Configure peer relationships
ONTAP 9
NetApp
November 28, 2022

This PDF was generated from https://docs.netapp.com/us-en/ontap/peering/create-cluster-relationship-93-later-task.html on November 28, 2022. Always check docs.netapp.com for the latest.
# Table of Contents

- Configure peer relationships ...................................................... 1
  - Create a cluster peer relationship ........................................... 1
  - Create an intercluster SVM peer relationship ............................ 4
  - Add an intercluster SVM peer relationship ............................... 5
Configure peer relationships

Create a cluster peer relationship

You can use the `cluster peer create` command to create a peer relationship between a local and remote cluster. After the peer relationship has been created, you can run `cluster peer create` on the remote cluster to authenticate it to the local cluster.

Before you begin

- You must have created intercluster LIFs on every node in the clusters that are being peered.
- The clusters must be running ONTAP 9.3 or later. (If the clusters are running ONTAP 9.2 or earlier, refer to the procedures in this archived document.)

Steps

1. On the destination cluster, create a peer relationship with the source cluster:

   ```
   cluster peer create -generate-passphrase -offer-expiration MM/DD/YY HH:MM:SS
   MM/DD/YYYY HH:MM:SS|1...7days|1...168hours -peer-addrs peer_LIF_IPs -initial-allowed-vserver
   -peers svm_name,..|* -ipspace ipspace
   ```

   If you specify both `-generate-passphrase` and `-peer-addrs`, only the cluster whose intercluster LIFs are specified in `-peer-addrs` can use the generated password.

   You can ignore the `-ipspace` option if you are not using a custom IPspace. For complete command syntax, see the man page.

   If you are creating the peering relationship in ONTAP 9.6 or later and you do not want cross-cluster peering communications to be encrypted, you must use the `-encryption-protocol-proposed none` option to disable encryption.

   The following example creates a cluster peer relationship with an unspecified remote cluster, and pre-authorizes peer relationships with SVMs `vs1` and `vs2` on the local cluster:

   ```
   cluster02::> cluster peer create -generate-passphrase -offer-expiration 2days -initial-allowed-vserver-peers vs1,vs2
   Passphrase: UCa+6lRVICXeL/gq1WrK7ShR
   Expiration Time: 6/7/2017 08:16:10 EST
   Initial Allowed Vserver Peers: vs1,vs2
   Intercluster LIF IP: 192.140.112.101
   Peer Cluster Name: Clus_7ShR (temporary generated)
   Warning: make a note of the passphrase - it cannot be displayed again.
   ```

   The following example creates a cluster peer relationship with the remote cluster at intercluster LIF IP addresses 192.140.112.103 and 192.140.112.104, and pre-authorizes a peer relationship with any SVM on the local cluster:
cluster02::> cluster peer create -generate-passphrase -peer-addrs 192.140.112.103,192.140.112.104 -offer-expiration 2days -initial-allowed-vserver-peers *

Passphrase: UCa+6lRVICxL/gq1WrK7ShR
Expiration Time: 6/7/2017 08:16:10 EST
Initial Allowed Vserver Peers: vs1,vs2
    Intercluster LIF IP: 192.140.112.101,192.140.112.102
    Peer Cluster Name: Clus_7ShR (temporary generated)
Warning: make a note of the passphrase - it cannot be displayed again.

The following example creates a cluster peer relationship with an unspecified remote cluster, and pre-authorizes peer relationships with SVMs vs1 and vs2 on the local cluster:

cluster02::> cluster peer create -generate-passphrase -offer-expiration 2days -initial-allowed-vserver-peers vs1,vs2

Passphrase: UCa+6lRVICxL/gq1WrK7ShR
Expiration Time: 6/7/2017 08:16:10 EST
Initial Allowed Vserver Peers: vs1,vs2
    Intercluster LIF IP: 192.140.112.101
    Peer Cluster Name: Clus_7ShR (temporary generated)
Warning: make a note of the passphrase - it cannot be displayed again.

2. On source cluster, authenticate the source cluster to the destination cluster:

    cluster peer create -peer-addrs peer_LIF_IPs -ipspace ipspace

For complete command syntax, see the man page.

The following example authenticates the local cluster to the remote cluster at intercluster LIF IP addresses 192.140.112.101 and 192.140.112.102:
cluster01::> cluster peer create -peer-addrs
192.140.112.101,192.140.112.102

Notice: Use a 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:

Clusters cluster02 and cluster01 are peered.

Enter the passphrase for the peer relationship when prompted.

3. Verify that the cluster peer relationship was created:

cluster peer show -instance

cluster01::> cluster peer show -instance

Peer Cluster Name: cluster02
Remote Intercluster Addresses: 192.140.112.101, 192.140.112.102
Availability of the Remote Cluster: Available
Remote Cluster Name: cluster2
Active IP Addresses: 192.140.112.101, 192.140.112.102
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

4. Check the connectivity and status of the nodes in the peer relationship:

cluster peer health show
cluster01::> cluster peer health show

<table>
<thead>
<tr>
<th>Node</th>
<th>cluster-Name</th>
<th>Node-Name</th>
<th>Ping-Status</th>
<th>RDB-Health</th>
<th>Cluster-Health</th>
<th>Avail...</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster01-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cluster02</td>
<td>cluster02</td>
<td>cluster02-01</td>
<td>interface_reachable</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>Data: interface_reachable</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ICMP: interface_reachable</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cluster02-02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data: interface_reachable</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ICMP: interface_reachable</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cluster01-02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cluster02</td>
<td>cluster02</td>
<td>cluster02-01</td>
<td>interface_reachable</td>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
<tr>
<td>Data: interface_reachable</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ICMP: interface_reachable</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cluster02-02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data: interface_reachable</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ICMP: interface_reachable</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Other ways to do this in ONTAP

<table>
<thead>
<tr>
<th>To perform these tasks with...</th>
<th>See this content...</th>
</tr>
</thead>
<tbody>
<tr>
<td>The redesigned System Manager (available with ONTAP 9.7 and later)</td>
<td>Prepare for mirroring and vaulting</td>
</tr>
<tr>
<td>System Manager Classic (available with ONTAP 9.7 and earlier)</td>
<td>Volume disaster recovery preparation overview</td>
</tr>
</tbody>
</table>

Create an intercluster SVM peer relationship

You can use the `vserver peer create` command to create a peer relationship between SVMs on local and remote clusters.

Before you begin

- The source and destination clusters must be peered.
- The clusters must be running ONTAP 9.3. (If the clusters are running ONTAP 9.2 or earlier, refer to the procedures in this archived document.)
- You must have "pre-authorized" peer relationships for the SVMs on the remote cluster.

For more information, see Creating a cluster peer relationship.

About this task

Previous releases of ONTAP let you authorize a peer relationship for only one SVM at a time. You needed to run the `vserver peer accept` command each time you authorized a pending SVM peer relationship.
Beginning with ONTAP 9.3, you can "pre-authorize" peer relationships for multiple SVMs by listing the SVMs in the `-initial-allowed-vserver` option when you create a cluster peer relationship. For more information, see Creating a cluster peer relationship.

**Steps**

1. On the data protection destination cluster, display the SVMs that are pre-authorized for peering:

   ```bash
   vserver peer permission show
   ```

<table>
<thead>
<tr>
<th>Peer Cluster</th>
<th>Vserver</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster02</td>
<td>vs1,vs2</td>
<td>snapmirror</td>
</tr>
</tbody>
</table>

2. On the data protection source cluster, create a peer relationship to a pre-authorized SVM on the data protection destination cluster:

   ```bash
   vserver peer create -vserver local_SVM -peer-vserver remote_SVM
   ```

   For complete command syntax, see the man page.

   The following example creates a peer relationship between the local SVM `pvs1` and the pre-authorized remote SVM `vs1`:

   ```bash
   cluster01::> vserver peer create -vserver pvs1 -peer-vserver vs1
   ```

3. Verify the SVM peer relationship:

   ```bash
   vserver peer show
   ```

<table>
<thead>
<tr>
<th>Remote Vserver</th>
<th>Peer Vserver</th>
<th>State</th>
<th>Peer Cluster</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>pvs1</td>
<td>vs1</td>
<td>peered</td>
<td>cluster02</td>
<td>snapmirror</td>
</tr>
</tbody>
</table>

**Add an intercluster SVM peer relationship**

If you create an SVM after configuring a cluster peer relationship, you will need to add a peer relationship for the SVM manually. You can use the `vserver peer create` command to create a peer relationship between SVMs. After the peer relationship has
been created, you can run `vserver peer accept` on the remote cluster to authorize the peer relationship.

**Before you begin**
The source and destination clusters must be peered.

**About this task**
You can create a peer relationships between SVMs in the same cluster for local data backup. For more information, see the `vserver peer create` man page.

Administrators occasionally use the `vserver peer reject` command to reject a proposed SVM peer relationship. If the relationship between SVMs is in the rejected state, you must delete the relationship before you can create a new one. For more information, see the `vserver peer delete` man page.

**Steps**
1. On the data protection source cluster, create a peer relationship with an SVM on the data protection destination cluster:
   ```bash
   vserver peer create -vserver local_SVM -peer-vserver remote_SVM -applications snapmirror|file-copy|lun-copy -peer-cluster remote_cluster
   ```
   The following example creates a peer relationship between the local SVM `pvs1` and the remote SVM `vs1`
   ```bash
   cluster01::> vserver peer create -vserver pvs1 -peer-vserver vs1 -applications snapmirror -peer-cluster cluster02
   ```
   If the local and remote SVMs have the same names, you must use a local name to create the SVM peer relationship:
   ```bash
   cluster01::> vserver peer create -vserver vs1 -peer-vserver vs1 -applications snapmirror -peer-cluster cluster01 -local-name cluster1vs1LocallyUniqueName
   ```
2. On the data protection source cluster, verify that the peer relationship has been initiated:
   ```bash
   vserver peer show-all
   ```
   For complete command syntax, see the man page.
   The following example shows that the peer relationship between SVM `pvs1` and SVM `vs1` has been initiated:
3. On the data protection destination cluster, display the pending SVM peer relationship:

   ```shell
   vserver peer show
   ```

   For complete command syntax, see the man page.

   The following example lists the pending peer relationships for `cluster02`:

   ```shell
   cluster02::> vserver peer show
   ```

<table>
<thead>
<tr>
<th>Vserver</th>
<th>Peer Vserver</th>
<th>State</th>
<th>Peer Cluster</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>vs1</td>
<td>pvs1</td>
<td>pending</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4. On the data protection destination cluster, authorize the pending peer relationship:

   ```shell
   vserver peer accept -vserver local_SVM -peer-vserver remote_SVM
   ```

   For complete command syntax, see the man page.

   The following example authorizes the peer relationship between the local SVM `vs1` and the remote SVM `pvs1`:

   ```shell
   cluster02::> vserver peer accept -vserver vs1 -peer-vserver pvs1
   ```

5. Verify the SVM peer relationship:

   ```shell
   vserver peer show
   ```
<table>
<thead>
<tr>
<th>Remote</th>
<th>Peer</th>
<th>Peer</th>
<th>Peering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vserver</td>
<td>Vserver</td>
<td>State</td>
<td>Peer Cluster</td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
<td>-------</td>
<td>--------------</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
<td>-------</td>
<td>--------------</td>
</tr>
<tr>
<td>pvs1</td>
<td>vs1</td>
<td>peered</td>
<td>cluster02</td>
</tr>
<tr>
<td>vs1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>