Cluster and SVM peering with the CLI

ONTAP 9

NetApp
November 28, 2022
# Table of Contents

Cluster and SVM peering with the CLI ................................................................. 1  
  Cluster and SVM peering overview with the CLI ................................................. 1  
  Prepare for cluster and SVM peering ............................................................. 1  
  Configure intercluster LIFs ............................................................................. 4  
  Configure peer relationships .......................................................................... 17  
  Enable cluster peering encryption on an existing peer relationship .................... 24  
  Remove cluster peering encryption from an existing peer relationship ............... 25  
  Where to find additional information ............................................................ 25
Cluster and SVM peering with the CLI

Cluster and SVM peering overview with the CLI

You can create peer relationships between source and destination clusters and between source and destination storage virtual machines (SVMs). You must create peer relationships between these entities before you can replicate Snapshot copies using SnapMirror.

ONTAP 9.3 offers enhancements that simplify the way you configure peer relationships between clusters and SVMs. The cluster and SVMs peering procedures are available for all ONTAP 9 versions. You should use the appropriate procedure for your version of ONTAP.

You perform the procedures using the command-line interface (CLI), not System Manager or an automated scripting tool.

Prepare for cluster and SVM peering

Peering basics

You must create peer relationships between source and destination clusters and between source and destination SVMs before you can replicate Snapshot copies using SnapMirror. A peer relationship defines network connections that enable clusters and SVMs to exchange data securely.

Clusters and SVMs in peer relationships communicate over the intercluster network using intercluster logical interfaces (LIFs). An intercluster LIF is a LIF that supports the "intercluster-core" network interface service and is typically created using the "default-intercluster" network interface service policy. You must create intercluster LIFs on every node in the clusters being peered.

Intercluster LIFs use routes that belong to the system SVM to which they are assigned. ONTAP automatically creates a system SVM for cluster-level communications within an IPspace.

Fan-out and cascade topologies are both supported. In a cascade topology, you need only create intercluster networks between the primary and secondary clusters and between the secondary and tertiary clusters. You need not create an intercluster network between the primary and the tertiary cluster.

It is possible (but not advisable) for an administrator to remove the intercluster-core service from the default-intercluster service policy. If this occurs, LIFs created using "default-intercluster" will not actually be intercluster LIFs. To confirm that the default-intercluster service policy contains the intercluster-core service, use the following command:

```
network interface service-policy show -policy default-intercluster
```
Connectivity requirements

Every intercluster LIF on the local cluster must be able to communicate with every intercluster LIF on the remote cluster.

Although it is not required, it is typically simpler to configure the IP addresses used for intercluster LIFs in the same subnet. The IP addresses can reside in the same subnet as data LIFs, or in a different subnet. The subnet used in each cluster must meet the following requirements:

- The subnet must belong to the broadcast domain that contains the ports that are used for intercluster communication.
- The subnet must have enough IP addresses available to allocate to one intercluster LIF per node.

For example, in a four-node cluster, the subnet used for intercluster communication must have four available IP addresses. Each node must have an intercluster LIF with an IP address on the intercluster network.

Intercluster LIFs can have an IPv4 address or an IPv6 address.

ONTAP 9 enables you to migrate your peering networks from IPv4 to IPv6 by optionally allowing both protocols to be present simultaneously on the intercluster LIFs. In earlier releases, all intercluster relationships for an entire cluster were either IPv4 or IPv6. This meant that changing protocols was a potentially disruptive event.

Port requirements

You can use dedicated ports for intercluster communication, or share ports used by the data network. Ports must meet the following requirements:

- All ports that are used to communicate with a given remote cluster must be in the same IPspace.

  You can use multiple IPspaces to peer with multiple clusters. Pair-wise full-mesh connectivity is required only within an IPspace.

- The broadcast domain that is used for intercluster communication must include at least two ports per node so that intercluster communication can fail over from one port to another port.

  Ports added to a broadcast domain can be physical network ports, VLANs, or interface groups (ifgrps).

  - All ports must be cabled.
  - All ports must be in a healthy state.
  - The MTU settings of the ports must be consistent.

Firewall requirements

Firewalls and the intercluster firewall policy must allow the following protocols:

- ICMP service

- TCP to the IP addresses of all the intercluster LIFs over the ports 10000, 11104, and 11105

- Bidirectional HTTPS between the intercluster LIFs

  Although HTTPS is not required when you set up cluster peering using the CLI, HTTPS is required later if
you use System Manager to configure data protection.

The default intercluster firewall policy allows access through the HTTPS protocol and from all IP addresses (0.0.0.0/0). You can modify or replace the policy if necessary.

Cluster requirement

Clusters must meet the following requirement:

- A cluster cannot be in a peer relationship with more than 255 clusters.

Use shared or dedicated ports

You can use dedicated ports for intercluster communication, or share ports used by the data network. In deciding whether to share ports, you need to consider network bandwidth, the replication interval, and port availability.

You can share ports on one peered cluster while using dedicated ports on the other.

Network bandwidth

If you have a high-speed network, such as 10 GbE, you might have enough local LAN bandwidth to perform replication using the same 10 GbE ports used for data access.

Even then, you should compare your available WAN bandwidth to your LAN bandwidth. If the available WAN bandwidth is significantly less than 10 GbE, you might need to use dedicated ports.

The one exception to this rule might be when all or many nodes in the cluster replicate data, in which case bandwidth utilization is typically spread across nodes.

If you are not using dedicated ports, the maximum transmission unit (MTU) size of the replication network should typically be the same as the MTU size of the data network.

Replication interval

If replication takes place in off-peak hours, you should be able to use data ports for replication even without a 10-GbE LAN connection.

If replication takes place during normal business hours, you need to consider the amount of data that will be replicated and whether it requires so much bandwidth that it could cause contention with data protocols. If network utilization by data protocols (SMB, NFS, iSCSI) is above 50%, you should use dedicated ports for intercluster communication, to allow for non-degraded performance if node failover occurs.

Port availability

If you determine that replication traffic is interfering with data traffic, you can migrate intercluster LIFs to any other intercluster-capable shared port on the same node.

You can also dedicate VLAN ports for replication. The bandwidth of the port is shared between all VLANs and the base port.
Use custom IPspaces to isolate replication traffic

You can use custom IPspaces to separate the interactions that a cluster has with its peers. Called designated intercluster connectivity, this configuration allows service providers to isolate replication traffic in multitenant environments.

Suppose, for example, that you want replication traffic between Cluster A and Cluster B to be separated from replication traffic between Cluster A and Cluster C. To accomplish this, you can create two IPspaces on Cluster A.

One IPspace contains the intercluster LIFs that you use to communicate with Cluster B. The other contains the intercluster LIFs that you use to communicate with Cluster C, as shown in the following illustration.

For custom IPspace configuration, see the Network Management Guide.

Configure intercluster LIFs

Configure intercluster LIFs on shared data ports

You can configure intercluster LIFs on ports shared with the data network. Doing so reduces the number of ports you need for intercluster networking.

Steps
1. List the ports in the cluster:

   network port show

   For complete command syntax, see the man page.

   The following example shows the network ports in cluster01:
cluster01:> network port show

<table>
<thead>
<tr>
<th>Speed (Mbps)</th>
<th>Node</th>
<th>Port</th>
<th>IPspace</th>
<th>Broadcast Domain</th>
<th>Link</th>
<th>MTU</th>
<th>Admin/Oper</th>
</tr>
</thead>
<tbody>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>------</td>
<td>--------------</td>
<td>------------------</td>
<td>------</td>
<td>--------</td>
<td>------------</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>------</td>
<td>--------------</td>
<td>------------------</td>
<td>------</td>
<td>--------</td>
<td>------------</td>
</tr>
</tbody>
</table>

2. Create intercluster LIFs on the system SVM:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>In ONTAP 9.6 and later:</strong></td>
<td>network interface create -vserver system_SVM -lif LIF_name -service -policy default-intercluster -home -node node -home-port port -address port_IP -netmask netmask</td>
</tr>
<tr>
<td><strong>In ONTAP 9.5 and earlier:</strong></td>
<td>network interface create -vserver system_SVM -lif LIF_name -role intercluster -home-node node -home -port port -address port_IP -netmask netmask</td>
</tr>
</tbody>
</table>

For complete command syntax, see the man page.

The following example creates intercluster LIFs `cluster01_icl01` and `cluster01_icl02`:
cluster01::> network interface create -vserver cluster01 -lif cluster01_icl01 -service-policy default-intercluster -home-node cluster01-01 -home-port e0c -address 192.168.1.201 -netmask 255.255.255.0

cluster01::> network interface create -vserver cluster01 -lif cluster01_icl02 -service-policy default-intercluster -home-node cluster01-02 -home-port e0c -address 192.168.1.202 -netmask 255.255.255.0

3. Verify that the intercluster LIFs were created:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>In ONTAP 9.6 and later:</td>
<td>network interface show -service-policy default-intercluster</td>
</tr>
<tr>
<td>In ONTAP 9.5 and earlier:</td>
<td>network interface show -role intercluster</td>
</tr>
</tbody>
</table>

For complete command syntax, see the man page.

cluster01::> network interface show -service-policy default-intercluster

<table>
<thead>
<tr>
<th>Logical</th>
<th>Status</th>
<th>Network</th>
<th>Current</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vserver</td>
<td>Interface</td>
<td>Admin/Oper Address/Mask</td>
<td>Node</td>
<td>Port</td>
</tr>
<tr>
<td>Home</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td></td>
</tr>
</tbody>
</table>

cluster01

cluster01_icl01

up/up 192.168.1.201/24  cluster01-01 e0c

true

cluster01_icl02

up/up 192.168.1.202/24  cluster01-02 e0c

true

4. Verify that the intercluster LIFs are redundant:
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>In ONTAP 9.6 and later:</td>
<td>network interface show -service-policy default-intercluster -failover</td>
</tr>
<tr>
<td>In ONTAP 9.5 and earlier:</td>
<td>network interface show -role intercluster -failover</td>
</tr>
</tbody>
</table>

For complete command syntax, see the man page.

The following example shows that the intercluster LIFs `cluster01_icl01` and `cluster01_icl02` on the e0c port will fail over to the e0d port.

```
custom01::> network interface show -service-policy default-intercluster -failover
```

<table>
<thead>
<tr>
<th>Vserver</th>
<th>Interface</th>
<th>Home:Port</th>
<th>Failover</th>
<th>Failover</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>cluster01_icl01</td>
<td>cluster01-01:e0c</td>
<td>local-only</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>192.168.1.201/24</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>cluster01_icl02</td>
<td>cluster01-02:e0c</td>
<td>local-only</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>192.168.1.201/24</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Configure intercluster LIFs on dedicated ports

You can configure intercluster LIFs on dedicated ports. Doing so typically increases the available bandwidth for replication traffic.

Steps
1. List the ports in the cluster:

   `network port show`

   For complete command syntax, see the man page.

   The following example shows the network ports in `cluster01`:
```
cluster01::> network port show

<table>
<thead>
<tr>
<th>Node</th>
<th>Port</th>
<th>IPspace</th>
<th>Broadcast Domain</th>
<th>Link</th>
<th>MTU</th>
<th>Admin/Oper</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>auto/1000</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>----------</td>
<td>------------------</td>
<td>------</td>
<td>-------</td>
<td>------------</td>
</tr>
<tr>
<td>cluster01-01</td>
<td>e0a</td>
<td>Cluster</td>
<td>Cluster</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
<tr>
<td>cluster01-01</td>
<td>e0b</td>
<td>Cluster</td>
<td>Cluster</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
<tr>
<td>cluster01-01</td>
<td>e0c</td>
<td>Default</td>
<td>Default</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
<tr>
<td>cluster01-01</td>
<td>e0d</td>
<td>Default</td>
<td>Default</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
<tr>
<td>cluster01-01</td>
<td>e0e</td>
<td>Default</td>
<td>Default</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
<tr>
<td>cluster01-01</td>
<td>e0f</td>
<td>Default</td>
<td>Default</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cluster01-02</td>
<td>e0a</td>
<td>Cluster</td>
<td>Cluster</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
<tr>
<td>cluster01-02</td>
<td>e0b</td>
<td>Cluster</td>
<td>Cluster</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
<tr>
<td>cluster01-02</td>
<td>e0c</td>
<td>Default</td>
<td>Default</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
<tr>
<td>cluster01-02</td>
<td>e0d</td>
<td>Default</td>
<td>Default</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
<tr>
<td>cluster01-02</td>
<td>e0e</td>
<td>Default</td>
<td>Default</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
<tr>
<td>cluster01-02</td>
<td>e0f</td>
<td>Default</td>
<td>Default</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
</tbody>
</table>
```

2. Determine which ports are available to dedicate to intercluster communication:

```bash
network interface show -fields home-port,curr-port
```

For complete command syntax, see the man page.

The following example shows that ports e0e and e0f have not been assigned LIFs:

```
ccluster01::> network interface show -fields home-port,curr-port

<table>
<thead>
<tr>
<th>vserver lif</th>
<th>home-port</th>
<th>curr-port</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster cluster01-01_clus1</td>
<td>e0a</td>
<td>e0a</td>
</tr>
<tr>
<td>Cluster cluster01-01_clus2</td>
<td>e0b</td>
<td>e0b</td>
</tr>
<tr>
<td>Cluster cluster01-02_clus1</td>
<td>e0a</td>
<td>e0a</td>
</tr>
<tr>
<td>Cluster cluster01-02_clus2</td>
<td>e0b</td>
<td>e0b</td>
</tr>
<tr>
<td>cluster01 cluster_mgmt</td>
<td>e0c</td>
<td>e0c</td>
</tr>
<tr>
<td>cluster01 cluster01-01_mgmt1</td>
<td>e0c</td>
<td>e0c</td>
</tr>
<tr>
<td>cluster01 cluster01-02_mgmt1</td>
<td>e0c</td>
<td>e0c</td>
</tr>
</tbody>
</table>
```

3. Create a failover group for the dedicated ports:
network interface failover-groups create -vserver system_SVM -failover-group failover_group -targets physical_or_logical_ports

The following example assigns ports e0e and e0f to the failover group intercluster01 on the system SVM cluster01:

    cluster01::> network interface failover-groups create -vserver cluster01 -failover-group intercluster01 -targets cluster01-01:e0e,cluster01-01:e0f,cluster01-02:e0e,cluster01-02:e0f

4. Verify that the failover group was created:

    network interface failover-groups show

For complete command syntax, see the man page.

    cluster01::> network interface failover-groups show

    Failover
    Vserver          Group            Targets
    ---------------- ----------------
    -----------------------------
    Cluster
    Cluster
    cluster01-01:e0a, cluster01-01:e0b, 
    cluster01-02:e0a, cluster01-02:e0b
    cluster01
    Default
    cluster01-01:e0c, cluster01-01:e0d, 
    cluster01-02:e0c, cluster01-02:e0d, 
    cluster01-01:e0e, cluster01-01:e0f
    cluster01-02:e0e, cluster01-02:e0f
    intercluster01
    intercluster01
    cluster01-01:e0e, cluster01-01:e0f
    cluster01-02:e0e, cluster01-02:e0f

5. Create intercluster LIFs on the system SVM and assign them to the failover group.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>In ONTAP 9.6 and later:</strong></td>
<td>network interface create -vserver system_SVM -lif LIF_name -service -policy default-intercluster -home -node node -home port port -address port_IP -netmask netmask -failover -group failover_group</td>
</tr>
</tbody>
</table>
Option | Description
--- | ---
**In ONTAP 9.5 and earlier:** | network interface create -vserver system_SVM -lif LIF_name -role intercluster -home-node node -home -port port -address port_IP -netmask netmask -failover-group failover_group

For complete command syntax, see the man page.

The following example creates intercluster LIFs cluster01_icl01 and cluster01_icl02 in the failover group intercluster01:

```
cluster01::> network interface create -vserver cluster01 -lif cluster01_icl01 -service-policy default-intercluster -home-node cluster01-01 -home-port e0e
-address 192.168.1.201
-netmask 255.255.255.0 -failover-group intercluster01

cluster01::> network interface create -vserver cluster01 -lif cluster01_icl02 -service-policy default-intercluster -home-node cluster01-02 -home-port e0e
-address 192.168.1.202
-netmask 255.255.255.0 -failover-group intercluster01
```

6. Verify that the intercluster LIFs were created:

Option | Description
--- | ---
**In ONTAP 9.6 and later:** | network interface show -service-policy default-intercluster

**In ONTAP 9.5 and earlier:** | network interface show -role intercluster

For complete command syntax, see the man page.
cluster01::> network interface show -service-policy default-intercluster

Current Is
Vserver     Interface  Admin/Oper Address/Mask       Node          Port
Home

---------- ---------- ---------- ------------------ --------------
------- ---- ------- ------- ------- ---------- -------

--- ----
cluster01

cluster01_icl01
  up/up       192.168.1.201/24   cluster01-01  e0e
true

cluster01_icl02
  up/up       192.168.1.202/24   cluster01-02  e0f
true

7. Verify that the intercluster LIFs are redundant:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>In ONTAP 9.6 and later:</td>
<td>network interface show -service-policy default-intercluster -failover</td>
</tr>
<tr>
<td>In ONTAP 9.5 and earlier:</td>
<td>network interface show -role intercluster -failover</td>
</tr>
</tbody>
</table>

For complete command syntax, see the man page.

The following example shows that the intercluster LIFs `cluster01_icl01` and `cluster01_icl02` on the SVM e0e port will fail over to the e0f port.

cluster01::> network interface show -service-policy default-intercluster -failover

Logical  Home  Failover  Failover
Vserver   Interface Node:Port Policy  Group
---------- --------------- ---------- ------ ---
cluster01
cluster01_icl01  cluster01-01:e0e  local-only

intercluster01

Failed Targets:  cluster01-01:e0e,
                 cluster01-01:e0f

cluster01_icl02  cluster01-02:e0e  local-only

intercluster01

Failed Targets:  cluster01-02:e0e,
                 cluster01-02:e0f
Configure intercluster LIFs in custom IPspaces

You can configure intercluster LIFs in custom IPspaces. Doing so allows you to isolate replication traffic in multitenant environments.

When you create a custom IPspace, the system creates a system storage virtual machine (SVM) to serve as a container for the system objects in that IPspace. You can use the new SVM as the container for any intercluster LIFs in the new IPspace. The new SVM has the same name as the custom IPspace.

Steps

1. List the ports in the cluster:

   network port show

   For complete command syntax, see the man page.

   The following example shows the network ports in cluster01:

   ```
   cluster01::> network port show

   (Mbps)
   Node  Port  IPspace  Broadcast Domain Link  MTU  Admin/Oper
   ------  ------  ---------  ---------------- -----  -------
   cluster01-01
   e0a    Cluster  Cluster  up     1500   auto/1000
   e0b    Cluster  Cluster  up     1500   auto/1000
   e0c    Default  Default  up     1500   auto/1000
   e0d    Default  Default  up     1500   auto/1000
   e0e    Default  Default  up     1500   auto/1000
   e0f    Default  Default  up     1500   auto/1000
   cluster01-02
   e0a    Cluster  Cluster  up     1500   auto/1000
   e0b    Cluster  Cluster  up     1500   auto/1000
   e0c    Default  Default  up     1500   auto/1000
   e0d    Default  Default  up     1500   auto/1000
   e0e    Default  Default  up     1500   auto/1000
   e0f    Default  Default  up     1500   auto/1000
   ```

2. Create custom IPspaces on the cluster:

   network ipspace create -ipspace ipspace

   The following example creates the custom IPspace ipspace-IC1:

   ```
   cluster01::> network ipspace create -ipspace ipspace-IC1
   ```
3. Determine which ports are available to dedicate to intercluster communication:

```bash
network interface show -fields home-port,curr-port
```

For complete command syntax, see the man page.

The following example shows that ports `e0e` and `e0f` have not been assigned LIFs:

```
class01::> network interface show -fields home-port,curr-port
vserver lif                  home-port curr-port
------- -------------------- --------- ---------
Cluster cluster01_clus1   e0a       e0a
Cluster cluster01_clus2   e0b       e0b
Cluster cluster02_clus1   e0a       e0a
Cluster cluster02_clus2   e0b       e0b
cluster01
    cluster_mgmt         e0c       e0c
cluster01
    cluster01-01_mgmt1   e0c       e0c
cluster01
    cluster01-02_mgmt1   e0c       e0c
```

4. Remove the available ports from the default broadcast domain:

```bash
network port broadcast-domain remove-ports -broadcast-domain Default -ports ports
```

A port cannot be in more than one broadcast domain at a time. For complete command syntax, see the man page.

The following example removes ports `e0e` and `e0f` from the default broadcast domain:

```
class01::> network port broadcast-domain remove-ports -broadcast-domain Default -ports
cluster01-01:e0e,cluster01-01:e0f,cluster01-02:e0e,cluster01-02:e0f
```

5. Verify that the ports have been removed from the default broadcast domain:

```bash
network port show
```

For complete command syntax, see the man page.

The following example shows that ports `e0e` and `e0f` have been removed from the default broadcast domain:
cluster01::> network port show

<table>
<thead>
<tr>
<th>Node</th>
<th>Port</th>
<th>IPspace</th>
<th>Broadcast Domain</th>
<th>Link</th>
<th>MTU</th>
<th>Admin/Oper</th>
</tr>
</thead>
<tbody>
<tr>
<td>------</td>
<td>------</td>
<td>---------</td>
<td>------------------</td>
<td>------</td>
<td>-------</td>
<td>------------</td>
</tr>
<tr>
<td>cluster01-01</td>
<td>e0a</td>
<td>Cluster</td>
<td>Cluster</td>
<td>up</td>
<td>9000</td>
<td>auto/1000</td>
</tr>
<tr>
<td></td>
<td>e0b</td>
<td>Cluster</td>
<td>Cluster</td>
<td>up</td>
<td>9000</td>
<td>auto/1000</td>
</tr>
<tr>
<td></td>
<td>e0c</td>
<td>Default</td>
<td>Default</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
<tr>
<td></td>
<td>e0d</td>
<td>Default</td>
<td>Default</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
<tr>
<td></td>
<td>e0e</td>
<td>Default</td>
<td>-</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
<tr>
<td></td>
<td>e0f</td>
<td>Default</td>
<td>-</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
<tr>
<td></td>
<td>e0g</td>
<td>Default</td>
<td>Default</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
<tr>
<td>cluster01-02</td>
<td>e0a</td>
<td>Cluster</td>
<td>Cluster</td>
<td>up</td>
<td>9000</td>
<td>auto/1000</td>
</tr>
<tr>
<td></td>
<td>e0b</td>
<td>Cluster</td>
<td>Cluster</td>
<td>up</td>
<td>9000</td>
<td>auto/1000</td>
</tr>
<tr>
<td></td>
<td>e0c</td>
<td>Default</td>
<td>Default</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
<tr>
<td></td>
<td>e0d</td>
<td>Default</td>
<td>Default</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
<tr>
<td></td>
<td>e0e</td>
<td>Default</td>
<td>-</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
<tr>
<td></td>
<td>e0f</td>
<td>Default</td>
<td>-</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
<tr>
<td></td>
<td>e0g</td>
<td>Default</td>
<td>Default</td>
<td>up</td>
<td>1500</td>
<td>auto/1000</td>
</tr>
</tbody>
</table>

6. Create a broadcast domain in the custom IPspace:

```
network port broadcast-domain create -ipspace ipspace -broadcast-domain broadcast_domain -mtu MTU -ports ports
```

The following example creates the broadcast domain `ipspace-IC1-bd` in the IPspace `ipspace-IC1`:

```
cluster01::> network port broadcast-domain create -ipspace ipspace-IC1 -broadcast-domain ipspace-IC1-bd -mtu 1500 -ports cluster01-01:e0e,cluster01-01:e0f, cluster01-02:e0e,cluster01-02:e0f
```

7. Verify that the broadcast domain was created:

```
network port broadcast-domain show
```

For complete command syntax, see the man page.
8. Create intercluster LIFs on the system SVM and assign them to the broadcast domain:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>In ONTAP 9.6 and later:</strong></td>
<td>network interface create -vserver system_SVM -lif LIF_name -service -policy default-intercluster -home -node node -home-port port -address port_IP -netmask netmask</td>
</tr>
<tr>
<td><strong>In ONTAP 9.5 and earlier:</strong></td>
<td>network interface create -vserver system_SVM -lif LIF_name -role intercluster -home-node node -home -port port -address port_IP -netmask netmask</td>
</tr>
</tbody>
</table>

The LIF is created in the broadcast domain that the home port is assigned to. The broadcast domain has a default failover group with the same name as the broadcast domain. For complete command syntax, see the man page.
The following example creates intercluster LIFs `cluster01_icl01` and `cluster01_icl02` in the broadcast domain `ipspace-IC1-bd`:

```
cluster01::> network interface create -vserver ipspace-IC1 -lif
             cluster01_icl01 -service-policy default-intercluster
             -home-node cluster01-01 -home-port e0e
             -address 192.168.1.201
             -netmask 255.255.255.0

cluster01::> network interface create -vserver ipspace-IC1 -lif
             cluster01_icl02 -service-policy default-intercluster
             -home-node cluster01-02 -home-port e0e
             -address 192.168.1.202
             -netmask 255.255.255.0
```

9. Verify that the intercluster LIFs were created:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>In ONTAP 9.6 and later:</strong></td>
<td>network interface show -service-policy default-intercluster</td>
</tr>
<tr>
<td><strong>In ONTAP 9.5 and earlier:</strong></td>
<td>network interface show -role intercluster</td>
</tr>
</tbody>
</table>

For complete command syntax, see the man page.

```
cluster01::> network interface show -service-policy default-intercluster

Current Is
Vserver  Interface  Admin/Oper Address/Mask       Node          Port
Home
------------------ ---------- ------------------ -------------- ------- ---- ------ ----
ipspace-IC1
   cluster01_icl01  up/up  192.168.1.201/24   cluster01-01  e0e
   true
   cluster01_icl02  up/up  192.168.1.202/24   cluster01-02  e0f
   true
```

10. Verify that the intercluster LIFs are redundant:
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>In ONTAP 9.6 and later:</strong></td>
<td>network interface show -service-policy default-intercluster -failover</td>
</tr>
<tr>
<td><strong>In ONTAP 9.5 and earlier:</strong></td>
<td>network interface show -role intercluster -failover</td>
</tr>
</tbody>
</table>

For complete command syntax, see the man page.

The following example shows that the intercluster LIFs `cluster01_icl01` and `cluster01_icl02` on the SVM `e0e` port fail over to the `e0f` port:

```
cluster01::> network interface show -service-policy default-intercluster -failover

Logical         Home                  Failover        Failover
Vserver  Interface       Node:Port             Policy          Group
------- --------------- --------------------- --------------- --------
ipspace-IC1
    cluster01_icl01 cluster01-01:e0e   local-only          intercluster01
    Failover Targets:  cluster01-01:e0e,
                        cluster01-01:e0f
    cluster01_icl02 cluster01-02:e0e   local-only          intercluster01
    Failover Targets:  cluster01-02:e0e,
                        cluster01-02:e0f
```

**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/YYYY
   ```
If you specify both \texttt{-generate-passphrase} and \texttt{-peer-addrs}, only the cluster whose intercluster LIFs are specified in \texttt{-peer-addrs} can use the generated password.

You can ignore the \texttt{-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 \texttt{-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 \texttt{vs1} and \texttt{vs2} on the local cluster:

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

<table>
<thead>
<tr>
<th>Passphrase: UCa+6lRVICxeL/gq1WrK7ShR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expiration Time: 6/7/2017 08:16:10 EST</td>
</tr>
<tr>
<td>Initial Allowed Vserver Peers: vs1,vs2</td>
</tr>
<tr>
<td>Intercluster LIF IP: 192.140.112.101</td>
</tr>
<tr>
<td>Peer Cluster Name: Clus_7ShR (temporary generated)</td>
</tr>
</tbody>
</table>

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 *
```

<table>
<thead>
<tr>
<th>Passphrase: UCa+6lRVICxeL/gq1WrK7ShR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expiration Time: 6/7/2017 08:16:10 EST</td>
</tr>
<tr>
<td>Initial Allowed Vserver Peers: vs1,vs2</td>
</tr>
<tr>
<td>Intercluster LIF IP: 192.140.112.101,192.140.112.102</td>
</tr>
<tr>
<td>Peer Cluster Name: Clus_7ShR (temporary generated)</td>
</tr>
</tbody>
</table>

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 \texttt{vs1} and \texttt{vs2} on the local cluster:
cluster peer create -generate-passphrase -offer-expiration 2days -initial-allowed-vserver-peers vs1,vs2

Passphrase: UCa+6lRVICeL/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:

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
Node       cluster-Name                Node-Name
Ping-Status               RDB-Health Cluster-Health  Avail...
---------- --------------------------- ---------  ---------------
--------
cluster01-01
cluster02    cluster02-01
Data: interface_reachable
ICMP: interface_reachable true       true            true
cluster02-02
Data: interface_reachable
ICMP: interface_reachable true       true            true

cluster01-02
cluster02    cluster02-01
Data: interface_reachable
ICMP: interface_reachable true       true            true
cluster02-02
Data: interface_reachable
ICMP: interface_reachable true       true            true

Other ways to do this in ONTAP
To perform these tasks with... | See this content...
---|---
The redesigned System Manager (available with ONTAP 9.7 and later) | Prepare for mirroring and vaulting
System Manager Classic (available with ONTAP 9.7 and earlier) | Volume disaster recovery preparation overview

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:

   `vserver peer permission show`

   ```bash
   cluster02::> vserver peer permission show
   Peer Cluster      Vserver               Applications
   -------------------  --------------------  -------------------
   cluster02          vs1,vs2               snapmirror
   ```

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

   `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`: 
3. Verify the SVM peer relationship:

vserver peer show

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

```
3. Verify the SVM peer relationship:

vserver peer show

```
cluster01::> vserver peer show

<table>
<thead>
<tr>
<th>Peer</th>
<th>Peer</th>
<th>Peering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote Vserver</td>
<td>Vserver</td>
<td>State</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>--------</td>
</tr>
<tr>
<td>----------</td>
<td>-----------</td>
<td>--------</td>
</tr>
<tr>
<td>pvs1</td>
<td>vs1</td>
<td>peered</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:

```
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`...
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:

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:

vserver peer show-all

For complete command syntax, see the man page.

The following example shows that the peer relationship between SVM\textsubscript{pvs1} and SVM\textsubscript{vs1} has been initiated:

<table>
<thead>
<tr>
<th>Peer</th>
<th>Peer</th>
<th>Peering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vserver</td>
<td>Vserver</td>
<td>State</td>
</tr>
<tr>
<td>-------</td>
<td>----------------</td>
<td>------------------</td>
</tr>
<tr>
<td>pvs1</td>
<td>vs1</td>
<td>initiated</td>
</tr>
</tbody>
</table>

3. On the data protection destination cluster, display the pending SVM peer relationship:

vserver peer show

For complete command syntax, see the man page.

The following example lists the pending peer relationships for cluster02:

<table>
<thead>
<tr>
<th>Peer</th>
<th>Peer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vserver</td>
<td>Vserver</td>
</tr>
<tr>
<td>-------</td>
<td>------------</td>
</tr>
<tr>
<td>vs1</td>
<td>pvs1</td>
</tr>
</tbody>
</table>

    pending

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

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:

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

5. Verify the SVM peer relationship:

```
vserver peer show
```

```
cluster01::> vserver peer show

Peer        Peer                           Peering
Remote
Vserver     Vserver     State        Peer Cluster      Applications
----------------- ----------- ------------ ----------------- --------------
------------------
pvs1        vs1         peered       cluster02         snapmirror
vs1
```

**Enable cluster peering encryption on an existing peer relationship**

Beginning with ONTAP 9.6, cluster peering encryption is enabled by default on all newly created cluster peering relationships. Cluster peering encryption uses a pre-shared key (PSK) and the Transport Security Layer (TLS) to secure cross-cluster peering communications. This adds an additional layer of security between the peered clusters.

**About this task**

If you are upgrading peered clusters to ONTAP 9.6 or later, and the peering relationship was created in ONTAP 9.5 or earlier, cluster peering encryption must be enabled manually after upgrading. Both clusters in the peering relationship must be running ONTAP 9.6 or later in order to enable cluster peering encryption.

**Steps**

1. On the destination cluster, enable encryption for communications with the source cluster:

   ```
   cluster peer modify source_cluster -auth-status-admin use-authentication -encryption-protocol-proposed tls-psk
   ```

2. When prompted enter a passphrase.

3. On the data protection source cluster, enable encryption for communication with the data protection destination cluster:

   ```
   cluster peer modify data_protection_destination_cluster -auth-status-admin use-authentication -encryption-protocol-proposed tls-psk
   ```
4. When prompted, enter the same passphrase entered on the destination cluster.

**Remove cluster peering encryption from an existing peer relationship**

By default, cluster peering encryption is enabled on all peer relationships created in ONTAP 9.6 or later. If you do not want to use encryption for cross-cluster peering communications, you can disable it.

**Steps**

1. On the destination cluster, modify communications with the source cluster to discontinue use of cluster peering encryption:
   - To remove encryption, but maintain authentication enter:
     ```bash
     cluster peer modify source_cluster -auth-status-admin use-authentication -encryption none
     ```
   - To remove encryption and authentication, enter:
     ```bash
     cluster peer modify source_cluster -auth-status no-authentication
     ```
2. When prompted enter a passphrase.
3. On the source cluster, disable encryption for communication with the destination cluster:
   - To remove encryption, but maintain authentication enter:
     ```bash
     cluster peer modify destination_cluster -auth-status-admin use-authentication -encrypt none
     ```
   - To remove encryption and authentication, enter:
     ```bash
     cluster peer modify destination_cluster -auth-status no-authentication
     ```
4. When prompted, enter the same passphrase entered on the destination cluster.

**Where to find additional information**

You can learn more about tasks related to cluster and SVM peering in NetApp’s extensive documentation library.

- **ONTAP concepts**
  
  Describes the concepts that inform ONTAP data management software, including data protection and transfer.

- **Data protection**
  
  Describes how to use the ONTAP CLI to perform SnapMirror replication.

- **Volume disaster recovery preparation**
  
  Describes how to use System Manager to quickly configure a destination volume for disaster recovery.

- **Volume disaster recovery preparation**
  
  Describes how to use System Manager to quickly recover a destination volume after a disaster.
• **Volume backup using SnapVault**
  Describes how to use System Manager to quickly configure a SnapVault relationship between volumes.

• **Volume restore management using SnapVault**
  Describes how to use System Manager to quickly restore files from a destination volume in a SnapVault relationship.

• **Archive and compliance using SnapLock technology**
  Describes how to replicate WORM files in a SnapLock volume.