



## **cluster commands**

### ONTAP commands

NetApp  
February 04, 2026

# Table of Contents

- cluster commands ..... 1
  - cluster add-node ..... 1
    - Description ..... 1
    - Parameters ..... 1
    - Examples ..... 1
    - Related Links ..... 2
  - cluster create ..... 2
    - Description ..... 2
    - Parameters ..... 2
    - Examples ..... 2
    - Related Links ..... 3
  - cluster join ..... 3
    - Description ..... 3
    - Parameters ..... 3
    - Examples ..... 3
    - Related Links ..... 4
  - cluster modify ..... 4
    - Description ..... 4
    - Parameters ..... 4
    - Examples ..... 5
  - cluster ping-cluster ..... 5
    - Description ..... 5
    - Parameters ..... 5
    - Examples ..... 5
  - cluster setup ..... 6
    - Description ..... 6
    - Parameters ..... 7
    - Examples ..... 7
    - Related Links ..... 11
  - cluster show ..... 11
    - Description ..... 11
    - Parameters ..... 12
    - Examples ..... 12
  - cluster unjoin ..... 13
    - Description ..... 13
    - Parameters ..... 13
    - Examples ..... 13
- cluster contact-info commands ..... 13
  - cluster contact-info modify ..... 13
  - cluster contact-info show ..... 15
- cluster date commands ..... 16
  - cluster date modify ..... 16
  - cluster date show ..... 17

cluster date zoneinfo load-from-uri . . . . .	19
cluster date zoneinfo show . . . . .	20
cluster ha commands . . . . .	20
cluster ha modify . . . . .	20
cluster ha show . . . . .	21
cluster identity commands . . . . .	21
cluster identity modify . . . . .	21
cluster identity show . . . . .	22
cluster image commands . . . . .	22
cluster image cancel-update . . . . .	22
cluster image pause-update . . . . .	23
cluster image resume-update . . . . .	23
cluster image show-update-history . . . . .	24
cluster image show-update-log-detail . . . . .	25
cluster image show-update-log . . . . .	28
cluster image show-update-progress . . . . .	31
cluster image show . . . . .	35
cluster image update . . . . .	36
cluster image validate . . . . .	38
cluster image package delete . . . . .	39
cluster image package get . . . . .	40
cluster image package show-repository . . . . .	40
cluster kernel-service commands . . . . .	41
cluster kernel-service show . . . . .	41
cluster log-forwarding commands . . . . .	43
cluster log-forwarding create . . . . .	43
cluster log-forwarding delete . . . . .	44
cluster log-forwarding modify . . . . .	45
cluster log-forwarding show . . . . .	46
cluster peer commands . . . . .	47
cluster peer create . . . . .	47
cluster peer delete . . . . .	57
cluster peer modify-local-name . . . . .	57
cluster peer modify . . . . .	58
cluster peer ping . . . . .	62
cluster peer show . . . . .	64
cluster peer connection show . . . . .	71
cluster peer health show . . . . .	73
cluster peer offer cancel . . . . .	76
cluster peer offer modify . . . . .	76
cluster peer offer show . . . . .	77
cluster peer policy modify . . . . .	78
cluster peer policy show . . . . .	79
cluster quorum-service commands . . . . .	80
cluster quorum-service options modify . . . . .	80

cluster quorum-service options show . . . . .	81
cluster ring commands . . . . .	82
cluster ring show . . . . .	82
cluster statistics commands . . . . .	83
cluster statistics show . . . . .	83
cluster time-service commands . . . . .	85
cluster time-service ntp security modify . . . . .	85
cluster time-service ntp security show . . . . .	86
cluster time-service ntp server create . . . . .	86
cluster time-service ntp server delete . . . . .	87
cluster time-service ntp server modify . . . . .	88
cluster time-service ntp server reset . . . . .	89
cluster time-service ntp server show . . . . .	89
cluster time-service ntp status show . . . . .	90

# cluster commands

## cluster add-node

Expand the cluster by discovering and adding new nodes

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

### Description

The `cluster add-node` command discovers and adds new nodes to the cluster. When the `-node-count` parameter is specified, the command attempts to add that many nodes to the cluster. The `-node-ip` parameter can be specified to directly add a node.

The [system node show-discovered](#) command displays all the nodes discovered on the local network.

### Parameters

#### { `-node-count` <integer> - Number of Nodes Being Added

Number of nodes to be added to the cluster. If fewer nodes are discovered, all the discovered nodes are added to the cluster and the command will fail since there are fewer nodes than specified. If more nodes are found than the number specified, the command will fail because there is no way to determine which nodes you intend to add to the cluster.



The `-node-count` parameter is supported on non-shared architecture platforms only.

#### | `-node-ip` <IP Address> - Cluster IP Address of Node }

Cluster IP address of the node to add. When this parameter is provided, the command directly adds the node.

#### [`-foreground` {`true`|`false`}] - Foreground Process

When set to `false` the command runs in the background as a job. The default is `true`, which causes the command to return after the operation completes.

#### [`-allow-mixed-version-join` <`true`>] - Allow a node at a different version to join Cluster

This parameter allows nodes with different, but compatible versions of Data ONTAP to be added to the cluster. A Data ONTAP best practice is to add nodes to the cluster that are of the same Data ONTAP version as the nodes in the cluster, but that may not always be possible.

### Examples

The following example adds a node using `-node-ip`:

```
cluster1:> cluster add-node -node-ip 192.168.1.5
[Job 22] Job succeeded.
```

The following example adds 3 nodes using `-node-count`.

```
cluster1::> cluster add-node -node-count 3
[Job 22] Job succeeded.
```

## Related Links

- [system node show-discovered](#)

# cluster create

Create a cluster

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

## Description

The `cluster create` command creates a cluster with one node. Once you create the cluster, add additional nodes to the cluster by using the [cluster join](#) command.

Note that single-node clusters do not require configuring the cluster network. A cluster network interface must be configured before other nodes can join the cluster.

## Parameters

**[`-license <License Code v2>`] - (DEPRECATED)-Base License**



This parameter is deprecated and may be removed in a future release of Data ONTAP.

Use this optional parameter to specify the base license for the cluster. Obtain this value from your sales or support representative.

**`-clustername <text>` - Cluster Name**

Use this parameter to specify the name of the cluster you are creating.

- The name must contain only the following characters: A-Z, a-z, 0-9, "-" or "\_".
- The first character must be one of the following characters: A-Z or a-z.
- The last character must be one of the following characters: A-Z, a-z or 0-9.
- The maximum supported length is 44 characters.
- The system reserves the following names: "all", "cluster", "local" and "localhost".

**[`-node-count <integer>`] - Node Count**

Use this parameter to specify the number of nodes in the cluster you are creating.

- `-node-count` parameter is supported on non-shared architecture platforms only.

## Examples

The following example creates a cluster named cluster1

```
cluster1::> cluster create -clustername cluster1
```

The following example creates a cluster named `cluster1` with `node-count 4` on a non-shared architecture platform.

```
cluster1::> cluster create -clustername cluster1 -node-count 4
```

## Related Links

- [cluster join](#)

## cluster join

(DEPRECATED)-Join an existing cluster using the specified member's IP address or by cluster name

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

## Description



This command is deprecated and may be removed in a future release of Data ONTAP. Use [cluster add-node](#) from a node in the cluster instead.

The `cluster join` command adds a node to an existing cluster. Use the [cluster create](#) command to create a cluster if one does not already exist.

Note that a cluster network interface must be configured for the cluster before other nodes can join the cluster.

## Parameters

**{ -clusteripaddr <IP Address> - IP Address of a Cluster Interface from a Node in the Cluster**

Use this parameter to specify the IP address of a cluster interface. This must be the IP address of a cluster interface of a node that is already in the cluster. This parameter is mutually exclusive with the `-cluster -name` parameter.

**| -cluster-name <text> - Cluster Name of the Cluster to Join }**

Deprecated. Use this parameter to specify the name of an existing cluster to join.

**[-allow-mixed-version-join <true>] - Allow a node at a different version to join Cluster**

This parameter allows nodes with different, but compatible versions of Data ONTAP to join the cluster. A Data ONTAP best practice is to join nodes to the cluster that are of the same Data ONTAP version as the nodes in the cluster, but that may not always be possible.

## Examples

The following example joins the local node to a cluster. The IP address `192.0.2.66` is the address of a cluster interface of a node that already belongs to the cluster.

```
node::> cluster join -clusteripaddr 192.0.2.66
```

## Related Links

- [cluster add-node](#)
- [cluster create](#)

# cluster modify

Modify cluster node membership attributes

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

## Description

The `cluster modify` command modifies the cluster attributes of a node, including its eligibility to participate in the cluster.

At the advanced privilege level, you can use the command to specify whether a node holds epsilon. Epsilon is an extra fractional vote that enables quorum to form using slightly weaker requirements. For example, two out of four eligible nodes are sufficient to form quorum if one of those two nodes holds epsilon.

## Parameters

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

Use this parameter to specify the name of the node to modify. If you do not specify a node, the command runs on the local node.

### **[-epsilon {true|false}] - Epsilon (privilege: advanced)**

Use this parameter with the value `true` to specify that the node holds Epsilon in the cluster. Use this parameter with the value `false` to specify that the node does not hold Epsilon in the cluster. In a cluster, only one node can be designated as Epsilon at any given time. You can designate a node as Epsilon to add weight to its voting in a cluster with an even number of nodes.

### **[-eligibility {true|false}] - Eligibility (privilege: advanced)**

Use this parameter with the value `true` to specify that the node is eligible to participate in the cluster. Use this parameter with the value `false` to specify that the node is not eligible to participate in the cluster.

If you modify a node as ineligible to participate in the cluster, the command prompts you for confirmation before it runs.

### **[-skip-quorum-check-before-ineligible <true>] - Skip Quorum Check Before Setting Node Ineligible (privilege: advanced)**

If this parameter is specified, quorum checks will be skipped prior to setting a node ineligible. When setting a node to ineligible, the operation will continue even if there is a possible data outage due to a quorum issue.



## Examples

This example modifies a node to make it eligible to participate in the cluster.

```
cluster1:*> cluster modify -node node3 -eligibility true
```

The following example removes epsilon from the node named node0 and adds it to the node named node1:

```
cluster1:*> cluster modify -node node0 -epsilon false
cluster1:*> cluster modify -node node1 -epsilon true
```

## cluster ping-cluster

Ping remote cluster interfaces and perform RPC server check

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

### Description

The `cluster ping-cluster` command probes network connectivity to remote cluster interfaces, and performs an RPC server check.

### Parameters

**-node <nodename> - Node (privilege: advanced)**

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

**[-use-sitelist {true|false}] - Use Sitelist for Cluster Interfaces (privilege: advanced)**

Use this parameter with the value `true` to specify that the command use the sitelist to determine any incomplete cluster IP information. Use this parameter with the value `false` to specify that the command not use the sitelist.

**[-skip-rpccheck {true|false}] - Skip RPC Server Check (privilege: advanced)**

Use this parameter with the value `true` to specify that the command not perform the `rpcinfo` check of remote hosts. Use this parameter with the value `false` to specify that the command perform the `rpcinfo` check. The `rpcinfo` check checks the status of the RPC servers on the remote hosts. By default, the `rpcinfo` check runs on the program number of the portmapper. Use the `-rpc-prognum` parameter to override this default.

**[-rpc-prognum <integer>] - RPC Server to Check (privilege: advanced)**

Use this parameter to override default behavior and run the `rpcinfo` check on the program number you specify. By default, the `rpcinfo` check runs on the program number of the portmapper.

## Examples

The following example shows typical output for this command.

```

cluster1::*> cluster ping-cluster -node node1
Host is node1
Getting addresses from network interface table...
Local = 10.254.231.102  10.254.91.42
Remote = 10.254.42.25   10.254.16.228
Ping status:
....
Basic connectivity succeeds on 4 path(s)
Basic connectivity fails on 0 path(s)
.....
Detected 1500 byte MTU on 4 path(s):
  Local 10.254.231.102 to Remote 10.254.16.228
  Local 10.254.231.102 to Remote 10.254.42.25
  Local 10.254.91.42 to Remote 10.254.16.228
  Local 10.254.91.42 to Remote 10.254.42.25
Larger than PMTU communication succeeds on 4 path(s)
RPC status:
2 paths up, 0 paths down (tcp check)
2 paths up, 0 paths down (udp check)

```

## cluster setup

### Setup wizard

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

### Description



Use of this command to join a node to an existing cluster is deprecated and might be removed in a future release of Data ONTAP. From a node in the cluster use the [system node show-discovered](#) command and then use the [cluster add-node](#) command.

The `cluster setup` command runs the cluster setup wizard, which can be used to either create a cluster or join a node to an existing cluster. When you run the cluster setup wizard, enter the appropriate information at the prompts. You will be asked to provide the following information to create a cluster:

- Node management interface port, IP address, netmask, default gateway
- Cluster name
- Cluster base license key



This parameter has been deprecated. It may be removed from a future release of Data ONTAP.

- Feature license keys
- Cluster administrator's password
- Cluster management interface port, IP address, netmask, default gateway

- DNS domain names
- Name server IP addresses
- Location

You will be asked to provide the following information to join a cluster:

- Node management interface port, IP address, netmask, default gateway
- Cluster IP address

The cluster management interface is used for managing the cluster. It provides one IP address to manage the cluster and will fail over to another node, if necessary. This is the preferred IP address for managing the cluster, but you can also manage the cluster by logging in to the node management IP address of a node in the cluster. Since the cluster management interface must be able to fail over, the port role for the interface must be "data" and typically the best choice for an IP address is one on the data network. The node management interface will not fail over, so an IP address on the management network and a port with the role "node management" is the best choice. Alternatively, you can assign an IP address on the data network to the cluster management interface - if that is better in your network topology - but the port must be a data port. The two examples below illustrate the cluster create and cluster join operations, respectively.

## Parameters

## Examples

The following example shows the create option of `cluster setup`.

```
node::> cluster setup
Welcome to the cluster setup wizard.

You can enter the following commands at any time:
  "help" or "?" - if you want to have a question clarified,
  "back" - if you want to change previously answered questions, and
  "exit" or "quit" - if you want to quit the cluster setup wizard.
  Any changes you made before quitting will be saved.

You can return to cluster setup at any time by typing "cluster setup".
To accept a default or omit a question, do not enter a value.

This system will send event messages and periodic reports to NetApp
Technical
Support. To disable this feature, enter
autosupport modify -support disable
within 24 hours.

Enabling AutoSupport can significantly speed problem determination and
resolution should a problem occur on your system.
For further information on AutoSupport, see:
http://support.netapp.com/autosupport/
```

Type yes to confirm and continue {yes}: yes

Enter the node management interface port [e0c]:

Enter the node management interface IP address: 192.0.2.66

Enter the node management interface netmask: 255.255.255.192

Enter the node management interface default gateway: 192.0.2.1

The node management interface has been modified to use port e0c with IP address 192.0.2.66.

Use your web browser to complete cluster setup by accessing <https://192.0.2.66>

Otherwise, press Enter to complete cluster setup using the command line interface:

Do you want to create a new cluster or join an existing cluster? {create, join}:

create

Do you intend for this node to be used as a single node cluster? {yes, no} [no]:

Existing cluster interface configuration found:

Port	MTU	IP	Netmask
e0a	9000	169.254.21.189	255.255.0.0
e0b	9000	169.254.29.73	255.255.0.0

Do you want to use this configuration? {yes, no} [yes]:

Enter the cluster administrator's (username "admin") password:

Retype the password:

Step 1 of 5: Create a Cluster

You can type "back", "exit", or "help" at any question.

Enter the cluster name: cluster1

Creating cluster cluster1

Starting cluster support services .

Cluster cluster1 has been created.

Step 2 of 5: Add Feature License Keys

You can type "back", "exit", or "help" at any question.

Enter an additional license key []:

Step 3 of 5: Set Up a Vserver for Cluster Administration

You can type "back", "exit", or "help" at any question.

Enter the cluster management interface port [e0d]:

```
Enter the cluster management interface IP address: 192.0.2.60
Enter the cluster management interface netmask: 255.255.255.192
Enter the cluster management interface default gateway [192.0.2.1]:
A cluster management interface on port e0d with IP address 192.0.2.60 has
been created. You can use this address to connect to and manage the
cluster.
```

```
Enter the DNS domain names: data.example.com
Enter the name server IP addresses: 192.0.2.147
DNS lookup for the admin Vserver will use the data.example.com domain.
```

```
Step 4 of 5: Configure Storage Failover (SFO)
You can type "back", "exit", or "help" at any question.
SFO is licensed.
SFO will be enabled when the partner joins the cluster.
Step 5 of 5: Set Up the Node
You can type "back", "exit", or "help" at any question.
```

```
Where is the controller located []: Sunnyvale
Cluster "cluster1" has been created.
```

To complete cluster setup, you must join each additional node to the cluster by running "system node show-discovered" and "cluster add-node" from a node in the cluster.

To complete system configuration, you can use either OnCommand System Manager or the Data ONTAP command-line interface.

To access OnCommand System Manager, point your web browser to the cluster management IP address (<https://192.0.2.60>).

To access the command-line interface, connect to the cluster management IP address (for example, `ssh admin@192.0.2.60`).

```
cluster1::>
```

An example of using `cluster setup` to join a cluster is shown below.

```
node::> cluster setup
Welcome to the cluster setup wizard.
```

You can enter the following commands at any time:

- "help" or "?" - if you want to have a question clarified,

"back" - if you want to change previously answered questions, and  
"exit" or "quit" - if you want to quit the cluster setup wizard.  
Any changes you made before quitting will be saved.

You can return to cluster setup at any time by typing "cluster setup".  
To accept a default or omit a question, do not enter a value.

This system will send event messages and periodic reports to NetApp  
Technical  
Support. To disable this feature, enter  
autosupport modify -support disable  
within 24 hours.

Enabling AutoSupport can significantly speed problem determination and  
resolution should a problem occur on your system.  
For further information on AutoSupport, see:  
<http://support.netapp.com/autosupport/>

Type yes to confirm and continue {yes}: yes

Enter the node management interface port [e0c]:  
Enter the node management interface IP address: 192.0.2.67  
Enter the node management interface netmask: 255.255.255.192  
Enter the node management interface default gateway: 192.0.2.1  
A node management interface on port e0c with IP address 192.0.2.67 has  
been created.

Use your web browser to complete cluster setup by accessing  
<https://192.0.2.67>

Otherwise, press Enter to complete cluster setup using the command line  
interface:

Do you want to create a new cluster or join an existing cluster? {create,  
join}:

join

Existing cluster interface configuration found:

Port	MTU	IP	Netmask
e0a	9000	169.254.31.170	255.255.0.0
e0b	9000	169.254.115.61	255.255.0.0

Do you want to use this configuration? {yes, no} [yes]:

Step 1 of 3: Join an Existing Cluster

You can type "back", "exit", or "help" at any question.

Enter the IP address of an interface on the private cluster network from  
the

```
cluster you want to join: 169.254.115.8
```

```
Joining cluster at address 169.254.115.8
```

```
This node has joined the cluster cluster1.
```

```
Step 2 of 3: Configure Storage Failover (SFO)
```

```
You can type "back", "exit", or "help" at any question.
```

```
SFO is licensed.
```

```
SFO will be enabled when the partner joins the cluster.
```

```
Step 3 of 3: Set Up the Node
```

```
You can type "back", "exit", or "help" at any question.
```

```
This node has been joined to cluster "cluster1".
```

To complete cluster setup, you must join each additional node to the cluster

by running "system node show-discovered" and "cluster add-node" from a node in the cluster.

To complete system configuration, you can use either OnCommand System Manager

or the Data ONTAP command-line interface.

To access OnCommand System Manager, point your web browser to the cluster management IP address (<https://192.0.2.60>).

To access the command-line interface, connect to the cluster management IP address (for example, `ssh admin@192.0.2.60`).

```
cluster1::>
```

## Related Links

- [system node show-discovered](#)
- [cluster add-node](#)

## cluster show

Display cluster node members

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

## Description

The `cluster show` command displays information about the nodes in a cluster.

## Parameters

**{ [-fields <fieldname>,...]**

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

**| [-instance ] }**

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

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

Selects the nodes that match this parameter value.

**[-node-uuid <UUID>] - UUID (privilege: advanced)**

Selects the nodes that match this parameter value.

**[-epsilon {true|false}] - Epsilon (privilege: advanced)**

Selects the nodes that match this parameter value. In a cluster, only one node can be designated as Epsilon at any given time. You can designate a node as Epsilon to add weight to its voting in a cluster with an even number of nodes.

**[-eligibility {true|false}] - Eligibility**

Selects the nodes that match this parameter value (true means eligible to participate in the cluster).

**[-health {true|false}] - Health**

Selects the nodes that match this parameter value (true means online).

## Examples

The following example displays information about all nodes in the cluster:

```
cluster1::> cluster show
Node           Health  Eligibility
-----
node0          true    true
node1          true    true
node2          true    true
node3          true    true
```

The following example displays information about the node named node1:

```
cluster1::> cluster show -node node1
Node: node1
Eligibility: true
Health: true
```



# cluster unjoin

Unjoin or remove a node from the cluster

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

## Description

The `cluster unjoin` command removes a node from a cluster.

Before you can remove a node from a cluster, you must shut down all of the node's shared resources, such as virtual interfaces to clients. If any of a node's shared resources are still active, the command fails. The failure message will display which active resources must be shut down before the node can be removed from the cluster.

## Parameters

**-node <nodename> - Node to unjoin (privilege: advanced)**

Use this parameter to specify the name of the node to remove from the cluster.

**[-skip-quorum-check-before-unjoin <true>] - Skip Quorum Check Before Unjoin (privilege: advanced)**

If this parameter is specified, quorum checks will be skipped prior to the unjoin. The operation will continue even if there is a possible data outage due to a quorum issue.

**[-skip-last-low-version-node-check <true>] - Skip the check that prevents unjoining the last low versioned node (privilege: advanced)**

This parameter allows the node with lowest version of Data ONTAP to be unjoined from the cluster.

## Examples

The following example shows how to remove the node named `node4` from the cluster .

```
cluster1::*> cluster unjoin -node node4
```

# cluster contact-info commands

## cluster contact-info modify

Modify contact information for the cluster

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

## Description

The `cluster contact-info modify` command modifies contact information for the cluster administrators. If any values contain spaces, you must enclose those values in quotes.

Use the [cluster contact-info show](#) command to display contact information for the cluster administrators.

## Parameters

### **[-primary-name <text>] - Name of Primary Contact**

Use this parameter to specify the name of the primary contact.

### **[-primary-phone <text>] - Phone Number of Primary Contact**

Use this parameter to specify the phone number of the primary contact.

### **[-primary-alt-phone <text>] - Alternate Phone Number of Primary Contact**

Use this parameter to specify the alternate phone number of the primary contact.

### **[-primary-email <text>] - Email Address or User ID of Primary Contact**

Use this parameter to specify the email address of the primary contact.

### **[-secondary-name <text>] - Name of Secondary Contact**

Use this parameter to specify the name of the secondary contact.

### **[-secondary-phone <text>] - Phone Number of Secondary Contact**

Use this parameter to specify the phone number of the secondary contact.

### **[-secondary-alt-phone <text>] - Alternate Phone Number of Secondary Contact**

Use this parameter to specify the alternate phone number of the secondary contact.

### **[-secondary-email <text>] - Email Address or User ID of Secondary Contact**

Use this parameter to specify the email address of the secondary contact.

### **[-business-name <text>] - Business Name**

Use this parameter to specify the name of the business responsible for this cluster.

### **[-address <text>] - Business Address**

Use this parameter to specify the street address of the business responsible for this cluster.

### **[-city <text>] - City Where Business Resides**

Use this parameter to specify the name of the city in which the business is located.

### **[-state <text>] - State Where Business Resides**

Use this parameter to specify the name of the state or province in which the business is located.

### **[-country <Country Code>] - 2-Character Country Code**

Use this parameter to specify the 2-character country code of the country in which the business is located.

### **[-zip-code <text>] - Postal Code Where Business Resides**

Use this parameter to specify the postal or ZIP code area in which the business is located.

## Examples

The following example changes the name and phone numbers of the secondary contact person for the cluster.

```
cluster1::> cluster contact-info modify -secondary-name "John Doe"
-secondary-phone 123.555.0156 -secondary-alt-phone 123.555.0178
```

The following example changes the mailing address of the business responsible for the cluster.

```
cluster1::> cluster contact-info modify -address "123 Example Avenue"
-city Exampleville -state "New Example" -zip-code 99999 -country US
```

## Related Links

- [cluster contact-info show](#)

## cluster contact-info show

Display contact information for the cluster

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

## Description

The `cluster contact-info show` command displays contact information for the cluster administrators.

## Examples

The following example shows example output for this command.

```
cluster1::> cluster contact-info show
Name of Primary Contact : Richard Roe
    Phone Number of Primary Contact : 123.555.0123
    Alternate Phone Number of Primary Contact : 123.555.0145
Email Address or User Id of Primary Contact : roe@example.com
    Name of Secondary Contact : John Doe
    Phone Number of Secondary Contact : 123.555.0167
    Alternate Phone Number of Secondary Contact : 123.555.0189
Email Address or User Id of Secondary Contact : doe@example.com
    Business Name : Example Dot Com
    Business Address : 123 Example Avenue
    City Where Business Resides : Exampleville
    State Where Business Resides : New Example
    2-Character Country Code : US
    Postal Code Where Business Resides : 99999
```

# cluster date commands

## cluster date modify

Modify the current date and time for the nodes in the cluster

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

### Description

The `cluster date modify` command sets the time zone, date, and time on every node in the cluster.

### Parameters

#### **[-timezone <Area/Location Timezone>] - Time Zone**

This parameter sets the timezone, specified in the Olson format.

#### **{ [-date {MM/DD/YYYY HH:MM:SS [{+|-}hh:mm]}] - Date and Time**

This parameter sets the date and time, in the format MM/DD/YYYY HH:MM:SS.

#### **| [-dateandtime <[[[[[cc]yy]mm]dd]hhmm[.ss]]>] - Date and Time**

This parameter sets the date and time information, in the format [[[[[cc]yy]mm]dd]hhmm[.ss]]. The argument for setting the date and time is interpreted as follows:

- cc First 2 digits of the year (e.g., 20 for 2011).
- yy Last 2 digits of year (e.g., 10 for 2010).
- mm Numeric month, a number from 01 to 12.
- dd Day, a number from 01 to 31.
- hh Hour, a number from 00 to 23.
- mm Minute, a number from 00 to 59.
- ss Second, a number from 00 to 59.

If the first two digits of the year are omitted, and the last two digits are greater than 68, a date in the 1900s is used. Otherwise, a date in the 2000s is used. If all four digits of the year are omitted, the default is the current year. If the month or day is omitted, the default is the current month or day, respectively. If the seconds are omitted, the default is set to 00. The system automatically handles the time changes for Daylight Saving and Standard time, and for leap seconds and years.

#### **| [-u, -utcdatetime <[[[[[cc]yy]mm]dd]hhmm[.ss]]>] - UTC Date and Time }**

This parameter sets the date and time information in Coordinated Universal Time (UTC), in the format [[[[[cc]yy]mm]dd]hhmm[.ss]]. `-u` is an alias for `-utcdatetime`. The argument for setting the date and time is interpreted as follows:

- cc First 2 digits of the year (e.g., 20 for 2011).
- yy Last 2 digits of year (e.g., 10 for 2010).
- mm Numeric month, a number from 01 to 12.
- dd Day, a number from 01 to 31.

- hh Hour, a number from 00 to 23.
- mm Minute, a number from 00 to 59.
- ss Second, a number from 00 to 59.

If the first two digits of the year are omitted, and the last two digits are greater than 68, a date in the 1900s is used. Otherwise, a date in the 2000s is used. If all four digits of the year are omitted, the default is the current year. If the month or day is omitted, the default is the current month or day, respectively. If the seconds are omitted, the default is set to 00. Time changes for Daylight Saving and Standard time, and for leap seconds and years, are handled automatically.

## Examples

The following example sets the date and time to January 1 2011, at 1:00 a.m.:

```
cluster1::> cluster date modify -date "01/01/2011 01:00:00"
```

The following example sets the date and time in the UTC format to May 22, 2011, at 09:25:00 a.m.:

```
cluster1::> cluster date modify -u 201105220925.00.
```

## cluster date show

Display the current date and time for the nodes in the cluster

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

### Description

The `cluster date show` command displays the time zone, date, and time settings for one or more nodes in the cluster. By default, the command displays date and time settings for all nodes in the cluster.

### Parameters

{ [-fields <fieldname>,...]

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

| [-utc ]

Displays date and time information in Coordinated Universal Time (UTC).

| [-utcdate ]

Displays date and time information in UTC.

| [-instance ] }

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

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

Selects the nodes that match this parameter value.

**[-timezone <Area/Location Timezone>] - Time Zone**

Selects the nodes that match this parameter value (specified in the Olson format).

**[-date {MM/DD/YYYY HH:MM:SS [{+|-}hh:mm]}] - Date and Time**

Selects the nodes that match this parameter value.

**[-utc-date <MM/DD/YYYY HH:MM:SS>] - UTC Date and Time**

Selects the nodes that match this parameter value.

**[-dateandtime <[[[[[cc]yy]mm]dd]hhmm[.ss]]>] - Date and Time**

Selects the nodes that match this parameter value (interpreted as follows):

- cc First 2 digits of the year (e.g., 20 for 2011).
- yy Last 2 digits of year (e.g., 11 for 2011).
- mm Numeric month, a number from 01 to 12.
- dd Day, a number from 01 to 31.
- hh Hour, a number from 00 to 23.
- mm Minute, a number from 00 to 59.
- ss Second, a number from 00 to 59.

**[-u, -utcdatetime <[[[[[cc]yy]mm]dd]hhmm[.ss]]>] - UTC Date and Time**

-u is used as an alias for -utcdatetime. Selects the nodes that match this parameter value (interpreted as follows):

- cc First 2 digits of the year (e.g., 20 for 2011).
- yy Last 2 digits of year (e.g., 11 for 2011).
- mm Numeric month, a number from 01 to 12.
- dd Day, a number from 01 to 31.
- hh Hour, a number from 00 to 23.
- mm Minute, a number from 00 to 59.
- ss Second, a number from 00 to 59.

**Examples**

The following example displays the date and time settings for all nodes in the cluster:

```
cluster1::> cluster date show
Node          Date          Timezone
-----
node0         10/06/2011 09:35:15 America/New_York
node1         10/06/2011 09:35:15 America/New_York
node2         10/06/2011 09:35:15 America/New_York
node3         10/06/2011 09:35:15 America/New_York
4 entries were displayed.
```

## cluster date zoneinfo load-from-uri

### Load timezone zoneinfo data

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

### Description

The `cluster date zoneinfo load-from-uri` command loads a new set of timezone zoneinfo data to replace the version installed in the cluster. Releases of Data ONTAP software contain the timezone data that is current at the time of release. If a change is made to the timezone between Data ONTAP releases, then an update can be made to the release data. For instance, if a change is made to when daylight saving time is observed for a country then an update to cluster zoneinfo data may be required.

Only zoneinfo files provided by NetApp for use in Data ONTAP should be used with this command.

To update the zoneinfo database do the following:

- Download the required zoneinfo file from the NetApp support website.
- Place the file on a local web server accessible without password from the cluster.
- Execute the `cluster date zoneinfo load-from-uri` command, passing the Universal Resource Identifier (URI) of the file as parameter.



The command need only be executed once for the cluster. The data will be distributed to each node of the cluster.

### Parameters

**-uri { (ftp|http) :// (hostname|IPv4 Address| [' IPv6 Address' ] ) ...} - URI of Timezone Zoneinfo Data**

URI of the new zoneinfo file.

### Examples

The following example loads a new version of the timezone zoneinfo database to the cluster:

```
cluster1::> cluster date zoneinfo load-from-uri
http://www.example.com/ontap_zoneinfo.zip
```

## cluster date zoneinfo show

Display cluster timezone zoneinfo information

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

### Description

Display information about the current timezone zoneinfo data.

### Examples

The following example shows the zoneinfo information for a cluster:

```
cluster1::> cluster date zoneinfo show
Cluster Zoneinfo Version: 2016f
```

## cluster ha commands

### cluster ha modify

Modify high-availability configuration of cluster management services

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

### Description

The `cluster ha modify` command enables or disables cluster high availability in a two-node cluster. Enable high availability when performing some procedures, such as replacing hardware.



This command is required to enable high availability if the cluster only has two nodes. Do not run this command in a cluster that has three or more nodes.



Cluster high availability for two-node clusters differs from the storage failover technology used between two nodes for storage high availability.

### Parameters

#### **[`-configured {true|false}`] - HA Configured**

Use this parameter with the value `true` to enable high availability mode in the cluster. Use this parameter with the value `false` to disable high availability mode in the cluster.

### Examples

The following example enables cluster high availability in a cluster.

```
cluster1::> cluster ha modify -configured true
```



## cluster ha show

Show high-availability configuration status for the cluster

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

### Description

The `cluster ha show` command displays the high-availability status of the cluster. Cluster high-availability mode applies only to two-node clusters.

### Examples

The following example displays the high-availability status for a two-node cluster:

```
cluster1::> cluster ha show
High Availability Configured: true
```

## cluster identity commands

### cluster identity modify

Modify the cluster's attributes

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

### Description

The `cluster identity modify` command changes a cluster's identity information.

### Parameters

#### **[`-name` <Cluster name>] - Cluster Name**

Use this parameter to specify a new name for the cluster.

- The name must contain only the following characters: A-Z, a-z, 0-9, "-" or "\_".
- The first character must be one of the following characters: A-Z or a-z.
- The last character must be one of the following characters: A-Z, a-z or 0-9.
- The maximum supported length is 44 characters.
- The system reserves the following names: "all", "cluster", "local" and "localhost".

#### **[`-location` <text>] - Cluster Location**

Use this parameter to specify the physical location of the cluster. For example, "Lab 5".

#### **[`-contact` <text>] - Cluster Contact**

Use this parameter to specify contact information for the cluster, such as a name or e-mail address.

## Examples

The following example renames the current cluster to *cluster2*:

```
cluster1::> cluster identity modify -name cluster2
```

## cluster identity show

Display the cluster's attributes including Name, Serial Number, Cluster UUID, Location and Contact

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

### Description

The `cluster identity show` command displays the identity information of the cluster.

## Examples

The following example displays the cluster's UUID, name, serial number, location and contact information:

```
cluster1::> cluster identity show
Cluster UUID: 1cd8a442-86d1-11e0-ae1c-123478563412
      Cluster Name: cluster1
Cluster Serial Number: 1-80-123456
      Cluster Location: Lab2
      Cluster Contact: jsmith@example.com

cluster1::>
```

## cluster image commands

### cluster image cancel-update

Cancel an update

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

### Description

The `cluster image cancel-update` command is used to cancel an update that is in either `paused-by-user` or `paused-by-error` state. The update cannot be canceled if it is not in a `paused` state.

## Examples

The following example displays a cancel-update operation:

```
cluster1::> cluster image cancel-update
```

```
Warning: The cancel operation can result in a mixed version  
cluster and/or mixed version HA pair. The cancel  
operation can take several minutes to complete.
```

```
Do you want to proceed with the cancel operation? {y|n}: y
```

```
Info: Canceling update. It may take a few minutes to finish canceling the  
update
```

## cluster image pause-update

### Pause an update

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

### Description

The `cluster image pause-update` command is used to pause a currently running update. The update pauses at the next predefined pause point (for example, after validation, download to the boot device, takeover completion, or giveback completion) which might take some time to reach. When the update reaches the pause point, it transitions into the pause-by-user state.

### Examples

The following example displays pause-update operation:

```
cluster1::> cluster image pause-update
```

```
Info: Pausing update. It may take a few minutes to finish pausing the  
update
```

## cluster image resume-update

### Resume an update

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

### Description

The `cluster image resume-update` command is used to resume an update that is currently paused in paused-by-user or paused-by-error state. If the update is not paused then an error is returned.

### Parameters

**`[-ignore-post-update-checks-failures {true|false}]` - Ignore Post-update-checks Phase Failures (privilege: advanced)**

Specifies whether the post update checks phase warnings and/or errors should be ignored. The default value is false.

## Examples

The following example shows an resume-update operation:

```
cluster1::> cluster image resume-update

Info: Resuming update...
```

## cluster image show-update-history

Display the update history

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

### Description

The `cluster image show-update-history` command displays the update history for each node. By default, the command displays the following information:

- Status
- Package version
- Start time
- Completion time
- Component ID
- Previous version
- Updated version

### 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.

**[-component-id <text>] - Component ID**

Displays updates for the specified component.

**[-start-time <MM/DD/YYYY HH:MM:SS>] - Start Time**

Displays updates with the specified start time.

**[-package-version <text>] - Package Version**

Displays updates for the specified package version.

**[*-status* {successful|canceled|back-out}] - Status**

Displays updates that completed with the specified status.

**[*-completion-time* <MM/DD/YYYY HH:MM:SS>] - Completion Time**

Displays updates with the specified completion time.

**[*-previous-version* <text>] - Previous Version**

Displays updates with the specified previous version.

**[*-updated-version* <text>] - Updated Version**

Displays updates with the specified updated version.

**Examples**

The following example displays history of automated nondisruptive updates:

```

cluster1::> cluster image show-update-history
Package      Start      Completion      Previous
Updated
Status      Version    Time            Time            Component ID    Version
Version
-----
-----
canceled    8.3        2/11/2014      2/11/2014      ssan-3240-      8.3      8.3
              12:05:51    12:05:51      55a
successful  8.3        2/11/2014      2/11/2014      ssan-3240-      8.3      8.3
              14:23:58    15:02:19      55a
successful  8.3        2/13/2014      2/18/2014      ssan-3240-      8.3      8.3
              16:48:42    09:45:30      55a
successful  8.3        2/18/2014      2/18/2014      ssan-3240-      8.3      8.3
              10:33:10    11:02:45      55a
canceled    8.3        2/11/2014      2/11/2014      ssan-3240-      8.3      8.3
              12:05:51    12:05:51      55b
successful  8.3        2/11/2014      2/11/2014      ssan-3240-      8.3      8.3
              14:23:58    15:54:43      55b
successful  8.3        2/13/2014      2/18/2014      ssan-3240-      8.3      8.3
              16:48:42    10:05:02      55b
successful  8.3        2/18/2014      2/18/2014      ssan-3240-      8.3      8.3
              10:33:10    11:22:02      55b
8 entries were displayed.

```

**cluster image show-update-log-detail**

Display detailed information about nondisruptive update events

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

## Description

The `cluster image show-update-log-detail` command displays detailed information about the currently running and previously run nondisruptive update events. By default, the command displays the following information:

- Node
- Transaction ID
- Time stamp
- Destination node
- Task phase
- Task name
- Task status
- Message

## Parameters

**{ [-fields <fieldname>,...]**

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

**| [-instance ] }**

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

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

Displays information only for the specified node.

**[-task-id <integer>] - Task Id (privilege: advanced)**

Displays information only for the specified task ID.

**[-posted-time <MM/DD/YYYY HH:MM:SS>] - Posted Time (privilege: advanced)**

Displays information that occurred at the specified time.

**[-msg-seq-no <integer>] - Message Sequence (privilege: advanced)**

Displays information only for the specified message sequence number.

**[-current-pid <integer>] - Process ID (privilege: advanced)**

Displays information only for the specified process ID.

**[-destination <text>] - Task Target node (privilege: advanced)**

Displays information only for the specified destination node.

**[-ndu-phase {validation|prereq-updates|ontap-updates|package-management|default-phase|post-update-checks}] - Update phase (privilege: advanced)**

Displays information only for the specified phase.

**[-task-name {initialize|mount-image|restart-hm|get-health|run-scripts|unmount-image|clear-alert|post-restart-hm|cleanup-rd|synch-image|do-download-job|do-failover-job|do-giveback-job|check-progress|complete-validation|invalid-task|default-task|do-postupdate-checks-task}] - Task Name (privilege: advanced)**

Displays information only for the specified task name.

**[-status {created|ready-to-run|running|completed|failed|pause\_req|paused|paused-error|cancel\_req|canceled|resume\_req|default\_status}] - Status Of Task (privilege: advanced)**

Displays information only for items with the specified status.

**[-message <text>] - Update Log Message (privilege: advanced)**

Displays information only for items with the specified message.

**[-msg-type <text>] - Type of Message (privilege: advanced)**

Displays information only for items with the specified message type.

**[-src-info <text>] - Source Information (privilege: advanced)**

Displays information only for items for the specified source.

## Examples

The following example displays detailed information automated nondisruptive updates:

```

cluster1::*> cluster image show-update-log-detail
      Time      Dest      Task  Task  Task
Node  TID Stamp   Node  Phase Name  Status Message
-----
node1  15  3/19/    MUM    ontap- initia ready- Created Task
      2014      update lize  to-run
      13:52:38      s
node1  15  3/19/    MUM    ontap- initia runnin Updated Task Status
      2014      update lize  g
      13:52:38      s
node1  16  3/19/    node1  ontap- do-    ready- Created Task
      2014      update downlo to-run
      13:52:38      s      ad-job
node1  16  3/19/    node1  ontap- do-    runnin Updated Task Status
      2014      update downlo g
      13:52:39      s      ad-job
node1  17  3/19/    node2  ontap- do-    ready- Created Task
      2014      update downlo to-run
      13:52:38      s      ad-job
node2  17  3/19/    node2  ontap- do-    runnin Updated Task Status
      2014      update downlo g
      13:52:38      s      ad-job
6 entries were displayed.

```

## cluster image show-update-log

Display the update transaction log

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

### Description

The `cluster image show-update-log` command displays detailed information about the currently running, or previously run nondisruptive updates. By default, the command displays the following information:

- Phase
- Transaction
- Transaction ID
- Component ID
- Time stamp
- Status



## 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.

**[-trans-id <integer>] - Transaction ID (privilege: advanced)**

Displays information for the step associated with the specified transaction ID.

**[-component-id {<nodename>|local}] - Component ID (privilege: advanced)**

Displays information for steps associated with the specified component.

**[-phase {validation|prereq-updates|ontap-updates|package-management|default-phase|post-update-checks}] - Transaction Phase (privilege: advanced)**

Displays information for steps associated with the specified update phase.

**[-trans-name {initialize|mount-image|restart-hm|get-health|run-scripts|unmount-image|clear-alert|post-restart-hm|cleanup-rd|synch-image|do-download-job|do-failover-job|do-giveback-job|check-progress|complete-validation|invalid-task|default-task|do-postupdate-checks-task}] - Transaction Name (privilege: advanced)**

Displays information for steps associated with the specified transaction.

**[-timestamp <MM/DD/YYYY HH:MM:SS>] - Timestamp (privilege: advanced)**

Displays information for steps associated with the specified timestamp.

**[-status {waiting|started|completed|paused-on-error|paused-by-user|pause-pending|cancel-pending|canceled|failed}] - Status (privilege: advanced)**

Displays information for steps matching the specified status.

## Examples

The following example displays information about automated nondisruptive update events:

```

cluster1::*> cluster image show-update-log
Trans
Phase      Transaction  Id    Component Id      Time Stamp  Status
-----
validation initialize    50    MUM                2/18/2014  completed
                                10:32:57
validation mount-image   51    node1              2/18/2014  completed
                                10:32:52
validation mount-image   52    node2              2/18/2014  completed
                                10:32:53
validation get-health   53    MUM                2/18/2014  completed
                                10:32:53
validation run-scripts  54    node1              2/18/2014  completed
                                10:32:53
validation run-scripts  55    node2              2/18/2014  completed
                                10:32:57
validation unmount-    56    node1              2/18/2014  completed
image
validation unmount-    57    node2              2/18/2014  completed
image
validation complete-  58    MUM                2/18/2014  completed
validation
package-   cleanup-    66    node1              3/14/2014  completed
management package
                                09:11:51
package-   cleanup-    67    node2              3/14/2014  completed
management package
                                09:11:51
package-   process-   68    node1              3/14/2014  completed
management package
                                09:13:41
package-   synch-     69    node2              3/14/2014  completed
management
                                09:14:25

13 entries were displayed.

```

## cluster image show-update-progress

Display the update progress

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

### Description

The `cluster image show-update-progress` command displays information about the current state of an update. By default, the command displays the following information:

- Update phase
- Status
- Estimated Duration
- Elapsed Duration

### 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.

**[-ndu-phase {validation|prereq-updates|ontap-updates|package-management|default-phase|post-update-checks}] - Update Phase**

Displays information about the specified update phase.

**[-phase-status {in-progress|waiting|paused-by-user|paused-on-error|completed|canceled|failed|pause-pending|cancel-pending}] - Phase Status**

Displays information about progress matching the specified phase status.

**[-phase-duration <text>] - Phase Duration**

Displays information about progress matching the specified phase duration.

**[-phase-comments <text>] - Phase Comments**

Displays information about progress matching the specified phase comments.

**[-elapsed-duration {<seconds>|[<d> days] <hh>:<mm>[:<ss>]}] - Elapsed duration of the phase**

Displays information about progress matching the specified elapsed duration.

**[-estimated-duration {<seconds>|[<d> days] <hh>:<mm>[:<ss>]}] - Estimated duration of the phase**

Displays information about progress matching the specified estimated duration.

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

Displays information about progress matching the specified phase description.

**[-subsystem-name <text>] - Subsystem Name**

Displays information about progress matching the specified subsystem name.

**[-subsystem-status <text>] - Subsystem Status**

Displays information about progress matching the specified subsystem status.

**[-subsystem-details <text>] - Subsystem Details**

Displays information about progress matching the specified subsystem details.

**[-subsystem-action <text>] - Subsystem Action**

Displays information about progress matching the specified subsystem action.

**Examples**

The following example shows the automated nondisruptive update of two nodes, nodeA and nodeB. In this case, nodeA's update is waiting, nodeB's update is in progress. nodeB's giveback operation is in progress.

```

cluster1::> cluster image show-update-progress
Estimated      Elapsed
Update Phase   Status          Duration        Duration
-----
Pre-update checks    completed        00:10:00        00:00:02
Data ONTAP updates  in-progress     01:23:00        00:32:07

Details:

Node name      Status          Status Description
-----
nodeA          waiting
nodeB          in-progress     Performing giveback operation.
3 entries were displayed.

cluster1::>

```

The following example shows the automated nondisruptive update of two nodes, nodeA and nodeB. In this case, automated nondisruptive update is paused-on-error in "Data ONTAP updates" phase. nodeA's update is waiting, nodeB's update is failed. "Status Description" show nodeB's error and action.

```
cluster1:> cluster image show-update-progress
```

Estimated	Elapsed		
Update Phase	Status	Duration	Duration
Pre-update checks	completed	00:10:00	00:00:02
Data ONTAP updates	paused-on-error	00:49:00	00:05:21

Details:

Node name	Status	Status Description
nodeA	waiting	
nodeB	failed	Error: Takeover of node "nodeB" is not possible. Action: Use the "storage failover show" command to view the cause of the failure.

2 entries were displayed.

Status: Paused - An error occurred in "Data ONTAP updates" phase. The non-disruptive update cannot continue until the error has been resolved. Resolve all issues, then use the "cluster image resume-update" command to resume the update.

```
cluster1:>
```

The following example shows that the automated nondisruptive update is paused-on-error in "Post-update checks" update phase and "Status Description" shows the error and action.

```

cluster1::> cluster image show-update-progress
Estimated      Elapsed
Update Phase   Status          Duration        Duration
-----
Data ONTAP updates  completed      02:19:00      00:00:03
Post-update checks  paused-on-error 00:10:00      00:00:02

Details:

Post-update Check  Status          Error-Action
-----
Cluster Quorum    Error           Error: Cluster is not in quorum.
Status

Action: Use the (privilege: advanced)
"cluster ring show" command to verify
all replication unit details.

5 entries were displayed.

Status: Paused - An error occurred in "Post-update checks" phase. The
non-disruptive update cannot continue until the error has been resolved.
Resolve all issues, then use the "cluster image resume-update" command
to resume the update.

cluster1::>

```

The following example shows that the automated nondisruptive update is completed on nodeA and nodeB.

```

cluster1::> cluster image show-update-progress
Estimated      Elapsed
Update Phase   Status          Duration        Duration
-----
Pre-update checks  completed      00:10:00      00:00:13
Data ONTAP updates  completed      01:23:00      01:15:11
Post-update checks  completed      00:10:00      00:00:02
3 entries were displayed.

Updated nodes: nodeA, nodeB.
cluster1:>

```

The following example shows the automated update of two-node MetroCluster configuration having clusters cluster\_A and cluster\_B. In this case, cluster\_A's update is waiting and cluster\_B's update is in progress. cluster\_B's switchback operation is in progress.

```

cluster_A::> cluster image show-update-progress
Estimated      Elapsed
Cluster                Duration          Duration      Status
-----
cluster_A                00:00:00          00:00:00      waiting
cluster_B                00:00:00          00:06:42      in-
progress

Details: Switchback in progress.

Waiting for partner cluster "sti60-vsimg-ucs134f_siteB" to be up.

cluster_A::>

```

The following example shows that the automated update is completed on both cluster\_A and cluster\_B in two-node MetroCluster configuration.

```

cluster_A::> cluster image show-update-progress
Estimated      Elapsed
Cluster                Duration          Duration      Status
-----
cluster_A                00:00:00          00:20:44      completed
cluster_B                00:00:00          00:10:43      completed

Details: MetroCluster updated successfully.

cluster_A::>

```

## cluster image show

Display currently running image information

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

### Description

The `cluster image show` command displays information about the version of Data ONTAP that is running on each node and the date/time when it was installed. By default, the command displays the following information:

- Node name
- Current version

- Installation date and time

## 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**

Displays information about the specified node.

**[-version <text>] - Current Version**

Displays information about the nodes running the specified version.

**[-date <MM/DD/YYYY HH:MM:SS>] - Date Installed**

Displays information about the nodes with the specified installation date.

## Examples

The following example displays information about currently running images on all nodes of the cluster:

```
cluster1::> cluster image show
                Current           Installation
Node           Version           Date
-----
node1          8.3                -
node2          8.3                -
2 entries were displayed.
```

## cluster image update

### Manage an update

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

### Description

The `cluster image update` command is used to initiate a Data ONTAP update. The update is preceded by a validation of the cluster to ensure that any issues that might affect the update are identified. There are two types of updates of a cluster. A rolling update updates Data ONTAP one HA pair at a time. This type of update is performed for clusters with fewer than eight nodes or when the `-force-rolling` option is specified for clusters with eight or more nodes. A batch update is used for clusters of eight or more nodes, and performs updates of multiple HA pairs at the same time.

There are predefined points in the update when the update can be paused (either by the user or by an error). These pause points occur after validation, after download to the boot device, after takeover has completed,



and after giveback has completed.

## Parameters

### **-version <text> - Update Version**

Specifies the Data ONTAP version to use to update the cluster.

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

Specifies the nodes that are to be updated. This parameter is not supported for updates of MetroCluster configurations and for two-stage upgrades.

### **[-estimate-only <>true>] - Estimate Only**

Creates a report of the steps that occur during the update without actually doing them.

### **[-pause-after {none|all}] - Update Pause**

Specifies that the update should pause at each predefined pause points (for example, after validation, after download to the boot device, after takeover, and after giveback) during the update.

### **[-ignore-validation-warning {true|false}] - Ignore Validation**

Specifies that the update should proceed even if the validation reports warnings.

### **[-skip-confirmation {true|false}] - Skip Confirmation**

Specifies that a validation that does not detect any error issues should not ask the user to confirm the update but simply proceed with the update.

### **[-force-rolling <>true>] - Force Rolling Update**

This option is used for clusters with eight or more nodes to specify that a rolling update (one HA pair at a time) should be done. This parameter is not supported for two-node MetroCluster.

### **[-stabilize-minutes <integer>] - Minutes to stabilize**

Specifies the number of minutes that the update should wait after a takeover or giveback is completed. This allows time for the clients to recover from the pause in I/O that occurs during takeover and giveback. This parameter is not supported for two-pack MetroCluster.

## Examples

The following example shows the update operation:

```
cluster1::> cluster image update -version 8.3
```

It can take several minutes to complete validation...

```
Pre-update Check      Status      Error-Action
```

```
-----  
-----
```

```
CIFS status           OK
```

```
Cluster health status OK
```

```
Cluster quorum status OK
```

```
Disk status           OK
```

```
High Availability
```

```
status                OK
```

```
LIF status            OK
```

```
LIFs on home node    OK
```

```
status
```

```
MetroCluster          OK
```

```
configuration status
```

```
SnapMirror status     OK
```

```
Overall Status        OK
```

```
10 entries were displayed.
```

```
Do you want to continue? {y|n}: y
```

```
Starting update...
```

## cluster image validate

Validates the cluster's update eligibility

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

### Description

The `cluster image validate` command checks for issues within the cluster that might lead to problems during the update.

### 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.

**[-version <text>] - Update Version**

Specifies the Data ONTAP version to use to validate the cluster.

**[-rolling <>true>] - Rolling Update**

Specify this optional parameter on a cluster with eight or more nodes to perform a rolling-update check. The default is to perform a batch-update check.



This parameter is only supported on a cluster with eight or more nodes, and is not supported for two-node MetroCluster.

**[`-nodes {<nodename>|local}`] - Nodes**

Specifies the nodes that are to be validated. This parameter is not supported for MetroCluster configurations and for two-stage upgrades.

**Examples**

The following example shows the validate operation:

```
cluster1::> cluster image validate -version 8.3

It can take several minutes to complete validation...
Pre-update Check      Status      Error-Action
-----
-----
CIFS status           OK
Cluster health status OK
Cluster quorum status OK
Disk status           OK
High Availability
status
LIF status            OK
LIFs on home node     OK
MetroCluster
configuration status
SnapMirror status     OK
Overall Status        OK
10 entries were displayed.
```

**cluster image package delete**

Remove a package from the cluster image package repository

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

**Description**

The `cluster image package delete` command deletes the specified version of the package from the package repository. The associated information about the package is also deleted from the update database.

**Parameters**

**`-version <text>` - Version To Be Deleted**

Specifies the package version that is to be deleted.

## Examples

The following example deletes the package with version 8.3:

```
cluster1::> cluster image package delete -version 8.3  
  
Package Delete Operation Completed Successfully
```

## cluster image package get

Fetch a package file from a URL into the cluster image package repository

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

### Description

The `cluster image package get` command fetches a Data ONTAP package file specified by the URL into the cluster. The package is stored in the cluster package repository and the information from the package is stored in the update database.

### Parameters

#### **-url <text> - Package URL**

Specifies the URL from which to get the package.

## Examples

The following example displays how to get a package from a URL:

```
cluster1::> cluster image package get -url http://example.com/image.tgz
```

## cluster image package show-repository

Display information about packages available in the cluster image package repository

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

### Description

The `cluster image package show-repository` command displays the package versions that are in the cluster package repository. By default, the command displays the following information:

- Package version

### Parameters

**{ [-fields <fieldname>,...]**

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

| [-detail ]

This parameter specifies that detailed information should be displayed.

| [-instance ] }

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

[-download-ver <text>] - Downloaded Version

Displays packages with the specified download version.

[-component-name <text>,...] - Component Name

Displays packages for the specified component.

[-component-version <text>,...] - Component Version

Displays packages with the specified component version.

[-package-build-time <MM/DD/YYYY HH:MM:SS>] - Package Build Time

Displays packages with the specified build time.

## Examples

The following example displays the packages in the cluster package repository:

```
cluster1::> cluster image package show-repository
Package Version Package Build Time
-----
8.3          9/12/2014 10:27:33
```

## cluster kernel-service commands

### cluster kernel-service show

Display cluster service state in the kernel

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

### Description

The `cluster kernel-service show` command displays the following information from the master node for each node in the cluster:

- Node name
- The quorum status of that node
- The availability status of that node
- The operational status of that node

## Parameters

**{ [-fields <fieldname>,...]**

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

**| [-instance ] }**

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

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

The node in the cluster where the information be being reported from. If this parameter is not specified, the command displays information about all nodes in the cluster.

**[-cluster-node <text>] - Cluster Node (privilege: advanced)**

The node in the cluster that the information listed is regarding. If this parameter is specified, the command displays information only about the nodes with the specified state value.

**[-status-quorum {out of quorum|in quorum}] - Quorum Status (privilege: advanced)**

The quorum status of the node specified by `-cluster-node` . If this parameter is specified, the command displays information only about the nodes with the specified state value.

**[-status-avail {false|true|unknown}] - Availability Status (privilege: advanced)**

The availability status of the node specified by `-cluster-node` . If this parameter is specified, the command displays information only about the nodes with the specified state value.

**[-status-oper {unknown|operational|not-operational}] - Operational Status (privilege: advanced)**

The operational status of the node specified by `-cluster-node` . If this parameter is specified, the command displays information only about the nodes with the specified state value.

## Examples

The following example displays information about all nodes in the cluster:

```

cluster1::*> cluster kernel-service show
Master          Cluster          Quorum          Availability
Operational
Node           Node           Status          Status          Status
-----
-----
cluster1-01    cluster1-01    in-quorum      true
operational
                cluster1-02    in-quorum      true
operational
2 entries were displayed.

cluster1::*> cluster kernel-service show -instance
Master Node: cluster1-01
      Cluster Node: cluster1-01
      Quorum Status: in-quorum
Availability Status: true
Operational Status: operational
Master Node: cluster1-01
      Cluster Node: cluster1-02
      Quorum Status: in-quorum
Availability Status: true
Operational Status: operational
2 entries were displayed.

```

## cluster log-forwarding commands

### cluster log-forwarding create

Create a log forwarding destination

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

#### Description

The `cluster log-forwarding create` command creates log forwarding destinations for remote logging.

#### Parameters

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

Host name or IPv4 or IPv6 address of the server to forward the logs to.

##### **[-port <integer>] - Destination Port**

The port that the destination server listen on.

### **[`-protocol {udp-unencrypted|tcp-unencrypted|tcp-encrypted}`] - Log Forwarding Protocol**

The protocols are used for sending messages to the destination. The protocols can be one of the following values:

- `_ udp-unencrypted _` - User Datagram Protocol with no security
- `_ tcp-unencrypted _` - Transmission Control Protocol with no security
- `_ tcp-encrypted _` - Transmission Control Protocol with Transport Layer Security (TLS)

### **[`-verify-server {true|false}`] - Verify Destination Server Identity**

When this parameter is set to `true`, the identity of the log forwarding destination is verified by validating its certificate. The value can be set to `true` only when the `tcp-encrypted` value is selected in the protocol field. When this value is `true` the remote server might be validated by OCSP. The OCSP validation for cluster logs is controlled with the [security config oosp enable -app audit\\_log](#) and [security config oosp disable -app audit\\_log](#).

### **[`-facility <Syslog Facility>`] - Syslog Facility**

The syslog facility to use for the forwarded logs.

### **[`-force <true>`] - Skip the Connectivity Test**

Normally, the `cluster log-forwarding create` command checks that the destination is reachable via an ICMP ping, and fails if it is not reachable. Setting this value to `true` bypasses the ping check so that the destination can be configured when it is unreachable.

## Examples

This example causes audit logs to be forwarded to a server at address 192.168.0.1, port 514 with USER facility.

```
cluster1::> cluster log-forwarding create -destination 192.168.0.1 -port 514 -facility user
```

## Related Links

- [security config oosp enable](#)
- [security config oosp disable](#)

## cluster log-forwarding delete

Delete a log forwarding destination

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

## Description

The `cluster log-forwarding delete` command deletes log forwarding destinations for remote logging.



## Parameters

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

Host name or IPv4 or IPv6 address of the server to delete the forwarding entry for.

### **-port <integer> - Destination Port**

The port that the destination server listen on.

## Examples

This example deletes the forwarding of all logs to the server at address 1.1.1.1, port 514.

```
cluster1::> cluster log-forwarding delete -destination 1.1.1.1 -port 514
```

## cluster log-forwarding modify

Modify log forwarding destination settings

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

## Description

The `cluster log-forwarding modify` command modifies log forwarding destinations for remote logging.

## Parameters

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

The host name or IPv4 or IPv6 address of the server to be modified.

### **-port <integer> - Destination Port**

The port that the destinations servers listen on.

### **[-verify-server {true|false}] - Verify Destination Server Identity**

When this parameter is set to `true`, the identity of the log forwarding destination is verified by validating its certificate. The value can be set to `true` only when the `tcp-encrypted` value is selected in the `protocol` field. When this value is `true` the remote server might be validated by OCSP. The OCSP validation for cluster logs is controlled with the [security config ocsf enable -app audit\\_log](#) and [security config ocsf disable -app audit\\_log](#).

### **[-facility <Syslog Facility>] - Syslog Facility**

The syslog facility to use for the forwarded logs.

## Examples

This example modifies the facility of audit logs that are forwarded to the destination server at address 192.168.0.1, port 514.

```
cluster1::> cluster log-forwarding modify -destination 192.168.0.1 -port
514 -facility local1
```

## Related Links

- [security config oosp enable](#)
- [security config oosp disable](#)

## cluster log-forwarding show

Display log forwarding destinations

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

## Description

The `cluster log-forwarding show` command displays log forwarding information:

- Destination (IPv4/IPv6/hostname)
- Port number
- List of log classes
- Facility

## 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 <Remote InetAddress>] - Destination Host**

If this optional parameter is specified, the command displays information about the forwarding destinations with the specified host name, IPv4 or IPv6 address.

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

If this optional parameter is specified, the command displays information about the forwarding destinations with the specified ports.

**[-protocol {udp-unencrypted|tcp-unencrypted|tcp-encrypted}] - Log Forwarding Protocol**

If this optional parameter is specified, the command displays information about the forwarding destinations with the specified protocols.

**[-verify-server {true|false}] - Verify Destination Server Identity**

If this optional parameter is specified, the command displays information about the forwarding destinations with the specified `verify-server` values.

### **[*-facility* <Syslog Facility>] - Syslog Facility**

If this optional parameter is specified, the command displays information about the forwarding destinations with the specified facility.

### **Examples**

```
cluster-1::> cluster log-forwarding show
Verify Syslog
Destination Host          Port    Protocol          Server  Facility
-----
192.168.0.1              514    udp-unencrypted  false  user
```

## **cluster peer commands**

### **cluster peer create**

Create a new cluster peer relationship

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

### **Description**

The `cluster peer create` command establishes a peer relationship between two clusters. Cluster peering enables independent clusters to coordinate and exchange data.

Before creating a new cluster peer relationship, make sure that both clusters are individually healthy and that there are no other peer relationships between the two clusters that might interfere with the new relationship.

You can create a cluster peer relationship using the IPv4 or IPv6 protocol. You may not use both protocols within a single relationship.

Use the [cluster show](#) and [cluster peer show](#) commands on each cluster to display health, peering eligibility, and peering information about the two clusters.

### **Parameters**

#### **[*-peer-addr*s <Remote InetAddress>,...] - Remote Intercluster Addresses**

Use this parameter to specify the names or IP addresses of the logical interfaces used for intercluster communication. Separate the addresses with commas.

The addresses you provide here are associated with the remote cluster until you modify or delete the relationship, regardless of whether the addresses are valid. Make sure to provide addresses which you know will remain available on the remote cluster. You can use the hostnames of the remote cluster's intercluster addresses, the IP addresses of the remote cluster's intercluster LIFs or both.

#### **[*-username* <text>] - Remote User Name**

Use this optional parameter to specify a username that runs a reciprocal `cluster peer create` command on the peered cluster. If you choose not to use the reciprocal creation option, by not supplying a username for reciprocal creation, you must run `cluster peer create` again on the remote cluster to complete the peering relationship.

If you specify the username for the remote cluster, you will be prompted to enter the associated remote password. These credentials are not stored, they are used only during creation to authenticate with the remote cluster and to enable the remote cluster to authorize the peering request. The provided username's profile must have access to the console application in the remote cluster.

Use the [security login role show](#) and [security login show](#) commands on each cluster to find user names and their privilege levels.

**[-no-authentication <true>] - Do Not Use Authentication**

Use this optional parameter when omitting the `-username` parameter to indicate that you will create an unauthenticated peering relationship.

**[-timeout <integer>] - Operation Timeout (seconds) (privilege: advanced)**

Use this optional parameter to specify a timeout value for peer communications. Specify the value in seconds. The default timeout value is 60 seconds.

**[-address-family {ipv4|ipv6}] - Address Family of Relationship**

Use this optional parameter to specify the address family of the cluster peer relationship. The default is based on existing relationships, existing local intercluster LIFs belonging to a particular address-family, and the addresses supplied to the `cluster peer create` command.

**[-offer-expiration {MM/DD/YYYY HH:MM:SS | {1..7}days | {1..168}hours}] - Passphrase Match Deadline**

Specifying `cluster peer create` normally creates an offer to establish authentication with a cluster that is a potential cluster peer to this cluster. Such offers expire unless they are accepted within some definite time. Use this optional parameter to specify the date and time at which this offer should expire, the time after which the offer will no longer be accepted.

**[-rpc-connect-timeout <integer>] - Timeout for RPC Connect (seconds) (privilege: advanced)**

Use this optional parameter to specify a timeout value for the RPC connect during peer communications. Specify the value in seconds. The default timeout value is 10 seconds.

**[-update-ping-timeout <integer>] - Timeout for Update Pings (seconds) (privilege: advanced)**

Use this optional parameter to specify a timeout value for pings while updating remote cluster information. Specify the value in seconds. The default timeout value is 5 seconds. This parameter applies only to cluster peer relationships using the IPv4 protocol.

**[-ipspace <IPspace>] - IPspace for the Relationship**

Use this optional parameter to specify the IPspace within which the cluster peering relationship is to operate. The default is to use the 'Default' IPspace.

**[-local-name <Cluster name>] - Peer Cluster Local Name**

Use this optional parameter to specify a unique local name to identify the remote cluster that is being peered. The local name must conform to the same rules as a cluster name. The default value is the remote cluster name.

**[-generate-passphrase <true>] - Use System-Generated passphrase**

Use this optional parameter alone to create cluster peer offer for the unidentified clusters or use it along with `-peer-addr` option to automatically generate the passphrase for the cluster peer operation with the peer cluster.

### **[`-initial-allowed-vserver-peers <Vserver Name>,...`] - Vservers allowed for auto peering**

Use this optional parameter to specify the list of Vservers for which reciprocal Vserver peering with peer cluster should be enabled. Upon the time of successful peering, Vserver peer permission entries will be created for the peer cluster for the specified list of Vservers.

### **Examples**

This example creates a peer relationship between cluster1 and cluster2. This reciprocal create executes the create command on both the local cluster and the remote cluster. The cluster peer create command can use the hostnames of cluster2's intercluster addresses, the IP addresses of cluster2's intercluster LIFs, or both. Note that the admin user's password was typed at the prompt, but was not displayed.

```
cluster1::> cluster peer create -peer-addr cluster2-d2,10.98.234.246
-username admin

Remote Password:

cluster1::> cluster peer show -instance
Peer Cluster Name: cluster2
    Remote Intercluster Addresses: cluster2-d2, 10.98.234.246
    Availability of the Remote Cluster: Available
        Remote Cluster Name: cluster2
        Active IP Addresses: 10.98.234.246, 10.98.234.243
        Cluster Serial Number: 1-80-123456
    Address Family of Relationship: ipv4
    Authentication Status Administrative: no-authentication
    Authentication Status Operational: absent
        Last Update Time: 02/05 21:05:41
    IPspace for the Relationship: Default
    Guidance for When Encryption Should Be Used: never
```

This example shows coordinated peer creation. The `cluster peer create` command was issued locally on each cluster. This does not require you to provide the username and password for the remote cluster. There is a password prompt, but if you are logged in as the admin user, you may simply press enter.

```
cluster1::> cluster peer create -peer-addr cluster2-d2, 10.98.234.246 -no
-authentication
```

Remote Password:

NOTICE: Addition of the local cluster information to the remote cluster  
has

failed with the following error: not authorized for that command. You may  
need to repeat this command on the remote cluster.

```
cluster1::> cluster peer show
```

Peer Cluster Name	Cluster Serial Number	Availability	Authentication
cluster2	1-80-123456	Available	absent

```
cluster2::> cluster peer create -peer-addr cluster1-d2 -no-authentication
```

Remote Password:

NOTICE: Addition of the local cluster information to the remote cluster  
has

failed with the following error: not authorized for that command. You may  
need to repeat this command on the remote cluster.

```
cluster2::> cluster peer show
```

Peer Cluster Name	Cluster Serial Number	Availability	Authentication
cluster1	1-80-654321	Available	absent

This example shows a reciprocal cluster peer create over IPv6 addresses, that establishes a cluster peer relationship with an IPv6 address family.

```

cluster1::> cluster peer create -peer-addr
FD20:8B1E:B255:C222:6A17:0BBD:E92C:4523 -username admin

Remote Password:

cluster1::> cluster peer show -instance
Address Family of Relationship: ipv6
                Peer Cluster Name: cluster2
                Remote Intercluster Addresses:
FD20:8B1E:B255:C222:6A17:0BBD:E92C:4523
                Availability of the Remote Cluster: Available
                Remote Cluster Name: cluster2
                Active IP Addresses:
FD20:8B1E:B255:C222:6A17:0BBD:E92C:4523
                Cluster Serial Number: 1-80-123456
                Address Family of Relationship: ipv6
                Authentication Status Administrative: no-authentication
                Authentication Status Operational: absent
                Last Update Time: 02/05 21:05:41
                IPspace for the Relationship: Default
                Guidance for When Encryption Should Be Used: never

```

This example shows creation of an authenticated peering relationship. It is an example of using the coordinated method to create a cluster peer relationship. The `cluster peer create` command is issued locally on each cluster. Before executing this pair of commands, a passphrase to be used with the commands is chosen and given at the prompts. The passphrase can be any text; it is prompted for twice on each cluster, and all four copies of the passphrase must agree. The passphrase does not echo on the screen. The passphrase must be longer than the minimum length as specified by the `cluster peer policy` on both clusters.

```
cluster1::> cluster peer create -peer-addr cluster2-d2, 10.98.234.246
```

```
Enter the passphrase:
```

```
Enter the passphrase again:
```

```
Notice: Now use the same passphrase in the "cluster peer create" command  
in the  
    other cluster.
```

```
cluster1::> cluster peer show
```

Peer	Cluster Name	Cluster Serial Number	Availability	Authentication
cluster2	-		Unavailable	pending

```
cluster2::> cluster peer create -peer-addr cluster1-d2
```

```
Enter the passphrase:
```

```
Enter the passphrase again:
```

```
cluster2::> cluster peer show
```

Peer	Cluster Name	Cluster Serial Number	Availability	Authentication
cluster1		1-80-654321	Available	ok

This example creates a peer relationship between cluster1 and cluster2. This reciprocal create executes the create command on both the local cluster and the remote cluster. The cluster peer create command can use the hostnames of cluster2's intercluster addresses, the IP addresses of cluster2's intercluster LIFs or both. Note that the admin user's password was typed at the prompt, but was not displayed. The `-local-name` parameter is specified to create a local name used to identify the peer cluster in cases where the name of the peer cluster is not unique or not descriptive.



```

cluster1::> create -peer-addr 10.98.191.193 -username admin -local-name
locallyUniqueName

cluster1::> cluster peer show
Peer Cluster Name          Cluster Serial Number Availability
Authentication
-----
-----
locallyUniqueName          1-80-000011          Available
absent

cluster1::> cluster peer show -instance
Peer Cluster Name: locallyUniqueName
    Remote Intercluster Addresses: 10.98.191.193
    Availability of the Remote Cluster: Available
    Remote Cluster Name: cluster2
    Active IP Addresses: 10.98.191.193
    Cluster Serial Number: 1-80-000011
    Address Family of Relationship: ipv4
    Authentication Status Administrative: no-authentication
    Authentication Status Operational: absent
    Last Update Time: 02/05 21:05:41
    IPspace for the Relationship: Default
    Guidance for when Encryption Should Be Used: never

```

The following example create a peer relationship between cluster1 and cluster2 using system-generated passphrases:

```

cluster1::> cluster peer create -peer-addr 10.98.191.193 -generate
-passphrase
Passphrase: UCa+6lRVICXeL/gq1WrK7ShR
Peer Cluster Name: cluster2
Initial Allowed Vserver Peers: -
Expiration Time: 6/7/2017 09:16:10 +5:30
Intercluster LIF IP: 10.140.106.185
Warning: make a note of the passphrase - it cannot be displayed again.

```

```

cluster1::> cluster peer show
Peer Cluster Name      Cluster Serial Number Availability
Authentication
-----
cluster2              -                Unavailable    pending

```

```

cluster2::> cluster peer create -peer-addr 10.140.106.185

```

Enter the passphrase:

Clusters cluster1 and cluster2 are peered.

```

cluster2::> cluster peer show
Peer Cluster Name      Cluster Serial Number Availability
Authentication
-----
cluster1              1-80-654321    Available     ok

```

The following example creates a cluster peer offer from cluster1 for an anonymous cluster using system-generated passphrase with offer expiration period of two days and the cluster2 uses the offer from cluster2 with the system-generated passphrase:

```

cluster1::> cluster peer create -generate-passphrase -offer-expiration
2days
Passphrase: UCa+6lRVICXeL/gq1WrK7ShR
Peer Cluster Name: Clus_7ShR (temporary generated)
Initial Allowed Vserver Peers: -
Expiration Time: 6/9/2017 08:16:10 +5:30
Intercluster LIF IP: 10.140.106.185
Warning: make a note of the passphrase - it cannot be displayed again.

```

```

cluster1::> cluster peer show
Peer Cluster Name      Cluster Serial Number Availability
Authentication
-----
Clus_7ShR              -                Unidentified    pending
cluster2::> cluster peer create -peer-addr 10.140.106.185

```

Enter the passphrase:

Clusters cluster1 and cluster2 are peered.

```

cluster2::> cluster peer show
Peer Cluster Name      Cluster Serial Number Availability
Authentication
-----
cluster1              1-80-654321      Available       ok

```

Cluster "cluster1" creates an offer with `initial-allowed-vserver-peers` option set to Vservers "vs1" and "vs2". And the peer cluster "cluster2" uses the offer and creates peer relationship with cluster1, upon the successful peer relationship establishment, Vserver peer permission entries are created for the Vservers "vs1" and "vs2" in cluster "cluster1" for the peer cluster "cluster2". The following example describes the usage of `initial-allowed-vserver-peers` option in the cluster peer creation workflow:

```

cluster1::> cluster peer create -generate-passphrase -initial-allowed
-vserver-peers vs1,vs2
Passphrase: UCa+6lRVICXeL/gq1WrK7ShR
      Peer Cluster Name: Clus_7ShR (temporary generated)
      Initial Allowed Vserver Peers: vs1,vs2
      Expiration Time: 6/7/2017 09:16:10 +5:30
      Intercluster LIF IP: 10.140.106.185
Warning: make a note of the passphrase - it cannot be displayed again.

```

```

cluster1::> cluster peer show
Peer Cluster Name      Cluster Serial Number Availability
Authentication
-----
Clus_7ShR              -                Unidentified    pending
cluster2::> cluster peer create -peer-addr 10.140.106.185

```

Enter the passphrase:

Clusters cluster1 and cluster2 are peered.

```

cluster2::> cluster peer show
Peer Cluster Name      Cluster Serial Number Availability
Authentication
-----
cluster1              1-80-654321        Available       ok
cluster1::> vserver peer permission show

```

```

Peer Cluster      Vserver      Applications
-----
cluster2
                vs1          snapmirror
                vs2          snapmirror
2 entries were displayed.

```

## Related Links

- [cluster show](#)
- [cluster peer show](#)
- [security login role show](#)
- [security login show](#)

## cluster peer delete

Delete a cluster peer relationship

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

### Description

The `cluster peer delete` command removes a peering relationship. It removes the relationship records, state data, and all associated jobs.

Before removing the relationship, the command verifies that no resources depend on the relationship. For example, if any SnapMirror relationships exist, the command denies the request to delete the peering relationship. You must remove all dependencies for the deletion to succeed. The `cluster peer delete` command removes only the local instance of the peer relationship. An administrator in the peer cluster must use the `cluster peer delete` command there as well to completely remove the relationship.

### Parameters

#### **-cluster <text> - Peer Cluster Name**

Use this parameter to specify the peering relationship to delete by specifying the name of the peered cluster.

### Examples

This example shows a failed deletion due to a SnapMirror dependency.

```
cluster2::> cluster peer delete -cluster cluster1
Error: command failed: Unable to delete peer relationship. Reason: A
SnapMirror source exists in this cluster
```

## cluster peer modify-local-name

Modify the local name for a cluster peer

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

### Description

The `cluster peer modify-local-name` command modifies the local name for a remote cluster. The new local name must be unique among all the local names for the remote clusters with which this cluster is peered.

### Parameters

#### **-name <text> - Cluster Peer Name**

Use this parameter to specify the existing local name for a peer cluster.

#### **-new-name <Cluster name> - Cluster Peer Local Name**

Use this parameter to specify the new local name of the peer cluster. The new local name must conform to the same rules as a cluster name.

## Examples

```
cluster2::> cluster peer modify-local-name -name cluster1 -new-name
cluster1A
```

## cluster peer modify

### Modify cluster peer relationships

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

### Description

The `cluster peer modify` command modifies the attributes of a peering relationship. When you modify a peer relationship and specify `-peer-addr`s, all of the remote addresses must respond, must be intercluster addresses, and must belong to the remote cluster that is being modified; or the modification request is denied.

### Parameters

#### **-cluster <text> - Peer Cluster Name**

Use this parameter to specify the peering relationship to modify by specifying the name of the peered cluster.

#### **[-peer-addr <Remote InetAddress>,...] - Remote Intercluster Addresses**

Use this parameter to specify the names or IP addresses of the logical interfaces used for intercluster communication. Separate the addresses with commas. The list of addresses you provide replaces the existing list of addresses.

#### **[-address-family {ipv4|ipv6}] - Address Family of Relationship**

Use this parameter to specify the address family of the names specified with the `peer-addr`s parameter.

#### **[-timeout <integer>] - Operation Timeout (seconds) (privilege: advanced)**

Use this parameter to specify a timeout value for peer communications. Specify the value in seconds.

#### **[-auth-status-admin {no-authentication|revoked|use-authentication}] - Authentication Status Administrative**

Use this parameter to adjust the authentication in use for the peer relationship. The defined values for this field are as follows.

- `no-authentication` - The cluster peer relationship uses no authentication.
- `use-authentication` - The cluster peer relationship is to be authenticated. After you use this value, you will be prompted for a passphrase to be used in determining a new authentication key, just as in the authenticated `cluster peer create` command or you can use the option `generate-passphrase` to automatically generate the passphrase.
- `revoked` - The cluster peer relationship is no longer to be trusted. Peering communication with this cluster peer is suspended until the two clusters set their `auth-status-admin` attributes either both to `no-authentication` or both to `use-authentication`.

Changes should be reflected on both clusters involved in a peering relationship.

**[-rpc-connect-timeout <integer>] - Timeout for RPC Connect (seconds) (privilege: advanced)**

Use this optional parameter to specify a timeout value for the RPC connect during peer communications. Specify the value in seconds.

**[-update-ping-timeout <integer>] - Timeout for Update Pings (seconds) (privilege: advanced)**

Use this optional parameter to specify a timeout value for pings while updating remote cluster information. Specify the value in seconds. This parameter applies only to cluster peer relationships using the IPv4 protocol.

**[-ipspace <IPspace>] - IPspace for the Relationship**

Use this optional parameter to specify that cluster peering communication for this remote cluster is to be done using local intercluster LIFs that are on ports in the named IPspace.

**[-generate-passphrase <>true>] - Use System-Generated passphrase**

Use this optional parameter along with `-auth-status-admin` option's `use-authentication` to automatically generate the passphrase which can be used for cluster peer operation.

## Examples

This example modifies the peering relationship to use a new IP address in the remote cluster for intercluster communications and revoke authentication.

View existing cluster peer configuration using following command :

```
cluster1::> cluster peer show -instance
Peer Cluster Name: cluster2
    Remote Intercluster Addresses: cluster2-d2, 10.98.234.246
    Availability of the Remote Cluster: Available
    Remote Cluster Name: cluster2
    Active IP Addresses: 10.98.234.246, 10.98.234.243
    Cluster Serial Number: 1-80-123456
    Remote Cluster Nodes: cluster2-01, cluster2-02
    Remote Cluster Health: true
    Unreachable Local Nodes: -
    Address Family of Relationship: ipv4
    Authentication Status Administrative: use-authentication
    Authentication Status Operational: ok
    Last Update Time: 02/05 21:05:41
    IPspace for the Relationship: Default
    Guidance for when Encryption Should Be Used: as-possible
```

Modify the cluster peer configuration using following command :

```
cluster1::> cluster peer modify -cluster cluster2 -peer-addr cluster2-  
d2,10.98.234.264 -auth-status-admin revoked  
Warning: This will discard the authentication key.  
  
Warning: You are removing authentication from the peering relationship  
with  
cluster "cluster2". Use the "cluster peer modify" command on  
cluster "cluster2" with the "-auth-status-admin  
no-authentication" parameter to complete authentication removal from  
the peering relationship.  
  
Do you want to continue?{y|n}:y
```

The following example modifies the peering relationship to use authentication with `-generate-passphrase` option.



```
cluster1::> cluster peer modify -cluster cluster2
-auth-status-admin use-authentication -generate-passphrase
```

Notice: Use the below system-generated passphrase in the "cluster peer modify"

command in the other cluster.

Passphrase: UCa+6lRVICXeL/gq1WrK7ShR

Expiration Time: 6/7/2017 09:16:10 +5:30

Peer Cluster Name: cluster2

Warning: make a note of the passphrase - it cannot be displayed again.

Until then, the operational authentication state of the relationship remains as "pending".

```
cluster1::> cluster peer offer show
```

Allowed

Peer Cluster Name	Authentication	Creation	Expiration
-----			
cluster2	ok-and-offer	6/7/2017 08:16:10	6/7/2017 09:16:10
-			

Modify cluster peer relationship in cluster2 with use-authentication option and use the auto-generated passphrase.

```
cluster2::> cluster peer modify -cluster cluster2 -auth-status-admin use-
authentication
```

Notice: Use a auto-generated passphrase or choose a passphrase of 8 or more characters.

To ensure the authenticity of the peering relationship, use a phrase or sequence of characters that would be hard to guess.

Enter the passphrase:

Confirm the passphrase:

## Related Links

- [cluster peer create](#)

## cluster peer ping

Initiate intercluster connectivity test

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

### Description

The `cluster peer ping` command displays the status of the network mesh used by the peering relationship. The command checks the network connection to each remote IP address known by the cluster. This includes all intercluster addresses. It is possible for a known address to be not present during the ping. These addresses are not checked, but the absence is temporary.

The most useful parameters for diagnosing problems are `-count` and `-packet-size`. Use the `-count` and `-packet-size` parameters to diagnose problems similarly to how you use them with the standard ping utility.

To display network connection status within a cluster, use the [network ping](#) command.

### 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.

**[-originating-node {<nodename>|local}] - Node that Initiates Ping**

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

**[-destination-cluster <Cluster name>] - Cluster to Ping**

Use this parameter to specify the peer cluster you wish to ping.

**[-destination-node <Peer Node Name>] - Node to Ping in Destination Cluster**

Use this parameter to specify a specific node in the destination cluster to ping.

**[-ip-address <IP Address>] - Active IP Address**

Use this parameter to specify the active IP address you wish to ping.

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

Use this parameter to specify the number of requests to be sent to the destination.

**[-status**

**{unknown\_node|internal\_error|unreachable|session\_reachable|interface\_reachable}] - Status of Ping Operation**

Use this parameter to display only ping results that have the status you specify.

**[-timeout <integer>] - Ping Timeout in Seconds**

Use this parameter to specify a timeout value in seconds for the ping operation.

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

Use this parameter to specify the number of data bytes to be sent in the ping packet.

### **[-ttl <integer>] - Time to Live/ Number of Hops**

Use this parameter to specify the maximum number of network hops a packet may make before it is considered a failure.

### **[-response-time <double>] - Response Time (ms)**

Use this parameter to display only nodes that have the response time (in milliseconds) that you specify. This parameter is most useful when specified with a range of values, such as >500

## **Examples**

This example shows a ping of cluster1 and cluster2 from cluster2. All nodes are reachable.

```
cluster2::> cluster peer ping
Node: node1          Destination Cluster: cluster2
Destination Node IP Address      Count  TTL RTT(ms) Status
-----
node1          10.98.228.230      1    255   0.209 interface_reachable
node2          10.98.228.234      1    255   0.42  interface_reachable
Node: node2          Destination Cluster: cluster2
Destination Node IP Address      Count  TTL RTT(ms) Status
-----
node1          10.98.228.230      1    255   0.358 interface_reachable
node2          10.98.228.234      1    255   0.17  interface_reachable
Node: node1          Destination Cluster: cluster1
Destination Node IP Address      Count  TTL RTT(ms) Status
-----
node3          10.98.229.22       1    255   0.336 interface_reachable
node4          10.98.229.29       1    255   0.354 interface_reachable
Node: node2          Destination Cluster: cluster1
Destination Node IP Address      Count  TTL RTT(ms) Status
-----
node3          10.98.229.22       1    255   0.354 interface_reachable
node4          10.98.229.29       1    255   0.336 interface_reachable
6 entries were displayed.
```

## **Related Links**

- [network ping](#)

## cluster peer show

Display peer cluster information

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

### Description

The `cluster peer show` command displays information about the peering relationships between the current cluster and other clusters. Cluster peering enables independent clusters to coordinate and exchange data.

### 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.

**[-cluster <text>] - Peer Cluster Name**

Selects the peered clusters that match this parameter value.

**[-cluster-uuid <UUID>] - Cluster UUID (privilege: advanced)**

Selects the peered clusters that match this parameter value.

**[-peer-addr <Remote InetAddress>,...] - Remote Intercluster Addresses**

Selects the peered clusters that match this parameter value (remote-host name or IP address).

**[-availability <availability>] - Availability of the Remote Cluster**

Selects the peered clusters that match this parameter value. This parameter can have four different values:

- Available - The peer cluster availability status will be *Available* only if all the nodes in the local cluster are able to contact all the nodes in the remote cluster.
- Partial - The peer cluster availability status will be *Partial* only if some nodes in the local cluster are not able to contact some or all nodes in the peer cluster.
- Unavailable - The peer cluster availability status will be *Unavailable* only if all the nodes in the local cluster are not able to contact any node in the peer cluster.
- Pending - The peer cluster availability status will be *Pending* while the system is creating in-memory health data.
- Unidentified - The peer cluster availability status will be *Unidentified* if the cluster peer offer is created for an anonymous cluster and is unused. When the offer is used, then the availability will get changed to any of the above mentioned status.

Note: If one or more nodes in the local cluster are offline or unreachable, then those nodes are not used to determine the availability status for the remote nodes.

**[-rcluster <text>] - Remote Cluster Name**

Selects the peered clusters that match this parameter value.

**[-ip-addr <Remote InetAddress>,...] - Active IP Addresses**

Selects the peered clusters that match this parameter value.

**[-serialnumber <Cluster Serial Number>] - Cluster Serial Number**

Selects the peered clusters that match this parameter value.

**[-remote-cluster-nodes <text>,...] - Remote Cluster Nodes**

Selects the peered clusters that match this parameter value.

**[-remote-cluster-health {true|false}] - Remote Cluster Health**

Selects the peered clusters that match this parameter value.

- true - This means that there is cluster quorum in the peer cluster.
- false - This means that there is no cluster quorum in the peer cluster.

**[-unreachable-local-nodes <text>,...] - Unreachable Local Nodes**

Selects the peered clusters that match this parameter value.

**[-timeout <integer>] - Operation Timeout (seconds) (privilege: advanced)**

Selects the peered clusters that match this parameter value.

**[-address-family {ipv4|ipv6}] - Address Family of Relationship**

Selects the peered clusters that have a relationship established using this protocol.

**[-auth-status-admin {no-authentication|revoked|use-authentication}] - Authentication Status Administrative**

Selects the peered clusters that match this parameter value, which must be chosen from the following values.

- no-authentication - The cluster peer relationship uses no authentication.
- use-authentication - The cluster peer relationship is authenticated.
- revoked - The cluster peer relationship is revoked until agreement can be reached.

**[-auth-status-operational {ok|absent|pending|expired|revoked|declined|refused|ok-and-offer|absent-but-offer|revoked-but-offer|key-mismatch|intent-mismatch|incapable}] - Authentication Status Operational**

Selects the peered clusters that match this parameter value, which must be one of the following values.

- ok - The clusters both use authentication and they have agreed on an authentication key.
- absent - The clusters agree not to use authentication.
- pending - This cluster has made an outstanding offer to authenticate with the other cluster, but agreement has not yet been reached.
- expired - This cluster's offer to authenticate with the other cluster expired before agreement was reached.
- revoked - This cluster has revoked any prior authentication agreement.
- declined - The other cluster has revoked the authentication agreement and is declining to communicate with this cluster.

- refused - The other cluster actively refuses the communication attempts, perhaps because its part of the peering relationship has been deleted.
- ok-and-offer - The clusters agree on an authentication key and are using it. In addition, this cluster has made an outstanding offer to re-authenticate with the other cluster.
- absent-but-offer - The clusters currently agree that neither side requires authentication of the other, but this cluster has made an outstanding offer to authenticate.
- revoked-but-offer - This cluster has revoked any authentication agreement, but it has made an outstanding offer to authenticate.
- intent-mismatch - The two clusters disagree on whether authentication is required.
- key-mismatch - The two clusters both believe that they are authenticated, but one of the shared secrets has become corrupted.
- incapable - The other cluster is no longer running a version of Data ONTAP that supports authenticated cluster peering.

**[-rpc-connect-timeout <integer>] - Timeout for RPC Connect (privilege: advanced)**

Selects the peered clusters that match this parameter value.

**[-update-ping-timeout <integer>] - Timeout for Update Pings (privilege: advanced)**

Selects the peered clusters that match this parameter value.

**[-last-updated <MM/DD/YYYY HH:MM:SS>] - Last Update Time**

Selects the peered clusters that match this parameter value.

**[-ipspace <IPspace>] - IPspace for the Relationship**

Selects the peered clusters whose relationships are to cross the named local IPspace. The default value is the IPspace name "Default". In relationships created before ONTAP 8.3.1, the initial value is "-" and is not updated to "Default" until an action is taken on a cluster peer relationship, such as creating, modifying, or deleting a relationship.

## Examples

This example shows the output of the cluster peer show command when all nodes in the local cluster are able to contact all nodes in the remote peer cluster. Additionally, the peer relationship is authenticated and operating correctly.

```
cluster1::> cluster peer show
Peer Cluster Name      Cluster Serial Number Availability
Authentication
-----
cluster2               1-80-123456      Available      ok
```

Detailed information for this scenario is shown below.

```

cluster1::> cluster peer show -instance
Peer Cluster Name: cluster2
    Remote Intercluster Addresses: cluster2-d2, 10.98.234.246
Availability of the Remote Cluster: Available
    Remote Cluster Name: cluster2
    Active IP Addresses: 10.98.234.246, 10.98.234.243
    Cluster Serial Number: 1-80-123456
    Remote Cluster Nodes: cluster2-01, cluster2-02
    Remote Cluster Health: true
    Unreachable Local Nodes: -
    Address Family of Relationship: ipv4
Authentication Status Administrative: use-authentication
    Authentication Status Operational: ok
        Last Update Time: 02/05 21:05:41
    IPspace for the Relationship: Default

```

This example shows the output of the cluster peer show command when some nodes in the local cluster are not able to contact some or all of the nodes in the remote peer cluster.

```

cluster1::> cluster peer show
Peer Cluster Name      Cluster Serial Number Availability
Authentication
-----
-----
cluster2                1-80-123456          Available      ok

```

Detailed information for this scenario is shown below.

```

cluster1::> cluster peer show -instance
Peer Cluster Name: cluster2
    Remote Intercluster Addresses: cluster2-d2, 10.98.234.246
Availability of the Remote Cluster: Partial
    Remote Cluster Name: cluster2
    Active IP Addresses: 10.98.234.246, 10.98.234.243
    Cluster Serial Number: 1-80-123456
    Remote Cluster Nodes: cluster2-01, cluster2-02
    Remote Cluster Health: false
    Unreachable Local Nodes: -
    Address Family of Relationship: ipv4
Authentication Status Administrative: use-authentication
    Authentication Status Operational: ok
        Last Update Time: 02/05 21:05:41
    IPspace for the Relationship: Default

```

This example shows the output of the cluster peer show command when some nodes in the local cluster cannot be contacted from the node where the command is executed, but all the other nodes including node on which command is executed are able to contact all nodes in the remote peer cluster.

```
cluster1::> cluster peer show
Peer Cluster Name      Cluster Serial Number Availability
Authentication
-----
-----
cluster2                1-80-123456          Available      ok
```

Detailed information for this scenario is shown below.

```
cluster1::> cluster peer show -instance
Peer Cluster Name: cluster2
    Remote Intercluster Addresses: cluster2-d2, 10.98.234.246
    Availability of the Remote Cluster: Available
        Remote Cluster Name: cluster2
        Active IP Addresses: 10.98.234.246, 10.98.234.243
        Cluster Serial Number: 1-80-123456
        Remote Cluster Nodes: cluster2-01, cluster2-02
        Remote Cluster Health: true
        Unreachable Local Nodes: cluster1-01
        Address Family of Relationship: ipv4
    Authentication Status Administrative: use-authentication
        Authentication Status Operational: ok
            Last Update Time: 02/05 21:05:41
                IPspace for the Relationship: Default
```

This example shows the output of the cluster peer show command when some nodes in the local cluster cannot be contacted from the node where the command is executed, and the node on which command is executed is also not able to contact the remote peer cluster.

```
cluster1::> cluster peer show
Peer Cluster Name      Cluster Serial Number Availability
Authentication
-----
-----
cluster2                1-80-123456          Unavailable    ok
```

Detailed information for this scenario is shown below.



```

cluster1::> cluster peer show -instance
Peer Cluster Name: cluster2
    Remote Intercluster Addresses: cluster2-d2, 10.98.234.246
Availability of the Remote Cluster: Unavailable
    Remote Cluster Name: cluster2
    Active IP Addresses: 10.98.234.246, 10.98.234.243
    Cluster Serial Number: 1-80-123456
    Remote Cluster Nodes: cluster2-01, cluster2-02
    Remote Cluster Health: -
    Unreachable Local Nodes: cluster1-01
    Address Family of Relationship: ipv4
Authentication Status Administrative: use-authentication
    Authentication Status Operational: ok
        Last Update Time: 02/05 21:05:41
    IPspace for the Relationship: Default

```

This example shows the output of the cluster peer show command when all the nodes in the local cluster are not able to contact any nodes in the remote peer cluster.

```

cluster1::> cluster peer show
Peer Cluster Name      Cluster Serial Number Availability
Authentication
-----
-----
cluster2                1-80-123456          Unavailable    ok

```

Detailed information for this scenario is shown below.

```

cluster1::> cluster peer show -instance
Peer Cluster Name: cluster2
    Remote Intercluster Addresses: cluster2-d2, 10.98.234.246
Availability of the Remote Cluster: Unavailable
    Remote Cluster Name: cluster2
    Active IP Addresses: 10.98.234.246, 10.98.234.243
    Cluster Serial Number: 1-80-123456
    Remote Cluster Nodes: cluster2-01, cluster2-02
    Remote Cluster Health: -
    Unreachable Local Nodes: -
    Address Family of Relationship: ipv4
Authentication Status Administrative: use-authentication
    Authentication Status Operational: ok
        Last Update Time: 02/05 21:05:41
    IPspace for the Relationship: Default

```

This example shows the output of the `cluster peer show` command while the system is creating in-memory health data.

```
cluster1::> cluster peer show
Peer Cluster Name      Cluster Serial Number Availability
Authentication
-----
-----
cluster2                1-80-123456          Pending          ok
```

Detailed information for this scenario is shown below.

```
cluster1::> cluster peer show -instance
Peer Cluster Name: cluster2
  Remote Intercluster Addresses: cluster2-d2, 10.98.234.246
  Availability of the Remote Cluster: Pending
    Remote Cluster Name: cluster2
    Active IP Addresses: 10.98.234.246, 10.98.234.243
    Cluster Serial Number: 1-80-123456
    Remote Cluster Nodes: -
    Remote Cluster Health: -
    Unreachable Local Nodes: -
  Address Family of Relationship: ipv4
Authentication Status Administrative: use-authentication
  Authentication Status Operational: ok
    Last Update Time: 02/05 21:05:41
  IPspace for the Relationship: Default
```

This example shows the output of the `cluster peer show` command for the offer created for an anonymous cluster:

```

cluster1::> cluster peer show
Peer Cluster Name          Cluster Serial Number Availability
Authentication
-----
-----
Clus_4gHR                  -                Unidentified   pending

```

Detailed information for this scenario is shown below.

```

cluster1::> cluster peer show -instance
Peer Cluster Name: Clus_4gHR
    Remote Intercluster Addresses: -
    Availability of the Remote Cluster: Unidentified
        Remote Cluster Name: Clus_4gHR
        Active IP Addresses: 10.98.234.246, 10.98.234.243
        Cluster Serial Number: -
        Remote Cluster Nodes: -
        Remote Cluster Health: -
        Unreachable Local Nodes: -
    Address Family of Relationship: ipv4
    Authentication Status Administrative: use-authentication
    Authentication Status Operational: ok
        Last Update Time: 02/05 21:05:41
    IPspace for the Relationship: Default

```

## cluster peer connection show

Show current peering connections for a cluster

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

### Description

The `cluster peer connection show` command displays information about the current TCP connections and how they are supporting the set of peering relationships.

### 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.

**[`-cluster-name <text>`] - Remote Cluster**

Selects the connections associated with the named peered cluster.

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

Selects the connections hosted by the given node.

**[`-connection-type {mgmt-client|mgmt-server|data}`] - Cluster Peering Connection Type**

Selects the connections of the named type. This parameter can have one of three different values:

- `Mgmt-client` - Management-plane client connections, created so that this node may make management-plane requests of other nodes.
- `Mgmt-server` - Management-plane server connections, over which this node services requests made by other nodes' `mgmt-client` connections.
- `Data` - Connections made between data-planes of different nodes.

**[`-index <integer>`] - Index of Connection**

Selects the connections with the given index value.

**[`-cluster-uuid <UUID>`] - Cluster UUID (privilege: advanced)**

Selects the connections to the cluster with the given cluster UUID.

**[`-auth-status-admin {no-authentication|revoked|use-authentication}`] - Authentication Status Administrative**

Selects connections to the peered clusters whose intended authentication matches this parameter value.

**[`-auth-status-operational {ok|absent|pending|expired|revoked|declined|refused|ok-and-offer|absent-but-offer|revoked-but-offer|key-mismatch|intent-mismatch|incapable}`] - Authentication Status Operational**

Selects connections to the peered clusters whose authentication state matches this parameter value.

**[`-authenticated {true|false}`] - Authenticated**

Selects connections that have been authenticated, or not, according to this parameter value.

**[`-port <integer>`] - Network Port**

Selects the connections that match this parameter value.

**[`-idle [<integer>h][<integer>m][<integer>s]>`] - Idle Time**

Selects the connections whose idle times match this parameter value.

**[`-address <IP Address>`] - Remote Network Address**

Selects the connections that have this parameter value as the remote network address.

**Examples**

This example shows the output of the cluster peer connection show command.

```

cluster1::> cluster peer connection show
Cluster Node           Connection Type  Auth  Encrypt Idle  Remote
Address
-----
cluster2
  node1
    data  true  true  6s
10.10.10.100
    data  true  true  7s
10.10.10.100
    data  true  true  20s
10.10.10.100
    data  true  true  11s
10.10.10.100
    data  true  true  7s
10.10.10.100
    data  true  true  7s
10.10.10.100
    data  true  true  11s
10.10.10.200
    data  true  true  11s
10.10.10.200
    data  true  true  48s
10.10.10.200
    data  true  true  48s
10.10.10.200
    data  true  true  37s
10.10.10.200
    data  true  true  37s
10.10.10.200
12 entries were displayed.

```

## cluster peer health show

### Check peer cluster health

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

### Description

The `cluster peer health show` command displays information about the health of the nodes in peer clusters from the perspective of the nodes in the local cluster. The command obtains health information by performing connectivity and status probes of each peer cluster's nodes from each node in the local cluster.

To enable quick access to remote cluster health information, remote cluster health status is periodically checked and cached. These cached results enable users and system features to quickly assess the availability

of remote resources. By default, this command accesses cached results. Use the `-bypass-cache true` option to force a current, non-cached check of remote cluster health.

## 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.

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

Selects the node that matches this parameter value.

**[-destination-cluster <Cluster name>] - Peer Cluster**

Selects the cluster that matches this parameter value.

**[-destination-node <Peer Node Name>] - Peer Node**

Selects the node that matches this parameter value.

**[-destination-cluster-uuid <UUID>] - Peer UUID**

Selects the cluster that matches this parameter value.

**[-data-ping**

**{unknown\_node|internal\_error|unreachable|session\_reachable|interface\_reachable}] - Status of Data Ping Operation**

Selects the nodes that match this parameter value.

**[-icmp-ping**

**{unknown\_node|internal\_error|unreachable|session\_reachable|interface\_reachable}] - Status of ICMP Ping Operation**

Selects the nodes that match this parameter value.

**[-node-health {true|false}] - RDB Health of the Node**

Selects the nodes that match this parameter value (`true` means healthy).

**[-cluster-health {true|false}] - Cluster Health**

Selects the nodes that match this parameter value (`true` means healthy).

**[-availability {true|false}] - Communication Indicator**

Selects the nodes that match this parameter value (`true` means communicating).

**[-bypass-cache {true|false}] - Bypass Cache and Determine Health**

Bypasses cached results to determine current cluster health (`true` means bypass the cache). Cached results may not be current, but they are displayed more quickly.

**[-last-updated <MM/DD/YYYY HH:MM:SS>] - Last Update Time**

Selects the nodes that match this parameter value.

## Examples

The following example shows typical output for this command in a cluster of two nodes that has a peer cluster of two nodes.

```
cluster1::> cluster peer health show
Node          Cluster-Name      Node-Name
          Ping-Status          RDB-Health Cluster-Health
Availability
-----
node1
      cluster2          node3
      Data: interface_reachable
      ICMP: interface_reachable true      true      true
          node4
      Data: interface_reachable
      ICMP: interface_reachable true      true      true
node2
      cluster2          node3
      Data: interface_reachable
      ICMP: interface_reachable true      true      true
          node4
      Data: interface_reachable
      ICMP: interface_reachable true      true      true
4 entries were displayed.
```

The following example shows detailed health information for node3 in cluster2 from the perspective of node1 in cluster1.

```
cluster1::> cluster peer health show -originating-node node1 -destination
-cluster cluster2 -destination-node node3 -instance
Local Node: node1
          Peer Cluster: cluster2
          Peer Node: node3
          Peer UUID: 5e4befb2-1f36-11d0-98c9-123476563412
Status of Data Ping Operation: interface_reachable
Status of ICMP Ping Operation: interface_reachable
          RDB health of the node: true
          Cluster Health: true
          Communication Indicator: true
          Last Update Time: 02/06 18:58:38
```

## cluster peer offer cancel

Cancel the outstanding offer to authenticate with a peer cluster

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

### Description

The `cluster peer offer cancel` command cancels an outstanding offer to authenticate with a potentially peered cluster. After the command completes, the given cluster can no longer establish authentication using the given authentication offer.

### Parameters

**-cluster <text> - Peer Cluster Name**

Use this parameter to specify which offer should be cancelled, by specifying the name of the cluster to which the offer is extended.

### Examples

The following example cancels the authentication offer to cluster2.

```
cluster1::> cluster peer offer cancel -cluster cluster2
```

## cluster peer offer modify

Modify an outstanding offer to authenticate with a peer cluster

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

### Description

The `cluster peer offer modify` command modifies the outstanding offer to authenticate with a potentially peered cluster. Every authentication offer has an expiration time, after which the offer will no longer be honored. This command is used to change that expiration time. To cancel the offer outright, use the [cluster peer offer cancel](#) command instead.

### Parameters

**-cluster <text> - Peer Cluster Name**

Use this parameter to specify the offer to be modified by indicating the name of the cluster to which it has been extended.

**[-offer-expiration {MM/DD/YYYY HH:MM:SS | {1..7}days | {1..168}hours}] - Authentication Offer Expiration Time**

Use this parameter to specify the new expiration time for the offer.

**[-initial-allowed-vserver-peers <Vserver Name>, ...] - Vservers Initially Allowed for Peering**

Use this optional parameter to specify the list of Vservers for which reciprocal Vserver peering with peer cluster should be enabled.



## Examples

This example modifies the expiration time for the authentication offer to push it out by an hour.

```
cluster1::> cluster peer offer show
Peer Cluster Name      Authentication Creation      Expiration
-----
cluster2               absent_but_offer
                        7/23/2013 14:45:47  7/23/2013
15:45:47

cluster1::> cluster peer offer modify -cluster cluster2 -offer-expiration
"7/23/2013 16:45:47"

cluster1::> cluster peer offer show
Peer Cluster Name      Authentication Creation      Expiration
-----
cluster2               absent_but_offer
                        7/23/2013 14:45:47  7/23/2013
16:45:47
```

## Related Links

- [cluster peer offer cancel](#)

## cluster peer offer show

Display outstanding offers to authenticate with a peer cluster

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

## Description

The `cluster peer offer show` command displays information about authentication offers still pending with potential peer clusters. By default, the command displays information about all unexpired offers made by the local cluster.

To display detailed information about a specific offer, run the command with the `-cluster` parameter.

## 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.

**[*-cluster* <text>] - Peer Cluster Name**

Selects the offer that matches this parameter value.

**[*-auth-status-operational* {ok|absent|pending|expired|revoked|declined|refused|ok-and-offer|absent-but-offer|revoked-but-offer|key-mismatch|intent-mismatch|incapable}] - Authentication Status Operational**

Selects the offers that match this parameter value.

**[*-offer-creation* <MM/DD/YYYY HH:MM:SS>] - Authentication Offer Creation Time**

Selects the offers that match this parameter value.

**[*-offer-expiration* {MM/DD/YYYY HH:MM:SS | {1..7}days | {1..168}hours}] - Authentication Offer Expiration Time**

Selects the offers that match this parameter value.

**[*-initial-allowed-vserver-peers* <Vserver Name>,...] - Vservers Initially Allowed for Peering**

Selects the offers that match this parameter value.

**[*-offer-creator* <text>] - Authentication Offer Creator**

Selects the offers that match this parameter value.

## Examples

The following example displays information about the outstanding authentication offers:

```
cluster1::> cluster peer offer show

Allowed
Peer Cluster Name      Authentication Creation      Expiration
Vserver Peers
-----
cluster2               absent_but_offer 7/11/2013 22:22:52 7/11/2013
23:22:52 vs1,vs2
```

## cluster peer policy modify

Modify the policy configuration for the cluster peering service

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

### Description

The `cluster peer policy modify` command modifies the prevailing policy settings. One setting governs whether unauthenticated cluster peer relationships can exist. The other setting specifies a minimum length for

passphrases.

## Parameters

### **`[-is-unauthenticated-access-permitted {true|false}] - Is Unauthenticated Cluster Peer Access Permitted`**

Use this parameter to specify whether unauthenticated peering relationships are allowed to exist. Setting the parameter value to *true* allows such relationships to exist. Setting the value to *false* prevents both the creation of unauthenticated peering relationships as well as the modification of existing peering relationships to be unauthenticated. Setting the value to *false* is not possible if the cluster currently is in any unauthenticated relationships.

### **`[-passphrase-minlength <integer>] - Passphrase Length Minimum`**

Use this parameter to specify a minimum length for passphrases as given to the [cluster peer create](#) or [cluster peer modify](#) commands in the future. The default value for this parameter is 8.

## Examples

This example modifies the peering policy to disallow unauthenticated intercluster communications.

```
cluster1::> cluster peer policy show
Is Unauthenticated Cluster Peer Communication Permitted:  true
                Minimum Length for a Passphrase:  8

cluster1::> cluster peer policy modify -is-unauthenticated-access
-permitted false

cluster1::> cluster peer policy show
Is Unauthenticated Cluster Peer Communication Permitted:  false
                Minimum Length for a Passphrase:  8
```

## Related Links

- [cluster peer create](#)
- [cluster peer modify](#)

## cluster peer policy show

Display the policy configuration for the cluster peering service

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

## Description

The `cluster peer policy show` command displays the prevailing cluster peer authentication policy. There are two policies at present: one to control whether any cluster peer relationships can be unauthenticated, and one to control the minimum length for a passphrase. If the policy is set to preclude unauthenticated peering relationships, then unauthenticated relationships cannot be created inadvertently. Passphrases of less than the minimum length may not be used. By default, this minimum length is set to 8, so

passphrases must be 8 characters long or longer.

## Examples

This example shows the cluster peer policy when unauthenticated relationships may not be created inadvertently.

```
cluster1::> cluster peer policy show
Is Unauthenticated Cluster Peer Communication Permitted:  false
Minimum Length for a Passphrase:  9
```

## cluster quorum-service commands

### cluster quorum-service options modify

Modify the settings for cluster quorum-service

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

#### Description

The `cluster quorum-service options modify` command modifies the values of cluster quorum services options.

#### Parameters

**`[-ignore-quorum-warning-confirmations {true|false}]` - Whether or Not Warnings Are Enabled (privilege: advanced)**

Specifies whether cluster quorum warnings and confirmations should be ignored when cluster operations could negatively impact cluster quorum:

- Halting a node ([system node halt](#) )
- Rebooting a node ([system node reboot](#) )
- Issuing a planned takeover ([storage failover takeover](#) )

The default setting is false.

#### Examples

The following example shows the usage of this command:

```
cluster1::> set advanced
```

```
Warning: These advanced commands are potentially dangerous; use them only  
when
```

```
    directed to do so by NetApp personnel.
```

```
    Do you want to continue? {y|n}: y
```

```
cluster1::*> cluster quorum-service options modify -ignore-quorum-warning  
-confirmations true
```

## Related Links

- [system node halt](#)
- [system node reboot](#)
- [storage failover takeover](#)

## cluster quorum-service options show

Display the settings for cluster quorum-service

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

## Description

The `cluster quorum-service options show` command displays the values of cluster quorum services options.

## Examples

The following example demonstrates showing the state of `ignore-quorum-warning-confirmations` when it is false and true.

```
cluster1::*> set advanced
```

```
Warning: These advanced commands are potentially dangerous; use them only  
when
```

```
    directed to do so by NetApp personnel.
```

```
    Do you want to continue? {y|n}: y
```

```
cluster1::*> cluster quorum-service options show
```

```
Ignore Quorum Warning Confirmations
```

```
-----
```

```
false
```

# cluster ring commands

## cluster ring show

Display cluster node member's replication rings

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

### Description

The `cluster ring show` command displays a cluster's ring-replication status. Support personnel might ask you to run this command to assist with troubleshooting.

### Parameters

**{ [-fields <fieldname>,...]**

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

**| [-instance ] }**

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

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

Selects the rings that match this parameter value.

**[-unitname {mgmt|vldb|vifmgr|bcomd|crs|availd}] - Unit Name (privilege: advanced)**

Selects the rings that match this parameter value. Possible values are:

- `mgmt` - The management application
- `vldb` - The volume location database
- `vifmgr` - The virtual-interface manager
- `bcomd` - The SAN management daemon
- `crs` - The configuration replication service

**[-online {master|secondary|offline}] - Status (privilege: advanced)**

Selects the rings that match this parameter value.

**[-epoch <integer>] - Epoch (privilege: advanced)**

Selects the rings that match this parameter value.

**[-master <nodename>] - Master Node (privilege: advanced)**

Selects the rings that match this parameter value.

**[-local <nodename>] - Local Node (privilege: advanced)**

Selects the rings that match this parameter value.

**[-db-epoch <integer>] - DB Epoch (privilege: advanced)**

Selects the rings that match this parameter value.

**[-db-trnxs <integer>] - DB Transaction (privilege: advanced)**

Selects the rings that match this parameter value.

**[-num-online <integer>] - Number Online (privilege: advanced)**

Selects the rings that match this parameter value.

**[-rdb-uuid <UUID>] - RDB UUID (privilege: advanced)**

Selects the rings that match this parameter value.

**Examples**

The following example displays information about all replication rings in a two-node cluster:

```

cluster1::*> cluster ring show
Node          UnitName Epoch    DB Epoch DB Trnxs Master   Online
-----
node0         mgmt      1          1         1068    node0    master
node0         vldb     1          1          98     node0    master
node0         vifmgr   1          1         350    node0    master
node0         bcomd    1          1          56     node0    master
node0         crs      1          1          88     node0    master
node1         mgmt      1          1         1068    node0    secondary
node1         vldb     1          1          98     node0    secondary
node1         vifmgr   1          1         350    node0    secondary
node1         bcomd    1          1          56     node0    secondary
node1         crs      1          1          88     node0    secondary
10 entries were displayed.

```

## cluster statistics commands

### cluster statistics show

Display cluster-wide statistics

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

#### Description

The `cluster statistics show` command displays the following information. Each item lists the current value and; if applicable, the change (delta) from the previous reported value.

- CPU busy percentage
- Average of CPU busy percentage (advanced privilege level only)

- Total number of NFS and CIFS operations
- Number of NFS operations
- Number of CIFS operations
- Number of cache operations (advanced privilege level only)
- Total amount of network data received (advanced privilege level only)
- Total amount of network data sent (advanced privilege level only)
- Number of packets received (advanced privilege level only)
- Number of packets sent (advanced privilege level only)
- Busy percentage for the data network
- Amount of data received on the data network
- Amount of data sent on the data network
- Busy percentage for the cluster network
- Amount of data received on the cluster network
- Amount of data sent on the cluster network
- Amount of data read from disk
- Amount of data written to disk

At the diagnostic privilege level, the command displays the following information:

- Average of CPU busy percentage
- CPU busy percentage
- Total number of operations
- Number of NFS operations
- Number of CIFS operations
- Number of Fcache operations
- Number of SpinFS operations
- Total amount of network traffic received
- Total amount of network traffic sent
- Percentage of data-network utilization
- Amount of data-network traffic received
- Amount of data-network traffic sent
- Percentage of cluster-network utilization
- Amount of cluster-network traffic received
- Amount of cluster-network traffic sent
- Amount of data read from disk
- Amount of data written to disk
- Number of packets received
- Number of packets sent



## Examples

The following example displays cluster statistics:

```
cluster1::> cluster statistics show
  Counter      Value      Delta
-----
  CPU Busy:    84%      +27
  Operations:
    Total:     951471448  7210/s:11s
    NFS:       12957951479  13759/s:11s
    CIFS:       342195460  230/s:11s
  Data Network:
    Busy:       0%      -
    Received:   1.98TB  3.18MB/s:11s
    Sent:       6.20TB  903KB/s:11s
  Cluster Network:
    Busy:       0%      -
    Received:   6.33TB  1.34MB/s:11s
    Sent:       6.24TB  3.54MB/s:11s
  Storage Disk:
    Read:       207TB  82.7MB/s:11s
    Write:      53.3TB  53.5MB/s:11s
```

## cluster time-service commands

### cluster time-service ntp security modify

Modify NTP security settings

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

#### Description

The `cluster time-service ntp security modify` command allows setting of security parameters related to the Network Time Protocol (NTP) subsystem.

#### Parameters

**`[-is-query-enabled {true|false}] - Is Querying of NTP Server Enabled? (privilege: advanced)`**

Setting this parameter to *true* allows querying of the NTP subsystem from systems external to the cluster. For example, querying a node using the standard "ntpq" command can be enabled by this command. The default setting is *false* to protect against possible security vulnerabilities. If querying of the NTP subsystem is disabled, the [cluster time-service ntp status show](#) command can be used to obtain similar information. Although querying of the NTP subsystem from external hosts can be disabled with this command, executing a local query to the localhost address is always enabled.

## Examples

The following example enables the querying of the NTP subsystem from clients external to the cluster:

```
cluster-1::> cluster time-service ntp security modify -is-query-enabled true
```

## Related Links

- [cluster time-service ntp status show](#)

## cluster time-service ntp security show

Display NTP security settings

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

## Description

The `cluster time-service ntp security show` command displays the configuration of security features related to the Network Time Protocol (NTP) subsystem.

## Examples

The following example displays the NTP security configuration of the cluster:

```
cluster1::> cluster time-service ntp security show  
External queries enabled?: true
```

## cluster time-service ntp server create

Add a NTP Server

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

## Description

The `cluster time-service ntp server create` command associates the cluster with an external network time server for time correction and adjustment by using the Network Time Protocol (NTP).

The command resolves the time server host name to an IP address and performs several validation checks. If an error is detected during validation, it is reported.

The validation checks performed by this command include the following:

- The NTP replies to an NTP query with the specified protocol version.
- The NTP reply indicates that the external time server is synchronized to another time server.
- The distance and dispersion of the NTP reply from the "root" or source clock are within the required limits.

## Parameters

### **-server <text>** - NTP Server Host Name, IPv4 or IPv6 Address

This specifies the host name or IP address of the external NTP server that is to be associated with the cluster for time correction and adjustment.

### **[-version {3|4|auto}]** - NTP Version for Server (default: auto)

This optionally specifies the NTP protocol version that should be used for communicating with the external NTP server. If the external NTP server does not support the specified protocol version, time exchange cannot take place.

The supported values for this parameter include the following:

- 3 - Use NTP protocol version 3, which is based on Internet Standard Request For Comments (RFC) #1305.
- 4 - Use NTP protocol version 4, which is based on Internet Standard RFC #5905.
- auto - Have Data ONTAP select the NTP protocol version.

The default setting is `auto`.

### **[-is-preferred {true|false}]** - Is Preferred NTP Server (default: false) (privilege: advanced)

This optionally specifies whether the external NTP server is the primary time source for correcting and adjusting the cluster time. The responses from this source will be used unless its time is outside the accepted selection range.

The default setting is `false`.

You use this parameter when a high quality radio (or GPS) based time server is being used with a set of non-radio based backup time servers.

## Examples

The following example associates the cluster with an NTP server named `ntp1.example.com`.

```
cluster1::> cluster time-service ntp server create -server  
ntp1.example.com
```

## cluster time-service ntp server delete

### Delete a NTP Server

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

### Description

The `cluster time-service ntp server delete` command removes the association between the cluster and an external network time server that uses the Network Time Protocol (NTP).

## Parameters

### **-server <text>** - NTP Server Host Name, IPv4 or IPv6 Address

This specifies the host name or IP address of an existing external NTP server that the cluster will disassociate from.

## Examples

The following example disassociates an NTP server named `ntp2.example.com` from the cluster:

```
cluster1::> cluster time-service ntp server delete -server
ntp2.example.com
```

## cluster time-service ntp server modify

### Modify NTP Server Options

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

### Description

The `cluster time-service ntp server modify` command modifies the configuration of an existing external network time server that uses the Network Time Protocol (NTP) for time correction and adjustment.

## Parameters

### **-server <text>** - NTP Server Host Name, IPv4 or IPv6 Address

This specifies the host name or IP address of an existing external NTP server that is to be modified.

### **[-version {3|4|auto}]** - NTP Version for Server (default: auto)

This optionally specifies the NTP protocol version that should be used for communicating with the external NTP server. If the external NTP server does not support the specified protocol version, time exchange cannot take place.

The supported values for this parameter include the following:

- 3 - Use NTP protocol version 3, which is based on Internet Standard Request For Comments (RFC) #1305.
- 4 - Use NTP protocol version 4, which is based on Internet Standard RFC #5905.
- auto - Have Data ONTAP select the NTP protocol version.

The default setting is `auto`.

### **[-is-preferred {true|false}]** - Is Preferred NTP Server (default: false) (privilege: advanced)

This optionally specifies whether the external NTP server is the primary time source for correcting and adjusting the cluster time. The responses from this source will be used unless its time is outside the accepted selection range

The default setting is `false`.

You use this parameter when a high quality radio (or GPS) based time server is being used with a set of non-radio based backup time servers.

This parameter is available only at the advanced privilege level and higher.

## Examples

The following example modifies the NTP version of an NTP server named ntp1.example.com. The NTP version is changed to 4.

```
cluster1::> cluster time-service ntp server modify -server
ntp1.example.com -version 4
```

## cluster time-service ntp server reset

Reset NTP server list to a default selection

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

### Description

The `cluster time-service ntp server reset` command replaces the current configuration with one of the selected configurations.

If none or more than one time service configuration is selected, the command will fail.

### Parameters

**[-use-public {true|false}] - Reset Server List to Public Identified Defaults (default: false)**  
(privilege: advanced)

When set to true, this specifies that the public NTP server list used by Data ONTAP should replace the current configuration.

The default setting is `false`.

## Examples

The following example replaces the current time service configuration with the default public NTP server list that is used by Data ONTAP.

```
cluster1::> cluster time-service ntp server reset -use-public true
```

## cluster time-service ntp server show

Display a list of NTP Servers

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

## Description

The `cluster time-service ntp server show` command displays the association between the cluster and external network time servers that use the Network Time Protocol (NTP).

## Parameters

**{ [-fields <fieldname>,...]**

If you specify the `-fields <fieldname>,...` parameter, the command only displays the fields that you specify. For example: `-fields server,version`.

**| [-instance ] }**

If the `-instance` parameter is specified, the command displays all the available field information.

**[-server <text>] - NTP Server Host Name, IPv4 or IPv6 Address**

If this parameter is specified, the command displays the external NTP servers that match the specified server name or IP address.

**[-version {3|4|auto}] - NTP Version for Server (default: auto)**

If this parameter is specified, the command displays the external NTP servers that use the specified NTP version.

**[-is-preferred {true|false}] - Is Preferred NTP Server (default: false) (privilege: advanced)**

If this parameter is specified, the command displays the external NTP server or servers that match the specified preferred server status.

**[-is-public {true|false}] - Is Public NTP Server Default (privilege: advanced)**

If this parameter is specified, the command displays the information for the external NTP servers that are either on the NTP server list defined by Data ONTAP (`true` `')` or not on the list (`false` `')`.

## Examples

The following example displays information about all external NTP time servers that are associated with the cluster:

```
cluster1::> cluster time-service ntp server show
Server                Version
-----
ntp1.example.com      auto
ntp2.example.com      auto
```

## **cluster time-service ntp status show**

Display status of the node's NTP client

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

## Description

The `cluster time-service ntp status show` command displays the status of the associations between the cluster and external network time servers that use the Network Time Protocol (NTP).

## 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 this parameter is specified, the command displays detailed information about all entries.

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

If this parameter is specified, the command displays information related to associations on the specified node.

**[-server <text>] - NTP Server Host Name, IPv4 or IPv6 Address (privilege: advanced)**

If this parameter is specified, the command displays information about the associations related to the specified NTP server. The server should be specified as it is configured in the [cluster time-service ntp server show](#) command.

**[-server-address <IP Address>] - Server IP Address (privilege: advanced)**

If this parameter is specified, the command displays information about the associations related to the NTP server with the specified IP address.

**[-is-peer-reachable {true|false}] - Is Peer Reachable and Responding to Polls? (privilege: advanced)**

If this parameter is specified as `true`, the command displays information about associations with the NTP servers that have been successfully polled.

**[-is-peer-selected {true|false}] - Is Peer Selected as Clock Source? (privilege: advanced)**

If this parameter is specified as `true`, the command displays information about associations with the NTP servers that have been selected as the current clock source.

**[-selection-state <State of NTP Peer Selection>] - State of Server Selection (privilege: advanced)**

If this parameter is specified, the command displays information about associations with the specified selection state.

**[-selection-state-text <text>] - Description of Server Selection State (privilege: advanced)**

If this parameter is specified, the command displays information about associations with the specified selection state description.

**[-poll-interval <integer>] - Poll Interval (secs) (privilege: advanced)**

If this parameter is specified, the command displays information about associations that have the specified polling interval.

**[`-time-last-poll <integer>`] - Time from Last Poll (secs) (privilege: advanced)**

If this parameter is specified, the command displays information about associations that are polled at the specified time.

**[`-offset <double>`] - Offset from Server Time (ms) (privilege: advanced)**

If this parameter is specified, the command displays information about associations with the specified offset from the NTP server.

**[`-delay <double>`] - Delay Time to Server (ms) (privilege: advanced)**

If this parameter is specified, the command displays information about associations with the specified travelling time to the NTP server.

**[`-jitter <double>`] - Maximum Offset Error (ms) (privilege: advanced)**

If this parameter is specified, the command displays information about associations with the specified offset error from the NTP server.

**[`-reachability <Hex String>`] - Reachability of Server (privilege: advanced)**

If this parameter is specified, the command displays information about associations with the specified reachability to the NTP server. Reachability is specified as a hexbyte that has a bit to represent the success of each of the last eight polls of the specified server. A set bit represents a successful poll. The least significant bit represents the most recent poll, the next most significant bit the poll before that, and so on.

**[`-server-stratum <integer>`] - Stratum of Server Clock (privilege: advanced)**

If this parameter is specified, the command displays information about associations with NTP servers with the specified clock stratum.

**[`-server-reference <text>`] - Reference Clock at Server (privilege: advanced)**

If this parameter is specified, the command displays information about associations with NTP servers using the specified clock as reference.

**[`-reported-errors <NTP Peer and Packet Errors>,...`] - Reported Packet and Peer Errors (privilege: advanced)**

If this parameter is specified, the command displays information about associations with the specified errors.

## Examples

The following example displays the status of the NTP associations of the cluster:



```

cluster-1::*>cluster time-service ntp status show
Node: node-1
Server                Reachable  Selection State      Offset
(ms)
-----
ntp1.eng.netapp.com   true      Currently Selected Server
39.122
ntp2.eng.netapp.com   true      Candidate Server
37.786
2 entries were displayed.

```

The following example displays the status of the association with the specified external NTP server:

```

cluster-1::*>cluster time-service ntp status show -instance -server
ntp1.example.com
Node: node-1
NTP Server Host Name, IPv4 or IPv6 Address: ntp1.example.com
Server IP Address: 10.56.32.33
Is Peer Reachable and Responding to Polls?: true
Is Peer Selected as Clock Source?: true
State of Server Selection: sys_peer
Description of Server Selection State: Currently Selected Server
Poll Interval (secs): 64
Time from Last Poll (secs): 1
Offset from Server Time (ms): 26.736
Delay Time to Server (ms): 61.772
Maximum Offset Error (ms): 3.064
Reachability of Server: 01
Stratum of Server Clock: 2
Reference Clock at Server: 10.56.68.21
Reported Packet and Peer Errors: -

```

## Related Links

- [cluster time-service ntp server show](#)

## Copyright information

Copyright © 2026 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

## Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.