# network commands

## ONTAP commands

NetApp
February 03, 2026

This PDF was generated from https://docs.netapp.com/us-en/ontap-cli-991/network-ping.html on February 03, 2026. Always check docs.netapp.com for the latest.

# Table of Contents


network commands . . . 1
- network ping . . . 1
  - Description . . . 1
  - Parameters . . . 1
  - Examples . . . 2
- network ping6 . . . 2
  - Description . . . 2
  - Parameters . . . 3
  - Examples . . . 4
- network test-path . . . 4
  - Description . . . 4
  - Parameters . . . 4
  - Examples . . . 5
- network traceroute . . . 5
  - Description . . . 5
  - Parameters . . . 5
  - Examples . . . 6
- network traceroute6 . . . 7
  - Description . . . 7
  - Parameters . . . 7
  - Examples . . . 8
- network arp commands . . . 8
  - network arp create . . . 8
  - network arp delete . . . 9
  - network arp show . . . 10
  - network arp active-entry delete . . . 10
  - network arp active-entry show . . . 11
- network bgp commands . . . 12
  - network bgp config create . . . 13
  - network bgp config delete . . . 13
  - network bgp config modify . . . 14
  - network bgp config show . . . 14
  - network bgp defaults modify . . . 15
  - network bgp defaults show . . . 16
  - network bgp peer-group create . . . 16
  - network bgp peer-group delete . . . 18
  - network bgp peer-group modify . . . 18
  - network bgp peer-group rename . . . 19
  - network bgp peer-group show . . . 19
  - network bgp vserver-status show . . . 21
- network cloud commands . . . 22
  - network cloud routing-table create . . . 22
  - network cloud routing-table delete . . . 22

network cloud routing-table show . . . 23
network connections commands . . . 24
network connections active show-clients . . . 24
network connections active show-lifs . . . 25
network connections active show-protocols . . . 26
network connections active show-services . . . 27
network connections active show . . . 29
network connections listening show . . . 31
network device-discovery commands . . . 33
network device-discovery show . . . 33
network fcp commands . . . 36
network fcp adapter modify . . . 36
network fcp adapter show . . . 38
network fcp topology show . . . 43
network fcp zone show . . . 45
network interface commands . . . 47
network interface create . . . 47
network interface delete . . . 52
network interface migrate-all . . . 53
network interface migrate . . . 54
network interface modify . . . 55
network interface rename . . . 58
network interface revert . . . 59
network interface show . . . 59
network interface start-cluster-check . . . 65
network interface capacity show . . . 66
network interface capacity details show . . . 67
network interface check cluster-connectivity show . . . 68
network interface check cluster-connectivity start . . . 69
network interface check failover show . . . 70
network interface dns-lb-stats show . . . 73
network interface failover-groups add-targets . . . 74
network interface failover-groups create . . . 75
network interface failover-groups delete . . . 76
network interface failover-groups modify . . . 76
network interface failover-groups remove-targets . . . 77
network interface failover-groups rename . . . 77
network interface failover-groups show . . . 78
network interface lif-weights show . . . 79
network interface service show . . . 80
network interface service-policy add-service . . . 81
network interface service-policy clone . . . 83
network interface service-policy create . . . 85
network interface service-policy delete . . . 87
network interface service-policy modify-service . . . 88

network interface service-policy remove-service . . . 90
network interface service-policy rename . . . 92
network interface service-policy restore-defaults . . . 93
network interface service-policy show . . . 95
network ipspace commands . . . 96
network ipspace create . . . 96
network ipspace delete . . . 97
network ipspace rename . . . 98
network ipspace show . . . 98
network ndp commands . . . 99
network ndp default-router delete-all . . . 99
network ndp default-router show . . . 100
network ndp neighbor create . . . 101
network ndp neighbor delete . . . 102
network ndp neighbor show . . . 102
network ndp neighbor active-entry delete . . . 103
network ndp neighbor active-entry show . . . 104
network ndp prefix delete-all . . . 105
network ndp prefix show . . . 106
network options commands . . . 107
network options cluster-health-notifications modify . . . 107
network options cluster-health-notifications show . . . 108
network options detect-switchless-cluster modify . . . 109
network options detect-switchless-cluster show . . . 109
network options ipv6 modify . . . 109
network options ipv6 show . . . 110
network options load-balancing modify . . . 111
network options load-balancing show . . . 111
network options multipath-routing modify . . . 111
network options multipath-routing show . . . 112
network options port-health-monitor disable-monitors . . . 112
network options port-health-monitor enable-monitors . . . 113
network options port-health-monitor modify . . . 114
network options port-health-monitor show . . . 115
network options send-soa modify . . . 116
network options send-soa show . . . 117
network options switchless-cluster modify . . . 117
network options switchless-cluster show . . . 118
network port commands . . . 118
network port delete . . . 118
network port modify . . . 119
network port show-address-filter-info . . . 120
network port show . . . 121
network port broadcast-domain add-ports . . . 125
network port broadcast-domain create . . . 126

network port broadcast-domain delete 127
network port broadcast-domain merge 127
network port broadcast-domain modify 128
network port broadcast-domain move 128
network port broadcast-domain remove-ports 129
network port broadcast-domain rename 129
network port broadcast-domain show 130
network port broadcast-domain split 131
network port ifgrp add-port 132
network port ifgrp create 134
network port ifgrp delete 135
network port ifgrp remove-port 135
network port ifgrp show 136
network port reachability repair 137
network port reachability scan 138
network port reachability show 138
network port vip create 141
network port vip delete 141
network port vip show 142
network port vlan create 143
network port vlan delete 143
network port vlan show 144
network qos-marking commands 145
network qos-marking modify 145
network qos-marking show 146
network route commands 147
network route create 147
network route delete 148
network route show-lifs 148
network route show 149
network route active-entry show 150
network subnet commands 153
network subnet add-ranges 153
network subnet create 154
network subnet delete 155
network subnet modify 155
network subnet remove-ranges 156
network subnet rename 157
network subnet show 158
network test-link commands 159
network test-link run-test 159
network test-link show 160
network test-link start-server 162
network test-link stop-server 163
network trace commands 163

network trace show . . . 163
network trace start . . . 164
network trace stop . . . 165
network trace file delete . . . 166
network trace file show . . . 166
network tuning commands . . . 167
network tuning icmp modify . . . 167
network tuning icmp show . . . 168
network tuning icmp6 modify . . . 168
network tuning icmp6 show . . . 169
network tuning tcp modify . . . 170
network tuning tcp show . . . 171

# network commands

## network ping

Ping

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

### Description

The `network ping` command displays whether a remote address is reachable and responsive, the (if specified) number of transmitted and received packets, and their round-trip time. The command requires a source node or logical interface from where the ping will be run, and a destination IP address. You can specify the source node by name, or a logical interface and its Vserver.

### Parameters

**{ -node <nodename> - Node**

Use this parameter to send the ping from the node you specify.

**| -lif <lif-name> - Logical Interface }**

Use this parameter to send the ping from the logical interface you specify.

**-vserver <vserver> - Vserver**

Use this parameter to send the ping from the Vserver where the intended logical interface resides. The default value is the system Vserver for cluster administrators.

**[-use-source-port {true|false}] - (DEPRECATED)-Use Source Port of Logical Interface (privilege: advanced)**

This parameter is only applicable when the `-lif` parameter is specified. When set to true, the ping packet will be sent out via the port which is currently hosting the IP address of the logical interface. Otherwise, the ping packet will be sent out via a port based on the routing table.

The `use-source-port` parameter is deprecated and may be removed in a future release of Data ONTAP.

**-destination <Remote InetAddress> - Destination**

Use this parameter to specify the remote internet address destination of the ping.

**[-s, -show-detail <true>] - Show Detail Output**

Use this parameter to display detailed output about the ping.

**[-R, -record-route <true>] - Record Route**

Use this parameter to display the route followed by the ping. You should set this option to `false` for pinging to succeed.

**[-v, -verbose <true>] - Show All ICMP Packets**

Use this parameter to display all ICMP packets.

**[-packet-size <integer>] - Packet Size**

Use this parameter to specify the number of data bytes to be sent in the ping packet. The default is 56 bytes, which is 64 ICMP data bytes total after 8 bytes of ICMP header data is added.

**[-count <integer>] - Count**

Use this parameter to specify the maximum number of ECHO_REQUESTS to be sent to the destination. The default is 20 requests. In the absence of the 'show-detail' option, ping reports that the destination is alive after receiving the first ECHO_REPLY response, independent of this value.

**[-wait <integer>] - Packet Send Wait Time (secs)**

Use this parameter to specify the number of seconds to wait between sending packets. The default is one second.

**[-flood <true>] - Flood Ping (privilege: advanced)**

Use this parameter to execute the command in flood mode. In flood mode, the command issues pings as fast as they are received, unless you specify a wait time.

**[-D, -disallow-fragmentation <true>] - Disallow Packet Fragmentation**

Use this parameter to prevent transport mechanisms from fragmenting ping packets in transit. Preventing fragmentation assures consistent packet size, making it easier to see transport bottlenecks.

**[-wait-response <integer>] - Packet Response Wait Time (ms)**

Use this parameter to specify the number of milliseconds to wait for each response packet. The default is 10000 milliseconds (10 seconds).

## Examples

This example shows a ping from node xena to the destination server 10.98.16.164 with the server responding that it is up and running.

```
cluster1::> network ping -node xena -destination 10.98.16.164
(network ping)
10.98.16.164 is alive
```

# network ping6

Ping an IPv6 address

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

## Description

The `network ping6` command uses the ICMPv6 protocol's mandatory ICMP6_ECHO_REQUEST datagram to elicit an ICMP6_ECHO_REPLY from a host or gateway. ICMP6_ECHO_REQUEST datagrams ("pings") have an IPv6 header, and ICMPv6 header formatted as documented in RFC2463.

## Parameters

**{ -node <nodename> - Node Name**

Use this parameter to originate ping6 from the specified node.

**| -lif <lif-name> - Logical Interface }**

Use this parameter to originate ping6 from the specified logical interface.

**-vserver <vserver name> - Vserver Name**

Use this parameter to originate ping6 from the specified Vserver. The default value is the system Vserver for cluster administrators.

**-destination <Remote InetAddress> - Destination**

Use this parameter to specify the IPv6 address of the destination node.

**[-b, -buffer-size <integer>] - Socket Buffer Size**

Use this parameter to set the socket buffer size.

**[-c, -count <integer>] - Max Requests to Send/Recieve**

Use this parameter to specify the maximum number of requests and replies. The default value is 20.

**[-H, -reverse-lookup <true>] - Reverse-lookup of IPv6 addresses**

Use this parameter to specify reverse-lookup of IPv6 addresses. Unless this parameter is specified, ping6 command does not attempt reverse lookup.

**[-i, -interval <integer>] - Wait between Packets (secs)**

Use this parameter to specify the delay time between packets in seconds. The default value is 1 second. This parameter is incompatible with the flood parameter.

**[-l, -preload <integer>] - Send Packets as Fast as Possible (privilege: advanced)**

Use this parameter if preload is required. If specified, ping6 sends that many packets as fast as possibile before falling into its normal mode of behaviour.

**[-use-source-port {true|false}] - Use Source Port of Logical Interface (privilege: advanced)**

This parameter is only applicable when the `-lif` parameter is specified. When set to true, the ping packet will be sent out via the port which is currently hosting the IP address of the logical interface. Otherwise, the ping packet will be sent out via a port based on the routing table.

**[-p, -pattern <text>] - Up to 16 'pad' Specified for Out Packet**

Use this parameter to fill the -16 'pad' bytes in the sent packet. This is useful for diagnosing data dependent problems in a network. For example, -pattern ff causes the sent packet to be filled with all ones.

**[-packet-size <integer>] - Packet Size**

Use this parameter to specify the number of data bytes to be sent. The default is 56, which translates to 64 ICMP data bytes when combined with the 8 bytes of ICMP header data.

**[-v, -verbose <true>] - Show All ICMP Packets**

Use this parameter to get verbose output. Verbose output displays both source address and destination addresses. Received ICMP packets other than ECHO_RESPONSE are listed. This parameter can be used only in conjunction with the show-detail parameter.

**[-s, -show-detail <true>] - Show Detail Output**

Use this parameter to display detailed output about the ping.

**[-f, -flood <true>] - Flood Ping (privilege: advanced)**

Use this parameter to output packets as fast as they come back or one hundred times per second, whichever is more. For every ECHO_REQUEST sent a period "." is printed, while for every ECHO_REPLY received a backspace is printed. This provides a rapid display of how many packets are being dropped. This can be very hard on a network and should be used with caution.

**[-D, -disable-fragmentation <true>] - Disable Fragmentation.**

Use this parameter to disallow fragmentation of the outgoing packets, if they do not fit in the Maximum Transmission Unit.

## Examples

This example shows a ping6 from node 'node1' to the destination server ipv6.google.com with the server responding that it is up and running.

```
cluster1::> network ping6 -node node1 -destination ipv6.google.com
    ipv6.google.com is alive.
```

# network test-path

Test path performance between two nodes

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

## Description

The `network test-path` command runs a performance test between two nodes. The command requires a source node, destination node, destination cluster, and application, or session type. All tests are run using intracluster or intercluster LIFs, depending on whether the test is between two nodes in the same cluster, or between nodes in peered clusters.

The test itself is different from most bandwidth test tools. It creates a "session" consisting of TCP connections between all possible paths between the nodes being tested. This is how internal Data ONTAP applications communicate between nodes. This means the test is using multiple paths, and thus the bandwidth reported might exceed the capacity of a single 10 Gb path.

## Parameters

**-source-node {<nodename>|local} - Node Initiating Session (privilege: advanced)**

Use this parameter to specify the node that initiates the test. Source-node parameter must be a member of the cluster in which the command is run.

**-destination-cluster <Cluster name> - Cluster Containing Passive Node (privilege: advanced)**

Use this parameter to specify the destination cluster; the local cluster, or a peered cluster.

**-destination-node <text> - Remote Node in Destination Cluster (privilege: advanced)**

Use this parameter to specify the destination node in the destination cluster

**-session-type {AsyncMirrorLocal|AsyncMirrorRemote|RemoteDataTransfer} - Type of Session to Test (privilege: advanced)**

The session type parameter is used to mimic the application settings used. A session consists of multiple TCP connections.

- AsyncMirrorLocal: settings used by SnapMirror between nodes in the same cluster
- AsyncMirrorRemote: settings used by SnapMirror between nodes in different clusters
- RemoteDataTransfer: settings used by Data ONTAP for remote data access between nodes in the same cluster

The default session-type is AsyncMirrorRemote.

## Examples

The following example runs a test between two nodes in the same cluster:

```
cluster1::*> network test-path -source-node node1 -destination-cluster
cluster1 -destination-node node2
Test Duration: 10.65 secs
   Send Throughput: 1092.65 MB/sec
Receive Throughput: 1092.65 MB/sec
           MB Sent: 11633.69
       MB Received: 11633.69
       Avg Latency:    64.40 ms
       Min Latency:     2.41 ms
       Max Latency:  2099.17 ms
```

# network traceroute

Traceroute

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

## Description

The `network traceroute` command performs a network probe from a node to a specified IP address. The command requires a source node or logical interface and a destination IP address. You can specify the source node by name, or specify a logical interface and its Vserver. The traceroute is performed between the source and destination.

## Parameters

**{ -node <nodename> - Node**

Use this parameter to originate the traceroute from the node you specify.

**| -lif <lif-name> - Logical Interface }**

Use this parameter to originate the traceroute from the specified network interface.

**-vserver <vserver> - LIF Owner**

Use this parameter to originate the traceroute from the Vserver where the intended logical interface resides. The default value is the system Vserver for cluster administrators.

**-destination <Remote InetAddress> - Destination**

Use this parameter to specify the remote internet address destination of the traceroute.

**[-m, -maxttl <integer>] - Maximum Number of Hops**

Use this parameter to specify the maximum number of hops (time-to-live) setting used by outgoing probe packets. The default is 30 hops.

**[-n, -numeric <true>] - Print Hop Numerically**

Use this parameter to print the hop addresses only numerically rather than symbolically and numerically.

**[-port <integer>] - Base UDP Port Number**

Use this parameter to specify the base UDP port number used in probes. The default is port 33434.

**[-packet-size <integer>] - Packet Size**

Use this parameter to specify the size of probe packets, in bytes.

**[-q, -nqueries <integer>] - Number of Queries**

Use this parameter to specify the number of probes per hop. The default is 3 probes.

**[-v, -verbose <true>] - Verbose Output**

Use this parameter to display all received ICMP packets, rather than just TIME_EXCEEDED and UNREACHABLE packets.

**[-w, -waittime <integer>] - Wait Between Packets (secs)**

Use this parameter to specify the time (in seconds) to wait for the response to a probe. The default is 5 seconds.

## Examples

This example shows a traceroute from node node1 to a destination address of 10.98.16.164, showing a maximum of five hops.

```
cluster1::> traceroute -node node1 -destination 10.98.16.164 -maxttl 5
 1   10.68.208.1 <10.68.208.1> 0.307 ms 293 ms 305 ms
 2   152.164.13.205 <152.164.13.205> 3.754 ms 3.722 ms 3.981 ms
 3   68.137.122.222 <68.137.122.222> 25.603 ms 24.947 ms 24,565 ms
 4   * * *
 5   * * *

traceroute to 10.98.16.164, 5 hops max, 52 byte packets
```

# network traceroute6

traceroute6

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

## Description

The `network traceroute6` command performs a network probe from a node to a specified IPv6 address. The command requires a source node or logical interface, Vserver from where traceroute6 will originate and a destination IPv6 address. traceroute6 is performed between the source and destination.

## Parameters

**{ -node <nodename> - Node**

Use this parameter to originate traceroute6 from the node you specify. This parameter is available only to cluster administrators.

**| -lif <lif-name> - Logical Interface }**

Use this parameter to originate traceroute6 from the logical interface you specify.

**-vserver <vserver name> - LIF Owner**

Use this parameter to originate traceroute6 from the Vserver you specify. The default value is the system Vserver for cluster administrators.

**[-d, -debug-mode <true>] - Debug Mode**

Use this parameter to enable socket level debugging. The default value is false.

**{ [-I, -icmp6 <true>] - ICMP6 ECHO instead of UDP**

Use this parameter to specify the use of ICMP6 ECHO instead of UDP datagrams for the probes. The default value is false.

**| [-U, -udp <true>] - UDP }**

Use this parameter to specify the use of UDP datagrams for the probes. The default value is true.

**[-n, -numeric <true>] - Print Hops Numerically**

Use this parameter to print the hop addresses only numerically rather than symbolically and numerically. The default value is false.

**[-v, -verbose <true>] - Verbose Output**

Use this parameter to display all received ICMP packets, rather than just TIME_EXCEEDED and UNREACHABLE packets. The default value is false.

**[-f, -first-hop <integer>] - Number of Hops to Skip in Trace**

Use this parameter to specify the number of hops to skip in trace. The default value is 1.

**[-g, -gateway <Remote InetAddress>] - Intermediate Gateway**

Use this parameter to specify the intermediate gateway.

**[-m, -hop-limit <integer>] - Maximum Number of Hops**

Use this parameter to specify the maximum hoplimit, upto 255. The default value is 64 hops.

**[-p, -port <integer>] - Base UDP Port Number**

Use this parameter to specify the base UDP port number used in probes. The default value is port 33434.

**[-q, -nqueries <integer>] - Number of Queries**

Use this parameter to specify the number of probes per hop. The default value is 3 probes.

**[-w, -wait-time <integer>] - Wait Between Packets (secs)**

Use this parameter to specify the delay time between probes in seconds. The default value is 5 seconds.

**-destination <Remote InetAddress> - Destination**

Use this parameter to specify the remote IPv6 address destination of traceroute6.

**[-packet-size <integer>] - Packet Size**

Use this parameter to specify the size of probe packets, in bytes. The default value is 16 bytes for ICMP6 ECHO and 12 bytes for UDP datagrams.

## Examples

The following example shows traceroute6 from node node1 to the destination fd20:8b1e:b255:4071:d255:1fcd:a8cd:b9e8.

```
cluster1::> network traceroute6 -node node1 -vserver vs1
                -destination 3ffe:b00:c18:1::10
traceroute6 to 3ffe:b00:c18:1::10 (3ffe:b00:c18:1::10)
              from 2001:0db8:0000:f101::2,
           64 hops max, 12 byte packets
1  2001:0db8:0000:f101::1 4.249 ms  2.021 ms  0.864 ms
2  3ffe:2000:0:400::1  0.831 ms  0.579 ms
3  3ffe:2000:0:1::132  227.693 ms  227.596 ms  227.439 ms
4  3ffe:c00:8023:2b::2 229.028 ms  228.267 ms  231.891 ms
5  3ffe:2e00:e:c::3  227.929 ms  228.696 ms  228.558 ms
6  3ffe:b00:c18:1::10  227.702 ms 227.806 ms  227.439 ms
```

# network arp commands

## network arp create

Create static ARP entry

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

### Description

The `network arp create` command creates a static ARP entry for a given Vserver. Statically created ARP

entries will be stored permanently in the Vserver context and will be used by the network stack.

#### Parameters

**-vserver <vserver name> - Vserver Name**

Use this parameter to specify the name of the Vserver on which the ARP entry is created.

**-remotehost <IP Address> - Remote IP Address**

Use this parameter to specify the IP address to be added as an ARP entry.

**-mac <MAC Address> - MAC Address**

Use this parameter to specify the MAC address (Ethernet address) for the host specified with -remotehost. Specify the MAC address as six hex bytes separated by colons.

#### Examples

The following example creates a static ARP entry on Vserver vs1 for the remote host with the IP address 10.63.0.2 having MAC address 40:55:39:25:27:c1

```
cluster1::> network arp create -vserver vs1 -remotehost 10.63.0.2 -mac
40:55:39:25:27:c1
```

## network arp delete

Delete static ARP entry

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

#### Description

The `network arp delete` command deletes static ARP entries from the Vserver and from the network stack.

#### Parameters

**-vserver <vserver name> - Vserver Name**

Use this parameter to specify the name of the Vserver from which the ARP entry is deleted.

**-remotehost <IP Address> - Remote IP Address**

Use this parameter to specify the IP address of the ARP entry being deleted.

#### Examples

The following example deletes the ARP entry for IP address 10.63.0.2 from the Vserver vs1.

```
cluster1::> network arp delete -vserver vs1 -remotehost 10.63.0.2
```

## network arp show

Display static ARP entries

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

### Description

The `network arp show` command displays static ARP entries present in a given Vserver. This command will not display dynamically learnt ARP entries in the network stack. Use the network arp active-entry show command to display dynamically learned ARP entries in the network stack.

### Parameters

**{ [-fields <fieldname>,…]**
Use this parameter to display only certain fields of the ARP table.

**| [-instance ] }**
Use this parameter to display all the fields of the ARP table.

**[-vserver <vserver name>] - Vserver Name**
Use this parameter to display ARP entries that are specific to a given Vserver.

**[-remotehost <IP Address>] - Remote IP Address**
Use this parameter to display ARP entries for the specified IP address

**[-mac <MAC Address>] - MAC Address**
Use this parameter to display ARP entry for the specified MAC address

**[-ipspace <IPspace>] - IPspace**
Use this parameter to specify the IPspace associated with the Vserver

### Examples

The following example displays static ARP entries from the Vserver vs1.

```
cluster1::> network arp show -vserver vs1
Vserver     Remote Host       MAC Address
----------- ----------------- -----------------
vs1
            10.238.0.2        40:55:39:25:27:c1
```

### Related Links

- network arp active-entry show

## network arp active-entry delete

Delete active ARP entry from a System or Admin Vserver

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

### Description

The `network arp active-entry delete` command deletes dynamically learned ARP entries from the network stack of a node. To delete statically configured ARP entries use the network arp delete command.

### Parameters

**`-node {<nodename>|local}` - Node (privilege: advanced)**

Use this parameter to specify the name of the node in which the ARP entry is deleted.

**`-vserver <vserver>` - System or Admin Vserver Name (privilege: advanced)**

Use this parameter to specify the name of the Vserver in which the ARP entry is deleted. Only Vservers with a type of Admin or System have dynamically learned ARP entries.

**`-subnet-group <IP Address/Mask>` - Subnet Group Name (privilege: advanced)**

Use this parameter to specify the name of the routing group in which the ARP entry is deleted.

**`-remotehost <text>` - Remote IP Address (privilege: advanced)**

Use this parameter to specify the IP address to be deleted from the active ARP entries.

**`-port <text>` - Port (privilege: advanced)**

Use this parameter to specify the name of the Port to be deleted from the active ARP entries.

### Examples

The following example deletes the active ARP entry with an IP address of 10.224.64.1, subnet group of 0.0.0.0/0, port e0c on node node2 in the Admin Vserver cluster1:

```
cluster1::network arp active-entry*> delete -node cluster1-01 -vserver
cluster1 -subnet-group 0.0.0.0/0 -remotehost 10.224.64.1 -port e0c
```

### Related Links

- network arp delete

## network arp active-entry show

Display active ARP entries organized by Vserver

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

### Description

The `network arp active-entry show ` command displays ARP entries present in the network stack of the node. The entries includes both dynamically learned ARP entries and user configured static ARP entries.

### Parameters

**{ [-fields <fieldname>,…]**

Use this parameter to display only certain fields of the active ARP table.

**| [-instance ] }**

Use this parameter to display all the fields of the active ARP table.

**[-node {<nodename>|local}] - Node (privilege: advanced)**

Use this parameter to display active ARP entries that are specific to a given node.

**[-vserver <vserver>] - System or Admin Vserver Name (privilege: advanced)**

Use this parameter to display active ARP entries that are specific to a given System or Admin Vserver. Data and Node Vservers will not have any active-arp entries.

**[-subnet-group <IP Address/Mask>] - Subnet Group Name (privilege: advanced)**

Use this parameter to display active ARP entries that are specific to a given subnet group.

**[-remotehost <text>] - Remote IP Address (privilege: advanced)**

Use this parameter to display active ARP entries for the specified IP address.

**[-port <text>] - Port (privilege: advanced)**

Use this parameter to display active ARP entries for the specified Port name.

**[-mac <text>] - MAC Address (privilege: advanced)**

Use this parameter to display the active ARP entry for the specified MAC address.

**[-ipspace <IPspace>] - IPspace (privilege: advanced)**

Use this parameter to specify the IPspace associated with the System or Admin Vserver.

### Examples

The following example displays active ARP entries for the Admin Vserver cluster1:

```
cluster1::*> network arp active-entry show -vserver cluster1

Node: node-01
Vserver: cluster1
Subnet Group: 169.254.0.0/16
Remote IP Address MAC Address       Port
----------------- ----------------- -------
169.254.106.95    0:55:39:27:d1:c1  lo
```

# network bgp commands

## network bgp config create

### Create BGP configuration

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

#### Description

The `network bgp config create` command is used to create the border gateway protocol (BGP) configuration for a node. It can be used to override the BGP parameters defined in the global BGP defaults.

#### Parameters

**`-node {<nodename>|local}` - Node**

This parameter specifies the node on which configuration details will be created.

**`-asn <integer>` - Autonomous System Number**

This parameter specifies the autonomous system number (ASN). The ASN attribute is a positive integer of the range from 1 to 4,294,967,295. It should typically be chosen from RFC6996 "Autonomous System (AS) Reservation for Private Use" or the AS number assigned to the operator's organization.

**`-hold-time <integer>` - Hold Time**

This parameter specifies the hold time in seconds. The default value is 180.

**`-router-id <IP Address>` - Router ID**

This parameter specifies the local router ID. The router-id value takes the form of an IPv4 address. The default router-id will be initialized using a local IPv4 address in admin vserver.

#### Examples

```
cluster1::> network bgp config create -node node1 -asn 10 -hold-time 180
-router-id 10.0.1.112
```

## network bgp config delete

### Delete BGP configuration

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

#### Description

The `network bgp config delete` command deletes a node's border gateway protocol (BGP) configuration. A BGP configuation cannot be deleted if there are BGP peer groups configured on the associated node.

#### Parameters

**`-node {<nodename>|local}` - Node**

This parameter specifies the node for which the BGP configuration will be deleted.

#### Examples

```
cluster1::> network bgp config delete -node node1
```

## network bgp config modify

### Modify BGP configuration

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

#### Description

The `network bgp config modify` command is used to modify a node's border gateway protocol (BGP) configuration.

#### Parameters

**`-node {<nodename>|local}` - Node**

This parameter specifies the node on which BGP configuration will be modified.

**`[-asn <integer>]` - Autonomous System Number**

This parameter specifies the autonomous system number (ASN). The ASN attribute is a positive integer of the range from 1 to 4,294,967,295. It should typically be chosen from RFC6996 "Autonomous System (AS) Reservation for Private Use" or the AS number assigned to the operator's organization.

**`[-hold-time <integer>]` - Hold Time**

This parameter specifies the hold time in seconds.

**`[-router-id <IP Address>]` - Router ID**

This parameter specifies the local router ID. The router-id value takes the form of an IPv4 address.

#### Examples

```
cluster1::> network bgp config modify -node node1 -router-id 1.1.1.1 -asn
20
```

## network bgp config show

### Display BGP configuration

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

The `network bgp config show` command displays the border gateway protocol (BGP) configuration for each node.

### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-node {<nodename>|local}] - Node**

This parameter selects the BGP configurations that match the specified node.

**[-asn <integer>] - Autonomous System Number**

This parameter selects the BGP configurations that match the specified autonomous system number.

**[-hold-time <integer>] - Hold Time**

This parameter selects BGP configurations that match the specified hold time.

**[-router-id <IP Address>] - Router ID**

This parameter selects the BGP configurations that match the specfied router ID.

### Examples

```
cluster1::> network bgp config show
        Autonomous
        System    Hold Time
    Node    Number    (seconds)  Router ID
    --------------  ---------------  ---------------  ---------
    node1    10    180    10.0.1.112
```

## network bgp defaults modify

Modify BGP defaults

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

### Description

The `network bgp defaults modify` command modifies the global defaults for border gateway protocol (BGP) configurations.

### Parameters

**[-asn <integer>] - Autonomous System Number**

This parameter specifies the autonomous system number (ASN). The ASN attribute is a positive integer. It should typically be chosen from RFC6996 "Autonomous System (AS) Reservation for Private Use", or the AS number assigned to the operator’s organization. The default ASN is 65501.

**[-hold-time <integer>] - Hold Time**

This parameter specifies the hold time in seconds. The default value is 180.

#### Examples

```
cluster1::> network bgp defaults modify -asn 20
```

## network bgp defaults show

Display BGP defaults

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

The `network bgp defaults show` command displays the global defaults for border gateway protocol (BGP) configurations.

#### Examples

```
cluster1::> network bgp defaults show
Autonomous
System Number  Hold Time
    (Seconds)
-------------  --------------
10    180
```

## network bgp peer-group create

Create a new BGP peer group

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

The `network bgp peer-group create` command is used to create a border gateway protocol (BGP) peer group. A BGP peer group will advertise VIP routes for the list of vservers in the peer group's `vserver-list` using the BGP LIF of the peer group. A BGP peer group will advertise VIP routes to a peer router using the border gateway protocol. The address of the peer router is identfied by the `peer-address` value.

#### Parameters

**-ipspace <IPspace> - IPspace Name**

This parameter specifies the IPspace of the peer group being created.

**-peer-group <text> - Peer Group Name**

This parameter specifies the name of the peer group being created.

**-bgp-lif <lif-name> - BGP LIF**

This parameter specifies the BGP interface (BGP LIF) of the peer group being created.

**-peer-address <IP Address> - Peer Router Address**

This parameter specifies the IP address of the peer router for the peer group being created.

**[-peer-asn <integer>] - Peer Router Autonomous number**

This parameter specifies the peer router autonomous system number (ASN) in the peer group being created. The default value is the value of the local node's ASN.

**-route-preference <integer> - Route Preference**

This parameter specifies the preference field in BGP update messages for VIP routes. If a router receives multiple VIP route announcements for the same VIP LIF from different BGP LIFs, it will install the one that has the highest preference value. The default route preference value is 100.

**[-asn-prepend-type <ASN Prepend type>] - ASN prepend type (privilege: advanced)**

This parameter specifies the ASN that will be prepended in the BGP attributes. The possible values are `local-asn` and `peer-asn` . The default behaviour is not to prepend any ASN.

**[-asn-prepend-count <integer>] - ASN prepend count (privilege: advanced)**

This parameter specifies the number of times ASN, as specified in `asn-prepend-type` will be prepended in the BGP path attributes. The default behaviour is not to prepend any ASN.

**[-community <BGP community>,…] - BGP Community (privilege: advanced)**

This parameter specifies the communities that will be included in the BGP path attributes. The default behaviour is not to include any community in BGP path attributes.

**[-med <integer>] - Multi Exit Discriminator (privilege: advanced)**

This parameter specifies the Multi Exit Discriminator (MED) attribute of BGP update messages, which can be used by routers for best path selection, in cases where more than one peer advertises the same route with similar attributes.

**[-use-peer-as-next-hop {true|false}] - Use Peer Address As Next Hop**

This parameter specifies whether the peer group uses the peer address as a next hop route. When the value is true, the peer address is used as the next hop router for packets sent from VIP LIFs via the port on which `bgp-lif` is configured. Internally, a default route with a gateway configured as the `peer-address` is added automatically on the node for all the Vservers in this peer group's IPspace. The route will be added for a Vserver only if it has a VIP LIF hosted on the current node of `bgp-lif` . Note that these automatically installed default routes are for VIP traffic; however, they can be used for non-VIP traffic as well if a Vserver hosts both VIP and non-VIP LIFs in the same subnet as `bgp-lif` . This route will have metric of 20 and will be used to forward traffic through the current port of `bgp-lif` . The default value of this parameter is false.

## Examples

```
cluster1::> network bgp peer-group create -peer-group group1 -ipspace
Default -bgp-lif bgp_lif -peer-address 10.0.1.112
```

## network bgp peer-group delete

### Delete a BGP peer group

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

The `network bgp peer-group delete` command is used to delete border gateway protocol (BGP) peer group configuration.

#### Parameters

**`-ipspace <IPspace>` - IPspace Name**

This parameter specifies the IPspace of the BGP peer group being deleted.

**`-peer-group <text>` - Peer Group Name**

This parameter specifies the name of the BGP peer group being deleted.

#### Examples

```
cluster1::> network bgp peer-group delete -ipspace Default -peer-group
group1
```

## network bgp peer-group modify

### Modify a BGP peer group

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

The `network bgp peer-group modify` command is used to modify a border gateway protocol (BGP) peer group configuration.

#### Parameters

**`-ipspace <IPspace>` - IPspace Name**

This parameter specifies the IPspace of the peer group being modified.

**`-peer-group <text>` - Peer Group Name**

This parameter specifies the name of the peer group being modified.

**`[-peer-address <IP Address>]` - Peer Router Address**

This parameter specifies an updated value for the IP address of the peer router.

**`[-use-peer-as-next-hop {true|false}]` - Use Peer Address As Next Hop**

This parameter specifies whether the peer group uses the peer address as a next hop route. When the value is true, the peer address is used as the next hop router for packets sent from VIP LIFs via the port on

which `bgp-lif` is configured. Internally, a default route with a gateway configured as the `peer-address` is added automatically on the node for all the Vservers in this peer group's IPspace. The route will be added for a Vserver only if it has a VIP LIF hosted on the current node of `bgp-lif` . Note that these automatically installed default routes are for VIP traffic; however, they can be used for non-VIP traffic as well if a Vserver hosts both VIP and non-VIP LIFs in the same subnet as `bgp-lif` . This route will have metric of 20 and will be used to forward traffic through the current port of `bgp-lif` . The default value of this parameter is false.

#### Examples

```
cluster1::> network bgp peer-group modify -ipspace Default -peer-group
peer1 -peer-address 10.10.10.10
```

## network bgp peer-group rename

Rename a BGP peer group

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

#### Description

The `network bgp peer-group rename` command is used to assign a new name to a BGP peer group.

#### Parameters

**`-ipspace <IPspace>` - IPspace Name (privilege: advanced)**

This parameter specifies the IPspace of the peer group being renamed.

**`-peer-group <text>` - Peer Group Name (privilege: advanced)**

The name of the peer group to be updated.

**`-new-name <text>` - New Name (privilege: advanced)**

The new name for the peer group.

#### Examples

```
cluster1::> network bgp peer-group rename -peer-group old_name -new-name
new_name
```

## network bgp peer-group show

Display BGP peer groups information

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

The `network bgp peer-group show` command displays the BGP peer groups configuration.

### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-ipspace <IPspace>] - IPspace Name**

This parameter selects peer groups that match the specified IPspace.

**[-peer-group <text>] - Peer Group Name**

This parameter selects peer groups that match the specified name.

**[-bgp-lif <lif-name>] - BGP LIF**

This parameter selects peer groups that match the specified BGP Interface.

**[-peer-address <IP Address>] - Peer Router Address**

This parameter selects peer groups that match the specified peer router address.

**[-peer-asn <integer>] - Peer Router Autonomous number**

This parameter selects peer groups that match the specified autonomous system number.

**[-state <BGP Session State>] - Peer Group State**

This parameter selects peer groups that match the specified BGP session state.

**[-bgp-node <nodename>] - BGP LIF Node**

This parameter selects peer groups that match the specified `bgp-node` value. This value is calculated based on the current node of the corresponding BGP LIF.

**[-bgp-port <netport>] - BGP LIF Port**

This parameter selects peer groups that match the specified `bgp-port` value. This value is calculated based on the current port of the associated BGP LIF.

**[-route-preference <integer>] - Route Preference**

This parameter selects peer groups that match the specified route preference value.

**[-asn-prepend-type <ASN Prepend type>] - ASN prepend type (privilege: advanced)**

This parameter selects peer groups that match the specified `asn-prepend-type` value. The possible values are `local-asn` and `peer-asn` .

**[-asn-prepend-count <integer>] - ASN prepend count (privilege: advanced)**

This parameter selects peer groups that match the specified `asn-prepend-count` value.

**[-community <BGP community>,…] - BGP Community (privilege: advanced)**

This parameter selects peer groups that match the specified `community` value.

**[-med <integer>] - Multi Exit Discriminator (privilege: advanced)**

This parameter selects peer groups that match the specified med value.

**[-use-peer-as-next-hop {true|false}] - Use Peer Address As Next Hop**

This parameter selects peer groups that match the specified use-peer-as-next-hop value.

### Examples

```
cluster1::> network bgp peer-group show
    IPspace: Default
    Peer        Local BGP Peer router                      Autonomous
    Group       Interface Address/subnet    state          Number     Node
Port
    ----------  --------- ----------------- -------------- ----------
------ -----
    gp1         bgp_lif1  10.0.5.37         up                    10
node1  e1a
    gp2         bgp_lif2  10.0.6.38         up                    12
node1  e2a
```

## network bgp vserver-status show

Display Vserver BGP status

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

The `network bgp vserver-status show` command displays the per-node border gateway protocol (BGP) status for each vserver. The BGP status for a particular vserver is "up" when at least one BGP peer group supporting that vserver is able to communicate with its peer router.

### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-node {<nodename>|local}] - Node**

This parameter selects the BGP status that match the specified node.

**[-vserver <vserver name>] - Vserver**

This parameter selects the BGP status for specified vserver.

**[-ipv4-status {unknown|unconfigured|up|down}] - IPv4 status**

This parameter selects the BGP status that matches the specified status for IPv4 address family.

**[-ipv6-status {unknown|unconfigured|up|down}] - IPv6 status**

This parameter selects the BGP status that matches the specified status for IPv6 address family.

#### Examples

```
cluster1::> network bgp vserver-status show
    Node                vserver    IPv4 status IPv6 status
    ------------------- ---------- ----------- -----------
    node1               vs1        up          up
```

# network cloud commands

## network cloud routing-table create

Create a new external routing table

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

#### Description

The `network cloud routing-table create` command creates a new external routing table.

#### Parameters

**-route-table-id <text> - Route Table ID (privilege: advanced)**

This parameter is used to provide the name of the external routing table to be created.

#### Examples

The following example creates an external routing table "eni-123456":

```
cluster1::> network cloud routing-table create -route-table-id eni-123456
```

## network cloud routing-table delete

Delete an existing external routing table

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

#### Description

The `network cloud routing-table delete` deletes an existing external routing table.

#### Parameters

**`-route-table-id <text>` - Route Table ID (privilege: advanced)**

This parameter is used to provide the name of an existing external routing table to be deleted.

#### Examples

The following example deletes the external routing table "eni-123456":

```
cluster1::> network cloud routing-table delete -route-table-id eni-123456
```

## network cloud routing-table show

Show existing external routing tables

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

#### Description

The `network cloud routing-table show` command retrieves the configured routing tables on mediator and displays them.

#### Parameters

**{ `[-fields <fieldname>,…]`**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| `[-instance ]` }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**`[-route-table-id <text>]` - Route Table ID (privilege: advanced)**

This parameter is the name of the external routing table to be shown.

#### Examples

The following example shows external routing tables, for this cluster, that are configured on the mediator

```
cluster1::> network cloud routing-table show
Route Table ID
 ----------------------
rtb-16924571
rtb-9c9245fb
rtb-a36ca1c4
3 entries were displayed.
```

# network connections commands

## network connections active show-clients

Show a count of the active connections by client

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

### Description

The `network connections active show-clients` command displays information about client connections, including the client's IP address and the number of client connections.

> The results of this command set are refreshed independently every 30 seconds and might not reflect the immediate state of the system.

### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-node {<nodename>|local}] - Node**

Use this parameter to display information only about the connections on the node you specify.

**[-vserver <vserver>] - Vserver**

This parameter is used by the system to break down the output per vserver.

**[-remote-address <Remote IP>] - Remote IP Address**

Use this parameter to display information only about the connections that use the remote IP address you specify.

**[-count <integer>] - Client Count**

Use this parameter to only clients with the number of active client connections you specify.

### Examples

The following example displays information about active client connections:

```
cluster1::> network connections active show-clients
Node    Vserver Name    Client IP Address     Count
------  --------------  -----------------     ------
node0   vs1             192.0.2.253                1
        vs2             192.0.2.252                2
        vs3             192.0.2.251                5
node1   vs1             192.0.2.250                1
        vs2             192.0.2.252                3
        vs2             customer.example.com       4
```

## network connections active show-lifs

Show a count of the active connections by logical interface

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

### Description

The `network connections active show-lifs` command displays the number of active connections on each logical interface, organized by node and Vserver.

> The results of this command set are refreshed independently every 30 seconds and might not reflect the immediate state of the system.

### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-node {<nodename>|local}] - Node**

Use this parameter to display information only about the connections on the node you specify.

**[-vserver <vserver>] - Vserver**

Use this parameter to display information only about the connections that are using the node or Vserver you specify.

**[-lif-name <lif-name>] - Logical Interface Name**

Use this parameter to display information only about the connections that are using the logical interface you specify.

**[-count <integer>] - Client Count**

Use this parameter to display only logical interfaces with the number of active client connections you specify.

**[-blocked-count <integer>] - (DEPRECATED)-Load Balancing Blocking Count**

> This parameter has been deprecated and may be removed in a future version of Data ONTAP.

Use this parameter to display information only about data logical interfaces blocked from migrating and the connection that is blocking it.

### Examples

The following example displays information about the servers and logical interfaces being used by all active connections:

```
cluster1::> network connections active show-lifs
Node      Vserver Name  Interface Name  Count
--------  ------------  --------------- ------
node0
          vs0           datalif1            3
          vs0           cluslif1            6
          vs0           cluslif2            5
node1
          vs0           datalif2            3
          vs0           cluslif1            3
          vs0           cluslif2            5
node2
          vs1           datalif2            1
          vs1           cluslif1            5
          vs1           cluslif2            3
node3
          vs1           datalif1            1
          vs1           cluslif1            2
          vs1           cluslif2            1
```

## network connections active show-protocols

Show a count of the active connections by protocol

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

### Description

The `network connections active show-protocols` command displays the number of active connections per protocol, organized by node.

> The results of this command set are refreshed independently every 30 seconds and might not reflect the immediate state of the system.

### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-node {<nodename>|local}] - Node**

Use this parameter to display information only about the connections on the node you specify.

**[-vserver <vserver>] - Vserver**

This parameter is used by the system to break down the output per vserver.

**[-proto {UDP|TCP}] - Protocol**

Use this parameter to display information only about the connections that use the network protocol you specify. Possible values include tcp (TCP), udp (UDP), and NA (not applicable).

**[-count <integer>] - Client Count**

Use this parameter to display only protocols with the number of active client connections you specify.

### Examples

The following example displays information about all network protocols being used by active connections:

```
cluster1::> network connections active show-protocols
Node     Vserver Name    Protocol   Count
-------  --------------  ---------  ------
node0
         vs1             UDP            19
         vs1             TCP            11
         vs2             UDP            17
node1
         vs1             UDP            14
         vs2             TCP            10
```

## network connections active show-services

Show a count of the active connections by service

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

### Description

The `network connections active show-services` command displays the number of active connections by protocol service, organized by node.

The results of this command set are refreshed independently every 30 seconds and might not reflect the immediate state of the system.

### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-node {<nodename>|local}] - Node**

Use this parameter to display information only about the connections on the node you specify.

**[-vserver <vserver>] - Vserver**

This parameter is used by the system to break down the output per vserver

**[-service <protocol service>] - Protocol Service**

Use this parameter to display information only about the connections that use the protocol service you specify. Possible values include: nfs, iscsi, and loopback.

**[-count <integer>] - Client Count**

Use this parameter to display information only about protocol services with the number of active client connections you specify.

### Examples

The following example displays information about all protocol services being used by active connections:

```
cluster1::> network connections active show-services
Node        Vserver Name    Service    Count
---------  --------------  ---------  ------
node0
            vs1             mount           3
            vs1             nfs            14
            vs1             nlm_v4          4
            vs1             cifs_srv        3
            vs1             port_map       18
            vs2             rclopcp        27
node1
            vs1             nfs             5
            vs2             rclopcp        12
            vs2             nfs             4
            vs2             port_map        8
```

## network connections active show

Show the active connections in this cluster

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

### Description

The `network connections active show` command displays information about active network connections.

> The results of this command set are refreshed independently every 30 seconds and might not reflect the immediate state of the system.

### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| [-print-ip-addresses ]**

Print IP addresses for remote hosts — do not attempt to resolve the addresses to a hostname.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-node {<nodename>|local}] - Node**

Selects the connections that match this parameter value.

**[-cid <Cid>] - Connection ID**

Selects the connections that match this parameter value.

**[-vserver <vserver>] - Vserver**

Selects the connections that match this parameter value.

**[-lif-name <lif-name>] - Logical Interface Name**

Selects the connections that match this parameter value.

**[-local-address <IP Address>] - Local IP address**

Selects the connections that match this parameter value.

**[-local-port <integer>] - Local Port**

Selects the connections that match this parameter value.

**[-remote-ip <InetAddress>] - Remote IP Address**

Selects the connections that match this parameter value.

**[-remote-host <Remote IP>] - Remote Host**

Selects the connections that match this parameter value.

**[-remote-port <integer>] - Remote Port**

Selects the connections that match this parameter value.

**[-proto {UDP|TCP}] - Protocol**

Selects the connections that match this parameter value. Possible values are tcp (TCP), udp (UDP), and NA (not applicable).

**[-lifid <integer>] - Logical Interface ID**

Selects the connections that match this parameter value.

**[-service <protocol service>] - Protocol Service**

Selects the connections that match this parameter value. Possible values include: nfs, iscsi, and loopback.

**[-lru {yes|no}] - Least Recently Used**

Selects the connections that match this parameter value.

**[-blocks-lb {true|false}] - Connection Blocks Load Balance Migrate**

Selects the logical interfaces that are blocked (true) or not blocked (false) from migrating due to an active client connection.

### Examples

The following example displays information about active network connections for the node named node0:

```
cluster1::> network connections active show node -node0

Vserver Interface        Remote
Name    Name:Local Port  IP Address:Port   Protocol/Service
------- ---------------- ----------------- ----------------
node0   cluslif1:7070    192.0.2.253:48621 UDP/rclopcp
node0   cluslif1:7070    192.0.2.253:48622 UDP/rclopcp
node0   cluslif2:7070    192.0.2.252:48644 UDP/rclopcp
node0   cluslif2:7070    192.0.2.250:48646 UDP/rclopcp
node0   cluslif1:7070    192.0.2.245:48621 UDP/rclopcp
node0   cluslif1:7070    192.0.2.245:48622 UDP/rclopcp
node0   cluslif2:7070    192.0.2.251:48644 UDP/rclopcp
node0   cluslif2:7070    192.0.2.251:48646 UDP/rclopcp
node0   cluslif1:7070    192.0.2.248:48621 UDP/rclopcp
node0   cluslif1:7070    192.0.2.246:48622 UDP/rclopcp
node0   cluslif2:7070    192.0.2.252:48644 UDP/rclopcp
node0   cluslif2:7070    192.0.2.250:48646 UDP/rclopcp
node0   cluslif1:7070    192.0.2.254:48621 UDP/rclopcp
node0   cluslif1:7070    192.0.2.253:48622 UDP/rclopcp
[...]
```

## network connections listening show

Show the listening connections in this cluster

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

### Description

The `network connections listening show` command displays information about network connections that are in an open and listening state.

### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-node {<nodename>|local}] - Node**

Selects the listening connections that match this parameter value.

**[-mgmt-cid <integer>] - Management Connection ID**

Selects the listening connections that match this parameter value.

**[-vserver <vserver>] - Vserver**

Selects the listening connections that match this parameter value.

**[-cid <integer>] - System Connection ID**

Selects the listening connections that match this parameter value.

**[-lif-name <lif-name>] - Logical Interface Name**

Selects the listening connections that match this parameter value.

**[-local-address <IP Address>] - Local IP Address**

Selects the listening connections that match this parameter value.

**[-local-port <integer>] - Local Port**

Selects the listening connections that match this parameter value.

**[-remote-ip <InetAddress>] - Remote IP Address**

Selects the listening connections that match this parameter value.

**[-remote-host <Remote IP>] - Remote Host**

Selects the listening connections that match this parameter value.

**[-remote-port <integer>] - Remote Port**

Selects the listening connections that match this parameter value.

**[-proto {UDP|TCP}] - Protocol**

Selects the listening connections that match this parameter value. Possible values include tcp (TCP), udp (UDP), and NA (not applicable).

**[-lifid <integer>] - Logical Interface ID**

Selects the listening connections that match this parameter value.

**[-service <protocol service>] - Protocol Service**

Selects the listening connections that match this parameter value. Possible values include: nfs, iscsi, and loopback.

**[-lru {yes|no}] - Least Recently Used**

Selects the listening connections that match this parameter value.

### Examples

The following example displays information about all listening network connections:

```
cluster1::> network connections listening show
Vserver Name Interface Name:Local Port  Protocol/Service
------------ -------------------------- -----------------
node0        cluslif1:7700              UDP/rclopcp
node0        cluslif2:7700              UDP/rclopcp
node1        cluslif1:7700              UDP/rclopcp
node1        cluslif2:7700              UDP/rclopcp
node2        cluslif1:7700              UDP/rclopcp
node2        cluslif2:7700              UDP/rclopcp
node3        cluslif1:7700              UDP/rclopcp
node3        cluslif2:7700              UDP/rclopcp
8 entries were displayed.
```

The following example displays detailed information about listening network connections for the node named node0:

```
cluster1::> network connections listening show -node node0
          Node: node0
Management Connection Id: 0
  System Connection Id: 0
         Vserver: vs0
 Logical Interface Name: datalif1
    Local IP address: 192.0.2.130
       Local Port: 111
    Remote IP address:
       Remote Port: 0
        Protocol: UDP
  Logical Interface Id: 1029
    Protocol Service: port_map
   least recently used: yes
          Node: node0
Management Connection Id: 1
  System Connection Id: 0
         Server: vs0
 Logical Interface Name: datalif2
    Local IP address: 192.0.2.131
       Local Port: 111
    Remote IP address:
       Remote Port: 0
        Protocol: UDP
  Logical Interface Id: 1030
    Protocol Service: port_map
   least recently used: yes
```

# network device-discovery commands

## network device-discovery show

Display device discovery information

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

The network device-discovery show command displays information about discovered devices. This information may be useful in determining the network topology or investigating connectivity issues. By default, the command displays the following information:

- Local interface
- Discovered device
- Discovered interface

- Discovered platform

### Parameters

**{ [-fields <fieldname>,…]**

Include the specified field or fields in the command output. Use '-fields ?' to display the valid fields.

**| [-instance ] }**

Use this parameter to display detailed information about all fields.

**[-node <nodename>] - Node**

Displays the discovery ports that match the node name.

**[-protocol {cdp|lldp}] - Protocol**

Displays the devices that are discovered by the given protocol.

**[-port <text>] - Port**

Displays the discovery ports that match the physical network port. For example, e0a will display devices discovered on port e0a.

**[-discovered-device <text>] - Discovered Device**

Displays the discovered devices that match the discovered device name.

**[-interface <text>] - Discovered Device Interface**

Displays the discovered devices that match this interface port name. The format is dependent on the reporting device. For example: FastEthernet0/12

**[-device-ip <IP Address>,…] - Discovered Device IP Addresses**

Displays the discovered devices that match the IP address(es). At present, only IPv4 addresses are included. It is recommended to use wildcards around the desired value.

**[-platform <text>] - Discovered Device Platform**

Displays the discovery ports that contain the platform of discovered devices. For example: N5K-C5010P-BF

**[-version <text>] - Discovered Device Version**

Displays the discovery ports that contain the version of discovered devices.

**[-chassis-id <text>] - Discovered Device Chassis ID**

Displays the discovered devices that match the chassis ID.

**[-system-name <text>] - Discovered Device System Name**

Displays the discovered devices that match the system name.

**[-hold-time-remaining <integer>] - Discovered Device's Remaining Hold Time**

Displays the discovered devices that match the remaining packet hold time in seconds. If an advertisement from the device isn't received before this time reaches zero, the entry will expire and be removed from the list. For example, "<120" will display discovered devices which will expire within the next 120 seconds.

**[-capabilities {router|trans-bridge|source-route-bridge|switch|host|igmp|repeater|phone}] - Discovered Device Capabilities**

Displays the discovered devices that match the capability or capabilities. Possible values:

- "router" - Router
- "trans-bridge" - Trans Bridge
- "source-route-bridge" - Source Route Bridge
- "switch" - Switch
- "host" - Host
- "igmp" - IGMP
- "repeater" - Repeater
- "phone" - Phone

**Examples**

```
cluster1::> network device-discovery show

Node/       Local  Discovered
Protocol    Port   Device                    Interface         Platform
----------- ------ ------------------------- ----------------
----------------
node1/cdp
            e0a    US-LS01-5010-F11-NX.example.com(SSI142311PD)
                                             Ethernet100/1/17  N5K-C5010P-
BF
            e0b    US-LS01-5010-F11-NX.example.com(SSI142311PD)
                                             Ethernet100/1/18  N5K-C5010P-
BF
            e1a    US-LS01-5010-F11-NX.example.com(SSI142311PD)
                                             Ethernet100/1/2   N5K-C5010P-
BF
node2/cdp
            e0a    US-LS01-5010-F11-NX.example.com(SSI142311PD)
                                             Ethernet100/1/19  N5K-C5010P-
BF
            e0b    US-LS01-5010-F11-NX.example.com(SSI142311PD)
                                             Ethernet100/1/20  N5K-C5010P-
BF
            e1a    US-LS01-5010-F11-NX.example.com(SSI142311PD)
                                             Ethernet100/1/4   N5K-C5010P-
BF
            e1c    US-LS01-5010-F11-NX.example.com(SSI142311PD)
                                             Ethernet100/1/36  N5K-C5010P-
BF
            e1d    US-LS01-5010-F11-NX.example.com(SSI142311PD)
                                             Ethernet100/1/35  N5K-C5010P-
BF
8 entries were displayed.
```

# network fcp commands

## network fcp adapter modify

Modify the fcp adapter settings

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

Modifies the FCP target adapter information.

The adapter argument is in the form Xy or Xy_z where X and z are integers and y is a letter. An example is 4a or 4a_1.

You cannot bring an adapter offline until all logical interfaces connected to that adapter are offline. Use the network interface modify command to take your logical interfaces offline.

The speed option sets the Fibre Channel link speed of an adapter. You can set adapters that support:

- 10Gb/s to 10 or auto
- 8Gb/s to 2, 4, 8 or auto
- 4Gb/s to 2, 4 or auto
- 2Gb/s to 2 or auto

By default, the link speed option is set to auto for auto negotiation. Setting the link speed to a specific value disables auto negotiation. Under certain conditions, a speed mismatch can prevent the adapter from coming online.

The system reports the actual link speed with the "Data Link Rate (Gbit)" field in the output of network fcp adapter show`-instance` .

### Parameters

**`-node {<nodename>|local}` - Node**

Specifies the node of the target adapter.

**`-adapter <text>` - Adapter**

Specifies the target adapter.

**`[-status-admin {down|up}]` - Administrative Status**

Specifies the desired (administrative) status of the adapter. To view the actual operational status, run network fcp adapter show`-fields` *`status-oper`* .

**`[-speed {1|2|4|8|10|16|32|auto}]` - Configured Speed**

Specifies the adapter configuration speed in Gigabytes.

### Examples

```
cluster1::> network fcp adapter modify -node node1 -adapter 0d -speed 2
```

Configures the speed of FCP adapter 0d on node1 to 2 Gb/s.

### Related Links

- network interface modify
- network fcp adapter show

## network fcp adapter show

Display FCP adapters

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

Displays FCP target adapter information. You can also use this information to determine if adapters are active and online.

The adapter argument is in the form Xy or Xy_z where X and z are integers and y is a letter. An example is 4a or 4a_1.

### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-node {<nodename>|local}] - Node**

If this parameter is specified, the command displays information only about the FCP target adapters that are present on the specified node.

**[-adapter <text>] - Adapter**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified name.

**[-description <text>] - Description**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified description.

**[-physical-protocol {fibre-channel|ethernet}] - Physical Protocol**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified physical protocol. Possible values are *fibre-channel* and *ethernet* .

**[-max-speed {1|2|4|8|10|16|32|auto}] - Maximum Speed**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified maxium speed.

**[-status-admin {down|up}] - Administrative Status**

If this parameter is specified, the command displays information only about the FCP target adapters that match the administrative state. Possible values are *up* and *down* .

**[-status-oper <text>] - Operational Status**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified operational status.

**[-status-extended <text>] - Extended Status**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified extended status.

**[-portaddr <Hex Integer>] - Host Port Address**

If this parameter is specified, the command displays information only about the FCP target adapters connected with the specified fabric port address.

**[-firmware-rev <text>] - Firmware Revision**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified firmware revision.

**[-data-link-rate <integer>] - Data Link Rate (Gbit)**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified data link rate.

**[-fabric-established {true|false}] - Fabric Established**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified fabric login establishment state.

**[-fabric-name <text>] - Fabric Name**

If this parameter is specified, the command displays information only about the FCP target adapters that are logged in to the fabric with the specified WWN.

**[-conn-established {loop|ptp}] - Connection Established**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified connection type. Possible values are *loop* and *ptp* .

**[-is-conn-established {true|false}] - Is Connection Established**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified connection established state.

**[-media-type {loop|ptp|auto}] - Mediatype**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified configured media type. Possible values are *loop* , *ptp* , and *auto* .

**[-speed {1|2|4|8|10|16|32|auto}] - Configured Speed**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified configured speed. If the adapter is set to auto-negotiate, then the value will be *auto* .

**[-data-protocols-supported {fcp|fc-nvme}] - Data Protocols Supported**

If this parameter is specified, the command displays information only about the FCP target adapters that may host LIFs with the specified data protocol. Possible values are *fcp* and *fc-nvme* .

**[-domain-id <integer>] - Domain ID**

If this parameter is specified, the command displays information only about the FCP target adapters with a domain identifier that matches the specified domain identifier.

**[-fc-wwnn <text>] - Adapter WWNN**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified world wide node name.

**[-fc-wwpn <text>] - Adapter WWPN**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified world wide port name.

**[-switch-port <text>] - Switch Port**

If this parameter is specified, the command displays information only about the FCP target adapters that are connected to the specified switch port.

**[-sfp-formfactor <text>] - Form Factor Of Transceiver**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified SFP form factor.

**[-sfp-vendor-name <text>] - Vendor Name Of Transceiver**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified SFP vendor name.

**[-sfp-part-number <text>] - Part Number Of Transceiver**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified SFP part number.

**[-sfp-rev <text>] - Revision Of Transceiver**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified SFP revision number.

**[-sfp-serial-number <text>] - Serial Number Of Transceiver**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified SFP serial number.

**[-sfp-fc-speed-capabilities <text>] - FC Capabilities Of Transceiver**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified SFP speed capabilities.

**[-sfp-vendor-oui <text>] - Vendor OUI Of Transceiver**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified SFP vendor OUI.

**[-sfp-wavelength <integer>] - Wavelength In Nanometers**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified SFP wavelength.

**[-sfp-date-code <text>] - Date Code Of Transceiver**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified SFP date code.

**[-is-sfp-optical-transceiver-valid {true|false}] - Validity Of Transceiver**

If this parameter is specified, the command displays information only about the FCP target adapters that

match whether the SFP is installed and valid.

**[-sfp-connector <text>] - Connector Used**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified SFP connector type.

**[-sfp-encoding <text>] - Encoding Used**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified SFP physical encoding.

**[-is-sfp-diagnostics-internally-calibrated {true|false}] - Is Internally Calibrated**

If this parameter is specified, the command displays information only about the FCP target adapters that match whether the SFP diagnostics are internally calibrated or not.

**[-sfp-diagnostic-monitoring-type <Hex Integer>] - Diagnostic Monitoring Type**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified SFP diagnostic monitoring type.

**[-sfp-ddm-capabilities <text>] - Status Monitoring Available**

If this parameter is specified, the command displays information only about the FCP target adapters that match whether the specified SFP digital diagnostics monitoring are supported or not.

**[-sfp-sff8472-compliance <Hex Integer>] - SFF-8472 Compliance**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified SFP SFF8472 compliance.

**[-sfp-rx-power <text>] - Received Optical Power**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified observed SFP receive power.

**[-is-sfp-rx-power-in-range {true|false}] - Is Received Power In Range**

If this parameter is specified, the command displays information only about the FCP target adapters that match whether the observed SFP receive power is within the valid range for the SFP.

**[-sfp-tx-power <text>] - SFP Transmitted Optical Power**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified SFP transmit power.

**[-is-sfp-tx-power-in-range {true|false}] - Is Xmit Power In Range**

If this parameter is specified, the command displays information only about the FCP target adapters that match whether the observed SFP transmit power is within the valid range for the SFP.

**[-sfp-ddm-status-control <Hex Integer>] - DDM Status**

If this parameter is specified, the command displays information only about the FCP target adapters that match the specified SFP DDM status and control.

**[-is-sfp-tx-in-disable {true|false}] - Is Xmit Disabled**

If this parameter is specified, the command displays information only about the FCP target adapters that match whether the observed SFP transmitter is in disabled state.

**[-is-sfp-tx-in-fault {true|false}] - Is Xmit In Fault**

If this parameter is specified, the command displays information only about the FCP target adapters that match whether the observed SFP transmitter is in faulted state.

**[-is-sfp-rx-in-los {true|false}] - Is Receiver In LOS**

If this parameter is specified, the command displays information only about the FCP target adapters that match whether the observed SFP receiver is in loss of signal state.

**Examples**

```
cluster1::> fcp adapter show
                     Connection  Host
Node         Adapter Established Port Address
------------ ------- ----------- ------------
sti6280-021   0a      ptp         30012c
```

The example above displays information regarding FCP adapters within cluster1.

```
cluster1::> fcp adapter show -instance -node sti6280-021 -adapter 0a
Node: sti6280-021
                      Adapter: 0a
                  Description: Fibre Channel Target Adapter 0a (QLogic
2532 (2562), rev. 2, 8G)
              Physical Protocol: fibre-channel
                  Maximum Speed: 8
         Administrative Status: up
            Operational Status: online
               Extended Status: ADAPTER UP
              Host Port Address: 30012c
              Firmware Revision: 5.8.0
         Data Link Rate (Gbit): 4
            Fabric Established: true
                   Fabric Name: 20:14:54:7f:ee:54:b9:01
        Connection Established: ptp
     Is Connection Established: true
                     Mediatype: ptp
              Configured Speed: auto
                  Adapter WWNN: 50:0a:09:80:8f:7f:8b:1c
                  Adapter WWPN: 50:0a:09:81:8f:7f:8b:1c
                   Switch Port: RTP-AG01-410B51:1/41
    Form Factor Of Transceiver: SFP
    Vendor Name Of Transceiver: OPNEXT,INC
    Part Number Of Transceiver: TRS2000EN-SC01
       Revision Of Transceiver: 0000
  Serial Number Of Transceiver: T10H64793
```

```
 FC Capabilities Of Transceiver: 10 (Gbit/sec)
      Vendor OUI Of Transceiver: 0:11:64
       Wavelength In Nanometers: 850
       Date Code Of Transceiver: 10:08:17
        Validity Of Transceiver: true
                 Connector Used: LC
                  Encoding Used: 64B66B
       Is Internally Calibrated: true
     Diagnostic Monitoring Type: 68
    Status Monitoring Available: fa {Rx_Loss_of_Sig, Tx_Fault, Tx_Disable}
            SFF-8472 Compliance: 5
         Received Optical Power: 441.3 (uWatts)
     Is Received Power In Range: true
  SFP Transmitted Optical Power: 600.4 (uWatts)
        Is Xmit Power In Range: true
                    DDM Status: 30
              Is Xmit Disabled: false
              Is Xmit In Fault: false
             Is Receiver In LOS: false
```

The example above displays detailed information regarding FCP adapter 0a in sti6280-021 within cluster1.

## network fcp topology show

### FCP topology interconnect elements per adapter

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

Display FCP topology interconnect elements per adapter.

#### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-node {<nodename>|local}] - Node**

Use this parameter to select the interconnect elements for adapters that are located on the node that you specify.

**[-adapter <text>] - Adapter**

Use this parameter to select the interconnect elements for the specified adapter.

**[-domain-id <integer>] - Domain Id**

Use this parameter to select the interconnect elements with the specified domain identifier.

**[-port-wwpn <text>] - Port WWPN**

Use this parameter to select the interconnect elements with the specified port world wide name.

**[-switch-name <text>] - Switch Name**

Use this parameter to select the interconnect elements with the specified switch.

**[-switch-vendor <text>] - Switch Vendor**

Use this parameter to select the interconnect elements with the specified vendor.

**[-switch-release <text>] - Switch Release**

Use this parameter to select the interconnect elements with the specified release.

**[-switch-wwpn <text>] - Switch WWPN**

Use this parameter to select the interconnect elements with the specified world wide port name.

**[-switch-wwn <text>] - Switch WWN**

Use this parameter to select the interconnect elements with the specified world wide name.

**[-port-count <integer>] - Port Count**

Use this parameter to select the interconnect elements with the specified port count.

**[-port-slot <text>] - Port Slot**

Use this parameter to select the interconnect elements with the specified port slot.

**[-port-state {Unknown|Online|Offline|Testing|Fault}] - Port State**

Use this parameter to select the interconnect elements with the specified port state.

**[-port-type {None|N-Port|NL-Port|FNL-Port|NX-Port|F-Port|FL-Port|E-Port|B-Port|TNP-Port|TF-Port|NV-Port|FV-Port|SD-Port|TE-Port|TL-Port}] - Port Type**

Use this parameter to select the interconnect elements with the specified port type.

**[-port-attached-wwpn <text>] - Attached Port WWPN**

Use this parameter to select the interconnect elements with the specified attached wwpn.

**[-port-attached-id <text>] - Attached Port Id**

Use this parameter to select the interconnect elements with the specified attached id.

**[-port-attached-visible <text>] - Visible**

Use this parameter to select the interconnect elements with the specified visiblity flag on attached port structure.

**Examples**

```
cluster1::> network fcp topology show
Switch connected to the adapter 0c
      Switch Name: ssan-fc0e-d58
    Switch Vendor: Cisco Systems, Inc.
   Switch Release: 5.2(1)N1(9)
    Switch Domain: 4
       Switch WWN: 20:05:00:05:9b:26:f4:c1
       Port Count: 20

Port    Port WWN                State     Type      Attached WWPN
Port Id
------- ----------------------- --------- ---------
---------------------- ------------
vfc9    20:08:00:05:9b:26:f4:ff Offline   None      -
-
vfc10   20:15:00:05:9b:26:f4:ff Online    TF-Port
50:0a:09:82:8d:92:4c:ff 0x0407c0  *
vfc11   20:16:00:05:9b:26:f4:ff Online    TF-Port
50:0a:09:81:8d:e2:4e:ec 0x040800  *

Switch connected to the adapter 0c
      Switch Name: ssan-fc0e-d58
    Switch Vendor: Cisco Systems, Inc.
   Switch Release: 5.2(1)N1(9)
    Switch Domain: 4
       Switch WWN: 20:05:00:05:9b:26:f4:c1
       Port Count: 20

Port    Port WWN                State     Type      Attached WWPN
Port Id
------- ----------------------- --------- ---------
---------------------- ------------
vfc20   20:13:00:05:9b:26:f4:ff Offline   None      -
-
vfc21   20:14:00:05:9b:26:f4:ff Online    TF-Port
50:0a:09:81:8d:92:4c:ff 0x0407a0  *

5 entries were displayed.
```

The example above show FCP topology interconnect information for the cluster.

## network fcp zone show

Display the active zone set information

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

Displays the active zone set information.

### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-node {<nodename>|local}] - Node**

Use this parameter to select the active zone set information for adapters that located on the node name that you specify.

**[-adapter <text>] - Adapter**

Use this parameter to select the active zone set information for the specified adapter.

**[-zoneset-name <text>] - Zoneset Name**

Use this parameter to select the active zone set information for the specified zone set name.

**[-zone-name <text>] - Zone Name**

Use this parameter to select the active zone set information for the specified zone name.

**[-unique <integer>] - Unique**

A unique index for each zoneset record.

**[-type-name <text>] - Type Name**

Use this parameter to select the active zone set information with the specified symbolic type.

**[-type <integer>] - Type**

Use this parameter to select the active zone set information with the specified port type.

**[-port-id <Hex Integer>] - Member Port Id**

Use this parameter to select the active zone set information with the specified member port id.

**[-domain-id <integer>] - Member Domain Id**

Use this parameter to select the active zone set information with the specified member domain id.

**[-port <integer>] - Member Port**

Use this parameter to select the active zone set information with the specified member port.

**[-wwn <text>] - Member WWN**

Use this parameter to select the active zone set information with the specified member WWN.

**[-zone-count <integer>] - Zone Count**

Use this parameter to select the active zone set information with the specified number of zones.

**[-zone-member-count <integer>] - Zone Member Count**

Use this parameter to select the active zone set information with the specified number of zone members in a zone.

**[-contents <text>] - Member Contents**

Use this parameter to select the active zone set information using any type.

**Examples**

```
cluster1::> network fcp adapter zone show
                                          Member
                       Zone Name          Type                WWN
 --------------------- ------------------ -------------------
-----------------
Active Zone Set on adapter 0c
   Zone Set Name: zoneset_name
                       zone_name_1        Port ID             -
                       zone_name_1        Port ID             -
                       zone_name_1        Port ID             -
                       zone_name_2        Domain ID/Port      -
                       zone_name_2        Domain ID/Port      -
                       zone_name_2        Domain ID/Port      -
                       zone_name_3        Fabric Port Name
00:00:00:00:00:00:00:00
                       zone_name_3        Fabric Port Name
01:00:00:00:00:00:00:00
                       zone_name_3        Fabric Port Name
02:00:00:00:00:00:00:00

9 entries were displayed.
```

The example above displays information regarding active zone set information for the cluster.

# network interface commands

## network interface create

Create a logical interface

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

**Description**

The `network interface create` command creates a logical interface (LIF).

A logical interface is an IP address associated with a physical network port. For logical interfaces using NAS data protocols, the interface can fail over or be migrated to a different physical port in the event of component failures, thereby continuing to provide network access despite the component failure. Logical interfaces using SAN data protocols do not support migration or failover.

On some cloud platforms, this operation might perform changes to the external route tables.

**Parameters**

**`-vserver <vserver>` - Vserver Name**

Use this parameter to specify the Vserver on which the LIF is created.

**`-lif <lif-name>` - Logical Interface Name**

Use this parameter to specify the name of the LIF that is created. For iSCSI and FC LIFs, the name cannot be more than 254 characters.

**`[-service-policy <text>]` - Service Policy**

Use this parameter to specify a service policy for the LIF. If no policy is specified, a default policy will be assigned automatically. Use the network interface service-policy show command to review available service policies.

**`[-role {undef|cluster|data|node-mgmt|intercluster|cluster-mgmt}]` - (DEPRECATED)-Role**

This parameter has been deprecated and may be removed in a future version of ONTAP. Use the `-service-policy` parameter instead.

Use this parameter to specify the role of the LIF. LIFs can have one of five roles:

- Cluster LIFs, which provide communication among the nodes in a cluster
- Intercluster LIFs, which provide communication among peered clusters
- Data LIFs, which provide data access to NAS and SAN clients
- Node-management LIFs, which provide access to cluster management functionality
- Cluster-management LIFs, which provide access to cluster management functionality

LIFs with the cluster-management role behave as LIFs with the node-management role except that cluster-management LIFs can failover between nodes.

**`[-data-protocol {nfs|cifs|iscsi|fcp|fcache|none|fc-nvme|s3}]` - Data Protocol**

Use this parameter to specify the list of data protocols that can be served by the LIF. The supported protocols are NFS, CIFS, iSCSI, FCP, and FC-NVMe. NFS and CIFS are available by default when you create a LIF. If you specify "none", the LIF does not support any data protocols. Also, none, iscsi, fcp or fc-nvme cannot be combined with any other protocols.

The data-protocol field must be specified when the LIF is created and cannot be modified later.

The NFS protocol relies on firewall services included in the built-in "data" and "mgmt-nfs" firewall policies. Assigning a different firewall policy might disrupt some NFS client implementations.

**-address <IP Address> - Network Address**

Use this parameter to specify the LIF's IP address.

A cluster LIF cannot be on the same subnet as a management or data LIF.

**{ -netmask <IP Address> - Netmask**

Use this parameter to specify the LIF's netmask.

**| -netmask-length <integer> - Bits in the Netmask**

Use this parameter to specify the length (in bits) of the LIF's netmask.

**| -is-vip <true> - Is VIP LIF**

Use this parameter to display only logical interfaces matching a specify "is-vip" flag. Specifying "true" matches only LIFs to implement a Virtual IP; "false" matches only LIFs that do not.

**{ [-auto <true>] - Allocate Link Local IPv4 Address**

Use this parameter to specify whether IPv4 link local addressing is enabled for this LIF.

**| [-subnet-name <subnet name>] - Subnet Name }**

Use this parameter to allocate the interface address from a subnet. If needed, a default route will be created for this subnet.

**[-home-node <nodename>] - Home Node**

Use this parameter to specify the LIF's home node. The home node is the node to which the LIF returns when the network interface revert command is run on the LIF.

**[-home-port {<netport>|<ifgrp>}] - Home Port**

Use this parameter to specify the LIF's home port or interface group. The home port is the port or interface group to which the LIF returns when the network interface revert command is run on the LIF.

**[-status-admin {up|down}] - Administrative Status**

Use this parameter to specify whether the initial administrative status of the LIF is up or down. The default setting is `up` . The administrative status can differ from the operational status For example, if you specify the status as up but a network problem prevents the interface from functioning, the operational status remains as down.

**[-failover-policy {system-defined|local-only|sfo-partner-only|disabled|broadcast-domain-wide}] - Failover Policy**

Use this parameter to specify the failover policy for the LIF.

- system-defined - The system determines appropriate failover targets for the LIF. The default behavior is that failover targets are chosen from the LIF's current hosting node and also from one other non-parter node when possible.
- local-only - The LIF fails over to a port on the local or home node of the LIF.
- sfo-partner-only - The LIF fails over to a port on the home node or SFO partner only.

- broadcast-domain-wide - The LIF fails over to a port in the same broadcast domain as the home port.
- disabled - Failover is disabled for the LIF.

The failover policy for cluster logical interfaces is local-only and cannot be changed. The default failover policy for data logical interfaces is system-defined. This value can be changed.

Logical interfaces for SAN protocols do not support failover. Thus, such interfaces will always show this parameter as `disabled` .

**[-firewall-policy <policy>] - Firewall Policy**

Use this parameter to specify the firewall policy for the LIF. A LIF can use a default firewall policy that corresponds to its role (management, cluster, intercluster, or data) or a custom firewall policy created by an administrator. View and modify existing firewall policies using the system services firewall policy show and system services firewall policy modify commands, respectively.

The NFS data protocol relies on firewall services included in the built-in "data" and "mgmt-nfs" firewall policies. Assigning a different firewall policy might disrupt some NFS client implementations.

**[-auto-revert {true|false}] - Auto Revert**

Use this parameter to specify whether a data LIF is automatically reverted to its home node under certain circumstances. These circumstances include startup, when the status of the management database changes to either master or secondary, or when the network connection is made. The default setting is `false` . If you set the value of this parameter to `true` , load balancing migration capability of the data LIF is disabled (the `-allow-lb-migrate` parameter is set to `false` ).

Logical interfaces for SAN traffic do not support auto-revet. Thus, this parameter is always `false` on such interfaces.

**[-dns-zone {<zone-name>|none}] - Fully Qualified DNS Zone Name**

Use this parameter to specify a unique, fully qualified domain name of a DNS zone to which this data LIF is added. You can associate a data LIF with a single DNS zone. All data LIFs included in a zone must be on the same Vserver. If a LIF is not added to a DNS zone the data LIF is created with the value `none` .

**[-listen-for-dns-query {true|false}] - DNS Query Listen Enable**

Use this parameter to specify if the LIF has to listen for DNS queries. The default value for this parameter is true.

**[-allow-lb-migrate {true|false}] - (DEPRECATED)-Load Balancing Migrate Allowed (privilege: advanced)**

This parameter has been deprecated and may be removed in a future version of Data ONTAP.

Use this parameter to specify whether load balancing migration is activated for this data LIF. The default value of this parameter is `false` . If you set the value of this parameter to `true` , automatic revert capability for this data LIF is disabled (the `-auto-revert` parameter is set to `false` ). Also, data LIFs that migrate as a result of load balancing adhere to network interface failover rules.

During times when a LIF is hosting active NFSv4, CIFS, or NRV connections, load balancing based LIF migrations between nodes will be temporarily disabled.

**[-lb-weight {load|0..100}] - Load Balanced Weight (privilege: advanced)**

Use this parameter to specify a load balancing weight for a data LIF. A valid numeric load balancing weight is any integer between 0 and 100. When you specify the same load balancing weight for all data LIFs in a DNS zone, client requests are uniformly distributed, similar to round-robin DNS. A data LIF with a low load balancing weight is made available for client requests less frequently than one that has a high load balancing weight. "load" is the default value of this parameter. If set to "load", node utilization statistics are used to dynamically assign the load balancing weight.

**[-failover-group <failover-group>] - Failover Group Name**

Use this parameter to specify the name of the failover group to associate with the LIF. Manage failover groups by using the `network interface failover-groups` command. Each broadcast domain has a default failover group which is created by the system automatically and has the same name as the broadcast domain. The failover group associated with the broadcast domain includes all ports in the broadcast domain. A logical interface's failover group is set to the failover group of the home port's broadcast domain by default, but this value can be modified.

Logical interfaces for SAN protocols do not support failover. Thus, this parameter cannot be specified for such interfaces.

**[-comment <text>] - Comment**

Use this parameter to specify the comment to associate with the LIF.

**[-force-subnet-association <true>] - Force the LIF's Subnet Association**

This command will fail if the IP address falls within the address range of a named subnet. Set this to true to acquire the address from the named subnet and assign the subnet to the LIF.

**[-is-dns-update-enabled {true|false}] - Is Dynamic DNS Update Enabled?**

If this parameter is set to *true* , then dynamic DNS update is sent to the DNS server for the particular LIF entry if dynamic DNS updates are enabled for the corresponding Vserver. This field is set to *true* by default for both IPv4 and IPv6 LIFs. DNS Update is not supported on LIFs not configured with either the NFS or CIFS protocol.

**[-probe-port <integer>] - Probe-port for Cloud Load Balancer**

Use this parameter to specify a probe-port for the LIF in the Azure environment. It is a required field in the Azure environment. If no probe-port is specified, an error would be returned.

**[-broadcast-domain <text>] - Broadcast Domain**

Use this parameter to display the broadcast domain that contains the home port of the logical interface.

## Examples

The following example creates an IPv4 LIF named datalif1 and an IPv6 LIF named datalif2 on a Vserver named vs0. Their home node is node0 and home port is e0c. The failover policy *broadcast-domain-wide* is assigned to both LIFs. The firewall policy is *data* and the LIFs are automatically reverted to their home node at startup and under other circumstances. The datalif1 has the IP address 192.0.2.130 and netmask 255.255.255.128, and datalif2 has the IP address 3ffe:1::aaaa and netmask length of 64.

```
cluster1::> network interface create -vserver vs0 -lif datalif1 -role data
-home-node node0 -home-port e0c -address 192.0.2.130 -netmask
255.255.255.128 -failover-policy broadcast-domain-wide -firewall-policy
data -auto-revert true
cluster1::> network interface create -vserver vs0 -lif datalif2 -role data
-home-node node0 -home-port e0c -address 3ffe:1::aaaa -netmask-length 64
-failover-policy broadcast-domain-wide -firewall-policy data -auto-revert
true
```

**Related Links**

- network interface service-policy show
- network interface revert
- system services firewall policy show
- system services firewall policy modify

## network interface delete

### Delete a logical interface

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

**Description**

The `network interface delete` command deletes a logical interface (LIF) from a Vserver. Only administratively down LIFs can be deleted. To make a LIF administratively down, use the network interface modify command to set the "status-admin" parameter to "down".

If the LIF is configured for a SAN protocol and is part of a portset, the LIF must be removed from the portset before it can be deleted. To determine if a LIF is in a portset, use the lun portset show command. To remove the LIF from the portset, use the lun portset remove command.

On some cloud platforms, this operation might perform changes to the external route tables.

**Parameters**

**`-vserver <vserver>` - Vserver Name**

Use this parameter to specify the Vserver on which the logical interface to be deleted is located.

**`-lif <lif-name>` - Logical Interface Name**

Use this parameter to specify the logical interface to delete.

**Examples**

The following example deletes a logical interface named cluslif3 that is located on a Vserver named vs0.

```
cluster1::> network interface delete -vserver vs0 -lif cluslif3
```

**Related Links**

- network interface modify
- lun portset show
- lun portset remove

## network interface migrate-all

Migrate all data logical interfaces away from the specified node

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

The `network interface migrate-all` command migrates all data logical interfaces from the node you specify.

> Manual migration of a logical interface can take up to 15 seconds to complete. Logical interface migration is a best-effort command and can only be completed if the destination node and port are operational. Logical interface migration requires that the logical interface be pre-configured with valid failover rules to facilitate failover to a remote node.

> Logical interfaces for SAN protocols do not support migration. Attempts to do so will result in an error.

> On some cloud platforms, this operation might perform changes to the external route tables.

### Parameters

**`-node <nodename>` - Node**

Use this parameter to specify the node from which all logical interfaces are migrated. Each data logical interface is migrated to another node in the cluster, assuming that the logical interface is configured with failover rules that specify an operational node and port.

**`[-port {<netport>|<ifgrp>}]` - Port**

Use this parameter to specify the port from which all logical interfaces are migrated. This option cannot be used with asynchronous migrations. If this parameter is not specified, then logical interfaces will be migrated away from all ports on the specified node.

### Examples

The following example migrates all data logical interfaces from the current (local) node.

```
cluster1::> network interface migrate-all -node local
```

## network interface migrate

Migrate a logical interface to a different port

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

The `network interface migrate` command migrates a logical interface to a port or interface group on the node you specify.

Manual migration of a logical interface can take up to 15 seconds to complete. Also, when you migrate a cluster logical interface, you must do so from the local node. Logical interface migration is a best-effort command, and can only be completed if the destination node and port are operational

Logical interfaces for SAN protocols do not support migration. Attempts to do so will result in an error.

On some cloud platforms, this operation might perform changes to the external route tables.

### Parameters

**`-vserver <vserver>` - Vserver Name**

Use this parameter to specify the Vserver that owns the logical interface that is to be migrated.

**`-lif <lif-name>` - Logical Interface Name**

Use this parameter to specify the logical interface that is to be migrated.

**`-destination-node <nodename>` - Destination Node**

Use this parameter to specify the node to which the logical interface is to be migrated.

**`[-destination-port {<netport>|<ifgrp>}]` - Destination Port**

Use this parameter to specify the port or interface group to which the logical interface is to be migrated.

**`[-force <true>]` - Force Migrate Data LIF Flag (privilege: advanced)**

Use this parameter to force the migration operation. Intended for disaster recovery only. Specifying this parameter implies the LIF's current node is down and will skip removing the LIF from its current node.

### Examples

The following example migrates a logical interface named datalif1 on a Vserver named vs0 to port e0c on a node named node2:

```
cluster1::> network interface migrate -vserver vs0 -lif datalif1 -dest
-node node2 -dest-port e0c
```

## network interface modify

### Modify a logical interface

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

The `network interface modify` command modifies attributes of a logical interface (LIF).

> You cannot modify some properties of an iSCSI or FCP LIF, such as `-home-node` or `-home-port` , if the LIF is in a portset. To modify these properties, first remove the LIF from the portset. To determine if a LIF is in a portset, use the lun portset show command. To remove the LIF from the portset, use the lun portset remove command.

> On some cloud platforms, this operation might perform changes to the external route tables.

#### Parameters

**`-vserver <vserver>` - Vserver Name**

Use this parameter to specify the Vserver on which the LIF to be modified is located.

**`-lif <lif-name>` - Logical Interface Name**

Use this parameter to specify the name of the LIF that is to be modified

**`[-service-policy <text>]` - Service Policy**

Use this parameter to modify the service policy associated with the LIF.

**`[-address <IP Address>]` - Network Address**

Use this parameter to modify the LIF's IP address.

> A cluster LIF cannot be on the same subnet as a data or management LIF.

**`{ [-netmask <IP Address>]` - Netmask**

Use this parameter to modify the LIF's netmask.

**`| [-netmask-length <integer>]` - Bits in the Netmask**

Use this parameter to modify the length (in bits) of the LIF's netmask.

**`| [-subnet-name <subnet name>]` - Subnet Name }**

Use this parameter to allocate the interface address from a subnet. Modifying this parameter will cause a new IP address to be allocated and assigned to the interface.

**`[-home-node <nodename>]` - Home Node**

Use this parameter to modify the LIF's home node. The home node is the node to which the LIF returns when the network interface revert command is run on that LIF.

**`[-home-port {<netport>|<ifgrp>}]` - Home Port**

Use this parameter to modify the LIF's home port. The home port is the port or interface group to which the

LIF returns when the network interface revert command is run on that LIF.

> If you change this parameter for a cluster or management LIF, you must reboot the storage system to force the change to take effect.

**[-status-admin {up|down}] - Administrative Status**

Use this parameter to modify the administrative status of the LIF. The administrative status can differ from the operational status. For example, if you specify the status as `up` but a network problem prevents the interface from functioning, the operational status remains as `down` .

**[-failover-policy {system-defined|local-only|sfo-partner-only|disabled|broadcast-domain-wide}] - Failover Policy**

Use this parameter to modify the failover policy for the LIF.

- system-defined - The system determines appropriate failover targets for the LIF. The default behavior is that failover targets are chosen from the LIF's current hosting node and also from one other non-partner node when possible.
- local-only - The LIF fails over to a port on the local or home node of the LIF.
- sfo-partner-only - The LIF fails over to a port on the home node or SFO partner only.
- broadcast-domain-wide - The LIF fails over to a port in the same broadcast domain as the home port.
- disabled - Failover is disabled for the LIF.

> The failover policy for cluster logical interfaces is local-only and cannot be changed. The default failover policy for data logical interfaces is system-defined. This value can be changed.

> Logical interfaces for SAN protocols do not support failover. Thus, such interfaces always show this parameter as `disabled` .

**[-firewall-policy <policy>] - Firewall Policy**

Use this parameter to set the firewall policy for the LIF. A LIF can use a default firewall policy that corresponds to its role (management, cluster, or data) or a custom firewall policy created by an administrator. When using a custom policy, the interface will fallback on its role's default policy for unspecified services. View existing firewall policies with the "system services firewall policy show " command. Modify existing firewall policies with the "system services firewall policy modify " command.

> The NFS data protocol relies on firewall services included in the built-in "data" and "mgmt-nfs" firewall policies. Assigning a different firewall policy policy might disrupt some NFS client implementations.

**[-auto-revert {true|false}] - Auto Revert**

Use this parameter to modify whether a data LIF is reverted automatically to its home node under certain circumstances. These circumstances would include startup, when the status of the management database changes to either master or secondary, and when the network connection is made. The default setting is `false` . If you set the value of this parameter to `true` , the load balancing migration capability of the data LIF is disabled (the `-allow-lb-migrate` parameter is set to `false` ).

Logical interfaces for SAN traffic do not support auto-revert. Thus, this parameter is always `false` on such interfaces.

**[-dns-zone {<zone-name>|none}] - Fully Qualified DNS Zone Name**

Use this parameter to modify the unique, fully qualified domain name of the DNS zone to which this data LIF belongs. You can associate a data LIF with a single DNS zone. All data LIFs included in a zone must be on the same Vserver. If you do not specify a value for this parameter, the data LIF is created with the value `none` .

**[-listen-for-dns-query {true|false}] - DNS Query Listen Enable**

Use this parameter to specify if the LIF has to listen for DNS queries. The default value for this parameter is true.

**[-allow-lb-migrate {true|false}] - (DEPRECATED)-Load Balancing Migrate Allowed (privilege: advanced)**

This parameter has been deprecated and may be removed in a future version of Data ONTAP.

Use this parameter to modify whether or not load balancing migration is enabled for this data LIF. The default value of this parameter is `false` . If you set the value of this parameter to `true` , the automatic revert capability of the data LIF is disabled (the `-auto-revert` parameter is set to `false` ). Also, data LIFs that migrate as a result of load balancing adhere to network interface failover rules.

During times when a LIF is hosting active NFSv4, CIFS, or NRV connections, load balancing based LIF migrations between nodes will be temporarily disabled.

**[-lb-weight {load|0..100}] - Load Balanced Weight (privilege: advanced)**

Use this parameter to modify the load balancing weight of the data LIF. A valid load balancing weight is any integer between 1 and 100. If you specify the same load balancing weight for all data LIFs in a DNS zone, client requests are uniformly distributed, similar to round-robin DNS. A data LIF with a low load balancing weight is made available for client requests less frequently than one that has a high load balancing weight.

**[-failover-group <failover-group>] - Failover Group Name**

Use this parameter to modify the name of the failover group to associate with the network interface. Manage failover groups using the `network interface failover-groups` command. Each broadcast domain has a default failover group which is created by the system automatically and has the same name as the broadcast domain. The failover group associated with the broadcast domain includes all ports in the broadcast domain. A logical interface's failover group is set to the failover group of the home port's broadcast domain by default, but this value can be modified.

Logical interfaces for SAN protocols do not support failover. Thus, this parameter cannot be specified for such interfaces.

**[-comment <text>] - Comment**

Use this parameter to modify the comment associated with the LIF.

**[-force-subnet-association <true>] - Force the LIF's Subnet Association**

This command will fail if the IP address falls within the address range of a named subnet. Set this to true to acquire the address from the named subnet and assign the subnet to the LIF.

**[-is-dns-update-enabled {true|false}] - Is Dynamic DNS Update Enabled?**

If this parameter is set to *true* , then dynamic DNS update is sent to the DNS server for the particular LIF entry if dynamic DNS updates are enabled for the corresponding Vserver. This field is set to *true* by default for both IPv4 and IPv6 LIFs. DNS Update is not supported on LIFs not configured with either the NFS or CIFS protocol.

### Examples

The following example modifies a LIF named datalif1 on a logical server named vs0. The LIF's netmask is modified to 255.255.255.128.

```
cluster1::> network interface modify -vserver vs0 -lif datalif1 -netmask
255.255.255.128
```

### Related Links

- lun portset show
- lun portset remove
- network interface revert
- system services firewall policy show
- system services firewall policy modify

## network interface rename

Rename a logical interface

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

Use the `network interface rename` command to change the name of an existing logical interface.

### Parameters

**-vserver <vserver> - Vserver Name**

Use this parameter to specify the Vserver on which the logical interface to rename is located.

**-lif <lif-name> - Logical Interface Name**

Use this parameter to specify the name of the logical interface to rename.

**-newname <text> - The new name for the interface**

Use this parameter to specify the new name of the logical interface. For iSCSI and FC LIFs, the name cannot be more than 254 characters.

### Examples

The following example renames a cluster logical interface named cluslif1 to cluslif4 on a Vserver named vs0.

```
cluster1::> network interface rename -vserver vs0 -lif cluslif1 -newname
cluslif4
```

## network interface revert

Revert a logical interface to its home port

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

The `network interface revert` command reverts a logical interface that is not currently on its home port to its home port, assuming that the home node and port are both operational. A logical interface's home port is specified when the logical interface is created. Determine a logical interface's home port by using the network interface show command.

When you revert a cluster logical interface, you must do so from the local node.

On some cloud platforms, this operation might perform changes to the external route tables.

### Parameters

**`-vserver <vserver>` - Vserver Name**

Use this parameter to specify the Vserver on which the logical interface to be reverted is located.

**`-lif <lif-name>` - Logical Interface Name**

Use this parameter to specify the logical interface that is to be reverted.

Logical interfaces for SAN protocols are always home. Thus, this command has no effect on such interfaces. The same applies to logical interfaces for NAS protocols that are already home.

### Examples

The following example returns any logical interfaces that are not currently on their home ports to their home ports.

```
cluster1::> network interface revert -vserver * -lif *
```

### Related Links

- network interface show

## network interface show

Display logical interfaces

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

### Description

The `network interface show` command displays information about logical interfaces.

Running the command with the `-failover` parameter displays information relevant to logical interface failover rules.

Running the command with the `-status` parameter displays information relevant to logical interface operational status.

Running the command with the `-by-ipspace` parameter displays information relevant to logical interfaces on a specific IPspace.

See the examples for more information.

You can specify additional parameters to display only information that matches those parameters. For example, to display information only about logical interfaces whose operational status is down, run the command with the `-status-oper down` parameter.

### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>,…` parameter, the command displays only the fields that you specify.

**| [-by-ipspace ]**

Use this parameter to display logical-interfaces sorted by IPspace and Vserver.

**| [-dns-zones ]**

Use this parameter to display logical-interfaces and whether the interface is associated with a Domain Name System (DNS) load balancing zone.

**| [-failover ]**

Use this parameter to display logical-interface failover information.

**| [-status ]**

Use this parameter to display detailed logical-interface status information.

**| [-instance ] }**

Use this parameter to display all the fields for the specified logical-interfaces.

**[-vserver <vserver>] - Vserver Name**

Use this parameter to display information only about logical interfaces on the Vserver you specify.

Use this parameter plus the `-lif` parameter to display detailed information only about the logical interface you specify.

**[-lif <lif-name>] - Logical Interface Name**

Use this parameter to display information only about logical interfaces that match the name you specify.

Use this parameter with the `-vserver` parameter to display detailed information only about the logical interface you specify.

**[-service-policy <text>] - Service Policy**

Use this parameter to display information only about logical interfaces that have the service policy you specify.

**[-services <LIF Service Name>,…] - Service List**

Use this parameter to display information only about logical interfaces that support all services in a comma-separated list of service names.

**[-role {undef|cluster|data|node-mgmt|intercluster|cluster-mgmt}] - (DEPRECATED)-Role**

This parameter has been deprecated and may be removed in a future version of ONTAP. Use either the `-service-policy` or `-services` parameter instead.

Use this parameter to display information only about logical interfaces that are associated with network ports that have the role you specify.

**[-data-protocol {nfs|cifs|iscsi|fcp|fcache|none|fc-nvme|s3}] - Data Protocol**

Use this parameter to display information only about logical interfaces that have the enabled data protocols you specify.

**[-address <IP Address>] - Network Address**

Use this parameter to display information only about logical interfaces that match the IP address or address range you specify.

**[-netmask <IP Address>] - Netmask**

Use this parameter to display information only about logical interfaces that have the netmask you specify.

**[-netmask-length <integer>] - Bits in the Netmask**

Use this parameter to display information only about logical interfaces with a netmask that has the number of bits you specify.

**[-is-vip <true>] - Is VIP LIF**

Use this parameter to display information only about logical interfaces that are VIP LIFs or not as you specify.

**[-subnet-name <subnet name>] - Subnet Name**

Use this parameter to display the logical interfaces that matches the subnet name.

**[-home-node <nodename>] - Home Node**

Use this parameter to display information only about logical interfaces that have the home node you specify.

**[-home-port {<netport>|<ifgrp>}] - Home Port**

Use this parameter to display information only about logical interfaces that have the home port or interface group you specify.

**[-curr-node <nodename>] - Current Node**

Use this parameter to display information only about logical interfaces that are currently located on the node you specify.

**[-curr-port {<netport>|<ifgrp>}] - Current Port**

Use this parameter to display information only about logical interfaces that are currently located on the port or interface group you specify.

**[-status-oper {up|down}] - Operational Status**

Use this parameter to display information only about logical interfaces that have the operational status you specify.

**[-status-extended <text>] - Extended Status**

Use this parameter to display information only about logical interfaces that match the extended status that you specify.

**[-numeric-id <integer>] - Numeric ID (privilege: advanced)**

Use this parameter to display information only about logical interfaces with the numeric ID (or range of IDs) you specify. The numeric ID is an integer that identifies the logical interface in the cluster.

**[-is-home {true|false}] - Is Home**

Use this parameter to display information only about logical interfaces that are (true) or are not (false) currently located on their home node and port.

**[-status-admin {up|down}] - Administrative Status**

Use this parameter to display information only about logical interfaces that have the administrative status you specify.

**[-failover-policy {system-defined|local-only|sfo-partner-only|disabled|broadcast-domain-wide}] - Failover Policy**

Use this parameter to display information only about logical interfaces that use the failover policy you specify.

**[-firewall-policy <policy>] - Firewall Policy**

Use this parameter to display information only about logical interfaces that use the firewall policies you specify.

**[-auto-revert {true|false}] - Auto Revert**

Use this parameter to display information only about logical interfaces that have auto-revert setting you specify.

**[-sticky {true|false}] - Sticky Flag (privilege: advanced)**

Use this parameter to display information only about logical interfaces that are "sticky". A sticky logical interface is one that has been manually migrated to another node and is not subject to auto-revert settings. A sticky logical interface remains at the migrated location until it is manually reverted or until it fails over to another node.

**[-dns-zone {<zone-name>|none}] - Fully Qualified DNS Zone Name**

Use this parameter to display information only about logical interfaces in the specified DNS zone.

**[-listen-for-dns-query {true|false}] - DNS Query Listen Enable**

Use this parameter to display information only about logical interfaces that have the DNS query listen value you specify.

**[-allow-lb-migrate {true|false}] - (DEPRECATED)-Load Balancing Migrate Allowed (privilege: advanced)**

> This parameter has been deprecated and may be removed in a future version of Data ONTAP.

Use this parameter to display information only about logical interfaces for which load balancing migration is activated (true) or not activated (false).

**[-lb-weight {load|0..100}] - Load Balanced Weight (privilege: advanced)**

Use this parameter to display information only about logical interfaces that have the load balancing weight you specify.

**[-failover-group <failover-group>] - Failover Group Name**

Use this parameter to display information only about logical interfaces that are in the failover group you specify. Logical interfaces in the same failover group are capable of failing over to the same set of ports.

**[-wwpn <text>] - FCP WWPN**

Use this parameter to display information only about logical interfaces that have the Fibre Channel Protocol port identifier (World Wide Port Name) you specify.

**[-address-family {ipv4|ipv6|ipv6z}] - Address family**

Use this parameter to view the address family that is in use on the interface. Only IPv4 and IPv6 non-zoned addresses can be configured. Configuration of IPv6z addresses is not allowed.

**[-comment <text>] - Comment**

Use this parameter to display information only about logical interfaces that have the comment you specify.

**[-ipspace <IPspace>] - IPspace of LIF**

Use this parameter to display information only about logical interfaces on the IPspace you specify.

**[-is-dns-update-enabled {true|false}] - Is Dynamic DNS Update Enabled?**

Use this parameter to display information only about logical interfaces that have (true) or do not have (false) dynamic DNS updates enabled for them.

**[-probe-port <integer>] - Probe-port for Cloud Load Balancer**

Use this parameter display the probe-port for the logical interface in the Azure environment.

**[-broadcast-domain <text>] - Broadcast Domain**

Use this parameter to display the broadcast domain that contains the home port of the logical interface.

**[-vserver-type <vserver type>] - Vserver Type**

Use this parameter to display information only about logical interfaces owned by Vservers of the specified type.

### Examples

The following example displays general information about all logical interfaces.

```
cluster1::> network interface show
            Logical    Status     Network            Current       Current
Is
Vserver     Interface  Admin/Oper Address/Mask       Node          Port
Home
----------- ---------- ---------- ------------------ ------------- -------
----
cluster1
            cluster_mgmt
                       up/up      192.0.2.1/192      node0         e0M
true
            node0_mgmt1
                       up/up      192.0.2.2/192      node0         e0M
true
            node1_mgmt1
                       up/up      192.0.2.3/192      node1         e0M
true
Cluster
            node0_clus1
                       up/up      192.0.2.66/192     node0         e0a
true
            node0_clus2
                       up/up      192.0.2.67/192     node0         e0b
true
            node1_clus1
                       up/up      192.0.2.68/192     node1         e0a
true
            node1_clus2
                       up/up      192.0.2.69/192     node1         e0b
true
```

The following example displays failover information about all logical interfaces.

```
cluster1::> network interface show -failover
         Logical         Home                  Failover        Failover
Vserver  Interface       Node:Port             Policy          Group
-------- --------------- --------------------- ---------------
---------------
cluster1
         cluster_mgmt    node0:e0M             broadcast-domain-wide
                                                               Default
                         Failover Targets: node0:e0M,
                                           node0:e0d,
                                           node0:e0e,
                                           node0:e0f,
                                           node1:e0M,
                                           node1:e0d,
                                           node1:e0e,
                                           node1:e0f
         node0_mgmt1     node0:e0M             local-only      Default
                         Failover Targets: node0:e0M,
                                           node0:e0d,
                                           node0:e0e,
                                           node0:e0f
         node1_mgmt1     node1:e0M             local-only      Default
                         Failover Targets: node1:e0M,
                                           node1:e0d,
                                           node1:e0e,
                                           node1:e0f
Cluster
         node0_clus1     node0:e0a             local-only      Cluster
                         Failover Targets: node0:e0a,
                                           node0:e0b
         node0_clus2     node0:e0a             local-only      Cluster
                         Failover Targets: node0:e0b,
                                           node0:e0a
         node1_clus1     node1:e0a             local-only      Cluster
                         Failover Targets: node1:e0a,
                                           node1:e0b
         node1_clus2     node1:e0a             local-only      Cluster
                         Failover Targets: node1:e0b,
                                           node1:e0a
```

### network interface start-cluster-check

(DEPRECATED)-Use the network interface check cluster-connectivity start command

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

### Description

> This command has been deprecated and may be removed in a future version of ONTAP. Use the `network interface check cluster-connectivity start` command instead.

The `network interface start-cluster-check` command initiates an accessibility check from every logical interface to every aggregate. Automatic checks run periodically, but this command manually initiates a check immediately.

This command produces no direct output. Any errors encountered during the check are reported in the event log. See the event log show command for more information.

### Examples

This example shows an execution of this command, with all parameters and output.

```
cluster1::> network interface start-cluster-check
```

### Related Links

- event log show

## network interface capacity show

Display the number of IP data LIFs capable of being configured on the cluster.

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

The `network interface capacity show` command displays the number of IP LIFs of role *`data`* supported on the cluster, as well as the number of IP LIFs of role *`data`* currently configured on the cluster.

> The number of IP LIFs of role *`data`* that are supported on a node depends on the hardware platform and the Cluster's Data ONTAP version. If one or more nodes in the cluster cannot support additional LIFs, then none of the nodes in the cluster can support additional LIFs.

### Examples

The following displays the IP data LIF capacity.

```
cluster1::> network interface capacity show
      IP Data LIF      IP Data LIF
  Supported Limit            Count
-----------------  ---------------
             1024              256
```

## network interface capacity details show

Display details about the IP data LIFs capable of being configured on each node.

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

### Description

The `network interface capacity details show` command displays the number of IP LIFs of role *`data`* that can be configured on each node, the number of IP data LIFs of role *`data`* that are supported on each node, and the number of IP data LIFs of role *`data`* that are configured to be homed on each node.

> The number of IP LIFs of role *`data`* that are supported on a node depends on the hardware platform and the Cluster's Data ONTAP version. If one or more nodes in the cluster cannot support additional LIFs, then none of the nodes in the cluster can support additional LIFs.

### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-node {<nodename>|local}] - Node Name (privilege: advanced)**

Use this parameter to specify the node for which to obtain data LIF capacity.

**[-capacity-for-node <integer>] - Number of IP data LIFs that can be configured on the node (privilege: advanced)**

This parameter specifies the number of IP LIFs of role *`data`* that can be configured on the node at the currently running Data ONTAP version. To view the version of a node, use the cluster image show command.

**[-limit-for-node <integer>] - Number of IP data LIFs that are supported on the node (privilege: advanced)**

This parameter specifies the number of IP LIFs of role *`data`* that are supported on the node at the current effective cluster version (ECV). To view the version of a node, use the cluster image show command.

**[-count-for-node <integer>] - Number of IP data LIFs that are assigned to the node (privilege: advanced)**

This parameter specifies the number of IP LIFs of role *`data`* currently configured to be homed on the node. To view LIFs homed on this node, use the network interface show -home-node command.

### Examples

The following displays the IP data LIF capacity.

```
cluster1::> network interface capacity details show
                        IP Data LIF       IP Data LIF      IP Data LIF
             Node          Capacity   Supported Limit            Count
-----------------  ---------------  ---------------  ---------------
            node1               512              512              128
            node2               512              512              128
```

**Related Links**

- cluster image show
- network interface show

## network interface check cluster-connectivity show

### Display the Cluster Connectivity Log

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

#### Description

The `network interface check cluster-connectivity show` command displays the cluster check result log.

#### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>,…` parameter, the command displays only the fields that you specify.

**| [-n ] (privilege: advanced)**

Selects IP addresses instead of LIF names.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-node {<nodename>|local}] - Node (privilege: advanced)**

Selects cluster health records that match the specified node name.

**[-record-number <integer>] - Record Number (privilege: advanced)**

Selects the cluster health records that match specified record number.

**[-date-time {MM/DD/YYYY HH:MM:SS [{+|-}hh:mm]}] - Record Date (privilege: advanced)**

Selects the cluster health records that match the specified date and time.

**[-src-name <text>] - Source LIF Name (privilege: advanced)**

Selects the cluster health records that match the specified source LIF name.

**[-src-addr <IP Address>] - Source IP Address (privilege: advanced)**

Selects the cluster health records that match the specified source IP address.

**[-dst-name <text>] - Destination LIF Name (privilege: advanced)**

Selects the cluster health records that match the specified destination LIF name.

**[-dst-addr <IP Address>] - Destination IP Address (privilege: advanced)**

Selects the cluster health records that match the specified destination IP address.

**[-loss {None|Partial|All}] - Packet Loss (privilege: advanced)**

Selects the cluster health records that match the specified loss value.

#### Examples

The following example displays the cluster connectivity records.

```
cluster1::> network interface check cluster-connectivity show
                                  Source          Destination     Packet
Node   Date                       LIF             LIF             Loss
------ -------------------------- --------------- ---------------
-----------
node1
       8/26/2020 10:35:36 -04:00  node1_clus_1    node1_clus_2    none
       8/26/2020 10:35:37 -04:00  node1_clus_1    node2_clus_1    none
       8/26/2020 10:35:38 -04:00  node1_clus_1    node2_clus_2    none
       8/26/2020 10:35:39 -04:00  node1_clus_2    node1_clus_1    none
       8/26/2020 10:35:40 -04:00  node1_clus_2    node2_clus_1    none
       8/26/2020 10:35:41 -04:00  node1_clus_2    node2_clus_2    none
node2
       8/26/2020 10:35:36 -04:00  node2_clus_1    node2_clus_2    none
       8/26/2020 10:35:37 -04:00  node2_clus_1    node1_clus_1    none
       8/26/2020 10:35:38 -04:00  node2_clus_1    node1_clus_2    none
       8/26/2020 10:35:39 -04:00  node2_clus_2    node2_clus_1    none
       8/26/2020 10:35:40 -04:00  node2_clus_2    node1_clus_1    none
       8/26/2020 10:35:41 -04:00  node2_clus_2    node1_clus_2    none
12 entries were displayed.
```

## network interface check cluster-connectivity start

Start the cluster check function

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

#### Description

The `network interface check cluster-connectivity start` command initiates an accessibility check from every logical interface to every aggregate. Automatic checks run periodically, but this command

manually initiates a check immediately.

This command produces no direct output. Any errors encountered during the check are reported via the cluster-check show and the event log. See the network interface check cluster-connectivity show and event log show commands for more information.

### Parameters

**`-node {<nodename>|local}` - Node (privilege: advanced)**

This optionally selects the node on which to initiate the accessibility check. If no node is specified, the accessibility check is initiated on the local node.

### Examples

This example shows an execution of this command with no parameter which will initiate the accessibility check on the local node.

```
cluster1::> network interface check cluster-connecivity start
```

### Related Links

- network interface check cluster-connectivity show
- event log show

## network interface check failover show

Discover if any LIFs might become inaccessible during a node outage, due to over-provisioning

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

This command identifies logical interfaces (LIFs) at risk of becoming inaccessible if their hosting nodes were to experience an outage. The source-nodes parameter is the only required input.

The tuple <destination-nodes, vserver-name, lif-name> is sufficient to uniquely identify a record in the returned listing. All fields other than source-nodes can be filtered on in the usual fashion. There are some examples of this filtering below.

### Parameters

**`{ [-fields <fieldname>,…]`**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**`| [-instance ] }`**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-destination-nodes <nodename>,…] - Set Of Nodes Over Capacity**

Use this parameter to display the nodes an at-risk LIF or LIFs could fail over to.

**[-vserver-name <vserver>] - Vserver Name**

Use this parameter to display only LIFs on the Vserver you specify.

**[-lif-name <lif-name>] - LIF Name**

Use this parameter to display at-risk information only about the LIF or LIFs whose name you specify.

**-source-nodes <nodename>,… - Nodes Going Down**

List of nodes to test. At-risk LIFs currently hosted on these nodes will be identified. The list should contain no more than half the nodes in the cluster.

**[-over-amount <integer>] - Amount Capacity Exceeded**

Use this parameter to select only at-risk LIFs associated with a set of destination nodes whose amount over capacity matches the number you specify.

Note that the number of LIFs considered to be at risk may be higher than the actual amount over capacity a given set of nodes is. Once a given set of nodes is determined to be potentially over capacity, all LIFs whose set of failover target nodes is an exact match are marked as at risk. The amount over capacity is an upper bound on the number LIFs which could become unhosted if LIFs were to fail over in a random order, each to a target randomly selected from that LIF's configured failover targets.

**[-failover-group <failover-group>] - Failover Group Name**

Use this parameter to display information only about at-risk LIFs whose failover-group you specify.

**[-failover-policy {system-defined|local-only|sfo-partner-only|disabled|broadcast-domain-wide}] - Failover Policy**

Use this parameter to display information only about at-risk LIFs whose failover-policy you specify.

### Examples

The following example shows all the at-risk LIFs for a specific two-node outage in a six-node cluster.

```
cluster1::> network interface check failover show -source-nodes
node1,node5

Destination Nodes: node2, node3, node4, node6
Amount Over Capacity: 2
Vserver             Logical Interface    Failover Group   Failover Policy
------------------  -------------------  ----------------
---------------------
vs0                 data1                Default          broadcast-
domain-wide
                    data2                Default          broadcast-
domain-wide
                    data3                Default          broadcast-
domain-wide
vs1                 data1                Custom_Name      broadcast-
domain-wide

Destination Nodes: node2
Amount Over Capacity: 1
Vserver             Logical Interface    Failover Group   Failover Policy
------------------  -------------------  ----------------
---------------------
vs0                 data6                Default          sfo-partner-only
vs1                 data7                Default          sfo-partner-only
```

The following example shows the same two-node outage scenario, but now with some filtering applied to the results.

```
cluster1::> network interface check failover show -source-nodes
node1,node5 -destination-nodes node2,node3,node4,node6 -failover-group
Def*

Destination Nodes: node2, node3, node4, node6
Amount Over Capacity: 2
Vserver             Logical Interface    Failover Group   Failover Policy
------------------  -------------------  ----------------
---------------------
vs0                 data1                Default          broadcast-
domain-wide
                    data2                Default          broadcast-
domain-wide
                    data3                Default          broadcast-
domain-wide
```

## network interface dns-lb-stats show

Show the DNS load-balancer stats for this node

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *advanced* privilege level.

### Description

The `network interface dns-lb-stats show` command displays the statistics for DNS load-balancing lookups for the zones belonging to the specified Vserver. These statistics represent the data for the Vserver on the local node. The following counts can be seen in the statistics output:

- success-count : Number of successful lookups.
- authoritative-count : Number of authoritative answers sent.
- nonauthoritative-count : Number of non authoritative answers sent.
- rr-set-missing-count : Number of times the RR set was missing.
- domain-missing-count : Number of times the domain was not be found.
- failure-count : Number of failed lookups.
- dropped-count : Number of lookups dropped.

### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-vserver <vserver>] - Vserver (privilege: advanced)**

Use this parameter to display DNS load-balancer statistics only for the specified Vservers.

**[-zone <text>] - DNS Zone (privilege: advanced)**

Use this parameter to display DNS load-balancer statistics only for the specified DNS zones.

**[-success-count <integer>] - Successful Lookup Count (privilege: advanced)**

Use this parameter to display DNS load-balancer statistics only for the specified number of successful lookups.

**[-authoritative-count <integer>] - Authoritative Answer Count (privilege: advanced)**

Use this parameter to display DNS load-balancer statistics only for the specified number of authoritative answers sent.

**[-nonauthoritative-count <integer>] - Non Authoritative Answer Count (privilege: advanced)**

Use this parameter to display DNS load-balancer statistics only for the specified number of non-authoritative answers sent.

**[-rr-set-missing-count <integer>] - RR Set Missing Count (privilege: advanced)**

Use this parameter to display DNS load-balancer statistics only for the specified number of times the RR set was missing.

**[-domain-missing-count <integer>] - Name Missing Count (privilege: advanced)**

Use this parameter to display DNS load-balancer statistics only for the specified number of times the domain was not found.

**[-failure-count <integer>] - Failed Lookup Count (privilege: advanced)**

Use this parameter to display DNS load-balancer statistics only for the specified number of failed lookups.

**[-dropped-count <integer>] - Dropped Count (privilege: advanced)**

Use this parameter to display DNS load-balancer statistics only for the specified number of dropped lookups.

### Examples

The following example displays stats for the zone "x.com".

```
cluster1::> network interface dns-lb-stats show -zone x.com
Vserver    DNS Zone           SUCCESS    AUTH  NOAUTH  NORR  NODOM  FAILED
DROP
---------  -----------------  -------   -----  ------ -----  -----  ------
-----
vs2
           x.com        5       5       0     0      0       0     0
```

## network interface failover-groups add-targets

Add failover targets to a failover group

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

The `network interface failover-groups add-targets` command enables you to add a list of failover targets such as network ports, interface groups, or VLANs to an existing logical interface failover group.

### Parameters

**-vserver <vserver> - Vserver Name**

Use this parameter to specify the name of the Vservers from which this failover group is accessible.

**-failover-group <text> - Failover Group Name**

Use this parameter to specify the failover group that you want to extend.

**-targets [node>:<port],… - Failover Targets**

Use this parameter to specify the failover targets such as network ports, interface groups, or VLANs you wish to add to the failover group.

#### Examples

This example shows the failover group "clyde" being extended to include additional failover targets.

```
cluster1::> network interface failover-group add-targets -vserver vs1
-failover-group clyde -targets xena1:e0c, xena1:e0d-100, xena2:a0a
```

## network interface failover-groups create

Create a new failover group

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

The `network interface failover-groups create` command creates a grouping of failover targets for logical interfaces on one or more nodes. Use this command to add a new network port or interface group to an existing failover group.

Interfaces for SAN protocols do not support failover. Such interfaces are not valid failover targets.

#### Parameters

**-vserver <vserver> - Vserver Name**

Use this parameter to specify the name of the Vservers from which this failover group is accessible.

**-failover-group <text> - Failover Group Name**

Use this parameter to specify the name of the logical interface failover group that you want to create.

**-targets [node>:<port],… - Failover Targets**

Use this parameter to specify the list of failover targets (network ports, interface groups, or VLANs on a node) belonging to this failover group.

#### Examples

The following example shows how to create a failover group named failover-group_2 containing ports e1e and e2e on node Xena.

```
cluster1::> network interface failover-groups create -vserver vs0
-failover-group failover-group_2 -targets xena:e1e,xena:e2e
```

## network interface failover-groups delete

Delete a failover group

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

The `network interface failover-groups delete` command deletes a logical interface failover group.

### Parameters

**`-vserver <vserver>` - Vserver Name**

Use this parameter to specify the name of the Vservers from which this failover group is accessible.

**`-failover-group <text>` - Failover Group Name**

Use this parameter to specify the name of the logical interface failover group to be deleted.

### Examples

The following example shows how to delete a failover group named failover-group_2.

```
cluster1::> network interface failover-groups delete -vserver vs1
-failover-group failover-group_2
```

## network interface failover-groups modify

Modify a failover group

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

The `network interface failover-groups modify` command enables you modify the list of network ports, interface groups, or VLANs belonging to an existing logical interface failover group. The specified list will overwrite the existing list of network ports, interface groups, and VLANs currently belonging to the logical interface failover group.

### Parameters

**`-vserver <vserver>` - Vserver Name**

Use this parameter to specify the name of the Vserver(s) from which this failover group is accessible.

**`-failover-group <text>` - Failover Group Name**

Use this parameter to specify the failover group that you want to modify.

**`[-targets [node>:<port],…]` - Failover Targets**

Use this parameter to specify the network ports, interface groups, or VLANs you wish to now belong to the failover group.

#### Examples

This example shows the failover group "clyde" being modified to now contain the specified network ports.

```
cluster1::> network interface failover-group modify -vserver vs1 -failover
-group clyde -targets xena1:e0c, xena1:e0d-100, xena2:a0a
```

## network interface failover-groups remove-targets

### Remove failover targets from a failover group

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

The `network interface failover-groups remove-targets` command enables you to specify a list of failover targets such as network ports, interface groups, or VLANs to be removed from an existing logical interface failover group.

#### Parameters

**`-vserver <vserver>` - Vserver Name**

Use this parameter to specify the name of the Vserver(s) from which this failover group is accessible.

**`-failover-group <text>` - Failover Group Name**

Use this parameter to specify the failover group that you want to remove failover targets from.

**`-targets [node>:<port],…` - Failover Targets**

Use this parameter to specify the failover targets such as network ports, interface groups, or VLANs you wish to remove from the failover group.

#### Examples

This example shows the failover targets xena1:e0c and xena1:e0d-100 being removed from the failover group "clyde".

```
cluster1::> network interface failover-group remove-targets -vserver vs1
-failover-group clyde -targets xena1:e0c, xena1:e0d-100, xena2:a0a
```

## network interface failover-groups rename

### Rename a logical interface failover Group

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

The `network interface failover-groups rename` command enables you to rename an existing logical interface failover group.

### Parameters

**-vserver <vserver> - Vserver Name**

Use this parameter to specify the name of the Vservers from which this failover group is accessible.

**-failover-group <text> - Failover Group Name**

Use this parameter to specify the failover group that you want to rename.

**-new-failover-group-name <text> - New name**

Use this parameter to specify the new name of the failover group.

### Examples

This example shows the failover group "clusterwide" being renamed "clyde".

```
cluster1::> network interface failover-group rename -failover -vserver vs1
-failover-group clusterwide -new-failover-group-name clyde
```

## network interface failover-groups show

Display logical interface failover groups

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

### Description

The `network interface failover-groups show` command displays information about logical interface failover groups.

### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-vserver <vserver>] - Vserver Name**

Use this parameter to display information only about the logical interface failover groups that have the target Vserver you specify.

**[-failover-group <text>] - Failover Group Name**

Use this parameter to display information only about the logical interface failover groups you specify.

**[-targets [node>:<port],…] - Failover Targets**

Use this parameter to display information only about the logical interface failover groups that have the failover target (physical port, interfacde group, or VLAN) you specify.

**[-broadcast-domain <Broadcast Domain>] - Broadcast Domain**

Use this parameter to display information only about the logical interface failover groups that have the broadcast domain you specify.

#### Examples

The following example displays information about all logical interface failover groups on a two node cluster.

```
cluster1::> network interface failover-groups show
                                  Failover
Vserver          Group            Targets
---------------- ----------------
--------------------------------------------
Cluster
                 Cluster
                                  node1:e1a, node1:e2a,
                                  node2:e1a, node2:e2a,
cluster1
                 Default
                                  node1:e0M, node1:e0a,
                                  node1:e0b, node1:e0c,
                                  node1:e0d, node2:e0M,
                                  node2:e0a, node2:e0b,
                                  node2:e0c, node2:e0d
```

## network interface lif-weights show

Show the load-balancer LIF weights

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *advanced* privilege level.

#### Description

The `network interface lif-weights show` command displays the weights assigned to each LIF in a DNS load-balancing zone in a Vserver.

#### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-vserver <vserver>] - Vserver (privilege: advanced)**

Use this parameter to display information only for the specified Vservers.

**[-zone <text>] - DNS Zone (privilege: advanced)**

Use this parameter to display information only for the specified DNS zones.

**[-address <IP Address>] - Network Address (privilege: advanced)**

Use this parameter to display information only for the specified IP addresses.

**[-weight <double>] - Load Balancer Weight (privilege: advanced)**

Use this parameter to display information only for the specified load balancer weights

#### Examples

The following example displays LIF weights for vserver "vs1".

```
cluster1::> network interface lif-weights show -vserver vs1
                                Network
Vserver    DNS Zone             Address         Weight
---------  -------------------  ------------  -------
vs1
           a.com                4.4.4.4       12.4206
           x.com                1.1.1.1       12.4206
           x.com                10.72.46.236  12.4206
3 entries were displayed.
```

## network interface service show

### Display available interface services

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

The `network interface service show` command displays available services for IP LIFs and the TCP or UDP ports that each service listens on. The ports listed in this table correspond to well-known ports that each service can be expected to open a listening socket. Services that do not listen for ingress connections are presented with an empty port list.

#### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>,…` parameter, the command displays only the fields that you specify.

**| [-restrictions ] (privilege: advanced)**

The `network interface service show-restrictions` command displays available services for IP LIFs and usage restrictions for each service. The restrictions determine which LIFs are permitted to use each service and what restrictions the service implies for the LIFs that do use it.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-service <LIF Service Name>] - Service Name**

Selects services that match the specified service name.

**[-ports <integer>,…] - TCP/UDP Port Numbers**

Selects services that contain all IP ports in the specified list.

**[-protocol-ports <text>,…] - Protocol: Port Numbers**

Selects services that match the `<protocol>:<port>` combination.

**[-vserver-policy <svc_vserver_policy>] - Vserver Restrictions**

Selects services that match a specific vserver restriction.

**[-failover-policy <svc_failover_policy>] - Failover Restrictions**

Selects services that match a specific interface failover restriction.

### Examples

The following example displays the built-in services.

```
cluster1::> network interface service show
Service                 Protocol:Port
----------------------- -------------
intercluster-core       tcp:11104
                        tcp:11105
management-bgp          tcp:179

2 entries were displayed.
```

## network interface service-policy add-service

Add an additional service entry to an existing service policy

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

### Description

The `network interface service-policy add-service` command adds an additional service to an existing service-policy. When an allowed address list is specified, the list applies to only the service being added. Existing services included in this policy are not impacted.

### Parameters

**-vserver <vserver name> - Vserver (privilege: advanced)**

Use this parameter to specify the name of the Vserver of the service policy to be updated.

**-policy <text> - Policy Name (privilege: advanced)**

Use this parameter to specify the name of service policy to be updated.

**-service <LIF Service Name> - Service entry to be added (privilege: advanced)**

Use this parameter to specify the name of service to be added to the existing service policy.

**[-allowed-addresses <IP Address/Mask>,…] - Allowed address ranges for the service (privilege: advanced)**

Use this parameter to specify a list of subnet masks for addresses that are allowed to access this service. Use the value `0.0.0.0/0` to represent the wildcard IPv4 address and `::/0` to represent the wildcard IPv6 address.

### Examples

The following example shows the addition of a service to an existing service policy.

```
cluster1::> network interface service-policy show -vserver cluster1
Vserver   Policy                    Service: Allowed Addresses
--------- ------------------------
--------------------------------------
cluster1
          default-intercluster      intercluster-core: 0.0.0.0/0
                                    management-https: 0.0.0.0/0
default-management         management-core: 0.0.0.0/0
                                    management-autosupport: 0.0.0.0/0
                                    management-ssh: 0.0.0.0/0
                                    management-https: 0.0.0.0/0
default-route-announce     management-bgp: 0.0.0.0/0

3 entries were displayed.

cluster1::> network interface service-policy add-service -vserver cluster1
-policy default-intercluster -service management-ssh

cluster1::> network interface service-policy show -vserver cluster1
Vserver   Policy                    Service: Allowed Addresses
--------- ------------------------
--------------------------------------
cluster1
          default-intercluster      intercluster-core: 0.0.0.0/0
                                    management-ssh: 0.0.0.0/0
                                    management-https: 0.0.0.0/0
default-management         management-core: 0.0.0.0/0
                                    management-autosupport: 0.0.0.0/0
                                    management-ssh: 0.0.0.0/0
                                    management-https: 0.0.0.0/0
default-route-announce     management-bgp: 0.0.0.0/0

3 entries were displayed.
```

## network interface service-policy clone

Clone an existing network service policy

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

### Description

The `network interface service-policy clone` command creates a new service policy that includes the same services and allowed addresses as an existing policy. Once the new service policy has been created, it can be modified as necessary without impacting the original policy.

### Parameters

**-vserver <vserver name> - Vserver (privilege: advanced)**

Use this parameter to specify the name of the Vserver of the service policy to be cloned.

**-policy <text> - Policy Name (privilege: advanced)**

Use this parameter to specify the name of the service policy to be cloned.

**-target-vserver <vserver name> - Vserver Name (privilege: advanced)**

Use this parameter to specify the name of the vserver on which the new service policy should be created.

**-target-policy <text> - New Service Policy Name (privilege: advanced)**

Use this parameter to specify the name of the new service policy.

### Examples

The following example shows the cloning of a service policy.

```
cluster1::> network interface service-policy show -vserver
cluster1,ipspace1
Vserver   Policy                    Service: Allowed Addresses
--------- -------------------------
----------------------------------------
cluster1
          custom1                   intercluster-core: 0.0.0.0/0
                                    management-core: 0.0.0.0/0
                                    management-ssh: 0.0.0.0/0
default-intercluster       intercluster-core: 0.0.0.0/0
                                    management-https: 0.0.0.0/0
default-management         management-core: 0.0.0.0/0
                                    management-autosupport: 0.0.0.0/0
                                    management-ssh: 0.0.0.0/0
                                    management-https: 0.0.0.0/0
default-route-announce     management-bgp: 0.0.0.0/0

ipspace1
          default-intercluster       intercluster-core: 0.0.0.0/0
                                    management-https: 0.0.0.0/0
default-management         management-core: 0.0.0.0/0
                                    management-autosupport: 0.0.0.0/0
                                    management-ssh: 0.0.0.0/0
                                    management-https: 0.0.0.0/0
default-route-announce     management-bgp: 0.0.0.0/0

7 entries were displayed.

cluster1::> network interface service-policy clone -vserver cluster1
```

```
-policy custom1 -target-vserver ipspace1 -target-policy custom2

cluster1::> network interface service-policy show -vserver
cluster1,ipspace1
Vserver   Policy                    Service: Allowed Addresses
--------- -------------------------
----------------------------------------
cluster1
          custom1                   intercluster-core: 0.0.0.0/0
                                    management-core: 0.0.0.0/0
                                    management-ssh: 0.0.0.0/0
default-intercluster       intercluster-core: 0.0.0.0/0
                                    management-https: 0.0.0.0/0
default-management         management-core: 0.0.0.0/0
                                    management-autosupport: 0.0.0.0/0
                                    management-ssh: 0.0.0.0/0
                                    management-https: 0.0.0.0/0
default-route-announce     management-bgp: 0.0.0.0/0

ipspace1
          custom2                   intercluster-core: 0.0.0.0/0
                                    management-core: 0.0.0.0/0
                                    management-ssh: 0.0.0.0/0
default-intercluster       intercluster-core: 0.0.0.0/0
                                    management-https: 0.0.0.0/0
default-management         management-core: 0.0.0.0/0
                                    management-autosupport: 0.0.0.0/0
                                    management-ssh: 0.0.0.0/0
                                    management-https: 0.0.0.0/0
default-route-announce     management-bgp: 0.0.0.0/0

8 entries were displayed.
```

## network interface service-policy create

Create a new service policy

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

### Description

The `network interface service-policy create` command creates a new service policy with a list of included services. LIFs can reference this policy to control the list of services that they are able to transport on their network. Services can represent applications accessed by a LIF as well as applications served by this cluster.

### Parameters

**-vserver <vserver name> - Vserver**

Use this parameter to specify the name of the Vserver on which the service policy will be created.

**-policy <text> - Policy Name**

Use this parameter to specify the name of service policy to be created.

**[-services <LIF Service Name>,…] - Included Services**

Use this parameter to specify a list of services that should be included in this policy.

**[-allowed-addresses <IP Address/Mask>,…] - Allowed Addresses**

Use this parameter to specify a list of subnet masks for addresses that are allowed to access the services in this policy. Use the value `0.0.0.0/0` to represent the wildcard IPv4 address and `::/0` to represent the wildcard IPv6 address.

### Examples

The following example shows the creation of a service policy with no initial services.

```
cluster1::> network interface service-policy create -vserver cluster1
-policy empty

cluster1::> network interface service-policy show -vserver cluster1
Vserver   Policy                    Service: Allowed Addresses
--------- -------------------------
-----------------------------------------
cluster1
          default-intercluster      intercluster-core: 0.0.0.0/0
                                    management-https: 0.0.0.0/0
default-management         management-core: 0.0.0.0/0
                                    management-autosupport: 0.0.0.0/0
                                    management-ssh: 0.0.0.0/0
                                    management-https: 0.0.0.0/0
default-route-announce     management-bgp: 0.0.0.0/0
empty                      -

4 entries were displayed.
```

The following example shows the creation of a new service policy with a specified service list.

```
cluster1::> network interface service-policy create -vserver cluster1
-policy custom -services intercluster-core,management-ssh

cluster1::> network interface service-policy show -vserver cluster1
Vserver   Policy                    Service: Allowed Addresses
--------- -------------------------
----------------------------------------
cluster1
          custom                    intercluster-core: 0.0.0.0/0
                                    management-ssh: 0.0.0.0/0
default-intercluster       intercluster-core: 0.0.0.0/0
                                    management-https: 0.0.0.0/0
default-management         management-core: 0.0.0.0/0
                                    management-autosupport: 0.0.0.0/0
                                    management-ssh: 0.0.0.0/0
                                    management-https: 0.0.0.0/0
default-route-announce     management-bgp: 0.0.0.0/0
empty                      -

5 entries were displayed.
```

## network interface service-policy delete

Delete an existing service policy

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

### Description

The `network interface service-policy delete` command deletes an existing service policy.

### Parameters

**`-vserver <vserver name>` - Vserver**
Use this parameter to specify the name of the Vserver of the service policy to be deleted.

**`-policy <text>` - Policy Name**
Use this parameter to specify the name of the service policy to be deleted.

### Examples

The following example shows the deletion of a service policy.

```
cluster1::> network interface service-policy show -vserver cluster1
Vserver   Policy                       Service: Allowed Addresses
--------- --------------------------
--------------------------------------
cluster1
          custom                       intercluster-core: 0.0.0.0/0
                                       management-ssh: 0.0.0.0/0
                                       management-https: 0.0.0.0/0
default-intercluster       intercluster-core: 0.0.0.0/0
                                       management-https: 0.0.0.0/0
default-management         management-core: 0.0.0.0/0
                                       management-autosupport: 0.0.0.0/0
                                       management-ssh: 0.0.0.0/0
                                       management-https: 0.0.0.0/0
default-route-announce     management-bgp: 0.0.0.0/0

4 entries were displayed.

cluster1::> network interface service-policy delete -vserver cluster1
-policy custom

cluster1::> network interface service-policy show -vserver cluster1
Vserver   Policy                       Service: Allowed Addresses
--------- --------------------------
----------------------------------------
cluster1
          default-intercluster       intercluster-core: 0.0.0.0/0
                                       management-https: 0.0.0.0/0
default-management         management-core: 0.0.0.0/0
                                       management-autosupport: 0.0.0.0/0
                                       management-ssh: 0.0.0.0/0
                                       management-https: 0.0.0.0/0
default-route-announce     management-bgp: 0.0.0.0/0

3 entries were displayed.
```

## network interface service-policy modify-service

Modify a service entry in an existing service policy

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

### Description

The `network interface service-policy modify-service` command modifies the policy-specific attributes of a service that is already included in a particular service policy. Other services in the policy are not

impacted by the change.

### Parameters

**`-vserver <vserver name>` - Vserver (privilege: advanced)**

Use this parameter to specify the name of the Vserver of the service policy to be updated.

**`-policy <text>` - Policy Name (privilege: advanced)**

Use this parameter to specify the name of service policy to be updated.

**`-service <LIF Service Name>` - Service entry to be modified (privilege: advanced)**

Use this parameter to specify the name of service to be updated.

**`-allowed-addresses <IP Address/Mask>,…` - Allowed address ranges for the service (privilege: advanced)**

Use this parameter to specify a list of subnet masks for addresses that are allowed to access this service. Use the value `0.0.0.0/0` to represent the wildcard IPv4 address and `::/0` to represent the wildcard IPv6 address.

### Examples

The following example shows the modification of a service on an existing service policy.

```
cluster1::> network interface service-policy show -vserver cluster1
Vserver   Policy                    Service: Allowed Addresses
--------- --------------------------
---------------------------------------
cluster1
          default-intercluster      intercluster-core: 0.0.0.0/0
                                    management-https: 0.0.0.0/0
default-management         management-core: 0.0.0.0/0
                                    management-autosupport: 0.0.0.0/0
                                    management-ssh: 0.0.0.0/0
                                    management-https: 0.0.0.0/0
default-route-announce     management-bgp: 0.0.0.0/0

3 entries were displayed.

cluster1::> network interface service-policy modify-service -vserver
cluster1 -policy default-management -service management-ssh -allowed
-addresses 10.1.0.0/16

cluster1::> network interface service-policy show -vserver cluster1
Vserver   Policy                    Service: Allowed Addresses
--------- --------------------------
---------------------------------------
cluster1
          default-intercluster      intercluster-core: 0.0.0.0/0
                                    management-https: 0.0.0.0/0
default-management         management-core: 0.0.0.0/0
                                    management-autosupport: 0.0.0.0/0
                                    management-ssh: 10.1.0.0/16
                                    management-https: 0.0.0.0/0
default-route-announce     management-bgp: 0.0.0.0/0

3 entries were displayed.
```

## network interface service-policy remove-service

Remove a service entry from an existing service policy

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

### Description

The `network interface service-policy remove-service` command removes an individual service from an existing service policy. Other services in the the policy are not impacted by the change.

### Parameters

**-vserver <vserver name> - Vserver (privilege: advanced)**

Use this parameter to specify the name of the Vserver of the service policy to be updated.

**-policy <text> - Policy Name (privilege: advanced)**

Use this parameter to specify the name of service policy to be updated.

**-service <LIF Service Name> - Service entry to be removed (privilege: advanced)**

Use this parameter to specify the name of service to be removed from the existing service policy.

### Examples

The following example shows the removal of a service from an existing service policy.

```
cluster1::> network interface service-policy show -vserver cluster1
Vserver   Policy                    Service: Allowed Addresses
--------- -------------------------
----------------------------------------
cluster1
          default-intercluster      intercluster-core: 0.0.0.0/0
                                    management-https: 0.0.0.0/0
default-management         management-core: 0.0.0.0/0
                                    management-autosupport: 0.0.0.0/0
                                    management-ssh: 0.0.0.0/0
                                    management-https: 0.0.0.0/0
default-route-announce     management-bgp: 0.0.0.0/0

3 entries were displayed.

cluster1::> network interface service-policy remove-service -vserver
cluster1 -policy default-management -service management-autosupport

cluster1::> network interface service-policy show -vserver cluster1
Vserver   Policy                    Service: Allowed Addresses
--------- -------------------------
----------------------------------------
cluster1
          default-intercluster      intercluster-core: 0.0.0.0/0
                                    management-https: 0.0.0.0/0
default-management         management-core: 0.0.0.0/0
                                    management-ssh: 0.0.0.0/0
                                    management-https: 0.0.0.0/0
default-route-announce     management-bgp: 0.0.0.0/0

3 entries were displayed.
```

## network interface service-policy rename

Rename an existing network service policy

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

### Description

The `network interface service-policy rename` command assigns a new name to an existing service policy without disrupting the LIFs using the policy.

### Parameters

**`-vserver <vserver name>` - Vserver (privilege: advanced)**

Use this parameter to specify the name of the Vserver of the service policy to be renamed.

**`-policy <text>` - Policy Name (privilege: advanced)**

Use this parameter to specify the name of the service policy to be renamed.

**`-new-name <text>` - New Service Policy Name (privilege: advanced)**

Use this parameter to specify the new name for the service policy.

### Examples

The following example shows the renaming of a service policy.

```
cluster1::> network interface service-policy show -vserver cluster1
Vserver   Policy                       Service: Allowed Addresses
--------- --------------------------
---------------------------------------
cluster1
          custom                       intercluster-core: 0.0.0.0/0
                                       management-autosupport: 0.0.0.0/0
                                       management-ssh: 0.0.0.0/0
                                       management-https: 0.0.0.0/0
default-intercluster       intercluster-core: 0.0.0.0/0
                                       management-https: 0.0.0.0/0
default-management         management-core: 0.0.0.0/0
                                       management-autosupport: 0.0.0.0/0
                                       management-ssh: 0.0.0.0/0
                                       management-https: 0.0.0.0/0
default-route-announce     management-bgp: 0.0.0.0/0

4 entries were displayed.

cluster1::> network interface service-policy rename -vserver cluster1
-policy custom -new-name system

cluster1::> network interface service-policy show -vserver cluster1
Vserver   Policy                       Service: Allowed Addresses
--------- --------------------------
---------------------------------------
cluster1
          system                       intercluster-core: 0.0.0.0/0
                                       management-autosupport: 0.0.0.0/0
                                       management-ssh: 0.0.0.0/0
                                       management-https: 0.0.0.0/0
default-intercluster       intercluster-core: 0.0.0.0/0
                                       management-https: 0.0.0.0/0
default-management         management-core: 0.0.0.0/0
                                       management-autosupport: 0.0.0.0/0
                                       management-ssh: 0.0.0.0/0
                                       management-https: 0.0.0.0/0
default-route-announce     management-bgp: 0.0.0.0/0

4 entries were displayed.
```

### network interface service-policy restore-defaults

Restore default settings to a service policy

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

### Description

The `network interface service-policy restore-defaults` command restores a built-in service-policy to its original state. The default list of services replaces any customizations that have been applied by an administrator. All included services will be updated to use the default allowed address list.

### Parameters

**`-vserver <vserver name>` - Vserver (privilege: advanced)**

Use this parameter to specify the name of the Vserver of the service policy to be restored.

**`-policy <text>` - Policy Name (privilege: advanced)**

Use this parameter to specify the name of the service policy to be restored.

### Examples

The following example shows the restoration of a service policy's default settings.

```
cluster1::> network interface service-policy show -vserver cluster1
Vserver   Policy                    Service: Allowed Addresses
--------- --------------------------
---------------------------------------
cluster1
          default-intercluster      intercluster-core: 10.1.0.0/16
                                    management-ssh: 10.1.0.0/16
                                    management-https: 10.1.0.0/16
default-management         management-core: 0.0.0.0/0
                                    management-autosupport: 0.0.0.0/0
                                    management-ssh: 0.0.0.0/0
                                    management-https: 0.0.0.0/0
default-route-announce     management-bgp: 0.0.0.0/0

3 entries were displayed.

cluster1::> network interface service-policy restore-defaults -vserver
cluster1 -policy default-intercluster

cluster1::> network interface service-policy show -vserver cluster1
Vserver   Policy                    Service: Allowed Addresses
--------- --------------------------
---------------------------------------
cluster1
          default-intercluster      intercluster-core: 0.0.0.0/0
                                    management-https: 0.0.0.0/0
default-management         management-core: 0.0.0.0/0
                                    management-autosupport: 0.0.0.0/0
                                    management-ssh: 0.0.0.0/0
                                    management-https: 0.0.0.0/0
default-route-announce     management-bgp: 0.0.0.0/0

3 entries were displayed.
```

## network interface service-policy show

Display existing service policies

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

### Description

The `network interface service-policy show` command displays existing service policies.

### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>,…` parameter, the command displays only the fields that you specify.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-vserver <vserver name>] - Vserver**

Selects service policies that match the specified vserver name.

**[-policy <text>] - Policy Name**

Selects service policies that match the specified service policy name.

**[-services <LIF Service Name>,…] - Included Services**

Selects service policies that contain all services in the specified list of service names.

**[-service-allowed-addresses <text>,…] - Service: Allowed Addresses**

Selects service policies that contain all "`<service>:<allowed-addresses>`" in the specified list of addresses.

### Examples

The following example displays the built-in service policies.

```
cluster1::> network interface service-policy show -vserver cluster1
Vserver   Policy                    Service: Allowed Addresses
--------- -------------------------
----------------------------------------
cluster1
          default-intercluster      intercluster-core: 0.0.0.0/0
                                    management-https: 0.0.0.0/0
default-management         management-core: 0.0.0.0/0
                                    management-autosupport: 0.0.0.0/0
                                    management-ssh: 0.0.0.0/0
                                    management-https: 0.0.0.0/0
default-route-announce     management-bgp: 0.0.0.0/0

3 entries were displayed.
```

# network ipspace commands

## network ipspace create

Create a new IPspace

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

IPspaces are distinct IP address spaces in which Storage Virtual Machines (SVMs) reside. The "Cluster" IPspace and "Default" IPspace are created by default. You can create more custom IPspaces when you need your SVMs to have overlapping IP addresses, or you need more control over networking configurations for cluster peering. Please reference the "Network Management Guide" for the limit of how many custom IPspaces are supported on your system..

### Parameters

**`-ipspace <IPspace>` - IPspace name**

The name of the IPspace to be created.

- The name must contain only the following characters: A-Z, a-z, 0-9, ".", "-" or "_".
- The first character of each label, delimited by ".", must be one of the following characters: A-Z or a-z.
- The last character of each label, delimited by ".", must be one of the following characters: A-Z, a-z or 0-9.
- The maximum supported length is 47 characters.
- The system reserves the following names: "all", "local" and "localhost".
- The system provides the following IPspaces: "Cluster" and "Default".

### Examples

The following example creates IPspace "ips1".

```
cluster1::> network ipspace create -name ips1
```

## network ipspace delete

Delete an IPspace

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

Delete an IPspace that contains no ports or Vservers.

### Parameters

**`-ipspace <IPspace>` - IPspace name**

The name of the IPspace to be deleted. If the IPspace is associated with one or more logical-interfaces, you must delete them before you can delete the IPspace.

### Examples

The following example deletes the IPspace "ips1".

```
cluster1::> network ipspace delete -ipspace ips1
```

## network ipspace rename

### Rename an IPspace

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

Rename an IPspace.

#### Parameters

**`-ipspace <IPspace>` - IPspace name**
The name of the IPspace to be renamed.

**`-new-name <IPspace>` - New Name**
The new name for the IPspace.

- The name must contain only the following characters: A-Z, a-z, 0-9, ".", "-" or "_".
- The first character of each label, delimited by ".", must be one of the following characters: A-Z or a-z.
- The last character of each label, delimited by ".", must be one of the following characters: A-Z, a-z or 0-9.
- The maximum supported length is 47 characters.
- The system reserves the following names: "all", "cluster", "local" and "localhost".

#### Examples

The following example renames IPspace "ips1" to "ips2".

```
cluster1::> network ipspace rename -ipspace ips1 -new-name ips2
```

## network ipspace show

### Display IPspace information

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

Display network IPspaces.

#### Parameters

**{ [-fields <fieldname>,…]**

Specify the fields to be displayed for each IPspace.

**| [-instance ] }**

Display all parameters of the IPspace objects.

**[-ipspace <IPspace>] - IPspace name**

Display the names of the IPspaces.

**[-ports [node>:<port],…] - Ports**

The list of network ports assigned to each IPspace.

**[-broadcast-domains <Broadcast Domain>,…] - Broadcast Domains**

The list of broadcast domains that belong to the IPspace.

**[-vservers <vserver name>,…] - Vservers**

The list of Vservers assigned to each IPspace.

### Examples

The following example displays general information about IPspaces.

```
cluster1::> network ipspace show
IPspace             Vserver List                  Broadcast Domains
------------------- ----------------------------
----------------------------
Cluster
                    Cluster                       -
Default
                    cluster1, vs1, vs2            br1, br2, br3
2 entries were displayed.
```

# network ndp commands

## network ndp default-router delete-all

Delete default routers on a given IPspace

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

### Description

The `network ndp default-router delete-all` command deletes default router lists from the specified IPspace.

#### Parameters

**`-ipspace <IPspace>` - IPspace Name (privilege: advanced)**

Use this parameter to specify the IPspace where the default routers are to be deleted.

#### Examples

The following example deletes default routers from IPspace ips1.

```
cluster1::*> network ndp default-router delete-all -ipspace ips1
```

## network ndp default-router show

### Display default routers

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

#### Description

The `network ndp default-router show` command displays Neighbor Discovery Protocol (NDP) default routers learned on a specified port.

#### Parameters

**`{ [-fields <fieldname>,…]`**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**`| [-instance ] }`**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**`[-node {<nodename>|local}]` - Node (privilege: advanced)**

Displays the NDP default routers from the specified node.

**`[-ipspace <IPspace>]` - IPspace name (privilege: advanced)**

Displays the NDP default routers from the specified IPspace.

**`[-port {<netport>|<ifgrp>}]` - Port (privilege: advanced)**

Displays the NDP default routers from the specified port.

**`[-router-addr <IP Address>]` - Router Address (privilege: advanced)**

Displays the default routers that have the specified IPv6 addresses.

**`[-flag {none|managed-address-DHCPv6|other-DHCPv6}]` - Flag (privilege: advanced)**

Displays the default routers that have the specified flag. The flag indicates whether addresses are available via DHCPv6 or other configuration information is available via DHCPv6.

**[-expire-time {[<integer>d][<integer>h][<integer>m][<integer>s]|never|expired}] - Expire Time (privilege: advanced)**

Displays the default routers that have the specified expire time.

#### Examples

The following example displays NDP default routers on local port e0f.

```
cluster1::*> network ndp default-router show -port e0f -node local

Node: node1
IPspace: Default
Port     Router Address              Flag            Expire Time
-------- -------------------------- --------------  --------------
e0f      fe80::5:73ff:fea0:107      none            0d0h23m9s
```

## network ndp neighbor create

### Create a static NDP neighbor entry

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *advanced* privilege level.

#### Description

The `network ndp neighbor create` command creates a static Neighbor Discovery Protocol (NDP) neighbor entry within a Vserver.

#### Parameters

**-vserver <vserver name> - Vserver Name (privilege: advanced)**

Use this parameter to specify the Vserver on which the NDP neighbor is to be created.

**-neighbor <IP Address> - Neighbor Address (privilege: advanced)**

Use this parameter to specify the neighbor's IPv6 address.

**-mac-address <MAC Address> - MAC Address (privilege: advanced)**

Use this parameter to specify the neighbor's MAC address.

#### Examples

The following example creates a NDP neighbor entry within Vserver vs0.

```
cluster1::*> network ndp neighbor create -vserver vs0 -neighbor 20:20::20
-mac-address 10:10:10:0:0:1
```

## network ndp neighbor delete

### Delete a static NDP neighbor entry

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *advanced* privilege level.

#### Description

The `network ndp neighbor delete` command deletes a static Neighbor Discovery Protocol (NDP) neighbor from a Vserver.

#### Parameters

**`-vserver <vserver name>` - Vserver Name (privilege: advanced)**

Use this parameter to specify the Vserver on which the NDP neighbor is to be deleted.

**`-neighbor <IP Address>` - Neighbor Address (privilege: advanced)**

Use this parameter to specify the neighbor's IPv6 address.

#### Examples

The following example deletes a NDP neighbor entry within Vserver vs0.

```
cluster1::*> network ndp neighbor delete -vserver vs0 -neighbor 20:20::20
```

## network ndp neighbor show

### Display static NDP neighbor entries

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *advanced* privilege level.

#### Description

The `network ndp neighbor show` command displays a group of static Neighbor Discovery Protocol (NDP) neighbors within one or more Vservers. You can view static NDP neighbors within specified Vservers, neighbors with specified IPv6 address, and neighbors with specified MAC address.

#### Parameters

**{ `[-fields <fieldname>,…]`**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| `[-instance ]` }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**`[-vserver <vserver name>]` - Vserver Name (privilege: advanced)**

Displays the static NDP neighbors that have the specified Vserver as their origin.

**[-neighbor <IP Address>] - Neighbor Address (privilege: advanced)**

Displays the static NDP neighbors that have the specified IPv6 address.

**[-mac-address <MAC Address>] - MAC Address (privilege: advanced)**

Displays the static NDP neighbors that have the specified MAC address.

#### Examples

The following example displays all of the static NDP neighbors configured on Vserver vs0.

```
cluster1::*> network ndp neighbor show -vserver vs0
Vserver            Neighbor                 MAC Address
------------------ ------------------------ -------------------
vs0
                   10:10::10                04:04:04:04:04:04
                   20:20::20                01:01:01:01:01:01
2 entries were displayed.
```

## network ndp neighbor active-entry delete

Delete active neighbor entry from a System or Admin Vserver

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

#### Description

The `network ndp neigbhor active-entry delete` command deletes a Network Discovery Protocol (NDP) neighbor entry on the specified port from a given Vserver's subnet group.

#### Parameters

**-node {<nodename>|local} - Node (privilege: advanced)**

Use this parameter to specify the node on which the neighbor entry is to be deleted.

**-vserver <vserver> - System or Admin Vserver Name (privilege: advanced)**

Use this parameter to specify the System or Admin Vserver on which the neighbor entry is to be deleted.

**-subnet-group <IP Address/Mask> - Subnet Group (privilege: advanced)**

Use this parameter to specify the subnet group from which the neighbor entry is to be deleted.

**-neighbor <IP Address> - Neighbor (privilege: advanced)**

Use this parameter to specify the IPv6 address of the neighbor entry which is to be deleted.

**-port {<netport>|<ifgrp>} - Port (privilege: advanced)**

Use this parameter to specify the port on which the neighbor entry is to be deleted.

#### Examples

The following example deletes a neighbor entry from the Admin Vserver cluster1:

```
cluster1::*> network ndp neighbor active-entry delete -vserver cluster1
-node local -subnet-group ::/0 -neighbor fe80:4::5:73ff:fea0:107 -port e0d
```

## network ndp neighbor active-entry show

Display active neighbor entries organized by Vserver

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

#### Description

The `network ndp neighbor active-entry show` command displays Network Discovery Protocol (NDP) neighbor cache entries on one or more nodes. You can view ndp neighbors within specified nodes and within specified System or Admin Vservers.

#### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| [-verbose ] (privilege: advanced)**

Displays the expire time, state, is-router, and probe count fields.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-node {<nodename>|local}] - Node (privilege: advanced)**

Displays the NDP neighbors from the specified node.

**[-vserver <vserver>] - System or Admin Vserver Name (privilege: advanced)**

Displays the NDP neighbors from the specified System or Admin Vserver. Data and Node Vservers do not have dynamic NDP neighbors.

**[-subnet-group <IP Address/Mask>] - Subnet Group (privilege: advanced)**

Displays the NDP neighbors in the specified subnet group.

**[-neighbor <IP Address>] - Neighbor (privilege: advanced)**

Displays the NDP neighbors that have the specified IPv6 address.

**[-port {<netport>|<ifgrp>}] - Port (privilege: advanced)**

Displays the NDP neighbors on the specified port.

**[-mac-address <MAC Address>] - MAC Address (privilege: advanced)**

Displays the NDP neighbors have the specified MAC address.

**[-expire-time {[<integer>d][<integer>h][<integer>m][<integer>s]|never|expired}] - Expire Time (privilege: advanced)**

Displays the NDP neighbors have the specified expire time.

**[-state {<nostate|incomplete|reachable|stale|delay|probe|unknown>}] - State (privilege: advanced)**

Displays the NDP neighbors in the specified state.

**[-is-router {true|false}] - Is Router (privilege: advanced)**

Displays the NDP neighbor which is a router.

**[-probe-count <integer>] - Probe Count (privilege: advanced)**

Displays the NDP neighbors with the specified probe count. Probe count is the number of times that this neighbor's MAC address has been queried.

**[-is-static {true|false}] - Is Static (privilege: advanced)**

Displays the NDP neighbors which are statically configured.

#### Examples

The following example displays NDP neighbors on the Admin Vserver cluster1:

```
cluster1::*> network ndp neighbor active-entry show -vserver cluster1

Node: node1
Vserver: cluster1
Subnet Group: ::/0
Neighbor                      MAC Address          Port
----------------------------- -------------------- --------
fe80:4::5:73ff:fea0:107       00:05:73:a0:01:07    e0d
fe80:4::226:98ff:fe0c:b6c1    00:26:98:0c:b6:c1    e0d
fe80:4::4255:39ff:fe25:27c1   40:55:39:25:27:c1    e0d
3 entries were displayed.
```

## network ndp prefix delete-all

Delete IPv6 prefixes on a given IPspace

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

#### Description

The `network ndp prefix delete-all` command deletes all prefixes learned from the specified IPspace.

#### Parameters

**`-ipspace <IPspace>` - IPspace Name (privilege: advanced)**

Use this parameter to specify the IPspace where the IPv6 prefixes are to be deleted.

#### Examples

The following example deletes all IPv6 prefixes within IPspace ips1.

```
cluster1::*> network ndp prefix delete-all -ipspace ips1
```

## network ndp prefix show

Display IPv6 prefixes

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

#### Description

The `network ndp prefix show` command displays IPv6 prefixes on one or more nodes.

#### Parameters

**{ `[-fields <fieldname>,…]`**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| `[-verbose ]` (privilege: advanced)**

Displays the valid-lifetime, preferred-lifetime, origin and advertising-router fields.

**| `[-instance ]` }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**`[-node {<nodename>|local}]` - Node (privilege: advanced)**

Displays the IPv6 prefixes from the specified node.

**`[-ipspace <IPspace>]` - IPspace name (privilege: advanced)**

Displays the IPv6 prefixes from the specified IPspace.

**`[-port {<netport>|<ifgrp>}]` - Port (privilege: advanced)**

Displays the IPv6 prefixes on the specified port.

**`[-prefix <IP Address/Mask>]` - Prefix (privilege: advanced)**

Displays the IPv6 prefixes with the specified prefix value.

**`[-flag {none|on-link|autonomous|on-link-autonomous}]` - Flag (privilege: advanced)**

Displays the IPv6 prefixes with the specified flag. The flag indicates whether a prefix is on-link and whether it can be used in autonomous address configuration.

**[-valid-lifetime {<unsigned integer>|infinity}] - Valid Lifetime (privilege: advanced)**

Displays the IPv6 prefixes having the specified valid lifetime in seconds.

**[-preferred-lifetime {<unsigned integer>|infinity}] - Preferred Lifetime (privilege: advanced)**

Displays the IPv6 prefixes having the specified preferred lifetime in seconds.

**[-expire-time {[<integer>d][<integer>h][<integer>m][<integer>s]|never|expired}] - Expire Time (privilege: advanced)**

Displays the IPv6 prefixes having the specified expire time.

**[-origin {router-advertise|renumber-request|static|kernel|unknown}] - Origin of the Prefix (privilege: advanced)**

Displays the IPv6 prefixes with the specified origin.

**[-advertising-router <IP Address>,…] - Router that Advertised the Prefix (privilege: advanced)**

Displays the IPv6 prefixes which are propagated by the specified router list.

#### Examples

The following example displays IPv6 prefixes on port e0f.

```
cluster1::*> network ndp prefix show -port e0f -node local

Node: node1
IPspace: Default
Port      Prefix                    Flag               Expire Time
--------- ------------------------- ------------------ -------------
e0f       fd20:8b1e:b255:814e::/64  on-link-autonomous 29d23h56m48s
```

# network options commands

## network options cluster-health-notifications modify

cluster health notification options

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

This command enables or disables cluster health notifications on the specified node.

#### Parameters

**-node {<nodename>|local} - Node**

This parameter specifies the node for which the cluster health notification status will be modified.

**[-enabled {true|false}] - Cluster Health Notifications Enabled**

Setting this parameter to *true* enables cluster health notification. Setting it to *false* disables cluster health notification.

### Examples

The following example modifies the cluster health notification status for a node:

```
cluster1::> network options cluster-health-notifications modify -node
node1 -enabled true
```

## network options cluster-health-notifications show

Display cluster health notification options

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

The `network options cluster-health-notifications show` command displays whether the node's cluster health notifications are enabled.

### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-node {<nodename>|local}] - Node**

This parameter specifies the node for which the cluster health notification status will be displayed.

**[-enabled {true|false}] - Cluster Health Notifications Enabled**

Selects the entries that match this parameter value.

### Examples

The following example displays the cluster health notification status for a node:

```
cluster1::> network options cluster-health-notifications show -node node1
                                  Node: node1
 Cluster Health Notifications Enabled: true
```

## network options detect-switchless-cluster modify

Modify the status of switchless cluster detection

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

### Description

This command enables or disables the automatic detection of a switchless cluster. A switchless cluster consists of two nodes where the cluster ports are directly connected without a switch between them.

### Parameters

**`[-enabled {true|false}]` - Enable Switchless Cluster Detection (privilege: advanced)**

This parameter specifies whether switchless cluster detection is enabled or not. Setting this parameter to *`true`* enables switchless cluster detection.

### Examples

```
The following example enables switchless cluster detection:
        cluster1::*> network options detect-switchless-cluster modify
-enabled true
```

## network options detect-switchless-cluster show

Display the status of switchless cluster detection

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

### Description

The network options detect-switchless-cluster show command displays whether switchless cluster detection is enabled.

### Examples

```
The following example displays whether switchless cluster detection is
enabled:
      cluster1::*> network options detect-switchless-cluster show
      Enable Detect Switchless Cluster: true
```

## network options ipv6 modify

Modify IPv6 options

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

This command sets the state of IPv6 options for the cluster.

#### Parameters

**`[-enabled <true>]` - IPv6 Enabled**

Setting this parameter to *`true`* enables IPv6 for the cluster. IPv6 cannot be disabled once it is enabled for the cluster. Call technical support for guidance regarding disabling IPv6.

**`[-is-ra-processing-enabled {true|false}]` - Router Advertisement (RA) Processing Enabled**

Setting this parameter to *`true`* enables cluster to process IPv6 router advertisements. Setting it to *`false`* disables router advertisement processing by the cluster.

#### Examples

```
The following example enables IPv6 for the cluster:
    cluster1::> network options ipv6 modify -enabled true

The following example enables IPv6 Router Advertisement processing for the
cluster:
    cluster1::> network options ipv6 modify -is-ra-processing-enabled true

The following example disables IPv6 Router Advertisement processing for
the
cluster:
    cluster1::> network options ipv6 modify -is-ra-processing-enabled
false
```

## network options ipv6 show

### Display IPv6 options

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

This command displays the current state of IPv6 options for the cluster.

#### Examples

```
cluster1::> network options ipv6 show

                                IPv6 Enabled: false
Router Advertisement (RA) Processing Enabled: false
```

## network options load-balancing modify

Modify load balancing algorithm

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

### Description

This command sets the state of geometric mean algorithm for load balancing

### Parameters

**`[-enable {true|false}]` - Geometric Mean Algorithm for load balancing (privilege: advanced)**

Setting this parameter to *`true`* enables the geometric mean algorithm for load balancing. Setting it to *`false`* disables the geometric mean algorithm for the cluster.

### Examples

```
The following example will enable the geometric mean algorithm for load
      balancing.
      cluster1::> network options load-balancing modify -enable true
The following example will disable the geometric mean algorithm for load
      balancing.
      cluster1::> network options load-balancing modify -enable false
```

## network options load-balancing show

Display load balancing algorithm

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

### Description

This command displays the use of geometric mean load balancing algorithm.

### Examples

```
cluster1::> network options load-balancing show
Geometric Mean Algorithm for load balancing: false
```

## network options multipath-routing modify

Modify multipath-routing

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

#### Description

The `network options multipath-routing modify` command is used to modify cluster-wide multipath routing configuration.

#### Parameters

**`[-is-enabled {true|false}]` - Is Multipath Routing Enabled (privilege: advanced)**

This parameter specifies whether multipath routing configuration is enabled or not. Setting this parameter to `_ true_` enables multipath routing for all nodes in the cluster.

#### Examples

```
cluster1::> network options multipath-routing modify -is-enabled true
```

## network options multipath-routing show

Display multipath-routing

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

#### Description

The `network options multipath-routing show` command displays the multipath routing configuration for the cluster.

#### Examples

```
cluster1::> network options multipath-routing show
                Is Multipath Routing Enabled: false
```

## network options port-health-monitor disable-monitors

Disable one or more port health monitors

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

#### Description

This command disables the given port health monitors for the given IPspaces in the cluster.

#### Parameters

**`-ipspace <IPspace>` - IPspace Name (privilege: advanced)**

The name of the IPspace for which the specified port health monitors are disabled.

**`-health-monitors {l2-reachability|link-flapping|crc-errors|vswitch-link}` - List of Port Health Monitors to Disable (privilege: advanced)**

The port health monitors to disable.

#### Examples

The following example disables the "l2_reachability" health monitor for the "Default" IPspace.

The status of the "link_flapping" monitor is unaffected by the command.

```
cluster1::*> network options port-health-monitor show

IPspace         Enabled Port Health Monitors
-------------   ----------------------------
Cluster         l2_reachability,
                link_flapping
Default         l2_reachability,
                link_flapping
2 entries were displayed.

cluster1::*> network options port-health-monitor disableMonitors -ipspace
Default -health-monitors l2_reachability

cluster1::*> network options port-health-monitor show

IPspace         Enabled Port Health Monitors
-------------   ----------------------------
Cluster         l2_reachability,
                link_flapping
Default         link_flapping
2 entries were displayed.
```

## network options port-health-monitor enable-monitors

Enable one or more port health monitors

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

#### Description

This command enables the given port health monitors for the given IPspaces in the cluster.

#### Parameters

**`-ipspace <IPspace>` - IPspace Name (privilege: advanced)**

The name of the IPspace for which the specified port health monitors are enabled.

**`-health-monitors {l2-reachability|link-flapping|crc-errors|vswitch-link}` - List of Port Health Monitors to Enable (privilege: advanced)**

The port health monitors to enable. Upon enabling the *`l2_reachability`* health monitor, it runs in an

"unpromoted" state. While in this state, the monitor does not mark any ports as unhealthy due to the l2_reachability health check. The monitor is promoted in the "Cluster" IPspace when the "Cluster" broadcast domain is found to have passed the *l2_reachability* health check. An EMS event called "vifmgr.hm.promoted" event is generated when the health monitor is promoted for the IPspace.

#### Examples

The following example enables the "l2_reachability" health monitor for the "Default" IPspace:

The status of the "link_flapping" monitor is unaffected by the command.

```
cluster1::*> network options port-health-monitor show

IPspace         Enabled Port Health Monitors
-------------   ----------------------------
Cluster         l2_reachability,
                link_flapping
Default         link_flapping
2 entries were displayed.

cluster1::*> network options port-health-monitor enableMonitors -ipspace
Default -health-monitors l2_reachability

cluster1::*> network options port-health-monitor show

IPspace         Enabled Port Health Monitors
-------------   ----------------------------
Cluster         l2_reachability,
                link_flapping
Default         l2_reachability,
                link_flapping
2 entries were displayed.
```

## network options port-health-monitor modify

### Modify port health monitors configuration

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

#### Description

This command modifies the enabled port health monitors for the given IPspaces in the cluster.

#### Parameters

**`-ipspace <IPspace>` - IPspace Name (privilege: advanced)**

The name of the IPspace for which enabled port health monitors are modified.

**[-health-monitors {l2-reachability|link-flapping|crc-errors|vswitch-link}] - List of Enabled Port Health Monitors (privilege: advanced)**

All of the port health monitors that you want to enable. This command enables any port health monitors in this list that are currently disabled, and it disables any currently enabled monitors that are not in this list. Upon enabling the *l2_reachability* health monitor, it runs in an "unpromoted" state. While in this state, the monitor does not mark any ports as unhealthy due to the l2_reachability health check. The monitor is promoted in the "Cluster" IPspace when the "Cluster" broadcast domain is found to have passed the *l2_reachability* health check. An EMS event called "vifmgr.hm.promoted" event is generated when the health monitor is promoted for the IPspace.

### Examples

The following example modifies the port health monitor configuration of the "Default" IPspace such that only the "link_flapping" port health monitor is enabled. enabled for all IPspaces in the cluster.

Only the specified monitor is enabled after the modify command is issued.

```
cluster1::*> network options port-health-monitor show

IPspace         Enabled Port Health Monitors
-------------   ----------------------------
Cluster         l2_reachability,
                link_flapping
Default         l2_reachability,
                link_flapping
2 entries were displayed.

cluster1::*> network options port-health-monitor modify -ipspace Default
-health-monitors link_flapping

cluster1::*> network options port-health-monitor show

IPspace         Enabled Port Health Monitors
-------------   ----------------------------
Cluster         l2_reachability,
                link_flapping
Default         link_flapping
2 entries were displayed.
```

## network options port-health-monitor show

Display port health monitors configuration

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

### Description

This command displays the enabled port health monitors for the IPspaces in the cluster.

### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-ipspace <IPspace>] - IPspace Name (privilege: advanced)**

Displays the port health monitors that are enabled only for the given IPspace name.

**[-health-monitors {l2-reachability|link-flapping|crc-errors|vswitch-link}] - List of Enabled Port Health Monitors (privilege: advanced)**

Displays the IPspaces that have the given monitors enabled.

### Examples

The following example lists all port health monitors that are enabled for all IPspaces in the cluster.

```
cluster1::*> network options port-health-monitor show

IPspace         Enabled Port Health Monitors
-------------   ----------------------------
Cluster         l2_reachability,
                link_flapping
Default         l2_reachability,
                link_flapping
2 entries were displayed.
```

## network options send-soa modify

Modify Send SOA settings

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

### Description

This command sets the status of sending statement of authority record in the DNS response.

### Parameters

**[-enable {true|false}] - Enable sending SOA (privilege: advanced)**

Setting this parameter to *true* enables sending the statement of authority (SOA) record in the DNS response. Setting it to *false* disables sending the statement of authority (SOA) record in the DNS

response for the cluster.

#### Examples

```
The following example will enable the sending of statement of authority
(SOA)
      in the DNS response.
      cluster1::> network options send-soa modify -enable true
The following example will disable the sending of statement of authority
(SOA)
      in the DNS response.
      cluster1::> network options send-soa modify -enable false
```

## network options send-soa show

### Display Send SOA settings

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

#### Description

This command displays whether sending the statement of authority record (SOA) in the DNS response is enabled or not.

#### Examples

```
cluster1::> network options send-soa show
Enable sending SOA: true
```

## network options switchless-cluster modify

### Modify switchless cluster network options

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

#### Description

This command sets whether the cluster network is in switchless or switched mode. A switchless cluster is physically formed by connecting two nodes back-to-back, without a switch between them.

#### Parameters

**`[-enabled {true|false}]` - Enable Switchless Cluster (privilege: advanced)**

This parameter specifies whether the switchless cluster is enabled or not. Setting this parameter to *`true`* enables the switchless cluster.

#### Examples

```
The following example enables the switchless cluster:
        cluster1::*> network options switchless-cluster modify -enabled
true
```

### network options switchless-cluster show

Display switchless cluster network options

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

#### Description

The network options switchless-cluster show command displays the attributes of a switchless cluster.

#### Examples

```
The following example displays the attributes of the switchless cluster:
            cluster1::*> network options switchless-cluster show
            Enable Switchless Cluster: true
```

## network port commands

### network port delete

Delete a network port

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

#### Description

The `network port delete` command deletes a network port that is no longer physically present on the storage system.

#### Parameters

**`-node {<nodename>|local}` - Node**

This specifies the node on which the port is located.

**`-port {<netport>|<ifgrp>}` - Port**

This specifies the port to delete.

#### Examples

The following example deletes port e0c from a node named node0. The command works only when the port does not physically exist on the storage system.

```
cluster1::*> network port delete -node node0 -port e0c
```

## network port modify

### Modify network port attributes

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

The `network port modify` command enables you to change the maximum transmission unit (MTU) setting, autonegotiation setting, administrative duplex mode, and administrative speed of a specified network port.

The MTU of ports that belong to broadcast-domains must be updated through the `broadcast-domain modify` command.

Modification of a port's IPspace will only work before a node is added to a cluster, when the cluster version is below Data ONTAP 8.3, or when the node is offline. To change the IPspace of a port once the node is in a Data ONTAP 8.3 cluster, the port should be added to a broadcast-domain that belongs to that IPspace.

#### Parameters

**`-node {<nodename>|local}` - Node**

Use this parameter to specify the node on which the port is located.

**`-port {<netport>|<ifgrp>}` - Port**

Use this parameter to specify the port that you want to modify.

**`[-mtu <integer>]` - MTU**

The port's MTU setting. The default setting for ports in the "Cluster" IPspace is 9000 bytes. All other ports use a default value of 1500 bytes.

**`[-autonegotiate-admin {true|false}]` - Auto-Negotiation Administrative**

Whether the port uses Ethernet autonegotiation to determine the highest speed and duplex mode that the port and its endpoint can support. The default setting when you create a port is `true` .

**`[-duplex-admin {auto|half|full}]` - Duplex Mode Administrative**

The administrative setting for the port's duplex mode. This is the duplex mode that you prefer the port to use. Depending on network limitations, the operational value can be different from the administrative setting. The default setting when you create a port is `full` .

**`[-speed-admin {auto|10|100|1000|10000|100000|40000|25000}]` - Speed Administrative**

The administrative speed setting, in megabits per second. This is the speed setting that you prefer the port to use. Depending on network limitations, the operational value can be lower than the administrative setting.

**`[-flowcontrol-admin {none|receive|send|full}]` - Flow Control Administrative**

The administrative flow control setting of the port. this is the flow control setting that you prefer the port to use. Depending on network and port limitations, the operational value can be different from the administrative setting.

**[-up-admin {true|false}] - Up Administrative (privilege: advanced)**

The administrative state of the port. If set to `true` , the port is used if it is operational. If set to `false` , the port is configured down.

**[-ipspace <IPspace>] - IPspace Name**

Use this parameter to specify the IPspace the network port is assigned to. Modification of a port's IPspace will only work before a node is added to a cluster, when the cluster version is below Data ONTAP 8.3, or when the node is offline. To change the IPspace of a port once the node is in a Data ONTAP 8.3 cluster, the port should be added to a broadcast-domain that belongs to that IPspace. If there is an inconsistency between the broadcast-domain and IPspace, this parameter can be set to bring the IPspace into alignment with the broadcast-domain.

**[-ignore-health-status {true|false}] - Ignore Port Health Status (privilege: advanced)**

Use this parameter to specify that the system ignore network port health status of the specified port for the purpose of hosting a logical interface.

#### Examples

The following example modifies port e0a on a node named node0 not to use auto-negotiation, to preferably use half duplex mode, and to preferably run at 100 Mbps.

```
cluster1::> network port modify -node node0 -port e0a -autonegotiate-admin
false -duplex-admin half -speed-admin 100
```

## network port show-address-filter-info

Print the port's address filter information

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

The `network port show-address-filter-info` command displays information about the port's address filter.

#### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**-node <nodename> - Node**

Use this parameter to specify the node.

**-port {<netport>|<ifgrp>} - Port**

Use this parameter to specify the port. For example, e0c.

**[-num-total <integer>] - Total Number Of Entries**

Use this parameter to specify the total number of entries.

**[-num-used <integer>] - Number Of Used Entries**

Use this parameter to specify the number of used entries.

**[-used-entries <text>,…] - The Used Entries**

Use this parameter to list the used entries.

#### Examples

The following example displays information of the given port's address filter on the specified node of the cluster.

```
cluster1::*> network port show-address-filter-info -node local -port e0c

Node: node1
        Total Number       Number of
Port      of Address    Used Address     Used Address
Name  Filter Entries  Filter Entries       Filter Entries
----  --------------  --------------   ------------------
e0c             1328               3     U 0 a0 98 40 e 6
                                          M 1 80 c2 0 0 e
                                          M 1 0 5e 0 0 fb
```

## network port show

### Display network port attributes

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

#### Description

The `network port show` command displays information about network ports. The command output indicates any inactive links, and lists the reason for the inactive status.

Some parameters can have "administrative" and "operational" values. The administrative setting is the preferred value for that parameter, which is set when the port is created or modified. The operational value is the actual current value of that parameter. Administrative and operational settings are not shown for virtual ports, '-' will be displayed. Please see the physical port hosting the target virtual port for these values.

If the operational duplex mode and speed of a port cannot be determined (for instance, if the link is down), that port's status is listed as *undef* , meaning undefined. This is different from '-', meaning no value.

#### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| [-health ]**

Use this parameter to display detailed health information for the specified network ports.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-node {<nodename>|local}] - Node**

Selects the network ports that match this parameter value. Use this parameter with the `-port` parameter to select a port.

**[-port {<netport>|<ifgrp>}] - Port**

Selects the network ports that match this parameter value. If you do not use this parameter, the command displays information about all network ports.

**[-link {off|up|down}] - Link**

Selects the network ports that match this parameter value.

**[-mtu <integer>] - MTU**

Selects the network ports that match this parameter value.

**[-autonegotiate-admin {true|false}] - Auto-Negotiation Administrative**

Selects the network ports that match this parameter value.

**[-autonegotiate-oper {true|false}] - Auto-Negotiation Operational**

Selects the network ports that match this parameter value.

**[-duplex-admin {auto|half|full}] - Duplex Mode Administrative**

Selects the network ports that match this parameter value.

**[-duplex-oper {auto|half|full}] - Duplex Mode Operational**

Selects the network ports that match this parameter value.

**[-speed-admin {auto|10|100|1000|10000|100000|40000|25000}] - Speed Administrative**

Selects the network ports that match this parameter value.

**[-speed-oper {auto|10|100|1000|10000|100000|40000|25000}] - Speed Operational**

Selects the network ports that match this parameter value.

**[-flowcontrol-admin {none|receive|send|full}] - Flow Control Administrative**

Selects the network ports that match this parameter value.

**[-flowcontrol-oper {none|receive|send|full}] - Flow Control Operational**

Selects the network ports that match this parameter value.

**[-mac <MAC Address>] - MAC Address**

Selects the network ports that match this parameter value.

**[-up-admin {true|false}] - Up Administrative (privilege: advanced)**

Selects the network ports that match this parameter value.

**[-type {physical|if-group|vlan|vip}] - Port Type**

Selects the network ports that match this parameter value.

**[-ifgrp-node <nodename>] - Interface Group Parent Node**

Selects the network ports that match this parameter value.

**[-ifgrp-port {<netport>|<ifgrp>}] - Interface Group Parent Port**

Selects the network ports that match this parameter value.

**[-ifgrp-distr-func {mac|ip|sequential|port}] - Distribution Function**

Selects the network ports that match this parameter value.

**[-ifgrp-mode {multimode|multimode_lacp|singlemode}] - Create Policy**

Selects the network ports that match this parameter value.

**[-vlan-node <nodename>] - Parent VLAN Node**

Selects the network ports that match this parameter value.

**[-vlan-port {<netport>|<ifgrp>}] - Parent VLAN Port**

Selects the network ports that match this parameter value.

**[-vlan-tag <integer>] - VLAN Tag**

Selects the network ports that match this parameter value.

**[-remote-device-id <text>] - Remote Device ID**

Selects the network ports that match this parameter value.

**[-ipspace <IPspace>] - IPspace Name**

Use this parameter to display information only about the ports that match the IPspace you specify.

**[-broadcast-domain <Broadcast Domain>] - Broadcast Domain**

Use this parameter to display information only about the ports that match the broadcast-domain you specify.

**[-mtu-admin <integer>] - MTU Administrative**

Selects the network ports that match this parameter value.

**[-health-status {healthy|degraded}] - Port Health Status**

Use this parameter to display information only about the ports that match the health-status you specify.

**[-ignore-health-status {true|false}] - Ignore Port Health Status**

Use this parameter to display information only about the ports that match the ignore-health-status you specify.

**[-health-degraded-reasons {l2-reachability|link-flapping|crc-errors|vswitch-link}] - Port Health Degraded Reasons**

Use this parameter to display information only about the ports that match the degraded-reason you specify.

**[-vm-network-name <text>] - Virtual Machine Network Name**

Use this parameter to display information only about the ports that match the network name you specify. Google Cloud Platform only.

### Examples

The following example displays information about all network ports.

```
cluster1::> network port show

Node: node1

Ignore
                                                  Speed(Mbps) Health
Health
Port      IPspace      Broadcast Domain Link MTU  Admin/Oper  Status
Status
--------- ------------ ---------------- ---- ---- ----------- --------
------
e0a       Cluster      Cluster          up   9000  auto/1000  healthy
false
e0b       Cluster      Cluster          up   9000  auto/1000  healthy
false
e0c       Default      Default          up   1500  auto/1000  degraded
false
e0d       Default      Default          up   1500  auto/1000  degraded
true
Node: node2

Ignore
                                                  Speed(Mbps) Health
Health
Port      IPspace      Broadcast Domain Link MTU  Admin/Oper  Status
Status
--------- ------------ ---------------- ---- ---- ----------- --------
------
e0a       Cluster      Cluster          up   9000  auto/1000  healthy
false
e0b       Cluster      Cluster          up   9000  auto/1000  healthy
false
e0c       Default      Default          up   1500  auto/1000  healthy
false
e0d       Default      Default          up   1500  auto/1000  healthy
false

8 entries were displayed.
```

The following example displays health information about all network ports.

```
cluster1::> network port show -health
                                  Ignore
                        Health    Health
Node       Port      Link Status    Status Degraded Reasons
--------   --------- ---- --------- ------ ----------------------
node1
           e0a       up   healthy   false  -
           e0b       up   healthy   false  -
           e0c       up   degraded  false  l2_reachability,
                                           link_flapping
           e0d       up   degraded  false  l2_reachability

node2
           e0a       up   healthy   false  -
           e0b       up   healthy   false  -
           e0c       up   healthy   false  -
           e0d       up   degraded  false  -

8 entries were displayed.
```

## network port broadcast-domain add-ports

Add ports to a layer 2 broadcast domain

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

Add ports to a broadcast domain.

> The IPSpace of the ports added will be updated to the IPSpace of the broadcast-domain. The ports will be added to the failover-group of the broadcast-domain. The MTU of the ports will be updated to the MTU of the broadcast-domain.

### Parameters

**`-ipspace <IPspace>` - IPspace Name**

The IPspace of the broadcast domain.

**`-broadcast-domain <Broadcast Domain>` - Layer 2 Broadcast Domain**

The broadcast domain for this port assignment.

**`-ports [node>:<port],…` - List of ports**

The ports to be added to this broadcast domain.

### Examples

The following example adds the port "e0d" on node "cluster1-1" and port "e0d" on node "cluster1-2" to broadcast domain "mgmt" in IPspace "Default".

```
cluster1::network port broadcast-domain> add-ports -ipspace Default
-broadcast-domain mgmt -ports cluster1-1:e0d, cluster1-2:e0d
```

## network port broadcast-domain create

Create a new layer 2 broadcast domain

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

Create a new broadcast domain.

> The IPSpace of the ports added will be updated to the IPSpace of the broadcast-domain. A failover-group will be generated containing the ports of the broadcast-domain. The MTU of all of the ports in the broadcast-domain will be updated to the MTU specified for the broadcast-domain.

### Parameters

**`[-ipspace <IPspace>]` - IPspace Name**

The IPspace to which the new broadcast domain belongs.

**`-broadcast-domain <Broadcast Domain>` - Layer 2 Broadcast Domain**

The name of the broadcast domain to be created. The name of the broadcast domain needs to be unique within the IPspace.

**`-mtu <integer>` - Configured MTU**

MTU of the broadcast domain.

**`[-ports [node>:<port],…]` - Ports**

The network ports to be added to the broadcast domain. Ports need to be added to the broadcast domain before interfaces can be hosted on the port. By default, no port will be added to the broadcast domain.

### Examples

The following example creates broadcast domain "mgmt" in IPspace "Default" with an MTU of 1500 and network ports e0c from node "gx1" and node "gx2".

```
cluster1::> network port broadcast-domain create -ipspace Default
-broadcast-domain mgmt -mtu 1500 -ports gx1:e0c,gx2:e0c
```

## network port broadcast-domain delete

Delete a layer 2 broadcast domain

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

Delete a broadcast domain that contains no ports.

### Parameters

**`-ipspace <IPspace>` - IPspace Name**
The IPspace to which the broadcast domain belongs

**`-broadcast-domain <Broadcast Domain>` - Layer 2 Broadcast Domain**
The name of the broadcast domain to be deleted.

### Examples

The following example deletes the broadcast domain "mgmt" in IPspace "Default".

```
cluster1::network port broadcast-domain> delete -ipspace Default
-broadcast-domain mgmt
```

## network port broadcast-domain merge

Merges two layer 2 broadcast domains

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

Merges a broadcast domain into an existing broadcast domain.

### Parameters

**`-ipspace <IPspace>` - IPspace Name**
The IPspace of the broadcast domain.

**`-broadcast-domain <Broadcast Domain>` - Layer 2 Broadcast Domain**
The merging broadcast domain.

**`-into-broadcast-domain <Broadcast Domain>` - Merge with This Layer 2 Broadcast Domain**
The target broadcast domain for the merge operation.

### Examples

The following example merges broadcast domain "bd-mgmt" in IPspace "Default" to broadcast domain "bd-data".

```
cluster1::network port broadcast-domain> merge -ipspace Default -broadcast
-domain bd-mgmt -into-broadcast-domain bd-data
```

## network port broadcast-domain modify

Modify a layer 2 broadcast domain

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

Modify a broadcast domain.

### Parameters

**`-ipspace <IPspace>` - IPspace Name**
The IPspace to which the broadcast domain belongs.

**`-broadcast-domain <Broadcast Domain>` - Layer 2 Broadcast Domain**
The name of the broadcast domain.

**`[-mtu <integer>]` - Configured MTU**
MTU of the broadcast domain.

### Examples

The following example modifies the mtu attribute of broadcast domain "mgmt" in IPspace "Default" to 1500

```
cluster1::network port broadcast-domain*> modify -ipspace Default
-broadcast-domain mgmt -mtu 1500
```

## network port broadcast-domain move

Move a layer 2 broadcast domain to another IPspace

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

Move a broadcast domain to another IPspace.

### Parameters

**`-ipspace <IPspace>` - IPspace Name**
The IPspace to which the broadcast domain belongs.

**-broadcast-domain <Broadcast Domain> - Layer 2 Broadcast Domain**

The name of the broadcast domain.

**-to-ipspace <IPspace> - Name of the destination IPspace**

The name of the destination IPspace.

#### Examples

The following example moves the broadcast domain named "mgmt" from IPspace "Default" to IPspace "Default-1".

```
cluster1::network port broadcast-domain> move -ipspace Default -broadcast
-domain mgmt -to-ipspace Default-1
```

## network port broadcast-domain remove-ports

Remove ports from a layer 2 broadcast domain

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

Remove port assignments from a broadcast domain.

#### Parameters

**-ipspace <IPspace> - IPspace Name**

The IPspace of the broadcast domain.

**-broadcast-domain <Broadcast Domain> - Layer 2 Broadcast Domain**

The broadcast domain of the ports.

**-ports [node>:<port],… - List of ports**

The ports to removed from the broadcast-domain.

#### Examples

The following example removes port "e0d" on node "cluster1-1" and port "e0d" on node "cluster1-2" from broadcast domain "mgmt" in IPspace "Default".

```
cluster1::network port broadcast-domain> remove-ports -ipspace Default
-broadcast-domain mgmt -ports cluster1-1:e0d, cluster1-2:e0d
```

## network port broadcast-domain rename

Rename a layer 2 broadcast domain

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

Rename a broadcast domain.

### Parameters

**`-ipspace <IPspace>` - IPspace Name**

The IPspace to which the broadcast domain belongs.

**`-broadcast-domain <Broadcast Domain>` - Layer 2 Broadcast Domain**

The name of the broadcast domain.

**`-new-name <text>` - New Name**

The new name of the broadcast domain.

### Examples

The following example renames the broadcast domain named "mgmt" to "mgmt2" in IPspace "Default".

```
cluster1::network port broadcast-domain> rename -ipspace Default
-broadcast-domain mgmt -new-name mgmt2
```

## network port broadcast-domain show

Display layer 2 broadcast domain information

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

Display broadcast domain information.

### Parameters

**{ `[-fields <fieldname>,…]`**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| `[-instance ]` }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**`[-ipspace <IPspace>]` - IPspace Name**

Selects the broadcast domains that match the IPspace name.

**`[-broadcast-domain <Broadcast Domain>]` - Layer 2 Broadcast Domain**

Selects the broadcast domains that match the broadcast domain name.

**`[-mtu <integer>]` - Configured MTU**

Selects the broadcast domains that match the MTU value. This field is the MTU that was configured by the

user, which might be different from the operational MTU.

**[-ports [node>:<port],…] - Ports**

Selects the broadcast domains that contain the network ports. For example, node1:e0a will display broadcast domains that contain node1:e0a network port.

**[-port-update-status {complete|in-progress|error|overridden-while-offline}] - Port Update Status**

Selects the broadcast domains that contain the network port status. For example, specifying "error" will display broadcast domains that contain "Error" network port status.

**[-port-update-status-details <text>,…] - Status Detail Description**

Selects the broadcast domains that contain the network port status detail text.

**[-port-update-status-combined {complete|in-progress|error|overridden-while-offline}] - Combined Port Update Status**

Selects the broadcast domains that contain the combined network port status. For example, specifying "error" will display broadcast domains that contain a combined network port status of "Error".

**[-failover-groups <failover-group>,…] - Failover Groups**

Selects the broadcast domains that contain the failover groups.

**[-subnet-names <subnet name>,…] - Subnet Names**

Selects the broadcast domains that contain the subnet name or names.

**[-is-vip {true|false}] - Is VIP Broadcast Domain**

Selects the broadcast domains that match a specific "is-vip" flag. Specifying "true" matches only broadcast domains associated with the Virtual IP feature; "false" matches only broadcast domains that do not.

### Examples

The following example displays general information about broadcast domains.

```
cluster1::> network port broadcast-domain show
IPspace Broadcast                                       Update
Name    Domain Name    MTU  Port List                   Status Details
------- ----------- ------  --------------------------- --------------
Cluster Cluster       9000
                            node1:e0a                   complete
                            node1:e0b                   complete
Default Default       1500
                            node1:e0c                   complete
                            node1:e0d                   complete
2 entries were displayed.
```

## network port broadcast-domain split

Splits a layer 2 broadcast domain into two layer 2 broadcast domains.

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

Splits ports from a broadcast domain into a new broadcast domain.

The following restrictions apply to this command:

- If the ports are in a failover group, all ports in the failover group must be provided. Use network interface failover-groups show to see which ports are in failover groups.
- If the ports have LIFs associated with them, the LIFs cannot be part of a subnet's ranges and the LIF's `curr-port` and `home-port` must both be provided. Use network interface show`-fields` *`subnet-name`* , *`home-node`* , *`home-port`* , *`curr-node`* , *`curr-port`* to see which ports have LIFs associated with them and whether the LIFs are part of a subnet's ranges. Use network subnet remove-ranges with the LIF's IP address and `-force-update-lif-associations` set to *`true`* to remove the LIF's association with a subnet.

### Parameters

**`-ipspace <IPspace>` - IPspace Name**

The IPspace of the broadcast domain.

**`-broadcast-domain <Broadcast Domain>` - Layer 2 Broadcast Domain**

The broadcast domain to split.

**`-new-broadcast-domain <Broadcast Domain>` - New Layer 2 Broadcast Domain Name**

The new broadcast domain.

**`-ports [node>:<port]`,… - List of Ports**

The ports to be split from this broadcast domain.

### Examples

The following example splits port "e0d" on node "cluster1-1" and port "e0d" on node "cluster1-2" from broadcast domain "bd-mgmt" in IPspace "Default" to broadcast domain "bd-data".

```
cluster1::> network port broadcast-domain split -ipspace Default
-broadcast-domain bd-mgmt -new-broadcast-domain bd-data -ports cluster1-
1:e0d, cluster1-2:e0d
```

### Related Links

- network interface failover-groups show
- network interface show
- network subnet remove-ranges

## network port ifgrp add-port

Add a port to an interface group

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

The `network port ifgrp add-port` command adds a network port to a port interface group. The port interface group must already exist. You can create a port interface group by using the network port ifgrp create command.

The following restrictions apply to port interface groups:

- A port that is already a member of a port interface group cannot be added to another port interface group.
- Cluster ports and management ports cannot be in a port interface group.
- A port to which a logical interface is already bound cannot be added to a port interface group.
- A port that already has an assigned failover role cannot be added to a port interface group.
- A VLAN port cannot be added to a port interface group.
- A port which attaches to a VLAN cannot be added to a port interface group.
- An interface group port cannot be added to a port interface group.
- A port that is assigned to a broadcast domain cannot be added to a port interface group.
- All ports in a port interface group must be physically located on the same node.

### Parameters

**`-node {<nodename>|local}` - Node**

The node on which the port interface group is located.

**`-ifgrp <ifgrp name>` - Interface Group Name**

The port interface group to which a port is to be added.

**`-port <netport>` - Specifies the name of port.**

The network port that is to be added to the port interface group.

**`[-skip-broadcast-domain-placement <true>]` - Skip Placement Into Broadcast Domain (privilege: advanced)**

When specified along with the first port added to the ifgrp, the ifgrp will not be placed into any broadcast domain.

### Examples

The following example adds port e0c to port interface group a1a on a node named node1:

```
cluster1::> network port ifgrp add-port -node node1 -ifgrp a1a -port e0c
```

### Related Links

- network port ifgrp create

## network port ifgrp create

Create a port interface group

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

The `network port ifgrp create` command creates a port interface group. See the documentation for the network port ifgrp add-port command for a list of restrictions on creating port interface groups.

### Parameters

**`-node {<nodename>|local}` - Node**

The node on which the port interface group will be created.

**`-ifgrp <ifgrp name>` - Interface Group Name**

The name of the port interface group that will be created. Port interface groups must be named using the syntax "a<number><letter>", where <number> is an integer in the range [0-999] without leading zeros and <letter> is a lowercase letter. For example, "a0a", "a0b", "a1c", and "a2a" are all valid port interface group names.

**`-distr-func {mac|ip|sequential|port}` - Distribution Function**

The distribution function of the port interface group that will be created. Valid values are:

- mac - Network traffic is distributed based on MAC addresses
- ip - Network traffic is distributed based on IP addresses
- sequential - Network traffic is distributed in round-robin fashion from the list of configured, available ports
- port - Network traffic is distributed based on the transport layer (TCP/UDP) ports

**`-mode {multimode|multimode_lacp|singlemode}` - Create Policy**

The create policy for the interface group that will be created. Valid values are:

- multimode - Bundle multiple member ports of the interface group to act as a single trunked port
- multimode_lacp - Bundle multiple member ports of the interface group using Link Aggregation Control Protocol
- singlemode - Provide port redundancy using member ports of the interface group for failover

### Examples

The following example creates a port interface group named a0a on node node0 with a distribution function of ip:

```
cluster1::> network port ifgrp create -node node0 -ifgrp a0a -distr-func
ip -mode multimode
```

### Related Links

- network port ifgrp add-port

## network port ifgrp delete

### Destroy a port interface group

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

The `network port ifgrp delete` command destroys a port interface group.

> When you delete an interface group port, it is automatically removed from failover rules and groups to which it belongs.

#### Parameters

**`-node {<nodename>|local}` - Node**
The node on which the port interface group is located.

**`-ifgrp <ifgrp name>` - Interface Group Name**
The port interface group that will be deleted.

#### Examples

The following example deletes port interface group a0b from a node named node0.

```
cluster1::> network port ifgrp delete -node node0 -ifgrp a0b
```

## network port ifgrp remove-port

### Remove a port from an interface group

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

The `network port ifgrp remove-port` command removes a network port from a port interface group.

#### Parameters

**`-node {<nodename>|local}` - Node**
The node on which the port interface group is located.

**`-ifgrp <ifgrp name>` - Interface Group Name**
The port interface group from which a port will be removed.

**-port <netport> - Specifies the name of port.**

The network port that will be removed from the port interface group.

#### Examples

The following example removes port e0d from port interface group a1a on a node named node1:

```
cluster1::> network port ifgrp remove-port -node node1 -ifgrp a1a -port
e0d
```

## network port ifgrp show

### Display port interface groups

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

The `network port ifgrp show` command displays information about port interface groups. By default, it displays information about all port interface groups on all nodes in the cluster.

#### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-node {<nodename>|local}] - Node**

Selects the port interface groups that match this parameter value. Use this parameter with the `-ifgrp` parameter to select information about a specific port interface group.

**[-ifgrp <ifgrp name>] - Interface Group Name**

Selects the port interface groups that match this parameter value. Use this parameter with the `-node` parameter, to select information about a specific port interface group.

**[-distr-func {mac|ip|sequential|port}] - Distribution Function**

Selects the port interface groups that match this parameter value.

**[-mode {multimode|multimode_lacp|singlemode}] - Create Policy**

Selects the port interface groups that match this parameter value.

**[-mac <MAC Address>] - MAC Address**

Selects the port interface groups that match this parameter value.

**[-activeports {full|partial|none}] - Port Participation**

Selects the port interface groups that match this parameter value. The value "partial" indicates that some but not all of the port interface group's ports are active. the value "full" indicates that all of the port interface group's ports are active.

**[-ports {<netport>|<ifgrp>}] - Network Ports**

Selects the port interface groups that match this parameter value.

**[-up-ports {<netport>|<ifgrp>}] - Up Ports**

Selects the port interface groups that match this parameter value. Displays only the ports that are up.

**[-down-ports {<netport>|<ifgrp>}] - Down Ports**

Selects the port interface groups that match this parameter value. Displays only the ports that are down.

### Examples

The following example displays information about all port interface groups.

```
cluster1::> network port ifgrp show
        Port     Distribution                Active
Node    ifgrp    Function     MAC Address    Ports   Ports
------  -------  ------------ -------------- ------- ---------------
node0
        a0a      ip           b8:f8:7a:20:00 partial e0c
node1
        a1a      ip           07:26:60:02:00 full    e0d
```

## network port reachability repair

Repair reachability

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

Repair network port configuration to match the detected reachable broadcast domain. If the reachability scan has completed with a reachability-status of misconfigured-reachability, meaning the network port does not have reachability to its configured broadcast domain, but has reachability to another broadcast domain, then repairing the network port's reachability will assign the port to the detected broadcast domain. Similarly, if the reachability scan has completed with a reachability status of no-reachability, then repairing the network port's reachability will assign the port to an empty broadcast domain. LIFs configured on the port will be adjusted to be configured on another port in their current broadcast domain if possible. Vlans on the specified port that do not have reachability to their configured broadcast domain will be removed. If the port was part of an ifgrp, the port will be removed from the ifgrp. If the port is not configured on a broadcast domain and has no reachability to any existing broadcast domains, it will be configured in a newly created broadcast domain.

### Parameters

**-node {<nodename>|local} - Node**

Selects the node on which the port resides.

**-port <netport> - Port**

Selects the port on which to repair configuration.

#### Examples

The following example applies the scanned broadcast domain reachability information to the specified port's configuration.

```
cluster1::> network port reachability repair -node node1 -port e0c
```

## network port reachability scan

### Perform a reachability scan

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

Scan network port configuration to detect reachable broadcast domains.

#### Parameters

**-node {<nodename>|local} - Node**

Selects the node on which the port resides.

**-port <netport> - Port**

Selects the port on which to scan broadcast domain reachability.

#### Examples

The following example applies the scanned broadcast domain reachability information to the specified port's configuration.

```
cluster1::> network port reachability scan -node node1 -port e0c
```

## network port reachability show

### Display Reachability Status

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

Display broadcast domain reachability information for the specified port. If the port is discovered, via a layer 2 reachability scan, to have reachability to broadcast domains other than the one on which it is expected, the

command will list the reachable broadcast domains and an appropriate reachability status message. If the reachable broadcast domain matches the expected one, the reachability status is displayed as Ok.

### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| [-detail ]**

When true, additional details regarding which broadcast domains have been found to be reachable from the specified network port are displayed.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-node {<nodename>|local}] - Node**

Selects the node on which the port resides.

**[-port <netport>] - Port**

Selects the port for which to display broadcast domain reachability information.

**[-expected-broadcast-domain [IPspace>:<Broadcast Domain]] - Expected Broadcast Domain**

The IPspace and broadcast domain currently assigned to the network port. If the specified port is an ifgrp member, the expected broadcast domain is the IPspace and broadcast domain currerntly assigned to the parent ifgrp.

**[-reachable-broadcast-domains [IPspace>:<Broadcast Domain],…] - Reachable Broadcast Domains**

The name of the IPspaces and broadcast domains that have reachability to the specified port, as discovered by a layer 2 reachability scan.

**[-reachability-status {unknown|ok|no-reachability|misconfigured-reachability|multi-domain-reachability|internal-error}] - Reachability Status**

The status of the broadcast domain reachability for the specified port. "Ok" if the expected broadcast domain matches the reachable broadcast domains, i.e., the port can reach other ports in the expected broadcast domain, but no ports configured in other broadcast domains. "No-reachability" if the port cannot reach any ports in the expected broadcast domain, and also cannot reach any ports in any other broadcast domains. "Misconfigured-reachability" if the port cannot reach any ports in the expected broadcast domain, but can reach ports in one other broadcast domain. "Multi-domain-reachability" if the port can reach other ports configured in multiple broadcast domains. "Unknown" if the port has not been link-up long enough for reachability to be determined.

**[-unreachable-ports [node>:<port],…] - Unreachable Ports**

The list of network ports that are expected in the same broadcast domain as the specified port but cannot be reached, either because those ports are down or because there is no network connectivity to those ports.

**[-unexpected-ports [node>:<port],…] - Unexpected Ports**

The list of network ports that are not expected in the same broadcast domain yet have network connectivity to the specified port.

**Examples**

The following example displays the broadcast domain reachability for the specified port.

```
cluster1::> network port reachability show -node node1 -port e0d
network port reachability show)
ode         Port     Expected Reachability        Reachability Status
----------- -------- ---------------------------
-------------------------
ode1        e0d      Default:Default              ok
```

The following example displays the detailed reachability for the 'e0d' port when it has misconfigured-reachability, i.e., it cannot reach the other 'e0d' port in the expected broadcast domain 'Default:Default', but can reach the 'e0c' ports configured in the 'Default:Default-3' broadcast domain.

```
cluster1::> network port reachability show -node node1 -port e0d -detail
network port reachability show)
ode         Port     Expected Reachability        Reachability Status
----------- -------- ---------------------------
-------------------------
ode1        e0d      Default:Default              misconfigured-
reachability
Actual Reachability: Default:Default-3
Unreachable Ports: node2:e0d
 Unexpected Ports: node2:e0c, node1:e0c
```

The following example displays the detailed reachability for the 'e0d' port when it has multi-domain-reachability, i.e., it can reach the other 'e0d' port in the expected broadcast domain 'Default:Default', but can also reach the 'e0c' ports configured in the 'Default:Default-3' broadcast domain.

```
cluster1::> network port reachability show -node node1 -port e0d -detail
network port reachability show)
ode         Port     Expected Reachability        Reachability Status
----------- -------- ---------------------------
-------------------------
ode1        e0d      Default:Default              multi-domain-
reachability
Actual Reachability: Default:Default,
                     Default:Default-3
Unreachable Ports: -
 Unexpected Ports: node2:e0c, node1:e0c
```

The following example displays the detailed reachability for the 'e0d' port when it has no-reachability, i.e., it cannot reach the other 'e0d' port in the expected broadcast domain 'Default:Default', and also cannot reach any other ports configured in broadcast domains.

```
cluster1::> network port reachability show -node node1 -port e0d -detail
network port reachability show)
ode         Port     Expected Reachability        Reachability Status
----------- -------- ---------------------------
-------------------------
ode1        e0d      Default:Default              no-reachability
Actual Reachability: -
Unreachable Ports: node2:e0d
 Unexpected Ports: -
```

## network port vip create

### Create a VIP port

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

#### Description

The `network port vip create` command creates a VIP port in the specified IPspace on the specified node. Only one VIP port can be created per IPspace on the given node.

#### Parameters

**`-node {<nodename>|local}` - Node**

The node where the VIP port should be created.

**`-port <netport>` - Network Port**

The name of the VIP port to be created in the format v<slot-number><port-letter>

**`-ipspace <IPspace>` - IPspace Name**

The IPspace where the VIP port should be created. The default value for this parameter is "Default", which identifies the default IPspace.

#### Examples

This example shows how to create a VIP port named v0a in ipspace ips on node1.

```
cluster1::> network port vip create -node node1 -port v0a -ipspace ips
```

## network port vip delete

### Delete a VIP port

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

### Description

The `network port vip delete` command deletes a VIP port.

### Parameters

**`-node {<nodename>|local}` - Node**

The node associated with the VIP port to be deleted.

**`-port <netport>` - Network Port**

The name of the VIP port to be deleted.

### Examples

This example shows how to delete VIP Port v0a on node1.

```
cluster1::> network port vip delete -node node1 -port v0a
```

## network port vip show

Display VIP ports

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

The `network port vip show` command displays information about VIP ports.

### Parameters

**`{ [-fields <fieldname>,…]`**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**`| [-instance ] }`**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**`[-node {<nodename>|local}]` - Node**

This parameter selects the VIP ports that match the specified node.

**`[-port <netport>]` - Network Port**

This parameter selects the VIP ports that match the specified port.

**`[-ipspace <IPspace>]` - IPspace Name**

This parameter selects the VIP ports that match the specified IPspace.

### Examples

The example below shows VIP port v0a is created in IPspace ips on node1.

```
cluster1::> network port vip show
    Node   VIP Port IPspace
    ------ -------- -------------
    node1  v0a      ips
```

## network port vlan create

Create a virtual LAN (VLAN)

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

The `network port vlan create` command attaches a VLAN to a network port on a specified node.

### Parameters

**`-node {<nodename>|local}` - Node**

The node to which the VLAN is to be attached.

> You cannot attach a VLAN to a cluster port.

**{ `-vlan-name {<netport>|<ifgrp>}` - VLAN Name**

The name of the VLAN that is to be attached. This name should be a combination of the name of the port or interface group and the VLAN ID, with a hyphen between, such as "e1c-80".

**| `-port {<netport>|<ifgrp>}` - Associated Network Port**

The network port to which the VLAN is to be attached.

**`-vlan-id <integer>` - Network Switch VLAN Identifier }**

The ID tag of the created VLAN.

**`[-skip-broadcast-domain-placement <true>]` - Skip Placement Into Broadcast Domain (privilege: advanced)**

When specified, the VLAN will not be placed into any broadcast domain.

### Examples

This example shows how to create VLAN e1c-80 attached to network port e1c on node1.

```
cluster1::> network port vlan create -node node1 -vlan-name e1c-80
```

## network port vlan delete

Delete a virtual LAN (VLAN)

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

The `network port vlan delete` command deletes a VLAN from a network port.

> When you delete a VLAN port, it is automatically removed from all failover rules and groups that use it.

### Parameters

**`-node {<nodename>|local}` - Node**

The node from which the VLAN is to be deleted.

**{ `-vlan-name {<netport>|<ifgrp>}` - VLAN Name**

The name of the VLAN that is to be deleted

**| `-port {<netport>|<ifgrp>}` - Associated Network Port**

The network port to which the VLAN is to be attached.

**`-vlan-id <integer>` - Network Switch VLAN Identifier }**

The ID tag of the deleted VLAN.

### Examples

This example shows how to delete VLAN e1c-80 from network port e1c on node1.

```
cluster1::> network port vlan delete -node node1 -vlan-name e1c-80
```

## network port vlan show

Display virtual LANs (VLANs)

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

The `network port vlan show` command displays information about network ports that are attached to VLANs. The command output indicates any inactive links and lists the reason for the inactive status.

If the operational duplex mode and speed cannot be determined (for instance, if the link is down), they are listed as `undef` , meaning undefined.

### Parameters

**{ `[-fields <fieldname>,…]`**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| `[-instance ]` }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-node {<nodename>|local}] - Node**

Selects the VLAN network ports that match this parameter value.

**{ [-vlan-name {<netport>|<ifgrp>}] - VLAN Name**

Selects the VLAN network ports that match this parameter value.

**| [-port {<netport>|<ifgrp>}] - Associated Network Port**

Selects the VLAN network ports that match this parameter value. If neither this parameter nor `-name` are used, the command displays information about all network ports.

**[-vlan-id <integer>] - Network Switch VLAN Identifier }**

Selects the VLAN network ports that match this parameter value.

**[-mac <MAC Address>] - MAC address**

Selects the VLAN network ports that match this parameter value.

### Examples

The example below shows VLAN e1b-70 attached to port e1b on node1.

```
cluster1::> network port vlan show
                  Network Network
Node   VLAN Name Port    VLAN ID  MAC Address
------ --------- ------- -------  ------------------
node1  e1b-70    e1b     70       00:15:17:76:7b:69
```

# network qos-marking commands

## network qos-marking modify

Modify the QoS marking values

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

The `network qos-marking modify` command modifies the QoS marking values for different protocols, for each IPspace.

### Parameters

**-ipspace <IPspace> - IPspace name**

Use this parameter to specify the IPspace for which the QoS marking entry is to be modified.

**-protocol <text> - Protocol**

Use this parameter to specify the protocol for which the QoS marking entry is to be modified. The possible values are NFS, CIFS, iSCSI, SnapMirror, SnapMirror-Sync, NDMP, FTP, HTTP-admin, HTTP-filesrv, SSH, Telnet, and SNMP.

**[-dscp <integer>] - DSCP Marking Value**

Use this parameter to specify the DSCP value. The possible values are 0 to 63.

**[-is-enabled {true|false}] - Is QoS Marking Enabled**

Use this parameter to enable or disable the QoS marking for the specified protocol and IPspace.

#### Examples

The following example modifies the QoS marking entry for the NFS protocol in the Default IPspace:

```
cluster1::> network qos-marking modify -ipspace Default -protocol NFS
-dscp 10 -is-enabled true
```

## network qos-marking show

Display the QoS marking values

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

The `network qos-marking show` command displays the QoS marking values for different protocols, for each IPspace.

#### Parameters

**{ [-fields <fieldname>,…]**

Use this parameter to display only certain fields of the QoS marking table.

**| [-instance ] }**

Use this parameter to display all the fields of the QoS marking table.

**[-ipspace <IPspace>] - IPspace name**

Use this parameter to display the QoS marking entries for the specified IPspace.

**[-protocol <text>] - Protocol**

Use this parameter to display the QoS marking entries for the specified protocol. The possible values are NFS, CIFS, iSCSI, SnapMirror, SnapMirror-Sync, NDMP, FTP, HTTP-admin, HTTP-filesrv, SSH, Telnet, and SNMP.

**[-dscp <integer>] - DSCP Marking Value**

Use this parameter to display the QoS marking entries matching the specified DSCP value. The possible values are 0 to 63.

**[-is-enabled {true|false}] - Is QoS Marking Enabled**

Use this parameter to display the QoS marking entries matching the specified flag.

#### Examples

The following example displays the QoS marking entries for the Default IPspace.

```
cluster1::> network qos-marking show -ipspace Default
IPspace              Protocol           DSCP  Enabled?
-------------------- ----------------- -----  --------
Default
                     CIFS                 10  false
                     FTP                  48  false
                     HTTP-admin           48  false
                     HTTP-filesrv         10  false
                     NDMP                 10  false
                     NFS                  10  true
                     SNMP                 48  false
                     SSH                  48  false
                     SnapMirror           10  false
                     SnapMirror-Sync      10  false
                     Telnet               48  false
                     iSCSI                10  false
12 entries were displayed.
```

# network route commands

## network route create

Create a static route

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

#### Description

The `network route create` command creates a static route within a Vserver.

#### Parameters

**`-vserver <vserver>` - Vserver Name**

Use this parameter to specify the Vserver on which the route is to be created.

**`-destination <IP Address/Mask>` - Destination/Mask**

Use this parameter to specify the IP address and subnet mask of the route's destination. The format for this value is: address, slash ("/"), mask. `0.0.0.0/0` is a valid destination value to create default IPv4 route. And `::/0` is a valid destination value to create default IPv6 route

**`-gateway <IP Address>` - Gateway**

Use this parameter to specify the IP address of the gateway server leading to the route's destination.

**[-metric <integer>] - Metric**

Use this parameter to specify the metric of the route.

#### Examples

The following example creates default routes within Vserver vs0 for IPv4 and IPv6.

```
cluster1::> network route create -vserver vs0 -destination 0.0.0.0/0
-gateway 10.61.208.1
cluster1::> network route create -vserver vs0 -destination ::/0 -gateway
3ffe:1::1
```

## network route delete

### Delete a static route

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

#### Description

The `network route delete` command deletes a static route from a Vserver.

#### Parameters

**-vserver <vserver> - Vserver Name**

Use this parameter to specify the Vserver on which the route is to be deleted.

**-destination <IP Address/Mask> - Destination/Mask**

Use this parameter to specify the IP address and subnet mask of the route's destination. The format for this value is: address, slash ("/"), mask. For example, `0.0.0.0/0` is a correctly formatted value for the -destination parameter.

**-gateway <IP Address> - Gateway**

Use this parameter to specify the gateway on which the route is to be deleted.

#### Examples

The following example deletes a route within Vserver vs0 for destination 0.0.0.0/0.

```
cluster1::network route delete -vserver vs0 -destination 0.0.0.0/0
```

## network route show-lifs

### Show the Logical Interfaces for each route entry

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

### Description

The `network route show-lifs` command displays the association of static routes and Logical Interfaces (LIFs) within one or more Vservers. You can view routes within specified Vservers, routes with specified destinations, and routes with specified gateways.

### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-vserver <vserver>] - Vserver Name**

Use this parameter to display only routes that have the specified Vserver as their origin.

**[-destination <IP Address/Mask>] - Destination/Mask**

Use this parameter to display only routes that have the specified IP address and subnet mask as their destination. The format for this value is: address, slash ("/"), mask. For example, `0.0.0.0/0` is a valid value for the -destination parameter.

**[-gateway <IP Address>] - Gateway**

Use this parameter to display only routes that have the specified IP address as their gateway.

**[-lifs <lif-name>,…] - Logical Interfaces**

Use this parameter to display only the routes that are associated with the specified Logical Interfaces (LIFs).

**[-address-family {ipv4|ipv6|ipv6z}] - Address Family**

Use this parameter to display only the routes that belong to specified address family.

## network route show

### Display static routes

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

### Description

The `network route show` command displays a group of static routes within one or more Vservers. You can view routes within specified Vservers, routes with specified destinations, and routes with specified gateways.

### Parameters

**{ [-fields <fieldname>,…]**

Use this parameter to display only certain fields of the routing tables.

**| [-instance ] }**

Use this parameter to display all fields of the routing tables.

**[-vserver <vserver>] - Vserver Name**

Use this parameter to display only routes that have the specified Vserver as their origin.

**[-destination <IP Address/Mask>] - Destination/Mask**

Use this parameter to display only routes that have the specified IP address and subnet mask as their destination. The format for this value is: address, slash ("/"), mask. The example below has `0.0.0.0/0` as a valid value for the -destination parameter.

**[-gateway <IP Address>] - Gateway**

Use this parameter to display only routes that have the specified IP address as their gateway.

**[-metric <integer>] - Metric**

Use this parameter to display only routes that have the specified metric.

**[-ipspace <IPspace>] - IPspace Name**

Use this parameter to optionally specify the IPspace associated with the Vserver. This parameter can be used in conjunction with the Vserver parameter in order to configure the same route across multiple Vservers within an IPspace.

**[-address-family {ipv4|ipv6|ipv6z}] - Address family of the route**

Use this parameter to display only the routes that have the specified address-family.

**Examples**

The following example displays information about all routing groups.

```
cluster1::> network route show
  (network route show)
Server              Destination     Gateway         Metric
------------------- --------------- --------------- ------
node1
                    0.0.0.0/0       10.61.208.1     20
node2
                    0.0.0.0/0       10.61.208.1     20
vs0
                    0.0.0.0/0       10.61.208.1     20
3 entries were displayed.
```

## network route active-entry show

### Display active routes

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *advanced* privilege level.

### Description

The `network route active-entry show` command displays installed routes on one or more nodes. You can view routes within specified nodes, within specified Vservers, routes in specified subnet groups, and routes with specified destinations.

### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| [-verbose ] (privilege: advanced)**

Use this parameter to display the reference count, use, interface, and Path MTU fields.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-vserver <vserver>] - Vserver Name (privilege: advanced)**

Displays the routes that have the specified Vserver as their origin.

**[-node {<nodename>|local}] - Node (privilege: advanced)**

Displays the routes from the specified node.

**[-address-type {ipv4|ipv6|ipv6z}] - Address Family (privilege: advanced)**

Displays the routes that have the specified IP address type.

**[-subnet-group <IP Address/Mask>] - Subnet Group (privilege: advanced)**

Displays the group of routes that belong to the specified subnet. Routes within the specified subnet group are used first before the default set. The "default" subnet group is a system-provided set of default routes.

**[-destination <text>] - Destination (privilege: advanced)**

Displays the routes that have the specified IP address or subnet as their destination. The format for the subnet is: <address>/<mask>. IPv6 address includes the scope value after percentage ("%"). 0.0.0.0/0, 169.254.4.60, ff02::%e0a/32 and fe80::250:56ff:fea6:db7c%e0b are valid values for this parameter.

**[-interface <text>] - Interface Name (privilege: advanced)**

Displays the routes that use the specified interface to transmit packets to the destination. A valid interface has the format of {<netport>|<ifgrp>}, such as "e0a", "e0a-1" and "a0a", or it can be a loopback interface, such as "lo" and "losk".

**[-route-interface-address <text>] - Route Interface Address (privilege: advanced)**

Displays the routes that use the specified IP address on the transmit interface.

**[-gateway <text>] - Route gateway (privilege: advanced)**

Displays the routes that have the specified gateway. The gateway can be an IP address, such as "10.10.2.1" and "fe80::1%lo", MAC address, such as "0:5:73:a0:1:7" or refer to a local link, such as "link#3".

**[-metric <integer>] - Metric (privilege: advanced)**

Displays the routes that have the specified metric.

**[-flags <text>] - Flags (privilege: advanced)**

Displays the routes that have the specified flags. The type string for "-flags" needs to be one or more of the following {U|G|H|R|D|S|1|2|L|C} in the order shown.

- U - Usable
- G - Gateway
- H - Host
- R - Reject
- D - Dynamic
- S - Static
- 1 - Protocol1
- 2 - Protocol2
- L - Llinfo
- C - Clone

Multiple values can be specified (for example: UHL).

**[-reference-count <integer>] - Reference Count (privilege: advanced)**

Displays the routes that have the specified reference count in the system.

**[-lookup-count <integer>] - Lookup Count (privilege: advanced)**

Displays the routes that have the specified use count (the count of lookups for the route).

**[-path-mtu <integer>] - Path MTU (privilege: advanced)**

Displays the routes that have the specified path maximum transmission unit.

### Examples

The following example displays active routes on all nodes in Vserver vs0 with subnet-group 10.10.10.0/24.

```
cluster1::*> network route active-entry show -vserver vs0 -subnet-group
10.10.10.0/24
  (network route active-entry show)

Vserver: vs0
Node: node1
Subnet Group: 10.10.10.0/24
Destination             Gateway              Interface   Metric  Flags
----------------------  -------------------  ---------   ------  -----
default                 10.10.10.1           e0c              0  UGS

Vserver: vs0
Node: node2
Subnet Group: 10.10.10.0/24
Destination             Gateway              Interface   Metric  Flags
----------------------  -------------------  ---------   ------  -----
default                 10.10.10.1           e0c              0  UGS
2 entries were displayed.
```

# network subnet commands

## network subnet add-ranges

Add new address ranges to a subnet

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

Add new address ranges to a subnet.

> All addresses in a range must be the same address family (IPv4 or IPv6) and must have the same subnet mask. Ranges that overlap or are next to existing ranges will be merged with the existing ranges.

### Parameters

**`-ipspace <IPspace>` - IPspace Name**

The IPspace in which the range resides.

**`-subnet-name <subnet name>` - Subnet Name**

The name of the subnet.

**`-ip-ranges {<ipaddr>|<ipaddr>-<ipaddr>}` - IP Ranges**

The list of ranges to add to the subnet.

**[-force-update-lif-associations <true>] - Force Update LIF Associations**

This command will fail if any service processor interfaces or network interfaces are using the IP addresses in the ranges provided. Using this parameter will associate any manually addressed interfaces with the subnet and will allow the command to succeed.

#### Examples

The following example allocates addresses for subnet *s1* in IPspace *Default*.

```
cluster1::> network subnet add-ranges -ipspace Default -subnet-name s1
    -ip-ranges "10.98.1.20-10.98.1.30, 10.98.1.35, 10.98.1.40-10.98.1.49"
```

## network subnet create

Create a new layer 3 subnet

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

Create a new subnet.

#### Parameters

**[-ipspace <IPspace>] - IPspace Name**

The IPspace to which the new subnet belongs.

**-subnet-name <subnet name> - Subnet Name**

The name of the subnet to be created. The name of the subnet needs to be unique within the IPspace.

**-broadcast-domain <Broadcast Domain> - Broadcast Domain**

The broadcast domain to which the new subnet belongs.

**-subnet <IP Address/Mask> - Layer 3 Subnet**

The address and mask of the subnet.

**[-gateway <IP Address>] - Gateway**

The gateway of the subnet.

**[-ip-ranges {<ipaddr>|<ipaddr>-<ipaddr>}] - IP Addresses or IP Address Ranges**

The IP ranges associated with this subnet.

**[-force-update-lif-associations <true>] - Change the subnet association**

This command will fail if any service processor interfaces or network interfaces are using the IP addresses in the ranges provided. Using this parameter will associate any manually addressed interfaces with the subnet and will allow the command to succeed.

#### Examples

The following examples create subnets named *s1* and *s6* in IPspace *Default* .

```
cluster1::> network subnet create -ipspace Default -broadcast-domain bd1
-subnet-name s1
    -subnet 192.168.1.0/24 -gateway 192.168.1.1 -ip-ranges "192.168.1.1-
192.168.1.100, 192.168.1.112, 192.168.1.145"
```

```
cluster1::> network subnet create -ipspace Default -broadcast-domain bd1
-subnet-name s6
    -subnet 3FFE::/64 -gateway 3FFE::1 -ip-ranges "3FFE::10-3FFE::20"
```

## network subnet delete

Delete an existing subnet object

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

Delete a subnet that contains no ports.

#### Parameters

**`-ipspace <IPspace>` - IPspace Name**
The IPspace to which the subnet belongs.

**`-subnet-name <subnet name>` - Subnet Name**
The name of the subnet to be deleted.

**`[-force-update-lif-associations <true>]` - Change the subnet association**
This command will fail if the subnet has ranges containing any existing service processor interface or network interface IP addresses. Setting this value to true will remove the network interface associations with the subnet and allow the command to succeed. However, it will not affect service processor interfaces.

#### Examples

The following example deletes subnet *s1* in IPspace *Default* .

```
cluster1::> network subnet delete -ipspace Default -subnet-name s1
```

## network subnet modify

Modify a layer 3 subnet

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

Modify a subnet.

### Parameters

**`-ipspace <IPspace>` - IPspace Name**
The IPspace to which the subnet belongs.

**`-subnet-name <subnet name>` - Subnet Name**
The name of the subnet to modify.

**`[-subnet <IP Address/Mask>]` - Layer 3 Subnet**
The new address and mask of the subnet.

**`[-gateway <IP Address>]` - Gateway**
The new gateway address.

**`[-ip-ranges {<ipaddr>|<ipaddr>-<ipaddr>}]` - IP Addresses or IP Address Ranges**
The new IP ranges for this subnet.

**`[-force-update-lif-associations <true>]` - Change the subnet association**
This command will fail if any existing service processor interfaces or network interfaces are using IP addresses in the IP ranges being added. It will also fail if any existing service processor interfaces or network interfaces are using IP addresses in the IP ranges being removed. Using this parameter will associate the interfaces with the IP addresses in the ranges being added to the subnet. It will also remove the subnet's association with the interfaces with IP addresses in the IP ranges being removed and will allow the command to succeed.

### Examples

The following example modifies the subnet address and gateway.

```
cluster1::> network subnet modify -ipspace Default -subnet-name s1 -subnet
192.168.2.0/24 -gateway 192.168.2.1
```

## network subnet remove-ranges

Remove address ranges from a subnet

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

Remove address ranges from a subnet.

#### Parameters

**`-ipspace <IPspace>` - IPspace Name**

The IPspace in which the range resides.

**`-subnet-name <subnet name>` - Subnet Name**

The name of the subnet.

**`-ip-ranges {<ipaddr>|<ipaddr>-<ipaddr>}` - IP Ranges**

IP ranges to remove.

**`[-force-update-lif-associations <true>]` - Force Update LIF Associations**

This command will fail if any existing service processor interfaces or network interfaces are using IP addresses in the ranges provided. Using this parameter will remove the subnet's association with those interfaces and allow the command to succeed.

#### Examples

The following example removes an address range with starting address of *`10.98.1.1`* from subnet *`s1`* in IPspace *`Default`* .

```
cluster1::> network subnet remove-ranges -ipspace Default -subnet-name s1
    -ip-ranges "10.98.1.1-10.98.1.30"
```

## network subnet rename

Rename a layer 3 subnet

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

Rename a Subnet.

#### Parameters

**`-ipspace <IPspace>` - IPspace Name**

The IPspace to which the subnet belongs.

**`-subnet-name <subnet name>` - Subnet Name**

The name of the subnet to rename.

**`-new-name <text>` - New Name**

The new name for the subnet.

#### Examples

The following example renames subnet *`s1`* to *`s3`* .

```
cluster1::> network subnet rename -ipspace Default -subnet s1 -new-name s3
```

## network subnet show

Display subnet information

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

Display subnet information.

### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-ipspace <IPspace>] - IPspace Name**

Selects the subnets that match the given IPspace name.

**[-subnet-name <subnet name>] - Subnet Name**

Selects the subnets that match the given subnet name.

**[-broadcast-domain <Broadcast Domain>] - Broadcast Domain**

Selects the subnets that match the given broadcast domain name.

**[-subnet <IP Address/Mask>] - Layer 3 Subnet**

Selects the subnets that match the given address and mask.

**[-gateway <IP Address>] - Gateway**

Selects the subnets that match the given gateway address.

**[-ip-ranges {<ipaddr>|<ipaddr>-<ipaddr>}] - IP Addresses or IP Address Ranges**

Selects the subnets that match the given IP range.

**[-total-count <integer>] - Total Address Count**

Selects the subnets that match the given total address count.

**[-used-count <integer>] - Used Address Count**

Selects the subnets that match the given number of addresses allocated.

**[-available-count <integer>] - Available Address Count**

Selects the subnets that match the given number of addresses available.

#### Examples

The following example displays general information about the subnets.

```
cluster1::> network subnet show
IPspace: Default
     Subnet                      Broadcast                    Avail/
     Name      Subnet            Domain    Gateway            Total   Ranges
     --------- ----------------- --------- ---------------- ---------
---------------
     s4        192.168.4.0/24    bd4       192.168.4.1         5/5
192.168.5.6-192.168.5.10
     s6        192.168.6.0/24    bd4       192.168.6.1         5/5
192.168.6.6-192.168.6.10
IPspace: ips1
     Subnet                      Broadcast                    Avail/
     Name      Subnet            Domain    Gateway            Total   Ranges
     --------- ----------------- --------- ---------------- ---------
---------------
     s10       192.168.6.0/24    bd10      192.168.6.1         0/0    -
     3 entries were displayed.
```

# network test-link commands

## network test-link run-test

Test link bandwidth

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

#### Description

The `network test-link run-test` command runs a performance test between two nodes. The command requires a source node, Vserver, and destination address.

Before executing the `network test-link run-test` command, the network test-link start-server command must be run to start a server on the node hosting the destination LIF. After all tests to that node are complete the network test-link stop-server command must be run to stop the server.

The test results are stored non-persistently and can be viewed using the network test-link show command. Results include input parameters, the bandwidth achieved, and the date and time of the test.

#### Parameters

**`-node {<nodename>|local}` - Node Name (privilege: advanced)**

Use this parameter to specify the node which initiates the test.

**-vserver <vserver> - Vserver (privilege: advanced)**

Use this parameter to specify the Vserver to access the destination LIF. DC (Data Channel) Vserver option is available only in an ONTAP Select or ONTAP Cloud cluster. It is a special vserver that hosts LIFs that are used to mirror data aggregates to partner node.

**-destination <Remote InetAddress> - Destination (privilege: advanced)**

Use this parameter to specify the destination IP address.

### Examples

The following example runs a test between the cluster LIFs, including the start and stop of the server side of the test:

```
cluster1::*> network test-link start-server -node node1

cluster1::*> network test-link run-test -node node2 -vserver Cluster
-destination 172.31.112.173
Node: node2
     Vserver: Cluster
 Destination: 172.31.112.173
Time of Test: 4/22/2016 15:33:18
        MB/s: 41.2678

cluster1::*> network test-link stop-server -node node1
cluster1::*> network test-link show
Node                Vserver             Destination       Time of Test
MB/s
-----------------   -----------------   ---------------
-------------------   ------------
node2               Cluster             172.31.112.173    4/22/2016
15:33:18         41.2678
```

### Related Links

- network test-link start-server
- network test-link stop-server
- network test-link show

## network test-link show

Display test results

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

### Description

The `network test-link show` command displays the results of prior network test-link run-test commands.

The test results are stored non-persistently and can be viewed using the `network test-link show` command. Results include input parameters, the bandwidth achieved, and the date and time of the test.

### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**-node {<nodename>|local} - Node (privilege: advanced)**

Selects the nodes that match this parameter value. Use this parameter to display the test results specific to a node. By default, the test results across all nodes are shown.

**-vserver <vserver> - Vserver (privilege: advanced)**

Use this parameter to display the test results specific to a Vserver. Use DC (Data Channel) Vserver option only in an ONTAP Select or ONTAP Cloud cluster to show network performance of links hosting DC LIFs. DC Vserver is a special Vserver that hosts LIFs that are used to mirror data aggregates to partner node

**[-destination <Remote InetAddress>] - Destination (privilege: advanced)**

Use this parameter to display the test results associated with the specified destination.

**[-timestamp <MM/DD/YYYY HH:MM:SS>] - Time of Test (privilege: advanced)**

Use this parameter to display the test results with the specified timestamp.

**[-bandwidth <double>] - MB/s (privilege: advanced)**

Use this parameter to display the test results only matching the specified bandwidth.

### Examples

The following example runs a test between the cluster LIFs twice and then demonstrates the show command results:

```
cluster1::*> network test-link run-test -node node2 -vserver Cluster
-destination 172.31.112.173
Node: node2
     Vserver: Cluster
 Destination: 172.31.112.173
Time of Test: 4/25/2016 10:37:52
        MB/s: 29.9946
cluster1::*> network test-link run-test -node node2 -vserver Cluster
-destination 172.31.112.173
Node: node2
     Vserver: Cluster
 Destination: 172.31.112.173
Time of Test: 4/25/2016 10:38:32
        MB/s: 39.8192
cluster1::network test-link*> show
Node                Vserver             Destination       Time of Test
MB/s
-----------------   -----------------   ---------------
-------------------   ------------
node2               Cluster             172.31.112.173    4/25/2016
10:38:32         39.8192
```

**Related Links**

- network test-link run-test

## network test-link start-server

Start server for bandwidth test

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

**Description**

The `network test-link start-server` command starts the server side of the `network test-link` test on the designated node.

Only one server at a time can be running for the `network test-link` command on a given node. If the `network test-link start-server` command is issued and a server is already running on the node, then the command is ignored, and the existing server continues to run.

The server started is listening on port 5201.

**Parameters**

**`-node {<nodename>|local}` - Node Name (privilege: advanced)**

Use this parameter to specify the node where the server is to be started.

#### Examples

The following example starts a server:

```
cluster1::*> network test-link start-server -node node1
```

### network test-link stop-server

Stop server for bandwidth test

**Availability:** This command is available to *cluster* administrators at the *advanced* privilege level.

#### Description

The `network test-link stop-server` command stops the `network test-link` server running on the designated node.

#### Parameters

**`-node {<nodename>|local}` - Node Name (privilege: advanced)**

Use this parameter to specify the node where the server is to be stopped.

#### Examples

The following example starts a server and stops it:

```
cluster1::*> network test-link start-server -node node1

cluster1::*> network test-link stop-server -node node1
```

## network trace commands

### network trace show

Show running network trace instances

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

The `network trace show` command shows currently running packet traces on a matching node.

#### Parameters

**{ `[-fields <fieldname>,…]`**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| [-instance ] }**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-node {<nodename>|local}] - Node Name**

Use this parameter optionally to show the details of running packet traces on a matching node.

**[-port {<netport>|<ifgrp>}] - Port**

Use this parameter optionally to show the details of running packet trace on a matching network interface.

#### Examples

The following example shows the details of running packet traces on nodes "node1" and "node2":

```
cluster1::> network trace show
Node       Port
---------- --------
node1
           e0a
node2
           e0c
```

## network trace start

trace start

**Availability:** This command is available to *cluster* and *Vserver* administrators at the *admin* privilege level.

#### Description

The `network trace start` command starts packet tracing with the given parameters. The best practice is to temporarily disable Snapshots on the root volume while the network trace is running.

#### Parameters

**-node {<nodename>|local} - Node Name**

Use this parameter to specify the node on which the packet trace should run.

**-port {<netport>|<ifgrp>} - Port**

Use this parameter to specify the network interface for packet tracing.

**[-address <IP Address>] - IP Address**

Use this parameter to optionally specify the address for packet tracing.

**[-protocol-port <integer>] - Protocol Port Number**

Use this parameter to optionally specify the protocol port number for packet tracing.

**[-buffer-size <integer>] - Buffer Size in KB**

Use this parameter to optionally specify the buffer size for packet tracing. The default buffer size is 4 KB.

**[-file-size <integer>] - Trace File Size in MB**

Use this parameter to optionally specify the trace file size for packet tracing. The default trace file size is 1 GB.

**[-rolling-traces <integer>] - Number of Rolling Trace Files**

Use this parameter to optionally specify the number of rolling trace files for packet tracing. The default number of rolling trace files is 2.

#### Examples

The following example starts packet tracing on node "node1" with address "10.98.16.164", network port "e0c", buffer size "10 KB", and protocol port number "10000":

```
cluster1::> network trace start -node node1
    -address 10.98.16.164 -port e0c -buffer-size 10 -protocol-port 10000
```

The following example disables Snapshots on the root volume. Snapshots should be re-enabled after the trace is stopped.

```
cluster1::> node run -node node1 vol options vol0 nosnap on
```

## network trace stop

### Stop an active network trace

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

The `network trace stop` command stops a running packet trace on a given network interface. The trace files could be located in /mroot/etc/log/packet_traces/.

#### Parameters

**-node {<nodename>|local} - Node Name**

Use this parameter to specify the node on which the packet tracing must be stopped.

**-port {<netport>|<ifgrp>} - Port**

Use this parameter to specify the network interface on which the packet tracing must be stopped.

#### Examples

The following example stops a packet trace on network interface "e0a" from node "node1":

```
cluster1::> network trace stop -node node1 -port e0a
```

## network trace file delete

### Delete a network trace file

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

The `network trace file delete` command deletes the network trace file from a matching node.

#### Parameters

**`-node {<nodename>|local}` - Node Name**

Use this parameter to delete the network trace file from a matching node.

**`-trace-file <text>` - Trace File**

Use this parameter to specify the network trace file to be deleted.

#### Examples

The following example deletes the list of network trace files from node "node1" using wildcard pattern:

```
cluster1::> network trace file delete -node node1 -trace-file *
```

## network trace file show

### Show list of network trace files

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

The `network trace file show` command shows the list of network trace files located in the /mroot/etc/log/packet_traces/ directory.

#### Parameters

**`{ [-fields <fieldname>,…]`**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**`| [-instance ] }`**

If you specify the `-instance` parameter, the command displays detailed information about all fields.

**[-node {<nodename>|local}] - Node Name**

Use this parameter to show the list of traces files of a matching node.

**[-trace-file <text>] - Trace File**

Use this parameter optionally to show the list of trace files with a matching trace-file name.

#### Examples

The following example shows the list of trace files on nodes "node1" and "node2":

```
cluster1::> network trace file show
Node            Trace File
--------------- ------------------------------
node1
                e0a_20170314_115624.trc0
node2
                e0c_20170314_115624.trc0
```

# network tuning commands

## network tuning icmp modify

Modify ICMP tuning options

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

This command displays options which can be used to fine tune icmp protocol behavior.

#### Parameters

**-node {<nodename>|local} - Node**

Sets this parameter to indicate on which node the ICMP tuning options are modified.

**[-is-drop-redirect-enabled {true|false}] - Drop redirect ICMP**

Sets this parameter to drop redirect ICMP message.

**[-tx-icmp-limit <integer>] - Maximum number of ICMP packets sent per second**

Sets the maximum number of ICMP messages including TCP RSTs can be sent per second.

**[-redirect-timeout <integer>] - Maximum seconds for route redirect timeout**

Sets this parameter to indicate the number of seconds after which the route is deleted. Value of zero means infinity. The default value is 300 seconds.

#### Examples

```
cluster1::> network tuning icmp modify -node node1 -is-drop-redirect
-enabled false
```

## network tuning icmp show

Show ICMP tuning options

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

This command displays the current state of the ICMP tuning options for the given node.

### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| [-instance ] }**

Displays all ICMP tuning options.

**[-node {<nodename>|local}] - Node**

Specifies the node for which the ICMP tuning options are displayed.

**[-is-drop-redirect-enabled {true|false}] - Drop redirect ICMP**

Displays all entries that match the "is-drop-redirect-enabled" value.

**[-tx-icmp-limit <integer>] - Maximum number of ICMP packets sent per second**

Displays all entries that match the "tx-icmp-limit" value.

**[-redirect-timeout <integer>] - Maximum seconds for route redirect timeout**

Displays all the entries that match the "redirect-timeout" value.

### Examples

```
cluster1::> network tuning icmp show
Drop Redirect Maximum ICMP      Redirect Timeout
Node      ICMP          Sends per Second  in Seconds
--------- ------------- ----------------  ----------------
node1
          true          100               300
```

## network tuning icmp6 modify

Modify ICMPv6 tuning options

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

This command displays options which can be used to fine tune icmpv6 protocol behavior.

### Parameters

**`-node {<nodename>|local}` - Node**

Sets this parameter to indicate on which node the ICMPv6 tuning options are modified.

**`[-is-v6-redirect-accepted {true|false}]` - Accept redirects via ICMPv6**

Sets this parameter to indicate whether or not redirect ICMPv6 messages are accepted.

**`[-redirect-v6-timeout <integer>]` - Maximum seconds for route redirect timeout**

Sets this parameter to indicate the number of seconds after which the route is deleted. Value of zero means infinity. The default value is 300 seconds.

**`[-tx-icmp6-err-limit <integer>]` - Maximum number of ICMPv6 error messages sent per second**

Sets the maximum number of ICMPv6 error messages that can be sent per second.

### Examples

```
cluster1::> network tuning icmp6 modify -node node1 -is-v6-redirect
-accepted false
```

## network tuning icmp6 show

Show ICMPv6 tuning options

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

### Description

This command displays the current state of the ICMPv6 tuning options for the given node.

### Parameters

**`{ [-fields <fieldname>,…]`**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**`| [-instance ] }`**

Displays all ICMPv6 tuning options.

**`[-node {<nodename>|local}]` - Node**

Specifies the node for which the ICMPv6 tuning options are displayed.

**[-is-v6-redirect-accepted {true|false}] - Accept redirects via ICMPv6**

Displays all entries that match the "is-v6-redirect-accepted" value.

**[-redirect-v6-timeout <integer>] - Maximum seconds for route redirect timeout**

Displays all the entries that match the "redirect-v6-timeout" value.

**[-tx-icmp6-err-limit <integer>] - Maximum number of ICMPv6 error messages sent per second**

Displays all entries that match the "tx-icmp6-err-limit" value.

#### Examples

```
cluster1::> network tuning icmp6 show
Accept Redirect Maximum ICMPv6 Error Redirect Timeout
Node      ICMPv6          Sends per Second     in Seconds
--------- --------------- -------------------- ----------------
node1
          true            100                  300
```

## network tuning tcp modify

### Modify TCP tuning options

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

This commands sets TCP tuning options on the node.

#### Parameters

**-node {<nodename>|local} - Node**

Indicates on which node the TCP tuning options will be modified.

**[-is-path-mtu-discovery-enabled {true|false}] - Path MTU discovery enabled**

Enables path MTU discovery feature.

**[-is-rfc3465-enabled {true|false}] - RFC3465 enabled**

Enables the rfc3465 feature.

**[-max-cwnd-increment <integer>] - Maximum congestion window segments incrementation**

Sets the maximum congestion window increment segements during slow start.

**[-is-rfc3390-enabled {true|false}] - RFC3390 enabled**

Enables the rfc3390 feature.

**[-is-sack-enabled {true|false}] - SACK support enabled**

Enables the selective ACK feature.

#### Examples

```
cluster1::> network tuning tcp modify -node node1 -is-path-mtu-discovery
-enabled false
```

## network tuning tcp show

Show TCP tuning options

**Availability:** This command is available to *cluster* administrators at the *admin* privilege level.

#### Description

This command displays the current state of the TCP tuning options for the given node.

#### Parameters

**{ [-fields <fieldname>,…]**

If you specify the `-fields <fieldname>, …` parameter, the command output also includes the specified field or fields. You can use '-fields ?' to display the fields to specify.

**| [-instance ] }**

Displays all TCP tuning options.

**[-node {<nodename>|local}] - Node**

Specifies the node for which the TCP tuninng options will be displayed.

**[-is-path-mtu-discovery-enabled {true|false}] - Path MTU discovery enabled**

Displays all entries that match the "is-path-mtu-discovery-enabled" value.

**[-is-rfc3465-enabled {true|false}] - RFC3465 enabled**

Displays all entries that match the "is-rfc3465-enabled" value.

**[-max-cwnd-increment <integer>] - Maximum congestion window segments incrementation**

Displays all entries that match the "max-cwnd-increment" value.

**[-is-rfc3390-enabled {true|false}] - RFC3390 enabled**

Displays all entries that match the "is-rfc3390-enabled" value.

**[-is-sack-enabled {true|false}] - SACK support enabled**

Displays all entries that match the "is-sack-enabled" value.

#### Examples

```
cluster1::> network tuning tcp show
           Path MTU            Maximum                     Selective
Node       Discovery  RFC3465 Congestion Window RFC3390 Ack
           Enabled    Enabled Incrementation    Enabled Enabled
---------- ---------  ------- ----------------- ------- --------
node1
           true       true    2                 true    true
```

## Copyright information

Copyright © 2026 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

## Trademark information

NETAPP, the NETAPP logo, and the marks listed at http://www.netapp.com/TM are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.