



# **network interface commands**

## **ONTAP 9.10.1 commands**

NetApp  
August 29, 2024

# Table of Contents

- network interface commands . . . . . 1
  - network interface create . . . . . 1
  - network interface delete . . . . . 5
  - network interface migrate-all . . . . . 6
  - network interface migrate . . . . . 7
  - network interface modify . . . . . 8
  - network interface rename . . . . . 12
  - network interface revert . . . . . 12
  - network interface show . . . . . 13
  - network interface start-cluster-check . . . . . 19
  - network interface capacity show . . . . . 20
  - network interface capacity details show . . . . . 21
  - network interface check cluster-connectivity show . . . . . 22
  - network interface check cluster-connectivity start . . . . . 23
  - network interface check failover show . . . . . 24
  - network interface dns-lb-stats show . . . . . 27
  - network interface failover-groups add-targets . . . . . 28
  - network interface failover-groups create . . . . . 29
  - network interface failover-groups delete . . . . . 30
  - network interface failover-groups modify . . . . . 30
  - network interface failover-groups remove-targets . . . . . 31
  - network interface failover-groups rename . . . . . 31
  - network interface failover-groups show . . . . . 32
  - network interface lif-weights show . . . . . 33
  - network interface service show . . . . . 34
  - network interface service-policy add-service . . . . . 35
  - network interface service-policy clone . . . . . 37
  - network interface service-policy create . . . . . 39
  - network interface service-policy delete . . . . . 41
  - network interface service-policy modify-service . . . . . 42
  - network interface service-policy remove-service . . . . . 44
  - network interface service-policy rename . . . . . 46
  - network interface service-policy restore-defaults . . . . . 47
  - network interface service-policy show . . . . . 49

# 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|nvme-roce|nvme-tcp}`] - 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-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 will always show this parameter as `disabled`.

**`[-firewall-policy <policy>] - (DEPRECATED)-Firewall Policy`**



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 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-revert. 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.

### **[`-rdma-protocols <roce>,...`] - Required RDMA offload protocols**

Defines RDMA offload protocols required by the LIF. A non-empty list will ensure that this LIF can only be moved to network ports that support the specified RDMA offload protocols.

## **Examples**

The following example creates an IPv4 LIF named `datlif1` and an IPv6 LIF named `datlif2` 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 service policy is `default-data-files` and the LIFs are automatically reverted to their home node at startup and under other circumstances. The `datlif1` has the IP address `192.0.2.130` and netmask `255.255.255.128`, and `datlif2` has the IP address `3ffe:1::aaaa` and netmask length of `64`.

```
cluster1::> network interface create -vserver vs0 -lif datlif1 -home-node
node0 -home-port e0c -address 192.0.2.130 -netmask 255.255.255.128
-failover-policy broadcast-domain-wide -service-policy default-data--files
-auto-revert true
cluster1::> network interface create -vserver vs0 -lif datlif2 -home-node
node0 -home-port e0c -address 3ffe:1::aaaa -netmask-length 64 -failover
-policy broadcast-domain-wide -service-policy default-data-files -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>]` - (DEPRECATED)-Firewall Policy**



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 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 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.

### **`[-rdma-protocols <roce>,...] - Required RDMA offload protocols`**

Defines RDMA offload protocols required by the LIF. A non-empty list will ensure that this LIF can only be moved to network ports that support the specified RDMA offload protocols.

## **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|fc|fcache|none|fc-nvme|s3|nvme-roce|nvme-tcp}`] - 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>`] - (DEPRECATED)-Firewall Policy**



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 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.

**`[-rdma-protocols <roce>,...]` - Required RDMA offload protocols**

Use this parameter to display the logical interfaces associated with one or more RDMA protocols.

## Examples

The following example displays general information about all logical interfaces.

```

cluster1::> network interface show

```

Is	Logical	Status	Network	Current	Current
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
```

Vserver	Logical Interface	Home Node:Port	Failover Policy	Failover 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 Supported Limit	IP Data LIF 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
```

Node	IP Data LIF Capacity	IP Data LIF Supported Limit	IP Data LIF 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
```

Node	Date			Source LIF	Destination LIF	Packet 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-connectivity 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, interface 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
```

Vserver	Group	Failover 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
```

Vserver	DNS Zone	Network 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 vservers 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 vservers 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 vservers 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.
```

## Copyright information

Copyright © 2024 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.