NetApp

cluster peer commands

ONTAP commands

NetApp
February 04, 2026

Table of Contents

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-addrs <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 | PnDTnHnMnS | PnW}]` **- 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-addrs` 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.

`[-encryption-protocol-proposed {none|tls-psk}]` **- Encryption Protocol To Be Used In Inter-Cluster Communication**

Use this optional parameter to specify how this cluster should use encryption in data connections to the other cluster. Specify 'tls-psk' to specify that TLS should be used with a Pre-Shared Key. Specify 'none' to use no encryption. Where authentication is used, the default is 'tls-psk'. Where authentication is not used, the default is 'none'.

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-addrs 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
 Encryption for Inter-Cluster Communication: none
```

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-addrs 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-addrs 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-addrs
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
 Encryption for Inter-Cluster Communication: none
```

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-addrs 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-addrs 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-addrs 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
 Encryption for Inter-Cluster Communication: none
```

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

```
cluster1::> cluster peer create -peer-addrs 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-addrs 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-addrs 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-addrs 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-addrs` , 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-addrs <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-addrs 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.

`[-encryption-protocol-proposed {none|tls-psk}]` - Encryption For Inter-Cluster Communication

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

- tls-psk - Use TLS with a Pre-Shared Key.

- none - Use no encryption.

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
 Encryption for Inter-Cluster Communication: tls-psk
```

Modify the cluster peer configuration using following command :

```
cluster1::> cluster peer modify -cluster cluster2 -peer-addrs 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
Vserver Peers
-------------------- ------------- -------------------
------------------- --------------
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-addrs <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-addrs <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.

`[-encryption-protocol-proposed {none|tls-psk}]` - Proposed Setting for Encryption of Inter-Cluster Communication

Selects the peered clusters that match the value of this parameter, that is, whether they are proposing to encrypt their cross-cluster communication or not.

`[-encryption-protocol {none|tls-psk}]` - Encryption Protocol For Inter-Cluster Communication

Selects the peered clusters that match the value of this parameter, that is, whether they are encrypting their cross-cluster communication or not.

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
 Encryption for Inter-Cluster Communication: none
```

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
 Encryption for Inter-Cluster Communication: none
```

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
 Encryption for Inter-Cluster Communication: none
```

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
 Encryption for Inter-Cluster Communication: none
```

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: 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
 Encryption for Inter-Cluster Communication: none
```

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
 Encryption for Inter-Cluster Communication: none
```

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
 Encryption for Inter-Cluster Communication: none
```

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
 Encryption for Inter-Cluster Communication: none
```

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 whose port matches 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                     Authentication: ok
        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 | PnDTnHnMnS | PnW}]` - 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 | PnDTnHnMnS | PnW}] - 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.

[-encryption-protocol-proposed {none|tls-psk}] - Encryption Protocol to Be Used For the Relationship

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.

`[-is-unencrypted-access-permitted {true|false}]` - Is Unencrypted Cluster Peer Access Permitted

Use this parameter to specify whether peering relationships that do not use encryption are allowed to exist. Setting the parameter value to *true* allows such relationships to exist. Setting the value to *false* prevents the creation of unauthenticated peering relationships and the modification of existing peering relationships to be unauthenticated, as well as preventing unencrypted peering relationships from being created and the modification of existing peering relationships to be unencrypted. Setting the value to *false* is not possible if the cluster currently is in any unauthenticated or unencrypted relationships.

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
    Is Unencrypted Cluster Peer Communication Permitted: true
```