python-neutronclient Documentation

Release 11.5.0.dev4

Neutron Contributors

Mar 06, 2025

CONTENTS

This is a client for OpenStack Networking API. It provides *Python API bindings* (the neutronclient module).

There is OpenStack Client (OSC). CLI which support the Networking API.

CHAPTER

ONE

USER DOCUMENTATION

1.1 Using CLI

There is OpenStackClient (OSC) which support the Networking API

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>]
    [--fit-width]
    [--print-empty]
    --local-as <local-as>
    [--ip-version {4,6}]
    [--advertise-floating-ip-host-routes]
```

(continues on next page)

(continued from previous page)

```
    [--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]
```

(continues on next page)

(continued from previous page)

```
    [--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 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>]
    [--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

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]
```

(continues on next page)

(continued from previous page)

```
    [--print-empty]
    --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]
```

(continues on next page)

(continued from previous page)

```
    [--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 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]
    [--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>]
    [--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]
```

(continues on next page)

(continued from previous page)

```
    [--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]
```

(continues on next page)

(continued from previous page)

```
    [--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>]
```

(continues on next page)

(continued from previous page)

```
    [--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

--disable

Disable this log (default is enabled)

--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

--disable
Disable this log (default is enabled)

--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.

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>]
    [--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 {l2,l3}]
```

-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 advertisementstowards this BGPVPN.

--type <TYPE>

BGP VPN type selection between IP VPN (l3) and Ethernet VPN (l2) (default: l3)

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
```

(continues on next page)

(continued from previous page)

```
    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]
```

(continues on next page)

(continued from previous page)

```
    <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]
    [--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]
```

(continues on next page)

(continued from previous page)

```
    [--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]
```

(continues on next page)

(continued from previous page)

```
    [--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>]
```

(continues on next page)

(continued from previous page)

```
    [--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
Flow classifier(s) 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]
```

(continues on next page)

(continued from previous page)

```
    [--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>]
    [--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=(mpls|nsh) (default is mpls) and symmetric=(true|false).

--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

Port chain(s) 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>]
    [--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|mpls|nsh) 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

Port pair(s) 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

Port pair group(s) 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]
```

(continues on next page)

(continued from previous page)

```
    [--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> ...]
```

service-graph

ID or name of the service graph(s) 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]
```

(continues on next page)

(continued from previous page)

```
    [--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,aes-xcbc,aes-cmac}]
    [--encryption-algorithm {3des,aes-128,aes-192,aes-256,aes-128-ccm-8,aes-
↪192-ccm-8,aes-256-ccm-8,aes-128-ccm-12,aes-192-ccm-12,aes-256-ccm-12,aes-
↪128-ccm-16,aes-192-ccm-16,aes-256-ccm-16,aes-128-gcm-8,aes-192-gcm-8,aes-
↪256-gcm-8,aes-128-gcm-12,aes-192-gcm-12,aes-256-gcm-12,aes-128-gcm-16,aes-
↪192-gcm-16,aes-256-gcm-16,aes-128-ctr,aes-192-ctr,aes-256-ctr}]
    [--phase1-negotiation-mode {main,aggressive}]
    [--ike-version {v1,v2}]
    [--pfs {group2,group5,group14,group15,group16,group17,group18,group19,
↪group20,group21,group22,group23,group24,group25,group26,group27,group28,
↪group29,group30,group31}]
    [--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,aes-xcbc,aes-cmac}]
    [--encryption-algorithm {3des,aes-128,aes-192,aes-256,aes-128-ccm-8,aes-
↪192-ccm-8,aes-256-ccm-8,aes-128-ccm-12,aes-192-ccm-12,aes-256-ccm-12,aes-
↪128-ccm-16,aes-192-ccm-16,aes-256-ccm-16,aes-128-gcm-8,aes-192-gcm-8,aes-
↪256-gcm-8,aes-128-gcm-12,aes-192-gcm-12,aes-256-gcm-12,aes-128-gcm-16,aes-
↪192-gcm-16,aes-256-gcm-16,aes-128-ctr,aes-192-ctr,aes-256-ctr}]
    [--phase1-negotiation-mode {main,aggressive}]
    [--ike-version {v1,v2}]
    [--pfs {group2,group5,group14,group15,group16,group17,group18,group19,
↪group20,group21,group22,group23,group24,group25,group26,group27,group28,
↪group29,group30,group31}]
    [--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,aes-xcbc,aes-cmac}]
    [--encapsulation-mode {tunnel,transport}]
    [--encryption-algorithm {3des,aes-128,aes-192,aes-256,aes-128-ccm-8,aes-
↪192-ccm-8,aes-256-ccm-8,aes-128-ccm-12,aes-192-ccm-12,aes-256-ccm-12,aes-
↪128-ccm-16,aes-192-ccm-16,aes-256-ccm-16,aes-128-gcm-8,aes-192-gcm-8,aes-
↪256-gcm-8,aes-128-gcm-12,aes-192-gcm-12,aes-256-gcm-12,aes-128-gcm-16,aes-
↪192-gcm-16,aes-256-gcm-16,aes-128-ctr,aes-192-ctr,aes-256-ctr}]
    [--lifetime units=UNITS,value=VALUE]
    [--pfs {group2,group5,group14,group15,group16,group17,group18,group19,
↪group20,group21,group22,group23,group24,group25,group26,group27,group28,
↪group29,group30,group31}]
    [--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,aes-xcbc,aes-cmac}]
    [--encapsulation-mode {tunnel,transport}]
    [--encryption-algorithm {3des,aes-128,aes-192,aes-256,aes-128-ccm-8,aes-
↪192-ccm-8,aes-256-ccm-8,aes-128-ccm-12,aes-192-ccm-12,aes-256-ccm-12,aes-
↪128-ccm-16,aes-192-ccm-16,aes-256-ccm-16,aes-128-gcm-8,aes-192-gcm-8,aes-
↪256-gcm-8,aes-128-gcm-12,aes-192-gcm-12,aes-256-gcm-12,aes-128-gcm-16,aes-
↪192-gcm-16,aes-256-gcm-16,aes-128-ctr,aes-192-ctr,aes-256-ctr}]
    [--lifetime units=UNITS,value=VALUE]
    [--pfs {group2,group5,group14,group15,group16,group17,group18,group19,
↪group20,group21,group22,group23,group24,group25,group26,group27,group28,
```

(continues on next page)

(continued from previous page)

```
↪group29,group30,group31}]
    [--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
```

(continues on next page)

(continued from previous page)

```
    --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

Ipsec 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

--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

Ipsec 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]
    [--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 removed. 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.

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']
```

CHAPTER

TWO

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 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 OSCs 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 wont be deprecated. It will be available along side 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. **Done** 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. **Done** 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 Z cycle. 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. **Done** 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 tool is now removed and all new CLI changes should be done in the `OpenStackClient (OSC)` and, if needed, also in the `OpenStack SDK`.

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 `openstack` Commands
Core	No	python-openstackclient
Extension (i.e. neutron stadium)	Yes	python-neutronclient (`neutronclient/osc/v2/<extension>`)
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 `openstack` Commands	Python Library to Change
python-openstackclient	openstacksdk
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/.