

---

# **python-neutronclient Documentation**

*Release 7.3.2.dev1*

**Neutron Contributors**

**Mar 06, 2024**



# CONTENTS

<b>1</b>	<b>User Documentation</b>	<b>3</b>
1.1	Using CLI	3
1.1.1	OpenStackClient	3
	Advanced Network Commands in OpenStack Client	3
	BGP Dynamic Routing	3
	bgp speaker create	3
	bgp speaker delete	5
	bgp speaker list	5
	bgp speaker set	6
	bgp speaker show	7
	bgp speaker show dragents	7
	bgp speaker add network	8
	bgp speaker remove network	9
	bgp speaker add peer	9
	bgp speaker remove peer	9
	bgp speaker list advertised routes	9
	bgp peer create	10
	bgp peer delete	12
	bgp peer list	12
	bgp peer set	13
	bgp peer show	13
	bgp dragent add speaker	14
	bgp dragent list	14
	bgp dragent remove speaker	15
	firewall group	15
	firewall group create	16
	firewall group delete	17
	firewall group list	17
	firewall group set	18
	firewall group show	20
	firewall group unset	20
	firewall group policy	21
	firewall group policy add rule	21
	firewall group policy create	22
	firewall group policy delete	23
	firewall group policy list	23
	firewall group policy remove rule	24
	firewall group policy set	24
	firewall group policy show	25

firewall group policy unset . . . . .	26
firewall group rule . . . . .	26
firewall group rule create . . . . .	27
firewall group rule delete . . . . .	29
firewall group rule list . . . . .	29
firewall group rule set . . . . .	30
firewall group rule show . . . . .	32
firewall group rule unset . . . . .	32
network log . . . . .	33
network loggable resources list . . . . .	33
network log create . . . . .	34
network log delete . . . . .	36
network log list . . . . .	36
network log set . . . . .	37
network log show . . . . .	37
network trunk . . . . .	38
network subport list . . . . .	38
network trunk create . . . . .	39
network trunk delete . . . . .	40
network trunk list . . . . .	41
network trunk set . . . . .	42
network trunk show . . . . .	42
network trunk unset . . . . .	43
bgpvpn . . . . .	43
bgpvpn create . . . . .	43
bgpvpn delete . . . . .	45
bgpvpn list . . . . .	45
bgpvpn set . . . . .	46
bgpvpn show . . . . .	47
bgpvpn unset . . . . .	48
bgpvpn network association create . . . . .	49
bgpvpn network association delete . . . . .	50
bgpvpn network association list . . . . .	50
bgpvpn network association show . . . . .	51
bgpvpn router association create . . . . .	52
bgpvpn router association delete . . . . .	53
bgpvpn router association list . . . . .	54
bgpvpn router association set . . . . .	55
bgpvpn router association show . . . . .	55
bgpvpn router association unset . . . . .	56
bgpvpn port association create . . . . .	56
bgpvpn port association delete . . . . .	58
bgpvpn port association list . . . . .	58
bgpvpn port association set . . . . .	59
bgpvpn port association show . . . . .	60
bgpvpn port association unset . . . . .	61
networking sfc . . . . .	61
sfc flow classifier create . . . . .	62
sfc flow classifier delete . . . . .	63
sfc flow classifier list . . . . .	63
sfc flow classifier set . . . . .	64

sfc flow classifier show . . . . .	65
sfc port chain create . . . . .	65
sfc port chain delete . . . . .	66
sfc port chain list . . . . .	67
sfc port chain set . . . . .	68
sfc port chain show . . . . .	68
sfc port chain unset . . . . .	69
sfc port pair create . . . . .	69
sfc port pair delete . . . . .	71
sfc port pair list . . . . .	71
sfc port pair set . . . . .	72
sfc port pair show . . . . .	72
sfc port pair group create . . . . .	73
sfc port pair group delete . . . . .	74
sfc port pair group list . . . . .	74
sfc port pair group set . . . . .	75
sfc port pair group show . . . . .	76
sfc port pair group unset . . . . .	76
sfc service graph create . . . . .	77
sfc service graph delete . . . . .	78
sfc service graph list . . . . .	78
sfc service graph set . . . . .	79
sfc service graph show . . . . .	79
network onboard subnets . . . . .	80
network onboard subnets . . . . .	80
VPN Endpoint Group . . . . .	80
vpn endpoint group create . . . . .	81
vpn endpoint group delete . . . . .	82
vpn endpoint group list . . . . .	82
vpn endpoint group set . . . . .	83
vpn endpoint group show . . . . .	83
VPN IKE Policy . . . . .	84
vpn ike policy create . . . . .	84
vpn ike policy delete . . . . .	86
vpn ike policy list . . . . .	86
vpn ike policy set . . . . .	87
vpn ike policy show . . . . .	88
VPN IPsec Policy . . . . .	88
vpn ipsec policy create . . . . .	89
vpn ipsec policy delete . . . . .	90
vpn ipsec policy list . . . . .	90
vpn ipsec policy set . . . . .	91
vpn ipsec policy show . . . . .	92
VPN IPsec Site Connection . . . . .	93
vpn ipsec site connection create . . . . .	93
vpn ipsec site connection delete . . . . .	95
vpn ipsec site connection list . . . . .	95
vpn ipsec site connection set . . . . .	96
vpn ipsec site connection show . . . . .	97
VPN Service . . . . .	98
vpn service create . . . . .	98

	vpn service delete . . . . .	99
	vpn service list . . . . .	99
	vpn service set . . . . .	100
	vpn service show . . . . .	101
1.1.2	neutron CLI . . . . .	102
	Using neutron CLI . . . . .	102
	Basic Usage . . . . .	102
	Using with os-client-config . . . . .	103
	Using with keystone token . . . . .	103
	Using noauth mode . . . . .	104
	Display options . . . . .	104
	Filtering . . . . .	104
	Changing displayed columns . . . . .	105
	Extra arguments for create/update operation . . . . .	106
	Key interpretation . . . . .	106
	Value interpretation . . . . .	106
	Specify a list value . . . . .	107
	Specify a dict value . . . . .	107
	Specify a list of dicts . . . . .	108
	Passing None as a value . . . . .	108
	Debugging . . . . .	108
	Display API-level communication . . . . .	108
	neutron CLI reference . . . . .	110
	neutron usage . . . . .	110
	neutron API v2.0 commands . . . . .	112
	address-scope-create . . . . .	112
	address-scope-delete . . . . .	113
	address-scope-list . . . . .	113
	address-scope-show . . . . .	115
	address-scope-update . . . . .	116
	agent-delete . . . . .	116
	agent-list . . . . .	116
	agent-show . . . . .	117
	agent-update . . . . .	118
	auto-allocated-topology-delete . . . . .	119
	auto-allocated-topology-show . . . . .	119
	availability-zone-list . . . . .	120
	bash-completion . . . . .	121
	bgp-dragent-list-hosting-speaker . . . . .	121
	bgp-dragent-speaker-add . . . . .	122
	bgp-dragent-speaker-remove . . . . .	122
	bgp-peer-create . . . . .	123
	bgp-peer-delete . . . . .	124
	bgp-peer-list . . . . .	124
	bgp-peer-show . . . . .	125
	bgp-peer-update . . . . .	126
	bgp-speaker-advertiseroute-list . . . . .	126
	bgp-speaker-create . . . . .	128
	bgp-speaker-delete . . . . .	129
	bgp-speaker-list . . . . .	129
	bgp-speaker-list-on-dragent . . . . .	130

bgp-speaker-network-add . . . . .	131
bgp-speaker-network-remove . . . . .	132
bgp-speaker-peer-add . . . . .	132
bgp-speaker-peer-remove . . . . .	132
bgp-speaker-show . . . . .	132
bgp-speaker-update . . . . .	133
dhcp-agent-list-hosting-net . . . . .	134
dhcp-agent-network-add . . . . .	135
dhcp-agent-network-remove . . . . .	135
ext-list . . . . .	135
ext-show . . . . .	136
firewall-create . . . . .	137
firewall-delete . . . . .	139
firewall-list . . . . .	139
firewall-policy-create . . . . .	140
firewall-policy-delete . . . . .	141
firewall-policy-insert-rule . . . . .	141
firewall-policy-list . . . . .	142
firewall-policy-remove-rule . . . . .	143
firewall-policy-show . . . . .	143
firewall-policy-update . . . . .	144
firewall-rule-create . . . . .	145
firewall-rule-delete . . . . .	146
firewall-rule-list . . . . .	147
firewall-rule-show . . . . .	148
firewall-rule-update . . . . .	149
firewall-show . . . . .	150
firewall-update . . . . .	151
flavor-associate . . . . .	151
flavor-create . . . . .	152
flavor-delete . . . . .	153
flavor-disassociate . . . . .	153
flavor-list . . . . .	153
flavor-profile-create . . . . .	154
flavor-profile-delete . . . . .	155
flavor-profile-list . . . . .	156
flavor-profile-show . . . . .	157
flavor-profile-update . . . . .	158
flavor-show . . . . .	158
flavor-update . . . . .	159
floatingip-associate . . . . .	160
floatingip-create . . . . .	160
floatingip-delete . . . . .	161
floatingip-disassociate . . . . .	162
floatingip-list . . . . .	162
floatingip-show . . . . .	163
ipsec-site-connection-create . . . . .	164
ipsec-site-connection-delete . . . . .	166
ipsec-site-connection-list . . . . .	166
ipsec-site-connection-show . . . . .	167
ipsec-site-connection-update . . . . .	168

l3-agent-list-hosting-router . . . . .	169
l3-agent-router-add . . . . .	170
l3-agent-router-remove . . . . .	171
lb-agent-hosting-pool . . . . .	171
lb-healthmonitor-associate . . . . .	172
lb-healthmonitor-create . . . . .	172
lb-healthmonitor-delete . . . . .	174
lb-healthmonitor-disassociate . . . . .	174
lb-healthmonitor-list . . . . .	174
lb-healthmonitor-show . . . . .	175
lb-healthmonitor-update . . . . .	176
lb-member-create . . . . .	177
lb-member-delete . . . . .	178
lb-member-list . . . . .	178
lb-member-show . . . . .	179
lb-member-update . . . . .	180
lb-pool-create . . . . .	180
lb-pool-delete . . . . .	182
lb-pool-list . . . . .	182
lb-pool-list-on-agent . . . . .	183
lb-pool-show . . . . .	184
lb-pool-stats . . . . .	185
lb-pool-update . . . . .	186
lb-vip-create . . . . .	186
lb-vip-delete . . . . .	188
lb-vip-list . . . . .	188
lb-vip-show . . . . .	189
lb-vip-update . . . . .	190
lbaas-agent-hosting-loadbalancer . . . . .	190
lbaas-healthmonitor-create . . . . .	191
lbaas-healthmonitor-delete . . . . .	193
lbaas-healthmonitor-list . . . . .	193
lbaas-healthmonitor-show . . . . .	194
lbaas-healthmonitor-update . . . . .	195
lbaas-l7policy-create . . . . .	196
lbaas-l7policy-delete . . . . .	197
lbaas-l7policy-list . . . . .	198
lbaas-l7policy-show . . . . .	199
lbaas-l7policy-update . . . . .	200
lbaas-l7rule-create . . . . .	200
lbaas-l7rule-delete . . . . .	202
lbaas-l7rule-list . . . . .	202
lbaas-l7rule-show . . . . .	203
lbaas-l7rule-update . . . . .	204
lbaas-listener-create . . . . .	205
lbaas-listener-delete . . . . .	206
lbaas-listener-list . . . . .	207
lbaas-listener-show . . . . .	208
lbaas-listener-update . . . . .	209
lbaas-loadbalancer-create . . . . .	209
lbaas-loadbalancer-delete . . . . .	211



lbaas-loadbalancer-list	211
lbaas-loadbalancer-list-on-agent	212
lbaas-loadbalancer-show	213
lbaas-loadbalancer-stats	214
lbaas-loadbalancer-status	215
lbaas-loadbalancer-update	215
lbaas-member-create	216
lbaas-member-delete	217
lbaas-member-list	217
lbaas-member-show	218
lbaas-member-update	219
lbaas-pool-create	220
lbaas-pool-delete	221
lbaas-pool-list	221
lbaas-pool-show	223
lbaas-pool-update	224
meter-label-create	224
meter-label-delete	225
meter-label-list	225
meter-label-rule-create	227
meter-label-rule-delete	228
meter-label-rule-list	228
meter-label-rule-show	229
meter-label-show	230
net-create	231
net-delete	233
net-external-list	233
net-ip-availability-list	235
net-ip-availability-show	236
net-list	237
net-list-on-dhcp-agent	239
net-show	241
net-update	242
port-create	243
port-delete	245
port-list	245
port-show	246
port-update	247
purge	248
qos-available-rule-types	248
qos-bandwidth-limit-rule-create	250
qos-bandwidth-limit-rule-delete	251
qos-bandwidth-limit-rule-list	251
qos-bandwidth-limit-rule-show	252
qos-bandwidth-limit-rule-update	253
qos-dscp-marking-rule-create	254
qos-dscp-marking-rule-delete	255
qos-dscp-marking-rule-list	255
qos-dscp-marking-rule-show	256
qos-dscp-marking-rule-update	257
qos-minimum-bandwidth-rule-create	258

qos-minimum-bandwidth-rule-delete . . . . .	259
qos-minimum-bandwidth-rule-list . . . . .	259
qos-minimum-bandwidth-rule-show . . . . .	260
qos-minimum-bandwidth-rule-update . . . . .	261
qos-policy-create . . . . .	262
qos-policy-delete . . . . .	263
qos-policy-list . . . . .	263
qos-policy-show . . . . .	264
qos-policy-update . . . . .	265
quota-default-show . . . . .	265
quota-delete . . . . .	266
quota-list . . . . .	266
quota-show . . . . .	267
quota-update . . . . .	268
rbac-create . . . . .	270
rbac-delete . . . . .	271
rbac-list . . . . .	271
rbac-show . . . . .	272
rbac-update . . . . .	273
router-create . . . . .	273
router-delete . . . . .	275
router-gateway-clear . . . . .	275
router-gateway-set . . . . .	275
router-interface-add . . . . .	276
router-interface-delete . . . . .	276
router-list . . . . .	276
router-list-on-l3-agent . . . . .	278
router-port-list . . . . .	279
router-show . . . . .	280
router-update . . . . .	281
security-group-create . . . . .	281
security-group-delete . . . . .	282
security-group-list . . . . .	283
security-group-rule-create . . . . .	284
security-group-rule-delete . . . . .	285
security-group-rule-list . . . . .	286
security-group-rule-show . . . . .	287
security-group-show . . . . .	288
security-group-update . . . . .	289
service-provider-list . . . . .	289
subnet-create . . . . .	290
subnet-delete . . . . .	292
subnet-list . . . . .	292
subnet-show . . . . .	294
subnet-update . . . . .	295
subnetpool-create . . . . .	295
subnetpool-delete . . . . .	297
subnetpool-list . . . . .	297
subnetpool-show . . . . .	298
subnetpool-update . . . . .	299
tag-add . . . . .	300

	tag-remove . . . . .	300
	tag-replace . . . . .	301
	vpn-endpoint-group-create . . . . .	301
	vpn-endpoint-group-delete . . . . .	302
	vpn-endpoint-group-list . . . . .	303
	vpn-endpoint-group-show . . . . .	304
	vpn-endpoint-group-update . . . . .	305
	vpn-ikepolicy-create . . . . .	305
	vpn-ikepolicy-delete . . . . .	306
	vpn-ikepolicy-list . . . . .	307
	vpn-ikepolicy-show . . . . .	308
	vpn-ikepolicy-update . . . . .	309
	vpn-ipsecpolicy-create . . . . .	310
	vpn-ipsecpolicy-delete . . . . .	311
	vpn-ipsecpolicy-list . . . . .	311
	vpn-ipsecpolicy-show . . . . .	312
	vpn-ipsecpolicy-update . . . . .	313
	vpn-service-create . . . . .	314
	vpn-service-delete . . . . .	315
	vpn-service-list . . . . .	315
	vpn-service-show . . . . .	317
	vpn-service-update . . . . .	318
1.2	neutronclient Python API . . . . .	318
1.2.1	Basic Usage . . . . .	318
<b>2</b>	<b>Contributor Guide</b> . . . . .	<b>321</b>
2.1	Contributor Guide . . . . .	321
2.1.1	Client command extension support . . . . .	321
	Minimum requirements from an extension . . . . .	321
	Minimum requirements to use canonical neutron CRUD commands framework . . . . .	321
	Precedence of command loading . . . . .	322
	neutronclient.extension entry_point . . . . .	322
2.1.2	CLI Option Guideline . . . . .	322
	General conventions . . . . .	322
	Options for boolean value . . . . .	323
	Options for dict value . . . . .	323
	Options for list value . . . . .	323
	Compatibility with extra arguments . . . . .	324
	Background . . . . .	324
	Use at least one required option . . . . .	324
	Options for Boolean value . . . . .	325
	Avoid using nargs in positional or optional arguments . . . . .	325
2.1.3	Transition to OpenStack Client . . . . .	326
	Overview . . . . .	327
	Transition Steps . . . . .	327
	Developer Guide . . . . .	328
	Developer References . . . . .	329
<b>3</b>	<b>History</b> . . . . .	<b>331</b>



This is a client for OpenStack Networking API. It provides *Python API bindings* (the neutronclient module) and *command-line interface (CLI)*.

There are two CLIs which support the Networking API: *neutron CLI* and *OpenStack Client (OSC)*. OpenStack Client provides the basic network commands and python-neutronclient provides extensions (aka OSC plugins) for advanced networking services.



## USER DOCUMENTATION

### 1.1 Using CLI

There are two CLIs which support the Networking API: `OpenStackClient (OSC)` and `neutron CLI` (deprecated).

#### 1.1.1 OpenStackClient

OpenStackClient provides [the basic network commands](#) and `python-neutronclient` provides *extensions* (aka OSC plugins) for advanced networking services.

#### Advanced Network Commands in OpenStack Client

The following list covers the extended commands for advanced network services available in `openstack` command.

These commands can be referenced by doing `openstack help` and the detail of individual command can be referred by `openstack help <command-name>`.

#### BGP Dynamic Routing

BGP dynamic routing enables announcement of project subnet prefixes via BGP. Admins create BGP speakers and BGP peers. BGP peers can be associated with BGP speakers, thereby enabling peering sessions with operator infrastructure. BGP speakers can be associated with networks, which controls which routes are announced to peers.

Network v2

#### `bgp speaker create`

Create a BGP speaker

```
openstack bgp speaker create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
```

(continues on next page)

(continued from previous page)

```
[--fit-width]
[--print-empty]
--local-as <local-as>
[--ip-version {4,6}]
[--advertise-floating-ip-host-routes]
[--no-advertise-floating-ip-host-routes]
[--advertise-tenant-networks]
[--no-advertise-tenant-networks]
[--project <project>]
[--project-domain <project-domain>]
<name>
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- local-as** <local-as>  
Local AS number. (Integer in [1, 4294967295] is allowed.)
- ip-version** <IP\_VERSION>  
IP version for the BGP speaker (default is 4)
- advertise-floating-ip-host-routes**  
Enable the advertisement of floating IP host routes by the BGP speaker. (default)
- no-advertise-floating-ip-host-routes**  
Disable the advertisement of floating IP host routes by the BGP speaker.
- advertise-tenant-networks**  
Enable the advertisement of tenant network routes by the BGP speaker. (default)
- no-advertise-tenant-networks**  
Disable the advertisement of tenant network routes by the BGP speaker.
- project** <project>  
Owners project (name or ID)
- project-domain** <project-domain>  
Domain the project belongs to (name or ID). This can be used in case collisions between project



names exist.

#### **name**

Name of the BGP speaker to create

This command is provided by the python-neutronclient plugin.

### **bgp speaker delete**

Delete a BGP speaker

```
openstack bgp speaker delete <bgp-speaker>
```

#### **bgp-speaker**

BGP speaker to delete (name or ID)

This command is provided by the python-neutronclient plugin.

### **bgp speaker list**

List BGP speakers

```
openstack bgp speaker list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [--agent <agent-id>]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- agent** <agent-id>  
List BGP speakers hosted by an agent (ID only)

This command is provided by the python-neutronclient plugin.

## bgp speaker set

Set BGP speaker properties

```
openstack bgp speaker set
  [--name NAME]
  [--advertise-floating-ip-host-routes]
  [--no-advertise-floating-ip-host-routes]
  [--advertise-tenant-networks]
  [--no-advertise-tenant-networks]
  <bgp-speaker>
```

- name** <NAME>  
New name for the BGP speaker
  - advertise-floating-ip-host-routes**  
Enable the advertisement of floating IP host routes by the BGP speaker. (default)
  - no-advertise-floating-ip-host-routes**  
Disable the advertisement of floating IP host routes by the BGP speaker.
  - advertise-tenant-networks**  
Enable the advertisement of tenant network routes by the BGP speaker. (default)
  - no-advertise-tenant-networks**  
Disable the advertisement of tenant network routes by the BGP speaker.
- bgp-speaker**  
BGP speaker to update (name or ID)

This command is provided by the python-neutronclient plugin.

## bgp speaker show

Show a BGP speaker

```

openstack bgp speaker show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  <bgp-speaker>

```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**bgp-speaker**  
BGP speaker to display (name or ID)

This command is provided by the python-neutronclient plugin.

## bgp speaker show dragents

(Deprecated) List dynamic routing agents hosting a BGP speaker

(Use bgp dragent list instead)

```

openstack bgp speaker show dragents
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]

```

(continues on next page)

(continued from previous page)

```
[--sort-column SORT_COLUMN]
[--sort-ascending | --sort-descending]
<bgp-speaker>
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric
- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- bgp-speaker**  
List dynamic routing agents hosting a BGP speaker (name or ID)

This command is provided by the python-neutronclient plugin.

### bgp speaker add network

Add a network to a BGP speaker

```
openstack bgp speaker add network <bgp-speaker> <network>
```

**bgp-speaker**  
BGP speaker (name or ID)

**network**  
Network to add (name or ID)

This command is provided by the python-neutronclient plugin.

## bgp speaker remove network

Remove a network from a BGP speaker

```
openstack bgp speaker remove network <bgp-speaker> <network>
```

### **bgp-speaker**

BGP speaker (name or ID)

### **network**

Network to remove (name or ID)

This command is provided by the python-neutronclient plugin.

## bgp speaker add peer

Add a peer to a BGP speaker

```
openstack bgp speaker add peer <bgp-speaker> <bgp-peer>
```

### **bgp-speaker**

BGP speaker (name or ID)

### **bgp-peer**

BGP Peer to add (name or ID)

This command is provided by the python-neutronclient plugin.

## bgp speaker remove peer

Remove a peer from a BGP speaker

```
openstack bgp speaker remove peer <bgp-speaker> <bgp-peer>
```

### **bgp-speaker**

BGP speaker (name or ID)

### **bgp-peer**

BGP Peer to remove (name or ID)

This command is provided by the python-neutronclient plugin.

## bgp speaker list advertised routes

List routes advertised

```
openstack bgp speaker list advertised routes
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
```

(continues on next page)

(continued from previous page)

```
[--fit-width]
[--print-empty]
[--sort-column SORT_COLUMN]
[--sort-ascending | --sort-descending]
<bgp-speaker>
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric
- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- bgp-speaker**  
BGP speaker (name or ID)

This command is provided by the python-neutronclient plugin.

## bgp peer create

Create a BGP peer

```
openstack bgp peer create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
```

(continues on next page)

(continued from previous page)

```

--peer-ip <peer-ip-address>
--remote-as <peer-remote-as>
[--auth-type <peer-auth-type>]
[--password <auth-password>]
[--project <project>]
[--project-domain <project-domain>]
<name>

```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- peer-ip** <peer-ip-address>  
Peer IP address
- remote-as** <peer-remote-as>  
Peer AS number. (Integer in [1, 4294967295] is allowed)
- auth-type** <peer-auth-type>  
Authentication algorithm. Supported algorithms: none (default), md5
- password** <auth-password>  
Authentication password
- project** <project>  
Owners project (name or ID)
- project-domain** <project-domain>  
Domain the project belongs to (name or ID). This can be used in case collisions between project names exist.
- name**  
Name of the BGP peer to create

This command is provided by the python-neutronclient plugin.

## bgp peer delete

Delete a BGP peer

```
openstack bgp peer delete <bgp-peer>
```

### bgp-peer

BGP peer to delete (name or ID)

This command is provided by the python-neutronclient plugin.

## bgp peer list

List BGP peers

```
openstack bgp peer list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

**--sort-ascending**  
sort the column(s) in ascending order



**--sort-descending**

sort the column(s) in descending order

This command is provided by the python-neutronclient plugin.

**bgp peer set**

Update a BGP peer

```
openstack bgp peer set
  [--name NAME]
  [--password <auth-password>]
  <bgp-peer>
```

**--name <NAME>**

Updated name of the BGP peer

**--password <auth-password>**

Updated authentication password

**bgp-peer**

BGP peer to update (name or ID)

This command is provided by the python-neutronclient plugin.

**bgp peer show**

Show information for a BGP peer

```
openstack bgp peer show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  <bgp-peer>
```

**-f <FORMATTER>, --format <FORMATTER>**

the output format, defaults to table

**-c COLUMN, --column COLUMN**

specify the column(s) to include, can be repeated to show multiple columns

**--noindent**

whether to disable indenting the JSON

**--prefix <PREFIX>**

add a prefix to all variable names

**--max-width <integer>**

Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**bgp-peer**

BGP peer to display (name or ID)

This command is provided by the python-neutronclient plugin.

## bgp dragent add speaker

Add a BGP speaker to a dynamic routing agent

```
openstack bgp dragent add speaker <agent-id> <bgp-speaker>
```

**agent-id**

ID of the dynamic routing agent

**bgp-speaker**

ID or name of the BGP speaker

This command is provided by the python-neutronclient plugin.

## bgp dragent list

List dynamic routing agents

```
openstack bgp dragent list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [--bgp-speaker <bgp-speaker>]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- bgp-speaker** <bgp-speaker>  
List dynamic routing agents hosting a BGP speaker (name or ID)

This command is provided by the python-neutronclient plugin.

### bgp dragent remove speaker

Removes a BGP speaker from a dynamic routing agent

```
openstack bgp dragent remove speaker <agent-id> <bgp-speaker>
```

**agent-id**  
ID of the dynamic routing agent

**bgp-speaker**  
ID or name of the BGP speaker

This command is provided by the python-neutronclient plugin.

### firewall group

A **firewall group** is a perimeter firewall management to Networking. Firewall group uses iptables to apply firewall policy to all VM ports and router ports within a project.

Network v2

## firewall group create

Create a new firewall group

```
openstack firewall group create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--name NAME]
  [--description <description>]
  [--ingress-firewall-policy <ingress-firewall-policy> | --no-ingress-
  ↪firewall-policy]
  [--egress-firewall-policy <egress-firewall-policy> | --no-egress-
  ↪firewall-policy]
  [--share | --no-share]
  [--enable | --disable]
  [--project <project>]
  [--project-domain <project-domain>]
  [--port <port> | --no-port]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--name** <NAME>  
Name for the firewall group

**--description** <description>  
Description of the firewall group

**--ingress-firewall-policy** <ingress-firewall-policy>  
Ingress firewall policy (name or ID)

**--no-ingress-firewall-policy**  
Detach ingress firewall policy from the firewall group

- egress-firewall-policy** <egress-firewall-policy>  
Egress firewall policy (name or ID)
- no-egress-firewall-policy**  
Detach egress firewall policy from the firewall group
- share**  
Share the firewall group to be used in all projects (by default, it is restricted to be used by the current project).
- no-share**  
Restrict use of the firewall group to the current project
- enable**  
Enable firewall group
- disable**  
Disable firewall group
- project** <project>  
Owners project (name or ID)
- project-domain** <project-domain>  
Domain the project belongs to (name or ID). This can be used in case collisions between project names exist.
- port** <port>  
Port(s) (name or ID) to apply firewall group. This option can be repeated
- no-port**  
Detach all port from the firewall group

This command is provided by the python-neutronclient plugin.

## firewall group delete

Delete firewall group(s)

```
openstack firewall group delete <firewall-group> [<firewall-group> ...]
```

### firewall-group

Firewall group(s) to delete (name or ID)

This command is provided by the python-neutronclient plugin.

## firewall group list

List firewall groups

```
openstack firewall group list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
```

(continues on next page)

(continued from previous page)

```

[--fit-width]
[--print-empty]
[--sort-column SORT_COLUMN]
[--sort-ascending | --sort-descending]
[--long]

```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric
- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- long**  
List additional fields in output

This command is provided by the python-neutronclient plugin.

## firewall group set

Set firewall group properties

```

openstack firewall group set
  [--name NAME]
  [--description <description>]
  [--ingress-firewall-policy <ingress-firewall-policy> | --no-ingress-
↪firewall-policy]
  [--egress-firewall-policy <egress-firewall-policy> | --no-egress-
↪firewall-policy]
  [--share | --no-share]

```

(continues on next page)

(continued from previous page)

```
[--enable | --disable]
[--port <port>]
[--no-port]
<firewall-group>
```

**--name** <NAME>

Name for the firewall group

**--description** <description>

Description of the firewall group

**--ingress-firewall-policy** <ingress-firewall-policy>

Ingress firewall policy (name or ID)

**--no-ingress-firewall-policy**

Detach ingress firewall policy from the firewall group

**--egress-firewall-policy** <egress-firewall-policy>

Egress firewall policy (name or ID)

**--no-egress-firewall-policy**

Detach egress firewall policy from the firewall group

**--share**

Share the firewall group to be used in all projects (by default, it is restricted to be used by the current project).

**--no-share**

Restrict use of the firewall group to the current project

**--enable**

Enable firewall group

**--disable**

Disable firewall group

**--port** <port>

Port(s) (name or ID) to apply firewall group. This option can be repeated

**--no-port**

Detach all port from the firewall group

**firewall-group**

Firewall group to update (name or ID)

This command is provided by the python-neutronclient plugin.

## firewall group show

Display firewall group details

```
openstack firewall group show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  <firewall-group>
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**firewall-group**  
Firewall group to show (name or ID)

This command is provided by the python-neutronclient plugin.

## firewall group unset

Unset firewall group properties

```
openstack firewall group unset
  [--port <port> | --all-port]
  [--ingress-firewall-policy]
  [--egress-firewall-policy]
  [--share]
  [--enable]
  <firewall-group>
```

**--port** <port>  
Port(s) (name or ID) to apply firewall group. This option can be repeated



- all-port**  
Remove all ports for this firewall group
- ingress-firewall-policy**  
Ingress firewall policy (name or ID) to delete
- egress-firewall-policy**  
Egress firewall policy (name or ID) to delete
- share**  
Restrict use of the firewall group to the current project
- enable**  
Disable firewall group
- firewall-group**  
Firewall group to unset (name or ID)

This command is provided by the python-neutronclient plugin.

## firewall group policy

A **firewall group policy** is an ordered collection of firewall rules. A firewall policy can be shared across projects. Thus it can also be made part of an audit workflow wherein the `firewall_policy` can be audited by the relevant entity that is authorized (and can be different from the projects which create or use the firewall group policy).

Network v2

## firewall group policy add rule

Insert a rule into a given firewall policy

```
openstack firewall group policy add rule
  [--insert-before <firewall-rule>]
  [--insert-after <firewall-rule>]
  <firewall-policy>
  <firewall-rule>
```

- insert-before <firewall-rule>**  
Insert the new rule before this existing rule (name or ID)
- insert-after <firewall-rule>**  
Insert the new rule after this existing rule (name or ID)
- firewall-policy**  
Firewall policy to insert rule (name or ID)
- firewall-rule**  
Firewall rule to be inserted (name or ID)

This command is provided by the python-neutronclient plugin.

## firewall group policy create

Create a new firewall policy

```
openstack firewall group policy create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--description DESCRIPTION]
  [--audited | --no-audited]
  [--share | --no-share]
  [--project <project>]
  [--project-domain <project-domain>]
  [--firewall-rule <firewall-rule> | --no-firewall-rule]
  <name>
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--description** <DESCRIPTION>  
Description of the firewall policy

**--audited**  
Enable auditing for the policy

**--no-audited**  
Disable auditing for the policy

**--share**  
Share the firewall policy to be used in all projects (by default, it is restricted to be used by the current project).

**--no-share**  
Restrict use of the firewall policy to the current project

- project** <project>  
Owners project (name or ID)
- project-domain** <project-domain>  
Domain the project belongs to (name or ID). This can be used in case collisions between project names exist.
- firewall-rule** <firewall-rule>  
Firewall rule(s) to apply (name or ID)
- no-firewall-rule**  
Unset all firewall rules from firewall policy
- name**  
Name for the firewall policy

This command is provided by the python-neutronclient plugin.

### firewall group policy delete

Delete firewall policy(s)

```
openstack firewall group policy delete
  <firewall-policy>
  [<firewall-policy> ...]
```

- firewall-policy**  
Firewall policy(s) to delete (name or ID)

This command is provided by the python-neutronclient plugin.

### firewall group policy list

List firewall policies

```
openstack firewall group policy list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [--long]
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**

whether to disable indenting the JSON

**--max-width** <integer>

Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN

specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

**--sort-ascending**

sort the column(s) in ascending order

**--sort-descending**

sort the column(s) in descending order

**--long**

List additional fields in output

This command is provided by the python-neutronclient plugin.

## firewall group policy remove rule

Remove a rule from a given firewall policy

```
openstack firewall group policy remove rule
  <firewall-policy>
  <firewall-rule>
```

**firewall-policy**

Firewall policy to remove rule (name or ID)

**firewall-rule**

Firewall rule to remove from policy (name or ID)

This command is provided by the python-neutronclient plugin.

## firewall group policy set

Set firewall policy properties

```
openstack firewall group policy set
  [--description DESCRIPTION]
  [--audited | --no-audited]
  [--share | --no-share]
  [--name <name>]
  [--firewall-rule <firewall-rule>]
```

(continues on next page)

(continued from previous page)

```
[--no-firewall-rule]
<firewall-policy>
```

**--description** <DESCRIPTION>  
Description of the firewall policy

**--audited**  
Enable auditing for the policy

**--no-audited**  
Disable auditing for the policy

**--share**  
Share the firewall policy to be used in all projects (by default, it is restricted to be used by the current project).

**--no-share**  
Restrict use of the firewall policy to the current project

**--name** <name>  
Name for the firewall policy

**--firewall-rule** <firewall-rule>  
Firewall rule(s) to apply (name or ID)

**--no-firewall-rule**  
Remove all firewall rules from firewall policy

**firewall-policy**  
Firewall policy to update (name or ID)

This command is provided by the python-neutronclient plugin.

## firewall group policy show

Display firewall policy details

```
openstack firewall group policy show
[-f {json,shell,table,value,yaml}]
[-c COLUMN]
[--noindent]
[--prefix PREFIX]
[--max-width <integer>]
[--fit-width]
[--print-empty]
<firewall-policy>
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>

add a prefix to all variable names

**--max-width** <integer>

Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**firewall-policy**

Firewall policy to show (name or ID)

This command is provided by the python-neutronclient plugin.

## firewall group policy unset

Unset firewall policy properties

```
openstack firewall group policy unset
  [--firewall-rule <firewall-rule> | --all-firewall-rule]
  [--audited]
  [--share]
  <firewall-policy>
```

**--firewall-rule** <firewall-rule>

Remove firewall rule(s) from the firewall policy (name or ID)

**--all-firewall-rule**

Remove all firewall rules from the firewall policy

**--audited**

Disable auditing for the policy

**--share**

Restrict use of the firewall policy to the current project

**firewall-policy**

Firewall policy to unset (name or ID)

This command is provided by the python-neutronclient plugin.

## firewall group rule

A **firewall group rule** represents a collection of attributes like ports, IP addresses which define match criteria and action (allow, or deny) that needs to be taken on the matched data traffic.

Network v2

## firewall group rule create

Create a new firewall rule

```

openstack firewall group rule create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--name <name>]
  [--description <description>]
  [--protocol {tcp,udp,icmp,any}]
  [--action {allow,deny,reject}]
  [--ip-version <ip-version>]
  [--source-ip-address <source-ip-address> | --no-source-ip-address]
  [--destination-ip-address <destination-ip-address> | --no-destination-
↪ip-address]
  [--source-port <source-port> | --no-source-port]
  [--destination-port <destination-port> | --no-destination-port]
  [--share | --no-share]
  [--enable-rule | --disable-rule]
  [--source-firewall-group <source-firewall-group> | --no-source-
↪firewall-group]
  [--destination-firewall-group <destination-firewall-group> | --no-
↪destination-firewall-group]
  [--project <project>]
  [--project-domain <project-domain>]

```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--name** <name>  
Name of the firewall rule

**--description** <description>

Description of the firewall rule

- protocol** <PROTOCOL>  
Protocol for the firewall rule
- action** <ACTION>  
Action for the firewall rule
- ip-version** <ip-version>  
Set IP version 4 or 6 (default is 4)
- source-ip-address** <source-ip-address>  
Source IP address or subnet
- no-source-ip-address**  
Detach source IP address
- destination-ip-address** <destination-ip-address>  
Destination IP address or subnet
- no-destination-ip-address**  
Detach destination IP address
- source-port** <source-port>  
Source port number or range(integer in [1, 65535] or range like 123:456)
- no-source-port**  
Detach source port number or range
- destination-port** <destination-port>  
Destination port number or range(integer in [1, 65535] or range like 123:456)
- no-destination-port**  
Detach destination port number or range
- share**  
Share the firewall rule to be used in all projects (by default, it is restricted to be used by the current project).
- no-share**  
Restrict use of the firewall rule to the current project
- enable-rule**  
Enable this rule (default is enabled)
- disable-rule**  
Disable this rule
- source-firewall-group** <source-firewall-group>  
Source firewall group (name or ID)
- no-source-firewall-group**  
No associated destination firewall group
- destination-firewall-group** <destination-firewall-group>  
Destination firewall group (name or ID)
- no-destination-firewall-group**  
No associated destination firewall group



**--project** <project>

Owners project (name or ID)

**--project-domain** <project-domain>

Domain the project belongs to (name or ID). This can be used in case collisions between project names exist.

This command is provided by the python-neutronclient plugin.

## firewall group rule delete

Delete firewall rule(s)

```
openstack firewall group rule delete
  <firewall-rule>
  [<firewall-rule> ...]
```

### firewall-rule

Firewall rule(s) to delete (name or ID)

This command is provided by the python-neutronclient plugin.

## firewall group rule list

List firewall rules that belong to a given tenant

```
openstack firewall group rule list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [--long]
```

**-f** <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

**-c** COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>

when to include quotes, defaults to nonnumeric

**--noindent**

whether to disable indenting the JSON

**--max-width** <integer>

Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN

specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

**--sort-ascending**

sort the column(s) in ascending order

**--sort-descending**

sort the column(s) in descending order

**--long**

List additional fields in output

This command is provided by the python-neutronclient plugin.

## firewall group rule set

Set firewall rule properties

```
openstack firewall group rule set
  [--name <name>]
  [--description <description>]
  [--protocol {tcp,udp,icmp,any}]
  [--action {allow,deny,reject}]
  [--ip-version <ip-version>]
  [--source-ip-address <source-ip-address> | --no-source-ip-address]
  [--destination-ip-address <destination-ip-address> | --no-destination-
  ↪ip-address]
  [--source-port <source-port> | --no-source-port]
  [--destination-port <destination-port> | --no-destination-port]
  [--share | --no-share]
  [--enable-rule | --disable-rule]
  [--source-firewall-group <source-firewall-group> | --no-source-
  ↪firewall-group]
  [--destination-firewall-group <destination-firewall-group> | --no-
  ↪destination-firewall-group]
  <firewall-rule>
```

**--name** <name>

Name of the firewall rule

**--description** <description>

Description of the firewall rule

**--protocol** <PROTOCOL>

Protocol for the firewall rule

**--action** <ACTION>

Action for the firewall rule

**--ip-version** <ip-version>  
Set IP version 4 or 6 (default is 4)

**--source-ip-address** <source-ip-address>  
Source IP address or subnet

**--no-source-ip-address**  
Detach source IP address

**--destination-ip-address** <destination-ip-address>  
Destination IP address or subnet

**--no-destination-ip-address**  
Detach destination IP address

**--source-port** <source-port>  
Source port number or range(integer in [1, 65535] or range like 123:456)

**--no-source-port**  
Detach source port number or range

**--destination-port** <destination-port>  
Destination port number or range(integer in [1, 65535] or range like 123:456)

**--no-destination-port**  
Detach destination port number or range

**--share**  
Share the firewall rule to be used in all projects (by default, it is restricted to be used by the current project).

**--no-share**  
Restrict use of the firewall rule to the current project

**--enable-rule**  
Enable this rule (default is enabled)

**--disable-rule**  
Disable this rule

**--source-firewall-group** <source-firewall-group>  
Source firewall group (name or ID)

**--no-source-firewall-group**  
No associated destination firewall group

**--destination-firewall-group** <destination-firewall-group>  
Destination firewall group (name or ID)

**--no-destination-firewall-group**  
No associated destination firewall group

**firewall-rule**  
Firewall rule to set (name or ID)

This command is provided by the python-neutronclient plugin.

## firewall group rule show

Display firewall rule details

```
openstack firewall group rule show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  <firewall-rule>
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**firewall-rule**  
Firewall rule to display (name or ID)

This command is provided by the python-neutronclient plugin.

## firewall group rule unset

Unset firewall rule properties

```
openstack firewall group rule unset
  [--source-ip-address]
  [--destination-ip-address]
  [--source-port]
  [--destination-port]
  [--share]
  [--enable-rule]
  [--source-firewall-group]
  [--destination-firewall-group]
  <firewall-rule>
```

- source-ip-address**  
Source IP address or subnet
- destination-ip-address**  
Destination IP address or subnet
- source-port**  
Source port number or range(integer in [1, 65535] or range like 123:456)
- destination-port**  
Destination port number or range(integer in [1, 65535] or range like 123:456)
- share**  
Restrict use of the firewall rule to the current project
- enable-rule**  
Disable this rule
- source-firewall-group**  
Source firewall group (name or ID)
- destination-firewall-group**  
Destination firewall group (name or ID)
- firewall-rule**  
Firewall rule to unset (name or ID)

This command is provided by the python-neutronclient plugin.

## network log

A **network log** is a container to group security groups or ports for logging. Specified resources can be logged via these event (ALL, ACCEPT or DROP).

Network v2

## network loggable resources list

List supported loggable resources

```
openstack network loggable resources list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [--long]
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

- quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric
- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- long**  
List additional fields in output

This command is provided by the python-neutronclient plugin.

## network log create

Create a new network log

```
openstack network log create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--description <description>]
  [--enable | --disable]
  [--project <project>]
  [--project-domain <project-domain>]
  [--event {ALL,ACCEPT,DROP}]
  --resource-type <resource-type>
  [--resource <resource>]
  [--target <target>]
  <name>
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- description** <description>  
Description of the network log
- enable**  
Enable this log (default is disabled)
- disable**  
Disable this log
- project** <project>  
Owners project (name or ID)
- project-domain** <project-domain>  
Domain the project belongs to (name or ID). This can be used in case collisions between project names exist.
- event** {ALL, ACCEPT, DROP}  
An event to store with log
- resource-type** <resource-type>  
Network log type(s). You can see supported type(s) with following command: \$ openstack network loggable resources list
- resource** <resource>  
Name or ID of resource (security group or firewall group) that used for logging. You can control for logging target combination with target option.
- target** <target>  
Port (name or ID) for logging. You can control for logging target combination with resource option.
- name**  
Name for the network log

This command is provided by the python-neutronclient plugin.

## network log delete

Delete network log(s)

```
openstack network log delete <network-log> [<network-log> ...]
```

### network-log

Network log(s) to delete (name or ID)

This command is provided by the python-neutronclient plugin.

## network log list

List network logs

```
openstack network log list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [--long]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

**--sort-ascending**  
sort the column(s) in ascending order



**--sort-descending**  
sort the column(s) in descending order

**--long**  
List additional fields in output

This command is provided by the python-neutronclient plugin.

## network log set

Set network log properties

```
openstack network log set
  [--description <description>]
  [--enable | --disable]
  [--name <name>]
  <network-log>
```

**--description** <description>  
Description of the network log

**--enable**  
Enable this log (default is disabled)

**--disable**  
Disable this log

**--name** <name>  
Name of the network log

**network-log**  
Network log to set (name or ID)

This command is provided by the python-neutronclient plugin.

## network log show

Display network log details

```
openstack network log show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  <network-log>
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**

whether to disable indenting the JSON

**--prefix** <PREFIX>

add a prefix to all variable names

**--max-width** <integer>

Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**network-log**

Network log to show (name or ID)

This command is provided by the python-neutronclient plugin.

## network trunk

A **network trunk** is a container to group logical ports from different networks and provide a single trunked vNIC for servers. It consists of one parent port which is a regular VIF and multiple subports which allow the server to connect to more networks.

Network v2

## network subport list

List all subports for a given network trunk

```
openstack network subport list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  --trunk <trunk>
```

**-f** <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

**-c** COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>

when to include quotes, defaults to nonnumeric

- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- trunk** <trunk>  
List subports belonging to this trunk (name or ID)

This command is provided by the python-neutronclient plugin.

## network trunk create

Create a network trunk for a given project

```
openstack network trunk create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--description <description>]
  --parent-port <parent-port>
  [--subport <port=,segmentation-type=,segmentation-id=>]
  [--enable | --disable]
  [--project <project>]
  [--project-domain <project-domain>]
  <name>
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON

- prefix** <PREFIX>  
add a prefix to all variable names
  - max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
  - fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
  - print-empty**  
Print empty table if there is no data to show.
  - description** <description>  
A description of the trunk
  - parent-port** <parent-port>  
Parent port belonging to this trunk (name or ID)
  - subport** <port=, segmentation-type=, segmentation-id=>  
Subport to add. Subport is of form port=<name or ID>,segmentation-type=,segmentation-ID= (subport) option can be repeated
  - enable**  
Enable trunk (default)
  - disable**  
Disable trunk
  - project** <project>  
Owners project (name or ID)
  - project-domain** <project-domain>  
Domain the project belongs to (name or ID). This can be used in case collisions between project names exist.
- name**  
Name of the trunk to create

This command is provided by the python-neutronclient plugin.

## network trunk delete

Delete a given network trunk

```
openstack network trunk delete <trunk> [<trunk> ...]
```

### trunk

Trunk(s) to delete (name or ID)

This command is provided by the python-neutronclient plugin.

## network trunk list

List all network trunks

```
openstack network trunk list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [--long]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

**--sort-ascending**  
sort the column(s) in ascending order

**--sort-descending**  
sort the column(s) in descending order

**--long**  
List additional fields in output

This command is provided by the python-neutronclient plugin.

## network trunk set

Set network trunk properties

```
openstack network trunk set
  [--name <name>]
  [--description <description>]
  [--subport <port=,segmentation-type=,segmentation-id=>]
  [--enable | --disable]
  <trunk>
```

**--name** <name>

Set trunk name

**--description** <description>

A description of the trunk

**--subport** <port=,segmentation-type=,segmentation-id=>

Subport to add. Subport is of form port=<name or ID>,segmentation-type=,segmentation-ID=(subport) option can be repeated

**--enable**

Enable trunk

**--disable**

Disable trunk

**trunk**

Trunk to modify (name or ID)

This command is provided by the python-neutronclient plugin.

## network trunk show

Show information of a given network trunk

```
openstack network trunk show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  <trunk>
```

**-f** <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

**-c** COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

**--noindent**

whether to disable indenting the JSON

**--prefix** <PREFIX>

add a prefix to all variable names

**--max-width** <integer>

Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**trunk**

Trunk to display (name or ID)

This command is provided by the python-neutronclient plugin.

## network trunk unset

Unset subports from a given network trunk

```
openstack network trunk unset --subport <subport> <trunk>
```

**--subport** <subport>

Subport to delete (name or ID of the port) (subport) option can be repeated

**trunk**

Unset subports from this trunk (name or ID)

This command is provided by the python-neutronclient plugin.

## bgpvpn

A **bgpvpn** resource contains a set of parameters to define a BGP-based VPN. BGP-based IP VPNs networks are widely used in the industry especially for enterprises. The networking BGP VPN project aims at supporting inter-connection between L3VPNs and Neutron resources, i.e. Networks, Routers and Ports.

Network v2

## bgpvpn create

Create BGP VPN resource

```
openstack bgpvpn create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--project <project>]
  [--project-domain <project-domain>]
```

(continues on next page)

(continued from previous page)

```
[--name <name>]
[--route-target <route-target>]
[--import-target <import-target>]
[--export-target <export-target>]
[--route-distinguisher <route-distinguisher>]
[--vni VNI]
[--local-pref LOCAL_PREF]
[--type {12,13}]
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- project** <project>  
Owners project (name or ID)
- project-domain** <project-domain>  
Domain the project belongs to (name or ID). This can be used in case collisions between project names exist.
- name** <name>  
Name of the BGP VPN
- route-target** <route-target>  
Add Route Target to import/export list (repeat option for multiple Route Targets)
- import-target** <import-target>  
Add Route Target to import list (repeat option for multiple Route Targets)
- export-target** <export-target>  
Add Route Target to export list (repeat option for multiple Route Targets)
- route-distinguisher** <route-distinguisher>  
Add Route Distinguisher to the list of Route Distinguishers from which a Route Distinguishers will be picked from to advertise a VPN route (repeat option for multiple Route Distinguishers)
- vni** <VNI>  
VXLAN Network Identifier to be used for this BGPVPN when a VXLAN encapsulation is used



**--local-pref** <LOCAL\_PREF>  
Default BGP LOCAL\_PREF to use in route advertisements towards this BGPVPN.

**--type** <TYPE>  
BGP VPN type selection between IP VPN (13) and Ethernet VPN (12) (default: 13)

This command is provided by the python-neutronclient plugin.

## bgpvpn delete

Delete BGP VPN resource(s)

```
openstack bgpvpn delete <bgpvpn> [<bgpvpn> ...]
```

### bgpvpn

BGP VPN(s) to delete (name or ID)

This command is provided by the python-neutronclient plugin.

## bgpvpn list

List BGP VPN resources

```
openstack bgpvpn list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [--project <project>]
  [--project-domain <project-domain>]
  [--long]
  [--property <key=value>]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN

specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

**--sort-ascending**

sort the column(s) in ascending order

**--sort-descending**

sort the column(s) in descending order

**--project** <project>

Owners project (name or ID)

**--project-domain** <project-domain>

Domain the project belongs to (name or ID). This can be used in case collisions between project names exist.

**--long**

List additional fields in output

**--property** <key=value>

Filter property to apply on returned BGP VPNs (repeat to filter on multiple properties)

This command is provided by the python-neutronclient plugin.

## bgpvpn set

Set BGP VPN properties

```
openstack bgpvpn set
  [--name <name>]
  [--route-target <route-target>]
  [--no-route-target]
  [--import-target <import-target>]
  [--no-import-target]
  [--export-target <export-target>]
  [--no-export-target]
  [--route-distinguisher <route-distinguisher>]
  [--no-route-distinguisher]
  [--vni VNI]
  [--local-pref LOCAL_PREF]
  <bgpvpn>
```

**--name** <name>

Name of the BGP VPN

**--route-target** <route-target>

Add Route Target to import/export list (repeat option for multiple Route Targets)

**--no-route-target**

Empty route target list

- import-target** <import-target>  
Add Route Target to import list (repeat option for multiple Route Targets)
  - no-import-target**  
Empty import route target list
  - export-target** <export-target>  
Add Route Target to export list (repeat option for multiple Route Targets)
  - no-export-target**  
Empty export route target list
  - route-distinguisher** <route-distinguisher>  
Add Route Distinguisher to the list of Route Distinguishers from which a Route Distinguishers will be picked from to advertise a VPN route (repeat option for multiple Route Distinguishers)
  - no-route-distinguisher**  
Empty route distinguisher list
  - vni** <VNI>  
VXLAN Network Identifier to be used for this BGPVPN when a VXLAN encapsulation is used
  - local-pref** <LOCAL\_PREF>  
Default BGP LOCAL\_PREF to use in route advertisementstowards this BGPVPN.
- bgpvpn**  
BGP VPN to update (name or ID)

This command is provided by the python-neutronclient plugin.

## bgpvpn show

Show information of a given BGP VPN

```
openstack bgpvpn show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  <bgpvpn>
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>

Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**bgpvpn**

BGP VPN to display (name or ID)

This command is provided by the python-neutronclient plugin.

### bgpvpn unset

Unset BGP VPN properties

```
openstack bgpvpn unset
  [--route-target <route-target>]
  [--all-route-target]
  [--import-target <import-target>]
  [--all-import-target]
  [--export-target <export-target>]
  [--all-export-target]
  [--route-distinguisher <route-distinguisher>]
  [--all-route-distinguisher]
  [--vni VNI]
  [--local-pref LOCAL_PREF]
  <bgpvpn>
```

**--route-target** <route-target>

Remove Route Target from import/export list (repeat option for multiple Route Targets)

**--all-route-target**

Empty route target list

**--import-target** <import-target>

Remove Route Target from import list (repeat option for multiple Route Targets)

**--all-import-target**

Empty import route target list

**--export-target** <export-target>

Remove Route Target from export list (repeat option for multiple Route Targets)

**--all-export-target**

Empty export route target list

**--route-distinguisher** <route-distinguisher>

Remove Route Distinguisher from the list of Route Distinguishers from which a Route Distinguishers will be picked from to advertise a VPN route (repeat option for multiple Route Distinguishers)

**--all-route-distinguisher**

Empty route distinguisher list

- vni** <VNI>  
VXLAN Network Identifier to be used for this BGPVPN when a VXLAN encapsulation is used
- local-pref** <LOCAL\_PREF>  
Default BGP LOCAL\_PREF to use in route advertisementstowards this BGPVPN.

**bgpvpn**  
BGP VPN to update (name or ID)

This command is provided by the python-neutronclient plugin.

## bgpvpn network association create

Create a BGP VPN network association

```
openstack bgpvpn network association create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--project <project>]
  [--project-domain <project-domain>]
  <bgpvpn>
  <network>
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- project** <project>  
Owners project (name or ID)
- project-domain** <project-domain>  
Domain the project belongs to (name or ID). This can be used in case collisions between project names exist.

### **bgpvpn**

BGP VPN to apply the network association (name or ID)

### **network**

Network to associate the BGP VPN (name or ID)

This command is provided by the python-neutronclient plugin.

## **bgpvpn network association delete**

Delete a BGP VPN network association(s) for a given BGP VPN

```
openstack bgpvpn network association delete
  <network
  association
  ID>
  [<network association ID> ...]
  <bgpvpn>
```

**network** association ID

Network association ID(s) to remove

### **bgpvpn**

BGP VPN the network association belongs to (name or ID)

This command is provided by the python-neutronclient plugin.

## **bgpvpn network association list**

List BGP VPN network associations for a given BGP VPN

```
openstack bgpvpn network association list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [--long]
  [--property <key=value>]
  <bgpvpn>
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- long**  
List additional fields in output
- property** <key=value>  
Filter property to apply on returned BGP VPNs (repeat to filter on multiple properties)

**bgpvpn**

BGP VPN listed associations belong to (name or ID)

This command is provided by the python-neutronclient plugin.

**bgpvpn network association show**

Show information of a given BGP VPN network association

```
openstack bgpvpn network association show
[-f {json,shell,table,value,yaml}]
[-c COLUMN]
[--noindent]
[--prefix PREFIX]
[--max-width <integer>]
[--fit-width]
[--print-empty]
<network
association
ID>
<bgpvpn>
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON

- prefix** <PREFIX>  
add a prefix to all variable names
  - max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
  - fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
  - print-empty**  
Print empty table if there is no data to show.
- network** association ID  
Network association ID to look up

**bgpvpn**  
BGP VPN the association belongs to (name or ID)

This command is provided by the python-neutronclient plugin.

### bgpvpn router association create

Create a BGP VPN router association

```
openstack bgpvpn router association create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--project <project>]
  [--project-domain <project-domain>]
  [--advertise_extra_routes | --no-advertise_extra_routes]
  <bgpvpn>
  <router>
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.



**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**--project** <project>

Owners project (name or ID)

**--project-domain** <project-domain>

Domain the project belongs to (name or ID). This can be used in case collisions between project names exist.

**--advertise\_extra\_routes**

Routes will be advertised to the BGP VPN (default)

**--no-advertise\_extra\_routes**

Routes from the router will not be advertised to the BGP VPN

**bgpvpn**

BGP VPN to apply the router association (name or ID)

**router**

Router to associate the BGP VPN (name or ID)

This command is provided by the python-neutronclient plugin.

### bgpvpn router association delete

Delete a BGP VPN router association(s) for a given BGP VPN

```
openstack bgpvpn router association delete
  <router
  association
  ID>
  [<router association ID> ...]
  <bgpvpn>
```

**router** association ID

Router association ID(s) to remove

**bgpvpn**

BGP VPN the router association belongs to (name or ID)

This command is provided by the python-neutronclient plugin.

## bgpvpn router association list

List BGP VPN router associations for a given BGP VPN

```
openstack bgpvpn router association list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [--long]
  [--property <key=value>]
  <bgpvpn>
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

**--sort-ascending**  
sort the column(s) in ascending order

**--sort-descending**  
sort the column(s) in descending order

**--long**  
List additional fields in output

**--property** <key=value>  
Filter property to apply on returned BGP VPNs (repeat to filter on multiple properties)

**bgpvpn**  
BGP VPN listed associations belong to (name or ID)

This command is provided by the python-neutronclient plugin.

### bgpvpn router association set

Set BGP VPN router association properties

```
openstack bgpvpn router association set
  [--advertise_extra_routes | --no-advertise_extra_routes]
  <router
  association
  ID>
  <bgpvpn>
```

#### **--advertise\_extra\_routes**

Routes will be advertised to the BGP VPN

#### **--no-advertise\_extra\_routes**

Routes from the router will not be advertised to the BGP VPN

**router** association ID

Router association ID to update

#### **bgpvpn**

BGP VPN the router association belongs to (name or ID)

This command is provided by the python-neutronclient plugin.

### bgpvpn router association show

Show information of a given BGP VPN router association

```
openstack bgpvpn router association show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  <router
  association
  ID>
  <bgpvpn>
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>

Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**router** association ID

Router association ID to look up

**bgpvpn**

BGP VPN the association belongs to (name or ID)

This command is provided by the python-neutronclient plugin.

### bgpvpn router association unset

Unset BGP VPN router association properties

```
openstack bgpvpn router association unset
  [--advertise_extra_routes | --no-advertise_extra_routes]
  <router
  association
  ID>
  <bgpvpn>
```

**--advertise\_extra\_routes**

Routes from the router will not be advertised to the BGP VPN

**--no-advertise\_extra\_routes**

Routes will be advertised to the BGP VPN

**router** association ID

Router association ID to update

**bgpvpn**

BGP VPN the router association belongs to (name or ID)

This command is provided by the python-neutronclient plugin.

### bgpvpn port association create

Create a BGP VPN port association

```
openstack bgpvpn port association create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
```

(continues on next page)

(continued from previous page)

```

[--print-empty]
[--project <project>]
[--project-domain <project-domain>]
[--advertise-fixed-ips | --no-advertise-fixed-ips]
[--prefix-route prefix=<cidr>[,local_pref=<integer>]]
[--bgpvpn-route bgpvpn=<BGP VPN ID or name>[,local_pref=<integer>]]
<bgpvpn>
<port>

```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- project** <project>  
Owners project (name or ID)
- project-domain** <project-domain>  
Domain the project belongs to (name or ID). This can be used in case collisions between project names exist.
- advertise-fixed-ips**  
Fixed IPs of the port will be advertised to the BGP VPN (default)
- no-advertise-fixed-ips**  
Fixed IPs of the port will not be advertised to the BGP VPN
- prefix-route** **prefix**=<cidr>[,**local\_pref**=<integer>]  
Add prefix route in CIDR notation. Optionally, can control the value of the BGP LOCAL\_PREF of the routes that will be advertised (repeat option for multiple prefix routes)
- bgpvpn-route** **bgpvpn**=<BGP VPN ID or name>[,**local\_pref**=<integer>]  
Add BGP VPN route for route leaking. Optionally, can control the value of the BGP LOCAL\_PREF of the routes that will be advertised (repeat option for multiple BGP VPN routes)
- bgpvpn**  
BGP VPN to apply the port association (name or ID)
- port**  
Port to associate the BGP VPN (name or ID)

This command is provided by the python-neutronclient plugin.

### bgpvpn port association delete

Delete a BGP VPN port association(s) for a given BGP VPN

```
openstack bgpvpn port association delete
  <port
  association
  ID>
  [<port association ID> ...]
  <bgpvpn>
```

**port** association ID  
Port association ID(s) to remove

**bgpvpn**  
BGP VPN the port association belongs to (name or ID)

This command is provided by the python-neutronclient plugin.

### bgpvpn port association list

List BGP VPN port associations for a given BGP VPN

```
openstack bgpvpn port association list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [--long]
  [--property <key=value>]
  <bgpvpn>
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN

specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

**--sort-ascending**

sort the column(s) in ascending order

**--sort-descending**

sort the column(s) in descending order

**--long**

List additional fields in output

**--property** <key=value>

Filter property to apply on returned BGP VPNs (repeat to filter on multiple properties)

**bgpvpn**

BGP VPN listed associations belong to (name or ID)

This command is provided by the python-neutronclient plugin.

**bgpvpn port association set**

Set BGP VPN port association properties

```
openstack bgpvpn port association set
  [--advertise-fixed-ips | --no-advertise-fixed-ips]
  [--prefix-route prefix=<cidr>[,local_pref=<integer>]]
  [--bgpvpn-route bgpvpn=<BGP VPN ID or name>[,local_pref=<integer>]]
  [--no-prefix-route]
  [--no-bgpvpn-route]
  <port
  association
  ID>
  <bgpvpn>
```

**--advertise-fixed-ips**

Fixed IPs of the port will be advertised to the BGP VPN

**--no-advertise-fixed-ips**

Fixed IPs of the port will not be advertised to the BGP VPN

**--prefix-route** prefix=<cidr>[,local\_pref=<integer>]

Add prefix route in CIDR notation. Optionally, can control the value of the BGP LOCAL\_PREF of the routes that will be advertised (repeat option for multiple prefix routes)

**--bgpvpn-route** bgpvpn=<BGP VPN ID or name>[,local\_pref=<integer>]

Add BGP VPN route for route leaking. Optionally, can control the value of the BGP LOCAL\_PREF of the routes that will be advertised (repeat option for multiple BGP VPN routes)

**--no-prefix-route**  
Empty prefix route list

**--no-bgpvpn-route**  
Empty BGP VPN route list

**port** association ID  
Port association ID to update

**bgpvpn**  
BGP VPN the port association belongs to (name or ID)

This command is provided by the python-neutronclient plugin.

## bgpvpn port association show

Show information of a given BGP VPN port association

```
openstack bgpvpn port association show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  <port
  association
  ID>
  <bgpvpn>
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**port** association ID  
Port association ID to look up



**bgpvpn**

BGP VPN the association belongs to (name or ID)

This command is provided by the python-neutronclient plugin.

**bgpvpn port association unset**

Unset BGP VPN port association properties

```
openstack bgpvpn port association unset
  [--advertise-fixed-ips | --no-advertise-fixed-ips]
  [--prefix-route <cidr>]
  [--bgpvpn-route <BGP VPN ID or name>]
  [--all-prefix-routes]
  [--all-bgpvpn-routes]
  <port
  association
  ID>
  <bgpvpn>
```

**--advertise-fixed-ips**

Fixed IPs of the port will not be advertised to the BGP VPN

**--no-advertise-fixed-ips**

Fixed IPs of the port will be advertised to the BGP VPN

**--prefix-route <cidr>**

Remove prefix route in CIDR notation (repeat option for multiple prefix routes)

**--bgpvpn-route <BGP VPN ID or name>**

Remove BGP VPN route (repeat option for multiple BGP VPN routes)

**--all-prefix-routes**

Empty prefix route list

**--all-bgpvpn-routes**

Empty BGP VPN route list

**port association ID**

Port association ID to update

**bgpvpn**

BGP VPN the port association belongs to (name or ID)

This command is provided by the python-neutronclient plugin.

**networking sfc**

**Service Function Chaining** is a mechanism for overriding the basic destination based forwarding that is typical of IP networks. Service Function Chains consist of an ordered sequence of Service Functions (SFs). SFs are virtual machines (or potentially physical devices) that perform a network function such as firewall, content cache, packet inspection, or any other function that requires processing of packets in a flow from point A to point B even though the SFs are not literally between point A and B from a routing table perspective.

Network v2

## sfc flow classifier create

Create a flow classifier

```
openstack sfc flow classifier create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--description <description>]
  [--protocol <protocol>]
  [--ethertype {IPv4,IPv6}]
  [--source-port <min-port>:<max-port>]
  [--destination-port <min-port>:<max-port>]
  [--source-ip-prefix <source-ip-prefix>]
  [--destination-ip-prefix <destination-ip-prefix>]
  [--logical-source-port <logical-source-port>]
  [--logical-destination-port <logical-destination-port>]
  [--l7-parameters L7_PARAMETERS]
  <name>
```

**-f** <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

**-c** COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

**--noindent**

whether to disable indenting the JSON

**--prefix** <PREFIX>

add a prefix to all variable names

**--max-width** <integer>

Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**--description** <description>

Description for the flow classifier

**--protocol** <protocol>

IP protocol name. Protocol name should be as per IANA standard.

**--ethertype** {IPv4,IPv6}

L2 ethertype, default is IPv4

**--source-port** <min-port>:<max-port>

Source protocol port (allowed range [1,65535]). Must be specified as a:b, where a=min-port and

b=max-port) in the allowed range.

- destination-port** <min-port>:<max-port>  
Destination protocol port (allowed range [1,65535]. Must be specified as a:b, where a=min-port and b=max-port) in the allowed range.
- source-ip-prefix** <source-ip-prefix>  
Source IP address in CIDR notation
- destination-ip-prefix** <destination-ip-prefix>  
Destination IP address in CIDR notation
- logical-source-port** <logical-source-port>  
Neutron source port (name or ID)
- logical-destination-port** <logical-destination-port>  
Neutron destination port (name or ID)
- l7-parameters** <L7\_PARAMETERS>  
Dictionary of L7 parameters. Currently, no value is supported for this option.

**name**

Name of the flow classifier

This command is provided by the python-neutronclient plugin.

### sfc flow classifier delete

Delete a given flow classifier

```
openstack sfc flow classifier delete <flow-classifier>
```

**flow-classifier**

Flow classifier to delete (name or ID)

This command is provided by the python-neutronclient plugin.

### sfc flow classifier list

List flow classifiers

```
openstack sfc flow classifier list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [--long]
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric
- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- long**  
List additional fields in output

This command is provided by the python-neutronclient plugin.

## sfc flow classifier set

Set flow classifier properties

```
openstack sfc flow classifier set  
  [--name <name>]  
  [--description <description>]  
  <flow-classifier>
```

- name** <name>  
Name of the flow classifier
- description** <description>  
Description for the flow classifier
- flow-classifier**  
Flow classifier to modify (name or ID)

This command is provided by the python-neutronclient plugin.

## sfc flow classifier show

Display flow classifier details

```

openstack sfc flow classifier show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  <flow-classifier>

```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**flow-classifier**  
Flow classifier to display (name or ID)

This command is provided by the python-neutronclient plugin.

## sfc port chain create

Create a port chain

```

openstack sfc port chain create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--description <description>]
  [--flow-classifier <flow-classifier>]

```

(continues on next page)

(continued from previous page)

```
[--chain-parameters correlation=<correlation-type>, symmetric=<boolean>]
--port-pair-group <port-pair-group>
<name>
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- description** <description>  
Description for the port chain
- flow-classifier** <flow-classifier>  
Add flow classifier (name or ID). This option can be repeated.
- chain-parameters** correlation=<correlation-type>, symmetric=<boolean>  
Dictionary of chain parameters. Supports correlation=(mplslsh) (default is mpl) and symmetric=(truefalse).
- port-pair-group** <port-pair-group>  
Add port pair group (name or ID). This option can be repeated.
- name**  
Name of the port chain

This command is provided by the python-neutronclient plugin.

## sfc port chain delete

Delete a given port chain

```
openstack sfc port chain delete <port-chain>
```

### port-chain

Port chain to delete (name or ID)

This command is provided by the python-neutronclient plugin.

## sfc port chain list

List port chains

```
openstack sfc port chain list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [--long]
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric
- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- long**  
List additional fields in output

This command is provided by the python-neutronclient plugin.

## sfc port chain set

Set port chain properties

```
openstack sfc port chain set
  [--name <name>]
  [--description <description>]
  [--flow-classifier <flow-classifier>]
  [--no-flow-classifier]
  [--port-pair-group <port-pair-group>]
  [--no-port-pair-group]
  <port-chain>
```

**--name** <name>

Name of the port chain

**--description** <description>

Description for the port chain

**--flow-classifier** <flow-classifier>

Add flow classifier (name or ID). This option can be repeated.

**--no-flow-classifier**

Remove associated flow classifiers from the port chain

**--port-pair-group** <port-pair-group>

Add port pair group (name or ID). Current port pair groups order is kept, the added port pair group will be placed at the end of the port chain. This option can be repeated.

**--no-port-pair-group**

Remove associated port pair groups from the port chain. At least one port-pair-group must be specified together.

**port-chain**

Port chain to modify (name or ID)

This command is provided by the python-neutronclient plugin.

## sfc port chain show

Display port chain details

```
openstack sfc port chain show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  <port-chain>
```

**-f** <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

**-c** COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns



**--noindent**

whether to disable indenting the JSON

**--prefix** <PREFIX>

add a prefix to all variable names

**--max-width** <integer>

Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**port-chain**

Port chain to display (name or ID)

This command is provided by the python-neutronclient plugin.

**sfc port chain unset**

Unset port chain properties

```
openstack sfc port chain unset
  [--flow-classifier <flow-classifier> | --all-flow-classifier]
  [--port-pair-group <port-pair-group>]
  <port-chain>
```

**--flow-classifier** <flow-classifier>

Remove flow classifier(s) from the port chain (name or ID). This option can be repeated.

**--all-flow-classifier**

Remove all flow classifiers from the port chain

**--port-pair-group** <port-pair-group>

Remove port pair group(s) from the port chain (name or ID). This option can be repeated.

**port-chain**

Port chain to unset (name or ID)

This command is provided by the python-neutronclient plugin.

**sfc port pair create**

Create a port pair

```
openstack sfc port pair create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
```

(continues on next page)

(continued from previous page)

```

[--fit-width]
[--print-empty]
[--description <description>]
[--service-function-parameters correlation=<correlation-type>, weight=
↪<weight>]
--ingress <ingress>
--egress <egress>
<name>

```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--description** <description>  
Description for the port pair

**--service-function-parameters** **correlation**=<correlation-type>, **weight**=<weight>  
Dictionary of service function parameters. Currently, **correlation**=(None|mplslsh) and **weight** are supported. **Weight** is an integer that influences the selection of a port pair within a port pair group for a flow. The higher the weight, the more flows will hash to the port pair. The default weight is 1.

**--ingress** <ingress>  
Ingress neutron port (name or ID)

**--egress** <egress>  
Egress neutron port (name or ID)

**name**  
Name of the port pair

This command is provided by the python-neutronclient plugin.

## sfc port pair delete

Delete a given port pair

```
openstack sfc port pair delete <port-pair>
```

### port-pair

Port pair to delete (name or ID)

This command is provided by the python-neutronclient plugin.

## sfc port pair list

List port pairs

```
openstack sfc port pair list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [--long]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

**--sort-ascending**  
sort the column(s) in ascending order

**--sort-descending**

sort the column(s) in descending order

**--long**

List additional fields in output

This command is provided by the python-neutronclient plugin.

## sfc port pair set

Set port pair properties

```
openstack sfc port pair set
  [--name <name>]
  [--description <description>]
  <port-pair>
```

**--name** <name>

Name of the port pair

**--description** <description>

Description for the port pair

**port-pair**

Port pair to modify (name or ID)

This command is provided by the python-neutronclient plugin.

## sfc port pair show

Display port pair details

```
openstack sfc port pair show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  <port-pair>
```

**-f** <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

**-c** COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

**--noindent**

whether to disable indenting the JSON

**--prefix** <PREFIX>

add a prefix to all variable names

**--max-width** <integer>

Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**port-pair**

Port pair to display (name or ID)

This command is provided by the python-neutronclient plugin.

## sfc port pair group create

Create a port pair group

```
openstack sfc port pair group create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--description <description>]
  [--port-pair <port-pair>]
  [--enable-tap | --disable-tap]
  [--port-pair-group-parameters lb-fields=<lb-fields>]
  <name>
```

**-f** <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

**-c** COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

**--noindent**

whether to disable indenting the JSON

**--prefix** <PREFIX>

add a prefix to all variable names

**--max-width** <integer>

Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

- description** <description>  
Description for the port pair group
- port-pair** <port-pair>  
Port pair (name or ID). This option can be repeated.
- enable-tap**  
Port pairs of this port pair group are deployed as passive tap service function
- disable-tap**  
Port pairs of this port pair group are deployed as l3 service function (default)
- port-pair-group-parameters** lb-fields=<lb-fields>  
Dictionary of port pair group parameters. Currently only one parameter lb-fields is supported. <lb-fields> is a & separated list of load-balancing fields.

**name**  
Name of the port pair group

This command is provided by the python-neutronclient plugin.

### sfc port pair group delete

Delete a given port pair group

```
openstack sfc port pair group delete <port-pair-group>
```

**port-pair-group**  
Port pair group to delete (name or ID)

This command is provided by the python-neutronclient plugin.

### sfc port pair group list

List port pair group

```
openstack sfc port pair group list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [--long]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- long**  
List additional fields in output

This command is provided by the python-neutronclient plugin.

## sfc port pair group set

Set port pair group properties

```
openstack sfc port pair group set
  [--name <name>]
  [--description <description>]
  [--port-pair <port-pair>]
  [--no-port-pair]
  <port-pair-group>
```

- name** <name>  
Name of the port pair group
- description** <description>  
Description for the port pair group
- port-pair** <port-pair>  
Port pair (name or ID). This option can be repeated.
- no-port-pair**  
Remove all port pair from port pair group
- port-pair-group**  
Port pair group to modify (name or ID)

This command is provided by the python-neutronclient plugin.

## sfc port pair group show

Display port pair group details

```
openstack sfc port pair group show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  <port-pair-group>
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**port-pair-group**  
Port pair group to display (name or ID)

This command is provided by the python-neutronclient plugin.

## sfc port pair group unset

Unset port pairs from port pair group

```
openstack sfc port pair group unset
  [--port-pair <port-pair> | --all-port-pair]
  <port-pair-group>
```

**--port-pair** <port-pair>  
Remove port pair(s) from the port pair group (name or ID). This option can be repeated.

**--all-port-pair**  
Remove all port pairs from the port pair group



**port-pair-group**

Port pair group to unset (name or ID)

This command is provided by the python-neutronclient plugin.

**sfc service graph create**

Create a service graph.

```
openstack sfc service graph create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--description DESCRIPTION]
  --branching-point SRC_CHAIN
  :DST_CHAIN_1,DST_CHAIN_2,DST_CHAIN_N
  <name>
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--description** <DESCRIPTION>  
Description for the service graph.

**--branching-point** SRC\_CHAIN:DST\_CHAIN\_1,DST\_CHAIN\_2,DST\_CHAIN\_N  
Service graph branching point: the key is the source Port Chain while the value is a list of destination Port Chains. This option can be repeated.

**name**  
Name of the service graph.

This command is provided by the python-neutronclient plugin.

## sfc service graph delete

Delete a given service graph.

```
openstack sfc service graph delete <service-graph>
```

### service-graph

ID or name of the service graph to delete.

This command is provided by the python-neutronclient plugin.

## sfc service graph list

List service graphs

```
openstack sfc service graph list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [--long]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

**--sort-ascending**  
sort the column(s) in ascending order

**--sort-descending**  
sort the column(s) in descending order

**--long**  
List additional fields in output

This command is provided by the python-neutronclient plugin.

## sfc service graph set

Set service graph properties

```
openstack sfc service graph set
  [--name <name>]
  [--description <description>]
  <service-graph>
```

**--name** <name>  
Name of the service graph

**--description** <description>  
Description for the service graph

**service-graph**  
Service graph to modify (name or ID)

This command is provided by the python-neutronclient plugin.

## sfc service graph show

Show information of a given service graph.

```
openstack sfc service graph show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  <service-graph>
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>

Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**service-graph**

ID or name of the service graph to display.

This command is provided by the python-neutronclient plugin.

## network onboard subnets

**network onboard subnets** enables a subnet to be adopted or onboarded into an existing subnet pool. The CIDR of the subnet is checked for uniqueness across any applicable address scopes and all subnets allocated from the target subnet pool. Once onboarded, the subnet CIDR is added to the prefix list of the subnet pool and the subnet appears as though it has been allocated from the subnet pool. The subnet also begins participating in the applicable address scope if the subnet pool belongs to one.

Network v2

## network onboard subnets

Onboard network subnets into a subnet pool

```
openstack network onboard subnets <network> <subnetpool>
```

**network**

Onboard all subnets associated with this network

**subnetpool**

Target subnet pool for onboarding subnets

This command is provided by the python-neutronclient plugin.

## VPN Endpoint Group

The **Endpoint Group** is used to configure multiple local and remote subnets in vpnservice object.

Network v2

## vpn endpoint group create

Create an endpoint group

```
openstack vpn endpoint group create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--description <description>]
  --type TYPE
  --value ENDPOINTS
  [--project <project>]
  [--project-domain <project-domain>]
  <name>
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--description** <description>  
Description for the endpoint group

**--type** <TYPE>  
Type of endpoints in group (e.g. subnet, cidr)

**--value** <ENDPOINTS>  
Endpoint(s) for the group. Must all be of the same type. (value) option can be repeated

**--project** <project>  
Owners project (name or ID)

**--project-domain** <project-domain>  
Domain the project belongs to (name or ID). This can be used in case collisions between project names exist.

**name**

Name for the endpoint group

This command is provided by the python-neutronclient plugin.

### vpn endpoint group delete

Delete endpoint group(s)

```
openstack vpn endpoint group delete
  <endpoint-group>
  [<endpoint-group> ...]
```

**endpoint-group**

Endpoint group(s) to delete (name or ID)

This command is provided by the python-neutronclient plugin.

### vpn endpoint group list

List endpoint groups that belong to a given project

```
openstack vpn endpoint group list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [--long]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- long**  
List additional fields in output

This command is provided by the python-neutronclient plugin.

## vpn endpoint group set

Set endpoint group properties

```
openstack vpn endpoint group set
  [--description <description>]
  [--name <name>]
  <endpoint-group>
```

- description** <description>  
Description for the endpoint group
- name** <name>  
Set a name for the endpoint group
- endpoint-group**  
Endpoint group to set (name or ID)

This command is provided by the python-neutronclient plugin.

## vpn endpoint group show

Display endpoint group details

```
openstack vpn endpoint group show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  <endpoint-group>
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**

whether to disable indenting the JSON

**--prefix** <PREFIX>

add a prefix to all variable names

**--max-width** <integer>

Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**endpoint-group**

Endpoint group to display (name or ID)

This command is provided by the python-neutronclient plugin.

## VPN IKE Policy

The **IKE Policy** is used for phases one and two negotiation of the VPN connection. You can specify both the authentication and encryption algorithms for connections.

Network v2

### vpn ike policy create

Create an IKE policy

```
openstack vpn ike policy create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--description <description>]
  [--auth-algorithm {sha1,sha256,sha384,sha512}]
  [--encryption-algorithm {aes-128,3des,aes-192,aes-256}]
  [--phase1-negotiation-mode {main,aggressive}]
  [--ike-version {v1,v2}]
  [--pfs {group5,group2,group14}]
  [--lifetime units=UNITS,value=VALUE]
  [--project <project>]
  [--project-domain <project-domain>]
  <name>
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table



**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--description** <description>  
Description of the IKE policy

**--auth-algorithm** <AUTH\_ALGORITHM>  
Authentication algorithm

**--encryption-algorithm** <ENCRYPTION\_ALGORITHM>  
Encryption algorithm

**--phase1-negotiation-mode** <PHASE1\_NEGOTIATION\_MODE>  
IKE Phase1 negotiation mode

**--ike-version** <IKE\_VERSION>  
IKE version for the policy

**--pfs** <PFS>  
Perfect Forward Secrecy

**--lifetime** units=UNITS,value=VALUE  
IKE lifetime attributes. units-seconds, default:seconds. value-non negative integer, default:3600.

**--project** <project>  
Owners project (name or ID)

**--project-domain** <project-domain>  
Domain the project belongs to (name or ID). This can be used in case collisions between project names exist.

**name**  
Name of the IKE policy

This command is provided by the python-neutronclient plugin.

## vpn ike policy delete

Delete IKE policy (policies)

```
openstack vpn ike policy delete <ike-policy> [<ike-policy> ...]
```

### ike-policy

IKE policy to delete (name or ID)

This command is provided by the python-neutronclient plugin.

## vpn ike policy list

List IKE policies that belong to a given project

```
openstack vpn ike policy list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [--long]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

**--sort-ascending**  
sort the column(s) in ascending order

**--sort-descending**  
sort the column(s) in descending order

**--long**  
List additional fields in output

This command is provided by the python-neutronclient plugin.

## vpn ike policy set

Set IKE policy properties

```
openstack vpn ike policy set
  [--description <description>]
  [--auth-algorithm {sha1,sha256,sha384,sha512}]
  [--encryption-algorithm {aes-128,3des,aes-192,aes-256}]
  [--phase1-negotiation-mode {main,aggressive}]
  [--ike-version {v1,v2}]
  [--pfs {group5,group2,group14}]
  [--lifetime units=UNITS,value=VALUE]
  [--name <name>]
  <ike-policy>
```

**--description** <description>  
Description of the IKE policy

**--auth-algorithm** <AUTH\_ALGORITHM>  
Authentication algorithm

**--encryption-algorithm** <ENCRYPTION\_ALGORITHM>  
Encryption algorithm

**--phase1-negotiation-mode** <PHASE1\_NEGOTIATION\_MODE>  
IKE Phase1 negotiation mode

**--ike-version** <IKE\_VERSION>  
IKE version for the policy

**--pfs** <PFS>  
Perfect Forward Secrecy

**--lifetime** units=UNITS,value=VALUE  
IKE lifetime attributes. units-seconds, default:seconds. value-non negative integer, default:3600.

**--name** <name>  
Name of the IKE policy

**ike-policy**  
IKE policy to set (name or ID)

This command is provided by the python-neutronclient plugin.

## vpn ike policy show

Display IKE policy details

```
openstack vpn ike policy show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  <ike-policy>
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**ike-policy**  
IKE policy to display (name or ID)

This command is provided by the python-neutronclient plugin.

## VPN IPsec Policy

The **IPsec Policy** specifies the authentication and encryption algorithms and encapsulation mode to use for the established VPN connection.

Network v2

## vpn ipsec policy create

Create an IPsec policy

```

openstack vpn ipsec policy create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--description <description>]
  [--auth-algorithm {sha1,sha256,sha384,sha512}]
  [--encapsulation-mode {tunnel,transport}]
  [--encryption-algorithm {3des,aes-128,aes-192,aes-256}]
  [--lifetime units=UNITS,value=VALUE]
  [--pfs {group2,group5,group14}]
  [--transform-protocol {esp,ah,ah-esp}]
  [--project <project>]
  [--project-domain <project-domain>]
  <name>

```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--description** <description>  
Description of the IPsec policy

**--auth-algorithm** <AUTH\_ALGORITHM>  
Authentication algorithm for IPsec policy

**--encapsulation-mode** <ENCAPSULATION\_MODE>  
Encapsulation mode for IPsec policy

**--encryption-algorithm** <ENCRYPTION\_ALGORITHM>  
Encryption algorithm for IPsec policy

- lifetime** units=UNITS,value=VALUE  
IPsec lifetime attributes. units-seconds, default:seconds. value-non negative integer, default:3600.
- pfs** <PFS>  
Perfect Forward Secrecy for IPsec policy
- transform-protocol** <TRANSFORM\_PROTOCOL>  
Transform protocol for IPsec policy
- project** <project>  
Owners project (name or ID)
- project-domain** <project-domain>  
Domain the project belongs to (name or ID). This can be used in case collisions between project names exist.

**name**  
Name of the IPsec policy

This command is provided by the python-neutronclient plugin.

### vpn ipsec policy delete

Delete IPsec policy(policies)

```
openstack vpn ipsec policy delete <ipsec-policy> [<ipsec-policy> ...]
```

**ipsec-policy**  
ipsec policy to delete (name or ID)

This command is provided by the python-neutronclient plugin.

### vpn ipsec policy list

List IPsec policies that belong to a given project

```
openstack vpn ipsec policy list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [--long]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- long**  
List additional fields in output

This command is provided by the python-neutronclient plugin.

## vpn ipsec policy set

Set IPsec policy properties

```
openstack vpn ipsec policy set
  [--description <description>]
  [--auth-algorithm {sha1,sha256,sha384,sha512}]
  [--encapsulation-mode {tunnel,transport}]
  [--encryption-algorithm {3des,aes-128,aes-192,aes-256}]
  [--lifetime units=UNITS,value=VALUE]
  [--pfs {group2,group5,group14}]
  [--transform-protocol {esp,ah,ah-esp}]
  [--name <name>]
  <ipsec-policy>
```

- description** <description>  
Description of the IPsec policy
- auth-algorithm** <AUTH\_ALGORITHM>  
Authentication algorithm for IPsec policy
- encapsulation-mode** <ENCAPSULATION\_MODE>  
Encapsulation mode for IPsec policy
- encryption-algorithm** <ENCRYPTION\_ALGORITHM>  
Encryption algorithm for IPsec policy
- lifetime** units=UNITS,value=VALUE  
IPsec lifetime attributes. units-seconds, default:seconds. value-non negative integer, default:3600.

- pfs** <PFS>  
Perfect Forward Secrecy for IPsec policy
- transform-protocol** <TRANSFORM\_PROTOCOL>  
Transform protocol for IPsec policy
- name** <name>  
Name of the IPsec policy

**ipsec-policy**

IPsec policy to set (name or ID)

This command is provided by the python-neutronclient plugin.

## vpn ipsec policy show

Display IPsec policy details

```
openstack vpn ipsec policy show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  <ipsec-policy>
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
  - c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
  - noindent**  
whether to disable indenting the JSON
  - prefix** <PREFIX>  
add a prefix to all variable names
  - max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
  - fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
  - print-empty**  
Print empty table if there is no data to show.
- ipsec-policy**  
IPsec policy to display (name or ID)

This command is provided by the python-neutronclient plugin.



## VPN IPsec Site Connection

Creates a site-to-site **IPsec Site Connection** for a VPN service.

Network v2

### vpn ipsec site connection create

Create an IPsec site connection

```
openstack vpn ipsec site connection create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--description <description>]
  [--dpd action=ACTION,interval=INTERVAL,timeout=TIMEOUT]
  [--mtu MTU]
  [--initiator {bi-directional,response-only}]
  [--peer-cidr PEER_CIDRS | --local-endpoint-group LOCAL_ENDPOINT_GROUP]
  [--peer-endpoint-group PEER_ENDPOINT_GROUP]
  [--enable | --disable]
  [--local-id LOCAL_ID]
  --peer-id PEER_ID
  --peer-address PEER_ADDRESS
  --psk PSK
  --vpnservice VPNSERVICE
  --ikepolicy IKEPOLICY
  --ipsecpolicy IPSECPOLICY
  [--project <project>]
  [--project-domain <project-domain>]
  <name>
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--description** <description>  
Description for the connection

**--dpd** action=ACTION, interval=INTERVAL, timeout=TIMEOUT  
Ipssec connection Dead Peer Detection attributes. action=hold,clear,disabled,restart,restart-by-peer.  
interval and timeout are non negative integers. interval should be less than timeout value. action,  
default:hold interval, default:30, timeout, default:120.

**--mtu** <MTU>  
MTU size for the connection

**--initiator** <INITIATOR>  
Initiator state

**--peer-cidr** <PEER\_CIDRS>  
Remote subnet(s) in CIDR format. Cannot be specified when using endpoint groups. Only appli-  
cable, if subnet provided for VPN service.

**--local-endpoint-group** <LOCAL\_ENDPOINT\_GROUP>  
Local endpoint group (name or ID) with subnet(s) for IPsec connection

**--peer-endpoint-group** <PEER\_ENDPOINT\_GROUP>  
Peer endpoint group (name or ID) with CIDR(s) for IPsec connection

**--enable**  
Enable IPsec site connection

**--disable**  
Disable IPsec site connection

**--local-id** <LOCAL\_ID>  
An ID to be used instead of the external IP address for a virtual router

**--peer-id** <PEER\_ID>  
Peer router identity for authentication. Can be IPv4/IPv6 address, e-mail address, key id, or FQDN

**--peer-address** <PEER\_ADDRESS>  
Peer gateway public IPv4/IPv6 address or FQDN

**--psk** <PSK>  
Pre-shared key string.

**--vpnservice** VPNSERVICE  
VPN service instance associated with this connection (name or ID)

**--ikepolicy** IKEPOLICY  
IKE policy associated with this connection (name or ID)

**--ipsecpolicy** IPSECPOLICY  
IPsec policy associated with this connection (name or ID)

**--project** <project>  
Owners project (name or ID)

**--project-domain** <project-domain>  
Domain the project belongs to (name or ID). This can be used in case collisions between project  
names exist.

**name**

Set friendly name for the connection

This command is provided by the python-neutronclient plugin.

**vpn ipsec site connection delete**

Delete IPsec site connection(s)

```
openstack vpn ipsec site connection delete
  <ipsec-site-connection>
  [<ipsec-site-connection> ...]
```

**ipsec-site-connection**

IPsec site connection to delete (name or ID)

This command is provided by the python-neutronclient plugin.

**vpn ipsec site connection list**

List IPsec site connections that belong to a given project

```
openstack vpn ipsec site connection list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [--long]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- long**  
List additional fields in output

This command is provided by the python-neutronclient plugin.

## vpn ipsec site connection set

Set IPsec site connection properties

```
openstack vpn ipsec site connection set
  [--description <description>]
  [--dpd action=ACTION,interval=INTERVAL,timeout=TIMEOUT]
  [--mtu MTU]
  [--initiator {bi-directional,response-only}]
  [--peer-cidr PEER_CIDRS | --local-endpoint-group LOCAL_ENDPOINT_GROUP]
  [--peer-endpoint-group PEER_ENDPOINT_GROUP]
  [--enable | --disable]
  [--local-id LOCAL_ID]
  [--peer-id PEER_ID]
  [--peer-address PEER_ADDRESS]
  [--name <name>]
  <ipsec-site-connection>
```

- description** <description>  
Description for the connection
- dpd** action=ACTION,interval=INTERVAL,timeout=TIMEOUT  
Ipssec connection Dead Peer Detection attributes. action-hold,clear,disabled,restart,restart-by-peer. interval and timeout are non negative integers. interval should be less than timeout value. action, default:hold interval, default:30, timeout, default:120.
- mtu** <MTU>  
MTU size for the connection
- initiator** <INITIATOR>  
Initiator state
- peer-cidr** <PEER\_CIDRS>  
Remote subnet(s) in CIDR format. Cannot be specified when using endpoint groups. Only applicable, if subnet provided for VPN service.
- local-endpoint-group** <LOCAL\_ENDPOINT\_GROUP>  
Local endpoint group (name or ID) with subnet(s) for IPsec connection
- peer-endpoint-group** <PEER\_ENDPOINT\_GROUP>  
Peer endpoint group (name or ID) with CIDR(s) for IPsec connection

- enable**  
Enable IPsec site connection
- disable**  
Disable IPsec site connection
- local-id** <LOCAL\_ID>  
An ID to be used instead of the external IP address for a virtual router
- peer-id** <PEER\_ID>  
Peer router identity for authentication. Can be IPv4/IPv6 address, e-mail address, key id, or FQDN
- peer-address** <PEER\_ADDRESS>  
Peer gateway public IPv4/IPv6 address or FQDN
- name** <name>  
Set friendly name for the connection
- ipsec-site-connection**  
IPsec site connection to set (name or ID)

This command is provided by the python-neutronclient plugin.

### vpn ipsec site connection show

Show information of a given IPsec site connection

```
openstack vpn ipsec site connection show
[-f {json,shell,table,value,yaml}]
[-c COLUMN]
[--noindent]
[--prefix PREFIX]
[--max-width <integer>]
[--fit-width]
[--print-empty]
<ipsec-site-connection>
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**ipsec-site-connection**

IPsec site connection to display (name or ID)

This command is provided by the python-neutronclient plugin.

## VPN Service

The **VPN Service** is associated with a router. After you create the service, it can contain multiple VPN connections.

Network v2

### vpn service create

Create an VPN service

```
openstack vpn service create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--description <description>]
  [--subnet <subnet>]
  [--flavor <flavor>]
  [--enable | --disable]
  --router ROUTER
  [--project <project>]
  [--project-domain <project-domain>]
  <name>
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

- print-empty**  
Print empty table if there is no data to show.
- description** <description>  
Description for the VPN service
- subnet** <subnet>  
Local private subnet (name or ID)
- flavor** <flavor>  
Flavor for the VPN service (name or ID)
- enable**  
Enable VPN service
- disable**  
Disable VPN service
- router** ROUTER  
Router for the VPN service (name or ID)
- project** <project>  
Owners project (name or ID)
- project-domain** <project-domain>  
Domain the project belongs to (name or ID). This can be used in case collisions between project names exist.
- name**  
Name for the VPN service

This command is provided by the python-neutronclient plugin.

### vpn service delete

Delete VPN service(s)

```
openstack vpn service delete <vpn-service> [<vpn-service> ...]
```

#### vpn-service

VPN service to delete (name or ID)

This command is provided by the python-neutronclient plugin.

### vpn service list

List VPN services that belong to a given project

```
openstack vpn service list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
```

(continues on next page)

(continued from previous page)

```
[--sort-column SORT_COLUMN]
[--sort-ascending | --sort-descending]
[--long]
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric
- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- long**  
List additional fields in output

This command is provided by the python-neutronclient plugin.

## vpn service set

Set VPN service properties

```
openstack vpn service set
  [--description <description>]
  [--subnet <subnet>]
  [--flavor <flavor>]
  [--enable | --disable]
  [--name <name>]
  <vpn-service>
```

- description** <description>  
Description for the VPN service



- subnet** <subnet>  
Local private subnet (name or ID)
- flavor** <flavor>  
Flavor for the VPN service (name or ID)
- enable**  
Enable VPN service
- disable**  
Disable VPN service
- name** <name>  
Name for the VPN service
- vpn-service**  
VPN service to modify (name or ID)

This command is provided by the python-neutronclient plugin.

## vpn service show

Display VPN service details

```
openstack vpn service show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  <vpn-service>
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.

### vpn-service

VPN service to display (name or ID)

This command is provided by the python-neutronclient plugin.

## 1.1.2 neutron CLI

**Warning:** neutron CLI is now deprecated and will be removed in the future. Use openstack CLI instead. See [openstack CLI command list](#) and *its extensions for advanced networking services*. The command mapping from neutron CLI to openstack CLI is available [here](#).

### Using neutron CLI

The **neutron** shell utility interacts with OpenStack Networking API from the command-line. It supports the entire features of OpenStack Networking API.

**Warning:** neutron CLI is now deprecated and will be removed in the future. Use openstack CLI instead. See [openstack CLI command list](#) and *its extensions for advanced networking services*. The command mapping from neutron CLI to openstack CLI is available [here](#).

### Basic Usage

In order to use the CLI, you must provide your OpenStack username, password, project, domain information for both user and project, and auth endpoint. Use the corresponding configuration options (`--os-username`, `--os-password`, `--os-project-name`, `--os-user-domain-id`, `os-project-domain-id`, and `--os-auth-url`), but it is easier to set them in environment variables.

```
export OS_USERNAME=user
export OS_PASSWORD=pass
export OS_PROJECT_NAME=project
export OS_USER_DOMAIN_ID=default
export OS_PROJECT_DOMAIN_ID=default
export OS_AUTH_URL=http://auth.example.com:5000/v3
```

If you are using Identity v2.0 API (DEPRECATED), you dont need to pass domain information.

```
export OS_USERNAME=user
export OS_PASSWORD=pass
export OS_TENANT_NAME=tenant
export OS_AUTH_URL=http://auth.example.com:5000/v2.0
```

Once youve configured your authentication parameters, you can run **neutron** commands. All commands take the form of:

```
neutron <command> [arguments...]
```

Run **neutron help** to get a full list of all possible commands, and run **neutron help <command>** to get detailed help for that command.

## Using with os-client-config

`os-client-config` provides more convenient way to manage a collection of client configurations and you can easily switch multiple OpenStack-based configurations.

To use `os-client-config`, you first need to prepare `~/.config/openstack/clouds.yaml` like the following.

```
clouds:
  devstack:
    auth:
      auth_url: http://auth.example.com:5000
      password: your-secret
      project_domain_id: default
      project_name: demo
      user_domain_id: default
      username: demo
      identity_api_version: '3'
      region_name: RegionOne
  devstack-admin:
    auth:
      auth_url: http://auth.example.com:35357
      password: another-secret
      project_domain_id: default
      project_name: admin
      user_domain_id: default
      username: admin
      identity_api_version: '3'
      region_name: RegionOne
```

Then, you need to specify a configuration name defined in the above `clouds.yaml`.

```
export OS_CLOUD=devstack
```

For more detail information, see the `os-client-config` documentation.

## Using with keystone token

The command-line tool will attempt to re-authenticate using your provided credentials for every request. You can override this behavior by manually supplying an auth token using `--os-url` and `--os-auth-token`. You can alternatively set these environment variables.

```
export OS_URL=http://neutron.example.org:9696/
export OS_TOKEN=3bcc3d3a03f44e3d8377f9247b0ad155
```

## Using noauth mode

If neutron server does not require authentication, besides these two arguments or environment variables (We can use any value as token.), we need manually supply `--os-auth-strategy` or set the environment variable.

```
export OS_AUTH_STRATEGY=noauth
```

## Display options

### Filtering

Neutron API supports filtering in the listing operation. **neutron** CLI supports this feature too.

To specify a filter in `*-list` command, you need to pass a pair of an attribute name and an expected value with the format of `--<attribute> <value>`. The example below retrieves ports owned by compute instances.

```
$ neutron port-list --device_owner network:dhcp
+-----+-----+-----+-----+
| id | name | mac_address | fixed_ |
| ips | | | |
+-----+-----+-----+-----+
| 8953d683-29ad-4be3-b73f-060727c7849b | | fa:16:3e:4b:9e:0a | {
| "subnet_id": "6b832dfe-f271-443c-abad-629961414a73", "ip_address": "10.0.
| 0.2"} |
| | | | | {
| "subnet_id": "cdcc616b-0cff-482f-96f5-06fc63d21247", "ip_address":
| "fd12:877c:1d66:0:f816:3eff:fe4b:9e0a"} |
+-----+-----+-----+-----+
| | | | |
+-----+-----+-----+-----+
```

You can also specify multiple filters. The example below retrieves security group rules applied to IPv4 traffic which belongs to a security group `bfa493f9-2b03-46d2-8399-b9b038a53bc1`.

```
$ neutron security-group-rule-list --security-group-id bfa493f9-2b03-46d2-
8399-b9b038a53bc1 --ethertype IPv4
+-----+-----+-----+-----+-----+
| id | id | security_group | direction |
| ethertype | protocol/port | remote | |
+-----+-----+-----+-----+-----+
| 65489805-0400-4bce-9bd9-16a81952263c | default | egress | IPv4 |
| any | any | | |
| 9429f336-4947-4643-bbd9-24528cc65648 | default | ingress | IPv4 |
| any | default (group) | | |
+-----+-----+-----+-----+-----+
(continues on next page)
```

(continued from previous page)

---

**Note:** Looking up UUID from name is not supported when specifying a filter. You need to use UUID to specify a specific resource.

---



---

**Note:** Filtering for dictionary or list attributes is not supported.

---

## Changing displayed columns

If you want displayed columns in a list operation, `-c` option can be used. `-c` can be specified multiple times and the column order will be same as the order of `-c` options.

```

$ neutron port-list -c id -c device_owner -c fixed_ips
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
↪-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
↪-----+
| id                                     | device_owner                               | fixed_
↪ips                                     |                                           |
↪                                     |                                           |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
↪-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
↪-----+
| 41ca1b9b-4bbd-4aa8-bcaa-31d3d5704205 | network:router_interface | {
↪"subnet_id": "6b832dfe-f271-443c-abad-629961414a73", "ip_address": "10.0.
↪0.1"}
| 8953d683-29ad-4be3-b73f-060727c7849b | network:dhcp                | {
↪"subnet_id": "6b832dfe-f271-443c-abad-629961414a73", "ip_address": "10.0.
↪0.2"}
|                                     |                               | {
↪"subnet_id": "cdcc616b-0cff-482f-96f5-06fc63d21247", "ip_address":
↪"fd12:877c:1d66:0:f816:3eff:fe4b:9e0a"} |
| a9da29f8-4504-4526-a5ce-cd3624fbd173 | neutron:LOADBALANCER        | {
↪"subnet_id": "6b832dfe-f271-443c-abad-629961414a73", "ip_address": "10.0.
↪0.3"}
|                                     |                               | {
↪"subnet_id": "cdcc616b-0cff-482f-96f5-06fc63d21247", "ip_address":
↪"fd12:877c:1d66:0:f816:3eff:feb1:ab71"} |
| d6a1ff96-0a99-416f-a4d6-65d9614cf64e | compute:nova                 | {
↪"subnet_id": "6b832dfe-f271-443c-abad-629961414a73", "ip_address": "10.0.
↪0.4"}
|                                     |                               | {
↪"subnet_id": "cdcc616b-0cff-482f-96f5-06fc63d21247", "ip_address":
↪"fd12:877c:1d66:0:f816:3eff:fe2c:348e"} |
| f4789225-26d0-409f-8047-82d2c7a87a95 | network:router_interface | {
↪"subnet_id": "cdcc616b-0cff-482f-96f5-06fc63d21247", "ip_address":
↪"fd12:877c:1d66::1"}
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
↪-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
↪-----+

```

## Extra arguments for create/update operation

**neutron** CLI has a mechanism called the *extra arguments* for `*-create` and `*-update` commands. It allows users to specify a set of *unknown options* which are not defined as options and not shown in the help text. **Unknown options MUST be placed at the end of the command line.** *unknown options* will be directly passed to the API layer. By this mechanism, you can pass an attribute which is not defined in the upstream **neutron** CLI. For example, when you are developing a new feature which add a new attribute to an existing resource, it is useful because we can test your feature without changing the existing neutron CLI.

For example, if you run the following command:

```
neutron resource-update <ID> --key1 value1 --key2 value2
```

where `resource` is some resource name and `--key1` and `--key2` are unknown options, then the following JSON will be sent to the neutron API:

```
PUT /v2.0/resources/<ID>
{
  "resource": {
    "key2": "value2",
    "key1": "value1"
  }
}
```

## Key interpretation

This means an option name (`--key1` in this case) must be one of valid resources of a corresponding resource. An option name `--foo_bar` is recognized as an attribute name `foo_bar`. `--foo-bar` is also interpreted as an attribute name `foo_bar`.

## Value interpretation

By default, if the number of values is 1, the option value is interpreted as a string and is passed to the API layer as specified in a command-line.

If the number of values is greater than 1, the option value is interpreted as a list and the result in the API layer will be same as when specifying a list as described below.

```
neutron resource-update <ID> key1 val1 val2 val3 key2 val4
```

In the above example, a value of `key1` is interpreted as `["val1", "val2", "val3"]` and a value of `key2` is interpreted as `val4`.

The extra argument mechanism supports more complex value like a list or a dict.

## Specify a list value

A command-line:

```
neutron resource-update <ID> --key list=true val1 val2 val3
```

will send the following in the API layer:

```
{
  "key": [
    "val1",
    "val2",
    "val3"
  ]
}
```

---

**Note:** If you want to specify a list value, it is recommended to specify `list=true`. When `list=true` is specified, specified values are interpreted as a list even regardless of the number of values.

If `list=true` is not specified, specified values are interpreted depends on the number of values how. If the number of values is more than 2, the specified values are interpreted as a list. If 1, the value is interpreted as a string.

---

## Specify a dict value

A command-line:

```
neutron resource-update <ID> --key type=dict key1=val1,key2=val2,key3=val3
```

will send the following in the API layer:

```
{
  "key": {
    "key1": "val1",
    "key2": "val2",
    "key3": "val3"
  }
}
```

---

**Note:** `type=bool True/False` and `type=int 10` are also supported.

---

## Specify a list of dicts

A command-line:

```
neutron resource-update <ID> --key type=dict list=true key1=val1 key2=val2_
↪key3=val3
```

will send the following in the API layer:

```
{
  "key": [
    {"key1": "val1"},
    {"key2": "val2"},
    {"key3": "val3"}
  ]
}
```

## Passing None as a value

There is a case where we would like to pass None (null in JSON) in the API layer. To do this:

```
neutron resource-update <ID> --key action=clear
```

The following body will be in the API layer:

```
{"key": null}
```

---

**Note:** If `action=clear` is specified, `list=true` or `type=dict` is ignored. It means when `action=clear` is specified None is always sent.

---

## Debugging

### Display API-level communication

`-v` (or `--verbose`, `--debug`) option displays a detail interaction with your neutron server. It is useful to debug what happens in the API level.

Here is an sample output of `net-show` command.

The first line show what parameters are recognized by `neutronclient`. It is sometimes useful to check if command-line parameters you specify are recognized properly.

```
$ neutron -v net-show mynetwork
DEBUG: neutronclient.neutron.v2_0.network.ShowNetwork get_
↪data(Namespace(columns=[], fields=[], formatter='table', id='mynetwork',
↪ max_width=0, noindent=False, prefix='', request_format='json', show_
↪details=False, variables=[]))
```

Next, `neutronclient` sends an authentication request to keystone to get a token which is used in further operations.



```

DEBUG: keystoneauth.session REQ: curl -g -i -X GET http://172.16.18.
↳47:5000 -H "Accept: application/json" -H "User-Agent: keystoneauth1"
DEBUG: keystoneauth.session RESP: [300] Content-Length: 593 Vary: X-Auth-
↳Token Keep-Alive: timeout=5, max=100 Server: Apache/2.4.7 (Ubuntu)
↳Connection: Keep-Alive Date: Fri, 27 Nov 2015 20:10:54 GMT Content-Type:
↳application/json
RESP BODY: {"versions": {"values": [{"status": "stable", "updated": "2015-
↳03-30T00:00:00Z", "media-types": [{"base": "application/json", "type":
↳"application/vnd.openstack.identity-v3+json"}], "id": "v3.4", "links": [{"
↳href": "http://172.16.18.47:5000/v3/", "rel": "self"}]}, {"status":
↳"stable", "updated": "2014-04-17T00:00:00Z", "media-types": [{"base":
↳"application/json", "type": "application/vnd.openstack.identity-v2.0+json
↳"}], "id": "v2.0", "links": [{"href": "http://172.16.18.47:5000/v2.0/",
↳"rel": "self"}, {"href": "http://docs.openstack.org/", "type": "text/html
↳", "rel": "describedby"}]}]}]}

DEBUG: keystoneauth.identity.v3.base Making authentication request to
↳http://172.16.18.47:5000/v3/auth/tokens

```

Neutronclient looks up a network ID corresponding to a given network name.

```

DEBUG: keystoneauth.session REQ: curl -g -i -X GET http://172.16.18.
↳47:9696/v2.0/networks.json?fields=id&name=mynetwork -H "User-Agent:
↳python-neutronclient" -H "Accept: application/json" -H "X-Auth-Token:
↳{SHA1}39300e7398d53a02afd183f13cb6afaef95ec4e5"
DEBUG: keystoneauth.session RESP: [200] Date: Fri, 27 Nov 2015 20:10:55
↳GMT Connection: keep-alive Content-Type: application/json; charset=UTF-8
↳Content-Length: 62 X-Openstack-Request-Id: req-cceb6e4-4f52-4874-alab-
↳5499abcba378
RESP BODY: {"networks": [{"id": "3698d3c7-d581-443e-bf86-53c4e3a738f7"}]}

```

Finally, neutronclient retrieves a detail of a given network using the resolved ID.

```

DEBUG: keystoneauth.session REQ: curl -g -i -X GET http://172.16.18.
↳47:9696/v2.0/networks/3698d3c7-d581-443e-bf86-53c4e3a738f7.json -H "User-
↳Agent: python-neutronclient" -H "Accept: application/json" -H "X-Auth-
↳Token: {SHA1}39300e7398d53a02afd183f13cb6afaef95ec4e5"
DEBUG: keystoneauth.session RESP: [200] Date: Fri, 27 Nov 2015 20:10:55
↳GMT Connection: keep-alive Content-Type: application/json; charset=UTF-8
↳Content-Length: 272 X-Openstack-Request-Id: req-261add00-d6d3-4ea7-becc-
↳105b60ac7369
RESP BODY: {"network": {"status": "ACTIVE", "subnets": [], "name":
↳"mynetwork", "admin_state_up": true, "tenant_id":
↳"8f0ebf767043483a987736c8c684178d", "mtu": 0, "router:external": false,
↳"shared": false, "port_security_enabled": true, "id": "3698d3c7-d581-
↳443e-bf86-53c4e3a738f7"}}

```

Field	Value
admin_state_up	True
id	3698d3c7-d581-443e-bf86-53c4e3a738f7
mtu	0
name	mynetwork
port_security_enabled	True
router:external	False

(continues on next page)

(continued from previous page)

shared	False	
status	ACTIVE	
subnets		
tenant_id	8f0ebf767043483a987736c8c684178d	
+-----+	+-----+	+-----+

## neutron CLI reference

**Warning:** neutron CLI is now deprecated and will be removed in the future. Use openstack CLI instead. See [openstack CLI command list](#) and *its extensions for advanced networking services*. The command mapping from neutron CLI to openstack CLI is available [here](#).

## neutron usage

Command-line interface to the Neutron APIs (neutron CLI version: 7.3.2)

```
neutron
  [--version]
  [-v]
  [-q]
  [-r NUM]
  [--os-service-type <os-service-type>]
  [--os-endpoint-type <os-endpoint-type>]
  [--service-type <service-type>]
  [--endpoint-type <endpoint-type>]
  [--os-auth-strategy <auth-strategy>]
  [--os-cloud <cloud>]
  [--os-auth-url <auth-url>]
  [--os-tenant-name <auth-tenant-name> | --os-project-name <auth-project-
  ↪name>]
  [--os-tenant-id <auth-tenant-id> | --os-project-id <auth-project-id>]
  [--os-username <auth-username>]
  [--os-user-id <auth-user-id>]
  [--os-user-domain-id <auth-user-domain-id>]
  [--os-user-domain-name <auth-user-domain-name>]
  [--os-project-domain-id <auth-project-domain-id>]
  [--os-project-domain-name <auth-project-domain-name>]
  [--os-cert <certificate>]
  [--os-cacert <ca-certificate>]
  [--os-key <key>]
  [--os-password <auth-password>]
  [--os-region-name <auth-region-name>]
  [--os-token <token>]
  [--http-timeout <seconds>]
  [--os-url <url>]
  [--insecure]
```

### --version

show programs version number and exit

### -v, --verbose, --debug

Increase verbosity of output and show tracebacks on errors. You can repeat this option.

- q, --quiet**  
Suppress output except warnings and errors.
- r NUM, --retries NUM**  
How many times the request to the Neutron server should be retried if it fails. Defaults to 0.
- os-service-type <os-service-type>**  
Defaults to env[OS\_NETWORK\_SERVICE\_TYPE] or network.
- os-endpoint-type <os-endpoint-type>**  
Defaults to env[OS\_ENDPOINT\_TYPE] or public.
- service-type <service-type>**  
DEPRECATED! Use os-service-type.
- endpoint-type <endpoint-type>**  
DEPRECATED! Use os-endpoint-type.
- os-auth-strategy <auth-strategy>**  
DEPRECATED! Only keystone is supported.
- os-cloud <cloud>**  
Defaults to env[OS\_CLOUD].
- os-auth-url <auth-url>**  
Authentication URL, defaults to env[OS\_AUTH\_URL].
- os-tenant-name <auth-tenant-name>**  
Authentication tenant name, defaults to env[OS\_TENANT\_NAME].
- os-project-name <auth-project-name>**  
Another way to specify tenant name. This option is mutually exclusive with os-tenant-name.  
Defaults to env[OS\_PROJECT\_NAME].
- os-tenant-id <auth-tenant-id>**  
Authentication tenant ID, defaults to env[OS\_TENANT\_ID].
- os-project-id <auth-project-id>**  
Another way to specify tenant ID. This option is mutually exclusive with os-tenant-id. Defaults to env[OS\_PROJECT\_ID].
- os-username <auth-username>**  
Authentication username, defaults to env[OS\_USERNAME].
- os-user-id <auth-user-id>**  
Authentication user ID (Env: OS\_USER\_ID)
- os-user-domain-id <auth-user-domain-id>**  
OpenStack user domain ID. Defaults to env[OS\_USER\_DOMAIN\_ID].
- os-user-domain-name <auth-user-domain-name>**  
OpenStack user domain name. Defaults to env[OS\_USER\_DOMAIN\_NAME].
- os-project-domain-id <auth-project-domain-id>**  
Defaults to env[OS\_PROJECT\_DOMAIN\_ID].
- os-project-domain-name <auth-project-domain-name>**  
Defaults to env[OS\_PROJECT\_DOMAIN\_NAME].

- os-cert** <certificate>  
Path of certificate file to use in SSL connection. This file can optionally be prepended with the private key. Defaults to env[OS\_CERT].
- os-cacert** <ca-certificate>  
Specify a CA bundle file to use in verifying a TLS (https) server certificate. Defaults to env[OS\_CACERT].
- os-key** <key>  
Path of client key to use in SSL connection. This option is not necessary if your key is prepended to your certificate file. Defaults to env[OS\_KEY].
- os-password** <auth-password>  
Authentication password, defaults to env[OS\_PASSWORD].
- os-region-name** <auth-region-name>  
Authentication region name, defaults to env[OS\_REGION\_NAME].
- os-token** <token>  
Authentication token, defaults to env[OS\_TOKEN].
- http-timeout** <seconds>  
Timeout in seconds to wait for an HTTP response. Defaults to env[OS\_NETWORK\_TIMEOUT] or None if not specified.
- os-url** <url>  
Defaults to env[OS\_URL].
- insecure**  
Explicitly allow neutronclient to perform insecure SSL (https) requests. The servers certificate will not be verified against any certificate authorities. This option should be used with caution.

## neutron API v2.0 commands

### address-scope-create

Create an address scope for a given tenant.

```
neutron address-scope-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  [--shared]
  NAME
  IP_VERSION
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- tenant-id** TENANT\_ID  
The owner tenant ID.
- shared**  
Set the address scope as shared.

**NAME**

Specify the name of the address scope.

**IP\_VERSION**

Specify the address family of the address scope.

This command is provided by the python-neutronclient plugin.

**address-scope-delete**

Delete an address scope.

```
neutron address-scope-delete ADDRESS_SCOPE [ADDRESS_SCOPE ...]
```

**ADDRESS\_SCOPE**

ID(s) or name(s) of address\_scope to delete.

This command is provided by the python-neutronclient plugin.

**address-scope-list**

List address scopes that belong to a given tenant.

```
neutron address-scope-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
```

(continues on next page)

(continued from previous page)

```
[--sort-ascending | --sort-descending]
[-D]
[-F FIELD]
[-P SIZE]
[--sort-key FIELD]
[--sort-dir {asc,desc}]
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric
- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D**, **--show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- P** SIZE, **--page-size** SIZE  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key** FIELD  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.
- sort-dir** {asc,desc}  
Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## address-scope-show

Show information about an address scope.

```

neutron address-scope-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  ADDRESS_SCOPE

```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**-D**, **--show-details**  
Show detailed information.

**-F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.

**ADDRESS\_SCOPE**  
ID or name of address\_scope to look up.

This command is provided by the python-neutronclient plugin.

## address-scope-update

Update an address scope.

```
neutron address-scope-update
  [--name NAME]
  [--shared {True,False}]
  ADDRESS_SCOPE
```

**--name** <NAME>

Updated name of the address scope.

**--shared** {True,False}

Set sharing of address scope. (True means shared)

**ADDRESS\_SCOPE**

ID or name of address\_scope to update.

This command is provided by the python-neutronclient plugin.

## agent-delete

Delete a given agent.

```
neutron agent-delete AGENT [AGENT ...]
```

**AGENT**

ID(s) of agent to delete.

This command is provided by the python-neutronclient plugin.

## agent-list

List agents.

```
neutron agent-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
```

**-f** <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

**-c** COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns



- quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric
- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- sort-key** FIELD  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.
- sort-dir** {asc,desc}  
Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## agent-show

Show information of a given agent.

```
neutron agent-show
[-f {json,shell,table,value,yaml}]
[-c COLUMN]
[--noindent]
[--prefix PREFIX]
[--max-width <integer>]
[--fit-width]
[--print-empty]
[-D]
```

(continues on next page)

(continued from previous page)

```
[-F FIELD]
AGENT
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
  - c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
  - noindent**  
whether to disable indenting the JSON
  - prefix** <PREFIX>  
add a prefix to all variable names
  - max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
  - fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
  - print-empty**  
Print empty table if there is no data to show.
  - D**, **--show-details**  
Show detailed information.
  - F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- AGENT**  
ID of agent to look up.

This command is provided by the python-neutronclient plugin.

## agent-update

Updates the admin status and description for a specified agent.

```
neutron agent-update
  [--admin-state-down]
  [--description DESCRIPTION]
AGENT
```

- admin-state-down**  
Set admin state up of the agent to false.
  - description** <DESCRIPTION>  
Description for the agent.
- AGENT**  
ID of agent to update.

This command is provided by the python-neutronclient plugin.

## auto-allocated-topology-delete

Delete the auto-allocated topology of a given tenant.

```
neutron auto-allocated-topology-delete [--tenant-id tenant-id]
```

**--tenant-id** tenant-id  
The owner tenant ID.

This command is provided by the python-neutronclient plugin.

## auto-allocated-topology-show

Show the auto-allocated topology of a given tenant.

```
neutron auto-allocated-topology-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--dry-run]
  [--tenant-id tenant-id]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--dry-run**  
Validate the requirements for auto-allocated-topology. (Does not return a topology.)

**--tenant-id** tenant-id  
The owner tenant ID.

This command is provided by the python-neutronclient plugin.

## availability-zone-list

List availability zones.

```
neutron availability-zone-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

**--sort-ascending**  
sort the column(s) in ascending order

**--sort-descending**  
sort the column(s) in descending order

**-D**, **--show-details**  
Show detailed information.

**-F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.

**-P** SIZE, **--page-size** SIZE

Specify retrieve unit of each request, then split one request to several requests.

**--sort-key** FIELD

Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.

**--sort-dir** {asc,desc}

Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## bash-completion

Prints all of the commands and options for bash-completion.

```
neutron bash-completion
```

This command is provided by the python-neutronclient plugin.

## bgp-dragent-list-hosting-speaker

List Dynamic Routing agents hosting a BGP speaker.

```
neutron bgp-dragent-list-hosting-speaker
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  BGP_SPEAKER
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN

specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

**--sort-ascending**

sort the column(s) in ascending order

**--sort-descending**

sort the column(s) in descending order

**-D, --show-details**

Show detailed information.

**-F** FIELD, **--field** FIELD

Specify the field(s) to be returned by server. You can repeat this option.

**BGP\_SPEAKER**

ID or name of the BGP speaker.

This command is provided by the python-neutronclient plugin.

## bgp-dragent-speaker-add

Add a BGP speaker to a Dynamic Routing agent.

```
neutron bgp-dragent-speaker-add BGP_DRAGENT_ID BGP_SPEAKER
```

**BGP\_DRAGENT\_ID**

ID of the Dynamic Routing agent.

**BGP\_SPEAKER**

ID or name of the BGP speaker.

This command is provided by the python-neutronclient plugin.

## bgp-dragent-speaker-remove

Removes a BGP speaker from a Dynamic Routing agent.

```
neutron bgp-dragent-speaker-remove BGP_DRAGENT_ID BGP_SPEAKER
```

**BGP\_DRAGENT\_ID**

ID of the Dynamic Routing agent.

**BGP\_SPEAKER**

ID or name of the BGP speaker.

This command is provided by the python-neutronclient plugin.

## bgp-peer-create

Create a BGP Peer.

```

neutron bgp-peer-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  --peer-ip PEER_IP_ADDRESS
  --remote-as PEER_REMOTE_AS
  [--auth-type PEER_AUTH_TYPE]
  [--password AUTH_PASSWORD]
  NAME

```

**-f** <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

**-c** COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

**--noindent**

whether to disable indenting the JSON

**--prefix** <PREFIX>

add a prefix to all variable names

**--max-width** <integer>

Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**--tenant-id** TENANT\_ID

The owner tenant ID.

**--peer-ip** PEER\_IP\_ADDRESS

Peer IP address.

**--remote-as** PEER\_REMOTE\_AS

Peer AS number. (Integer in [1, 4294967295] is allowed.)

**--auth-type** PEER\_AUTH\_TYPE

Authentication algorithm. Supported algorithms: none(default), md5

**--password** AUTH\_PASSWORD

Authentication password.

**NAME**

Name of the BGP peer to create.

This command is provided by the python-neutronclient plugin.

### bgp-peer-delete

Delete a BGP peer.

```
neutron bgp-peer-delete BGP_PEER [BGP_PEER ...]
```

#### **BGP\_PEER**

ID(s) or name(s) of bgp\_peer to delete.

This command is provided by the python-neutronclient plugin.

### bgp-peer-list

List BGP peers.

```
neutron bgp-peer-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.



- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- P** SIZE, **--page-size** SIZE  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key** FIELD  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.
- sort-dir** {asc, desc}  
Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## bgp-peer-show

Show information of a given BGP peer.

```
neutron bgp-peer-show
  [-f {json, shell, table, value, yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  BGP_PEER
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>

Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**-D, --show-details**

Show detailed information.

**-F FIELD, --field FIELD**

Specify the field(s) to be returned by server. You can repeat this option.

**BGP\_PEER**

ID or name of bgp\_peer to look up.

This command is provided by the python-neutronclient plugin.

## bgp-peer-update

Update BGP Peers information.

```
neutron bgp-peer-update
  [--name NAME]
  [--password AUTH_PASSWORD]
  BGP_PEER
```

**--name** <NAME>

Updated name of the BGP peer.

**--password** AUTH\_PASSWORD

Updated authentication password.

**BGP\_PEER**

ID or name of bgp\_peer to update.

This command is provided by the python-neutronclient plugin.

## bgp-speaker-advertiseroute-list

List routes advertised by a given BGP speaker.

```
neutron bgp-speaker-advertiseroute-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
```

(continues on next page)

(continued from previous page)

```

[--sort-ascending | --sort-descending]
[-D]
[-F FIELD]
[-P SIZE]
[--sort-key FIELD]
[--sort-dir {asc,desc}]
BGP_SPEAKER

```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric
- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- P** SIZE, **--page-size** SIZE  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key** FIELD  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.
- sort-dir** {asc,desc}  
Sorts the list in the specified direction. You can repeat this option.

**BGP\_SPEAKER**

ID or name of the BGP speaker.

This command is provided by the python-neutronclient plugin.

**bgp-speaker-create**

Create a BGP Speaker.

```
neutron bgp-speaker-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  --local-as LOCAL_AS
  [--ip-version {4,6}]
  [--advertise-floating-ip-host-routes {True,False}]
  [--advertise-tenant-networks {True,False}]
  NAME
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--tenant-id** TENANT\_ID  
The owner tenant ID.

**--local-as** LOCAL\_AS  
Local AS number. (Integer in [1, 4294967295] is allowed.)

**--ip-version** <IP\_VERSION>  
IP version for the BGP speaker (default is 4).

**--advertise-floating-ip-host-routes** {True,False}

Whether to enable or disable the advertisement of floating-ip host routes by the BGP speaker. By default floating ip host routes will be advertised by the BGP speaker.

**--advertise-tenant-networks** {True,False}

Whether to enable or disable the advertisement of tenant network routes by the BGP speaker. By default tenant network routes will be advertised by the BGP speaker.

#### NAME

Name of the BGP speaker to create.

This command is provided by the python-neutronclient plugin.

### bgp-speaker-delete

Delete a BGP speaker.

```
neutron bgp-speaker-delete BGP_SPEAKER [BGP_SPEAKER ...]
```

#### BGP\_SPEAKER

ID(s) or name(s) of bgp\_speaker to delete.

This command is provided by the python-neutronclient plugin.

### bgp-speaker-list

List BGP speakers.

```
neutron bgp-speaker-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- P** SIZE, **--page-size** SIZE  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key** FIELD  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.
- sort-dir** {asc,desc}  
Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## bgp-speaker-list-on-dragent

List BGP speakers hosted by a Dynamic Routing agent.

```
neutron bgp-speaker-list-on-dragent
[-f {csv,json,table,value,yaml}]
[-c COLUMN]
[--quote {all,minimal,none,nonnumeric}]
[--noindent]
[--max-width <integer>]
[--fit-width]
[--print-empty]
[--sort-column SORT_COLUMN]
[--sort-ascending | --sort-descending]
[-D]
[-F FIELD]
BGP_DRAGENT_ID
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric
- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- BGP\_DRAGENT\_ID**  
ID of the Dynamic Routing agent.

This command is provided by the python-neutronclient plugin.

### bgp-speaker-network-add

Add a network to the BGP speaker.

```
neutron bgp-speaker-network-add BGP_SPEAKER NETWORK
```

**BGP\_SPEAKER**  
ID or name of the BGP speaker.

**NETWORK**  
ID or name of the network to add.

This command is provided by the python-neutronclient plugin.

## bgp-speaker-network-remove

Remove a network from the BGP speaker.

```
neutron bgp-speaker-network-remove BGP_SPEAKER NETWORK
```

### **BGP\_SPEAKER**

ID or name of the BGP speaker.

### **NETWORK**

ID or name of the network to remove.

This command is provided by the python-neutronclient plugin.

## bgp-speaker-peer-add

Add a peer to the BGP speaker.

```
neutron bgp-speaker-peer-add BGP_SPEAKER BGP_PEER
```

### **BGP\_SPEAKER**

ID or name of the BGP speaker.

### **BGP\_PEER**

ID or name of the BGP peer to add.

This command is provided by the python-neutronclient plugin.

## bgp-speaker-peer-remove

Remove a peer from the BGP speaker.

```
neutron bgp-speaker-peer-remove BGP_SPEAKER BGP_PEER
```

### **BGP\_SPEAKER**

ID or name of the BGP speaker.

### **BGP\_PEER**

ID or name of the BGP peer to remove.

This command is provided by the python-neutronclient plugin.

## bgp-speaker-show

Show information of a given BGP speaker.

```
neutron bgp-speaker-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
```

(continues on next page)



(continued from previous page)

```

[--fit-width]
[--print-empty]
[-D]
[-F FIELD]
BGP_SPEAKER

```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- BGP\_SPEAKER**  
ID or name of bgp\_speaker to look up.

This command is provided by the python-neutronclient plugin.

## bgp-speaker-update

Update BGP Speakers information.

```

neutron bgp-speaker-update
  [--name NAME]
  [--advertise-floating-ip-host-routes {True,False}]
  [--advertise-tenant-networks {True,False}]
  BGP_SPEAKER

```

- name** <NAME>  
Name of the BGP speaker to update.
- advertise-floating-ip-host-routes** {True,False}  
Whether to enable or disable the advertisement of floating-ip host routes by the BGP speaker. By default floating ip host routes will be advertised by the BGP speaker.

**--advertise-tenant-networks** {True,False}

Whether to enable or disable the advertisement of tenant network routes by the BGP speaker. By default tenant network routes will be advertised by the BGP speaker.

**BGP\_SPEAKER**

ID or name of bgp\_speaker to update.

This command is provided by the python-neutronclient plugin.

## dhcp-agent-list-hosting-net

List DHCP agents hosting a network.

```
neutron dhcp-agent-list-hosting-net
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  NETWORK
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

**--sort-ascending**  
sort the column(s) in ascending order

- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F FIELD, --field FIELD**  
Specify the field(s) to be returned by server. You can repeat this option.

**NETWORK**

Network to query.

This command is provided by the python-neutronclient plugin.

**dhcp-agent-network-add**

Add a network to a DHCP agent.

```
neutron dhcp-agent-network-add DHCP_AGENT NETWORK
```

**DHCP\_AGENT**

ID of the DHCP agent.

**NETWORK**

Network to add.

This command is provided by the python-neutronclient plugin.

**dhcp-agent-network-remove**

Remove a network from a DHCP agent.

```
neutron dhcp-agent-network-remove DHCP_AGENT NETWORK
```

**DHCP\_AGENT**

ID of the DHCP agent.

**NETWORK**

Network to remove.

This command is provided by the python-neutronclient plugin.

**ext-list**

List all extensions.

```
neutron ext-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
```

(continues on next page)

(continued from previous page)

```

[--sort-column SORT_COLUMN]
[--sort-ascending | --sort-descending]
[-D]
[-F FIELD]

```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric
- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## ext-show

Show information of a given resource.

```

neutron ext-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]

```

(continues on next page)

(continued from previous page)

```

[--fit-width]
[--print-empty]
[-D]
[-F FIELD]
EXTENSION

```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- D**, **--show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- EXTENSION**  
ID of extension to look up.

This command is provided by the python-neutronclient plugin.

## firewall-create

Create a firewall.

```

neutron firewall-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  [--name NAME]
  [--description DESCRIPTION]

```

(continues on next page)

(continued from previous page)

```
[--router ROUTER | --no-routers]
[--admin-state-down]
POLICY
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
  - c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
  - noindent**  
whether to disable indenting the JSON
  - prefix** <PREFIX>  
add a prefix to all variable names
  - max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
  - fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
  - print-empty**  
Print empty table if there is no data to show.
  - tenant-id** TENANT\_ID  
The owner tenant ID.
  - name** <NAME>  
Name for the firewall.
  - description** <DESCRIPTION>  
Description for the firewall.
  - router** ROUTER  
ID or name of the router associated with the firewall (requires FWaaS router insertion extension to be enabled). This option can be repeated.
  - no-routers**  
Associate no routers with the firewall (requires FWaaS router insertion extension).
  - admin-state-down**  
Set admin state up to false.
- POLICY**  
ID or name of the firewall policy associated to this firewall.

This command is provided by the python-neutronclient plugin.

## firewall-delete

Delete a given firewall.

```
neutron firewall-delete FIREWALL [FIREWALL ...]
```

### **FIREWALL**

ID(s) or name(s) of firewall to delete.

This command is provided by the python-neutronclient plugin.

## firewall-list

List firewalls that belong to a given tenant.

```
neutron firewall-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F FIELD, --field FIELD**  
Specify the field(s) to be returned by server. You can repeat this option.
- P SIZE, --page-size SIZE**  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key FIELD**  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of `sort_dir` and `sort_key` values. Extra `sort_dir` options are ignored. Missing `sort_dir` options use the default `asc` value.
- sort-dir {asc, desc}**  
Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## firewall-policy-create

Create a firewall policy.

```
neutron firewall-policy-create
  [-f {json, shell, table, value, yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  [--shared]
  [--audited]
  [--description DESCRIPTION]
  [--firewall-rules FIREWALL_RULES]
  NAME
```

- f <FORMATTER>, --format <FORMATTER>**  
the output format, defaults to `table`
- c COLUMN, --column COLUMN**  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix <PREFIX>**  
add a prefix to all variable names



- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- tenant-id** TENANT\_ID  
The owner tenant ID.
- shared**  
Create a shared policy.
- audited**  
Sets audited to True.
- description** <DESCRIPTION>  
Description for the firewall policy.
- firewall-rules** <FIREWALL\_RULES>  
Ordered list of whitespace-delimited firewall rule names or IDs; e.g., firewall-rules rule1 rule2

**NAME**

Name for the firewall policy.

This command is provided by the python-neutronclient plugin.

**firewall-policy-delete**

Delete a given firewall policy.

```
neutron firewall-policy-delete FIREWALL_POLICY [FIREWALL_POLICY ...]
```

**FIREWALL\_POLICY**

ID(s) or name(s) of firewall\_policy to delete.

This command is provided by the python-neutronclient plugin.

**firewall-policy-insert-rule**

Insert a rule into a given firewall policy.

```
neutron firewall-policy-insert-rule
  [--insert-before FIREWALL_RULE]
  [--insert-after FIREWALL_RULE]
  FIREWALL_POLICY
  FIREWALL_RULE
```

- insert-before** FIREWALL\_RULE  
Insert before this rule.

**--insert-after** FIREWALL\_RULE  
Insert after this rule.

**FIREWALL\_POLICY**  
ID or name of firewall\_policy to update.

**FIREWALL\_RULE**  
New rule to insert.

This command is provided by the python-neutronclient plugin.

## firewall-policy-list

List firewall policies that belong to a given tenant.

```
neutron firewall-policy-list
[-f {csv,json,table,value,yaml}]
[-c COLUMN]
[--quote {all,minimal,none,nonnumeric}]
[--noindent]
[--max-width <integer>]
[--fit-width]
[--print-empty]
[--sort-column SORT_COLUMN]
[--sort-ascending | --sort-descending]
[-D]
[-F FIELD]
[-P SIZE]
[--sort-key FIELD]
[--sort-dir {asc,desc}]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F FIELD, --field FIELD**  
Specify the field(s) to be returned by server. You can repeat this option.
- P SIZE, --page-size SIZE**  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key FIELD**  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of `sort_dir` and `sort_key` values. Extra `sort_dir` options are ignored. Missing `sort_dir` options use the default `asc` value.
- sort-dir {asc,desc}**  
Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

### firewall-policy-remove-rule

Remove a rule from a given firewall policy.

```
neutron firewall-policy-remove-rule FIREWALL_POLICY FIREWALL_RULE
```

#### **FIREWALL\_POLICY**

ID or name of `firewall_policy` to update.

#### **FIREWALL\_RULE**

ID or name of the firewall rule to be removed from the policy.

This command is provided by the python-neutronclient plugin.

### firewall-policy-show

Show information of a given firewall policy.

```
neutron firewall-policy-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  FIREWALL_POLICY
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- D**, **--show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.

#### **FIREWALL\_POLICY**

ID or name of firewall\_policy to look up.

This command is provided by the python-neutronclient plugin.

### **firewall-policy-update**

Update a given firewall policy.

```
neutron firewall-policy-update
  [--description DESCRIPTION]
  [--firewall-rules FIREWALL_RULES]
  [--name NAME]
  [--shared {True,False}]
  [--audited {True,False}]
  FIREWALL_POLICY
```

- description** <DESCRIPTION>  
Description for the firewall policy.
- firewall-rules** <FIREWALL\_RULES>  
Ordered list of whitespace-delimited firewall rule names or IDs; e.g., firewall-rules rule1 rule2
- name** <NAME>  
Name for the firewall policy.
- shared** {True,False}  
Update the sharing status of the policy. (True means shared).

**--audited** {True,False}  
Update the audit status of the policy. (True means auditing is enabled).

#### **FIREWALL\_POLICY**

ID or name of firewall\_policy to update.

This command is provided by the python-neutronclient plugin.

### **firewall-rule-create**

Create a firewall rule.

```
neutron firewall-rule-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  [--shared]
  [--name NAME]
  [--description DESCRIPTION]
  [--source-ip-address SOURCE_IP_ADDRESS]
  [--destination-ip-address DESTINATION_IP_ADDRESS]
  [--source-port SOURCE_PORT]
  [--destination-port DESTINATION_PORT]
  [--enabled {True,False}]
  --protocol
  {tcp,udp,icmp,any}
  --action
  {allow,deny,reject}
  [--ip-version {4,6}]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--tenant-id** TENANT\_ID  
The owner tenant ID.

**--shared**  
Set shared flag for the firewall rule.

**--name** <NAME>  
Name for the firewall rule.

**--description** <DESCRIPTION>  
Description for the firewall rule.

**--source-ip-address** <SOURCE\_IP\_ADDRESS>  
Source IP address or subnet.

**--destination-ip-address** <DESTINATION\_IP\_ADDRESS>  
Destination IP address or subnet.

**--source-port** <SOURCE\_PORT>  
Source port (integer in [1, 65535] or range in a:b).

**--destination-port** <DESTINATION\_PORT>  
Destination port (integer in [1, 65535] or range in a:b).

**--enabled** {True,False}  
Whether to enable or disable this rule.

**--protocol** <PROTOCOL>  
Protocol for the firewall rule.

**--action** <ACTION>  
Action for the firewall rule.

**--ip-version** <IP\_VERSION>  
IP version for the firewall rule (default is 4).

This command is provided by the python-neutronclient plugin.

### firewall-rule-delete

Delete a given firewall rule.

```
neutron firewall-rule-delete FIREWALL_RULE [FIREWALL_RULE ...]
```

#### **FIREWALL\_RULE**

ID(s) or name(s) of firewall\_rule to delete.

This command is provided by the python-neutronclient plugin.

## firewall-rule-list

List firewall rules that belong to a given tenant.

```
neutron firewall-rule-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric
- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D**, **--show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.

**-P** SIZE, **--page-size** SIZE

Specify retrieve unit of each request, then split one request to several requests.

**--sort-key** FIELD

Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.

**--sort-dir** {asc,desc}

Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## firewall-rule-show

Show information of a given firewall rule.

```
neutron firewall-rule-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  FIREWALL_RULE
```

**-f** <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

**-c** COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

**--noindent**

whether to disable indenting the JSON

**--prefix** <PREFIX>

add a prefix to all variable names

**--max-width** <integer>

Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**-D**, **--show-details**

Show detailed information.

**-F** FIELD, **--field** FIELD

Specify the field(s) to be returned by server. You can repeat this option.



**FIREWALL\_RULE**

ID or name of firewall\_rule to look up.

This command is provided by the python-neutronclient plugin.

**firewall-rule-update**

Update a given firewall rule.

```
neutron firewall-rule-update
  [--shared {True,False}]
  [--ip-version {4,6}]
  [--name NAME]
  [--description DESCRIPTION]
  [--source-ip-address SOURCE_IP_ADDRESS]
  [--destination-ip-address DESTINATION_IP_ADDRESS]
  [--source-port SOURCE_PORT]
  [--destination-port DESTINATION_PORT]
  [--enabled {True,False}]
  [--protocol {tcp,udp,icmp,any}]
  [--action {allow,deny,reject}]
  FIREWALL_RULE
```

**--shared** {True,False}

Update the shared flag for the firewall rule.

**--ip-version** <IP\_VERSION>

Update IP version for the firewall rule.

**--name** <NAME>

Name for the firewall rule.

**--description** <DESCRIPTION>

Description for the firewall rule.

**--source-ip-address** <SOURCE\_IP\_ADDRESS>

Source IP address or subnet.

**--destination-ip-address** <DESTINATION\_IP\_ADDRESS>

Destination IP address or subnet.

**--source-port** <SOURCE\_PORT>

Source port (integer in [1, 65535] or range in a:b).

**--destination-port** <DESTINATION\_PORT>

Destination port (integer in [1, 65535] or range in a:b).

**--enabled** {True,False}

Whether to enable or disable this rule.

**--protocol** <PROTOCOL>

Protocol for the firewall rule.

**--action** <ACTION>

Action for the firewall rule.

**FIREWALL\_RULE**

ID or name of firewall\_rule to update.

This command is provided by the python-neutronclient plugin.

## firewall-show

Show information of a given firewall.

```
neutron firewall-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  FIREWALL
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**-D**, **--show-details**  
Show detailed information.

**-F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.

**FIREWALL**  
ID or name of firewall to look up.

This command is provided by the python-neutronclient plugin.

## firewall-update

Update a given firewall.

```
neutron firewall-update
  [--name NAME]
  [--description DESCRIPTION]
  [--router ROUTER | --no-routers]
  [--policy POLICY]
  [--admin-state-up {True,False}]
  FIREWALL
```

**--name** <NAME>

Name for the firewall.

**--description** <DESCRIPTION>

Description for the firewall.

**--router** ROUTER

ID or name of the router associated with the firewall (requires FWaaS router insertion extension to be enabled). This option can be repeated.

**--no-routers**

Associate no routers with the firewall (requires FWaaS router insertion extension).

**--policy** POLICY

ID or name of the firewall policy associated to this firewall.

**--admin-state-up** {True,False}

Update the admin state for the firewall (True means UP).

**FIREWALL**

ID or name of firewall to update.

This command is provided by the python-neutronclient plugin.

## flavor-associate

Associate a Neutron service flavor with a flavor profile.

```
neutron flavor-associate FLAVOR FLAVOR_PROFILE
```

**FLAVOR**

ID or name of the flavor to associate.

**FLAVOR\_PROFILE**

ID of the flavor profile to be associated with the flavor.

This command is provided by the python-neutronclient plugin.

## flavor-create

Create a Neutron service flavor.

```
neutron flavor-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  [--description DESCRIPTION]
  [--enabled {True,False}]
  NAME
  SERVICE_TYPE
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--tenant-id** TENANT\_ID  
The owner tenant ID.

**--description** <DESCRIPTION>  
Description for the flavor.

**--enabled** {True,False}  
Sets enabled flag.

**NAME**  
Name for the flavor.

**SERVICE\_TYPE**  
Service type to which the flavor applies to: e.g. VPN. (See service-provider-list for loaded examples.)

This command is provided by the python-neutronclient plugin.

## flavor-delete

Delete a given Neutron service flavor.

```
neutron flavor-delete FLAVOR [FLAVOR ...]
```

### FLAVOR

ID(s) or name(s) of flavor to delete.

This command is provided by the python-neutronclient plugin.

## flavor-disassociate

Disassociate a Neutron service flavor from a flavor profile.

```
neutron flavor-disassociate FLAVOR FLAVOR_PROFILE
```

### FLAVOR

ID or name of the flavor to be disassociated.

### FLAVOR\_PROFILE

ID of the flavor profile to be disassociated from the flavor.

This command is provided by the python-neutronclient plugin.

## flavor-list

List Neutron service flavors.

```
neutron flavor-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- P** SIZE, **--page-size** SIZE  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key** FIELD  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.
- sort-dir** {asc,desc}  
Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## flavor-profile-create

Create a Neutron service flavor profile.

```
neutron flavor-profile-create
[-f {json,shell,table,value,yaml}]
[-c COLUMN]
[--noindent]
[--prefix PREFIX]
[--max-width <integer>]
[--fit-width]
[--print-empty]
[--tenant-id TENANT_ID]
[--description DESCRIPTION]
```

(continues on next page)

(continued from previous page)

```
[--driver DRIVER]
[--metainfo METAINFO]
[--enabled {True,False}]
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- tenant-id** TENANT\_ID  
The owner tenant ID.
- description** <DESCRIPTION>  
Description for the flavor profile.
- driver** <DRIVER>  
Python module path to driver.
- metainfo** <METAINFO>  
Metainfo for the flavor profile.
- enabled** {True,False}  
Sets enabled flag.

This command is provided by the python-neutronclient plugin.

### flavor-profile-delete

Delete a given Neutron service flavor profile.

```
neutron flavor-profile-delete SERVICE_PROFILE [SERVICE_PROFILE ...]
```

#### SERVICE\_PROFILE

ID(s) or name(s) of service\_profile to delete.

This command is provided by the python-neutronclient plugin.

## flavor-profile-list

List Neutron service flavor profiles.

```
neutron flavor-profile-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

**--sort-ascending**  
sort the column(s) in ascending order

**--sort-descending**  
sort the column(s) in descending order

**-D**, **--show-details**  
Show detailed information.

**-F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.



- P** SIZE, **--page-size** SIZE  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key** FIELD  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.
- sort-dir** {asc,desc}  
Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## flavor-profile-show

Show information about a given Neutron service flavor profile.

```
neutron flavor-profile-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  SERVICE_PROFILE
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- D**, **--show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.

#### **SERVICE\_PROFILE**

ID or name of service\_profile to look up.

This command is provided by the python-neutronclient plugin.

### **flavor-profile-update**

Update a given Neutron service flavor profile.

```
neutron flavor-profile-update
  [--description DESCRIPTION]
  [--driver DRIVER]
  [--metainfo METAINFO]
  [--enabled {True,False}]
  SERVICE_PROFILE
```

**--description** <DESCRIPTION>

Description for the flavor profile.

**--driver** <DRIVER>

Python module path to driver.

**--metainfo** <METAINFO>

Metainfo for the flavor profile.

**--enabled** {True,False}

Sets enabled flag.

#### **SERVICE\_PROFILE**

ID or name of service\_profile to update.

This command is provided by the python-neutronclient plugin.

### **flavor-show**

Show information about a given Neutron service flavor.

```
neutron flavor-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  FLAVOR
```

**-f** <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

**-c** COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- FLAVOR**  
ID or name of flavor to look up.

This command is provided by the python-neutronclient plugin.

## flavor-update

Update a Neutron service flavor.

```
neutron flavor-update
  [--name NAME]
  [--description DESCRIPTION]
  [--enabled {True,False}]
  FLAVOR
```

- name** <NAME>  
Name for the flavor.
- description** <DESCRIPTION>  
Description for the flavor.
- enabled** {True,False}  
Sets enabled flag.
- FLAVOR**  
ID or name of flavor to update.

This command is provided by the python-neutronclient plugin.

## floatingip-associate

Create a mapping between a floating IP and a fixed IP.

```
neutron floatingip-associate
  [--fixed-ip-address FIXED_IP_ADDRESS]
  FLOATINGIP_ID
  PORT
```

**--fixed-ip-address** <FIXED\_IP\_ADDRESS>  
IP address on the port (only required if port has multiple IPs).

**FLOATINGIP\_ID**  
ID of the floating IP to associate.

**PORT**  
ID or name of the port to be associated with the floating IP.

This command is provided by the python-neutronclient plugin.

## floatingip-create

Create a floating IP for a given tenant.

```
neutron floatingip-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  [--description DESCRIPTION]
  [--port-id PORT_ID]
  [--fixed-ip-address FIXED_IP_ADDRESS]
  [--floating-ip-address FLOATING_IP_ADDRESS]
  [--subnet SUBNET_ID]
  [--dns-domain DNS_DOMAIN]
  [--dns-name DNS_NAME]
  FLOATING_NETWORK
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**--tenant-id** TENANT\_ID

The owner tenant ID.

**--description** <DESCRIPTION>

Description of the floating IP.

**--port-id** <PORT\_ID>

ID of the port to be associated with the floating IP.

**--fixed-ip-address** <FIXED\_IP\_ADDRESS>

IP address on the port (only required if port has multiple IPs).

**--floating-ip-address** <FLOATING\_IP\_ADDRESS>

IP address of the floating IP

**--subnet** <SUBNET\_ID>

Subnet ID on which you want to create the floating IP.

**--dns-domain** <DNS\_DOMAIN>

Assign DNS domain to the floatingip (requires DNS integration extension)

**--dns-name** <DNS\_NAME>

Assign DNS name to the floatingip (requires DNS integration extension)

**FLOATING\_NETWORK**

ID or name of the network from which the floating IP is allocated.

This command is provided by the python-neutronclient plugin.

## floatingip-delete

Delete a given floating IP.

```
neutron floatingip-delete FLOATINGIP [FLOATINGIP ...]
```

**FLOATINGIP**

ID(s) of floatingip to delete.

This command is provided by the python-neutronclient plugin.

## floatingip-disassociate

Remove a mapping from a floating IP to a fixed IP.

```
neutron floatingip-disassociate FLOATINGIP_ID
```

### FLOATINGIP\_ID

ID of the floating IP to disassociate.

This command is provided by the python-neutronclient plugin.

## floatingip-list

List floating IPs that belong to a given tenant.

```
neutron floatingip-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F FIELD, --field FIELD**  
Specify the field(s) to be returned by server. You can repeat this option.
- P SIZE, --page-size SIZE**  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key FIELD**  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of `sort_dir` and `sort_key` values. Extra `sort_dir` options are ignored. Missing `sort_dir` options use the default `asc` value.
- sort-dir {asc, desc}**  
Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## floatingip-show

Show information of a given floating IP.

```
neutron floatingip-show
  [-f {json, shell, table, value, yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  FLOATINGIP
```

- f <FORMATTER>, --format <FORMATTER>**  
the output format, defaults to table
- c COLUMN, --column COLUMN**  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix <PREFIX>**  
add a prefix to all variable names
- max-width <integer>**  
Maximum display width, <1 to disable. You can also use the `CLIFF_MAX_TERM_WIDTH` environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**-D, --show-details**

Show detailed information.

**-F FIELD, --field FIELD**

Specify the field(s) to be returned by server. You can repeat this option.

**FLOATINGIP**

ID of floatingip to look up.

This command is provided by the python-neutronclient plugin.

## ipsec-site-connection-create

Create an IPsec site connection.

```
neutron ipsec-site-connection-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  [--admin-state-down]
  --vpnservice-id VPNSERVICE
  --ikepolicy-id IKEPOLICY
  --ipsecpolicy-id IPSECPOLICY
  [--name NAME]
  [--description DESCRIPTION]
  [--dpd action=ACTION,interval=INTERVAL,timeout=TIMEOUT]
  [--local-ep-group LOCAL_EP_GROUP]
  [--peer-ep-group PEER_EP_GROUP]
  [--peer-cidr PEER_CIDRS]
  --peer-id PEER_ID
  --peer-address PEER_ADDRESS
  --psk PSK
  [--mtu MTU]
  [--initiator {bi-directional,response-only}]
```

**-f <FORMATTER>, --format <FORMATTER>**

the output format, defaults to table

**-c COLUMN, --column COLUMN**

specify the column(s) to include, can be repeated to show multiple columns

**--noindent**

whether to disable indenting the JSON



- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- tenant-id** TENANT\_ID  
The owner tenant ID.
- admin-state-down**  
Set admin state up to false.
- vpnservice-id** VPNSERVICE  
VPN service instance ID associated with this connection.
- ikepolicy-id** IKEPOLICY  
IKE policy ID associated with this connection.
- ipsecpolicy-id** IPSECPOLICY  
IPsec policy ID associated with this connection.
- name** <NAME>  
Set friendly name for the connection.
- description** <DESCRIPTION>  
Set a description for the connection.
- dpd** action=ACTION, interval=INTERVAL, timeout=TIMEOUT  
Ipssec connection. Dead Peer Detection attributes. action=hold,clear,disabled,restart,restart-by-peer. interval and timeout are non negative integers. interval should be less than timeout value. action, default:hold interval, default:30, timeout, default:120.
- local-ep-group** <LOCAL\_EP\_GROUP>  
Local endpoint group ID/name with subnet(s) for IPsec connection.
- peer-ep-group** <PEER\_EP\_GROUP>  
Peer endpoint group ID/name with CIDR(s) for IPsec connection.
- peer-cidr** <PEER\_CIDRS>  
[DEPRECATED in Mitaka] Remote subnet(s) in CIDR format. Cannot be specified when using endpoint groups. Only applicable, if subnet provided for VPN service.
- peer-id** <PEER\_ID>  
Peer router identity for authentication. Can be IPv4/IPv6 address, e-mail address, key id, or FQDN.
- peer-address** <PEER\_ADDRESS>  
Peer gateway public IPv4/IPv6 address or FQDN.
- psk** <PSK>  
Pre-shared key string.

**--mtu** <MTU>  
MTU size for the connection, default:1500.

**--initiator** <INITIATOR>  
Initiator state in lowercase, default:bi-directional

This command is provided by the python-neutronclient plugin.

### ipsec-site-connection-delete

Delete a given IPsec site connection.

```
neutron ipsec-site-connection-delete
    IPSEC_SITE_CONNECTION
    [IPSEC_SITE_CONNECTION ...]
```

**IPSEC\_SITE\_CONNECTION**  
ID(s) or name(s) of IPsec site connection to delete.

This command is provided by the python-neutronclient plugin.

### ipsec-site-connection-list

List IPsec site connections that belong to a given tenant.

```
neutron ipsec-site-connection-list
    [-f {csv,json,table,value,yaml}]
    [-c COLUMN]
    [--quote {all,minimal,none,nonnumeric}]
    [--noindent]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
    [-D]
    [-F FIELD]
    [-P SIZE]
    [--sort-key FIELD]
    [--sort-dir {asc,desc}]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- P** SIZE, **--page-size** SIZE  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key** FIELD  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.
- sort-dir** {asc, desc}  
Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## ipsec-site-connection-show

Show information of a given IPsec site connection.

```
neutron ipsec-site-connection-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  IPSEC_SITE_CONNECTION
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

- noindent**  
whether to disable indenting the JSON
  - prefix** <PREFIX>  
add a prefix to all variable names
  - max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
  - fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
  - print-empty**  
Print empty table if there is no data to show.
  - D, --show-details**  
Show detailed information.
  - F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- IPSEC\_SITE\_CONNECTION**  
ID or name of IPsec site connection to look up.

This command is provided by the python-neutronclient plugin.

### ipsec-site-connection-update

Update a given IPsec site connection.

```
neutron ipsec-site-connection-update
  [--admin-state-up {True,False}]
  [--name NAME]
  [--description DESCRIPTION]
  [--dpd action=ACTION,interval=INTERVAL,timeout=TIMEOUT]
  [--local-ep-group LOCAL_EP_GROUP]
  [--peer-ep-group PEER_EP_GROUP]
  [--peer-cidr PEER_CIDRS]
  [--peer-id PEER_ID]
  [--peer-address PEER_ADDRESS]
  [--psk PSK]
  [--mtu MTU]
  [--initiator {bi-directional,response-only}]
  IPSEC_SITE_CONNECTION
```

- admin-state-up** {True,False}  
Update the administrative state. (True meaning Up)
- name** <NAME>  
Set friendly name for the connection.
- description** <DESCRIPTION>  
Set a description for the connection.

- dpd** `action=ACTION, interval=INTERVAL, timeout=TIMEOUT`  
 Isec connection. Dead Peer Detection attributes. `action=hold,clear,disabled,restart,restart-by-peer`. `interval` and `timeout` are non negative integers. `interval` should be less than `timeout` value. `action`, default:hold `interval`, default:30, `timeout`, default:120.
- local-ep-group** `<LOCAL_EP_GROUP>`  
 Local endpoint group ID/name with subnet(s) for IPsec connection.
- peer-ep-group** `<PEER_EP_GROUP>`  
 Peer endpoint group ID/name with CIDR(s) for IPsec connection.
- peer-cidr** `<PEER_CIDRS>`  
 [DEPRECATED in Mitaka] Remote subnet(s) in CIDR format. Cannot be specified when using endpoint groups. Only applicable, if subnet provided for VPN service.
- peer-id** `<PEER_ID>`  
 Peer router identity for authentication. Can be IPv4/IPv6 address, e-mail address, key id, or FQDN.
- peer-address** `<PEER_ADDRESS>`  
 Peer gateway public IPv4/IPv6 address or FQDN.
- psk** `<PSK>`  
 Pre-shared key string.
- mtu** `<MTU>`  
 MTU size for the connection, default:1500.
- initiator** `<INITIATOR>`  
 Initiator state in lowercase, default:bi-directional

**IPSEC\_SITE\_CONNECTION**

ID or name of IPsec site connection to update.

This command is provided by the python-neutronclient plugin.

**l3-agent-list-hosting-router**

List L3 agents hosting a router.

```
neutron l3-agent-list-hosting-router
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  ROUTER
```

- f** `<FORMATTER>`, **--format** `<FORMATTER>`  
 the output format, defaults to table

- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric
- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.

**ROUTER**

Router to query.

This command is provided by the python-neutronclient plugin.

### **l3-agent-router-add**

Add a router to a L3 agent.

```
neutron l3-agent-router-add L3_AGENT ROUTER
```

**L3\_AGENT**

ID of the L3 agent.

**ROUTER**

Router to add.

This command is provided by the python-neutronclient plugin.

### l3-agent-router-remove

Remove a router from a L3 agent.

```
neutron l3-agent-router-remove L3_AGENT ROUTER
```

#### L3\_AGENT

ID of the L3 agent.

#### ROUTER

Router to remove.

This command is provided by the python-neutronclient plugin.

### lb-agent-hosting-pool

Get loadbalancer agent hosting a pool.

Deriving from ListCommand though server will return only one agent to keep common output format for all agent schedulers

```
neutron lb-agent-hosting-pool
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  POOL
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.

**POOL**  
Pool to query.

This command is provided by the python-neutronclient plugin.

### lb-healthmonitor-associate

Create a mapping between a health monitor and a pool.

```
neutron lb-healthmonitor-associate HEALTH_MONITOR_ID POOL
```

**HEALTH\_MONITOR\_ID**  
Health monitor to associate.

**POOL**  
ID of the pool to be associated with the health monitor.

This command is provided by the python-neutronclient plugin.

### lb-healthmonitor-create

Create a health monitor.

```
neutron lb-healthmonitor-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  [--admin-state-down]
  [--expected-codes EXPECTED_CODES]
  [--http-method HTTP_METHOD]
  [--url-path URL_PATH]
  --delay DELAY
  --max-retries MAX_RETRIES
  --timeout TIMEOUT
```

(continues on next page)



(continued from previous page)

```
--type
{PING,TCP,HTTP,HTTPS}
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- tenant-id** TENANT\_ID  
The owner tenant ID.
- admin-state-down**  
Set admin state up to false.
- expected-codes** <EXPECTED\_CODES>  
The list of HTTP status codes expected in response from the member to declare it healthy. This attribute can contain one value, or a list of values separated by comma, or a range of values (e.g. 200-299). If this attribute is not specified, it defaults to 200.
- http-method** <HTTP\_METHOD>  
The HTTP method used for requests by the monitor of type HTTP.
- url-path** <URL\_PATH>  
The HTTP path used in the HTTP request used by the monitor to test a member health. This must be a string beginning with a / (forward slash).
- delay** <DELAY>  
The time in seconds between sending probes to members.
- max-retries** <MAX\_RETRIES>  
Number of permissible connection failures before changing the member status to INACTIVE. [1..10]
- timeout** <TIMEOUT>  
Maximum number of seconds for a monitor to wait for a connection to be established before it times out. The value must be less than the delay value.
- type** <TYPE>  
One of the predefined health monitor types.

This command is provided by the python-neutronclient plugin.

### lb-healthmonitor-delete

Delete a given health monitor.

```
neutron lb-healthmonitor-delete HEALTH_MONITOR [HEALTH_MONITOR ...]
```

#### HEALTH\_MONITOR

ID(s) of health\_monitor to delete.

This command is provided by the python-neutronclient plugin.

### lb-healthmonitor-disassociate

Remove a mapping from a health monitor to a pool.

```
neutron lb-healthmonitor-disassociate HEALTH_MONITOR_ID POOL
```

#### HEALTH\_MONITOR\_ID

Health monitor to disassociate.

#### POOL

ID of the pool to be disassociated with the health monitor.

This command is provided by the python-neutronclient plugin.

### lb-healthmonitor-list

List health monitors that belong to a given tenant.

```
neutron lb-healthmonitor-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

- quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric
- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- P** SIZE, **--page-size** SIZE  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key** FIELD  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.
- sort-dir** {asc,desc}  
Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## lb-healthmonitor-show

Show information of a given health monitor.

```
neutron lb-healthmonitor-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
```

(continues on next page)

(continued from previous page)

```
[--print-empty]
[-D]
[-F FIELD]
HEALTH_MONITOR
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- D**, **--show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.

**HEALTH\_MONITOR**

ID of health\_monitor to look up.

This command is provided by the python-neutronclient plugin.

**lb-healthmonitor-update**

Update a given health monitor.

```
neutron lb-healthmonitor-update HEALTH_MONITOR
```

**HEALTH\_MONITOR**

ID of health\_monitor to update.

This command is provided by the python-neutronclient plugin.

## lb-member-create

Create a member.

```
neutron lb-member-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  [--admin-state-down]
  [--weight WEIGHT]
  --address ADDRESS
  --protocol-port PROTOCOL_PORT
  POOL
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--tenant-id** TENANT\_ID  
The owner tenant ID.

**--admin-state-down**  
Set admin state up to false.

**--weight** <WEIGHT>  
Weight of pool member in the pool (default:1, [0..256]).

**--address** <ADDRESS>  
IP address of the pool member on the pool network.

**--protocol-port** <PROTOCOL\_PORT>  
Port on which the pool member listens for requests or connections.

**POOL**  
ID or name of the pool this vip belongs to.

This command is provided by the python-neutronclient plugin.

### lb-member-delete

Delete a given member.

```
neutron lb-member-delete MEMBER [MEMBER ...]
```

#### MEMBER

ID(s) or name(s) of member to delete.

This command is provided by the python-neutronclient plugin.

### lb-member-list

List members that belong to a given tenant.

```
neutron lb-member-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- P** SIZE, **--page-size** SIZE  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key** FIELD  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.
- sort-dir** {asc, desc}  
Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## lb-member-show

Show information of a given member.

```
neutron lb-member-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  MEMBER
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>

Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**-D, --show-details**

Show detailed information.

**-F FIELD, --field FIELD**

Specify the field(s) to be returned by server. You can repeat this option.

**MEMBER**

ID of member to look up.

This command is provided by the python-neutronclient plugin.

## lb-member-update

Update a given member.

```
neutron lb-member-update MEMBER
```

**MEMBER**

ID or name of member to update.

This command is provided by the python-neutronclient plugin.

## lb-pool-create

Create a pool.

```
neutron lb-pool-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  [--admin-state-down]
  [--description DESCRIPTION]
  --lb-method
  {ROUND_ROBIN,LEAST_CONNECTIONS,SOURCE_IP}
  --name NAME
  --protocol
  {HTTP,HTTPS,TCP}
  --subnet-id SUBNET
  [--provider PROVIDER]
```



- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- tenant-id** TENANT\_ID  
The owner tenant ID.
- admin-state-down**  
Set admin state up to false.
- description** <DESCRIPTION>  
Description of the pool.
- lb-method** <LB\_METHOD>  
The algorithm used to distribute load between the members of the pool.
- name** <NAME>  
The name of the pool.
- protocol** <PROTOCOL>  
Protocol for balancing.
- subnet-id** SUBNET  
The subnet on which the members of the pool will be located.
- provider** <PROVIDER>  
Provider name of the loadbalancer service.

This command is provided by the python-neutronclient plugin.

## lb-pool-delete

Delete a given pool.

```
neutron lb-pool-delete POOL [POOL ...]
```

### POOL

ID(s) or name(s) of pool to delete.

This command is provided by the python-neutronclient plugin.

## lb-pool-list

List pools that belong to a given tenant.

```
neutron lb-pool-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F FIELD, --field FIELD**  
Specify the field(s) to be returned by server. You can repeat this option.
- P SIZE, --page-size SIZE**  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key FIELD**  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of `sort_dir` and `sort_key` values. Extra `sort_dir` options are ignored. Missing `sort_dir` options use the default `asc` value.
- sort-dir {asc, desc}**  
Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## lb-pool-list-on-agent

List the pools on a loadbalancer agent.

```
neutron lb-pool-list-on-agent
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  LBAAS_AGENT
```

- f <FORMATTER>, --format <FORMATTER>**  
the output format, defaults to `table`
- c COLUMN, --column COLUMN**  
specify the column(s) to include, can be repeated to show multiple columns
- quote <QUOTE\_MODE>**  
when to include quotes, defaults to `nonnumeric`
- noindent**  
whether to disable indenting the JSON
- max-width <integer>**  
Maximum display width, `<1` to disable. You can also use the `CLIFF_MAX_TERM_WIDTH` environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN

specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

**--sort-ascending**

sort the column(s) in ascending order

**--sort-descending**

sort the column(s) in descending order

**-D, --show-details**

Show detailed information.

**-F** FIELD, **--field** FIELD

Specify the field(s) to be returned by server. You can repeat this option.

**LBAAS\_AGENT**

ID of the loadbalancer agent to query.

This command is provided by the python-neutronclient plugin.

## lb-pool-show

Show information of a given pool.

```
neutron lb-pool-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  POOL
```

**-f** <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

**-c** COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

**--noindent**

whether to disable indenting the JSON

**--prefix** <PREFIX>

add a prefix to all variable names

- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- D, --show-details**  
Show detailed information.
- F FIELD, --field FIELD**  
Specify the field(s) to be returned by server. You can repeat this option.

**POOL**

ID or name of pool to look up.

This command is provided by the python-neutronclient plugin.

**lb-pool-stats**

Retrieve stats for a given pool.

```
neutron lb-pool-stats
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  POOL
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**-D, --show-details**

Show detailed information.

**-F FIELD, --field FIELD**

Specify the field(s) to be returned by server. You can repeat this option.

**POOL**

ID or name of pool to look up.

This command is provided by the python-neutronclient plugin.

## lb-pool-update

Update a given pool.

```
neutron lb-pool-update POOL
```

**POOL**

ID or name of pool to update.

This command is provided by the python-neutronclient plugin.

## lb-vip-create

Create a vip.

```
neutron lb-vip-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  [--address ADDRESS]
  [--admin-state-down]
  [--connection-limit CONNECTION_LIMIT]
  [--description DESCRIPTION]
  --name NAME
  --protocol-port PROTOCOL_PORT
  --protocol
  {TCP,HTTP,HTTPS}
  --subnet-id SUBNET
  POOL
```

**-f <FORMATTER>, --format <FORMATTER>**  
the output format, defaults to table

**-c COLUMN, --column COLUMN**  
specify the column(s) to include, can be repeated to show multiple columns

- noindent**  
whether to disable indenting the JSON
  - prefix** <PREFIX>  
add a prefix to all variable names
  - max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
  - fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
  - print-empty**  
Print empty table if there is no data to show.
  - tenant-id** TENANT\_ID  
The owner tenant ID.
  - address** <ADDRESS>  
IP address of the vip.
  - admin-state-down**  
Set admin state up to false.
  - connection-limit** <CONNECTION\_LIMIT>  
The maximum number of connections per second allowed for the vip. Valid values: a positive integer or -1 for unlimited (default).
  - description** <DESCRIPTION>  
Description of the vip to be created.
  - name** <NAME>  
Name of the vip to be created.
  - protocol-port** <PROTOCOL\_PORT>  
TCP port on which to listen for client traffic that is associated with the vip address.
  - protocol** <PROTOCOL>  
Protocol for balancing.
  - subnet-id** SUBNET  
The subnet on which to allocate the vip address.
- POOL**  
ID or name of the pool to which this vip belongs.

This command is provided by the python-neutronclient plugin.

## lb-vip-delete

Delete a given vip.

```
neutron lb-vip-delete VIP [VIP ...]
```

### VIP

ID(s) or name(s) of vip to delete.

This command is provided by the python-neutronclient plugin.

## lb-vip-list

List vips that belong to a given tenant.

```
neutron lb-vip-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated



- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F FIELD, --field FIELD**  
Specify the field(s) to be returned by server. You can repeat this option.
- P SIZE, --page-size SIZE**  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key FIELD**  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of `sort_dir` and `sort_key` values. Extra `sort_dir` options are ignored. Missing `sort_dir` options use the default `asc` value.
- sort-dir {asc, desc}**  
Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## lb-vip-show

Show information of a given vip.

```
neutron lb-vip-show
  [-f {json, shell, table, value, yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
VIP
```

- f <FORMATTER>, --format <FORMATTER>**  
the output format, defaults to table
- c COLUMN, --column COLUMN**  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix <PREFIX>**  
add a prefix to all variable names
- max-width <integer>**  
Maximum display width, <1 to disable. You can also use the `CLIFF_MAX_TERM_WIDTH` environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**-D, --show-details**

Show detailed information.

**-F FIELD, --field FIELD**

Specify the field(s) to be returned by server. You can repeat this option.

**VIP**

ID or name of vip to look up.

This command is provided by the python-neutronclient plugin.

## lb-vip-update

Update a given vip.

```
neutron lb-vip-update VIP
```

**VIP**

ID or name of vip to update.

This command is provided by the python-neutronclient plugin.

## lbaas-agent-hosting-loadbalancer

Get lbaas v2 agent hosting a loadbalancer.

Deriving from ListCommand though server will return only one agent to keep common output format for all agent schedulers

```
neutron lbaas-agent-hosting-loadbalancer
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  LOADBALANCER
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

- quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric
- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.

**LOADBALANCER**

LoadBalancer to query.

This command is provided by the python-neutronclient plugin.

**lbaas-healthmonitor-create**

LBaaS v2 Create a healthmonitor.

```
neutron lbaas-healthmonitor-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  --delay DELAY
  [--name NAME]
  --timeout TIMEOUT
  [--http-method HTTP_METHOD]
  [--url-path URL_PATH]
  --max-retries MAX_RETRIES
```

(continues on next page)

(continued from previous page)

```
[--expected-codes EXPECTED_CODES]
[--admin-state-down]
--type
{PING,TCP,HTTP,HTTPS}
--pool POOL
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- tenant-id** TENANT\_ID  
The owner tenant ID.
- delay** <DELAY>  
The time in seconds between sending probes to members.
- name** <NAME>  
Name of the health monitor.
- timeout** <TIMEOUT>  
Maximum number of seconds for a monitor to wait for a connection to be established before it times out. The value must be less than the delay value.
- http-method** <HTTP\_METHOD>  
The HTTP method used for requests by the monitor of type HTTP.
- url-path** <URL\_PATH>  
The HTTP path used in the HTTP request used by the monitor to test a member health. This must be a string beginning with a / (forward slash).
- max-retries** <MAX\_RETRIES>  
Number of permissible connection failures before changing the member status to INACTIVE. [1..10].
- expected-codes** <EXPECTED\_CODES>  
The list of HTTP status codes expected in response from the member to declare it healthy. This attribute can contain one value, or a list of values separated by comma, or a range of values (e.g. 200-299). If this attribute is not specified, it defaults to 200.

**--admin-state-down**

Set admin state up to false.

**--type** <TYPE>

One of the predefined health monitor types.

**--pool** <POOL>

ID or name of the pool that this healthmonitor will monitor.

This command is provided by the python-neutronclient plugin.

**lbaas-healthmonitor-delete**

LBaaS v2 Delete a given healthmonitor.

```
neutron lbaas-healthmonitor-delete HEALTHMONITOR [HEALTHMONITOR ...]
```

**HEALTHMONITOR**

ID(s) or name(s) of healthmonitor to delete.

This command is provided by the python-neutronclient plugin.

**lbaas-healthmonitor-list**

LBaaS v2 List healthmonitors that belong to a given tenant.

```
neutron lbaas-healthmonitor-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
```

**-f** <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

**-c** COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>

when to include quotes, defaults to nonnumeric

**--noindent**

whether to disable indenting the JSON

- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- P** SIZE, **--page-size** SIZE  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key** FIELD  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.
- sort-dir** {asc,desc}  
Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## lbaas-healthmonitor-show

LBaaS v2 Show information of a given healthmonitor.

```
neutron lbaas-healthmonitor-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  HEALTHMONITOR
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- D**, **--show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.

**HEALTHMONITOR**

ID or name of healthmonitor to look up.

This command is provided by the python-neutronclient plugin.

**lbaas-healthmonitor-update**

LBaaS v2 Update a given healthmonitor.

```
neutron lbaas-healthmonitor-update
  [--delay DELAY]
  [--name NAME]
  [--timeout TIMEOUT]
  [--http-method HTTP_METHOD]
  [--url-path URL_PATH]
  [--max-retries MAX_RETRIES]
  [--expected-codes EXPECTED_CODES]
  [--admin-state-up {True,False}]
  HEALTHMONITOR
```

- delay** <DELAY>  
The time in seconds between sending probes to members.
- name** <NAME>  
Name of the health monitor.
- timeout** <TIMEOUT>  
Maximum number of seconds for a monitor to wait for a connection to be established before it times out. The value must be less than the delay value.

**--http-method** <HTTP\_METHOD>

The HTTP method used for requests by the monitor of type HTTP.

**--url-path** <URL\_PATH>

The HTTP path used in the HTTP request used by the monitor to test a member health. This must be a string beginning with a / (forward slash).

**--max-retries** <MAX\_RETRIES>

Number of permissible connection failures before changing the member status to INACTIVE. [1..10].

**--expected-codes** <EXPECTED\_CODES>

The list of HTTP status codes expected in response from the member to declare it healthy. This attribute can contain one value, or a list of values separated by comma, or a range of values (e.g. 200-299). If this attribute is not specified, it defaults to 200.

**--admin-state-up** {True,False}

Update the administrative state of the health monitor (True meaning Up).

#### HEALTHMONITOR

ID or name of healthmonitor to update.

This command is provided by the python-neutronclient plugin.

### lbaas-l7policy-create

LBaaS v2 Create L7 policy.

```
neutron lbaas-l7policy-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  [--name NAME]
  [--description DESCRIPTION]
  --action ACTION
  [--redirect-pool REDIRECT_POOL]
  [--redirect-url REDIRECT_URL]
  [--position POSITION]
  [--admin-state-down]
  --listener LISTENER
```

**-f** <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

**-c** COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

**--noindent**

whether to disable indenting the JSON

**--prefix** <PREFIX>

add a prefix to all variable names



- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- tenant-id** TENANT\_ID  
The owner tenant ID.
- name** <NAME>  
Name of the policy.
- description** <DESCRIPTION>  
Description of the policy.
- action** ACTION  
Action type of the policy.
- redirect-pool** <REDIRECT\_POOL>  
ID or name of the pool for REDIRECT\_TO\_POOL action type.
- redirect-url** <REDIRECT\_URL>  
URL for REDIRECT\_TO\_URL action type. This should be a valid URL string.
- position** <POSITION>  
L7 policy position in ordered policies list. This must be an integer starting from 1. Not specifying the position will place the policy at the tail of existing policies list.
- admin-state-down**  
Set admin state up to false.
- listener** LISTENER  
ID or name of the listener this policy belongs to.

This command is provided by the python-neutronclient plugin.

### **lbaas-l7policy-delete**

LBaaS v2 Delete a given L7 policy.

```
neutron lbaas-l7policy-delete L7POLICY [L7POLICY ...]
```

#### **L7POLICY**

ID(s) or name(s) of l7policy to delete.

This command is provided by the python-neutronclient plugin.

## lbaas-l7policy-list

LBaaS v2 List L7 policies that belong to a given listener.

```
neutron lbaas-l7policy-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

**--sort-ascending**  
sort the column(s) in ascending order

**--sort-descending**  
sort the column(s) in descending order

**-D**, **--show-details**  
Show detailed information.

**-F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.

- P** SIZE, **--page-size** SIZE  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key** FIELD  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.
- sort-dir** {asc,desc}  
Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## lbaas-l7policy-show

LBaaS v2 Show information of a given L7 policy.

```
neutron lbaas-l7policy-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  L7POLICY
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- D**, **--show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.

### L7POLICY

ID or name of l7policy to look up.

This command is provided by the python-neutronclient plugin.

### lbaas-l7policy-update

LBaaS v2 Update a given L7 policy.

```
neutron lbaas-l7policy-update
  [--name NAME]
  [--description DESCRIPTION]
  [--action ACTION]
  [--redirect-pool REDIRECT_POOL]
  [--redirect-url REDIRECT_URL]
  [--position POSITION]
  [--admin-state-up {True,False}]
  L7POLICY
```

**--name** <NAME>

Name of the policy.

**--description** <DESCRIPTION>

Description of the policy.

**--action** ACTION

Action type of the policy.

**--redirect-pool** <REDIRECT\_POOL>

ID or name of the pool for REDIRECT\_TO\_POOL action type.

**--redirect-url** <REDIRECT\_URL>

URL for REDIRECT\_TO\_URL action type. This should be a valid URL string.

**--position** <POSITION>

L7 policy position in ordered policies list. This must be an integer starting from 1. Not specifying the position will place the policy at the tail of existing policies list.

**--admin-state-up** {True,False}

Specify the administrative state of the policy (True meaning Up).

### L7POLICY

ID or name of l7policy to update.

This command is provided by the python-neutronclient plugin.

### lbaas-l7rule-create

LBaaS v2 Create L7 rule.

```
neutron lbaas-l7rule-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
```

(continues on next page)

(continued from previous page)

```

[--max-width <integer>]
[--fit-width]
[--print-empty]
[--tenant-id TENANT_ID]
--type
{HOST_NAME,PATH,FILE_TYPE,HEADER,COOKIE}
--compare-type
{REGEX,STARTS_WITH,ENDS_WITH,CONTAINS,EQUAL_TO}
[--invert-compare]
[--key KEY]
--value VALUE
[--admin-state-down]
L7POLICY

```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- tenant-id** TENANT\_ID  
The owner tenant ID.
- type** <TYPE>  
Rule type.
- compare-type** <COMPARE\_TYPE>  
Rule compare type.
- invert-compare**  
Invert the compare type.
- key** <KEY>  
Key to compare. Relevant for HEADER and COOKIE types only.
- value** <VALUE>  
Value to compare.
- admin-state-down**  
Set admin state up to false

#### L7POLICY

ID or name of L7 policy this rule belongs to.

This command is provided by the python-neutronclient plugin.

#### lbaas-l7rule-delete

LBaaS v2 Delete a given L7 rule.

```
neutron lbaas-l7rule-delete RULE [RULE ...] L7POLICY
```

#### RULE

ID(s) or name(s) of rule to delete.

#### L7POLICY

ID or name of L7 policy this rule belongs to.

This command is provided by the python-neutronclient plugin.

#### lbaas-l7rule-list

LBaaS v2 List L7 rules that belong to a given L7 policy.

```
neutron lbaas-l7rule-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
  L7POLICY
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- P** SIZE, **--page-size** SIZE  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key** FIELD  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.
- sort-dir** {asc, desc}  
Sorts the list in the specified direction. You can repeat this option.

**L7POLICY**

ID or name of L7 policy this rule belongs to.

This command is provided by the python-neutronclient plugin.

**lbaas-l7rule-show**

LBaaS v2 Show information of a given rule.

```
neutron lbaas-l7rule-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  RULE
  L7POLICY
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.

**RULE**

ID or name of rule to look up.

**L7POLICY**

ID or name of L7 policy this rule belongs to.

This command is provided by the python-neutronclient plugin.

## lbaas-l7rule-update

LBaaS v2 Update a given L7 rule.

```
neutron lbaas-l7rule-update
  [--type {HOST_NAME,PATH,FILE_TYPE,HEADER,COOKIE}]
  [--compare-type {REGEX,STARTS_WITH,ENDS_WITH,CONTAINS,EQUAL_TO}]
  [--invert-compare]
  [--key KEY]
  [--value VALUE]
  [--admin-state-up {True,False}]
  RULE
  L7POLICY
```

- type** <TYPE>  
Rule type.
- compare-type** <COMPARE\_TYPE>  
Rule compare type.
- invert-compare**  
Invert the compare type.



**--key** <KEY>  
Key to compare. Relevant for HEADER and COOKIE types only.

**--value** <VALUE>  
Value to compare.

**--admin-state-up** {True,False}  
Specify the administrative state of the rule (True meaning Up).

**RULE**  
ID or name of rule to update.

**L7POLICY**  
ID or name of L7 policy this rule belongs to.

This command is provided by the python-neutronclient plugin.

## lbaas-listener-create

LBaaS v2 Create a listener.

```
neutron lbaas-listener-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  [--description DESCRIPTION]
  [--connection-limit CONNECTION_LIMIT]
  [--default-pool DEFAULT_POOL]
  [--admin-state-down]
  [--name NAME]
  [--default-tls-container-ref DEFAULT_TLS_CONTAINER_REF]
  [--sni-container-refs SNI_CONTAINER_REFS [SNI_CONTAINER_REFS ...]]
  [--loadbalancer LOADBALANCER]
  --protocol
  {TCP,HTTP,HTTPS,TERMINATED_HTTPS}
  --protocol-port PORT
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**--tenant-id** TENANT\_ID

The owner tenant ID.

**--description** <DESCRIPTION>

Description of the listener.

**--connection-limit** <CONNECTION\_LIMIT>

The maximum number of connections per second allowed for the listener. Positive integer or -1 for unlimited (default).

**--default-pool** <DEFAULT\_POOL>

Default pool for the listener.

**--admin-state-down**

Set admin state up to false.

**--name** <NAME>

The name of the listener. At least one of default-pool or loadbalancer must be specified.

**--default-tls-container-ref** <DEFAULT\_TLS\_CONTAINER\_REF>

Default TLS container reference to retrieve TLS information.

**--sni-container-refs** <SNI\_CONTAINER\_REFS>

List of TLS container references for SNI.

**--loadbalancer** LOADBALANCER

ID or name of the load balancer.

**--protocol** <PROTOCOL>

Protocol for the listener.

**--protocol-port** PORT

Protocol port for the listener.

This command is provided by the python-neutronclient plugin.

## lbaas-listener-delete

LBaaS v2 Delete a given listener.

```
neutron lbaas-listener-delete LISTENER [LISTENER ...]
```

**LISTENER**

ID(s) or name(s) of listener to delete.

This command is provided by the python-neutronclient plugin.

## lbaas-listener-list

LBaaS v2 List listeners that belong to a given tenant.

```

neutron lbaas-listener-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]

```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

**--sort-ascending**  
sort the column(s) in ascending order

**--sort-descending**  
sort the column(s) in descending order

**-D**, **--show-details**  
Show detailed information.

**-F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.

**-P** SIZE, **--page-size** SIZE

Specify retrieve unit of each request, then split one request to several requests.

**--sort-key** FIELD

Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.

**--sort-dir** {asc,desc}

Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## lbaas-listener-show

LBaaS v2 Show information of a given listener.

```
neutron lbaas-listener-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  LISTENER
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**-D**, **--show-details**  
Show detailed information.

**-F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.

**LISTENER**

ID or name of listener to look up.

This command is provided by the python-neutronclient plugin.

**lbaas-listener-update**

LBaaS v2 Update a given listener.

```
neutron lbaas-listener-update
  [--description DESCRIPTION]
  [--connection-limit CONNECTION_LIMIT]
  [--default-pool DEFAULT_POOL]
  [--name NAME]
  [--admin-state-up {True,False}]
  LISTENER
```

**--description** <DESCRIPTION>

Description of the listener.

**--connection-limit** <CONNECTION\_LIMIT>

The maximum number of connections per second allowed for the listener. Positive integer or -1 for unlimited (default).

**--default-pool** <DEFAULT\_POOL>

Default pool for the listener.

**--name** <NAME>

Name of the listener.

**--admin-state-up** {True,False}

Specify the administrative state of the listener. (True meaning Up)

**LISTENER**

ID or name of listener to update.

This command is provided by the python-neutronclient plugin.

**lbaas-loadbalancer-create**

LBaaS v2 Create a loadbalancer.

```
neutron lbaas-loadbalancer-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  [--description DESCRIPTION]
  [--name NAME]
  [--admin-state-down]
  [--provider PROVIDER]
```

(continues on next page)

(continued from previous page)

```
[--flavor FLAVOR]
[--vip-address VIP_ADDRESS]
VIP_SUBNET
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- tenant-id** TENANT\_ID  
The owner tenant ID.
- description** <DESCRIPTION>  
Description of the load balancer.
- name** NAME  
Name of the load balancer.
- admin-state-down**  
Set admin state up to false.
- provider** <PROVIDER>  
Provider name of the load balancer service.
- flavor** <FLAVOR>  
ID or name of the flavor.
- vip-address** <VIP\_ADDRESS>  
VIP address for the load balancer.
- VIP\_SUBNET**  
Load balancer VIP subnet.

This command is provided by the python-neutronclient plugin.

## lbaas-loadbalancer-delete

LBaaS v2 Delete a given loadbalancer.

```
neutron lbaas-loadbalancer-delete LOADBALANCER [LOADBALANCER ...]
```

### LOADBALANCER

ID(s) or name(s) of loadbalancer to delete.

This command is provided by the python-neutronclient plugin.

## lbaas-loadbalancer-list

LBaaS v2 List loadbalancers that belong to a given tenant.

```
neutron lbaas-loadbalancer-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F FIELD, --field FIELD**  
Specify the field(s) to be returned by server. You can repeat this option.
- P SIZE, --page-size SIZE**  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key FIELD**  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of `sort_dir` and `sort_key` values. Extra `sort_dir` options are ignored. Missing `sort_dir` options use the default `asc` value.
- sort-dir {asc, desc}**  
Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## lbaas-loadbalancer-list-on-agent

List the loadbalancers on a loadbalancer v2 agent.

```
neutron lbaas-loadbalancer-list-on-agent
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  LBAAS_AGENT
```

- f <FORMATTER>, --format <FORMATTER>**  
the output format, defaults to `table`
- c COLUMN, --column COLUMN**  
specify the column(s) to include, can be repeated to show multiple columns
- quote <QUOTE\_MODE>**  
when to include quotes, defaults to `nonnumeric`
- noindent**  
whether to disable indenting the JSON
- max-width <integer>**  
Maximum display width, `<1` to disable. You can also use the `CLIFF_MAX_TERM_WIDTH` environment variable, but the parameter takes precedence.



- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- LBAAS\_AGENT**  
ID of the loadbalancer agent to query.

This command is provided by the python-neutronclient plugin.

### lbaas-loadbalancer-show

LBaaS v2 Show information of a given loadbalancer.

```
neutron lbaas-loadbalancer-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  LOADBALANCER
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names

- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- D, --show-details**  
Show detailed information.
- F FIELD, --field FIELD**  
Specify the field(s) to be returned by server. You can repeat this option.

#### **LOADBALANCER**

ID or name of loadbalancer to look up.

This command is provided by the python-neutronclient plugin.

### **lbaas-loadbalancer-stats**

Retrieve stats for a given loadbalancer.

```
neutron lbaas-loadbalancer-stats
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  LOADBALANCER
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**-D, --show-details**

Show detailed information.

**-F FIELD, --field FIELD**

Specify the field(s) to be returned by server. You can repeat this option.

**LOADBALANCER**

ID or name of loadbalancer to look up.

This command is provided by the python-neutronclient plugin.

**lbaas-loadbalancer-status**

Retrieve status for a given loadbalancer.

The only output is a formatted JSON tree, and the table format does not support this type of data.

```
neutron lbaas-loadbalancer-status LOADBALANCER
```

**LOADBALANCER**

ID or name of loadbalancer to show.

This command is provided by the python-neutronclient plugin.

**lbaas-loadbalancer-update**

LBaaS v2 Update a given loadbalancer.

```
neutron lbaas-loadbalancer-update
  [--admin-state-up {True,False}]
  [--description DESCRIPTION]
  [--name NAME]
  LOADBALANCER
```

**--admin-state-up {True,False}**

Update the administrative state of the load balancer (True meaning Up).

**--description <DESCRIPTION>**

Description of the load balancer.

**--name NAME**

Name of the load balancer.

**LOADBALANCER**

ID or name of loadbalancer to update.

This command is provided by the python-neutronclient plugin.

## lbaas-member-create

LBaaS v2 Create a member.

```
neutron lbaas-member-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  [--name NAME]
  [--weight WEIGHT]
  [--admin-state-down]
  --subnet SUBNET
  --address ADDRESS
  --protocol-port PROTOCOL_PORT
  POOL
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--tenant-id** TENANT\_ID  
The owner tenant ID.

**--name** <NAME>  
Name of the member.

**--weight** <WEIGHT>  
Weight of the member in the pool (default:1, [0..256]).

**--admin-state-down**  
Set admin state up to false.

**--subnet** <SUBNET>  
Subnet ID or name for the member.

**--address** <ADDRESS>

IP address of the pool member in the pool.

**--protocol-port** <PROTOCOL\_PORT>

Port on which the pool member listens for requests or connections.

**POOL**

ID or name of the pool that this member belongs to.

This command is provided by the python-neutronclient plugin.

### lbaas-member-delete

LBaaS v2 Delete a given member.

```
neutron lbaas-member-delete MEMBER [MEMBER ...] POOL
```

**MEMBER**

ID(s) or name(s) of member to delete.

**POOL**

ID or name of the pool that this member belongs to.

This command is provided by the python-neutronclient plugin.

### lbaas-member-list

LBaaS v2 List members that belong to a given pool.

```
neutron lbaas-member-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
  POOL
```

**-f** <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

**-c** COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>

when to include quotes, defaults to nonnumeric

- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- P** SIZE, **--page-size** SIZE  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key** FIELD  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.
- sort-dir** {asc,desc}  
Sorts the list in the specified direction. You can repeat this option.

## POOL

ID or name of the pool that this member belongs to.

This command is provided by the python-neutronclient plugin.

## lbaas-member-show

LBaaS v2 Show information of a given member.

```
neutron lbaas-member-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
```

(continues on next page)

(continued from previous page)

```

[--print-empty]
[-D]
[-F FIELD]
MEMBER
POOL

```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- D**, **--show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- MEMBER**  
ID or name of member to look up.
- POOL**  
ID or name of the pool that this member belongs to.

This command is provided by the python-neutronclient plugin.

## lbaas-member-update

LBaaS v2 Update a given member.

```

neutron lbaas-member-update
[--admin-state-up {True,False}]
[--name NAME]
[--weight WEIGHT]
MEMBER
POOL

```

- admin-state-up** {True,False}  
Update the administrative state of the member (True meaning Up).

**--name** <NAME>

Name of the member.

**--weight** <WEIGHT>

Weight of the member in the pool (default:1, [0..256]).

**MEMBER**

ID or name of member to update.

**POOL**

ID or name of the pool that this member belongs to.

This command is provided by the python-neutronclient plugin.

## lbaas-pool-create

LBaaS v2 Create a pool.

```
neutron lbaas-pool-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  [--description DESCRIPTION]
  [--name NAME]
  --lb-algorithm
  {ROUND_ROBIN,LEAST_CONNECTIONS,SOURCE_IP}
  [--admin-state-down]
  [--listener LISTENER]
  [--loadbalancer LOADBALANCER]
  --protocol
  {HTTP,HTTPS,TCP}
  [--session-persistence type=TYPE[,cookie_name=COOKIE_NAME]]
```

**-f** <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

**-c** COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

**--noindent**

whether to disable indenting the JSON

**--prefix** <PREFIX>

add a prefix to all variable names

**--max-width** <integer>

Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable



- print-empty**  
Print empty table if there is no data to show.
- tenant-id** TENANT\_ID  
The owner tenant ID.
- description** <DESCRIPTION>  
Description of the pool.
- name** <NAME>  
The name of the pool.
- lb-algorithm** <LB\_ALGORITHM>  
The algorithm used to distribute load between the members of the pool.
- admin-state-down**  
Set admin state up to false.
- listener** <LISTENER>  
Listener whose default-pool should be set to this pool. At least one of listener or loadbalancer must be specified.
- loadbalancer** <LOADBALANCER>  
Loadbalancer with which this pool should be associated. At least one of listener or loadbalancer must be specified.
- protocol** <PROTOCOL>  
Protocol for balancing.
- session-persistence** type=TYPE [, cookie\_name=COOKIE\_NAME]  
The type of session persistence to use and associated cookie name.

This command is provided by the python-neutronclient plugin.

### lbaas-pool-delete

LBaaS v2 Delete a given pool.

```
neutron lbaas-pool-delete POOL [POOL ...]
```

#### POOL

ID(s) or name(s) of pool to delete.

This command is provided by the python-neutronclient plugin.

### lbaas-pool-list

LBaaS v2 List pools that belong to a given tenant.

```
neutron lbaas-pool-list
  [-f {csv, json, table, value, yaml}]
  [-c COLUMN]
  [--quote {all, minimal, none, nonnumeric}]
  [--noindent]
  [--max-width <integer>]
```

(continues on next page)

(continued from previous page)

```

[--fit-width]
[--print-empty]
[--sort-column SORT_COLUMN]
[--sort-ascending | --sort-descending]
[-D]
[-F FIELD]
[-P SIZE]
[--sort-key FIELD]
[--sort-dir {asc,desc}]

```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric
- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D**, **--show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- P** SIZE, **--page-size** SIZE  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key** FIELD  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.

**--sort-dir** {asc,desc}

Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## lbaas-pool-show

LBaaS v2 Show information of a given pool.

```
neutron lbaas-pool-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  POOL
```

**-f** <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

**-c** COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

**--noindent**

whether to disable indenting the JSON

**--prefix** <PREFIX>

add a prefix to all variable names

**--max-width** <integer>

Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**-D**, **--show-details**

Show detailed information.

**-F** FIELD, **--field** FIELD

Specify the field(s) to be returned by server. You can repeat this option.

**POOL**

ID or name of pool to look up.

This command is provided by the python-neutronclient plugin.

## lbaas-pool-update

LBaaS v2 Update a given pool.

```
neutron lbaas-pool-update
  [--admin-state-up {True,False}]
  [--session-persistence type=TYPE[,cookie_name=COOKIE_NAME]
  |
  --no-session-persistence
  ]
  [--description DESCRIPTION]
  [--name NAME]
  [--lb-algorithm {ROUND_ROBIN,LEAST_CONNECTIONS,SOURCE_IP}]
  POOL
```

**--admin-state-up** {True,False}

Update the administrative state of the pool (True meaning Up).

**--session-persistence** type=TYPE[,cookie\_name=COOKIE\_NAME]

The type of session persistence to use and associated cookie name.

**--no-session-persistence**

Clear session persistence for the pool.

**--description** <DESCRIPTION>

Description of the pool.

**--name** <NAME>

The name of the pool.

**--lb-algorithm** <LB\_ALGORITHM>

The algorithm used to distribute load between the members of the pool.

**POOL**

ID or name of pool to update.

This command is provided by the python-neutronclient plugin.

## meter-label-create

Create a metering label for a given tenant.

```
neutron meter-label-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  [--description DESCRIPTION]
  [--shared]
  NAME
```

**-f** <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- tenant-id** TENANT\_ID  
The owner tenant ID.
- description** <DESCRIPTION>  
Description of the metering label to be created.
- shared**  
Set the label as shared.

**NAME**

Name of the metering label to be created.

This command is provided by the python-neutronclient plugin.

**meter-label-delete**

Delete a given metering label.

```
neutron meter-label-delete METERING_LABEL [METERING_LABEL ...]
```

**METERING\_LABEL**

ID(s) or name(s) of metering\_label to delete.

This command is provided by the python-neutronclient plugin.

**meter-label-list**

List metering labels that belong to a given tenant.

```
neutron meter-label-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
```

(continues on next page)

(continued from previous page)

```
[--fit-width]
[--print-empty]
[--sort-column SORT_COLUMN]
[--sort-ascending | --sort-descending]
[-D]
[-F FIELD]
[-P SIZE]
[--sort-key FIELD]
[--sort-dir {asc,desc}]
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric
- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D**, **--show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- P** SIZE, **--page-size** SIZE  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key** FIELD  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.

**--sort-dir** {asc, desc}

Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## meter-label-rule-create

Create a metering label rule for a given label.

```
neutron meter-label-rule-create
  [-f {json, shell, table, value, yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  [--direction {ingress, egress}]
  [--excluded]
  LABEL
  REMOTE_IP_PREFIX
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--tenant-id** TENANT\_ID  
The owner tenant ID.

**--direction** <DIRECTION>  
Direction of traffic, default: ingress.

**--excluded**  
Exclude this CIDR from the label, default: not excluded.

**LABEL**  
ID or name of the label.

## REMOTE\_IP\_PREFIX

CIDR to match on.

This command is provided by the python-neutronclient plugin.

## meter-label-rule-delete

Delete a given metering label.

```
neutron meter-label-rule-delete
    METERING_LABEL_RULE
    [METERING_LABEL_RULE ...]
```

## METERING\_LABEL\_RULE

ID(s) of metering\_label\_rule to delete.

This command is provided by the python-neutronclient plugin.

## meter-label-rule-list

List metering labels that belong to a given label.

```
neutron meter-label-rule-list
    [-f {csv,json,table,value,yaml}]
    [-c COLUMN]
    [--quote {all,minimal,none,nonnumeric}]
    [--noindent]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
    [-D]
    [-F FIELD]
    [-P SIZE]
    [--sort-key FIELD]
    [--sort-dir {asc,desc}]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.



- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- P** SIZE, **--page-size** SIZE  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key** FIELD  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.
- sort-dir** {asc,desc}  
Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## meter-label-rule-show

Show information of a given metering label rule.

```
neutron meter-label-rule-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  METERING_LABEL_RULE
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

- noindent**  
whether to disable indenting the JSON
  - prefix** <PREFIX>  
add a prefix to all variable names
  - max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
  - fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
  - print-empty**  
Print empty table if there is no data to show.
  - D, --show-details**  
Show detailed information.
  - F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- METERING\_LABEL\_RULE**  
ID of metering\_label\_rule to look up.

This command is provided by the python-neutronclient plugin.

## meter-label-show

Show information of a given metering label.

```
neutron meter-label-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  METERING_LABEL
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names

- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- D, --show-details**  
Show detailed information.
- F FIELD, --field FIELD**  
Specify the field(s) to be returned by server. You can repeat this option.

**METERING\_LABEL**

ID or name of metering\_label to look up.

This command is provided by the python-neutronclient plugin.

**net-create**

Create a network for a given tenant.

```
neutron net-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  [--admin-state-down]
  [--shared]
  [--provider:network_type <network_type>]
  [--provider:physical_network <physical_network_name>]
  [--provider:segmentation_id <segmentation_id>]
  [--vlan-transparent {True,False}]
  [--description DESCRIPTION]
  [--qos-policy QOS_POLICY]
  [--availability-zone-hint AVAILABILITY_ZONE]
  [--dns-domain DNS_DOMAIN]
  NAME
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON

- prefix** <PREFIX>  
add a prefix to all variable names
  - max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
  - fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
  - print-empty**  
Print empty table if there is no data to show.
  - tenant-id** TENANT\_ID  
The owner tenant ID.
  - admin-state-down**  
Set admin state up to false.
  - shared**  
Set the network as shared.
  - provider:network\_type** <network\_type>  
The physical mechanism by which the virtual network is implemented.
  - provider:physical\_network** <physical\_network\_name>  
Name of the physical network over which the virtual network is implemented.
  - provider:segmentation\_id** <segmentation\_id>  
VLAN ID for VLAN networks or tunnel-id for GRE/VXLAN networks.
  - vlan-transparent** {True,False}  
Create a VLAN transparent network.
  - description** <DESCRIPTION>  
Description of network.
  - qos-policy** <QOS\_POLICY>  
ID or name of the QoS policy that should be attached to the resource.
  - availability-zone-hint** AVAILABILITY\_ZONE  
Availability Zone for the network (requires availability zone extension, this option can be repeated).
  - dns-domain** <DNS\_DOMAIN>  
Assign DNS domain to the network (requires DNS integration extension)
- NAME**  
Name of the network to be created.

This command is provided by the python-neutronclient plugin.

## net-delete

Delete a given network.

```
neutron net-delete NETWORK [NETWORK ...]
```

### NETWORK

ID(s) or name(s) of network to delete.

This command is provided by the python-neutronclient plugin.

## net-external-list

List external networks that belong to a given tenant.

```
neutron net-external-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
  [--tenant-id TENANT_ID]
  [--name NAME]
  [--admin-state-up {True,False}]
  [--status STATUS]
  [--shared {True,False}]
  [--router:external {True,False}]
  [--tags TAG]
  [--tags-any TAG]
  [--not-tags TAG]
  [--not-tags-any TAG]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- P** SIZE, **--page-size** SIZE  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key** FIELD  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.
- sort-dir** {asc,desc}  
Sorts the list in the specified direction. You can repeat this option.
- tenant-id** <TENANT\_ID>  
Filter networks belonging to the given tenant.
- name** <NAME>  
Filter networks according to their name.
- admin-state-up** {True,False}  
Filter and list the networks whose administrative state is active
- status** <STATUS>  
Filter networks according to their operation status.(For example: ACTIVE, ERROR etc)
- shared** {True,False}  
Filter and list the networks which are shared.
- router:external** {True,False}  
Filter and list the networks which are external.
- tags** TAG  
Filter and list networks which has all given tags. Multiple tags can be set like tags <tag[,tag]>
- tags-any** TAG  
Filter and list networks which has any given tags. Multiple tags can be set like tags-any <tag[,tag]>

**--not-tags** TAG  
Filter and list networks which does not have all given tags. Multiple tags can be set like not-tags <tag[,tag]>

**--not-tags-any** TAG  
Filter and list networks which does not have any given tags. Multiple tags can be set like not-tags-any <tag[,tag]>

This command is provided by the python-neutronclient plugin.

## net-ip-availability-list

List IP usage of networks

```
neutron net-ip-availability-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
  [--ip-version {4,6}]
  [--network-id NETWORK_ID]
  [--network-name NETWORK_NAME]
  [--tenant-id TENANT_ID]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- sort-key** FIELD  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.
- sort-dir** {asc,desc}  
Sorts the list in the specified direction. You can repeat this option.
- ip-version** <IP\_VERSION>  
Returns IP availability for the network subnets with a given IP version. Default: 4
- network-id** <NETWORK\_ID>  
Returns IP availability for the network matching a given network ID.
- network-name** <NETWORK\_NAME>  
Returns IP availability for the network matching a given name.
- tenant-id** <TENANT\_ID>  
Returns IP availability for the networks with a given tenant ID.

This command is provided by the python-neutronclient plugin.

## net-ip-availability-show

Show IP usage of specific network

```
neutron net-ip-availability-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  NETWORK
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns



- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.

**NETWORK**

ID or name of network to look up.

This command is provided by the python-neutronclient plugin.

**net-list**

List networks that belong to a given tenant.

```
neutron net-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
  [--tenant-id TENANT_ID]
  [--name NAME]
  [--admin-state-up {True,False}]
  [--status STATUS]
  [--shared {True,False}]
  [--router:external {True,False}]
  [--tags TAG]
  [--tags-any TAG]
  [--not-tags TAG]
  [--not-tags-any TAG]
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

- quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric
- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- P** SIZE, **--page-size** SIZE  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key** FIELD  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.
- sort-dir** {asc,desc}  
Sorts the list in the specified direction. You can repeat this option.
- tenant-id** <TENANT\_ID>  
Filter networks belonging to the given tenant.
- name** <NAME>  
Filter networks according to their name.
- admin-state-up** {True,False}  
Filter and list the networks whose administrative state is active
- status** <STATUS>  
Filter networks according to their operation status.(For example: ACTIVE, ERROR etc)
- shared** {True,False}  
Filter and list the networks which are shared.

- router:external** {True,False}  
Filter and list the networks which are external.
- tags** TAG  
Filter and list networks which has all given tags. Multiple tags can be set like tags <tag[,tag]>
- tags-any** TAG  
Filter and list networks which has any given tags. Multiple tags can be set like tags-any <tag[,tag]>
- not-tags** TAG  
Filter and list networks which does not have all given tags. Multiple tags can be set like not-tags <tag[,tag]>
- not-tags-any** TAG  
Filter and list networks which does not have any given tags. Multiple tags can be set like not-tags-any <tag[,tag]>

This command is provided by the python-neutronclient plugin.

### net-list-on-dhcp-agent

List the networks on a DHCP agent.

```
neutron net-list-on-dhcp-agent
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
  [--tenant-id TENANT_ID]
  [--name NAME]
  [--admin-state-up {True,False}]
  [--status STATUS]
  [--shared {True,False}]
  [--router:external {True,False}]
  [--tags TAG]
  [--tags-any TAG]
  [--not-tags TAG]
  [--not-tags-any TAG]
  DHCP_AGENT
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

- quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric
- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- P** SIZE, **--page-size** SIZE  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key** FIELD  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.
- sort-dir** {asc,desc}  
Sorts the list in the specified direction. You can repeat this option.
- tenant-id** <TENANT\_ID>  
Filter networks belonging to the given tenant.
- name** <NAME>  
Filter networks according to their name.
- admin-state-up** {True,False}  
Filter and list the networks whose administrative state is active
- status** <STATUS>  
Filter networks according to their operation status.(For example: ACTIVE, ERROR etc)
- shared** {True,False}  
Filter and list the networks which are shared.

- router:external** {True,False}  
Filter and list the networks which are external.
- tags** TAG  
Filter and list networks which has all given tags. Multiple tags can be set like tags <tag[,tag]>
- tags-any** TAG  
Filter and list networks which has any given tags. Multiple tags can be set like tags-any <tag[,tag]>
- not-tags** TAG  
Filter and list networks which does not have all given tags. Multiple tags can be set like not-tags <tag[,tag]>
- not-tags-any** TAG  
Filter and list networks which does not have any given tags. Multiple tags can be set like not-tags-any <tag[,tag]>
- DHCP\_AGENT**  
ID of the DHCP agent.

This command is provided by the python-neutronclient plugin.

## net-show

Show information of a given network.

```
neutron net-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  NETWORK
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**-D, --show-details**

Show detailed information.

**-F FIELD, --field FIELD**

Specify the field(s) to be returned by server. You can repeat this option.

**NETWORK**

ID or name of network to look up.

This command is provided by the python-neutronclient plugin.

## net-update

Update networks information.

```
neutron net-update
  [--name NAME]
  [--description DESCRIPTION]
  [--qos-policy QOS_POLICY | --no-qos-policy]
  [--dns-domain DNS_DOMAIN | --no-dns-domain]
  NETWORK
```

**--name <NAME>**

Name of the network.

**--description <DESCRIPTION>**

Description of this network.

**--qos-policy <QOS\_POLICY>**

ID or name of the QoS policy that should be attached to the resource.

**--no-qos-policy**

Detach QoS policy from the resource.

**--dns-domain <DNS\_DOMAIN>**

Assign DNS domain to the network (requires DNS integration extension.)

**--no-dns-domain**

Unassign DNS domain from the network (requires DNS integration extension.)

**NETWORK**

ID or name of network to update.

This command is provided by the python-neutronclient plugin.

## port-create

Create a port for a given tenant.

```

neutron port-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  [--name NAME]
  [--description DESCRIPTION]
  [--fixed-ip subnet_id=SUBNET,ip_address=IP_ADDR]
  [--device-id DEVICE_ID]
  [--device-owner DEVICE_OWNER]
  [--admin-state-down]
  [--mac-address MAC_ADDRESS]
  [--vnic-type <direct | direct-physical | macvtap | normal | baremetal>]
  [--binding-profile BINDING_PROFILE]
  [--security-group SECURITY_GROUP | --no-security-groups]
  [--extra-dhcp-opt EXTRA_DHCP_OPTS]
  [--qos-policy QOS_POLICY]
  [--allowed-address-pair ip_address=IP_ADDR|CIDR[,mac_address=MAC_ADDR]
  |
  --no-allowed-address-pairs
  ]
  [--dns-name DNS_NAME]
  NETWORK

```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--tenant-id** TENANT\_ID  
The owner tenant ID.

- name** <NAME>  
Name of this port.
- description** <DESCRIPTION>  
Description of this port.
- fixed-ip** subnet\_id=SUBNET,ip\_address=IP\_ADDR  
Desired IP and/or subnet for this port: subnet\_id=<name\_or\_id>,ip\_address=<ip>. You can repeat this option.
- device-id** <DEVICE\_ID>  
Device ID of this port.
- device-owner** <DEVICE\_OWNER>  
Device owner of this port.
- admin-state-down**  
Set admin state up to false.
- mac-address** <MAC\_ADDRESS>  
MAC address of this port.
- vnic-type** <direct | direct-physical | macvtap | normal | baremetal>  
VNIC type for this port.
- binding-profile** <BINDING\_PROFILE>  
Custom data to be passed as binding:profile.
- security-group** SECURITY\_GROUP  
Security group associated with the port. You can repeat this option.
- no-security-groups**  
Associate no security groups with the port.
- extra-dhcp-opt** <EXTRA\_DHCP\_OPTS>  
Extra dhcp options to be assigned to this port: opt\_name=<dhcp\_option\_name>,opt\_value=<value>,ip\_version={4  
You can repeat this option.
- qos-policy** <QOS\_POLICY>  
ID or name of the QoS policy that should be attached to the resource.
- allowed-address-pair** ip\_address=IP\_ADDR|CIDR[,mac\_address=MAC\_ADDR]  
Allowed address pair associated with the port. ip\_address parameter is required. IP address or CIDR can be specified for ip\_address. mac\_address parameter is optional. You can repeat this option.
- no-allowed-address-pairs**  
Associate no allowed address pairs with the port.
- dns-name** <DNS\_NAME>  
Assign DNS name to the port (requires DNS integration extension)

**NETWORK**

ID or name of the network this port belongs to.

This command is provided by the python-neutronclient plugin.



## port-delete

Delete a given port.

```
neutron port-delete PORT [PORT ...]
```

### PORT

ID(s) or name(s) of port to delete.

This command is provided by the python-neutronclient plugin.

## port-list

List ports that belong to a given tenant.

```
neutron port-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F FIELD, --field FIELD**  
Specify the field(s) to be returned by server. You can repeat this option.
- P SIZE, --page-size SIZE**  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key FIELD**  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of `sort_dir` and `sort_key` values. Extra `sort_dir` options are ignored. Missing `sort_dir` options use the default `asc` value.
- sort-dir {asc, desc}**  
Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## port-show

Show information of a given port.

```
neutron port-show
  [-f {json, shell, table, value, yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  PORT
```

- f <FORMATTER>, --format <FORMATTER>**  
the output format, defaults to table
- c COLUMN, --column COLUMN**  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix <PREFIX>**  
add a prefix to all variable names
- max-width <integer>**  
Maximum display width, <1 to disable. You can also use the `CLIFF_MAX_TERM_WIDTH` environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**-D, --show-details**

Show detailed information.

**-F FIELD, --field FIELD**

Specify the field(s) to be returned by server. You can repeat this option.

**PORT**

ID or name of port to look up.

This command is provided by the python-neutronclient plugin.

**port-update**

Update ports information.

```
neutron port-update
  [--name NAME]
  [--description DESCRIPTION]
  [--fixed-ip subnet_id=SUBNET,ip_address=IP_ADDR]
  [--device-id DEVICE_ID]
  [--device-owner DEVICE_OWNER]
  [--admin-state-up {True,False}]
  [--security-group SECURITY_GROUP | --no-security-groups]
  [--extra-dhcp-opt EXTRA_DHCP_OPTS]
  [--qos-policy QOS_POLICY | --no-qos-policy]
  [--allowed-address-pair ip_address=IP_ADDR|CIDR[,mac_address=MAC_ADDR]
  |
  --no-allowed-address-pairs
  ]
  [--dns-name DNS_NAME | --no-dns-name]
  PORT
```

**--name <NAME>**

Name of this port.

**--description <DESCRIPTION>**

Description of this port.

**--fixed-ip subnet\_id=SUBNET,ip\_address=IP\_ADDR**

Desired IP and/or subnet for this port: subnet\_id=<name\_or\_id>,ip\_address=<ip>. You can repeat this option.

**--device-id <DEVICE\_ID>**

Device ID of this port.

**--device-owner <DEVICE\_OWNER>**

Device owner of this port.

**--admin-state-up <ADMIN\_STATE\_UP>**

Set admin state up for the port.

- security-group** SECURITY\_GROUP  
Security group associated with the port. You can repeat this option.
- no-security-groups**  
Associate no security groups with the port.
- extra-dhcp-opt** <EXTRA\_DHCP\_OPTS>  
Extra dhcp options to be assigned to this port: opt\_name=<dhcp\_option\_name>,opt\_value=<value>,ip\_version={4  
You can repeat this option.
- qos-policy** <QOS\_POLICY>  
ID or name of the QoS policy that shouldbe attached to the resource.
- no-qos-policy**  
Detach QoS policy from the resource.
- allowed-address-pair** ip\_address=IP\_ADDR|CIDR[,mac\_address=MAC\_ADDR]  
Allowed address pair associated with the port. ip\_address parameter is required. IP address or  
CIDR can be specified for ip\_address. mac\_address parameter is optional. You can repeat this  
option.
- no-allowed-address-pairs**  
Associate no allowed address pairs with the port.
- dns-name** <DNS\_NAME>  
Assign DNS name to the port (requires DNS integration extension.)
- no-dns-name**  
Unassign DNS name from the port (requires DNS integration extension.)

**PORT**

ID or name of port to update.

This command is provided by the python-neutronclient plugin.

## purge

Delete all resources that belong to a given tenant.

```
neutron purge TENANT
```

**TENANT**

ID of Tenant owning the resources to be deleted.

This command is provided by the python-neutronclient plugin.

## qos-available-rule-types

List available qos rule types.

```
neutron qos-available-rule-types  
  [-f {csv,json,table,value,yaml}]  
  [-c COLUMN]  
  [--quote {all,minimal,none,nonnumeric}]  
  [--noindent]
```

(continues on next page)

(continued from previous page)

```

[--max-width <integer>]
[--fit-width]
[--print-empty]
[--sort-column SORT_COLUMN]
[--sort-ascending | --sort-descending]
[-D]
[-F FIELD]
[-P SIZE]
[--sort-key FIELD]
[--sort-dir {asc,desc}]

```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric
- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D**, **--show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- P** SIZE, **--page-size** SIZE  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key** FIELD  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.

**--sort-dir** {asc,desc}

Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## qos-bandwidth-limit-rule-create

Create a qos bandwidth limit rule.

```
neutron qos-bandwidth-limit-rule-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  [--max-kbps MAX_KBPS]
  [--max-burst-kbps MAX_BURST_KBPS]
  QOS_POLICY
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--tenant-id** TENANT\_ID  
The owner tenant ID.

**--max-kbps** <MAX\_KBPS>  
Maximum bandwidth in kbps.

**--max-burst-kbps** <MAX\_BURST\_KBPS>  
Maximum burst bandwidth in kbps.

**QOS\_POLICY**  
ID or name of the QoS policy.

This command is provided by the python-neutronclient plugin.

## qos-bandwidth-limit-rule-delete

Delete a given qos bandwidth limit rule.

```
neutron qos-bandwidth-limit-rule-delete
    BANDWIDTH_LIMIT_RULE
    [BANDWIDTH_LIMIT_RULE ...]
    QOS_POLICY
```

### **BANDWIDTH\_LIMIT\_RULE**

ID(s) of bandwidth\_limit\_rule to delete.

### **QOS\_POLICY**

ID or name of the QoS policy.

This command is provided by the python-neutronclient plugin.

## qos-bandwidth-limit-rule-list

List all qos bandwidth limit rules belonging to the specified policy.

```
neutron qos-bandwidth-limit-rule-list
    [-f {csv,json,table,value,yaml}]
    [-c COLUMN]
    [--quote {all,minimal,none,nonnumeric}]
    [--noindent]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
    [-D]
    [-F FIELD]
    [-P SIZE]
    [--sort-key FIELD]
    [--sort-dir {asc,desc}]
    QOS_POLICY
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment

variable `CLIFF_FIT_WIDTH=1` to always enable

**--print-empty**

Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN

specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

**--sort-ascending**

sort the column(s) in ascending order

**--sort-descending**

sort the column(s) in descending order

**-D, --show-details**

Show detailed information.

**-F** FIELD, **--field** FIELD

Specify the field(s) to be returned by server. You can repeat this option.

**-P** SIZE, **--page-size** SIZE

Specify retrieve unit of each request, then split one request to several requests.

**--sort-key** FIELD

Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of `sort_dir` and `sort_key` values. Extra `sort_dir` options are ignored. Missing `sort_dir` options use the default `asc` value.

**--sort-dir** {asc,desc}

Sorts the list in the specified direction. You can repeat this option.

**QOS\_POLICY**

ID or name of the QoS policy.

This command is provided by the `python-neutronclient` plugin.

## qos-bandwidth-limit-rule-show

Show information about the given qos bandwidth limit rule.

```
neutron qos-bandwidth-limit-rule-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  BANDWIDTH_LIMIT_RULE
  QOS_POLICY
```

**-f** <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table



- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.

**BANDWIDTH\_LIMIT\_RULE**

ID of bandwidth\_limit\_rule to look up.

**QOS\_POLICY**

ID or name of the QoS policy.

This command is provided by the python-neutronclient plugin.

**qos-bandwidth-limit-rule-update**

Update the given qos bandwidth limit rule.

```
neutron qos-bandwidth-limit-rule-update
  [--max-kbps MAX_KBPS]
  [--max-burst-kbps MAX_BURST_KBPS]
  BANDWIDTH_LIMIT_RULE
  QOS_POLICY
```

- max-kbps** <MAX\_KBPS>  
Maximum bandwidth in kbps.
- max-burst-kbps** <MAX\_BURST\_KBPS>  
Maximum burst bandwidth in kbps.

**BANDWIDTH\_LIMIT\_RULE**

ID of bandwidth\_limit\_rule to update.

**QOS\_POLICY**

ID or name of the QoS policy.

This command is provided by the python-neutronclient plugin.

## qos-dscp-marking-rule-create

Create a QoS DSCP marking rule.

```
neutron qos-dscp-marking-rule-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  --dscp-mark DSCP_MARK
  QOS_POLICY
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--tenant-id** TENANT\_ID  
The owner tenant ID.

**--dscp-mark** <DSCP\_MARK>  
DSCP mark: value can be 0, even numbers from 8-56, excluding 42, 44, 50, 52, and 54.

**QOS\_POLICY**  
ID or name of the QoS policy.

This command is provided by the python-neutronclient plugin.

## qos-dscp-marking-rule-delete

Delete a given qos dscp marking rule.

```
neutron qos-dscp-marking-rule-delete
    DSCP_MARKING_RULE
    [DSCP_MARKING_RULE ...]
    QOS_POLICY
```

### DSCP\_MARKING\_RULE

ID(s) of dscp\_marking\_rule to delete.

### QOS\_POLICY

ID or name of the QoS policy.

This command is provided by the python-neutronclient plugin.

## qos-dscp-marking-rule-list

List all QoS DSCP marking rules belonging to the specified policy.

```
neutron qos-dscp-marking-rule-list
    [-f {csv,json,table,value,yaml}]
    [-c COLUMN]
    [--quote {all,minimal,none,nonnumeric}]
    [--noindent]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
    [-D]
    [-F FIELD]
    [-P SIZE]
    [--sort-key FIELD]
    [--sort-dir {asc,desc}]
    QOS_POLICY
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment

variable `CLIFF_FIT_WIDTH=1` to always enable

**--print-empty**

Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN

specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

**--sort-ascending**

sort the column(s) in ascending order

**--sort-descending**

sort the column(s) in descending order

**-D, --show-details**

Show detailed information.

**-F** FIELD, **--field** FIELD

Specify the field(s) to be returned by server. You can repeat this option.

**-P** SIZE, **--page-size** SIZE

Specify retrieve unit of each request, then split one request to several requests.

**--sort-key** FIELD

Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of `sort_dir` and `sort_key` values. Extra `sort_dir` options are ignored. Missing `sort_dir` options use the default `asc` value.

**--sort-dir** {asc,desc}

Sorts the list in the specified direction. You can repeat this option.

**QOS\_POLICY**

ID or name of the QoS policy.

This command is provided by the `python-neutronclient` plugin.

## qos-dscp-marking-rule-show

Show information about the given qos dscp marking rule.

```
neutron qos-dscp-marking-rule-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  DSCP_MARKING_RULE
  QOS_POLICY
```

**-f** <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.

**DSCP\_MARKING\_RULE**

ID of dscp\_marking\_rule to look up.

**QOS\_POLICY**

ID or name of the QoS policy.

This command is provided by the python-neutronclient plugin.

**qos-dscp-marking-rule-update**

Update the given QoS DSCP marking rule.

```
neutron qos-dscp-marking-rule-update
  --dscp-mark DSCP_MARK
  DSCP_MARKING_RULE
  QOS_POLICY
```

- dscp-mark** <DSCP\_MARK>  
DSCP mark: value can be 0, even numbers from 8-56, excluding 42, 44, 50, 52, and 54.

**DSCP\_MARKING\_RULE**

ID of dscp\_marking\_rule to update.

**QOS\_POLICY**

ID or name of the QoS policy.

This command is provided by the python-neutronclient plugin.

## qos-minimum-bandwidth-rule-create

Create a qos minimum bandwidth rule.

```
neutron qos-minimum-bandwidth-rule-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  --min-kbps MIN_KBPS
  --direction
  {egress}
  QOS_POLICY
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--tenant-id** TENANT\_ID  
The owner tenant ID.

**--min-kbps** <MIN\_KBPS>  
QoS minimum bandwidth assurance, expressed in kilobits per second.

**--direction** <DIRECTION>  
Traffic direction.

**QOS\_POLICY**  
ID or name of the QoS policy.

This command is provided by the python-neutronclient plugin.

## qos-minimum-bandwidth-rule-delete

Delete a given qos minimum bandwidth rule.

```
neutron qos-minimum-bandwidth-rule-delete
  MINIMUM_BANDWIDTH_RULE
  [MINIMUM_BANDWIDTH_RULE ...]
  QOS_POLICY
```

### MINIMUM\_BANDWIDTH\_RULE

ID(s) of minimum\_bandwidth\_rule to delete.

### QOS\_POLICY

ID or name of the QoS policy.

This command is provided by the python-neutronclient plugin.

## qos-minimum-bandwidth-rule-list

List all qos minimum bandwidth rules belonging to the specified policy.

```
neutron qos-minimum-bandwidth-rule-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
  QOS_POLICY
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment

variable `CLIFF_FIT_WIDTH=1` to always enable

**--print-empty**

Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN

specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

**--sort-ascending**

sort the column(s) in ascending order

**--sort-descending**

sort the column(s) in descending order

**-D, --show-details**

Show detailed information.

**-F** FIELD, **--field** FIELD

Specify the field(s) to be returned by server. You can repeat this option.

**-P** SIZE, **--page-size** SIZE

Specify retrieve unit of each request, then split one request to several requests.

**--sort-key** FIELD

Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of `sort_dir` and `sort_key` values. Extra `sort_dir` options are ignored. Missing `sort_dir` options use the default `asc` value.

**--sort-dir** {asc,desc}

Sorts the list in the specified direction. You can repeat this option.

**QOS\_POLICY**

ID or name of the QoS policy.

This command is provided by the `python-neutronclient` plugin.

## qos-minimum-bandwidth-rule-show

Show information about the given qos minimum bandwidth rule.

```
neutron qos-minimum-bandwidth-rule-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  MINIMUM_BANDWIDTH_RULE
  QOS_POLICY
```

**-f** <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table



- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.

**MINIMUM\_BANDWIDTH\_RULE**

ID of minimum\_bandwidth\_rule to look up.

**QOS\_POLICY**

ID or name of the QoS policy.

This command is provided by the python-neutronclient plugin.

**qos-minimum-bandwidth-rule-update**

Update the given qos minimum bandwidth rule.

```
neutron qos-minimum-bandwidth-rule-update
  --min-kbps MIN_KBPS
  --direction
  {egress}
  MINIMUM_BANDWIDTH_RULE
  QOS_POLICY
```

- min-kbps** <MIN\_KBPS>  
QoS minimum bandwidth assurance, expressed in kilobits per second.

- direction** <DIRECTION>  
Traffic direction.

**MINIMUM\_BANDWIDTH\_RULE**

ID of minimum\_bandwidth\_rule to update.

**QOS\_POLICY**

ID or name of the QoS policy.

This command is provided by the python-neutronclient plugin.

## qos-policy-create

Create a qos policy.

```
neutron qos-policy-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  [--description DESCRIPTION]
  [--shared]
  NAME
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--tenant-id** TENANT\_ID  
The owner tenant ID.

**--description** <DESCRIPTION>  
Description of the QoS policy to be created.

**--shared**  
Accessible by other tenants. Set shared to True (default is False).

**NAME**  
Name of the QoS policy to be created.

This command is provided by the python-neutronclient plugin.

## qos-policy-delete

Delete a given qos policy.

```
neutron qos-policy-delete POLICY [POLICY ...]
```

### POLICY

ID(s) or name(s) of policy to delete.

This command is provided by the python-neutronclient plugin.

## qos-policy-list

List QoS policies that belong to a given tenant connection.

```
neutron qos-policy-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F FIELD, --field FIELD**  
Specify the field(s) to be returned by server. You can repeat this option.
- P SIZE, --page-size SIZE**  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key FIELD**  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of `sort_dir` and `sort_key` values. Extra `sort_dir` options are ignored. Missing `sort_dir` options use the default `asc` value.
- sort-dir {asc, desc}**  
Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## qos-policy-show

Show information of a given qos policy.

```
neutron qos-policy-show
  [-f {json, shell, table, value, yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  POLICY
```

- f <FORMATTER>, --format <FORMATTER>**  
the output format, defaults to table
- c COLUMN, --column COLUMN**  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix <PREFIX>**  
add a prefix to all variable names
- max-width <integer>**  
Maximum display width, <1 to disable. You can also use the `CLIFF_MAX_TERM_WIDTH` environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**-D, --show-details**

Show detailed information.

**-F FIELD, --field FIELD**

Specify the field(s) to be returned by server. You can repeat this option.

**POLICY**

ID or name of policy to look up.

This command is provided by the python-neutronclient plugin.

**qos-policy-update**

Update a given qos policy.

```
neutron qos-policy-update
  [--name NAME]
  [--description DESCRIPTION]
  [--shared | --no-shared]
  POLICY
```

**--name <NAME>**

Name of the QoS policy.

**--description <DESCRIPTION>**

Description of the QoS policy.

**--shared**

Accessible by other tenants. Set shared to True (default is False).

**--no-shared**

Not accessible by other tenants. Set shared to False.

**POLICY**

ID or name of policy to update.

This command is provided by the python-neutronclient plugin.

**quota-default-show**

Show default quotas for a given tenant.

```
neutron quota-default-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
```

(continues on next page)

(continued from previous page)

```

[--fit-width]
[--print-empty]
[--tenant-id tenant-id]

```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- tenant-id** tenant-id  
The owner tenant ID.

This command is provided by the python-neutronclient plugin.

## quota-delete

Delete defined quotas of a given tenant.

```
neutron quota-delete [--tenant-id tenant-id]
```

- tenant-id** tenant-id  
The owner tenant ID.

This command is provided by the python-neutronclient plugin.

## quota-list

List quotas of all tenants who have non-default quota values.

```

neutron quota-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]

```

(continues on next page)

(continued from previous page)

```
[--print-empty]
[--sort-column SORT_COLUMN]
[--sort-ascending | --sort-descending]
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric
- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order

This command is provided by the python-neutronclient plugin.

## quota-show

Show quotas for a given tenant.

```
neutron quota-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id tenant-id]
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- tenant-id** tenant-id  
The owner tenant ID.

This command is provided by the python-neutronclient plugin.

## quota-update

Update a given tenants quotas.

```
neutron quota-update
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id tenant-id]
  [--network networks]
  [--subnet subnets]
  [--port ports]
  [--router routers]
  [--floatingip floatingips]
  [--security-group security_groups]
  [--security-group-rule security_group_rules]
  [--vip vips]
  [--pool pools]
  [--member members]
  [--health-monitor health_monitors]
  [--loadbalancer loadbalancers]
  [--listener listeners]
  [--rbac-policy rbac_policies]
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table



- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- tenant-id** tenant-id  
The owner tenant ID.
- network** networks  
The limit of networks.
- subnet** subnets  
The limit of subnets.
- port** ports  
The limit of ports.
- router** routers  
The limit of routers.
- floatingip** floatingips  
The limit of floating IPs.
- security-group** security\_groups  
The limit of security groups.
- security-group-rule** security\_group\_rules  
The limit of security groups rules.
- vip** vips  
The limit of vips.
- pool** pools  
The limit of pools.
- member** members  
The limit of pool members.
- health-monitor** health\_monitors  
The limit of health monitors.
- loadbalancer** loadbalancers  
The limit of load balancers.
- listener** listeners  
The limit of listeners.

**--rbac-policy** rbac\_policies  
The limit of RBAC policies.

This command is provided by the python-neutronclient plugin.

## rbac-create

Create a RBAC policy for a given tenant.

```
neutron rbac-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  --type
  {network,qos-policy}
  [--target-tenant TARGET_TENANT]
  --action
  {access_as_external,access_as_shared}
  RBAC_OBJECT
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--tenant-id** TENANT\_ID  
The owner tenant ID.

**--type** <TYPE>  
Type of the object that RBAC policy affects.

**--target-tenant** <TARGET\_TENANT>  
ID of the tenant to which the RBAC policy will be enforced.

**--action** <ACTION>  
Action for the RBAC policy.

**RBAC\_OBJECT**  
ID or name of the RBAC object.

This command is provided by the python-neutronclient plugin.

### rbac-delete

Delete a RBAC policy.

```
neutron rbac-delete RBAC_POLICY [RBAC_POLICY ...]
```

**RBAC\_POLICY**  
ID(s) of rbac\_policy to delete.

This command is provided by the python-neutronclient plugin.

### rbac-list

List RBAC policies that belong to a given tenant.

```
neutron rbac-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- P** SIZE, **--page-size** SIZE  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key** FIELD  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.
- sort-dir** {asc, desc}  
Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## rbac-show

Show information of a given RBAC policy.

```
neutron rbac-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  RBAC_POLICY
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- D, --show-details**  
Show detailed information.
- F FIELD, --field FIELD**  
Specify the field(s) to be returned by server. You can repeat this option.
- RBAC\_POLICY**  
ID of rbac\_policy to look up.

This command is provided by the python-neutronclient plugin.

## rbac-update

Update RBAC policy for given tenant.

```
neutron rbac-update [--target-tenant TARGET_TENANT] RBAC_POLICY
```

- target-tenant** <TARGET\_TENANT>  
ID of the tenant to which the RBAC policy will be enforced.

**RBAC\_POLICY**  
ID of rbac\_policy to update.

This command is provided by the python-neutronclient plugin.

## router-create

Create a router for a given tenant.

```
neutron router-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
```

(continues on next page)

(continued from previous page)

```
[--admin-state-down]
[--description DESCRIPTION]
[--flavor FLAVOR]
[--distributed {True,False}]
[--ha {True,False}]
[--availability-zone-hint AVAILABILITY_ZONE]
NAME
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
  - c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
  - noindent**  
whether to disable indenting the JSON
  - prefix** <PREFIX>  
add a prefix to all variable names
  - max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
  - fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
  - print-empty**  
Print empty table if there is no data to show.
  - tenant-id** TENANT\_ID  
The owner tenant ID.
  - admin-state-down**  
Set admin state up to false.
  - description** <DESCRIPTION>  
Description of router.
  - flavor** <FLAVOR>  
ID or name of flavor.
  - distributed** {True,False}  
Create a distributed router.
  - ha** {True,False}  
Create a highly available router.
  - availability-zone-hint** AVAILABILITY\_ZONE  
Availability Zone for the router (requires availability zone extension, this option can be repeated).
- NAME**  
Name of the router to be created.

This command is provided by the python-neutronclient plugin.

## router-delete

Delete a given router.

```
neutron router-delete ROUTER [ROUTER ...]
```

### ROUTER

ID(s) or name(s) of router to delete.

This command is provided by the python-neutronclient plugin.

## router-gateway-clear

Remove an external network gateway from a router.

```
neutron router-gateway-clear ROUTER
```

### ROUTER

ID or name of the router.

This command is provided by the python-neutronclient plugin.

## router-gateway-set

Set the external network gateway for a router.

```
neutron router-gateway-set
  [--enable-snat]
  [--disable-snat]
  [--fixed-ip subnet_id=SUBNET,ip_address=IP_ADDR]
ROUTER
EXTERNAL-NETWORK
```

### --enable-snat

Enable source NAT on the router gateway.

### --disable-snat

Disable source NAT on the router gateway.

### --fixed-ip subnet\_id=SUBNET,ip\_address=IP\_ADDR

Desired IP and/or subnet on external network: subnet\_id=<name\_or\_id>,ip\_address=<ip>. You can specify both of subnet\_id and ip\_address or specify one of them as well. You can repeat this option.

### ROUTER

ID or name of the router.

### EXTERNAL-NETWORK

ID or name of the external network for the gateway.

This command is provided by the python-neutronclient plugin.

## router-interface-add

Add an internal network interface to a router.

```
neutron router-interface-add ROUTER INTERFACE
```

### ROUTER

ID or name of the router.

### INTERFACE

The format is SUBNET|subnet=SUBNET|port=PORT. Either a subnet or port must be specified. Both ID and name are accepted as SUBNET or PORT. Note that subnet= can be omitted when specifying a subnet.

This command is provided by the python-neutronclient plugin.

## router-interface-delete

Remove an internal network interface from a router.

```
neutron router-interface-delete ROUTER INTERFACE
```

### ROUTER

ID or name of the router.

### INTERFACE

The format is SUBNET|subnet=SUBNET|port=PORT. Either a subnet or port must be specified. Both ID and name are accepted as SUBNET or PORT. Note that subnet= can be omitted when specifying a subnet.

This command is provided by the python-neutronclient plugin.

## router-list

List routers that belong to a given tenant.

```
neutron router-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table



- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric
- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- P** SIZE, **--page-size** SIZE  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key** FIELD  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.
- sort-dir** {asc,desc}  
Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## router-list-on-l3-agent

List the routers on a L3 agent.

```
neutron router-list-on-l3-agent
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  L3_AGENT
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

**--sort-ascending**  
sort the column(s) in ascending order

**--sort-descending**  
sort the column(s) in descending order

**-D**, **--show-details**  
Show detailed information.

**-F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.

**L3\_AGENT**  
ID of the L3 agent to query.

This command is provided by the python-neutronclient plugin.

## router-port-list

List ports that belong to a given tenant, with specified router.

```
neutron router-port-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
  ROUTER
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric
- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.

**-F** FIELD, **--field** FIELD

Specify the field(s) to be returned by server. You can repeat this option.

**-P** SIZE, **--page-size** SIZE

Specify retrieve unit of each request, then split one request to several requests.

**--sort-key** FIELD

Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.

**--sort-dir** {asc, desc}

Sorts the list in the specified direction. You can repeat this option.

**ROUTER**

ID or name of the router to look up.

This command is provided by the python-neutronclient plugin.

## router-show

Show information of a given router.

```
neutron router-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  ROUTER
```

**-f** <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

**-c** COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

**--noindent**

whether to disable indenting the JSON

**--prefix** <PREFIX>

add a prefix to all variable names

**--max-width** <integer>

Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**-D, --show-details**  
Show detailed information.

**-F FIELD, --field FIELD**  
Specify the field(s) to be returned by server. You can repeat this option.

**ROUTER**  
ID or name of router to look up.

This command is provided by the python-neutronclient plugin.

## router-update

Update routers information.

```
neutron router-update
  [--name NAME]
  [--description DESCRIPTION]
  [--admin-state-up {True,False}]
  [--distributed {True,False}]
  [--route destination=CIDR,nexthop=IP_ADDR | --no-routes]
  ROUTER
```

**--name <NAME>**  
Updated name of the router.

**--description <DESCRIPTION>**  
Description of router.

**--admin-state-up {True,False}**  
Specify the administrative state of the router (True means Up).

**--distributed {True,False}**  
True means this router should operate in distributed mode.

**--route destination=CIDR,nexthop=IP\_ADDR**  
Route to associate with the router. You can repeat this option.

**--no-routes**  
Remove routes associated with the router.

**ROUTER**  
ID or name of router to update.

This command is provided by the python-neutronclient plugin.

## security-group-create

Create a security group.

```
neutron security-group-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
```

(continues on next page)

(continued from previous page)

```
[--fit-width]
[--print-empty]
[--tenant-id TENANT_ID]
[--description DESCRIPTION]
NAME
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
  - c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
  - noindent**  
whether to disable indenting the JSON
  - prefix** <PREFIX>  
add a prefix to all variable names
  - max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
  - fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
  - print-empty**  
Print empty table if there is no data to show.
  - tenant-id** TENANT\_ID  
The owner tenant ID.
  - description** <DESCRIPTION>  
Description of the security group to be created.
- NAME**  
Name of the security group to be created.

This command is provided by the python-neutronclient plugin.

### security-group-delete

Delete a given security group.

```
neutron security-group-delete SECURITY_GROUP [SECURITY_GROUP ...]
```

**SECURITY\_GROUP**  
ID(s) or name(s) of security\_group to delete.

This command is provided by the python-neutronclient plugin.

## security-group-list

List security groups that belong to a given tenant.

```

neutron security-group-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]

```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

**--sort-ascending**  
sort the column(s) in ascending order

**--sort-descending**  
sort the column(s) in descending order

**-D**, **--show-details**  
Show detailed information.

**-F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.

**-P** SIZE, **--page-size** SIZE

Specify retrieve unit of each request, then split one request to several requests.

**--sort-key** FIELD

Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.

**--sort-dir** {asc,desc}

Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## security-group-rule-create

Create a security group rule.

```
neutron security-group-rule-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  [--description DESCRIPTION]
  [--direction {ingress,egress}]
  [--ethertype ETHERTYPE]
  [--protocol PROTOCOL]
  [--port-range-min PORT_RANGE_MIN]
  [--port-range-max PORT_RANGE_MAX]
  [--remote-ip-prefix REMOTE_IP_PREFIX]
  [--remote-group-id REMOTE_GROUP]
  SECURITY_GROUP
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable



- print-empty**  
Print empty table if there is no data to show.
  - tenant-id** TENANT\_ID  
The owner tenant ID.
  - description** <DESCRIPTION>  
Description of security group rule.
  - direction** <DIRECTION>  
Direction of traffic: ingress/egress.
  - ethertype** <ETHERTYPE>  
IPv4/IPv6
  - protocol** <PROTOCOL>  
Protocol of packet. Allowed values are [icmp, icmpv6, tcp, udp] and integer representations [0-255].
  - port-range-min** <PORT\_RANGE\_MIN>  
Starting port range. For ICMP it is type.
  - port-range-max** <PORT\_RANGE\_MAX>  
Ending port range. For ICMP it is code.
  - remote-ip-prefix** <REMOTE\_IP\_PREFIX>  
CIDR to match on.
  - remote-group-id** REMOTE\_GROUP  
ID or name of the remote security group to which the rule is applied.
- SECURITY\_GROUP**  
ID or name of the security group to which the rule is added.

This command is provided by the python-neutronclient plugin.

### security-group-rule-delete

Delete a given security group rule.

```
neutron security-group-rule-delete
    SECURITY_GROUP_RULE
    [SECURITY_GROUP_RULE ...]
```

**SECURITY\_GROUP\_RULE**  
ID(s) of security\_group\_rule to delete.

This command is provided by the python-neutronclient plugin.

## security-group-rule-list

List security group rules that belong to a given tenant.

```
neutron security-group-rule-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
  [--no-nameconv]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

**--sort-ascending**  
sort the column(s) in ascending order

**--sort-descending**  
sort the column(s) in descending order

**-D**, **--show-details**  
Show detailed information.

**-F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.

- P** SIZE, **--page-size** SIZE  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key** FIELD  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.
- sort-dir** {asc,desc}  
Sorts the list in the specified direction. You can repeat this option.
- no-nameconv**  
Do not convert security group ID to its name.

This command is provided by the python-neutronclient plugin.

### security-group-rule-show

Show information of a given security group rule.

```
neutron security-group-rule-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  SECURITY_GROUP_RULE
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- D**, **--show-details**  
Show detailed information.

**-F** FIELD, **--field** FIELD

Specify the field(s) to be returned by server. You can repeat this option.

**SECURITY\_GROUP\_RULE**

ID of security\_group\_rule to look up.

This command is provided by the python-neutronclient plugin.

## security-group-show

Show information of a given security group.

```
neutron security-group-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  SECURITY_GROUP
```

**-f** <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

**-c** COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

**--noindent**

whether to disable indenting the JSON

**--prefix** <PREFIX>

add a prefix to all variable names

**--max-width** <integer>

Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**-D, --show-details**

Show detailed information.

**-F** FIELD, **--field** FIELD

Specify the field(s) to be returned by server. You can repeat this option.

**SECURITY\_GROUP**

ID or name of security\_group to look up.

This command is provided by the python-neutronclient plugin.

## security-group-update

Update a given security group.

```
neutron security-group-update
  [--name NAME]
  [--description DESCRIPTION]
  SECURITY_GROUP
```

**--name** <NAME>

Updated name of the security group.

**--description** <DESCRIPTION>

Updated description of the security group.

**SECURITY\_GROUP**

ID or name of security\_group to update.

This command is provided by the python-neutronclient plugin.

## service-provider-list

List service providers.

```
neutron service-provider-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- P** SIZE, **--page-size** SIZE  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key** FIELD  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.
- sort-dir** {asc, desc}  
Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## subnet-create

Create a subnet for a given tenant.

```
neutron subnet-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  [--name NAME]
  [--description DESCRIPTION]
  [--gateway GATEWAY_IP | --no-gateway]
  [--allocation-pool start=IP_ADDR,end=IP_ADDR]
  [--host-route destination=CIDR,nexthop=IP_ADDR]
  [--dns-nameserver DNS_NAMESERVER]
  [--disable-dhcp]
  [--enable-dhcp]
```

(continues on next page)

(continued from previous page)

```

[--ip-version {4,6}]
[--ipv6-ra-mode {dhcpv6-stateful,dhcpv6-stateless,slaac}]
[--ipv6-address-mode {dhcpv6-stateful,dhcpv6-stateless,slaac}]
[--subnetpool SUBNETPOOL]
[--use-default-subnetpool]
[--prefixlen PREFIX_LENGTH]
[--segment SEGMENT]
NETWORK
[CIDR]

```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- tenant-id** TENANT\_ID  
The owner tenant ID.
- name** <NAME>  
Name of this subnet.
- description** <DESCRIPTION>  
Description of this subnet.
- gateway** GATEWAY\_IP  
Gateway IP of this subnet.
- no-gateway**  
Do not configure a gateway for this subnet.
- allocation-pool** start=IP\_ADDR,end=IP\_ADDR  
Allocation pool IP addresses for this subnet (This option can be repeated).
- host-route** destination=CIDR,nexthop=IP\_ADDR  
Additional route (This option can be repeated).
- dns-nameserver** DNS\_NAMESERVER  
DNS name server for this subnet (This option can be repeated).
- disable-dhcp**  
Disable DHCP for this subnet.

**--enable-dhcp**

Enable DHCP for this subnet.

**--ip-version** <IP\_VERSION>

IP version to use, default is 4. Note that when subnetpool is specified, IP version is determined from the subnetpool and this option is ignored.

**--ipv6-ra-mode** <IPV6\_RA\_MODE>

IPv6 RA (Router Advertisement) mode.

**--ipv6-address-mode** <IPV6\_ADDRESS\_MODE>

IPv6 address mode.

**--subnetpool** SUBNETPOOL

ID or name of subnetpool from which this subnet will obtain a CIDR.

**--use-default-subnetpool**

Use default subnetpool for ip\_version, if it exists.

**--prefixlen** PREFIX\_LENGTH

Prefix length for subnet allocation from subnetpool.

**--segment** SEGMENT

ID of segment with which this subnet will be associated.

**NETWORK**

Network ID or name this subnet belongs to.

**CIDR**

CIDR of subnet to create.

This command is provided by the python-neutronclient plugin.

## subnet-delete

Delete a given subnet.

```
neutron subnet-delete SUBNET [SUBNET ...]
```

**SUBNET**

ID(s) or name(s) of subnet to delete.

This command is provided by the python-neutronclient plugin.

## subnet-list

List subnets that belong to a given tenant.

```
neutron subnet-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
```

(continues on next page)



(continued from previous page)

```

[--sort-column SORT_COLUMN]
[--sort-ascending | --sort-descending]
[-D]
[-F FIELD]
[-P SIZE]
[--sort-key FIELD]
[--sort-dir {asc,desc}]

```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric
- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D**, **--show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- P** SIZE, **--page-size** SIZE  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key** FIELD  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.
- sort-dir** {asc,desc}  
Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## subnet-show

Show information of a given subnet.

```
neutron subnet-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  SUBNET
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**-D**, **--show-details**  
Show detailed information.

**-F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.

**SUBNET**  
ID or name of subnet to look up.

This command is provided by the python-neutronclient plugin.

## subnet-update

Update subnets information.

```

neutron subnet-update
  [--name NAME]
  [--description DESCRIPTION]
  [--gateway GATEWAY_IP | --no-gateway]
  [--allocation-pool start=IP_ADDR,end=IP_ADDR]
  [--host-route destination=CIDR,nexthop=IP_ADDR]
  [--dns-nameserver DNS_NAMESERVER]
  [--disable-dhcp]
  [--enable-dhcp]
  SUBNET

```

**--name** <NAME>

Name of this subnet.

**--description** <DESCRIPTION>

Description of this subnet.

**--gateway** GATEWAY\_IP

Gateway IP of this subnet.

**--no-gateway**

Do not configure a gateway for this subnet.

**--allocation-pool** start=IP\_ADDR,end=IP\_ADDR

Allocation pool IP addresses for this subnet (This option can be repeated).

**--host-route** destination=CIDR,nexthop=IP\_ADDR

Additional route (This option can be repeated).

**--dns-nameserver** DNS\_NAMESERVER

DNS name server for this subnet (This option can be repeated).

**--disable-dhcp**

Disable DHCP for this subnet.

**--enable-dhcp**

Enable DHCP for this subnet.

**SUBNET**

ID or name of subnet to update.

This command is provided by the python-neutronclient plugin.

## subnetpool-create

Create a subnetpool for a given tenant.

```

neutron subnetpool-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]

```

(continues on next page)

(continued from previous page)

```
[--fit-width]
[--print-empty]
[--tenant-id TENANT_ID]
[--description DESCRIPTION]
[--min-prefixlen MIN_PREFIXLEN]
[--max-prefixlen MAX_PREFIXLEN]
[--default-prefixlen DEFAULT_PREFIXLEN]
--pool-prefix PREFIXES
[--is-default {True,False}]
[--shared]
[--address-scope ADDRSCOPE]
NAME
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- tenant-id** TENANT\_ID  
The owner tenant ID.
- description** <DESCRIPTION>  
Description of subnetpool.
- min-prefixlen** <MIN\_PREFIXLEN>  
Subnetpool minimum prefix length.
- max-prefixlen** <MAX\_PREFIXLEN>  
Subnetpool maximum prefix length.
- default-prefixlen** <DEFAULT\_PREFIXLEN>  
Subnetpool default prefix length.
- pool-prefix** <PREFIXES>  
Subnetpool prefixes (This option can be repeated).
- is-default** {True,False}  
Specify whether this should be the default subnetpool (True meaning default).

**--shared**

Set the subnetpool as shared.

**--address-scope** ADDRSCOPE

ID or name of the address scope with which the subnetpool is associated. Prefixes must be unique across address scopes.

**NAME**

Name of the subnetpool to be created.

This command is provided by the python-neutronclient plugin.

**subnetpool-delete**

Delete a given subnetpool.

```
neutron subnetpool-delete SUBNETPOOL [SUBNETPOOL ...]
```

**SUBNETPOOL**

ID(s) or name(s) of subnetpool to delete.

This command is provided by the python-neutronclient plugin.

**subnetpool-list**

List subnetpools that belong to a given tenant.

```
neutron subnetpool-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- P** SIZE, **--page-size** SIZE  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key** FIELD  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.
- sort-dir** {asc,desc}  
Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## subnetpool-show

Show information of a given subnetpool.

```
neutron subnetpool-show
[-f {json,shell,table,value,yaml}]
[-c COLUMN]
[--noindent]
[--prefix PREFIX]
[--max-width <integer>]
[--fit-width]
[--print-empty]
[-D]
[-F FIELD]
SUBNETPOOL
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- D**, **--show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.

**SUBNETPOOL**

ID or name of subnetpool to look up.

This command is provided by the python-neutronclient plugin.

**subnetpool-update**

Update subnetpools information.

```
neutron subnetpool-update
  [--description DESCRIPTION]
  [--min-prefixlen MIN_PREFIXLEN]
  [--max-prefixlen MAX_PREFIXLEN]
  [--default-prefixlen DEFAULT_PREFIXLEN]
  [--pool-prefix PREFIXES]
  [--is-default {True,False}]
  [--name NAME]
  [--address-scope ADDRSCOPE | --no-address-scope]
  SUBNETPOOL
```

- description** <DESCRIPTION>  
Description of subnetpool.
- min-prefixlen** <MIN\_PREFIXLEN>  
Subnetpool minimum prefix length.
- max-prefixlen** <MAX\_PREFIXLEN>  
Subnetpool maximum prefix length.

**--default-prefixlen** <DEFAULT\_PREFIXLEN>  
Subnetpool default prefix length.

**--pool-prefix** <PREFIXES>  
Subnetpool prefixes (This option can be repeated).

**--is-default** {True,False}  
Specify whether this should be the default subnetpool (True meaning default).

**--name** <NAME>  
Updated name of the subnetpool.

**--address-scope** ADDRSCOPE  
ID or name of the address scope with which the subnetpool is associated. Prefixes must be unique across address scopes.

**--no-address-scope**  
Detach subnetpool from the address scope.

**SUBNETPOOL**  
ID or name of subnetpool to update.

This command is provided by the python-neutronclient plugin.

## tag-add

Add a tag into the resource.

```
neutron tag-add
  --resource-type
  {network,subnet,port,router,subnetpool}
  --resource RESOURCE
  --tag TAG
```

**--resource-type** <RESOURCE\_TYPE>  
Resource Type.

**--resource** <RESOURCE>  
Resource name or ID.

**--tag** <TAG>  
Tag to be added.

This command is provided by the python-neutronclient plugin.

## tag-remove

Remove a tag on the resource.

```
neutron tag-remove
  --resource-type
  {network,subnet,port,router,subnetpool}
  --resource RESOURCE
  [--all | --tag TAG]
```



**--resource-type** <RESOURCE\_TYPE>

Resource Type.

**--resource** <RESOURCE>

Resource name or ID.

**--all**

Remove all tags on the resource.

**--tag** <TAG>

Tag to be removed.

This command is provided by the python-neutronclient plugin.

## tag-replace

Replace all tags on the resource.

```
neutron tag-replace
  --resource-type
  {network,subnet,port,router,subnetpool}
  --resource RESOURCE
  --tag TAG
```

**--resource-type** <RESOURCE\_TYPE>

Resource Type.

**--resource** <RESOURCE>

Resource name or ID.

**--tag** TAG

Tag (This option can be repeated).

This command is provided by the python-neutronclient plugin.

## vpn-endpoint-group-create

Create a VPN endpoint group.

```
neutron vpn-endpoint-group-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  [--name NAME]
  [--description DESCRIPTION]
  --type TYPE
  --value ENDPOINTS
```

**-f** <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- tenant-id** TENANT\_ID  
The owner tenant ID.
- name** <NAME>  
Set a name for the endpoint group.
- description** <DESCRIPTION>  
Set a description for the endpoint group.
- type** <TYPE>  
Type of endpoints in group (e.g. subnet, cidr, vlan).
- value** <ENDPOINTS>  
Endpoint(s) for the group. Must all be of the same type.

This command is provided by the python-neutronclient plugin.

### vpn-endpoint-group-delete

Delete a given VPN endpoint group.

```
neutron vpn-endpoint-group-delete ENDPOINT_GROUP [ENDPOINT_GROUP ...]
```

#### ENDPOINT\_GROUP

ID(s) or name(s) of endpoint\_group to delete.

This command is provided by the python-neutronclient plugin.

## vpn-endpoint-group-list

List VPN endpoint groups that belong to a given tenant.

```
neutron vpn-endpoint-group-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

**--sort-ascending**  
sort the column(s) in ascending order

**--sort-descending**  
sort the column(s) in descending order

**-D**, **--show-details**  
Show detailed information.

**-F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.

**-P** SIZE, **--page-size** SIZE

Specify retrieve unit of each request, then split one request to several requests.

**--sort-key** FIELD

Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.

**--sort-dir** {asc,desc}

Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## vpn-endpoint-group-show

Show a specific VPN endpoint group.

```
neutron vpn-endpoint-group-show
[-f {json,shell,table,value,yaml}]
[-c COLUMN]
[--noindent]
[--prefix PREFIX]
[--max-width <integer>]
[--fit-width]
[--print-empty]
[-D]
[-F FIELD]
ENDPOINT_GROUP
```

**-f** <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

**-c** COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

**--noindent**

whether to disable indenting the JSON

**--prefix** <PREFIX>

add a prefix to all variable names

**--max-width** <integer>

Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**-D**, **--show-details**

Show detailed information.

**-F** FIELD, **--field** FIELD

Specify the field(s) to be returned by server. You can repeat this option.

**ENDPOINT\_GROUP**

ID or name of endpoint\_group to look up.

This command is provided by the python-neutronclient plugin.

**vpn-endpoint-group-update**

Update a given VPN endpoint group.

```
neutron vpn-endpoint-group-update
  [--name NAME]
  [--description DESCRIPTION]
  ENDPOINT_GROUP
```

**--name** <NAME>

Set a name for the endpoint group.

**--description** <DESCRIPTION>

Set a description for the endpoint group.

**ENDPOINT\_GROUP**

ID or name of endpoint\_group to update.

This command is provided by the python-neutronclient plugin.

**vpn-ikepolicy-create**

Create an IKE policy.

```
neutron vpn-ikepolicy-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  [--description DESCRIPTION]
  [--auth-algorithm {sha1,sha256,sha384,sha512}]
  [--encryption-algorithm ENCRYPTION_ALGORITHM]
  [--phase1-negotiation-mode {main}]
  [--ike-version {v1,v2}]
  [--pfs PFS]
  [--lifetime units=UNITS,value=VALUE]
  NAME
```

**-f** <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

**-c** COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

**--noindent**

whether to disable indenting the JSON

- prefix** <PREFIX>  
add a prefix to all variable names
  - max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
  - fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
  - print-empty**  
Print empty table if there is no data to show.
  - tenant-id** TENANT\_ID  
The owner tenant ID.
  - description** <DESCRIPTION>  
Description of the IKE policy.
  - auth-algorithm** <AUTH\_ALGORITHM>  
Authentication algorithm, default:sha1.
  - encryption-algorithm** <ENCRYPTION\_ALGORITHM>  
Encryption algorithm, default:aes-128.
  - phase1-negotiation-mode** <PHASE1\_NEGOTIATION\_MODE>  
IKE Phase1 negotiation mode, default:main.
  - ike-version** <IKE\_VERSION>  
IKE version for the policy, default:v1.
  - pfs** <PFS>  
Perfect Forward Secrecy, default:group5.
  - lifetime** units=UNITS,value=VALUE  
IKE lifetime attributes. units-seconds, default:seconds. value-non negative integer, default:3600.
- NAME**  
Name of the IKE policy.

This command is provided by the python-neutronclient plugin.

### vpn-ikepolicy-delete

Delete a given IKE policy.

```
neutron vpn-ikepolicy-delete IKEPOLICY [IKEPOLICY ...]
```

#### **IKEPOLICY**

ID(s) or name(s) of IKE policy to delete.

This command is provided by the python-neutronclient plugin.

## vpn-ikepolicy-list

List IKE policies that belong to a tenant.

```
neutron vpn-ikepolicy-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric

**--noindent**  
whether to disable indenting the JSON

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

**--sort-ascending**  
sort the column(s) in ascending order

**--sort-descending**  
sort the column(s) in descending order

**-D**, **--show-details**  
Show detailed information.

**-F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.

**-P** SIZE, **--page-size** SIZE

Specify retrieve unit of each request, then split one request to several requests.

**--sort-key** FIELD

Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.

**--sort-dir** {asc,desc}

Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## vpn-ikepolicy-show

Show information of a given IKE policy.

```
neutron vpn-ikepolicy-show
[-f {json,shell,table,value,yaml}]
[-c COLUMN]
[--noindent]
[--prefix PREFIX]
[--max-width <integer>]
[--fit-width]
[--print-empty]
[-D]
[-F FIELD]
IKEPOLICY
```

**-f** <FORMATTER>, **--format** <FORMATTER>

the output format, defaults to table

**-c** COLUMN, **--column** COLUMN

specify the column(s) to include, can be repeated to show multiple columns

**--noindent**

whether to disable indenting the JSON

**--prefix** <PREFIX>

add a prefix to all variable names

**--max-width** <integer>

Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**

Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**

Print empty table if there is no data to show.

**-D**, **--show-details**

Show detailed information.

**-F** FIELD, **--field** FIELD

Specify the field(s) to be returned by server. You can repeat this option.



**IKEPOLICY**

ID or name of IKE policy to look up.

This command is provided by the python-neutronclient plugin.

**vpn-ikepolicy-update**

Update a given IKE policy.

```
neutron vpn-ikepolicy-update
  [--name NAME]
  [--description DESCRIPTION]
  [--auth-algorithm {sha1,sha256,sha384,sha512}]
  [--encryption-algorithm ENCRYPTION_ALGORITHM]
  [--phase1-negotiation-mode {main}]
  [--ike-version {v1,v2}]
  [--pfs PFS]
  [--lifetime units=UNITS,value=VALUE]
  IKEPOLICY
```

**--name** <NAME>

Updated name of the IKE policy.

**--description** <DESCRIPTION>

Description of the IKE policy.

**--auth-algorithm** <AUTH\_ALGORITHM>

Authentication algorithm, default:sha1.

**--encryption-algorithm** <ENCRYPTION\_ALGORITHM>

Encryption algorithm, default:aes-128.

**--phase1-negotiation-mode** <PHASE1\_NEGOTIATION\_MODE>

IKE Phase1 negotiation mode, default:main.

**--ike-version** <IKE\_VERSION>

IKE version for the policy, default:v1.

**--pfs** <PFS>

Perfect Forward Secrecy, default:group5.

**--lifetime** **units**=UNITS,value=VALUE

IKE lifetime attributes. **units**-seconds, default:seconds. **value**-non negative integer, default:3600.

**IKEPOLICY**

ID or name of IKE policy to update.

This command is provided by the python-neutronclient plugin.

## vpn-ipsecpolicy-create

Create an IPsec policy.

```
neutron vpn-ipsecpolicy-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  [--auth-algorithm {sha1,sha256,sha384,sha512}]
  [--description DESCRIPTION]
  [--encapsulation-mode {tunnel,transport}]
  [--encryption-algorithm ENCRYPTION_ALGORITHM]
  [--lifetime units=UNITS,value=VALUE]
  [--pfs PFS]
  [--transform-protocol {esp,ah,ah-esp}]
  NAME
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--tenant-id** TENANT\_ID  
The owner tenant ID.

**--auth-algorithm** <AUTH\_ALGORITHM>  
Authentication algorithm for IPsec policy, default:sha1.

**--description** <DESCRIPTION>  
Description of the IPsec policy.

**--encapsulation-mode** <ENCAPSULATION\_MODE>  
Encapsulation mode for IPsec policy, default:tunnel.

**--encryption-algorithm** <ENCRYPTION\_ALGORITHM>  
Encryption algorithm for IPsec policy, default:aes-128.

- lifetime** units=UNITS,value=VALUE  
IPsec lifetime attributes. units-seconds, default:seconds. value-non negative integer, default:3600.
- pfs** <PFS>  
Perfect Forward Secrecy for IPsec policy, default:group5.
- transform-protocol** <TRANSFORM\_PROTOCOL>  
Transform protocol for IPsec policy, default:esp.

**NAME**

Name of the IPsec policy.

This command is provided by the python-neutronclient plugin.

**vpn-ipsecpolicy-delete**

Delete a given IPsec policy.

```
neutron vpn-ipsecpolicy-delete IPSECPOLICY [IPSECPOLICY ...]
```

**IPSECPOLICY**

ID(s) or name(s) of IPsec policy to delete.

This command is provided by the python-neutronclient plugin.

**vpn-ipsecpolicy-list**

List IPsec policies that belong to a given tenant connection.

```
neutron vpn-ipsecpolicy-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric
- noindent**  
whether to disable indenting the JSON

- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- P** SIZE, **--page-size** SIZE  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key** FIELD  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.
- sort-dir** {asc,desc}  
Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.

## vpn-ipsecpolicy-show

Show information of a given IPsec policy.

```
neutron vpn-ipsecpolicy-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  IPSECPOLICY
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- D**, **--show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.

**IPSECPOLICY**

ID or name of IPsec policy to look up.

This command is provided by the python-neutronclient plugin.

**vpn-ipsecpolicy-update**

Update a given IPsec policy.

```
neutron vpn-ipsecpolicy-update
  [--name NAME]
  [--auth-algorithm {sha1,sha256,sha384,sha512}]
  [--description DESCRIPTION]
  [--encapsulation-mode {tunnel,transport}]
  [--encryption-algorithm ENCRYPTION_ALGORITHM]
  [--lifetime units=UNITS,value=VALUE]
  [--pfs PFS]
  [--transform-protocol {esp,ah,ah-esp}]
  IPSECPOLICY
```

- name** <NAME>  
Updated name of the IPsec policy.
- auth-algorithm** <AUTH\_ALGORITHM>  
Authentication algorithm for IPsec policy, default:sha1.
- description** <DESCRIPTION>  
Description of the IPsec policy.

- encapsulation-mode** <ENCAPSULATION\_MODE>  
Encapsulation mode for IPsec policy, default:tunnel.
- encryption-algorithm** <ENCRYPTION\_ALGORITHM>  
Encryption algorithm for IPsec policy, default:aes-128.
- lifetime** units=UNITS,value=VALUE  
IPsec lifetime attributes. units-seconds, default:seconds. value-non negative integer, default:3600.
- pfs** <PFS>  
Perfect Forward Secrecy for IPsec policy, default:group5.
- transform-protocol** <TRANSFORM\_PROTOCOL>  
Transform protocol for IPsec policy, default:esp.

#### IPSECPOLICY

ID or name of IPsec policy to update.

This command is provided by the python-neutronclient plugin.

### vpn-service-create

Create a VPN service.

```
neutron vpn-service-create
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--tenant-id TENANT_ID]
  [--admin-state-down]
  [--name NAME]
  [--description DESCRIPTION]
ROUTER
[SUBNET]
```

- f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table
- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- noindent**  
whether to disable indenting the JSON
- prefix** <PREFIX>  
add a prefix to all variable names
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**--tenant-id** TENANT\_ID  
The owner tenant ID.

**--admin-state-down**  
Set admin state up to false.

**--name** <NAME>  
Name for the VPN service.

**--description** <DESCRIPTION>  
Description for the VPN service.

**ROUTER**  
Router unique identifier for the VPN service.

**SUBNET**  
[DEPRECATED in Mitaka] Unique identifier for the local private subnet.

This command is provided by the python-neutronclient plugin.

### vpn-service-delete

Delete a given VPN service.

```
neutron vpn-service-delete VPNSERVICE [VPNSERVICE ...]
```

**VPNSERVICE**  
ID(s) or name(s) of VPN service to delete.

This command is provided by the python-neutronclient plugin.

### vpn-service-list

List VPN service configurations that belong to a given tenant.

```
neutron vpn-service-list
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [-D]
  [-F FIELD]
  [-P SIZE]
  [--sort-key FIELD]
  [--sort-dir {asc,desc}]
```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

- c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns
- quote** <QUOTE\_MODE>  
when to include quotes, defaults to nonnumeric
- noindent**  
whether to disable indenting the JSON
- max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.
- fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable
- print-empty**  
Print empty table if there is no data to show.
- sort-column** SORT\_COLUMN  
specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
- sort-ascending**  
sort the column(s) in ascending order
- sort-descending**  
sort the column(s) in descending order
- D, --show-details**  
Show detailed information.
- F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.
- P** SIZE, **--page-size** SIZE  
Specify retrieve unit of each request, then split one request to several requests.
- sort-key** FIELD  
Sorts the list by the specified fields in the specified directions. You can repeat this option, but you must specify an equal number of sort\_dir and sort\_key values. Extra sort\_dir options are ignored. Missing sort\_dir options use the default asc value.
- sort-dir** {asc,desc}  
Sorts the list in the specified direction. You can repeat this option.

This command is provided by the python-neutronclient plugin.



## vpn-service-show

Show information of a given VPN service.

```

neutron vpn-service-show
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN]
  [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [-D]
  [-F FIELD]
  VPNSERVICE

```

**-f** <FORMATTER>, **--format** <FORMATTER>  
the output format, defaults to table

**-c** COLUMN, **--column** COLUMN  
specify the column(s) to include, can be repeated to show multiple columns

**--noindent**  
whether to disable indenting the JSON

**--prefix** <PREFIX>  
add a prefix to all variable names

**--max-width** <integer>  
Maximum display width, <1 to disable. You can also use the CLIFF\_MAX\_TERM\_WIDTH environment variable, but the parameter takes precedence.

**--fit-width**  
Fit the table to the display width. Implied if max-width greater than 0. Set the environment variable CLIFF\_FIT\_WIDTH=1 to always enable

**--print-empty**  
Print empty table if there is no data to show.

**-D**, **--show-details**  
Show detailed information.

**-F** FIELD, **--field** FIELD  
Specify the field(s) to be returned by server. You can repeat this option.

### VPNSERVICE

ID or name of VPN service to look up.

This command is provided by the python-neutronclient plugin.

## vpn-service-update

Update a given VPN service.

```
neutron vpn-service-update
  [--name NAME]
  [--description DESCRIPTION]
  [--admin-state-up {True,False}]
  VPNSERVICE
```

**--name** <NAME>

Name for the VPN service.

**--description** <DESCRIPTION>

Description for the VPN service.

**--admin-state-up** {True,False}

Update the admin state for the VPN Service.(True means UP)

**VPNSERVICE**

ID or name of VPN service to update.

This command is provided by the python-neutronclient plugin.

## 1.2 neutronclient Python API

### 1.2.1 Basic Usage

First create a client instance using a keystoneauth Session. For more information on this keystoneauth API, see [Using Sessions](#).

```
from keystoneauth1 import identity
from keystoneauth1 import session
from neutronclient.v2_0 import client
username='username'
password='password'
project_name='demo'
project_domain_id='default'
user_domain_id='default'
auth_url='http://auth.example.com:5000/v3'
auth = identity.Password(auth_url=auth_url,
                        username=username,
                        password=password,
                        project_name=project_name,
                        project_domain_id=project_domain_id,
                        user_domain_id=user_domain_id)
sess = session.Session(auth=auth)
neutron = client.Client(session=sess)
```

If you are using Identity v2.0 API (DEPRECATED), create an auth plugin using the appropriate parameters and `keystoneauth1.identity` will handle Identity API version discovery. Then you can create a Session and a Neutronclient just like the previous example.

```
auth = identity.Password(auth_url=auth_url,
                        username=username,
                        password=password,
                        project_name=project_name)
# create a Session and a Neutronclient
```

Now you can call various methods on the client instance.

```
network = {'name': 'mynetwork', 'admin_state_up': True}
neutron.create_network({'network':network})
networks = neutron.list_networks(name='mynetwork')
print networks
network_id = networks['networks'][0]['id']
neutron.delete_network(network_id)
```

Alternatively, you can create a client instance using an auth token and a service endpoint URL directly.

```
from neutronclient.v2_0 import client
neutron = client.Client(endpoint_url='http://192.168.206.130:9696/',
                       token='d3f9226f27774f338019aa2611112ef6')
```

You can get X-Openstack-Request-Id as request\_ids from the result.

```
network = {'name': 'mynetwork', 'admin_state_up': True}
neutron.create_network({'network':network})
networks = neutron.list_networks(name='mynetwork')
print networks.request_ids
# -> ['req-978a0160-7ab0-44f0-8a93-08e9a4e785fa']
```



## CONTRIBUTOR GUIDE

In the *Contributor Guide*, you will find information on neutronclients lower level programming details or APIs as well as the transition to OpenStack client.

### 2.1 Contributor Guide

In the Contributor Guide, you will find information on neutronclients lower level programming details or APIs as well as the transition to OpenStack client.

#### 2.1.1 Client command extension support

The client command extension adds support for extending the neutron client while considering ease of creation. Extensions strongly conform to preexisting neutron commands (`/neutron/v2_0/`).

A sample extension can be seen at: `neutronclient/neutron/v2_0/contrib/_fox_sockets.py`

#### Minimum requirements from an extension

- NeutronClientExtension subclasses must have a `shell_command` class variable if the command is to be available to the CLI (`shell.py`)

Example: `neutronclient.neutron.v2_0.contrib._fox_sockets.FoxInSocketsList`

#### Minimum requirements to use canonical neutron CRUD commands framework

Neutron commands are cliff commands, commands in extension can use their own way to finish their tasks. But if they want to make use of the canonical neutron CRUD commands framework, the extension should:

- have a class that subclasses NeutronClientExtension to provide the requisite resource name, version support, and resource collection and object paths for a resource the commands will process.

Example: `neutronclient.neutron.v2_0.contrib._fox_sockets.FoxInSocket`

- have a class that subclasses from the ClientExtensionList to provide resource object list function. This is because most commands need the list function to get object ID via `neutronclient.neutron.v2_0.__init__.find_resource_by_id`.

Example: `neutronclient.neutron.v2_0.contrib._fox_sockets.FoxInSocketsList`

- if needed, subclass `ClientExtensionUpdate` to implement update of the resource object.  
Example: `neutronclient.neutron.v2_0.contrib._fox_sockets.FoxInSocketsUpdate`
- if needed, subclass `ClientExtensionDelete` to implement deletion of the resource object.  
Example: `neutronclient.neutron.v2_0.contrib._fox_sockets.FoxInSocketsDelete`
- if needed, subclass `ClientExtensionShow` to get the detail of the resource object.  
Example: `neutronclient.neutron.v2_0.contrib._fox_sockets.FoxInSocketsShow`

## Precedence of command loading

- hard coded commands are loaded first
- external commands (installed in the environment) are loaded then

Commands that have the same name will be overwritten by commands that are loaded later. To change the execution of a command for your particular extension you only need to override the `execute` method.

Currently this extension support is limited to top-level resources. Parent/child relationships may be added if desired.

## neutronclient.extension entry\_point

To activate the commands in a specific extension module, add an entry in `setup.cfg` under `neutronclient.extension`. For example:

```
[entry_points]
neutronclient.extension =
    fox_sockets = neutronclient.neutron.v2_0.contrib._fox_sockets
```

### 2.1.2 CLI Option Guideline

This document describes the conventions of neutron CLI options.

#### General conventions

1. Option names should be delimited by a hyphen instead of an underscore. This is the common guidelines across all OpenStack CLIs.
  - Good: `--ip-version`
  - Not Good: `--ip_version`
2. Use at least one required option for `*-create` command. If all options are optional, we typically use `name` field as a required option.
3. When you need to specify an ID of a resource, it is better to provide another way to specify the resource like `name` or other reasonable field.
4. If an attribute name in API is `foo_id`, the corresponding option should be `--foo` instead of `--foo-id`.
  - It is because we usually support ID and `name` to specify a resource.

5. Do not use `nargs='?'` without a special reason.
  - The behavior of `nargs='?'` option for python argparse is bit tricky and may lead to unexpected option parsing different from the help message. The detail is described in the *Background section* below.
6. (option) Avoid using positional options as much as possible.
  - Positional arguments should be limited to attributes which will be required in the long future.
7. We honor existing options and should keep compatibilities when adding or changing options.

## Options for boolean value

Use the form of `--option-name {True|False}`.

- For a new option, it is recommended.
- It is suggested to use `common.utils.add_boolean_argument` in an implementation. It allows `true/false` in addition to `True/False`.
- For existing options, migration to the recommended form is not necessarily required. All backward-compatibility should be kept without reasonable reasons.

## Options for dict value

Some API attributes take a dictionary.

`--foo key1=val1, key2=val2` is usually used.

This means `{"key1": "val1", "key2": "val2"}` is passed in the API layer.

Examples:

- `--host-route destination=CIDR, nexthop=IP_ADDR` for a subnet
- `--fixed-ip subnet_id=SUBNET, ip_address=IP_ADDR` for a port.

## Options for list value

Some attributes take a list.

In this case, we usually use:

- Define an option per element (Use a singular form as an option name)
- Allow to specify the option multiple times

For Example, **port-create** has `--security-group` option. `--security-group SG1 --security-group SG2` generates `{"security_groups": ["SG1", "SG2"]}` in the API layer.

This convention applies to a case of a list of dict. `--allocation-pool` and `--host-route` for a subnet are examples.

## Compatibility with extra arguments

*extra arguments* supports various types of option specifications. At least the following patterns needs to be considered when defining a new option. For more detail, see *Extra arguments for create/update operation*.

- Normal options with value
- Boolean options : `--foo True, --bar=False`
- List options : `--bars list=true val1 val2, --bars val1 val2`
- Dict options : `--foo type=dict key1=val1, key2=val2`
- List of Dict options : `--bars list=true type=dict key1=val1, key2=val2 key3=val3, key4=val4`
- `action=clear`

For normal options with value, there are four patterns to specify an option as extra arguments.

- `--admin-state-up True` (a space between option name and value)
- `--admin-state-up=True` (= between option name and value)
- `--admin_state_up True` (underscore is used as delimiter)
- `--admin_state_up=True` (underscore is used as delimiter)

## Background

There are a lot of opinions on which form of options are better or not. This section tries to capture the reason of the current choice.

## Use at least one required option

As a convention, **neutron** CLI requires one required argument.

If all options are optional in the API level and we have `name` field, we usually use `name` as a required parameter. Requiring at least one argument has the following benefits:

- If we run `neutron *-create` without a required argument, we will have a brief help message without detail option help. It is convenient.
- We can avoid miss operation by just hitting `neutron *-create`. Requiring at least one parameter is a good balance.

Even though we can change this convention to allow to create a resource without `name` field, it will bring confusions to existing users.

There may be opinion that it is inconsistent with API level requirement or Horizon behavior, but even if `neutron` CLI requires `name` field there is no bad impact on regular users. Considering possible confusion if we change it, it looks better to keep it as-is.



## Options for Boolean value

- `--enable-foo/--disable-foo` or similar patterns (including `--admin-state-down`) is not suggested because we need two exclusive options for one attribute in REST API. It is meaningless.
- It is not recommended to have an option only to specify non-default value. For example, we have `--shared` or `--admin-state-down` options for `net-create`. This form only works for `*-create` and does not work for `*-update`. It leads to having different options for `*-create` and `*-update`.
- A flag option like `--enable-dhcp` (without value) also has a problem when considering the compatibility with *extra argument*. We can specify `-enable-dhcp True/False` or `--enable-dhcp=True/False` in the *extra argument* mechanism. If we introduce `--enable-dhcp` (without value), the form of `-enable-dhcp True/False` cannot be used now. This is another reason we don't use a flag style option for a boolean parameter.

## Avoid using nargs in positional or optional arguments

The behavior of `nargs='?'` option for python argparse is bit tricky. When we use `nargs='?'` and if the order of command-line options is changed then the command-line parser may fail to parse the arguments correctly. Two examples of such failures are provided below.

Example 1: This example shows how the actual behavior can differ from the provided help message. In the below block, help message at [5] says `--bb CC` is a valid format but the argument parsing for the same format fails at [7].

```
In [1]: import argparse
In [2]: parser = argparse.ArgumentParser()
In [3]: parser.add_argument('--bb', nargs='?')
In [4]: parser.add_argument('cc')

In [5]: parser.print_help()
usage: ipython [-h] [--bb [BB]] cc

positional arguments:
  cc

optional arguments:
  -h, --help  show this help message and exit
  --bb [BB]

In [6]: parser.parse_args('--bb 1 X'.split())
Out[6]: Namespace(bb='1', cc='X')

In [7]: parser.parse_args('--bb X'.split())
usage: ipython [-h] [--bb [BB]] cc
ipython: error: too few arguments
An exception has occurred, use %tb to see the full traceback.

SystemExit: 2
```

Example 2: This example shows how fragile `nargs='?'` can be when user specifies options in different order from the help message.

```
In [1]: import argparse
In [2]: parser = argparse.ArgumentParser()
In [3]: parser.add_argument('--a', help='option a')
In [4]: parser.add_argument('--b', help='option b')
In [5]: parser.add_argument('x', help='positional arg X')
In [6]: parser.add_argument('y', nargs='?', help='positional arg Y')
In [7]: parser.print_help()
usage: ipython [-h] [--a A] [--b B] x [y]

positional arguments:
  x          positional arg X
  y          positional arg Y

optional arguments:
  -h, --help  show this help message and exit
  --a A       option a
  --b B       option b

In [8]: parser.parse_args('--a 1 --b 2 X Y'.split())
Out[8]: Namespace(a='1', b='2', x='X', y='Y')

In [9]: parser.parse_args('X Y --a 1 --b 2'.split())
Out[9]: Namespace(a='1', b='2', x='X', y='Y')

In [10]: parser.parse_args('X --a 1 --b 2 Y'.split())
usage: ipython [-h] [--a A] [--b B] x [y]
ipython: error: unrecognized arguments: Y
An exception has occurred, use %tb to see the full traceback.

SystemExit: 2

To exit: use 'exit', 'quit', or Ctrl-D.
To exit: use 'exit', 'quit', or Ctrl-D.
```

Note: Most CLI users don't care about the order of the command-line options. Hence, such fragile behavior should be avoided.

### 2.1.3 Transition to OpenStack Client

This document details the transition roadmap for moving the neutron clients OpenStack Networking API support, both the Python library and the neutron command-line interface (CLI), to the [OpenStack Client \(OSC\)](#) and the [OpenStack Python SDK](#). This transition is being guided by the [Deprecate individual CLIs in favour of OSC OpenStack spec](#). See the [Neutron RFE](#), [OSC neutron support etherpad](#) and details below for the overall progress of this transition.

## Overview

This transition will result in the neutron clients `neutron` CLI being deprecated and then eventually removed. The `neutron` CLI will be replaced by OSC's networking support available via the `openstack` CLI. This is similar to the deprecation and removal process for the `keystone` clients `keystone` CLI. The neutron clients Python library won't be deprecated. It will be available alongside the networking support provided by the OpenStack Python SDK.

Users of the neutron clients command extensions will need to transition to the `OSC plugin system` before the `neutron` CLI is removed. Such users will maintain their OSC plugin commands within their own project and will be responsible for deprecating and removing their `neutron` CLI extension.

## Transition Steps

1. **Done:** OSC adds OpenStack Python SDK as a dependency. See the following patch set: <https://review.opendev.org/#/c/138745/>
2. **Done:** OSC switches its networking support for the `network` command object to use the OpenStack Python SDK instead of the neutron clients Python library. See the following patch set: <https://review.opendev.org/#/c/253348/>
3. **Done:** OSC removes its `python-neutronclient` dependency. See the following patch set: <https://review.opendev.org/#/c/255545/>
4. **In Progress:** OpenStack Python SDK releases version 1.0 to guarantee backwards compatibility of its networking support and OSC updates its dependencies to include OpenStack Python SDK version 1.0 or later. See the following blueprint: <https://blueprints.launchpad.net/python-openstackclient/+spec/network-command-sdk-support>
5. **Done:** OSC switches its networking support for the `ip floating`, `ip floating pool`, `ip fixed`, `security group`, and `security group rule` command objects to use the OpenStack Python SDK instead of the nova clients Python library when neutron is enabled. When nova network is enabled, then the nova clients Python library will continue to be used. See the following OSC bugs:
  - **Done** Floating IP CRUD
  - **Done** Port CRUD
  - **Done** Security Group CRUD
  - **Done** Security Group Rule CRUD
6. **In Progress:** OSC continues enhancing its networking support. At this point and when applicable, enhancements to the `neutron` CLI must also be made to the `openstack` CLI and possibly the OpenStack Python SDK. Users of the neutron clients command extensions should start their transition to the OSC plugin system. See the developer guide section below for more information on this step.
7. **In Progress:** Deprecate the `neutron` CLI. Running the CLI after it has been `deprecated` will issue a warning message: `neutron CLI is deprecated and will be removed in the future. Use openstack CLI instead.` In addition, no new features will be added to the CLI, though fixes to the CLI will be assessed on a case by case basis.
8. **Not Started:** Remove the `neutron` CLI after two deprecation cycles once the criteria below have been met.

- The networking support provide by the `openstack` CLI is functionally equivalent to the `neutron` CLI and it contains sufficient functional and unit test coverage.
- [Neutron Stadium](#) projects, Neutron documentation and [DevStack](#) use `openstack` CLI instead of `neutron` CLI.
- Most users of the neutron clients command extensions have transitioned to the OSC plugin system and use the `openstack` CLI instead of the `neutron` CLI.

## Developer Guide

The `neutron` CLI version 6.x, without extensions, supports over 200 commands while the `openstack` CLI version 3.3.0 supports over 70 networking commands. Of the 70 commands, some do not have all of the options or arguments of their `neutron` CLI equivalent. With this large functional gap, a few critical questions for developers during this transition are Which CLI do I change?, Where does my CLI belong?, and Which Python library do I change? The answer depends on the state of a command and the state of the overall transition. Details are outlined in the tables below. Early stages of the transition will require dual maintenance.

### Which CLI do I change?

neutron Command	openstack Command	CLI to Change
Exists	Doesnt Exist	<code>neutron</code>
Exists	In Progress	<code>neutron</code> and <code>openstack</code> (update related blueprint or bug)
Exists	Exists	<code>openstack</code> (assumes command parity resulting in <code>neutron</code> being deprecated)
Doesnt Exist	Doesnt Exist	<code>openstack</code>

### Where does my CLI belong?

If you are developing an API in any of the [neutron repos](#) the client-side support must be generally located in either the `openstackclient` or `neutronclient` repos. Whether the actual code goes into one or the other repo it depends on the nature of the feature, its maturity level, and/or the depth of feedback required during the development.

The table below provides an idea of what goes where. Generally speaking, we consider Core anything that is L2 and L3 related or that it has been located in the `neutron` repo for quite sometime, e.g. QoS or Metering, or that it is available in each `neutron` deployment irrespective of its configuration (e.g. auto-allocated-topology). Any client feature that falls into this categorization will need to be contributed in OSC. Any other that does not, will need to go into `neutronclient`, assuming that its server-side is located in a `neutron` controlled repo. This is a general guideline, when in doubt, please reach out to a member of the `neutron` core team for clarifications.

Networking Commands	OSC Plugin	OpenStack Project for <code>openstack</code> Commands
Core	No	<code>python-openstackclient</code>
Extension (i.e. <code>neutron stadium</code> )	Yes	<code>python-neutronclient</code> ( <code>neutronclient/osc/v2/&lt;extension&gt;</code> )
Other	Yes	Applicable project owning networking resource

When a repo stops being under neutron governance, its client-side counterpart will have to go through deprecation. Bear in mind that for grandfathered extensions like FWaaS v1, VPNaaS, and LBaaS v1, this is not required as the neutronclient is already deprecated on its own.

### Which Python library do I change?

OpenStack Project for <code>openstack</code> Commands	Python Library to Change
python-openstackclient	openstacksdk
python-neutronclient	python-neutronclient
Other	Applicable project owning network resource

**Important:** The actual name of the command object and/or action in OSC may differ from those used by neutron in order to follow the OSC command structure and to avoid name conflicts. The *network* prefix must be used to avoid name conflicts if the command object name is highly likely to have an ambiguous meaning. Developers should get new command objects and actions approved by the OSC team before proceeding with the implementation.

The Core group includes network resources that provide core `neutron` project features (e.g. network, subnet, port, etc.) and not advanced features in the `neutron` project (e.g. trunk, etc.) or advanced services in separate projects (FWaaS, LBaaS, VPNaaS, dynamic routing, etc.). The Other group applies projects other than the core `neutron` project. Contact the neutron PTL or core team with questions on network resource classification.

When adding or updating an `openstack` networking command to `python-openstackclient`, changes may first be required to the OpenStack Python SDK to support the underlying networking resource object, properties and/or actions. Once the OpenStack Python SDK changes are merged, the related OSC changes can be merged. The OSC changes may require an update to the OSC `openstacksdk` version in the `requirements.txt` file.

When adding an `openstack` networking command to `python-openstackclient`, you can optionally propose an `OSC command spec` which documents the new command interface before proceeding with the implementation.

Users of the `neutron` clients command extensions must adopt the `OSC plugin` system for this transition. Such users will maintain their OSC plugin within their own project and should follow the guidance in the table above to determine which command to change.

## Developer References

- See `OSC neutron support etherpad` to determine if an `openstack` command is in progress.
- See `OSC command list` to determine if an `openstack` command exists.
- See `OSC command spec list` to determine if an `openstack` command spec exists.
- See `OSC plugin command list` to determine if an `openstack` plugin command exists.
- See `OSC command structure` to determine the current `openstack` command objects, plugin objects and actions.
- See `OSC human interface guide` for guidance on creating new OSC command interfaces.
- See `OSC plugin` for information on the OSC plugin system to be used for `neutron` CLI extensions.
- Create an OSC blueprint: <https://blueprints.launchpad.net/python-openstackclient/>

- Report an OSC bug: <https://bugs.launchpad.net/python-openstackclient/+filebug>
- Report an OpenStack Python SDK bug: <https://bugs.launchpad.net/python-openstacksdk/+filebug>

---

**Note:** neutron CLI has been deprecated from Ocata release. We do not add, change and drop any existing commands any more. We only accept changes on OSC plugin, neutronclient python bindings and bug fixes on the deprecated CLI (`neutron` command).

---

---

**CHAPTER  
THREE**

---

**HISTORY**

Release notes is available at <http://docs.openstack.org/releasenotes/python-neutronclient/>.