



# **Data protection and disaster recovery**

## **ONTAP 9**

NetApp  
April 27, 2023

This PDF was generated from [https://docs.netapp.com/us-en/ontap/concept\\_dp\\_overview.html](https://docs.netapp.com/us-en/ontap/concept_dp_overview.html) on April 27, 2023. Always check docs.netapp.com for the latest.

# Table of Contents

- Data protection and disaster recovery . . . . . 1
  - Data protection with System Manager . . . . . 1
  - Cluster and SVM peering with the CLI . . . . . 14
  - Data protection with the CLI . . . . . 40

# Data protection and disaster recovery

## Data protection with System Manager

### Data protection overview with System Manager

The topics in this section show you how to configure and manage data protection with System Manager in ONTAP 9.7 and later releases.

If you are using System Manager in ONTAP 9.7 or earlier, see [ONTAP System Manager Classic documentation](#)

Protect your data by creating and managing Snapshot copies, mirrors, vaults, and mirror-and-vault relationships.

*SnapMirror* is disaster recovery technology, designed for failover from primary storage to secondary storage at a geographically remote site. As its name implies, SnapMirror creates a replica, or mirror, of your working data in secondary storage from which you can continue to serve data in the event of a catastrophe at the primary site.

A *vault* is designed for disk-to-disk Snapshot copy replication for standards compliance and other governance-related purposes. In contrast to a SnapMirror relationship, in which the destination usually contains only the Snapshot copies currently in the source volume, a vault destination typically retains point-in-time Snapshot copies created over a much longer period.

Beginning with ONTAP 9.10.1, you can create data protection relationships between S3 buckets using S3 SnapMirror. Destination buckets can be on local or remote ONTAP systems, or on non-ONTAP systems such as StorageGRID and AWS. For more information, see [S3 SnapMirror overview](#).

### Create custom data protection policies

You can create custom data protection policies with System Manager when the existing default protection policies are not appropriate for your needs. Beginning with ONTAP 9.11.1, you can use System Manager to create custom mirror and vault policies, to display and select legacy policies. This capability is also available in ONTAP 9.8P12 and later patches of ONTAP 9.8.

Create custom protection policies on both the source and destination cluster.

#### Steps

1. Click **Protection > Local Policy Settings**.
2. Under **Protection Policies**, click .
3. In the **Protection Policies** pane, click  **Add**.
4. Enter the new policy name, and select the policy scope.
5. Choose a policy type. To add a vault-only or mirror-only policy, choose **Asynchronous**, and click **Use a legacy policy type**.
6. Complete the required fields.
7. Click **Save**.


8. Repeat these steps on the other cluster.

## Configure Snapshot copies

You can create Snapshot copy policies to specify the maximum number of Snapshot copies that are automatically created and how frequently they are created. The policy specifies when to create Snapshot copies, how many copies to retain, and how to name them.

This procedure creates a Snapshot copy policy on the local cluster only.

### Steps

1. Click **Protection > Overview > Local Policy Settings**.
2. Under **Snapshot Policies**, click , and then click **+ Add**.
3. Type the policy name, select the policy scope, and under **Schedules**, click **+ Add** to enter the schedule details.

## Calculate reclaimable space before deleting Snapshot copies

Beginning with ONTAP 9.10.1, you can use System Manager to select Snapshot copies you want to delete and calculate the reclaimable space before you delete them.

### Steps

1. Click **Storage > Volumes**.
2. Select the volume from which you want to delete Snapshot copies.
3. Click **Snapshot Copies**.
4. Select one or more Snapshot copies.
5. Click **Calculate Reclaimable Space**.

## Enable or disable client access to Snapshot copy directory


Beginning with ONTAP 9.10.1, you can use System Manager to enable or disable client systems to access to a Snapshot copy directory on a volume. Enabling access makes the Snapshot copy directory visible to clients and allows Windows clients to map a drive to the Snapshot copies directory to view and access its contents.

You can enable or disable access to a volume's Snapshot copy directory by editing the volume settings or by editing the volume's share settings.

### Enable or disable client access to Snapshot copy directory by editing a volume

The Snapshot copy directory on a volume is accessible to clients by default.

### Steps

1. Click **Storage > Volumes**.
2. Select the volume containing the Snapshot copies directory you want to either show or hide.
3. Click  and select **Edit**.

4. In the **Snapshot Copies (Local) Settings** section, select or deselect **Show the Snapshot copies directory to clients**.
5. Click **Save**.

### Enable or disable client access to Snapshot copy directory by editing a share

The Snapshot copy directory on a volume is accessible to clients by default.

#### Steps

1. Click **Storage > Shares**.
2. Select the volume containing the Snapshot copies directory you want to either show or hide.
3. Click  and select **Edit**.
4. In the **Share Properties** section, select or deselect **Allow clients to access Snapshot copies directory**.
5. Click **Save**.

### Recover from Snapshot copies

You can recover a volume to an earlier point in time by restoring from a Snapshot copy.

This procedure restores a volume from a Snapshot copy.

#### Steps

1. Click **Storage** and select a volume.
2. Under **Snapshot Copies**, click  next to the Snapshot copy you want to restore, and select **Restore**.

### Prepare for mirroring and vaulting

You can protect your data by replicating it to a remote cluster for data backup and disaster recovery purposes.

Several default protection policies are available. You must have created your protection policies if you want to use custom policies.



#### Steps

1. In the local cluster, click **Protection > Overview**.
2. Expand **Intercluster Settings**. Click **Add Network Interfaces** and add intercluster network interfaces for the cluster.

Repeat this step on the remote cluster.

3. In the remote cluster, click **Protection > Overview**. Click  in the Cluster Peers section and click **Generate**

## Passphrase.

4. Copy the generated passphrase and paste it in the local cluster.
5. In the local cluster, under Cluster Peers, click **Peer Clusters** and peer the local and remote clusters.
6. Optionally, under Storage VM Peers, click  and then **Peer Storage VMs** to peer the storage VMs.
7. Click **Protect Volumes** to protect your volumes. To protect your LUNs, click **Storage > LUNs**, select a LUN to protect, and then click .

Select the protection policy based on the type of data protection you need.

8. To verify the volumes and LUNs are successfully protected from the local cluster, click **Storage > Volumes** or **Storage > LUNs** and, expand the volume/LUN view.

## Other ways to do this in ONTAP

To perform these tasks with...	See this content...
System Manager Classic (available with ONTAP 9.7 and earlier)	<a href="#">Volume disaster recovery preparation overview</a>
The ONTAP command line interface	<a href="#">Create a cluster peer relationship</a>

## Configure mirrors and vaults

Create a mirror and vault of a volume to protect data in case of a disaster and to have multiple archived versions of data to which you can roll back. Beginning with ONTAP 9.11.1, you can use System Manager to select pre-created and custom mirror and vault policies, to display and select legacy policies, and to override the transfer schedules defined in a protection policy when protecting volumes and storage VMs. This capability is also available in ONTAP 9.8P12 and later patches of ONTAP 9.8.



If you are using ONTAP 9.8P12 or later ONTAP 9.8 patch release and you configured SnapMirror using System Manager, you should use ONTAP 9.9.1P13 or later and ONTAP 9.10.1P10 or later patch releases if you plan to upgrade to ONTAP 9.9.1 or ONTAP 9.10.1 releases.

This procedure creates a data protection policy on a remote cluster. The source cluster and destination cluster use intercluster network interfaces for exchanging data. The procedure assumes the [intercluster network interfaces are created and the clusters containing the volumes are peered](#) (paired). You can also peer storage VMs for data protection; however, if storage VMs are not peered, but permissions are enabled, storage VMs are automatically peered when the protection relationship is created.



## Steps

1. Select the volume or LUN to protect: click **Storage > Volumes** or **Storage > LUNs**, and then click the desired volume or LUN name.
2. Click .

3. Select the destination cluster and storage VM.
4. The asynchronous policy is selected by default. To select a synchronous policy, click **More Options**.
5. Click **Protect**.
6. Click the **SnapMirror (Local or Remote)** tab for the selected volume or LUN to verify that protection is set up correctly.

#### Other ways to do this in ONTAP


To perform these tasks with...	See this content...
System Manager Classic (available with ONTAP 9.7 and earlier)	<a href="#">Volume backup using SnapVault overview</a>
The ONTAP command line interface	<a href="#">Create a replication relationship</a>

## Resynchronize a protection relationship

When your original source volume is available again after a disaster, you can resynchronize data from the destination volume and reestablish the protection relationship.

This procedure replaces the data in the original source volume in an asynchronous relationship so that you can start serving data from the original source volume again and resume the original protection relationship.

#### Steps

1. Click **Protection > Relationships** and then click the broken off relationship you want to resynchronize.
2. Click  and then select **Resync**.
3. Under **Relationships**, monitor the resynchronization progress by checking the relationship state. The state changes to "Mirrored" when resynchronization is complete.

## Restore a volume from an earlier Snapshot copy

When data in a volume is lost or corrupted, you can roll back your data by restoring from an earlier Snapshot copy.

This procedure replaces the current data on the source volume with data from an earlier Snapshot copy version. You should perform this task on the destination cluster.

#### Steps

1. Click **Protection > Relationships**, and then click the source volume name.
2. Click  and then select **Restore**.
3. Under **Source**, the source volume is selected by default. Click **Other Volume** if you want to choose a volume other than the source.
4. Under **Destination**, choose the Snapshot copy you want to restore.
5. If your source and destination are located on different clusters, on the remote cluster, click **Protection > Relationships** to monitor the restore progress.

## Other ways to do this in ONTAP

To perform these tasks with...	See this content...
System Manager Classic (available with ONTAP 9.7 and earlier)	<a href="#">Volume restore using SnapVault overview</a>
The ONTAP command line interface	<a href="#">Restore the contents of a volume from a SnapMirror destination</a>

## Recover from Snapshot copies

You can recover a volume to an earlier point in time by restoring from a Snapshot copy.

This procedure restores a volume from a Snapshot copy.

### Steps

1. Click **Storage** and select a volume.
2. Under **Snapshot Copies**, click  next to the Snapshot copy you want to restore, and select **Restore**.

## Restore to a new volume

Beginning with ONTAP 9.8, you can use System Manager to restore backed up data on the destination volume to a volume other than the original source.

When you restore to a different volume, you can select an existing volume, or you can create a new volume.

### Steps

1. Select the desired protection relationship: click **Protection > Relationships**.
2. Click  and click **Restore**.
3. In the **Source** section, select **Other Volume** and select the cluster and Storage VM.
4. Select either **Existing volume** or **Create a new volume**.
5. If you are creating a new volume, enter the volume name.
6. In the **Destination** section, select the Snapshot copy to restore.
7. Click **Save**.
8. Under **Relationships**, monitor the restore progress by viewing **Transfer Status** for the relationship.

## Reverse Resynchronizing a Protection Relationship

Beginning with ONTAP 9.8, you can use System Manager to perform a reverse resynchronization operation to delete an existing protection relationship and reverse the functions of the source and destination volumes. Then you use the destination volume to serve data while you repair or replace the source, update the source, and reestablish the original configuration of the systems.

When you perform a reverse resynch operation, any data on the source volume that is newer than the data in the common Snapshot copy is deleted.

### Steps



1. Select the desired protection relationship: click **Protection > Relationships**.
2. Click  and click **Reverse Resync**.
3. Under **Relationships**, monitor the reverse resynchronization progress by viewing **Transfer Status** for the relationship.

## Serve data from a SnapMirror destination

To serve data from a mirror destination when a source becomes unavailable, stop scheduled transfers to the destination, and then break the SnapMirror relationship to make the destination writable.



### Steps

1. Select the desired protection relationship: click **Protection > Relationships**, and then click the desired volume name.
2. Click .
3. Stop scheduled transfers : click **Pause**.
4. Make the destination writable: click **Break**.
5. Go to the main **Relationships** page to verify that the relationship state displays as "broken off".

### Next steps:

When the disabled source volume is available again, you should resynchronize the relationship to copy the current data to the original source volume. This process replaces the data on the original source volume.

### Other ways to do this in ONTAP

To perform these tasks with...	See this content...
System Manager Classic (available with ONTAP 9.7 and earlier)	<a href="#">Volume disaster recovery overview</a>
The ONTAP command line interface	<a href="#">Activate the destination volume</a>

## Configure storage VM disaster recovery

Using System Manager, you can create an storage VM disaster recovery (storage VM DR) relationship to replicate one storage VM configuration to another. In the event of a disaster at the primary site, you can quickly activate the destination storage VM.

Complete this procedure from the destination. If you need to create a new protection policy, for instance, when your source storage VM has SMB configured, you should use System Manager to create the policy and select the **Copy source storage VM configuration** option in the **Add Protection Policy** window.

For details see [Create custom data protection policies](#).

## Steps

1. On the destination cluster, click **Protection > Relationships**.
2. Under **Relationships**, click **Protect** and choose **Storage VMs (DR)**.
3. Select a protection policy. If you created a custom protection policy, select it, then choose the source cluster and storage VM you want to replicate. You can also create a new destination storage VM by entering a new storage VM name.
4. Click **Save**.

## Serve data from an SVM DR destination

Beginning with ONTAP 9.8, you can use System Manager to activate a destination storage VM after a disaster. Activating the destination storage VM makes the SVM destination volumes writable and enables you to serve data to clients.

## Steps

1. If the source cluster is accessible, verify that the SVM is stopped: navigate to **Storage > Storage VMs** and check the **State** column for the SVM.
2. If the source SVM state is "Running", stop it: select  and choose **Stop**.
3. On the destination cluster, locate the desired protection relationship: navigate to **Protection > Relationships**.
4. Click  and choose **Activate Destination Storage VM**.

## Reactivate a source storage VM

Beginning with ONTAP 9.8, you can use System Manager to reactivate a source storage VM after a disaster. Reactivating the source storage VM stops the destination storage VM, and it reenables replication from the source to the destination.

## About this task

When you reactivate the source storage VM, System Manager performs the following operations in the background:

- Creates a reverse SVM DR relationship from the original destination to original source using SnapMirror resync
- Stops the destination SVM
- Updates the SnapMirror relationship
- Breaks the SnapMirror relationship
- Restarts the original SVM
- Issues a SnapMirror resync of the original source back to the original destination
- Cleans up the SnapMirror relationships

## Steps

1. Select the desired protection relationship: click **Protection > Relationships**.
2. Click  and click **Reactivate Source Storage VM**.
3. Under **Relationships**, monitor the source reactivation progress by viewing **Transfer Status** for the

protection relationship.

## Resynchronize a destination storage VM

Beginning with ONTAP 9.8, you can use System Manager to resynchronize the data and configuration details from the source storage VM to the destination storage VM in a broken protection relationship and reestablish the relationship.

ONTAP 9.11.1 introduces an option to bypass a full data warehouse rebuild when you perform a disaster recovery rehearsal, enabling you to return to production faster.

You perform the resync operation only from the destination of the original relationship. The resync deletes any data in the destination storage VM that is newer than the data in the source storage VM.

### Steps

1. Select the desired protection relationship: click **Protection > Relationships**.
2. Optionally, select **Perform a quick resync** to bypass a full data warehouse rebuild during a disaster recovery rehearsal.
3. Click  and click **Resync**.
4. Under **Relationships**, monitor the resynchronization progress by viewing **Transfer Status** for the relationship.

## Back up data to the cloud using SnapMirror

Beginning with ONTAP 9.9.1, you can back up your data to the cloud and to restore your data from cloud storage to a different volume by using System Manager. You can use either StorageGRID or ONTAP S3 as your cloud object store.

Before using the SnapMirror Cloud feature, you should request a SnapMirror Cloud API license key from the NetApp Support Site: [Request SnapMirror Cloud API license key](#).

Following the instructions, you should provide a simple description of your business opportunity and request the API key by sending an email to the provided email address. You should receive an email response within 24 hours with further instructions on how to acquire the API key.

### Add a cloud object store

Before you configure SnapMirror Cloud backups, you need to add a StorageGRID or ONTAP S3 cloud object store.

### Steps

1. Click **Protection > Overview > Cloud Object Stores**.
2. Click  **Add**.

### Back up using the default policy

You can quickly configure a SnapMirror Cloud backup for an existing volume using the default cloud protection policy, DailyBackup.

### Steps

1. Click **Protection > Overview** and select **Back Up Volumes to Cloud**.

2. If this is your first time backing up to the cloud, enter your SnapMirror Cloud API license key in the license field as indicated.
3. Click **Authenticate and Continue**.
4. Select a source volume.
5. Select a cloud object store.
6. Click **Save**.

### Create a custom cloud backup policy

If you do not want to use the default DailyBackup cloud policy for your SnapMirror Cloud backups, you can create your own policy.

#### Steps

1. Click **Protection > Overview > Local Policy Settings** and select **Protection Policies**.
2. Click **Add** and enter the new policy details.
3. In the **Policy Type** section, select **Back up to Cloud** to indicate that you are creating a cloud policy.
4. Click **Save**.

### Create a backup from the Volumes page

You can use the System Manager **Volumes** page to when you want to select and create cloud backups for multiple volumes at one time or when you want to use a custom protection policy.

#### Steps

1. Click **Storage > Volumes**.
2. Select the volumes you want to back up to the cloud, and click **Protect**.
3. In the **Protect Volume** window, click **More Options**.
4. Select a policy.


You can select the default policy, DailyBackup, or a custom cloud policy you created.

5. Select a cloud object store.
6. Click **Save**.

### Restore from the cloud

You can use System Manager to restore backed up data from cloud storage to a different volume on the source cluster.

#### Steps


1. Click **Storage > Volumes**.
2. Select the **Back Up to Cloud** tab.
3. Click  next to the source volume you want to restore, and select **Restore**.
4. Under **Source**, select a storage VM and then enter the name of the volume to which you want the data restored.
5. Under **Destination**, select the Snapshot copy you want to restore.

6. Click **Save**.

## Delete a SnapMirror Cloud relationship

You can use System Manager to delete a cloud relationship.


### Steps

1. Click **Storage > Volumes** and select the volume you want to delete.
2. Click  next to the source volume and select **Delete**.
3. Select **Delete the cloud object store endpoint (optional)** if you want to delete the cloud object store endpoint.
4. Click **Delete**.

## Remove a cloud object store

You can use System Manager to remove a cloud object store if it is not part of a cloud backup relationship. When a cloud object store is part of a cloud backup relationship, it cannot be deleted.

### Steps

1. Click **Protection > Overview > Cloud Object Stores**.
2. Select the object store you want to delete, click  and select **Delete**.

## Back up data using Cloud Backup

Beginning with ONTAP 9.9.1, you can use System Manager to back up data in the cloud using Cloud Backup.



Cloud Backup supports FlexVol read-write volumes and data-protection (DP) volumes. FlexGroup volumes and SnapLock volumes are not supported.

### Before you begin

You should perform the following procedures to establish an account in BlueXP. For the service account, you need to create the role as "Account Admin". (Other service account roles do not have the required privileges needed to establish a connection from System Manager.)

1. [Create an account in BlueXP](#).
2. [Create a connector in BlueXP](#) with one of the following cloud providers:
  - Microsoft Azure
  - Amazon Web Services (AWS)
  - Google Cloud Platform (GCP)
  - StorageGrid (ONTAP 9.10.1)



Beginning with ONTAP 9.10.1, you can select StorageGrid as a cloud backup provider, but only if BlueXP is deployed on premises. The BlueXP connector must be installed on premises and available through the BlueXP software-as-a-service (SaaS) application.

3. [Subscribe to Cloud Backup Service in BlueXP](#) (requires the appropriate license).

4. [Generate an access key and a secret key using BlueXP.](#)

## Register the cluster with BlueXP

You can register the cluster with BlueXP by using either BlueXP or System Manager.

### Steps

1. In System Manager, go to **Protection Overview**.
2. Under **Cloud Backup Service**, provide the following details:
  - Client ID
  - Client secret key
3. Select **Register and Continue**.

## Enable Cloud Backup

After the cluster is registered with BlueXP, you need to enable the Cloud Backup and initiate the first backup to the cloud.

### Steps

1. In System Manager, click **Protection > Overview**, then scroll to the **Cloud Backup Service** section.
2. Enter the **Client ID** and **Client Secret**.



Beginning with ONTAP 9.10.1, you can learn about the cost of using the cloud by clicking **Learn more about the cost of using the cloud**.

3. Click **Connect and Enable Cloud Backup Service**.
4. On the **Enable Cloud Backup Service** page, provide the following details, depending on the provider you selected.

For this cloud provider...	Enter the following data...
Azure	<ul style="list-style-type: none"><li>• Azure Subscription ID</li><li>• Region</li><li>• Resource group name (existing or new)</li></ul>
AWS	<ul style="list-style-type: none"><li>• AWS Account ID</li><li>• Access key</li><li>• Secret key</li><li>• Region</li></ul>
Google Cloud Project (GCP)	<ul style="list-style-type: none"><li>• Google Cloud Project name</li><li>• Google Cloud Access key</li><li>• Google Cloud Secret key</li><li>• Region</li></ul>

StorageGrid  
(ONTAP 9.10.1 and later, and only  
for on-premises deployment of  
BlueXP)

- Server
- SG Access Key
- SG Secret Key

5. Select a **Protection policy**:

- **Existing policy**: Choose an existing policy.
- **New Policy**: Specify a name and set up a transfer schedule.



Beginning with ONTAP 9.10.1, you can specify whether you want to enable archiving with Azure or AWS.



If you enable archiving for a volume with Azure or AWS, you cannot disable the archiving.

If you enable archiving for Azure or AWS, specify the following:

- The number of days after which the volume is archived.
- The number of backups to retain in the archive. Specify “0” (zero) to archive up to the latest backup.
- For AWS, select the archive storage class.

6. Select the volumes you want to back up.

7. Select **Save**.

### Edit the protection policy used for Cloud Backup

You can change which protection policy is used with Cloud Backup.

#### Steps

1. In System Manager, click **Protection > Overview**, then scroll to the **Cloud Backup Service** section.
2. Click , then **Edit**.
3. Select a **Protection policy**:
  - **Existing policy**: Choose an existing policy.
  - **New Policy**: Specify a name and set up a transfer schedule.



Beginning with ONTAP 9.10.1, you can specify whether you want to enable archiving with Azure or AWS.



If you enable archiving for a volume with Azure or AWS, you cannot disable the archiving.

If you enable archiving for Azure or AWS, specify the following:

- The number of days after which the volume is archived.
- The number of backups to retain in the archive. Specify “0” (zero) to archive up to the latest backup.

- For AWS, select the archive storage class.

4. Select **Save**.

## Protect new volumes or LUNs on the cloud

When you create a new volume or LUN, you can establish a SnapMirror protection relationship that enables backing up to the cloud for the volume or LUN.

### Before you begin

- You should have a SnapMirror license.
- Intercluster LIFs should be configured.
- NTP should be configured.
- Cluster must be running ONTAP 9.9.1.

### About this task

You cannot protect new volumes or LUNs on the cloud for the following cluster configurations:

- The cluster cannot be in a MetroCluster environment.
- SVM-DR is not supported.
- FlexGroups cannot be backed up using Cloud Backup.

### Steps

1. When provisioning a volume or LUN, on the **Protection** page in System Manager, select the checkbox labeled **Enable SnapMirror (Local or Remote)**.
2. Select the Cloud Backup policy type.
3. If the Cloud Backup is not enabled, select **Enable Cloud Backup Service**.

## Protect existing volumes or LUNs on the cloud

You can establish a SnapMirror protection relationship for existing volumes and LUNs.

### Steps

1. Select an existing volume or LUN, and click **Protect**.
2. On the **Protect Volumes** page, specify **Backup using Cloud Backup Service** for the protection policy.
3. Click **Protect**.
4. On the **Protection** page, select the checkbox labeled **Enable SnapMirror (Local or Remote)**.
5. Select **Enable Cloud Backup Service**.

## Restore data from backup files

You can perform backup management operations, such as restoring data, updating relationships, and deleting relationships, only when using the BlueXP interface. Refer to [Restoring data from backup files](#) for more information.

# 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
```

### Prerequisites for cluster peering

Before you set up cluster peering, you should confirm that the connectivity, port, IP address, subnet, firewall, and cluster-naming requirements are met.

#### 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.



Beginning with ONTAP 9.10.1, firewall policies are deprecated and wholly replaced with LIF service policies. For more information, see [Configure firewall policies for LIFs](#).

### 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
```

(Mbps)						Speed
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
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

## 2. Create intercluster LIFs on the system SVM:

Option	Description
<b>In ONTAP 9.6 and later:</b>	<code>network interface create -vserver <i>system_SVM</i> -lif <i>LIF_name</i> -service -policy default-intercluster -home -node <i>node</i> -home-port <i>port</i> -address <i>port_IP</i> -netmask <i>netmask</i></code>
<b>In ONTAP 9.5 and earlier:</b>	<code>network interface create -vserver <i>system_SVM</i> -lif <i>LIF_name</i> -role intercluster -home-node <i>node</i> -home -port <i>port</i> -address <i>port_IP</i> -netmask <i>netmask</i></code>

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:

Option	Description
<b>In ONTAP 9.6 and later:</b>	network interface show -service-policy default-intercluster
<b>In ONTAP 9.5 and earlier:</b>	network interface show -role intercluster

For complete command syntax, see the man page.

```
cluster01::> network interface show -service-policy default-intercluster
Current Is
Vserver      Logical      Status      Network      Current
Home
-----
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:

Option	Description
In ONTAP 9.6 and later:	network interface show -service-policy default-intercluster -failover
In ONTAP 9.5 and earlier:	network interface show -role intercluster -failover

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.

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

Vserver	Logical Interface	Home Node:Port	Failover Policy	Failover Group
cluster01	cluster01_icl01	cluster01-01:e0c	local-only	
192.168.1.201/24			Failover Targets: cluster01-01:e0c, cluster01-01:e0d	
	cluster01_icl02	cluster01-02:e0c	local-only	
192.168.1.201/24			Failover Targets: cluster01-02:e0c, cluster01-02:e0d	

### 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
```

(Mbps)		Speed				
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. Determine which ports are available to dedicate to intercluster communication:

```
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:

```
cluster01::> network interface show -fields home-port,curr-port
vserver lif                home-port curr-port
-----
Cluster cluster01-01_clus1 e0a      e0a
Cluster cluster01-01_clus2 e0b      e0b
Cluster cluster01-02_clus1 e0a      e0a
Cluster cluster01-02_clus2 e0b      e0b
cluster01
    cluster_mgmt            e0c      e0c
cluster01
    cluster01-01_mgmt1      e0c      e0c
cluster01
    cluster01-02_mgmt1      e0c      e0c
```

## 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

Vserver          Group          Failover
-----          -
Targets
-----
Cluster
Cluster
cluster01        Cluster
                  cluster01-01:e0a, cluster01-01:e0b,
                  cluster01-02:e0a, cluster01-02:e0b
                  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
                  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.

Option	Description
In ONTAP 9.6 and later:	network interface create -vserver <i>system_SVM</i> -lif <i>LIF_name</i> -service -policy default-intercluster -home -node <i>node</i> -home- port <i>port</i> -address <i>port_IP</i> -netmask <i>netmask</i> -failover -group <i>failover_group</i>

Option	Description
<b>In ONTAP 9.5 and earlier:</b>	<code>network interface create -vserver <i>system_SVM</i> -lif <i>LIF_name</i> -role intercluster -home-node <i>node</i> -home-port <i>port</i> -address <i>port_IP</i> -netmask <i>netmask</i> -failover-group <i>failover_group</i></code>

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
<b>In ONTAP 9.6 and later:</b>	<code>network interface show -service-policy default-intercluster</code>
<b>In ONTAP 9.5 and earlier:</b>	<code>network interface show -role intercluster</code>

For complete command syntax, see the man page.

```

cluster01::> network interface show -service-policy default-intercluster
          Logical      Status      Network      Current
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:

Option	Description
<b>In ONTAP 9.6 and later:</b>	<code>network interface show -service-policy default-intercluster -failover</code>
<b>In ONTAP 9.5 and earlier:</b>	<code>network interface show -role intercluster -failover</code>

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
                                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 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
```

						Speed
(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:

```
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:

```
cluster01::> 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:

```
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:

```
cluster01::> 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:

```
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
```

Node	Port	IPspace	Broadcast Domain	Link	MTU	Speed (Mbps) Admin/Oper
-----						
cluster01-01						
	e0a	Cluster	Cluster	up	9000	auto/1000
	e0b	Cluster	Cluster	up	9000	auto/1000
	e0c	Default	Default	up	1500	auto/1000
	e0d	Default	Default	up	1500	auto/1000
	e0e	Default	-	up	1500	auto/1000
	e0f	Default	-	up	1500	auto/1000
	e0g	Default	Default	up	1500	auto/1000
cluster01-02						
	e0a	Cluster	Cluster	up	9000	auto/1000
	e0b	Cluster	Cluster	up	9000	auto/1000
	e0c	Default	Default	up	1500	auto/1000
	e0d	Default	Default	up	1500	auto/1000
	e0e	Default	-	up	1500	auto/1000
	e0f	Default	-	up	1500	auto/1000
	e0g	Default	Default	up	1500	auto/1000

#### 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.

```

cluster01::> network port broadcast-domain show
IPspace Broadcast
Name      Domain Name      MTU      Port List
-----
Cluster Cluster      9000
      cluster01-01:e0a      complete
      cluster01-01:e0b      complete
      cluster01-02:e0a      complete
      cluster01-02:e0b      complete
Default Default      1500
      cluster01-01:e0c      complete
      cluster01-01:e0d      complete
      cluster01-01:e0f      complete
      cluster01-01:e0g      complete
      cluster01-02:e0c      complete
      cluster01-02:e0d      complete
      cluster01-02:e0f      complete
      cluster01-02:e0g      complete
ipspace-IC1
      ipspace-IC1-bd
      1500
      cluster01-01:e0e      complete
      cluster01-01:e0f      complete
      cluster01-02:e0e      complete
      cluster01-02:e0f      complete

```

8. Create intercluster LIFs on the system SVM and assign them to the broadcast domain:

Option	Description
<b>In ONTAP 9.6 and later:</b>	<pre> network interface create -vserver system_SVM -lif LIF_name -service -policy default-intercluster -home -node node -home-port port -address port_IP -netmask netmask </pre>
<b>In ONTAP 9.5 and earlier:</b>	<pre> network interface create -vserver system_SVM -lif LIF_name -role intercluster -home-node node -home -port port -address port_IP -netmask netmask </pre>

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:

Option	Description
In ONTAP 9.6 and later:	<code>network interface show -service-policy default-intercluster</code>
In ONTAP 9.5 and earlier:	<code>network interface show -role intercluster</code>

For complete command syntax, see the man page.

```
cluster01::> network interface show -service-policy default-intercluster
Logical      Status      Network      Current
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:



Option	Description
In ONTAP 9.6 and later:	network interface show -service-policy default-intercluster -failover
In ONTAP 9.5 and earlier:	network interface show -role intercluster -failover

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
```

Vserver	Logical Interface	Home Node:Port	Failover Policy	Failover 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
HH:MM:SS|1...7days|1...168hours -peer-addr peer_LIF_IPs -initial-allowed-vserver
```

```
-peers svm_name,...|* -ipspace ipspace
```

If you specify both `-generate-passphrase` and `-peer-addr`s, only the cluster whose intercluster LIFs are specified in `-peer-addr`s 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-addr
192.140.112.103,192.140.112.104 -offer-expiration 2days -initial-allowed
-vserver-peers *
```

```
                Passphrase: UCa+6lRVICXeL/gq1WrK7ShR
                Expiration Time: 6/7/2017 08:16:10 EST
Initial Allowed Vserver Peers: vs1,vs2
                Intercluster LIF IP: 192.140.112.101,192.140.112.102
                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+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.

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

```
cluster peer create -peer-addr peer_LIF_IPs -ip-space ip-space
```

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-addr
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)	<a href="#">Prepare for mirroring and vaulting</a>
System Manager Classic (available with ONTAP 9.7 and earlier)	<a href="#">Volume disaster recovery preparation overview</a>

## 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
```

```
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`:

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

### 3. Verify the SVM peer relationship:

```
vserver peer show
```

```
cluster01::> vserver peer show
```

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

### 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<sub>pvs1</sub> and SVM<sub>vs1</sub> has been initiated:

```
cluster01::> vserver peer show-all
```

Vserver	Peer Vserver	Peer State	Peer Cluster	Peering Applications
pvs1	vs1	initiated	Cluster02	snapmirror

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:

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

Vserver	Peer Vserver	Peer State
vs1	pvs1	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
```

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

## 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:

```
cluster peer modify source_cluster -auth-status-admin use-authentication  
-encryption none
```

- To remove encryption and authentication, enter:

```
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:

```
cluster peer modify destination_cluster -auth-status-admin use-  
authentication -encrypt none
```

- To remove encryption and authentication, enter:

```
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.

## Data protection with the CLI

### Data protection overview with the CLI

You can use CLI commands to manage Snapshot copies on a local ONTAP system and to replicate Snapshot copies to a remote system using SnapMirror. You can replicate Snapshot copies for disaster recovery or long-term retention.

Use these procedures under the following circumstances:

- You want to understand the range of ONTAP backup and recovery capabilities.
- You want to use the command-line interface (CLI), not System Manager, an automated scripting tool, or [SnapCenter Software](#).
- You have already created peer relationships between the source and destination clusters and the source and destination SVMs.

#### [Cluster and SVM peering](#)

- You are backing up volumes or SVMs from AFF or FAS storage systems to AFF or FAS storage systems.
  - If you are replicating Element volumes to ONTAP, or ONTAP LUNs to an Element system, see the [NetApp Element software documentation](#).

#### [Replication between NetApp element software and ONTAP](#)

- Beginning with ONTAP 9.10.1, you can create data protection relationships between S3 buckets using S3 SnapMirror. For more information, see [S3 SnapMirror overview](#).
- You want to provide data protection using online methods, not tape.

### Other ways to do this in ONTAP

To perform these tasks with...	Refer to...
The redesigned System Manager (available with ONTAP 9.7 and later)	<a href="#">Prepare for mirroring and vaulting</a>
System Manager Classic (available with ONTAP 9.7 and earlier)	<a href="#">Volume disaster recovery preparation overview</a>

## Manage local Snapshot copies

### Manage local Snapshot copies overview

A *Snapshot copy* is a read-only, point-in-time image of a volume. The image consumes minimal storage space and incurs negligible performance overhead because it records only changes to files since the last Snapshot copy.

You can use a Snapshot copy to restore the entire contents of a volume, or to recover individual files or LUNs. Snapshot copies are stored in the directory `.snapshot` on the volume.

In ONTAP 9.3 and earlier, a volume can contain up to 255 Snapshot copies. In ONTAP 9.4 and later, a FlexVol volume can contain up to 1023 Snapshot copies.



Beginning with ONTAP 9.8, FlexGroup volumes can contain 1023 Snapshot copies. For more information, see [Protect FlexGroup volumes using Snapshot copies](#).

### Configure custom Snapshot policies

#### Configure custom Snapshot policies overview

A *Snapshot policy* defines how the system creates Snapshot copies. The policy specifies when to create Snapshot copies, how many copies to retain, and how to name them. For example, a system might create one Snapshot copy every day at 12:10 a.m., retain the two most recent copies, and name the copies “*daily.timestamp*.”

The default policy for a volume automatically creates Snapshot copies on the following schedule, with the oldest Snapshot copies deleted to make room for newer copies:

- A maximum of six hourly Snapshot copies taken five minutes past the hour.
- A maximum of two daily Snapshot copies taken Monday through Saturday at 10 minutes after midnight.
- A maximum of two weekly Snapshot copies taken every Sunday at 15 minutes after midnight.

Unless you specify a Snapshot policy when you create a volume, the volume inherits the Snapshot policy associated with its containing storage virtual machine (SVM).

#### When to configure a custom Snapshot policy

If the default Snapshot policy is not appropriate for a volume, you can configure a custom policy that modifies the frequency, retention, and name of Snapshot copies. The schedule will be dictated mainly by the rate of change of the active file system.

You might back up a heavily used file system like a database every hour, while you back up rarely used files once a day. Even for a database, you will typically run a full backup once or twice a day, while backing up transaction logs every hour.

Other factors are the importance of the files to your organization, your Service Level Agreement (SLA), your Recovery Point Objective (RPO), and your Recovery Time Objective (RTO). Generally speaking, you should retain only as many Snapshot copies as necessary.

## Create a Snapshot job schedule

A Snapshot policy requires at least one Snapshot copy job schedule. You can use the `job schedule cron create` command to create a job schedule.

### About this task

By default, ONTAP forms the names of Snapshot copies by appending a timestamp to the job schedule name.

If you specify values for both day of the month and day of the week, the values are considered independently. For example, a cron schedule with the day specification `Friday` and the day of the month specification `13` runs every Friday and on the 13th day of each month, not just on every Friday the 13th.

### Step

1. Create a job schedule:

```
job schedule cron create -name job_name -month month -dayofweek day_of_week  
-day day_of_month -hour hour -minute minute
```

For `-month`, `-dayofweek`, and `-hour`, you can specify `all` to run the job every month, day of the week, and hour, respectively.

Beginning with ONTAP 9.10.1, you can include the Vserver for your job schedule:

```
job schedule cron create -name job_name -vserver Vserver_name -month month  
-dayofweek day_of_week -day day_of_month -hour hour -minute minute
```

The following example creates a job schedule named `myweekly` that runs on Saturdays at 3:00 a.m.:

```
cluster1::> job schedule cron create -name myweekly -dayofweek  
"Saturday" -hour 3 -minute 0
```

The following example creates a schedule named `myweeklymulti` that specifies multiple days, hours and minutes:

```
job schedule cron create -name myweeklymulti -dayofweek  
"Monday,Wednesday,Sunday" -hour 3,9,12 -minute 0,20,50
```

## Create a Snapshot policy

A Snapshot policy specifies when to create Snapshot copies, how many copies to retain, and how to name them. For example, a system might create one Snapshot copy every day at 12:10 a.m., retain the two most recent copies, and name them “`daily.timestamp`.” A Snapshot policy can contain up to five job schedules.

### About this task

By default, ONTAP forms the names of Snapshot copies by appending a timestamp to the job schedule name:

```
daily.2017-05-14_0013/          hourly.2017-05-15_1106/
daily.2017-05-15_0012/          hourly.2017-05-15_1206/
hourly.2017-05-15_1006/         hourly.2017-05-15_1306/
```

You can substitute a prefix for the job schedule name if you prefer.

The `snapmirror-label` option is for SnapMirror replication. For more information, see [Defining a rule for a policy](#).

## Step

1. Create a Snapshot policy:

```
volume snapshot policy create -vserver SVM -policy policy_name -enabled
true|false -schedule1 schedule1_name -count1 copies_to_retain -prefix1
snapshot_prefix -snapmirror-label1 snapshot_label ... -schedule1 schedule5_name
-count5 copies_to_retain-prefix5 snapshot_prefix -snapmirror-label5
snapshot_label
```

The following example creates a Snapshot policy named `snap_policy_daily` that runs on a daily schedule. The policy has a maximum of five Snapshot copies, each with the name `daily.timestamp` and the SnapMirror label `daily`:

```
cluster1::> volume snapshot policy create -vserver vs0 -policy
snap_policy_daily -schedule1 daily -count1 5 -snapmirror-label1 daily
```

## Manage the Snapshot copy reserve

### Manage the Snapshot copy reserve overview

The *Snapshot copy reserve* sets aside a percentage of disk space for Snapshot copies, five percent by default. Because Snapshot copies use space in the active file system when the Snapshot copy reserve is exhausted, you might want to increase the Snapshot copy reserve as needed. Alternatively, you can autodelete Snapshot copies when the reserve is full.

### When to increase the Snapshot copy reserve

In deciding whether to increase the Snapshot reserve, it's important to remember that a Snapshot copy records only changes to files since the last Snapshot copy was made. It consumes disk space only when blocks in the active file system are modified or deleted.

This means that the rate of change of the file system is the key factor in determining the amount of disk space used by Snapshot copies. No matter how many Snapshot copies you create, they will not consume disk space if the active file system has not changed.

A FlexVol volume containing database transaction logs, for example, might have a Snapshot copy reserve as large as 20% to account for its greater rate of change. Not only will you want to create more Snapshot copies

to capture the more frequent updates to the database, you will also want to have a larger Snapshot copy reserve to handle the additional disk space the Snapshot copies consume.



A Snapshot copy consists of pointers to blocks rather than copies of blocks. You can think of a pointer as a “claim” on a block: ONTAP “holds” the block until the Snapshot copy is deleted.



*A Snapshot copy consumes disk space only when blocks in the active file system are modified or deleted.*

How deleting protected files can lead to less file space than expected

A Snapshot copy points to a block even after you delete the file that used the block. This explains why an exhausted Snapshot copy reserve might lead to the counter-intuitive result in which deleting an entire file system results in less space being available than the file system occupied.

Consider the following example. Before deleting any files, the `df` command output is as follows:

Filesystem	kbytes	used	avail	capacity
/vol/vol0/	3000000	3000000	0	100%
/vol/vol0/.snapshot	1000000	500000	500000	50%

After deleting the entire file system and making a Snapshot copy of the volume, the `df` command generates the following output:

Filesystem	kbytes	used	avail	capacity
/vol/vol0/	3000000	2500000	500000	83%
/vol/vol0/.snapshot	1000000	3500000	0	350%

As the output shows, the entire 3 GB formerly used by the active file system is now being used by Snapshot copies, in addition to the 0.5 GB used before the deletion.

Because the disk space used by the Snapshot copies now exceeds the Snapshot copy reserve, the overflow of 2.5 GB “spills” into the space reserved for active files, leaving you with 0.5 GB free space for files where you might reasonably have expected 3 GB.

#### Monitor Snapshot copy disk consumption

You can monitor Snapshot copy disk consumption using the `df` command. The command displays the amount of free space in the active file system and the Snapshot copy reserve.

#### Step

1. Display Snapshot copy disk consumption: `df`

The following example shows Snapshot copy disk consumption:

```
cluster1::> df
Filesystem      kbytes  used   avail  capacity
/vol/vol0/      3000000 3000000 0        100%
/vol/vol0/.snapshot 1000000 500000  500000   50%
```

#### Check available Snapshot copy reserve on a volume

You might want to check how much Snapshot copy reserve is available on a volume by using the `snapshot-reserve-available` parameter with the `volume show` command.

#### Step

1. Check the Snapshot copy reserve available on a volume:

```
vol show -vserver SVM -volume volume -fields snapshot-reserve-available
```

For complete command syntax, see the man page.

The following example displays the available Snapshot copy reserve for `vol1`:

```
cluster1::> vol show -vserver vs0 -volume vol1 -fields snapshot-reserve-
available

vserver volume snapshot-reserve-available
-----
vs0      vol1      4.84GB
```

### Modify the Snapshot copy reserve

You might want to configure a larger Snapshot copy reserve to prevent Snapshot copies from using space reserved for the active file system. You can decrease the Snapshot copy reserve when you no longer need as much space for Snapshot copies.

#### Step

1. Modify the Snapshot copy reserve:

```
volume modify -vserver SVM -volume volume -percent-snapshot-space snap_reserve
```

For complete command syntax, see the man page.

The following example sets the Snapshot copy reserve for `vol1` to 10 percent:

```
cluster1::> volume modify -vserver vs0 -volume vol1 -percent-snapshot
-space 10
```

### Autodelete Snapshot copies

You can use the `volume snapshot autodelete modify` command to trigger automatic deletion of Snapshot copies when the Snapshot reserve is exceeded. By default, the oldest Snapshot copies are deleted first.

#### About this task

LUN and file clones are deleted when there are no more Snapshot copies to be deleted.

#### Step

1. Autodelete Snapshot copies:

```
volume snapshot autodelete modify -vserver SVM -volume volume -enabled
true|false -trigger volume|snap_reserve
```

For complete command syntax, see the man page.

The following example autodeletes Snapshot copies for `vol1` when the Snapshot copy reserve is



exhausted:

```
cluster1::> volume snapshot autodelete modify -vserver vs0 -volume voll  
-enabled true -trigger snap_reserve
```

## Restore files from Snapshot copies

### Restore a file from a Snapshot copy on an NFS or SMB client

A user on an NFS or SMB client can restore a file directly from a Snapshot copy without the intervention of a storage system administrator.

Every directory in the file system contains a subdirectory named `.snapshot` accessible to NFS and SMB users. The `.snapshot` subdirectory contains subdirectories corresponding to the Snapshot copies of the volume:

```
$ ls .snapshot  
daily.2017-05-14_0013/          hourly.2017-05-15_1106/  
daily.2017-05-15_0012/          hourly.2017-05-15_1206/  
hourly.2017-05-15_1006/         hourly.2017-05-15_1306/
```

Each subdirectory contains the files referenced by the Snapshot copy. If users accidentally delete or overwrite a file, they can restore the file to the parent read-write directory by copying the file from the Snapshot subdirectory to the read-write directory:

```
$ ls my.txt  
ls: my.txt: No such file or directory  
$ ls .snapshot  
daily.2017-05-14_0013/          hourly.2017-05-15_1106/  
daily.2017-05-15_0012/          hourly.2017-05-15_1206/  
hourly.2017-05-15_1006/         hourly.2017-05-15_1306/  
$ ls .snapshot/hourly.2017-05-15_1306/my.txt  
my.txt  
$ cp .snapshot/hourly.2017-05-15_1306/my.txt .  
$ ls my.txt  
my.txt
```

### Enable and disable NFS and SMB client access to Snapshot copy directory

To determine whether the Snapshot copy directory is visible to NFS and SMB clients to restore a file or LUN from a Snapshot copy, you can enable and disable access to the Snapshot copy directory using the `-snapdir-access` option of the `volume modify` command.

### Steps

### 1. Check the Snapshot directory access status:

```
volume show -vserver SVM_name -volume vol_name -fields snapdir-access
```

Example:

```
clus1::> volume show -vserver vs0 -volume vol1 -fields snapdir-access
vserver volume snapdir-access
-----
vs0      vol1    false
```

### 2. Enable or disable the Snapshot copy directory access:

```
volume modify -vserver SVM_name -volume vol_name -snapdir-access true|false
```

The following example enables Snapshot copy directory access on vol1:

```
clus1::> volume modify -vserver vs0 -volume vol1 -snapdir-access true
Volume modify successful on volume vol1 of Vserver vs0.
```

## Restore a single file from a Snapshot copy

You can use the `volume snapshot restore-file` command to restore a single file or LUN from a Snapshot copy. You can restore the file to a different location in the parent read-write volume if you do not want to replace an existing file.

### About this task

If you are restoring an existing LUN, a LUN clone is created and backed up in the form of a Snapshot copy. During the restore operation, you can read to and write from the LUN.

Files with streams are restored by default.

### Steps

#### 1. List the Snapshot copies in a volume:

```
volume snapshot show -vserver SVM -volume volume
```

For complete command syntax, see the man page.

The following example shows the Snapshot copies in vol1:

```
clus1::> volume snapshot show -vserver vs1 -volume vol1
```

Vserver	Volume	Snapshot	State	Size	Total%	Used%
vs1	vol1	hourly.2013-01-25_0005	valid	224KB	0%	0%
		daily.2013-01-25_0010	valid	92KB	0%	0%
		hourly.2013-01-25_0105	valid	228KB	0%	0%
		hourly.2013-01-25_0205	valid	236KB	0%	0%
		hourly.2013-01-25_0305	valid	244KB	0%	0%
		hourly.2013-01-25_0405	valid	244KB	0%	0%
		hourly.2013-01-25_0505	valid	244KB	0%	0%

7 entries were displayed.

## 2. Restore a file from a Snapshot copy:

```
volume snapshot restore-file -vserver SVM -volume volume -snapshot snapshot  
-path file_path -restore-path destination_path
```

For complete command syntax, see the man page.

The following example restores the file `myfile.txt`:

```
cluster1::> volume snapshot restore-file -vserver vs0 -volume vol1  
-snapshot daily.2013-01-25_0010 -path /myfile.txt
```

### Restore part of a file from a Snapshot copy

You can use the `volume snapshot partial-restore-file` command to restore a range of data from a Snapshot copy to a LUN or to an NFS or SMB container file, assuming you know the starting byte offset of the data and the byte count. You might use this command to restore one of the databases on a host that stores multiple databases in the same LUN.

Beginning in ONTAP 9.12.1, partial restore is available for volumes in an SM-BC relationship.

### Steps

1. List the Snapshot copies in a volume:

```
volume snapshot show -vserver SVM -volume volume
```

For complete command syntax, see the man page.

The following example shows the Snapshot copies in `vol1`:

```
clus1::> volume snapshot show -vserver vs1 -volume vol1
```

Vserver	Volume	Snapshot	State	Size	Total%	Used%
vs1	vol1	hourly.2013-01-25_0005	valid	224KB	0%	0%
		daily.2013-01-25_0010	valid	92KB	0%	0%
		hourly.2013-01-25_0105	valid	228KB	0%	0%
		hourly.2013-01-25_0205	valid	236KB	0%	0%
		hourly.2013-01-25_0305	valid	244KB	0%	0%
		hourly.2013-01-25_0405	valid	244KB	0%	0%
		hourly.2013-01-25_0505	valid	244KB	0%	0%

7 entries were displayed.

## 2. Restore part of a file from a Snapshot copy:

```
volume snapshot partial-restore-file -vserver SVM -volume volume -snapshot  
snapshot -path file_path -start-byte starting_byte -byte-count byte_count
```

The starting byte offset and byte count must be multiples of 4,096.

The following example restores the first 4,096 bytes of the file `myfile.txt`:

```
cluster1::> volume snapshot partial-restore-file -vserver vs0 -volume  
vol1 -snapshot daily.2013-01-25_0010 -path /myfile.txt -start-byte 0  
-byte-count 4096
```

### Restore the contents of a volume from a Snapshot copy

You can use the `volume snapshot restore` command to restore the contents of a volume from a Snapshot copy.

#### About this task

If the volume has SnapMirror relationships, manually replicate all mirror copies of the volume immediately after you restore from a Snapshot copy. Not doing so can result in unusable mirror copies that must be deleted and recreated.

## 1. List the Snapshot copies in a volume:

```
volume snapshot show -vserver SVM -volume volume
```

The following example shows the Snapshot copies in `vol1`:

```
clus1::> volume snapshot show -vserver vs1 -volume vol1
```

Vserver	Volume	Snapshot	State	Size	Total%	Used%
vs1	vol1	hourly.2013-01-25_0005	valid	224KB	0%	0%
		daily.2013-01-25_0010	valid	92KB	0%	0%
		hourly.2013-01-25_0105	valid	228KB	0%	0%
		hourly.2013-01-25_0205	valid	236KB	0%	0%
		hourly.2013-01-25_0305	valid	244KB	0%	0%
		hourly.2013-01-25_0405	valid	244KB	0%	0%
		hourly.2013-01-25_0505	valid	244KB	0%	0%

7 entries were displayed.

## 2. Restore the contents of a volume from a Snapshot copy:

```
volume snapshot restore -vserver SVM -volume volume -snapshot snapshot
```

The following example restores the contents of vol1:

```
cluster1::> volume snapshot restore -vserver vs0 -volume vol1 -snapshot  
daily.2013-01-25_0010
```

## SnapMirror volume replication

### About SnapMirror volume replication

Traditionally, ONTAP replication technologies served the need for disaster recovery (DR) and data archiving. In ONTAP 9.3, these technologies were combined in a way that allows you to configure disaster recovery and archiving on the same destination volume.

### Asynchronous SnapMirror disaster recovery basics

*SnapMirror* is disaster recovery technology, designed for failover from primary storage to secondary storage at a geographically remote site. As its name implies, SnapMirror creates a replica, or *mirror*, of your working data in secondary storage from which you can continue to serve data in the event of a catastrophe at the primary site.

If the primary site is still available to serve data, you can simply transfer any needed data back to it, and not serve clients from the mirror at all. As the failover use case implies, the controllers on the secondary system should be equivalent or nearly equivalent to the controllers on the primary system to serve data efficiently from mirrored storage.

### Data protection relationships

Data is mirrored at the volume level. The relationship between the source volume in primary storage and the

destination volume in secondary storage is called a *data protection relationship*. The clusters in which the volumes reside and the SVMs that serve data from the volumes must be *peered*. A peer relationship enables clusters and SVMs to exchange data securely.

### Cluster and SVM peering

The figure below illustrates SnapMirror data protection relationships.



### Scope of data protection relationships

You can create a data protection relationship directly between volumes or between the SVMs that own the volumes. In an *SVM data protection relationship*, all or part of the SVM configuration, from NFS exports and SMB shares to RBAC, is replicated, as well as the data in the volumes that the SVM owns.

You can also use SnapMirror for two special data protection applications:

- A *load-sharing mirror* copy of the SVM root volume ensures that data remains accessible in the event of a node outage or failover.
- A data protection relationship between *SnapLock volumes* lets you replicate WORM files to secondary storage.

### Archive and compliance using SnapLock technology

### How SnapMirror data protection relationships are initialized

The first time you invoke SnapMirror, it performs a *baseline transfer* from the source volume to the destination volume. The *SnapMirror policy* for the relationship defines the contents of the baseline and any updates.

A baseline transfer under the default SnapMirror policy `MirrorAllSnapshots` involves the following steps:

- Make a Snapshot copy of the source volume.
- Transfer the Snapshot copy and all the data blocks it references to the destination volume.

- Transfer the remaining, less recent Snapshot copies on the source volume to the destination volume for use in case the “active” mirror is corrupted.

#### How SnapMirror data protection relationships are updated

Updates are asynchronous, following the schedule you configure. Retention mirrors the Snapshot policy on the source.

At each update under the `MirrorAllSnapshots` policy, SnapMirror creates a Snapshot copy of the source volume and transfers that Snapshot copy and any Snapshot copies that have been made since the last update. In the following output from the `snapmirror policy show` command for the `MirrorAllSnapshots` policy, note the following:

- `Create Snapshot` is “true”, indicating that `MirrorAllSnapshots` creates a Snapshot copy when SnapMirror updates the relationship.
- `MirrorAllSnapshots` has rules “`sm_created`” and “`all_source_snapshots`”, indicating that both the Snapshot copy created by SnapMirror and any Snapshot copies that have been made since the last update are transferred when SnapMirror updates the relationship.

```
cluster_dst::> snapmirror policy show -policy MirrorAllSnapshots -instance

                Vserver: vs0
    SnapMirror Policy Name: MirrorAllSnapshots
    SnapMirror Policy Type: async-mirror
            Policy Owner: cluster-admin
                Tries Limit: 8
            Transfer Priority: normal
    Ignore accesstime Enabled: false
        Transfer Restartability: always
    Network Compression Enabled: false
            Create Snapshot: true
                Comment: Asynchronous SnapMirror policy for mirroring
all snapshots
                        and the latest active file system.
    Total Number of Rules: 2
            Total Keep: 2
                Rules: SnapMirror Label      Keep  Preserve Warn
Schedule Prefix
-----
-----
                        sm_created            1  false      0  -
-
                        all_source_snapshots  1  false      0  -
-
```

## MirrorLatest policy

The preconfigured `MirrorLatest` policy works exactly the same way as `MirrorAllSnapshots`, except that only the Snapshot copy created by SnapMirror is transferred at initialization and update.

```
Rules: SnapMirror Label      Keep  Preserve Warn
Schedule Prefix
-----
sm_created                1  false      0 -
```

## SnapMirror Synchronous disaster recovery basics

Beginning with ONTAP 9.5, SnapMirror Synchronous (SM-S) technology is supported on all FAS and AFF platforms that have at least 16 GB of memory and on all ONTAP Select platforms. SnapMirror Synchronous technology is a per-node, licensed feature that provides synchronous data replication at the volume level.

This functionality addresses the regulatory and national mandates for synchronous replication in financial, healthcare, and other regulated industries where zero data loss is required.

The limit on the number of SnapMirror Synchronous replication operations per HA pair depends on the controller model.

The following table lists the number of SnapMirror Synchronous operations that are allowed per HA pair according to platform type and ONTAP release.

Platform	Releases earlier than ONTAP 9.9.1	ONTAP 9.9.1	ONTAP 9.10.1	ONTAP 9.11.1\ONTAP 9.12.1
AFF	80	160	200	400
FAS	40	80	80	80
ONTAP Select	20	40	40	40

## Supported features

ONTAP 9.12.1 supports non-disruptive SnapMirror Synchronous operations (NDO) on AFF/ASA platforms, only. Support for non-disruptive operations enables you to perform many common maintenance tasks without scheduling down time. Operations supported include takeover and giveback, and volume move, provided that a single node is surviving among each of the two clusters.

The following features are supported for SnapMirror Synchronous technology in ONTAP 9.10.1; provided all nodes in the source and destination cluster are running ONTAP 9.10.1:

- NFSv4.2



- NVMe/TCP

In ONTAP 9.5 and later, SnapMirror Synchronous technology supports the NFSv3, FC, and iSCSI protocols over all networks for which the latency does not exceed 10ms.

SnapMirror Synchronous supports source and destination volumes on FabricPool aggregates with a tiering policy of None, Snapshot, or Auto. The destination volume in a FabricPool aggregate cannot be set to All tiering policy.

The following features are supported for SnapMirror Synchronous technology in ONTAP 9.7:

- Replication of application-created Snapshot copies  
If a Snapshot copy is tagged with the appropriate label at the time of the `snapshot create` operation, using the CLI or the ONTAP API, SnapMirror Synchronous replicates the Snapshot copies, both user created or those created with external scripts, after quiescing the applications. Scheduled Snapshot copies created using a Snapshot policy are not replicated. For more information about replicating application-created Snapshot copies, see the Knowledge Base article: [How to replicate application created snapshots with SnapMirror Synchronous](#).
- FC-NVMe
- LUN clones and NVMe namespace clones  
LUN clones backed by application-created Snapshot copies are also supported.

The following features are supported for SnapMirror Synchronous technology in ONTAP 9.6; provided all nodes in the source and destination cluster are running ONTAP 9.6:

- SVM DR
  - A SnapMirror Synchronous source can also be a SVM DR source, for example, a fan-out configuration with SM-S as one leg and SVM DR as the other.
  - A SnapMirror Synchronous source cannot be an SVM DR destination because SM-S does not support cascading a DP source.  
You must release the synchronous relationship before performing an SVM DR flip resync in the destination cluster.
  - A SnapMirror Synchronous destination cannot be an SVM DR source because SVM DR does not support replication of DP volumes.  
A flip resync of the synchronous source would result in the SVM DR excluding the DP volume in the destination cluster.
- NFSv4.0 and NFSv4.1
- SMB 2.0 or later
- Mixed protocol access (NFSv3 and SMB)
- Antivirus on the primary volume of the SnapMirror Synchronous relationship
- Hard or soft quotas on the primary volume of the SnapMirror Synchronous relationship  
The quota rules are not replicated to the destination; therefore, the quota database is not replicated to the destination.
- FPolicy on the primary volume of the SnapMirror Synchronous relationship
- SnapMirror Synchronous mirror-mirror cascade  
The relationship from the destination volume of the SnapMirror Synchronous relationship must be an asynchronous SnapMirror relationship.
- Timestamp parity between source and destination volumes for NAS

If you have upgraded from ONTAP 9.5 to ONTAP 9.6, the timestamp is replicated only for any new and modified files in the source volume. The timestamp of existing files in the source volume is not synchronized.

- Removal of high metadata operation frequency limitation
- Security for sensitive data in-transit using TLS 1.2 encryption
- Clone autodelete

### Unsupported features

The following features are not supported with Synchronous SnapMirror relationships:

- Tamperproof Snapshot copies
- Consistency groups
- MetroCluster configurations
- SFMoD
- SFCoD
- VVol
- Mixed SAN and NVMe access  
LUNs and NVMe namespaces are not supported on the same volume or SVM.
- SnapLock volumes
- FlexGroup volumes
- FlexCache volumes
- SnapRestore
- DP\_Optimized (DPO) systems
- Tape backup or restore using dump and SMTape on the destination volume
- Tape based restore to the source volume
- Throughput floor (QoS Min) for source volumes
- In a fan-out configuration, only one relationship can be a SnapMirror Synchronous relationship; all the other relationships from the source volume must be asynchronous SnapMirror relationships.
- Global throttling

### Modes of operation

SnapMirror Synchronous has two modes of operation based on the type of the SnapMirror policy used:

- **Sync mode**

In Sync mode, application I/O operations are sent in parallel to the primary and secondary storage systems. If the write to the secondary storage is not completed for any reason, the application is allowed to continue writing to the primary storage. When the error condition is corrected, SnapMirror Synchronous technology automatically resynchronizes with the secondary storage and resumes replicating from primary storage to secondary storage in Synchronous mode.

In Sync mode, RPO=0 and RTO is very low until a secondary replication failure occurs at which time RPO and RTO become indeterminate, but equal the time to repair the issue that caused secondary replication to fail and for the resync to complete.

- **StrictSync mode**

SnapMirror Synchronous can optionally operate in StrictSync mode. If the write to the secondary storage is not completed for any reason, the application I/O fails, thereby ensuring that the primary and secondary storage are identical. Application I/O to the primary resumes only after the SnapMirror relationship returns to the InSync status. If the primary storage fails, application I/O can be resumed on the secondary storage, after failover, with no loss of data.

In StrictSync mode RPO is always zero, and RTO is very low.

### Relationship status

The status of a SnapMirror Synchronous relationship is always in the InSync status during normal operation. If the SnapMirror transfer fails for any reason, the destination is not in sync with the source and can go to the OutofSync status.

For SnapMirror Synchronous relationships, the system automatically checks the relationship status (InSync or OutofSync) at a fixed interval. If the relationship status is OutofSync, ONTAP automatically triggers the auto resync process to bring back the relationship to the InSync status. Auto resync is triggered only if the transfer fails due to any operation, such as unplanned storage failover at source or destination or a network outage. User-initiated operations such as `snapmirror quiesce` and `snapmirror break` do not trigger auto resync.

If the relationship status becomes OutofSync for a SnapMirror Synchronous relationship in the StrictSync mode, all I/O operations to the primary volume are stopped. The OutofSync state for SnapMirror Synchronous relationship in the Sync mode is not disruptive to the primary and I/O operations are allowed on the primary volume.

### Related information

[NetApp Technical Report 4733: SnapMirror Synchronous configuration and best practices](#)

### About workloads supported by StrictSync and Sync policies

StrictSync and Sync policies support all LUN-based applications with FC, iSCSI, and FC-NVMe protocols, as well as NFSv3 and NFSv4 protocols for enterprise applications such as databases, VMWare, quota, SMB, and so on. Beginning with ONTAP 9.6, SnapMirror Synchronous can be used for enterprise file services such as electronic design automation (EDA), home directories, and software build workloads.

In ONTAP 9.5, for a Sync policy, you need to consider a few important aspects while selecting the NFSv3 or NFSv4 workloads. The amount of data read or write operations by workloads is not a consideration, as Sync policy can handle high read or write IO workloads. In ONTAP 9.5, workloads that have excessive file creation, directory creation, file permission changes, or directory permission changes may not be suitable (these are referred to as high-metadata workloads). A typical example of a high-metadata workload is a DevOps workload in which you create multiple test files, run automation, and delete the files. Another example is parallel build workload that generate multiple temporary files during compilation. The impact of a high rate of write metadata activity is that it can cause synchronization between mirrors to temporarily break which stalls the read and write IOs from the client.

Beginning with ONTAP 9.6, these limitations are removed and SnapMirror Synchronous can be used for enterprise file services workloads that include multiuser environments, such as home directories and software build workloads.

### Related information

[SnapMirror Synchronous Configuration and Best Practices](#)

## Vault archiving using SnapMirror technology

SnapMirror vault policies replace SnapVault technology in ONTAP 9.3 and later. You use a SnapMirror vault policy for disk-to-disk Snapshot copy replication for standards compliance and other governance-related purposes. In contrast to a SnapMirror relationship, in which the destination usually contains only the Snapshot copies currently in the source volume, a vault destination typically retains point-in-time Snapshot copies created over a much longer period.

You might want to keep monthly Snapshot copies of your data over a 20-year span, for example, to comply with government accounting regulations for your business. Since there is no requirement to serve data from vault storage, you can use slower, less expensive disks on the destination system.

The figure below illustrates SnapMirror vault data protection relationships.



*A SnapVault data protection relationship typically retains point-in-time Snapshot copies created over a longer period than the Snapshot copies on the source volume.*

### How vault data protection relationships are initialized

The SnapMirror policy for the relationship defines the contents of the baseline and any updates.

A baseline transfer under the default vault policy `XDPDefault` makes a Snapshot copy of the source volume, then transfers that copy and the data blocks it references to the destination volume. Unlike SnapMirror relationships, a vault backup does not include older Snapshot copies in the baseline.

### How vault data protection relationships are updated

Updates are asynchronous, following the schedule you configure. The rules you define in the policy for the relationship identify which new Snapshot copies to include in updates and how many copies to retain. The labels defined in the policy ("monthly," for example) must match one or more labels defined in the Snapshot policy on the source. Otherwise, replication fails.

At each update under the `XDPDefault` policy, SnapMirror transfers Snapshot copies that have been made since the last update, provided they have labels matching the labels defined in the policy rules. In the following

output from the `snapmirror policy show` command for the `XDPDefault` policy, note the following:

- `Create Snapshot` is “false”, indicating that `XDPDefault` does not create a Snapshot copy when `SnapMirror` updates the relationship.
- `XDPDefault` has rules “daily” and “weekly”, indicating that all Snapshot copies with matching labels on the source are transferred when `SnapMirror` updates the relationship.

```
cluster_dst:> snapmirror policy show -policy XDPDefault -instance

                Vserver: vs0
SnapMirror Policy Name: XDPDefault
SnapMirror Policy Type: vault
                Policy Owner: cluster-admin
                Tries Limit: 8
                Transfer Priority: normal
Ignore accesstime Enabled: false
                Transfer Restartability: always
Network Compression Enabled: false
                Create Snapshot: false
                Comment: Default policy for XDP relationships with
daily and weekly
                        rules.
                Total Number of Rules: 2
                Total Keep: 59
                Rules: SnapMirror Label      Keep  Preserve Warn
Schedule Prefix
-----
-----
                        daily                7   false    0  -
-
                        weekly               52   false    0  -
-
```

**SnapMirror unified replication basics**

*SnapMirror unified replication* allows you to configure disaster recovery and archiving on the same destination volume. When unified replication is appropriate, it offers benefits in reducing the amount of secondary storage you need, limiting the number of baseline transfers, and decreasing network traffic.

**How unified data protection relationships are initialized**

As with `SnapMirror`, unified data protection performs a baseline transfer the first time you invoke it. The `SnapMirror` policy for the relationship defines the contents of the baseline and any updates.

A baseline transfer under the default unified data protection policy `MirrorAndVault` makes a Snapshot copy of the source volume, then transfers that copy and the data blocks it references to the destination volume. Like

vault archiving, unified data protection does not include older Snapshot copies in the baseline.

**How unified data protection relationships are updated**

At each update under the `MirrorAndVault` policy, `SnapMirror` creates a Snapshot copy of the source volume and transfers that Snapshot copy and any Snapshot copies that have been made since the last update, provided they have labels matching the labels defined in the Snapshot policy rules. In the following output from the `snapmirror policy show` command for the `MirrorAndVault` policy, note the following:

- `Create Snapshot` is “true”, indicating that `MirrorAndVault` creates a Snapshot copy when `SnapMirror` updates the relationship.
- `MirrorAndVault` has rules “sm\_created”, “daily”, and “weekly”, indicating that both the Snapshot copy created by `SnapMirror` and the Snapshot copies with matching labels on the source are transferred when `SnapMirror` updates the relationship.

```
cluster_dst:> snapmirror policy show -policy MirrorAndVault -instance

                Vserver: vs0
    SnapMirror Policy Name: MirrorAndVault
    SnapMirror Policy Type: mirror-vault
            Policy Owner: cluster-admin
            Tries Limit: 8
        Transfer Priority: normal
    Ignore accesstime Enabled: false
    Transfer Restartability: always
    Network Compression Enabled: false
            Create Snapshot: true
            Comment: A unified Synchronous SnapMirror and
SnapVault policy for
                                mirroring the latest file system and daily
and weekly snapshots.
        Total Number of Rules: 3
            Total Keep: 59
                Rules: SnapMirror Label      Keep  Preserve Warn
Schedule Prefix
-----
-----
                                sm_created      1  false      0 -
-
                                daily              7  false      0 -
-
                                weekly             52  false      0 -
-
```

**Unified7year policy**

The preconfigured `Unified7year` policy works exactly the same way as `MirrorAndVault`, except that a

fourth rule transfers monthly Snapshot copies and retains them for seven years.

Rules: SnapMirror Label		Keep	Preserve	Warn
Schedule	Prefix			
-----	-----	----	-----	----
	sm_created	1	false	0 -
-	daily	7	false	0 -
-	weekly	52	false	0 -
-	monthly	84	false	0 -
-				

**Protect against possible data corruption**

Unified replication limits the contents of the baseline transfer to the Snapshot copy created by SnapMirror at initialization. At each update, SnapMirror creates another Snapshot copy of the source and transfers that Snapshot copy and any new Snapshot copies that have labels matching the labels defined in the Snapshot policy rules.

You can protect against the possibility that an updated Snapshot copy is corrupted by creating a copy of the last transferred Snapshot copy on the destination. This “local copy” is retained regardless of the retention rules on the source, so that even if the Snapshot originally transferred by SnapMirror is no longer available on the source, a copy of it will be available on the destination.

**When to use unified data replication**

You need to weigh the benefit of maintaining a full mirror against the advantages that unified replication offers in reducing the amount of secondary storage, limiting the number of baseline transfers, and decreasing network traffic.

The key factor in determining the appropriateness of unified replication is the rate of change of the active file system. A traditional mirror might be better suited to a volume holding hourly Snapshot copies of database transaction logs, for example.

**XDP replaces DP as the SnapMirror default**

Beginning with ONTAP 9.3, SnapMirror extended data protection (XDP) mode replaces SnapMirror data protection (DP) mode as the SnapMirror default.

Before upgrading to ONTAP 9.12.1, you must convert existing DP-type relationships to XDP before you can upgrade to ONTAP 9.12.1 and later releases. For more information, see [Convert an existing DP-type relationship to XDP](#).

Until ONTAP 9.3, SnapMirror invoked in DP mode and SnapMirror invoked in XDP mode used different replication engines, with different approaches to version-dependence:

- SnapMirror invoked in DP mode used a *version-dependent* replication engine in which the ONTAP version was required to be the same on primary and secondary storage:

```
cluster_dst:> snapmirror create -type DP -source-path ... -destination
-path ...
```

- SnapMirror invoked in XDP mode used a *version-flexible* replication engine that supported different ONTAP versions on primary and secondary storage:

```
cluster_dst:> snapmirror create -type XDP -source-path ...
-destination-path ...
```

With improvements in performance, the significant benefits of version-flexible SnapMirror outweigh the slight advantage in replication throughput obtained with version-dependent mode. For this reason, beginning with ONTAP 9.3, XDP mode has been made the new default, and any invocations of DP mode on the command line or in new or existing scripts are automatically converted to XDP mode.

Existing relationships are not affected. If a relationship is already of type DP, it will continue to be of type DP. Beginning with ONTAP 9.5, MirrorAndVault is the new default policy when no data protection mode is specified or when XDP mode is specified as the relationship type. The table below shows the behavior you can expect.

If you specify...	The type is...	The default policy (if you do not specify a policy) is...
DP	XDP	MirrorAllSnapshots (SnapMirror DR)
Nothing	XDP	MirrorAndVault (unified replication)
XDP	XDP	MirrorAndVault (unified replication)

As the table shows, the default policies assigned to XDP in different circumstances ensure that the conversion maintains the functional equivalence of the old types. Of course, you can use different policies as needed, including policies for unified replication:

If you specify...	And the policy is...	The result is...
DP	MirrorAllSnapshots	SnapMirror DR
	XDPDefault	SnapVault
	MirrorAndVault	Unified replication
XDP	MirrorAllSnapshots	SnapMirror DR
	XDPDefault	SnapVault
	MirrorAndVault	Unified replication



The only exceptions to conversion are as follows:

- SVM data protection relationships continue to default to DP mode in ONTAP 9.3 and earlier.

Beginning with ONTAP 9.4, SVM data protection relationships default to XDP mode.

- Root volume load-sharing data protection relationships continue to default to DP mode.
- SnapLock data protection relationships continue to default to DP mode in ONTAP 9.4 and earlier.

Beginning with ONTAP 9.5, SnapLock data protection relationships default to XDP mode.

- Explicit invocations of DP continue to default to DP mode if you set the following cluster-wide option:

```
options replication.create_data_protection_rels.enable on
```

This option is ignored if you do not explicitly invoke DP.

### When a destination volume grows automatically

During a data protection mirror transfer, the destination volume grows automatically in size if the source volume has grown, provided there is available space in the aggregate that contains the volume.

This behavior occurs irrespective of any automatic growth setting on the destination. You cannot limit the volume's growth or prevent ONTAP from growing it.

By default, data protection volumes are set to the `grow_shrink` autosize mode, which enables the volume to grow or shrink in response to the amount of used space. The max-autosize for data protection volumes is equal to the maximum FlexVol size and is platform dependent. For example:

- FAS2220, default DP volume max-autosize = 60TB
- FAS6220, default DP volume max-autosize = 70TB
- FAS8200, default DP volume max-autosize = 100TB

For more information, see [NetApp Hardware Universe](#).

### Fan-out and cascade data protection deployments

You can use a *fan-out* deployment to extend data protection to multiple secondary systems. You can use a *cascade* deployment to extend data protection to tertiary systems.

Both fan-out and cascade deployments support any combination of SnapMirror DR, SnapVault, or unified replication; however, SnapMirror Synchronous relationships (supported beginning with ONTAP 9.5) support only fan-out deployments with one or more asynchronous SnapMirror relationships and do not support cascade deployments. Only one relationship in the fan-out configuration can be a SnapMirror Synchronous relationship, all the other relationships from the source volume must be asynchronous SnapMirror relationships. [SnapMirror Business Continuity](#) (supported beginning with ONTAP 9.8) also supports fan-out configurations.



You can use a *fan-in* deployment to create data protection relationships between multiple primary systems and a single secondary system. Each relationship must use a different volume on the secondary system.



You should be aware that volumes that are part of a fan-out or cascade configuration can take longer to resynchronize. It is not uncommon to see the SnapMirror relationship reporting the status "preparing" for an extended time period.

### How fan-out deployments work

SnapMirror supports *multiple-mirrors* and *mirror-vault* fan-out deployments.

A multiple-mirrors fan-out deployment consists of a source volume that has a mirror relationship to multiple secondary volumes.



A mirror-vault fan-out deployment consists of a source volume that has a mirror relationship to a secondary volume and a SnapVault relationship to a different secondary volume.



Beginning with ONTAP 9.5, you can have fan-out deployments with SnapMirror Synchronous relationships;

however, only one relationship in the fan-out configuration can be a SnapMirror Synchronous relationship, all the other relationships from the source volume must be asynchronous SnapMirror relationships.



#### How cascade deployments work

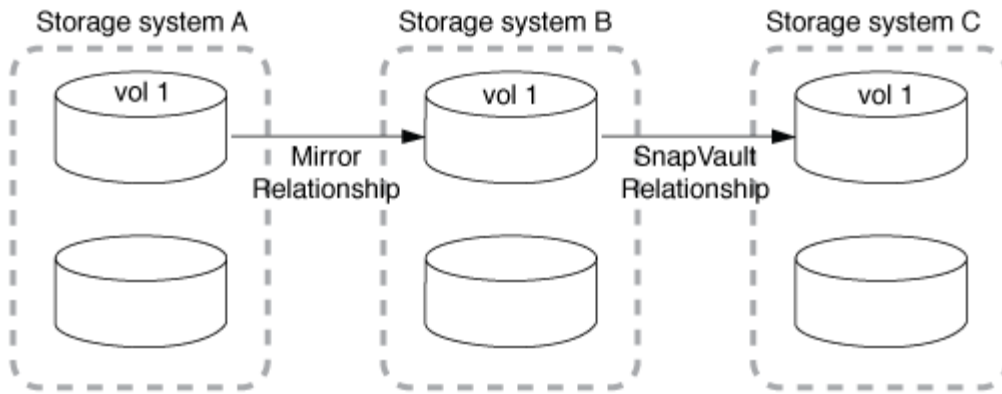
SnapMirror supports *mirror-mirror*, *mirror-vault*, *vault-mirror*, and *vault-vault* cascade deployments.

A mirror-mirror cascade deployment consists of a chain of relationships in which a source volume is mirrored to a secondary volume, and the secondary volume is mirrored to a tertiary volume. If the secondary volume becomes unavailable, you can synchronize the relationship between the primary and tertiary volumes without performing a new baseline transfer.

Beginning with ONTAP 9.6, SnapMirror Synchronous relationships are supported in a mirror-mirror cascade deployment. Only the primary and secondary volumes can be in a SnapMirror Synchronous relationship. The relationship between the secondary volumes and tertiary volumes must be asynchronous.



A mirror-vault cascade deployment consists of a chain of relationships in which a source volume is mirrored to a secondary volume, and the secondary volume is vaulted to a tertiary volume.



Vault-mirror and, beginning with ONTAP 9.2, vault-vault cascade deployments are also supported:

- A vault-mirror cascade deployment consists of a chain of relationships in which a source volume is vaulted to a secondary volume, and the secondary volume is mirrored to a tertiary volume.
- (Beginning with ONTAP 9.2) A vault-vault cascade deployment consists of a chain of relationships in which a source volume is vaulted to a secondary volume, and the secondary volume is vaulted to a tertiary volume.

### Further Reading

- [Resume protection in a fan-out configuration with SM-BC](#)

## SnapMirror licensing

### SnapMirror licensing overview

Beginning with ONTAP 9.3, licensing has been simplified for replicating between ONTAP instances. In ONTAP 9 releases, the SnapMirror license supports both vault and mirror relationships. Users can now purchase a SnapMirror license to support ONTAP replication for both backup and disaster recovery use cases.

Prior to the ONTAP 9.3 release, two licenses were available to support different replication use cases. A SnapVault license was needed to configure *vault* relationships between ONTAP instances, where the DP instance could retain a higher number of Snapshot copies to support backup use cases where retention times are longer. A SnapMirror license was needed to configure *mirror* relationships between ONTAP instances, where each ONTAP instance would maintain the same number of snapshot copies (that is, a *mirror* image) to support disaster recovery use cases where cluster failovers would be possible. Both SnapMirror and SnapVault licenses can continue to be used and supported for ONTAP 8.x and 9.x releases.

SnapVault licenses continue to function and are supported for both ONTAP 8.x and 9.x releases, but they are no longer being sold. The SnapMirror license continues to be available and can be used in place of SnapVault and can be used for both mirror and vault configurations.

For ONTAP asynchronous replication, beginning with ONTAP 9.3 a single unified replication engine is used to configure extended data protection mode (XDP) policies, where the SnapMirror license can be configured for a mirror policy, a vault policy, or a mirror-vault policy. A SnapMirror license is required on both the source and destination clusters. A SnapVault license is not required if a SnapMirror license is already installed. The SnapMirror asynchronous perpetual license is included in the Data Protection bundle which you can purchase for your ONTAP clusters. The Data Protection bundle price is based on the raw capacity of the cluster.

Data protection configuration limits are determined using several factors, including your ONTAP version,

hardware platform, and the licenses installed. For more information, see [Hardware Universe](#).

## SnapMirror Synchronous license

Beginning with ONTAP 9.5, SnapMirror Synchronous relationships are supported. You require the following licenses for creating a SnapMirror Synchronous relationship:

- The SnapMirror Synchronous license is required on both the source cluster and the destination cluster.

The SnapMirror Synchronous license is enabled with either the Premium bundle or the Data Protection bundle.

If your system was purchased before June 2019 with a Premium or Flash Bundle, you can download a NetApp master key to get the required SnapMirror Synchronous license from the NetApp Support Site: [Master License Keys](#)

- The SnapMirror license is required on both the source cluster and the destination cluster.

## SnapMirror Cloud license

Beginning with ONTAP 9.8, the SnapMirror Cloud license provides asynchronous replication of Snapshot copies from ONTAP instances to object storage endpoints. Replication targets can be configured using both on-premises object stores as well as S3 and S3-compatible public cloud object storage services. SnapMirror Cloud relationships are supported from ONTAP systems to pre-qualified object storage targets. ONTAP 9.8 approved object storage targets include ONTAP S3, StorageGRID, AWS S3 Standard, S3 Standard-IA, and S3 One Zone-IA, Microsoft Azure Blob Premium, Hot and Cool, and GCP Standard and Nearline storage.

SnapMirror Cloud is not available as a standalone license and is available only with purchase of the Hybrid Cloud Bundle. Hybrid Cloud Bundle is a term-based subscription license that is priced based on capacity. Only one license is needed per ONTAP cluster. Capacity is defined as the “used” capacity (not raw capacity) within any volume which is protected by SnapMirror Cloud. Users will purchase this license based on the total used capacity of volumes on the cluster being backed up by SnapMirror Cloud. As of October 2021, the Hybrid Cloud Bundle includes only a SnapMirror Cloud license (previously Hybrid Cloud Bundle included a FabricPool license, which was removed from the bundle effective October 2021). In addition to SnapMirror Cloud, the async SnapMirror license is also required and is available only with the purchase of the Data Protection Bundle.

You require the following licenses for creating a SnapMirror Cloud relationship:

- Both a SnapMirror license (purchased through Data Protection Bundle, or through Premium Bundle) and a SnapMirror Cloud license (purchased through Hybrid Cloud Bundle) are needed for replicating directly to the object store endpoint.
- When configuring a multi-policy replication workflow (for example, Disk-to-Disk-to-Cloud), a SnapMirror license is required on all ONTAP instances, while the SnapMirror Cloud license is only required for the source cluster which is replicating directly to the object storage endpoint.

SnapMirror Cloud is an end user license which can be purchased from NetApp or from an approved NetApp reseller partner. The SnapMirror Cloud license provides end user entitlement but does not enable asynchronous ONTAP to object storage replication. To invoke ONTAP APIs for SnapMirror Cloud, a unique API key from an authorized application is required. Authorized and licensed applications used to orchestrate SnapMirror Cloud replication include System Manager, and are also available from multiple third-party application providers. These authorized applications will embed the unique API key to invoke ONTAP APIs. A combination of the SnapMirror Cloud end user license and an authorized third-party backup application is required to orchestrate and enable SnapMirror Cloud replication.

Beginning with ONTAP 9.9.1, you can use System Manager for SnapMirror Cloud replication. For more information, see [Back up to the cloud](#).

A list of authorized SnapMirror Cloud third-party applications is published on the NetApp web site.

**Data Protection Optimized (DPO)**

Beginning with ONTAP 9.1, new ONTAP data protection features were packaged with the FAS8200 as part of a solution called the Data Protection Bundle. This new hardware and software bundle included a new DP\_Optimized (DPO) license that provided unique ONTAP features for secondary workloads. With the introduction of ONTAP 9.3 the DPO license increased the number of volumes per node from 1,000 to 1,500. Also introduced with ONTAP 9.3 were new configurations of the Data Protection Bundle based on configurations of FAS2620.

The DPO license was specifically designed for ONTAP clusters that were to be dedicated as secondary targets for SnapMirror replication. In addition to increasing the maximum volumes per node on the DPO controller, the DPO license also modified controller QoS settings to support greater replication traffic at the expense of application I/O. For this reason, the DPO license should never be installed on a cluster that supports application I/O, as application performance would be impacted. Later, Data Protection Bundles based on the FAS8200 and FAS2620 were offered as a solution and included programmatic free licenses based on the customer environment. When purchasing the solution bundles, free SnapMirror licenses would be provided for select older clusters which replicated to the DPO secondary. While the DPO license is needed on the Data Protection solution cluster, primary clusters from the following platform list would be provided free SnapMirror licenses. Primary clusters not included in this list would require purchase of SnapMirror licenses. The DPO hardware and software bundle was based on both FAS2620 and FAS8200 systems which are both EOA status and no are longer available.

- FAS2200 Series
- FAS3000 Series
- FAS6000 Series
- FAS8000 Series

**Data Protection Optimized (DPO) License**

Data Protection hardware and software solution bundles introduced with ONTAP 9.1 and 9.3 were based on FAS8200 and FAS2620 only. As these platforms matured and new platforms were introduced new requests to support ONTAP features for secondary replication use cases increased. As a result, a new standalone DPO license was introduced in November 2018 with the ONTAP 9.5 release.

The standalone DPO license was supported on both FAS and AFF platforms and could be purchased pre-configured with new clusters or added to deployed clusters as a software upgrade in the field. Because these new DPO licenses were not part of a hardware and software solution bundle, they carried a lower price, and free SnapMirror licenses for primary clusters were not provided. Secondary clusters configured with the a la carte DPO license must also purchase a SnapMirror license, and all primary clusters replicating to the DPO secondary cluster must purchase a SnapMirror license.

Additional ONTAP features were delivered with the DPO across multiple ONTAP releases.

Feature	9.3	9.4	9.5	9.6	9.7+
Max vols/node	1500	1500	1500	1500/2500	1500/2500

Max concurrent repl sessions	100	200	200	200	200
Workload bias*	client apps	Apps/SM	SnapMirror	SnapMirror	SnapMirror
Cross volume aggregate deduplication for HDD	No	Yes	Yes	Yes	Yes

- Details about priority for the SnapMirror backoff (workload bias) feature:
- Client: cluster I/O priority is set to client workloads (production apps), not SnapMirror traffic.
- Equality: SnapMirror replication requests have equal priority to I/O for production apps.
- SnapMirror: all SnapMirror I/O requests have higher priority than I/O for production apps.

**Table 1: Max FlexVolumes per node across ONTAP releases**

	<b>9.3—9.5 Without DPO</b>	<b>9.3—9.5 With DPO</b>	<b>9.6 Without DPO</b>	<b>9.6 With DPO</b>	<b>9.7—9.9.1 Without DPO</b>	<b>9.7—9.9.1 With DPO</b>
FAS2620	1000	1500	1000	1500	1000	1500
FAS2650	1000	1500	1000	1500	1000	1500
FAS2720	1000	1500	1000	1500	1000	1500
FAS2750	1000	1500	1000	1500	1000	1500
A200	1000	1500	1000	1500	1000	1500
A220	1000	1500	1000	1500	1000	1500
FAS8200/8300	1000	1500	1000	2500	1000	2500
A300	1000	1500	1000	2500	2500	2500
A400	1000	1500	1000	2500	2500	2500
FAS8700/9000	1000	1500	1000	2500	1000	2500
A700	1000	1500	1000	2500	2500	2500
A700s	1000	1500	1000	2500	2500	2500

A800	1000	1500	1000	2500	2500	2500
------	------	------	------	------	------	------

For the latest maximum FlexVol volume support for your configuration, see [Hardware Universe](#).

## Considerations for all new DPO installations

- After it is enabled, the DPO license feature cannot be disabled or undone.
- Installation of the DPO license requires a re-boot of ONTAP or failover to enable.
- The DPO solution is intended for secondary storage workloads; application workload performance on DPO clusters may be impacted
- The DPO license is supported on a select list of NetApp storage platform models.
- DPO features vary by ONTAP release. Refer to the compatibility table for reference.
- New FAS and AFF systems are not qualified with DPO. DPO licenses cannot be purchased for clusters not listed above.

## Install a SnapMirror Cloud license

Beginning with ONTAP 9.8, SnapMirror Cloud provides asynchronous snapshot replication from ONTAP to object storage endpoints. SnapMirror Cloud relationships can only be configured using pre-qualified third-party backup applications. To configure ONTAP to object storage replication, both SnapMirror and SnapMirror Cloud licenses are required on the ONTAP source cluster configured for replication to the object store endpoint.

### About this task

The SnapMirror Cloud license is a single-instance cluster-wide license, which means it does not need to be installed on every node in the cluster. It is a term-based license where both term and backup capacity are enforced. In addition to this end user license, SnapMirror Cloud requires an authorized and approved backup application to configure and invoke ONTAP APIs for replication. Both SnapMirror Cloud end user license and authorized app are necessary to utilize SnapMirror Cloud replication.

SnapMirror Cloud licenses are acquired through purchase of the Hybrid Cloud Bundle, which can be purchased with 1 or 3 year terms in 1 TB increments. The Hybrid Cloud Bundle includes a license for SnapMirror Cloud. Each license has a unique serial number. Purchases of the Hybrid Cloud Bundle are based on capacity, where the purchased capacity of the Hybrid Cloud Bundle is applied to the SnapMirror Cloud license.

The SnapMirror Cloud license can be installed on the cluster using the ONTAP command line or System Manager.

### Steps

1. Download two NetApp License File (NLF) for SnapMirror Cloud from the NetApp Support Site.

[NetApp Support](#)

2. Use System Manager to upload the SnapMirror Cloud NLF file to the cluster:
  - a. Click **Configuration > Licenses**.
  - b. In the **Cluster Settings** pane, click **Licenses**.



- c. In the **Packages** window, click **Add**.
- d. In the **Add License Packages** dialog box, click **Choose Files** to select the NLF you downloaded, and then click **Add** to upload the file to the cluster.

## Related information

[NetApp Software License Search](#)

## DPO systems feature enhancements

Beginning with ONTAP 9.6, the maximum number of FlexVol volumes supported increases when the DP\_Optimized (DPO) license is installed. Beginning with ONTAP 9.4, systems with the DPO license support SnapMirror backoff, cross-volume background deduplication, cross-volume inline deduplication, use of Snapshot blocks as donors, and compaction.

Beginning with ONTAP 9.6, the maximum supported number of FlexVol volumes on secondary or data protection systems has increased, enabling you to scale up to 2,500 FlexVol volumes per node, or up to 5,000 in failover mode. The increase in FlexVol volumes is enabled with the DP\_Optimized (DPO) license. A SnapMirror license is still required on both the source and destination nodes.

Beginning with ONTAP 9.4, the following feature enhancements are made to DPO systems:

- SnapMirror backoff: In DPO systems, replication traffic is given the same priority that client workloads are given.

SnapMirror backoff is disabled by default on DPO systems.

- Volume background deduplication and cross-volume background deduplication: Volume background deduplication and cross-volume background deduplication are enabled in DPO systems.

You can run the `storage aggregate efficiency cross-volume-dedupe start -aggregate aggregate_name -scan-old-data true` command to deduplicate the existing data. The best practice is to run the command during off-peak hours to reduce the impact on performance.

- Increased savings by using Snapshot blocks as donors: The data blocks that are not available in the active file system but are trapped in Snapshot copies are used as donors for volume deduplication.

The new data can be deduplicated with the data that was trapped in Snapshot copies, effectively sharing the Snapshot blocks as well. The increased donor space provides more savings, especially when the volume has a large number of Snapshot copies.

- Compaction: Data compaction is enabled by default on DPO volumes.

## Manage SnapMirror volume replication

### SnapMirror replication workflow

SnapMirror offers three types of data protection relationship: SnapMirror DR, archive (previously known as SnapVault), and unified replication. You can follow the same basic workflow to configure each type of relationship.

Beginning with general availability in ONTAP 9.9.1, SnapMirror Business Continuity (SM-BC) provides Zero

Recovery Time Objective (Zero RTO) or Transparent Application Failover (TAF) to enable automatic failover of business-critical applications in SAN environments. SM-BC is supported in a configuration of either two AFF clusters or two All SAN Array (ASA) clusters.

NetApp Documentation: SnapMirror Business Continuity

For each type of SnapMirror data protection relationship, the workflow is the same: create a destination volume, create a job schedule, specify a policy, create and initialize the relationship.

Beginning with ONTAP 9.3, you can use the `snapmirror protect` command to configure a data protection relationship in a single step. Even if you use `snapmirror protect`, you need to understand each step in the workflow.



Configure a replication relationship in one step

Beginning with ONTAP 9.3, you can use the `snapmirror protect` command to

configure a data protection relationship in a single step. You specify a list of volumes to be replicated, an SVM on the destination cluster, a job schedule, and a SnapMirror policy. `snapmirror protect` does the rest.

### What you'll need

- The source and destination clusters and SVMs must be peered.

#### Cluster and SVM peering

- The language on the destination volume must be the same as the language on the source volume.

### About this task

The `snapmirror protect` command chooses an aggregate associated with the specified SVM. If no aggregate is associated with the SVM, it chooses from all the aggregates in the cluster. The choice of aggregate is based on the amount of free space and the number of volumes on the aggregate.

The `snapmirror protect` command then performs the following steps:

- Creates a destination volume with an appropriate type and amount of reserved space for each volume in the list of volumes to be replicated.
- Configures a replication relationship appropriate for the policy you specify.
- Initializes the relationship.

The name of the destination volume is of the form *source\_volume\_name\_dst*. In case of a conflict with an existing name, the command appends a number to the volume name. You can specify a prefix and/or suffix in the command options. The suffix replaces the system-supplied `dst` suffix.

In ONTAP 9.3 and earlier, a destination volume can contain up to 251 Snapshot copies. In ONTAP 9.4 and later, a destination volume can contain up to 1019 Snapshot copies.



Initialization can be time-consuming. `snapmirror protect` does not wait for initialization to complete before the job finishes. For this reason, you should use the `snapmirror show` command rather than the `job show` command to determine when initialization is complete.

Beginning with ONTAP 9.5, SnapMirror Synchronous relationships can be created by using the `snapmirror protect` command.

### Step

1. Create and initialize a replication relationship in one step:

```
snapmirror protect -path-list SVM:volume|cluster://SVM/volume, ... -destination
-vserver destination_SVM -policy policy -schedule schedule -auto-initialize
true|false -destination-volume-prefix prefix -destination-volume-suffix suffix
```



You must run this command from the destination SVM or the destination cluster. The `-auto-initialize` option defaults to "true".

The following example creates and initializes a SnapMirror DR relationship using the default `MirrorAllSnapshots` policy:

```
cluster_dst:> snapmirror protect -path-list svm1:volA, svm1:volB
-destination-vserver svm_backup -policy MirrorAllSnapshots -schedule
replication_daily
```



You can use a custom policy if you prefer. For more information, see [Creating a custom replication policy](#).

The following example creates and initializes a SnapVault relationship using the default XDPDefault policy:

```
cluster_dst:> snapmirror protect -path-list svm1:volA, svm1:volB
-destination-vserver svm_backup -policy XDPDefault -schedule
replication_daily
```

The following example creates and initializes a unified replication relationship using the default MirrorAndVault policy:

```
cluster_dst:> snapmirror protect -path-list svm1:volA, svm1:volB
-destination-vserver svm_backup -policy MirrorAndVault
```

The following example creates and initializes a SnapMirror Synchronous relationship using the default Sync policy:

```
cluster_dst:> snapmirror protect -path-list svm1:volA, svm1:volB
-destination-vserver svm_sync -policy Sync
```



For SnapVault and unified replication policies, you might find it useful to define a schedule for creating a copy of the last transferred Snapshot copy on the destination. For more information, see [Defining a schedule for creating a local copy on the destination](#).

## After you finish

Use the `snapmirror show` command to verify that the SnapMirror relationship was created. For complete command syntax, see the man page.

## Configure a replication relationship one step at a time

### Create a destination volume

You can use the `volume create` command on the destination to create a destination volume. The destination volume should be the same or greater in size than the source volume.

### Step

## 1. Create a destination volume:

```
volume create -vserver SVM -volume volume -aggregate aggregate -type DP -size size
```

For complete command syntax, see the man page.

The following example creates a 2-GB destination volume named `volA_dst`:

```
cluster_dst::> volume create -vserver SVM_backup -volume volA_dst  
-aggregate node01_aggr -type DP -size 2GB
```

## Create a replication job schedule

You can use the `job schedule cron create` command to create a replication job schedule. The job schedule determines when SnapMirror automatically updates the data protection relationship to which the schedule is assigned.

### About this task

You assign a job schedule when you create a data protection relationship. If you do not assign a job schedule, you must update the relationship manually.

### Step

#### 1. Create a job schedule:

```
job schedule cron create -name job_name -month month -dayofweek day_of_week  
-day day_of_month -hour hour -minute minute
```

For `-month`, `-dayofweek`, and `-hour`, you can specify `all` to run the job every month, day of the week, and hour, respectively.

Beginning with ONTAP 9.10.1, you can include the Vserver for your job schedule:

```
job schedule cron create -name job_name -vserver Vserver_name -month month  
-dayofweek day_of_week -day day_of_month -hour hour -minute minute
```

The following example creates a job schedule named `my_weekly` that runs on Saturdays at 3:00 a.m.:

```
cluster_dst::> job schedule cron create -name my_weekly -dayofweek  
"Saturday" -hour 3 -minute 0
```

## Customize a replication policy

### Create a custom replication policy

You can create a custom replication policy if the default policy for a relationship is not suitable. You might want to compress data in a network transfer, for example, or modify

the number of attempts SnapMirror makes to transfer Snapshot copies.

You can use a default or custom policy when you create a replication relationship. For a custom archive (formerly SnapVault) or unified replication policy, you must define one or more *rules* that determine which Snapshot copies are transferred during initialization and update. You might also want to define a schedule for creating local Snapshot copies on the destination.

The *policy type* of the replication policy determines the type of relationship it supports. The table below shows the available policy types.

Policy type	Relationship type
async-mirror	SnapMirror DR
vault	SnapVault
mirror-vault	Unified replication
strict-sync-mirror	SnapMirror Synchronous in the StrictSync mode (supported beginning with ONTAP 9.5)
sync-mirror	SnapMirror Synchronous in the Sync mode (supported beginning with ONTAP 9.5)



When you create a custom replication policy, it is a good idea to model the policy after a default policy.

## Step

1. Create a custom replication policy:

```
snapmirror policy create -vserver SVM -policy policy -type async-  
mirror|vault|mirror-vault|strict-sync-mirror|sync-mirror -comment comment  
-tries transfer_tries -transfer-priority low|normal -is-network-compression  
-enabled true|false
```

For complete command syntax, see the man page.

Beginning with ONTAP 9.5, you can specify the schedule for creating a common Snapshot copy schedule for SnapMirror Synchronous relationships by using the `-common-snapshot-schedule` parameter. By default, the common Snapshot copy schedule for SnapMirror Synchronous relationships is one hour. You can specify a value from 30 minutes to two hours for the Snapshot copy schedule for SnapMirror Synchronous relationships.

The following example creates a custom replication policy for SnapMirror DR that enables network compression for data transfers:

```
cluster_dst::> snapmirror policy create -vserver svm1 -policy  
DR_compressed -type async-mirror -comment "DR with network compression  
enabled" -is-network-compression-enabled true
```

The following example creates a custom replication policy for SnapVault:

```
cluster_dst:> snapmirror policy create -vserver svml -policy
my_snapvault -type vault
```

The following example creates a custom replication policy for unified replication:

```
cluster_dst:> snapmirror policy create -vserver svml -policy my_unified
-type mirror-vault
```

The following example creates a custom replication policy for SnapMirror Synchronous relationship in the StrictSync mode:

```
cluster_dst:> snapmirror policy create -vserver svml -policy
my_strictsync -type strict-sync-mirror -common-snapshot-schedule
my_sync_schedule
```

### After you finish

For “vault” and “mirror-vault” policy types, you must define rules that determine which Snapshot copies are transferred during initialization and update.

Use the `snapmirror policy show` command to verify that the SnapMirror policy was created. For complete command syntax, see the man page.

### Define a rule for a policy

For custom policies with the “vault” or “mirror-vault” policy type, you must define at least one rule that determines which Snapshot copies are transferred during initialization and update. You can also define rules for default policies with the “vault” or “mirror-vault” policy type.

### About this task

Every policy with the “vault” or “mirror-vault” policy type must have a rule that specifies which Snapshot copies to replicate. The rule “bi-monthly”, for example, indicates that only Snapshot copies assigned the SnapMirror label “bi-monthly” should be replicated. You specify the SnapMirror label when you configure the Snapshot policy on the source.

Each policy type is associated with one or more system-defined rules. These rules are automatically assigned to a policy when you specify its policy type. The table below shows the system-defined rules.

System-defined rule	Used in policy types	Result
sm_created	async-mirror, mirror-vault, Sync, StrictSync	A Snapshot copy created by SnapMirror is transferred on initialization and update.

all_source_snapshots	async-mirror	New Snapshot copies on the source are transferred on initialization and update.
daily	vault,mirror-vault	New Snapshot copies on the source with the SnapMirror label “daily” are transferred on initialization and update.
weekly	vault,mirror-vault	New Snapshot copies on the source with the SnapMirror label “weekly” are transferred on initialization and update.
monthly	mirror-vault	New Snapshot copies on the source with the SnapMirror label “monthly” are transferred on initialization and update.
app_consistent	Sync, StrictSync	Snapshot copies with the SnapMirror label “app_consistent” on source are synchronously replicated to the destination. Supported Beginning with ONTAP 9.7.

Except for the “async-mirror” policy type, you can specify additional rules as needed, for default or custom policies. For example:

- For the default `MirrorAndVault` policy, you might create a rule called “bi-monthly” to match Snapshot copies on the source with the “bi-monthly” SnapMirror label.
- For a custom policy with the “mirror-vault” policy type, you might create a rule called “bi-weekly” to match Snapshot copies on the source with the “bi-weekly” SnapMirror label.

## Step

1. Define a rule for a policy:

```
snapmirror policy add-rule -vserver SVM -policy policy_for_rule -snapmirror
-label snapmirror-label -keep retention_count
```

For complete command syntax, see the man page.

The following example adds a rule with the SnapMirror label `bi-monthly` to the default `MirrorAndVault` policy:

```
cluster_dst::> snapmirror policy add-rule -vserver svml -policy
MirrorAndVault -snapmirror-label bi-monthly -keep 6
```

The following example adds a rule with the SnapMirror label `bi-weekly` to the custom `my_snapvault`



policy:

```
cluster_dst:> snapmirror policy add-rule -vserver svml -policy  
my_snapvault -snapmirror-label bi-weekly -keep 26
```

The following example adds a rule with the SnapMirror label `app_consistent` to the custom `Sync` policy:

```
cluster_dst:> snapmirror policy add-rule -vserver svml -policy Sync  
-snapmirror-label app_consistent -keep 1
```

You can then replicate Snapshot copies from the source cluster that match this SnapMirror label:

```
cluster_src:> snapshot create -vserver vs1 -volume voll -snapshot  
snapshot1 -snapmirror-label app_consistent
```

### Define a schedule for creating a local copy on the destination

For SnapVault and unified replication relationships, you can protect against the possibility that an updated Snapshot copy is corrupted by creating a copy of the last transferred Snapshot copy on the destination. This “local copy” is retained regardless of the retention rules on the source, so that even if the Snapshot originally transferred by SnapMirror is no longer available on the source, a copy of it will be available on the destination.

#### About this task

You specify the schedule for creating a local copy in the `-schedule` option of the `snapmirror policy add-rule` command.

#### Step

1. Define a schedule for creating a local copy on the destination:

```
snapmirror policy add-rule -vserver SVM -policy policy_for_rule -snapmirror  
-label snapmirror-label -schedule schedule
```

For complete command syntax, see the man page. For an example of how to create a job schedule, see [Creating a replication job schedule](#).

The following example adds a schedule for creating a local copy to the default `MirrorAndVault` policy:

```
cluster_dst:> snapmirror policy add-rule -vserver svml -policy  
MirrorAndVault -snapmirror-label my_monthly -schedule my_monthly
```

The following example adds a schedule for creating a local copy to the custom `my_unified` policy:

```
cluster_dst:> snapmirror policy add-rule -vserver svm1 -policy
my_unified -snapmirror-label my_monthly -schedule my_monthly
```

### Create a replication relationship

The relationship between the source volume in primary storage and the destination volume in secondary storage is called a *data protection relationship*. You can use the `snapmirror create` command to create SnapMirror DR, SnapVault, or unified replication data protection relationships.

### What you'll need

- The source and destination clusters and SVMs must be peered.

#### Cluster and SVM peering

- The language on the destination volume must be the same as the language on the source volume.

### About this task

Until ONTAP 9.3, SnapMirror invoked in DP mode and SnapMirror invoked in XDP mode used different replication engines, with different approaches to version-dependence:

- SnapMirror invoked in DP mode used a *version-dependent* replication engine in which the ONTAP version was required to be the same on primary and secondary storage:

```
cluster_dst:> snapmirror create -type DP -source-path ... -destination
-path ...
```

- SnapMirror invoked in XDP mode used a *version-flexible* replication engine that supported different ONTAP versions on primary and secondary storage:

```
cluster_dst:> snapmirror create -type XDP -source-path ...
-destination-path ...
```

With improvements in performance, the significant benefits of version-flexible SnapMirror outweigh the slight advantage in replication throughput obtained with version-dependent mode. For this reason, beginning with ONTAP 9.3, XDP mode has been made the new default, and any invocations of DP mode on the command line or in new or existing scripts are automatically converted to XDP mode.

Existing relationships are not affected. If a relationship is already of type DP, it will continue to be of type DP. The table below shows the behavior you can expect.

If you specify...	The type is...	The default policy (if you do not specify a policy) is...
DP	XDP	MirrorAllSnapshots (SnapMirror DR)

Nothing	XDP	MirrorAllSnapshots (SnapMirror DR)
XDP	XDP	XDPDefault (SnapVault)

See also the examples in the procedure below.

The only exceptions to conversion are as follows:

- SVM data protection relationships continue to default to DP mode.

Specify XDP explicitly to obtain XDP mode with the default `MirrorAllSnapshots` policy.

- Load-sharing data protection relationships continue to default to DP mode.
- SnapLock data protection relationships continue to default to DP mode.
- Explicit invocations of DP continue to default to DP mode if you set the following cluster-wide option:

```
options replication.create_data_protection_rels.enable on
```

This option is ignored if you do not explicitly invoke DP.

In ONTAP 9.3 and earlier, a destination volume can contain up to 251 Snapshot copies. In ONTAP 9.4 and later, a destination volume can contain up to 1019 Snapshot copies.

Beginning with ONTAP 9.5, SnapMirror Synchronous relationships are supported.

## Step

1. From the destination cluster, create a replication relationship:

```
snapmirror create -source-path SVM:volume|cluster://SVM/volume, ... -destination
-path SVM:volume|cluster://SVM/volume, ... -type DP|XDP -schedule schedule
-policy policy
```

For complete command syntax, see the man page.



The `schedule` parameter is not applicable when creating SnapMirror Synchronous relationships.

The following example creates a SnapMirror DR relationship using the default `MirrorLatest` policy:

```
cluster_dst::> snapmirror create -source-path svm1:volA -destination
-path svm_backup:volA_dst -type XDP -schedule my_daily -policy
MirrorLatest
```

The following example creates a SnapVault relationship using the default `XDPDefault` policy:

```
cluster_dst:> snapmirror create -source-path svm1:volA -destination  
-path svm_backup:volA_dst -type XDP -schedule my_daily -policy  
XDPDefault
```

The following example creates a unified replication relationship using the default `MirrorAndVault` policy:

```
cluster_dst:> snapmirror create -source-path svm1:volA -destination-path  
svm_backup:volA_dst -type XDP -schedule my_daily -policy MirrorAndVault
```

The following example creates a unified replication relationship using the custom `my_unified` policy:

```
cluster_dst:> snapmirror create -source-path svm1:volA -destination  
-path svm_backup:volA_dst -type XDP -schedule my_daily -policy  
my_unified
```

The following example creates a SnapMirror Synchronous relationship using the default `Sync` policy:

```
cluster_dst:> snapmirror create -source-path svm1:volA -destination  
-path svm_backup:volA_dst -type XDP -policy Sync
```

The following example creates a SnapMirror Synchronous relationship using the default `StrictSync` policy:

```
cluster_dst:> snapmirror create -source-path svm1:volA -destination  
-path svm_backup:volA_dst -type XDP -policy StrictSync
```

The following example creates a SnapMirror DR relationship. With the DP type automatically converted to XDP and with no policy specified, the policy defaults to the `MirrorAllSnapshots` policy:

```
cluster_dst:> snapmirror create -source-path svm1:volA -destination  
-path svm_backup:volA_dst -type DP -schedule my_daily
```

The following example creates a SnapMirror DR relationship. With no type or policy specified, the policy defaults to the `MirrorAllSnapshots` policy:

```
cluster_dst:> snapmirror create -source-path svm1:volA -destination  
-path svm_backup:volA_dst -schedule my_daily
```

The following example creates a SnapMirror DR relationship. With no policy specified, the policy defaults to the `XDPDefault` policy:

```
cluster_dst:> snapmirror create -source-path svm1:volA -destination  
-path svm_backup:volA_dst -type XDP -schedule my_daily
```

The following example creates a SnapMirror Synchronous relationship with the predefined policy SnapCenterSync:

```
cluster_dst:> snapmirror create -source-path svm1:volA -destination  
-path svm_backup:volA_dst -type XDP -policy SnapCenterSync
```



The predefined policy SnapCenterSync is of type Sync. This policy replicates any Snapshot copy that is created with the snapmirror-label of "app\_consistent".

### After you finish

Use the `snapmirror show` command to verify that the SnapMirror relationship was created. For complete command syntax, see the man page.

### 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)	<a href="#">Configure mirrors and vaults</a>
System Manager Classic (available with ONTAP 9.7 and earlier)	<a href="#">Volume backup using SnapVault overview</a>

### Initialize a replication relationship

For all relationship types, initialization performs a *baseline transfer*: it makes a Snapshot copy of the source volume, then transfers that copy and all the data blocks it references to the destination volume. Otherwise, the contents of the transfer depend on the policy.

#### What you'll need

The source and destination clusters and SVMs must be peered.

#### [Cluster and SVM peering](#)

#### About this task

Initialization can be time-consuming. You might want to run the baseline transfer in off-peak hours.

Beginning with ONTAP 9.5, SnapMirror Synchronous relationships are supported.

#### Step

1. Initialize a replication relationship:

```
snapmirror initialize -source-path SVM:volume|cluster://SVM/volume, ...  
-destination-path SVM:volume|cluster://SVM/volume, ...
```

For complete command syntax, see the man page.



You must run this command from the destination SVM or the destination cluster.

The following example initializes the relationship between the source volume `volA` on `svm1` and the destination volume `volA_dst` on `svm_backup`:

```
cluster_dst::> snapmirror initialize -source-path svm1:volA -destination
-path svm_backup:volA_dst
```

#### Example: Configure a vault-vault cascade

An example will show in concrete terms how you can configure replication relationships one step at a time. You can use the vault-vault cascade deployment configured in the example to retain more than 251 Snapshot copies labeled “my-weekly”.

#### What you’ll need

- The source and destination clusters and SVMs must be peered.
- You must be running ONTAP 9.2 or later. Vault-vault cascades are not supported in earlier ONTAP releases.

#### About this task

The example assumes the following:

- You have configured Snapshot copies on the source cluster with the SnapMirror labels “my-daily”, “my-weekly”, and “my-monthly”.
- You have configured destination volumes named “volA” on the secondary and tertiary destination clusters.
- You have configured replication job schedules named “my\_snapvault” on the secondary and tertiary destination clusters.

The example shows how to create replication relationships based on two custom policies:

- The “snapvault\_secondary” policy retains 7 daily, 52 weekly, and 180 monthly Snapshot copies on the secondary destination cluster.
- The “snapvault\_tertiary policy” retains 250 weekly Snapshot copies on the tertiary destination cluster.

#### Steps

1. On the secondary destination cluster, create the “snapvault\_secondary” policy:

```
cluster_secondary::> snapmirror policy create -policy snapvault_secondary
-type vault -comment "Policy on secondary for vault to vault cascade" -vserver
svm_secondary
```

2. On the secondary destination cluster, define the “my-daily” rule for the policy:

```
cluster_secondary::> snapmirror policy add-rule -policy snapvault_secondary
-snapmirror-label my-daily -keep 7 -vserver svm_secondary
```

3. On the secondary destination cluster, define the “my-weekly” rule for the policy:

```
cluster_secondary::> snapmirror policy add-rule -policy snapvault_secondary  
-snapmirror-label my-weekly -keep 52 -vserver svm_secondary
```

4. On the secondary destination cluster, define the “my-monthly” rule for the policy:

```
cluster_secondary::> snapmirror policy add-rule -policy snapvault_secondary  
-snapmirror-label my-monthly -keep 180 -vserver svm_secondary
```

5. On the secondary destination cluster, verify the policy:

```
cluster_secondary::> snapmirror policy show snapvault_secondary -instance
```

```

                Vserver: svm_secondary
SnapMirror Policy Name: snapvault_secondary
SnapMirror Policy Type: vault
                Policy Owner: cluster-admin
                Tries Limit: 8
                Transfer Priority: normal
Ignore accesstime Enabled: false
                Transfer Restartability: always
Network Compression Enabled: false
                Create Snapshot: false
                Comment: Policy on secondary for vault to vault
cascade
                Total Number of Rules: 3
                        Total Keep: 239
                                Rules: SnapMirror Label          Keep  Preserve Warn
Schedule Prefix
-----
-----
                                my-daily              7    false      0 -
-
                                my-weekly             52    false      0 -
-
                                my-monthly            180    false      0 -
-

```

6. On the secondary destination cluster, create the relationship with the source cluster:

```
cluster_secondary::> snapmirror create -source-path svm_primary:volA  
-destination-path svm_secondary:volA -type XDP -schedule my_snapvault -policy  
snapvault_secondary
```

7. On the secondary destination cluster, initialize the relationship with the source cluster:

```
cluster_secondary::> snapmirror initialize -source-path svm_primary:volA
```

```
-destination-path svm_secondary:volA
```

8. On the tertiary destination cluster, create the “snapvault\_tertiary” policy:

```
cluster_tertiary::> snapmirror policy create -policy snapvault_tertiary -type
vault -comment "Policy on tertiary for vault to vault cascade" -vserver
svm_tertiary
```

9. On the tertiary destination cluster, define the “my-weekly” rule for the policy:

```
cluster_tertiary::> snapmirror policy add-rule -policy snapvault_tertiary
-snapmirror-label my-weekly -keep 250 -vserver svm_tertiary
```

10. On the tertiary destination cluster, verify the policy:

```
cluster_tertiary::> snapmirror policy show snapvault_tertiary -instance
```

```

                Vserver: svm_tertiary
SnapMirror Policy Name: snapvault_tertiary
SnapMirror Policy Type: vault
                Policy Owner: cluster-admin
                Tries Limit: 8
                Transfer Priority: normal
Ignore accesstime Enabled: false
                Transfer Restartability: always
Network Compression Enabled: false
                Create Snapshot: false
                Comment: Policy on tertiary for vault to vault
cascade
                Total Number of Rules: 1
                        Total Keep: 250
                                Rules: SnapMirror Label          Keep  Preserve Warn
Schedule Prefix
-----
-----
                                my-weekly          250  false          0  -
-
```

11. On the tertiary destination cluster, create the relationship with the secondary cluster:

```
cluster_tertiary::> snapmirror create -source-path svm_secondary:volA
-destination-path svm_tertiary:volA -type XDP -schedule my_snapvault -policy
snapvault_tertiary
```

12. On the tertiary destination cluster, initialize the relationship with the secondary cluster:

```
cluster_tertiary::> snapmirror initialize -source-path svm_secondary:volA
-destination-path svm_tertiary:volA
```



## Convert an existing DP-type relationship to XDP

You can easily convert an existing DP-type relationship to XDP to take advantage of version-flexible SnapMirror.

### About this task

- If you are upgrading to ONTAP 9.12.1 or later, you must convert DP-type relationships to XDP before upgrading. ONTAP 9.12.1 and later does not support DP-type relationships.
- SnapMirror does not automatically convert existing DP-type relationships to XDP. To convert the relationship, you need to break and delete the existing relationship, create a new XDP relationship, and resync the relationship. For background information, see [XDP replaces DP as the SnapMirror default](#).
- When planning your conversion, you should be aware that background preparation and the data warehousing phase of an XDP SnapMirror relationship can take a long time. It is not uncommon to see the SnapMirror relationship reporting the status "preparing" for an extended time period.



After you convert a SnapMirror relationship type from DP to XDP, space-related settings, such as autosize and space guarantee are no longer replicated to the destination.

### Steps

1. From the destination cluster, ensure that the SnapMirror relationship is type DP, that the mirror state is SnapMirrored, the relationship status is Idle, and the relationship is healthy:

```
snapmirror show -destination-path SVM:volume|cluster://SVM/volume
```

The following example shows the output from the `snapmirror show` command:

```
cluster_dst::>snapmirror show -destination-path svm_backup:volA_dst

Source Path: svm1:volA
Destination Path: svm_backup:volA_dst
Relationship Type: DP
SnapMirror Schedule: -
Tries Limit: -
Throttle (KB/sec): unlimited
Mirror State: SnapMirrored
Relationship Status: Idle
Transfer Snapshot: -
Snapshot Progress: -
Total Progress: -
Snapshot Checkpoint: -
Newest Snapshot: snapmirror.10af643c-32d1-11e3-954b-
123478563412_2147484682.2014-06-27_100026
Newest Snapshot Timestamp: 06/27 10:00:55
Exported Snapshot: snapmirror.10af643c-32d1-11e3-954b-
123478563412_2147484682.2014-06-27_100026
Exported Snapshot Timestamp: 06/27 10:00:55
Healthy: true
```



You might find it helpful to retain a copy of the `snapmirror show` command output to keep track existing of the relationship settings.

2. From the source and the destination volumes, ensure that both volumes have a common Snapshot copy:

```
volume snapshot show -vserver SVM -volume volume
```

The following example shows the `volume snapshot show` output for the source and the destination volumes:

```
cluster_src:> volume snapshot show -vserver vsml -volume volA
---Blocks---
Vserver Volume Snapshot State Size Total% Used%
-----
svm1 volA
weekly.2014-06-09_0736 valid 76KB 0% 28%
weekly.2014-06-16_1305 valid 80KB 0% 29%
daily.2014-06-26_0842 valid 76KB 0% 28%
hourly.2014-06-26_1205 valid 72KB 0% 27%
hourly.2014-06-26_1305 valid 72KB 0% 27%
hourly.2014-06-26_1405 valid 76KB 0% 28%
hourly.2014-06-26_1505 valid 72KB 0% 27%
hourly.2014-06-26_1605 valid 72KB 0% 27%
daily.2014-06-27_0921 valid 60KB 0% 24%
hourly.2014-06-27_0921 valid 76KB 0% 28%
snapmirror.10af643c-32d1-11e3-954b-123478563412_2147484682.2014-06-
27_100026
valid 44KB 0% 19%
11 entries were displayed.
```

```
cluster_dest:> volume snapshot show -vserver svm_backup -volume volA_dst
---Blocks---
Vserver Volume Snapshot State Size Total% Used%
-----
svm_backup volA_dst
weekly.2014-06-09_0736 valid 76KB 0% 30%
weekly.2014-06-16_1305 valid 80KB 0% 31%
daily.2014-06-26_0842 valid 76KB 0% 30%
hourly.2014-06-26_1205 valid 72KB 0% 29%
hourly.2014-06-26_1305 valid 72KB 0% 29%
hourly.2014-06-26_1405 valid 76KB 0% 30%
hourly.2014-06-26_1505 valid 72KB 0% 29%
hourly.2014-06-26_1605 valid 72KB 0% 29%
daily.2014-06-27_0921 valid 60KB 0% 25%
hourly.2014-06-27_0921 valid 76KB 0% 30%
snapmirror.10af643c-32d1-11e3-954b-123478563412_2147484682.2014-06-
27_100026
```

### 3. To ensure scheduled updates will not run during the conversion, quiesce the existing DP-type relationship:

```
snapmirror quiesce -source-path SVM:volume|cluster://SVM/volume, ...
-destination-path SVM:volume|cluster://SVM/volume, ...
```

For complete command syntax, see the man page.



You must run this command from the destination SVM or the destination cluster.

The following example quiesces the relationship between the source volume `volA` on `svm1` and the destination volume `volA_dst` on `svm_backup`:

```
cluster_dst:> snapmirror quiesce -destination-path svm_backup:volA_dst
```

#### 4. Break the existing DP-type relationship:

```
snapmirror break -destination-path SVM:volume|cluster://SVM/volume, ...
```

For complete command syntax, see the man page.



You must run this command from the destination SVM or the destination cluster.

The following example breaks the relationship between the source volume `volA` on `svm1` and the destination volume `volA_dst` on `svm_backup`:

```
cluster_dst:> snapmirror break -destination-path svm_backup:volA_dst
```

#### 5. If automatic deletion of Snapshot copies is enabled on the destination volume, disable it:

```
volume snapshot autodelete modify -vserver SVM -volume volume -enabled false
```

The following example disables Snapshot copy autodelete on the destination volume `volA_dst`:

```
cluster_dst:> volume snapshot autodelete modify -vserver svm_backup  
-volume volA_dst -enabled false
```

#### 6. Delete the existing DP-type relationship:

```
snapmirror delete -destination-path SVM:volume|cluster://SVM/volume, ...
```

For complete command syntax, see the man page.



You must run this command from the destination SVM or the destination cluster.

The following example deletes the relationship between the source volume `volA` on `svm1` and the destination volume `volA_dst` on `svm_backup`:

```
cluster_dst:> snapmirror delete -destination-path svm_backup:volA_dst
```

#### 7. You can use the output you retained from the `snapmirror show` command to create the new XDP-type

relationship:

```
snapmirror create -source-path SVM:volume|cluster://SVM/volume, ... -destination  
-path SVM:volume|cluster://SVM/volume, ... -type XDP -schedule schedule -policy  
policy
```

The new relationship must use the same source and destination volume. For complete command syntax, see the man page.



You must run this command from the destination SVM or the destination cluster.

The following example creates a SnapMirror DR relationship between the source volume `volA` on `svm1` and the destination volume `volA_dst` on `svm_backup` using the default `MirrorAllSnapshots` policy:

```
cluster_dst:> snapmirror create -source-path svm1:volA -destination  
-path svm_backup:volA_dst  
-type XDP -schedule my_daily -policy MirrorAllSnapshots
```

#### 8. Resync the source and destination volumes:

```
snapmirror resync -source-path SVM:volume|cluster://SVM/volume, ... -destination  
-path SVM:volume|cluster://SVM/volume, ...
```

To improve resync time, you can use the `-quick-resync` option, but you should be aware that storage efficiency savings can be lost. For complete command syntax, see the man page: [SnapMirror resync command](#).



You must run this command from the destination SVM or the destination cluster. Although resync does not require a baseline transfer, it can be time-consuming. You might want to run the resync in off-peak hours.

The following example resyncs the relationship between the source volume `volA` on `svm1` and the destination volume `volA_dst` on `svm_backup`:

```
cluster_dst:> snapmirror resync -source-path svm1:volA -destination  
-path svm_backup:volA_dst
```

#### 9. If you disabled automatic deletion of Snapshot copies, reenale it:

```
volume snapshot autodelete modify -vserver SVM -volume volume -enabled true
```

#### After you finish

1. Use the `snapmirror show` command to verify that the SnapMirror relationship was created. For complete command syntax, see the man page.
2. Once the SnapMirror XDP destination volume begins updating Snapshot copies as defined by the SnapMirror policy, you can use the output of `snapmirror list-destinations` command from the source cluster to display the new SnapMirror XDP relationship.

## Convert the type of a SnapMirror relationship

Beginning with ONTAP 9.5, SnapMirror Synchronous is supported. You can convert an asynchronous SnapMirror relationship to a SnapMirror Synchronous relationship or vice versa without performing a baseline transfer.

### About this task

You cannot convert an asynchronous SnapMirror relationship to a SnapMirror Synchronous relationship or vice versa by changing the SnapMirror policy

### Steps

- **Converting an asynchronous SnapMirror relationship to a SnapMirror Synchronous relationship**

- a. From the destination cluster, delete the asynchronous SnapMirror relationship:

```
snapmirror delete -destination-path SVM:volume
```

```
cluster2::>snapmirror delete -destination-path vs1_dr:vol1
```

- b. From the source cluster, release the SnapMirror relationship without deleting the common Snapshot copies:

```
snapmirror release -relationship-info-only true -destination-path  
dest_SVM:dest_volume
```

```
cluster1::>snapmirror release -relationship-info-only true  
-destination-path vs1_dr:vol1
```

- c. From the destination cluster, create a SnapMirror Synchronous relationship:

```
snapmirror create -source-path src_SVM:src_volume -destination-path  
dest_SVM:dest_volume -policy sync-mirror
```

```
cluster2::>snapmirror create -source-path vs1:vol1 -destination-path  
vs1_dr:vol1 -policy sync
```

- d. Resynchronize the SnapMirror Synchronous relationship:

```
snapmirror resync -destination-path dest_SVM:dest_volume
```

```
cluster2::>snapmirror resync -destination-path vs1_dr:vol1
```

- **Converting a SnapMirror Synchronous relationship to an asynchronous SnapMirror relationship**

- a. From the destination cluster, quiesce the existing SnapMirror Synchronous relationship:

```
snapmirror quiesce -destination-path dest_SVM:dest_volume
```

```
cluster2::> snapmirror quiesce -destination-path vs1_dr:vol1
```

- b. From the destination cluster, delete the asynchronous SnapMirror relationship:

```
snapmirror delete -destination-path SVM:volume
```

```
cluster2::>snapmirror delete -destination-path vs1_dr:vol1
```

- c. From the source cluster, release the SnapMirror relationship without deleting the common Snapshot copies:

```
snapmirror release -relationship-info-only true -destination-path  
dest_SVM:dest_volume
```

```
cluster1::>snapmirror release -relationship-info-only true  
-destination-path vs1_dr:vol1
```

- d. From the destination cluster, create an asynchronous SnapMirror relationship:

```
snapmirror create -source-path src_SVM:src_volume -destination-path  
dest_SVM:dest_volume -policy MirrorAllSnapshots
```

```
cluster2::>snapmirror create -source-path vs1:vol1 -destination-path  
vs1_dr:vol1 -policy sync
```

- e. Resynchronize the SnapMirror Synchronous relationship:

```
snapmirror resync -destination-path dest_SVM:dest_volume
```

```
cluster2::>snapmirror resync -destination-path vs1_dr:vol1
```

### Convert the mode of a SnapMirror Synchronous relationship

Beginning with ONTAP 9.5, SnapMirror Synchronous relationships are supported. You can convert the mode of a SnapMirror Synchronous relationship from StrictSync to Sync or vice versa.

#### About this task

You cannot modify the policy of a Snapmirror Synchronous relationship to convert its mode.

#### Steps

1. From the destination cluster, quiesce the existing SnapMirror Synchronous relationship:

```
snapmirror quiesce -destination-path dest_SVM:dest_volume
```

```
cluster2::> snapmirror quiesce -destination-path vs1_dr:vol1
```

2. From the destination cluster, delete the existing SnapMirror Synchronous relationship:

```
snapmirror delete -destination-path dest_SVM:dest_volume
```

```
cluster2::> snapmirror delete -destination-path vs1_dr:vol1
```

3. From the source cluster, release the SnapMirror relationship without deleting the common Snapshot copies:

```
snapmirror release -relationship-info-only true -destination-path  
dest_SVM:dest_volume
```

```
cluster1::> snapmirror release -relationship-info-only true -destination  
-path vs1_dr:vol1
```

4. From the destination cluster, create a SnapMirror Synchronous relationship by specifying the mode to which you want to convert the SnapMirror Synchronous relationship:

```
snapmirror create -source-path vs1:vol1 -destination-path dest_SVM:dest_volume  
-policy Sync|StrictSync
```

```
cluster2::> snapmirror create -source-path vs1:vol1 -destination-path  
vs1_dr:vol1 -policy Sync
```

5. From the destination cluster, resynchronize the SnapMirror relationship:

```
snapmirror resync -destination-path dest_SVM:dest_volume
```

```
cluster2::> snapmirror resync -destination-path vs1_dr:vol1
```

## Serve data from a SnapMirror DR destination volume

### Make the destination volume writeable

You need to make the destination volume writeable before you can serve data from the volume to clients. You can use the `snapmirror quiesce` command to stop scheduled transfers to the destination, the `snapmirror abort` command to stop ongoing transfers, and the `snapmirror break` command to make the destination writeable.



## About this task

You must perform this task from the destination SVM or the destination cluster.

### Steps

1. Stop scheduled transfers to the destination:

```
snapmirror quiesce -source-path SVM:volume|cluster://SVM/volume, ...  
-destination-path SVM:volume|cluster://SVM/volume, ...
```

For complete command syntax, see the man page.

The following example stops scheduled transfers between the source volume `volA` on `svm1` and the destination volume `volA_dst` on `svm_backup`:

```
cluster_dst:> snapmirror quiesce -source-path svm1:volA -destination  
-path svm_backup:volA_dst
```

2. Stop ongoing transfers to the destination:

```
snapmirror abort -source-path SVM:volume|cluster://SVM/volume, ... -destination  
-path SVM:volume|cluster://SVM/volume, ...
```

For complete command syntax, see the man page.



This step is not required for SnapMirror Synchronous relationships (supported beginning with ONTAP 9.5).

The following example stops ongoing transfers between the source volume `volA` on `svm1` and the destination volume `volA_dst` on `svm_backup`:

```
cluster_dst:> snapmirror abort -source-path svm1:volA -destination-path  
svm_backup:volA_dst
```

3. Break the SnapMirror DR relationship:

```
snapmirror break -source-path SVM:volume|cluster://SVM/volume, ... -destination  
-path SVM:volume|cluster://SVM/volume, ...
```

For complete command syntax, see the man page.

The following example breaks the relationship between the source volume `volA` on `svm1` and the destination volume `volA_dst` on `svm_backup`:

```
cluster_dst:> snapmirror break -source-path svm1:volA -destination-path  
svm_backup:volA_dst
```

## 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)	<a href="#">Serve data from a SnapMirror destination</a>
System Manager Classic (available with ONTAP 9.7 and earlier)	<a href="#">Volume disaster recovery overview</a>

### Configure the destination volume for data access

After making the destination volume writeable, you must configure the volume for data access. NAS clients, NVMe subsystem, and SAN hosts can access the data from the destination volume until the source volume is reactivated.

NAS environment:

1. Mount the NAS volume to the namespace using the same junction path that the source volume was mounted to in the source SVM.
2. Apply the appropriate ACLs to the SMB shares at the destination volume.
3. Assign the NFS export policies to the destination volume.
4. Apply the quota rules to the destination volume.
5. Redirect clients to the destination volume.
6. Remount the NFS and SMB shares on the clients.

SAN environment:

1. Map the LUNs in the volume to the appropriate initiator group.
2. For iSCSI, create iSCSI sessions from the SAN host initiators to the SAN LIFs.
3. On the SAN client, perform a storage re-scan to detect the connected LUNs.

For information about NVMe environment, see [SAN administration](#).

### Reactivate the original source volume

You can reestablish the original data protection relationship between the source and destination volumes when you no longer need to serve data from the destination.

#### About this task

- The procedure below assumes that the baseline in the original source volume is intact. If the baseline is not intact, you must create and initialize the relationship between the volume you are serving data from and the original source volume before performing the procedure.
- Background preparation and the data warehousing phase of an XDP SnapMirror relationship can take a long time. It is not uncommon to see the SnapMirror relationship reporting the status "preparing" for an extended time period.

#### Steps

1. Delete the original data protection relationship:

```
snapmirror delete -source-path SVM:volume|cluster://SVM/volume, ... -destination  
-path SVM:volume|cluster://SVM/volume, ...
```

For complete command syntax, see the man page.

You must run this command from the destination SVM or the destination cluster.

The following example deletes the relationship between the original source volume, volA on svm1, and the volume you are serving data from, volA\_dst on svm\_backup:

```
cluster_dst::> snapmirror delete -source-path svm1:volA -destination  
-path svm_backup:volA_dst
```

## 2. Reverse the original data protection relationship:

```
snapmirror resync -source-path SVM:volume|cluster://SVM/volume, ... -destination  
-path SVM:volume|cluster://SVM/volume, ...
```

For complete command syntax, see the man page.



You must run this command from the original source SVM or the original source cluster. Although resync does not require a baseline transfer, it can be time-consuming. You might want to run the resync in off-peak hours. The command fails if a common Snapshot copy does not exist on the source and destination. Use `snapmirror initialize` to re-initialize the relationship.

The following example reverses the relationship between the original source volume, volA on svm1, and the volume you are serving data from, volA\_dst on svm\_backup:

```
cluster_src::> snapmirror resync -source-path svm_backup:volA_dst  
-destination-path svm1:volA
```

## 3. When you are ready to reestablish data access to the original source, stop access to the original destination volume. One way to do this is to stop the original destination SVM:

```
vserver stop -vserver SVM
```

For complete command syntax, see the man page.



You must run this command from the original destination SVM or the original destination cluster. This command stops user access to the entire original destination SVM. You may want to stop access to the original destination volume using other methods.

The following example stops the original destination SVM:

```
cluster_dst::> vserver stop svm_backup
```

#### 4. Update the reversed relationship:

```
snapmirror update -source-path SVM:volume|cluster://SVM/volume, ... -destination  
-path SVM:volume|cluster://SVM/volume, ...
```

For complete command syntax, see the man page.



You must run this command from the original source SVM or the original source cluster.

The following example updates the relationship between the volume you are serving data from, `volA_dst` on `svm_backup`, and the original source volume, `volA` on `svm1`:

```
cluster_src::> snapmirror update -source-path svm_backup:volA_dst  
-destination-path svm1:volA
```

#### 5. From the original source SVM or the original source cluster, stop scheduled transfers for the reversed relationship:

```
snapmirror quiesce -source-path SVM:volume|cluster://SVM/volume, ...  
-destination-path SVM:volume|cluster://SVM/volume, ...
```

For complete command syntax, see the man page.



You must run this command from the original source SVM or the original source cluster.

The following example stops scheduled transfers between the original destination volume, `volA_dst` on `svm_backup`, and the original source volume, `volA` on `svm1`:

```
cluster_src::> snapmirror quiesce -source-path svm_backup:volA_dst  
-destination-path svm1:volA
```

#### 6. When the final update is complete and the relationship indicates "Quiesced" for the relationship status, run the following command from the original source SVM or the original source cluster to break the reversed relationship::

```
snapmirror break -source-path SVM:volume|cluster://SVM/volume, ... -destination  
-path SVM:volume|cluster://SVM/volume, ...
```

For complete command syntax, see the man page.



You must run this command from the original source SVM or the source cluster.

The following example breaks the relationship between the original destination volume, `volA_dst` on `svm_backup`, and the original source volume, `volA` on `svm1`:

```
cluster_scr:> snapmirror break -source-path svm_backup:volA_dst  
-destination-path svm1:volA
```

7. From the original source SVM or the original source cluster, delete the reversed data protection relationship:

```
snapmirror delete -source-path SVM:volume|cluster://SVM/volume, ... -destination  
-path SVM:volume|cluster://SVM/volume, ...
```

For complete command syntax, see the man page.



You must run this command from the original source SVM or the original source cluster.

The following example deletes the reversed relationship between the original source volume, volA on svm1, and the volume you are serving data from, volA\_dst on svm\_backup:

```
cluster_src:> snapmirror delete -source-path svm_backup:volA_dst  
-destination-path svm1:volA
```

8. Release the reversed relationship from the original destination SVM or the original destination cluster.

```
snapmirror release -source-path SVM:volume|cluster://SVM/volume, ...  
-destination-path SVM:volume|cluster://SVM/volume, ...
```



You must run this command from the original destination SVM or the original destination cluster.

The following example releases the reversed relationship between the original destination volume, volA\_dst on svm\_backup, and the original source volume, volA on svm1:

```
cluster_dst:> snapmirror release -source-path svm_backup:volA_dst  
-destination-path svm1:volA
```

1. If needed, start the original destination SVM:

```
vserver start -vserver SVM
```

For complete command syntax, see the man page.

The following example starts the original destination SVM:

```
cluster_dst:> vserver start svm_backup
```

2. Reestablish the original data protection relationship from the original destination:

```
snapmirror resync -source-path SVM:volume|cluster://SVM/volume, ... -destination
-path SVM:volume|cluster://SVM/volume, ...
```

For complete command syntax, see the man page.

The following example reestablishes the relationship between the original source volume, `volA` on `svm1`, and the original destination volume, `volA_dst` on `svm_backup`:

```
cluster_dst::> snapmirror resync -source-path svm1:volA -destination
-path svm_backup:volA_dst
```

### After you finish

Use the `snapmirror show` command to verify that the SnapMirror relationship was created. For complete command syntax, see the man page.

## Restore files from a SnapMirror destination volume

### Restore a single file, LUN, or NVMe namespace from a SnapMirror destination

You can restore a single file, LUN, a set of files or LUNs from a Snapshot copy, or an NVMe namespace from a SnapMirror destination volume. Beginning with ONTAP 9.7, you can also restore NVMe namespaces from a SnapMirror Synchronous destination. You can restore files to the original source volume or to a different volume.

### What you'll need

To restore a file or LUN from a SnapMirror Synchronous destination (supported beginning with ONTAP 9.5), you must first delete and release the relationship.

### About this task

The volume to which you are restoring files or LUNs (the destination volume) must be a read-write volume:

- SnapMirror performs an *incremental restore* if the source and destination volumes have a common Snapshot copy (as is typically the case when you are restoring to the original source volume).
- Otherwise, SnapMirror performs a *baseline restore*, in which the specified Snapshot copy and all the data blocks it references are transferred to the destination volume.

### Steps

1. List the Snapshot copies in the destination volume:

```
volume snapshot show -vserver SVM -volume volume
```

For complete command syntax, see the man page.

The following example shows the Snapshot copies on the `vserverB:secondary1` destination:

```
cluster_dst:> volume snapshot show -vserver vserverB -volume secondary1
```

Vserver	Volume	Snapshot	State	Size	Total% Used%
-----	-----	-----	-----	-----	-----
vserverB	secondary1	hourly.2013-01-25_0005	valid	224KB	0%
		daily.2013-01-25_0010	valid	92KB	0%
		hourly.2013-01-25_0105	valid	228KB	0%
		hourly.2013-01-25_0205	valid	236KB	0%
		hourly.2013-01-25_0305	valid	244KB	0%
		hourly.2013-01-25_0405	valid	244KB	0%
		hourly.2013-01-25_0505	valid	244KB	0%

7 entries were displayed.

2. Restore a single file or LUN or a set of files or LUNs from a Snapshot copy in a SnapMirror destination volume:

```
snapmirror restore -source-path SVM:volume|cluster://SVM/volume, ...
-destination-path SVM:volume|cluster://SVM/volume, ... -source-snapshot snapshot
-file-list source_file_path,@destination_file_path
```

For complete command syntax, see the man page.



You must run this command from the destination SVM or the destination cluster.

The following command restores the files `file1` and `file2` from the Snapshot copy `daily.2013-01-25_0010` in the original destination volume `secondary1`, to the same location in the active file system of the original source volume `primary1`:

```
cluster_dst:> snapmirror restore -source-path vserverB:secondary1
-destination-path vserverA:primary1 -source-snapshot daily.2013-01-
25_0010 -file-list /dir1/file1,/dir2/file2
```

```
[Job 3479] Job is queued: snapmirror restore for the relationship with
destination vserverA:primary1
```

The following command restores the files `file1` and `file2` from the Snapshot copy `daily.2013-01-25_0010` in the original destination volume `secondary1`, to a different location in the active file system of the original source volume `primary1`.

The destination file path begins with the `@` symbol followed by the path of the file from the root of the original source volume. In this example, `file1` is restored to `/dir1/file1.new` and `file2` is restored to `/dir2.new/file2` on `primary1`:

```
cluster_dst:> snapmirror restore -source-path vserverB:secondary1
-destination-path vserverA:primary1 -source-snapshot daily.2013-01-
25_0010 -file-list
/dir/file1,@/dir1/file1.new,/dir2/file2,@/dir2.new/file2
```

```
[Job 3479] Job is queued: snapmirror restore for the relationship with
destination vserverA:primary1
```

The following command restores the files `file1` and `file3` from the Snapshot copy `daily.2013-01-25_0010` in the original destination volume `secondary1`, to different locations in the active file system of the original source volume `primary1`, and restores `file2` from `snap1` to the same location in the active file system of `primary1`.

In this example, the file `file1` is restored to `/dir1/file1.new` and `file3` is restored to `/dir3.new/file3`:

```
cluster_dst:> snapmirror restore -source-path vserverB:secondary1
-destination-path vserverA:primary1 -source-snapshot daily.2013-01-
25_0010 -file-list
/dir/file1,@/dir1/file1.new,/dir2/file2,/dir3/file3,@/dir3.new/file3
```

```
[Job 3479] Job is queued: snapmirror restore for the relationship with
destination vserverA:primary1
```

### Restore the contents of a volume from a SnapMirror destination

You can restore the contents of an entire volume from a Snapshot copy in a SnapMirror destination volume. You can restore the volume's contents to the original source volume or to a different volume.

#### About this task

The destination volume for the restore operation must be one of the following:

- A read-write volume, in which case SnapMirror performs an *incremental restore*, provided that the source and destination volumes have a common Snapshot copy (as is typically the case when you are restoring to the original source volume).





The command fails if there is not a common Snapshot copy. You cannot restore the contents of a volume to an empty read-write volume.

- An empty data protection volume, in which case SnapMirror performs a *baseline restore*, in which the specified Snapshot copy and all the data blocks it references are transferred to the source volume.

Restoring the contents of a volume is a disruptive operation. SMB traffic must not be running on the SnapVault primary volume when a restore operation is running.

If the destination volume for the restore operation has compression enabled, and the source volume does not have compression enabled, disable compression on the destination volume. You need to re-enable compression after the restore operation is complete.

Any quota rules defined for the destination volume are deactivated before the restore is performed. You can use the `volume quota modify` command to reactivate quota rules after the restore operation is complete.

### Steps

1. List the Snapshot copies in the destination volume:

```
volume snapshot show -vserver SVM -volume volume
```

For complete command syntax, see the man page.

The following example shows the Snapshot copies on the `vserverB:secondary1` destination:

```
cluster_dst::> volume snapshot show -vserver vserverB -volume secondary1
```

Vserver Used%	Volume	Snapshot	State	Size	Total%
-----	-----	-----	-----	-----	-----
vserverB 0%	secondary1	hourly.2013-01-25_0005	valid	224KB	0%
0%		daily.2013-01-25_0010	valid	92KB	0%
0%		hourly.2013-01-25_0105	valid	228KB	0%
0%		hourly.2013-01-25_0205	valid	236KB	0%
0%		hourly.2013-01-25_0305	valid	244KB	0%
0%		hourly.2013-01-25_0405	valid	244KB	0%
0%		hourly.2013-01-25_0505	valid	244KB	0%

7 entries were displayed.

## 2. Restore the contents of a volume from a Snapshot copy in a SnapMirror destination volume:

```
snapmirror restore -source-path SVM:volume|cluster://SVM/volume, ...  
-destination-path SVM:volume|cluster://SVM/volume, ... -source-snapshot snapshot
```

For complete command syntax, see the man page.



You must run this command from the destination SVM or the destination cluster.

The following command restores the contents of the original source volume `primary1` from the Snapshot copy `daily.2013-01-25_0010` in the original destination volume `secondary1`:

```
cluster_dst:> snapmirror restore -source-path vserverB:secondary1  
-destination-path vserverA:primary1 -source-snapshot daily.2013-01-  
25_0010
```

Warning: All data newer than Snapshot copy `daily.2013-01-25_0010` on volume `vserverA:primary1` will be deleted.

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

```
[Job 34] Job is queued: snapmirror restore from source  
vserverB:secondary1 for the snapshot daily.2013-01-25_0010.
```

## 3. Remount the restored volume and restart all applications that use the volume.

### 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)	<a href="#">Restore a volume from an earlier Snapshot copy</a>
System Manager Classic (available with ONTAP 9.7 and earlier)	<a href="#">Volume restore using SnapVault overview</a>

### Update a replication relationship manually

You might need to update a replication relationship manually if an update fails because the source volume has been moved.

#### About this task

SnapMirror aborts any transfers from a moved source volume until you update the replication relationship manually.

Beginning with ONTAP 9.5, SnapMirror Synchronous relationships are supported. Although the source and destination volumes are in sync at all times in these relationships, the view from the secondary cluster is synchronized with the primary only on an hourly basis. If you want to view the point-in-time data at the destination, you should perform a manual update by running the `snapmirror update` command.

## Step

1. Update a replication relationship manually:

```
snapmirror update -source-path SVM:volume|cluster://SVM/volume, ... -destination  
-path SVM:volume|cluster://SVM/volume, ...
```

For complete command syntax, see the man page.



You must run this command from the destination SVM or the destination cluster. The command fails if a common Snapshot copy does not exist on the source and destination. Use `snapmirror initialize` to re-initialize the relationship.

The following example updates the relationship between the source volume `volA` on `svm1` and the destination volume `volA_dst` on `svm_backup`:

```
cluster_src::> snapmirror update -source-path svm1:volA -destination  
-path svm_backup:volA_dst
```

## Resynchronize a replication relationship

You need to resynchronize a replication relationship after you make a destination volume writeable, after an update fails because a common Snapshot copy does not exist on the source and destination volumes, or if you want to change the replication policy for the relationship.

### About this task

- Although `resync` does not require a baseline transfer, it can be time-consuming. You might want to run the `resync` in off-peak hours.
- Volumes that are part of a fan-out or cascade configuration can take longer to resynchronize. It is not uncommon to see the SnapMirror relationship reporting the status "preparing" for an extended time period.

## Step

1. Resync the source and destination volumes:

```
snapmirror resync -source-path SVM:volume|cluster://SVM/volume, ... -destination  
-path SVM:volume|cluster://SVM/volume, ... -type DP|XDP -schedule schedule  
-policy policy
```

For complete command syntax, see the man page.



You must run this command from the destination SVM or the destination cluster.

The following example resyncs the relationship between the source volume `volA` on `svm1` and the destination volume `volA_dst` on `svm_backup`:

```
cluster_dst:> snapmirror resync -source-path svm1:volA -destination  
-path svm_backup:volA_dst
```

## Delete a volume replication relationship

You can use the `snapmirror delete` and `snapmirror release` commands to delete a volume replication relationship. You can then delete unneeded destination volumes manually.

### About this task

The `snapmirror release` command deletes any SnapMirror-created Snapshot copies from the source. You can use the `-relationship-info-only` option to preserve the Snapshot copies.

### Steps

1. Quiesce the replication relationship:

```
snapmirror quiesce -destination-path SVM:volume|cluster://SVM/volume
```

```
cluster_dst:> snapmirror quiesce -destination-path svm_backup:volA_dst
```

2. (Optional) Break the replication relationship if you require the destination volume to be a read/write volume. You can skip this step if you plan to delete the destination volume or if you don't need the volume to be read/write:

```
snapmirror break -source-path SVM:volume|cluster://SVM/volume, ... -destination  
-path SVM:volume|cluster://SVM/volume, ...
```

```
cluster_dst:> snapmirror break -source-path svm1:volA -destination-path  
svm_backup:volA_dst
```

3. Delete the replication relationship:

```
snapmirror delete -source-path SVM:volume|cluster://SVM/volume, ... -destination  
-path SVM:volume|cluster://SVM/volume, ...
```

For complete command syntax, see the man page.



You must run this command from the destination cluster or destination SVM.

The following example deletes the relationship between the source volume `volA` on `svm1` and the destination volume `volA_dst` on `svm_backup`:

```
cluster_dst:> snapmirror delete -source-path svm1:volA -destination  
-path svm_backup:volA_dst
```

#### 4. Release replication relationship information from the source SVM:

```
snapmirror release -source-path SVM:volume|cluster://SVM/volume, ...  
-destination-path SVM:volume|cluster://SVM/volume, ...
```

For complete command syntax, see the man page.



You must run this command from the source cluster or source SVM.

The following example releases information for the specified replication relationship from the source SVM svm1:

```
cluster_src::> snapmirror release -source-path svm1:volA -destination  
-path svm_backup:volA_dst
```

### Manage storage efficiency

SnapMirror preserves storage efficiency on the source and destination volumes, with one exception, when postprocess data compression is enabled on the destination. In that case, all storage efficiency is lost on the destination. To correct this issue, you need to disable postprocess compression on the destination, update the relationship manually, and re-enable storage efficiency.

#### What you'll need

- The source and destination clusters and SVMs must be peered.

#### Cluster and SVM peering

- You must disable postprocess compression on the destination.

#### About this task

You can use the `volume efficiency show` command to determine whether efficiency is enabled on a volume. For more information, see the man pages.

You can check if SnapMirror is maintaining storage efficiency by viewing the SnapMirror audit logs and locating the transfer description. If the transfer description displays `transfer_desc=Logical Transfer`, SnapMirror is not maintaining storage efficiency. If the transfer description displays `transfer_desc=Logical Transfer with Storage Efficiency`, SnapMirror is maintaining storage efficiency. For example:

```
Fri May 22 02:13:02 CDT 2020 ScheduledUpdate[May 22 02:12:00]:cc0fbc29-  
b665-11e5-a626-00a09860c273 Operation-Uid=39fbcf48-550a-4282-a906-  
df35632c73a1 Group=none Operation-Cookie=0 action=End source=<sourcepath>  
destination=<destpath> status=Success bytes_transferred=117080571  
network_compression_ratio=1.0:1 transfer_desc=Logical Transfer - Optimized  
Directory Mode
```

## Logical Transfer with storage

Beginning with ONTAP 9.3, manual update is no longer required to re-enable storage efficiency. If SnapMirror detects that postprocess compression has been disabled, it automatically re-enables storage efficiency at the next scheduled update. Both the source and the destination must be running ONTAP 9.3.

Beginning with ONTAP 9.3, AFF systems manage storage efficiency settings differently from FAS systems after a destination volume is made writeable:

- After you make a destination volume writeable using the `snapmirror break` command, the caching policy on the volume is automatically set to “auto” (the default).



This behavior is applicable to FlexVol volumes, only, and it does not apply to FlexGroup volumes.

- On resync, the caching policy is automatically set to “none”, and deduplication and inline compression are automatically disabled, regardless of your original settings. You must modify the settings manually as needed.



Manual updates with storage efficiency enabled can be time-consuming. You might want to run the operation in off-peak hours.

### Step

1. Update a replication relationship and re-enable storage efficiency:

```
snapmirror update -source-path SVM:volume|cluster://SVM/volume, ... -destination  
-path SVM:volume|cluster://SVM/volume, ... -enable-storage-efficiency true
```

For complete command syntax, see the man page.



You must run this command from the destination SVM or the destination cluster. The command fails if a common Snapshot copy does not exist on the source and destination. Use `snapmirror initialize` to re-initialize the relationship.

The following example updates the relationship between the source volume `volA` on `svm1` and the destination volume `volA_dst` on `svm_backup`, and re-enables storage efficiency:

```
cluster_dst::> snapmirror update -source-path svm1:volA -destination  
-path svm_backup:volA_dst -enable-storage-efficiency true
```

## Use SnapMirror global throttling

Global network throttling is available for all SnapMirror and SnapVault transfers at a per-node level.

### About this task

SnapMirror global throttling restricts the bandwidth used by incoming and/or outgoing SnapMirror and SnapVault transfers. The restriction is enforced cluster wide on all nodes in the cluster.

For example, if the outgoing throttle is set to 100 MBps, each node in the cluster will have the outgoing

bandwidth set to 100 MBps. If global throttling is disabled, it is disabled on all nodes.

Although data transfer rates are often expressed in bits per second (bps), the throttle values must be entered in kilobytes per second (KBps).



In ONTAP 9.9.1 and earlier releases, the throttle has no effect on `volume move` transfers or load-sharing mirror transfers. Beginning with ONTAP 9.10.0, you can specify an option to throttle a volume move operations. For details, see [How to throttle volume move in ONTAP 9.10 and later](#).

Global throttling works with the per-relationship throttle feature for SnapMirror and SnapVault transfers. The per-relationship throttle is enforced until the combined bandwidth of per-relationship transfers exceeds the value of the global throttle, after which the global throttle is enforced. A throttle value 0 implies that global throttling is disabled.



SnapMirror global throttling has no effect on SnapMirror Synchronous relationships when they are In-Sync. However, the throttle does effect SnapMirror Synchronous relationships when they perform an asynchronous transfer phase such as an initialization operation or after an Out Of Sync event. For this reason, enabling global throttling with SnapMirror Synchronous relationships is not recommended.

## Steps

1. Enable global throttling:

```
options -option-name replication.throttle.enable on|off
```

The following example shows how to enable SnapMirror global throttling on `cluster_dst`:

```
cluster_dst::> options -option-name replication.throttle.enable on
```

2. Specify the maximum total bandwidth used by incoming transfers on the destination cluster:

```
options -option-name replication.throttle.incoming.max_kbs KBps
```

The recommended minimum throttle bandwidth is 4 KBps and the maximum is up to 2 TBps. The default value for this option is `unlimited`, which means there is no limit on total bandwidth used.

The following example shows how to set the maximum total bandwidth used by incoming transfers to 100 Mbps:

```
cluster_dst::> options -option-name  
replication.throttle.incoming.max_kbs 12500
```



100 Mbps = 12500 KBps

3. Specify the maximum total bandwidth used by outgoing transfers on the source cluster:

```
options -option-name replication.throttle.outgoing.max_kbs KBps
```

KBps is the maximum transfer rate in kilobytes per second. Valid transfer rate values are 1 to 125000. The default value for this option is `unlimited`, which means there is no limit on total bandwidth used.

The following example shows how to set the maximum total bandwidth used by outgoing transfers to 100 Mbps:

```
cluster_src::> options -option-name  
replication.throttle.outgoing.max_kbs 12500
```

## About SnapMirror SVM replication

You can use SnapMirror to create a data protection relationship between SVMs. In this type of data protection relationship, all or part of the SVM's configuration, from NFS exports and SMB shares to RBAC, is replicated, as well as the data in the volumes that the SVM owns.

### Supported relationship types

Only data-serving SVMs can be replicated. The following data protection relationship types are supported:

- *SnapMirror DR*, in which the destination typically contains only the Snapshot copies currently on the source.

Beginning with ONTAP 9.9.1, this behavior changes when you are using the mirror-vault policy. Beginning with ONTAP 9.9.1, you can create different Snapshot policies on the source and destination, and the Snapshot copies on the destination are not overwritten by Snapshot copies on the source:

- They are not overwritten from the source to the destination during normal scheduled operations, updates and resync
- They are not deleted during break operations.
- They are not deleted during flip-resync operations.  
When you configure an SVM DR relationship using the mirror-vault policy using ONTAP 9.9.1 and later, the policy behaves as follows:
- User-defined Snapshot copy policies at the source are not copied to the destination.
- System-defined Snapshot copy policies are not copied to the destination.
- Volume association with user and system defined Snapshot policies are not copied to the destination.

SVM.

- Beginning with ONTAP 9.2, *SnapMirror unified replication*, in which the destination is configured for both DR and long-term retention.

Details about these relationship types can be found here: [Understanding SnapMirror volume replication](#).

The *policy type* of the replication policy determines the type of relationship it supports. The following table shows the available policy types.

Policy type	Relationship type
-------------	-------------------



async-mirror	SnapMirror DR
mirror-vault	Unified replication

## XDP replaces DP as the SVM replication default in ONTAP 9.4

Beginning with ONTAP 9.4, SVM data protection relationships default to XDP mode. SVM data protection relationships continue to default to DP mode in ONTAP 9.3 and earlier.

Existing relationships are not affected by the new default. If a relationship is already of type DP, it will continue to be of type DP. The following table shows the behavior you can expect.

If you specify...	The type is...	The default policy (if you do not specify a policy) is...
DP	XDP	MirrorAllSnapshots (SnapMirror DR)
Nothing	XDP	MirrorAllSnapshots (SnapMirror DR)
XDP	XDP	MirrorAndVault (Unified replication)

Details about the changes in the default can be found here: [XDP replaces DP as the SnapMirror default](#).



Version-independence is not supported for SVM replication. In an SVM DR configuration, the destination SVM must be on a cluster running the same ONTAP version as the source SVM cluster to support failover and fail back operations.

## Compatible ONTAP versions for SnapMirror relationships

### How SVM configurations are replicated

The content of an SVM replication relationship is determined by the interaction of the following fields:

- The `-identity-preserve true` option of the `snapmirror create` command replicates the entire SVM configuration.

The `-identity-preserve false` option replicates only the volumes and authentication and authorization configurations of the SVM, and the protocol and name service settings listed in [Configurations replicated in SVM DR relationships](#).

- The `-discard-configs network` option of the `snapmirror policy create` command excludes LIFs and related network settings from SVM replication, for use in cases where the source and destination SVMs are in different subnets.
- The `-vserver-dr-protection unprotected` option of the `volume modify` command excludes the specified volume from SVM replication.

Otherwise, SVM replication is almost identical to volume replication. You can use virtually the same workflow for SVM replication as you use for volume replication.

## Support details

The following table shows support details for SnapMirror SVM replication.

Resource or feature	Support details
Relationship types	<ul style="list-style-type: none"><li>• SnapMirror DR</li><li>• Beginning with ONTAP 9.2, SnapMirror unified replication</li></ul>
Replication scope	Intercluster only. You cannot replicate SVMs in the same cluster.
Version-independence	Not supported.
Deployment types	<ul style="list-style-type: none"><li>• Single source to single destination</li><li>• Beginning with ONTAP 9.4, fan-out. You can fan-out to two destinations only.</li></ul> <p>By default, only one -identity-preserve true relationship is allowed per source SVM.</p>
Autonomous Ransomware Protection	<ul style="list-style-type: none"><li>• Supported beginning with ONTAP 9.12.1. For more information, see <a href="#">Autonomous Ransomware Protection</a></li></ul>
Volume encryption	<ul style="list-style-type: none"><li>• Encrypted volumes on the source are encrypted on the destination.</li><li>• Onboard Key Manager or KMIP servers must be configured on the destination.</li><li>• New encryption keys are generated at the destination.</li><li>• If the destination does not contain a node that supports volume .encryption, replication succeeds, but the destination volumes are not encrypted.</li></ul>
FabricPool	Beginning with ONTAP 9.6, SnapMirror SVM replication is supported with FabricPools.

MetroCluster	<p>Beginning with ONTAP 9.11.1, both sides of a SVM DR relationship within a MetroCluster configuration can act as a source for additional SVM DR configurations.</p> <p>Beginning with ONTAP 9.5, SnapMirror SVM replication is supported on MetroCluster configurations.</p> <ul style="list-style-type: none"> <li>• A MetroCluster configuration cannot be the destination of an SVM DR relationship.</li> <li>• Only an active SVM within a MetroCluster configuration can be the source of an SVM DR relationship.</li> </ul> <p>A source can be a sync-source SVM before switchover or a sync-destination SVM after switchover.</p> <ul style="list-style-type: none"> <li>• When a MetroCluster configuration is in a steady state, the MetroCluster sync-destination SVM cannot be the source of an SVM DR relationship, since the volumes are not online.</li> <li>• When the sync-source SVM is the source of an SVM DR relationship, the source SVM DR relationship information is replicated to the MetroCluster partner.</li> <li>• During the switchover and switchback processes, replication to the SVM DR destination might fail.</li> </ul> <p>However, after the switchover or switchback process completes, the next SVM DR scheduled updates will succeed.</p>
SnapMirror Synchronous	Not supported with SVM DR.

## Configurations replicated in SVM DR relationships

The following table shows the interaction of the `snapmirror create -identity-preserve` option and the `snapmirror policy create -discard-configs network set` option:

Configuration replicated		<code>-identity-preserve true</code>		<code>-identity-preserve false</code>
		Policy without <code>-discard-configs network set</code>	Policy with <code>-discard-configs network set</code>	

Network	NAS LIFs	Yes	No	No
	LIF Kerberos configuration	Yes	No	No
	SAN LIFs	No	No	No
	Firewall policies	Yes	Yes	No
	Routes	Yes	No	No
	Broadcast domain	No	No	No
	Subnet	No	No	No
	IPspace	No	No	No
SMB	SMB server	Yes	Yes	No
	Local groups and local user	Yes	Yes	Yes
	Privilege	Yes	Yes	Yes
	Shadow copy	Yes	Yes	Yes
	BranchCache	Yes	Yes	Yes
	Server options	Yes	Yes	Yes
	Server security	Yes	Yes	No
	Home directory, share	Yes	Yes	Yes
	Symlink	Yes	Yes	Yes
	Fpolicy policy, Fsecurity policy, and Fsecurity NTFS	Yes	Yes	Yes
	Name mapping and group mapping	Yes	Yes	Yes
	Audit information	Yes	Yes	Yes

NFS	Export policies	Yes	Yes	No
	Export policy rules	Yes	Yes	No
	NFS server	Yes	Yes	No
RBAC	Security certificates	Yes	Yes	No
	Login user, public key, role, and role configuration	Yes	Yes	Yes
	SSL	Yes	Yes	No
Name services	DNS and DNS hosts	Yes	Yes	No
	UNIX user and UNIX group	Yes	Yes	Yes
	Kerberos realm and Kerberos keyblocks	Yes	Yes	No
	LDAP and LDAP client	Yes	Yes	No
	Netgroup	Yes	Yes	No
	NIS	Yes	Yes	No
	Web and web access	Yes	Yes	No
Volume	Object	Yes	Yes	Yes
	Snapshot copies, Snapshot policy, and autodelete policy	Yes	Yes	Yes
	Efficiency policy	Yes	Yes	Yes
	Quota policy and quota policy rule	Yes	Yes	Yes
	Recovery queue	Yes	Yes	Yes

Root volume	Namespace	Yes	Yes	Yes
	User data	No	No	No
	Qtrees	No	No	No
	Quotas	No	No	No
	File-level QoS	No	No	No
	Attributes: state of the root volume, space guarantee, size, autosize, and total number of files	No	No	No
Storage QoS	QoS policy group	Yes	Yes	Yes
Fibre Channel (FC)		No	No	No
iSCSI		No	No	No
LUNs	Object	Yes	Yes	Yes
	igroups	No	No	No
	portsets	No	No	No
	Serial numbers	No	No	No
SNMP	v3 users	Yes	Yes	No

## SVM DR storage limits

The following table shows the recommended maximum number of volumes and SVM DR relationships supported per storage object. You should be aware that limits are often platform dependent. Refer to the [Hardware Universe](#) to learn the limits for your specific configuration.

Storage object	Limit
SVM	300 Flexible volumes
HA pair	1,000 Flexible Volumes
Cluster	128 SVM DR relationships

## Manage SnapMirror SVM replication

### Replicate SVM configurations

#### SnapMirror SVM replication workflow

SnapMirror SVM replication involves creating the destination SVM, creating a replication job schedule, and creating and initializing a SnapMirror relationship.



This workflow assumes that you are already using a default policy or a custom replication policy.



#### Criteria for placing volumes on destination SVMs

When replicating volumes from the source SVM to the destination SVM, it's important to know the criteria for selecting aggregates.

Aggregates are selected based on the following criteria:

- Volumes are always placed on non-root aggregates.
- Non-root aggregates are selected based on the available free space and the number of volumes already hosted on the aggregate.

Aggregates with more free space and fewer volumes are given priority. The aggregate with the highest

priority is selected.

- Source volumes on FabricPool aggregates are placed on FabricPool aggregates on the destination with the same tiering-policy.
- If a volume on the source SVM is located on a Flash Pool aggregate, then the volume is placed on a Flash Pool aggregate on the destination SVM, if such an aggregate exists and has enough free space.
- If the `-space-guarantee` option of the volume that is replicated is set to `volume`, only aggregates with free space greater than the volume size are considered.
- The volume size grows automatically on the destination SVM during replication, based on the source volume size.

If you want to pre-reserve the size on the destination SVM, you must resize the volume. The volume size does not shrink automatically on the destination SVM based on the source SVM.

If you want to move a volume from one aggregate to another, you can use the `volume move` command on the destination SVM.

### Replicate an entire SVM configuration

You can use the `-identity-preserve true` option of the `snapmirror create` command to replicate an entire SVM configuration.

#### Before you begin

The source and destination clusters and SVMs must be peered.

For more information, see [Create a cluster peer relationship](#) and [Create an SVM intercluster peer relationship](#).

For complete command syntax, see the man page.

#### About this task

This workflow assumes that you are already using a default policy or a custom replication policy.

Beginning with ONTAP 9.9.1, when you use the mirror-vault policy, you can create different Snapshot policies on the source and destination SVM, and the Snapshot copies on the destination are not overwritten by Snapshot copies on the source. For more information, see [Understanding SnapMirror SVM replication](#).

#### Steps

1. Create a destination SVM:

```
vserver create -vserver SVM_name -subtype dp-destination
```

The SVM name must be unique across the source and destination clusters.

The following example creates a destination SVM named `svm_backup`:

```
cluster_dst:> vserver create -vserver svm_backup -subtype dp-destination
```

2. From the destination cluster, create an SVM peer relationship using the `vserver peer create` command.

For more information, see [Create an SVM intercluster peer relationship](#).



### 3. Create a replication job schedule:

```
job schedule cron create -name job_name -month month -dayofweek day_of_week
-day day_of_month -hour hour -minute minute
```

For `-month`, `-dayofweek`, and `-hour`, you can specify `all` to run the job every month, day of the week, and hour, respectively.

The following example creates a job schedule named `my_weekly` that runs on Saturdays at 3:00 a.m.:

```
cluster_dst:> job schedule cron create -name my_weekly -dayofweek
saturday -hour 3 -minute 0
```

### 4. From the destination SVM or the destination cluster, create a replication relationship:

```
snapmirror create -source-path SVM_name: -destination-path SVM_name: -type
DP|XDP -schedule schedule -policy policy -identity-preserve true
```



You must enter a colon (:) after the SVM name in the `-source-path` and `-destination-path` options.

The following example creates a SnapMirror DR relationship using the default `MirrorAllSnapshots` policy:

```
cluster_dst:> snapmirror create -source-path svm1: -destination-path
svm_backup: -type XDP -schedule my_daily -policy MirrorAllSnapshots
-identity-preserve true
```

The following example creates a unified replication relationship using the default `MirrorAndVault` policy:

```
cluster_dst:> snapmirror create -source-path svm1: -destination-path
svm_backup: -type XDP -schedule my_daily -policy MirrorAndVault
-identity-preserve true
```

Assuming you have created a custom policy with the policy type `async-mirror`, the following example creates a SnapMirror DR relationship:

```
cluster_dst:> snapmirror create -source-path svm1: -destination-path
svm_backup: -type XDP -schedule my_daily -policy my_mirrored -identity
-preserve true
```

Assuming you have created a custom policy with the policy type `mirror-vault`, the following example creates a unified replication relationship:

```
cluster_dst:> snapmirror create -source-path svm1: -destination-path  
svm_backup: -type XDP -schedule my_daily -policy my_unified -identity  
-preserve true
```

#### 5. Stop the destination SVM:

```
vserver stop
```

*SVM name*

The following example stops a destination SVM named dvs1:

```
cluster_dst:> vserver stop -vserver dvs1
```

#### 6. From the destination SVM or the destination cluster, initialize the SVM replication relationship: +

```
snapmirror initialize -source-path SVM_name: -destination-path SVM_name:
```

The following example initializes the relationship between the source SVM, `svm1`, and the destination SVM, `svm_backup`:

```
cluster_dst:> snapmirror initialize -source-path svm1: -destination  
-path svm_backup:
```

#### Exclude LIFs and related network settings from SVM replication

If the source and destination SVMs are in different subnets, you can use the `-discard-configs network` option of the `snapmirror policy create` command to exclude LIFs and related network settings from SVM replication.

#### What you'll need

The source and destination clusters and SVMs must be peered.

For more information, see [Create a cluster peer relationship](#) and [Create an SVM intercluster peer relationship](#).

#### About this task

The `-identity-preserve` option of the `snapmirror create` command must be set to `true` when you create the SVM replication relationship.

For complete command syntax, see the man page.

#### Steps

##### 1. Create a destination SVM:

```
vserver create -vserver SVM -subtype dp-destination
```

The SVM name must be unique across the source and destination clusters.

The following example creates a destination SVM named `svm_backup`:

```
cluster_dst:> vserver create -vserver svm_backup -subtype dp-destination
```

2. From the destination cluster, create an SVM peer relationship using the `vserver peer create` command.

For more information, see [Create an SVM intercluster peer relationship](#).

3. Create a job schedule:

```
job schedule cron create -name job_name -month month -dayofweek day_of_week  
-day day_of_month -hour hour -minute minute
```

For `-month`, `-dayofweek`, and `-hour`, you can specify `all` to run the job every month, day of the week, and hour, respectively.

The following example creates a job schedule named `my_weekly` that runs on Saturdays at 3:00 a.m.:

```
cluster_dst:> job schedule cron create -name my_weekly -dayofweek  
"Saturday" -hour 3 -minute 0
```

4. Create a custom replication policy:

```
snapmirror policy create -vserver SVM -policy policy -type async-  
mirror|vault|mirror-vault -comment comment -tries transfer_tries -transfer  
-priority low|normal -is-network-compression-enabled true|false -discard  
-configs network
```

For complete command syntax, see the man page.

The following example creates a custom replication policy for SnapMirror DR that excludes LIFs:

```
cluster_dst:> snapmirror policy create -vserver svm1 -policy  
DR_exclude_LIFs -type async-mirror -discard-configs network
```

The following example creates a custom replication policy for unified replication that excludes LIFs:

```
cluster_dst:> snapmirror policy create -vserver svm1 -policy  
unified_exclude_LIFs -type mirror-vault -discard-configs network
```

5. From the destination SVM or the destination cluster, run the following command to create a replication relationship:

```
snapmirror create -source-path SVM: -destination-path SVM: -type DP|XDP
```

```
-schedule schedule -policy policy -identity-preserve true|false
```



You must enter a colon (:) after the SVM name in the `-source-path` and `-destination-path` options. See the examples below.

The following example creates a SnapMirror DR relationship that excludes LIFs:

```
cluster_dst:> snapmirror create -source-path svm1: -destination-path  
svm_backup: -type XDP -schedule my_daily -policy DR_exclude_LIFs  
-identity-preserve true
```

The following example creates a SnapMirror unified replication relationship that excludes LIFs:

```
cluster_dst:> snapmirror create -source-path svm1: -destination-path  
svm_backup: -type XDP -schedule my_daily -policy unified_exclude_LIFs  
-identity-preserve true
```

## 6. Stop the destination SVM:

```
vserver stop
```

*SVM name*

The following example stops a destination SVM named `dvs1`:

```
cluster_dst:> vserver stop -vserver dvs1
```

## 7. From the destination SVM or the destination cluster, initialize a replication relationship:

```
snapmirror initialize -source-path SVM: -destination-path SVM:
```

For complete command syntax, see the man page.

The following example initializes the relationship between the source, `svm1` and the destination, `svm_backup`:

```
cluster_dst:> snapmirror initialize -source-path svm1: -destination  
-path svm_backup:
```

## After you finish

You must configure the network and protocols on the destination SVM for data access in the event a disaster occurs.

## Exclude network, name service, and other settings from SVM replication

You can use the `-identity-preserve false` option of the `snapmirror create` command to replicate only the volumes and security configurations of an SVM. Some protocol and name service settings are also preserved.

### What you'll need

The source and destination clusters and SVMs must be peered.

For more information, see [Create a cluster peer relationship](#) and [Create an SVM intercluster peer relationship](#).

### About this task

For a list of preserved protocol and name service settings, see [Configurations replicated in SVM DR relationships](#).

For complete command syntax, see the man page.

### Steps

1. Create a destination SVM:

```
vserver create -vserver SVM -subtype dp-destination
```

The SVM name must be unique across the source and destination clusters.

The following example creates a destination SVM named `svm_backup`:

```
cluster_dst:> vserver create -vserver svm_backup -subtype dp-destination
```

2. From the destination cluster, create an SVM peer relationship using the `vserver peer create` command.

For more information, see [Create an SVM intercluster peer relationship](#).

3. Create a replication job schedule:

```
job schedule cron create -name job_name -month month -dayofweek day_of_week  
-day day_of_month -hour hour -minute minute
```

For `-month`, `-dayofweek`, and `-hour`, you can specify `all` to run the job every month, day of the week, and hour, respectively.

The following example creates a job schedule named `my_weekly` that runs on Saturdays at 3:00 a.m.:

```
cluster_dst:> job schedule cron create -name my_weekly -dayofweek  
"Saturday" -hour 3 -minute 0
```

4. Create a replication relationship that excludes network, name service, and other configuration settings:

```
snapmirror create -source-path SVM: -destination-path SVM: -type DP|XDP
```

```
-schedule schedule -policy policy -identity-preserve false
```



You must enter a colon (:) after the SVM name in the `-source-path` and `-destination-path` options. See the examples below. You must run this command from the destination SVM or the destination cluster.

The following example creates a SnapMirror DR relationship using the default `MirrorAllSnapshots` policy. The relationship excludes network, name service, and other configuration settings from SVM replication:

```
cluster_dst:> snapmirror create -source-path svm1: -destination-path  
svm_backup: -type XDP -schedule my_daily -policy MirrorAllSnapshots  
-identity-preserve false
```

The following example creates a unified replication relationship using the default `MirrorAndVault` policy. The relationship excludes network, name service, and other configuration settings:

```
cluster_dst:> snapmirror create svm1: -destination-path svm_backup:  
-type XDP -schedule my_daily -policy MirrorAndVault -identity-preserve  
false
```

Assuming you have created a custom policy with the policy type `async-mirror`, the following example creates a SnapMirror DR relationship. The relationship excludes network, name service, and other configuration settings from SVM replication:

```
cluster_dst:> snapmirror create -source-path svm1: -destination-path  
svm_backup: -type XDP -schedule my_daily -policy my_mirrored -identity  
-preserve false
```

Assuming you have created a custom policy with the policy type `mirror-vault`, the following example creates a unified replication relationship. The relationship excludes network, name service, and other configuration settings from SVM replication:

```
cluster_dst:> snapmirror create -source-path svm1: -destination-path  
svm_backup: -type XDP -schedule my_daily -policy my_unified -identity  
-preserve false
```

##### 5. Stop the destination SVM:

```
vserver stop
```

*SVM name*

The following example stops a destination SVM named `dvs1`:

```
destination_cluster::> vserver stop -vserver dvs1
```

6. If you are using SMB, you must also configure an SMB server.

See [SMB only: Creating an SMB server](#).

7. From the destination SVM or the destination cluster, initialize the SVM replication relationship:

```
snapmirror initialize -source-path SVM_name: -destination-path SVM_name:
```

### After you finish

You must configure the network and protocols on the destination SVM for data access in the event a disaster occurs.

### Specify aggregates to use for SVM DR relationships

After a disaster recovery SVM is created, you can use the `aggr-list` option with `vserver modify` command to limit which aggregates are used to host SVM DR destination volumes.

### Step

1. Create a destination SVM:

```
vserver create -vserver SVM -subtype dp-destination
```

2. Modify the disaster recovery SVM's `aggr-list` to limit the aggregates that are used to host the disaster recovery SVM's volume:

```
cluster_dest::> vserver modify -vserver SVM -aggr-list <comma-separated-list>
```

### SMB only: Create a SMB server

If the source SVM has an SMB configuration, and you chose to set `identity-preserve` to `false`, you must create a SMB server for the destination SVM. SMB server is required for some SMB configurations, such as shares during initialization of the SnapMirror relationship.

### Steps

1. Start the destination SVM by using the `vserver start` command.

```
destination_cluster::> vserver start -vserver dvs1  
[Job 30] Job succeeded: DONE
```

2. Verify that the destination SVM is in the `running` state and subtype is `dp-destination` by using the `vserver show` command.

```
destination_cluster::> vserver show
```

Vserver	Type	Subtype	Admin State	Operational State	Root Volume
Aggregate					
-----					
dvs1	data	dp-destination	running	running	-

3. Create a LIF by using the `network interface create` command.

```
destination_cluster::>network interface create -vserver dvs1 -lif NAS1  
-role data -data-protocol cifs -home-node destination_cluster-01 -home  
-port a0a-101 -address 192.0.2.128 -netmask 255.255.255.128
```

4. Create a route by using the `network route create` command.

```
destination_cluster::>network route create -vserver dvs1 -destination  
0.0.0.0/0  
-gateway 192.0.2.1
```

## Network management

5. Configure DNS by using the `vserver services dns create` command.

```
destination_cluster::>vserver services dns create -domains  
mydomain.example.com -vserver  
dvs1 -name-servers 192.0.2.128 -state enabled
```

6. Add the preferred domain controller by using the `vserver cifs domain preferred-dc add` command.

```
destination_cluster::>vserver cifs domain preferred-dc add -vserver dvs1  
-preferred-dc  
192.0.2.128 -domain mydomain.example.com
```

7. Create the SMB server by using the `vserver cifs create` command.

```
destination_cluster::>vserver cifs create -vserver dvs1 -domain  
mydomain.example.com  
-cifs-server CIFS1
```



8. Stop the destination SVM by using the `vserver stop` command.

```
destination_cluster::> vserver stop -vserver dvs1
[Job 46] Job succeeded: DONE
```

### Exclude volumes from SVM replication

By default, all RW data volumes of the source SVM are replicated. If you do not want to protect all the volumes on the source SVM, you can use the `-vserver-dr -protection unprotected` option of the `volume modify` command to exclude volumes from SVM replication.

#### Steps

1. Exclude a volume from SVM replication:

```
volume modify -vserver SVM -volume volume -vserver-dr-protection unprotected
```

For complete command syntax, see the man page.

The following example excludes the volume `volA_src` from SVM replication:

```
cluster_src::> volume modify -vserver SVM1 -volume volA_src -vserver-dr
-protection unprotected
```

If you later want to include a volume in the SVM replication that you originally excluded, run the following command:

```
volume modify -vserver SVM -volume volume -vserver-dr-protection protected
```

The following example includes the volume `volA_src` in the SVM replication:

```
cluster_src::> volume modify -vserver SVM1 -volume volA_src -vserver-dr
-protection protected
```

2. Create and initialize the SVM replication relationship as described in [Replicating an entire SVM configuration](#).

### Serve data from an SVM DR destination

#### SVM disaster recovery workflow

To recover from a disaster and serve data from the destination SVM, you must activate the destination SVM. Activating the destination SVM involves stopping scheduled SnapMirror transfers, aborting ongoing SnapMirror transfers, breaking the replication relationship, stopping the source SVM, and starting the destination SVM.



#### Make SVM destination volumes writeable

You need to make SVM destination volumes writeable before you can serve data to clients. The procedure is largely identical to the procedure for volume replication, with one exception. If you set `-identity-preserve true` when you created the SVM replication relationship, you must stop the source SVM before activating the destination SVM.

#### About this task

For complete command syntax, see the man page.



In a disaster recovery scenario, you cannot perform a SnapMirror update from the source SVM to the disaster recovery destination SVM because your source SVM and its data will be inaccessible, and because updates since the last resync might be bad or corrupt.

## Steps

1. From the destination SVM or the destination cluster, stop scheduled transfers to the destination:

```
snapmirror quiesce -source-path SVM: -destination-path SVM:
```



You must enter a colon (:) after the SVM name in the `-source-path` and `-destination-path` options. See the example below.

The following example stops scheduled transfers between the source SVM `svm1` and the destination SVM `svm_backup`:

```
cluster_dst::> snapmirror quiesce -source-path svm1: -destination-path  
svm_backup:
```

2. From the destination SVM or the destination cluster, stop ongoing transfers to the destination:

```
snapmirror abort -source-path SVM: -destination-path SVM:
```



You must enter a colon (:) after the SVM name in the `-source-path` and `-destination-path` options. See the example below.

The following example stops ongoing transfers between the source SVM `svm1` and the destination SVM `svm_backup`:

```
cluster_dst::> snapmirror abort -source-path svm1: -destination-path  
svm_backup:
```

3. From the destination SVM or the destination cluster, break the replication relationship:

```
snapmirror break -source-path SVM: -destination-path SVM:
```



You must enter a colon (:) after the SVM name in the `-source-path` and `-destination-path` options. See the example below.

The following example breaks the relationship between the source SVM `svm1` and the destination SVM `svm_backup`:

```
cluster_dst::> snapmirror break -source-path svm1: -destination-path  
svm_backup:
```

4. If you set `-identity-preserve true` when you created the SVM replication relationship, stop the

source SVM:

```
vserver stop -vserver SVM
```

The following example stops the source SVM `svm1`:

```
cluster_src::> vserver stop svm1
```

#### 5. Start the destination SVM:

```
vserver start -vserver SVM
```

The following example starts the destination SVM `svm_backup`:

```
cluster_dst::> vserver start svm_backup
```

#### After you finish

Configure SVM destination volumes for data access, as described in [Configuring the destination volume for data access](#).

#### Reactivate the source SVM

##### Source SVM reactivation workflow

If the source SVM exists after a disaster, you can reactivate it and protect it by recreating the SVM disaster recovery relationship.



### Reactivate the original source SVM

You can reestablish the original data protection relationship between the source and destination SVM when you no longer need to serve data from the destination. The procedure is largely identical to the procedure for volume replication, with one exception. You must stop the destination SVM before reactivating the source SVM.

#### What you'll need

If you have increased the size of destination volume while serving data from it, before you reactivate the source volume, you should manually increase max-autosize on the original source volume to ensure it can grow sufficiently.

#### When a destination volume grows automatically

#### About this task

Beginning with ONTAP 9.11.1, you can reduce resynchronization time during a disaster recovery rehearsal by using the `-quick-resync true` option of the `snapmirror resync` command while performing a reverse resync of an SVM DR relationship. A quick resync can reduce the time it takes to return to production by bypassing the data warehouse rebuild and restore operations.



Quick resync does not preserve the storage efficiency of the destination volumes. Enabling quick resync might increase the volume space used by the destination volumes.

This procedure assumes that the baseline in the original source volume is intact. If the baseline is not intact, you must create and initialize the relationship between the volume you are serving data from and the original

source volume before performing the procedure.

For complete command syntax on commands, see the man page.

## Steps

1. From the original source SVM or the original source cluster, create a reverse SVM DR relationship using the same configuration, policy, and identity-preserve setting as the original SVM DR relationship:

```
snapmirror create -source-path SVM: -destination-path SVM:
```



You must enter a colon (:) after the SVM name in the `-source-path` and `-destination-path` options. See the example below.

The following example creates a relationship between the SVM from which you are serving data, `svm_backup`, and the original source SVM, `svm1`:

```
cluster_src::> snapmirror create -source-path svm_backup: -destination  
-path svm1:
```

2. From the original source SVM or the original source cluster, run the following command to reverse the data protection relationship:

```
snapmirror resync -source-path SVM: -destination-path SVM:
```



You must enter a colon (:) after the SVM name in the `-source-path` and `-destination-path` options. See the example below.

Although `resync` does not require a baseline transfer, it can be time-consuming. You might want to run the `resync` in off-peak hours.



The command fails if a common Snapshot copy does not exist on the source and destination. Use `snapmirror initialize` to reinitialize the relationship.

The following example reverses the relationship between the original source SVM, `svm1`, and the SVM from which you are serving data, `svm_backup`:

```
cluster_src::> snapmirror resync -source-path svm_backup: -destination  
-path svm1:
```

Example using `-quick-resync` option:

```
cluster_src::> snapmirror resync -source-path svm_backup: -destination  
-path svm1: -quick-resync true
```

3. When you are ready to reestablish data access to the original source SVM, stop the original destination SVM to disconnect any clients currently connected to the original destination SVM.

```
vserver stop -vserver SVM
```

The following example stops the original destination SVM which is currently serving data:

```
cluster_dst::> vserver stop svm_backup
```

4. Verify that the original destination SVM is in the stopped state by using the `vserver show` command.

```
cluster_dst::> vserver show
```

Vserver	Type	Subtype	Admin State	Operational State	Root Volume
Aggregate					
-----	-----	-----	-----	-----	-----
-----					
svm_backup	data	default	stopped	stopped	rv
aggr1					

5. From the original source SVM or the original source cluster, run the following command to perform the final update of the reversed relationship to transfer all changes from the original destination SVM to the original source SVM:

```
snapmirror update -source-path SVM: -destination-path SVM:
```



You must enter a colon (:) after the SVM name in the `-source-path` and `-destination-path` options. See the example below.

The following example updates the relationship between the original destination SVM from which you are serving data, `svm_backup`, and the original source SVM, `svm1`:

```
cluster_src::> snapmirror update -source-path svm_backup: -destination-path svm1:
```

6. From the original source SVM or the original source cluster, run the following command to stop scheduled transfers for the reversed relationship:

```
snapmirror quiesce -source-path SVM: -destination-path SVM:
```



You must enter a colon (:) after the SVM name in the `-source-path` and `-destination-path` options. See the example below.

The following example stops scheduled transfers between the SVM you are serving data from, `svm_backup`, and the original SVM, `svm1`:

```
cluster_src::> snapmirror quiesce -source-path svm_backup: -destination  
-path svm1:
```

7. When the final update is complete and the relationship indicates "Quiesced" for the relationship status, run the following command from the original source SVM or the original source cluster to break the reversed relationship:

```
snapmirror break -source-path SVM: -destination-path SVM:
```



You must enter a colon (:) after the SVM name in the `-source-path` and `-destination-path` options. See the example below.

The following example breaks the relationship between the original destination SVM from which you were serving data, `svm_backup`, and the original source SVM, `svm1`:

```
cluster_src::> snapmirror break -source-path svm_backup: -destination  
-path svm1:
```

8. If the original source SVM was previously stopped, from the original source cluster, start the original source SVM:

```
vserver start -vserver SVM
```

The following example starts the original source SVM:

```
cluster_src::> vserver start svm1
```

9. From the original destination SVM or the original destination cluster, reestablish the original data protection relationship:

```
snapmirror resync -source-path SVM: -destination-path SVM:
```



You must enter a colon (:) after the SVM name in the `-source-path` and `-destination-path` options. See the example below.

The following example reestablishes the relationship between the original source SVM, `svm1`, and the original destination SVM, `svm_backup`:

```
cluster_dst::> snapmirror resync -source-path svm1: -destination-path  
svm_backup:
```

10. From the original source SVM or the original source cluster, run the following command to delete the reversed data protection relationship:

```
snapmirror delete -source-path SVM: -destination-path SVM:
```





You must enter a colon (:) after the SVM name in the `-source-path` and `-destination-path` options. See the example below.

The following example deletes the reversed relationship between the original destination SVM, `svm_backup`, and the original source SVM, `svm1`:

```
cluster_src::> snapmirror delete -source-path svm_backup: -destination-path svm1:
```

11. From the original destination SVM or the original destination cluster, release the reversed data protection relationship:

```
snapmirror release -source-path SVM: -destination-path SVM:
```



You must enter a colon (:) after the SVM name in the `-source-path` and `-destination-path` options. See the example below.

The following example releases the reversed relationship between the original destination SVM, `svm_backup`, and the original source SVM, `svm1`:

```
cluster_dst::> snapmirror release -source-path svm_backup: -destination-path svm1:
```

### After you finish

Use the `snapmirror show` command to verify that the SnapMirror relationship was created. For complete command syntax, see the man page.

### Reactivate the original source SVM (FlexGroup volumes only)

You can reestablish the original data protection relationship between the source and destination SVM when you no longer need to serve data from the destination. To reactivate the original source SVM when you are using FlexGroup volumes, you need to perform some additional steps, including deleting the original SVM DR relationship and releasing the original relationship before you reverse the relationship. You also need to release the reversed relationship and recreate the original relationship before stopping scheduled transfers.

### Steps

1. From the original destination SVM or the original destination cluster, delete the original SVM DR relationship:

```
snapmirror delete -source-path SVM: -destination-path SVM:
```



You must enter a colon (:) after the SVM name in the `-source-path` and `-destination-path` options. See the example below.

The following example deletes the original relationship between the original source SVM, `svm1`, and the original destination SVM, `svm_backup`:

```
cluster_dst::> snapmirror delete -source-path svm1: -destination-path  
svm_backup:
```

2. From the original source SVM or the original source cluster, release the original relationship while keeping the Snapshot copies intact:

```
snapmirror release -source-path SVM: -destination-path SVM: -relationship-info  
-only true
```



You must enter a colon (:) after the SVM name in the `-source-path` and `-destination-path` options. See the example below.

The following example releases the original relationship between the original source SVM, `svm1`, and the original destination SVM, `svm_backup`.

```
cluster_src::> snapmirror release -source-path svm1: -destination-path  
svm_backup: -relationship-info-only true
```

3. From the original source SVM or the original source cluster, create a reverse SVM DR relationship using the same configuration, policy, and identity-preserve setting as the original SVM DR relationship:

```
snapmirror create -source-path SVM: -destination-path SVM:
```



You must enter a colon (:) after the SVM name in the `-source-path` and `-destination-path` options. See the example below.

The following example creates a relationship between the SVM from which you are serving data, `svm_backup`, and the original source SVM, `svm1`:

```
cluster_src::> snapmirror create -source-path svm_backup: -destination  
-path svm1:
```

4. From the original source SVM or the original source cluster, run the following command to reverse the data protection relationship:

```
snapmirror resync -source-path SVM: -destination-path SVM:
```



You must enter a colon (:) after the SVM name in the `-source-path` and `-destination-path` options. See the example below.

Although `resync` does not require a baseline transfer, it can be time-consuming. You might want to run the `resync` in off-peak hours.



The command fails if a common Snapshot copy does not exist on the source and destination. Use `snapmirror initialize` to reinitialize the relationship.

The following example reverses the relationship between the original source SVM, `svm1`, and the SVM from which you are serving data, `svm_backup`:

```
cluster_src::> snapmirror resync -source-path svm_backup: -destination
-path svm1:
```

5. When you are ready to reestablish data access to the original source SVM, stop the original destination SVM to disconnect any clients currently connected to the original destination SVM.

```
vserver stop -vserver SVM
```

The following example stops the original destination SVM which is currently serving data:

```
cluster_dst::> vserver stop svm_backup
```

6. Verify that the original destination SVM is in the stopped state by using the `vserver show` command.

```
cluster_dst::> vserver show
```

Vserver	Type	Subtype	Admin State	Operational State	Root Volume
Aggregate					
-----	-----	-----	-----	-----	-----
svm_backup	data	default	stopped	stopped	rv
aggr1					

7. From the original source SVM or the original source cluster, run the following command to perform the final update of the reversed relationship to transfer all changes from the original destination SVM to the original source SVM:

```
snapmirror update -source-path SVM: -destination-path SVM:
```



You must enter a colon (:) after the SVM name in the `-source-path` and `-destination-path` options. See the example below.

The following example updates the relationship between the original destination SVM from which you are serving data, `svm_backup`, and the original source SVM, `svm1`:

```
cluster_src::> snapmirror update -source-path svm_backup: -destination
-path svm1:
```

8. From the original source SVM or the original source cluster, run the following command to stop scheduled transfers for the reversed relationship:

```
snapmirror quiesce -source-path SVM: -destination-path SVM:
```



You must enter a colon (:) after the SVM name in the `-source-path` and `-destination-path` options. See the example below.

The following example stops scheduled transfers between the SVM you are serving data from, `svm_backup`, and the original SVM, `svm1`:

```
cluster_src::> snapmirror quiesce -source-path svm_backup: -destination  
-path svm1:
```

9. When the final update is complete and the relationship indicates "Quiesced" for the relationship status, run the following command from the original source SVM or the original source cluster to break the reversed relationship:

```
snapmirror break -source-path SVM: -destination-path SVM:
```



You must enter a colon (:) after the SVM name in the `-source-path` and `-destination-path` options. See the example below.

The following example breaks the relationship between the original destination SVM from which you were serving data, `svm_backup`, and the original source SVM, `svm1`:

```
cluster_src::> snapmirror break -source-path svm_backup: -destination  
-path svm1:
```

10. If the original source SVM was previously stopped, from the original source cluster, start the original source SVM:

```
vserver start -vserver SVM
```

The following example starts the original source SVM:

```
cluster_src::> vserver start svm1
```

11. From the original source SVM or the original source cluster, delete the reversed SVM DR relationship:

```
snapmirror delete -source-path SVM: -destination-path SVM:
```



You must enter a colon (:) after the SVM name in the `-source-path` and `-destination-path` options. See the example below.

The following example deletes the reversed relationship between the original destination SVM,

svm\_backup, and the original source SVM, svm1:

```
cluster_src:> snapmirror delete -source-path svm_backup: -destination  
-path svm1:
```

12. From the original destination SVM or the original destination cluster, release the reversed relationship while keeping the Snapshot copies intact:

```
snapmirror release -source-path SVM: -destination-path SVM: -relationship-info  
-only true
```



You must enter a colon (:) after the SVM name in the -source-path and -destination -path options. See the example below.

The following example releases the reversed relationship between the original destination SVM, svm\_backup, and the original source SVM, svm1:

```
cluster_dst:> snapmirror release -source-path svm_backup: -destination  
-path svm1: -relationship-info-only true
```

13. From the original destination SVM or the original destination cluster, recreate the original relationship. Use the same configuration, policy, and identity-preserve setting as the original SVM DR relationship:

```
snapmirror create -source-path SVM: -destination-path SVM:
```



You must enter a colon (:) after the SVM name in the -source-path and -destination -path options. See the example below.

The following example creates a relationship between the original source SVM, svm1, and the original destination SVM, svm\_backup:

```
cluster_dst:> snapmirror create -source-path svm1: -destination-path  
svm_backup:
```

14. From the original destination SVM or the original destination cluster, reestablish the original data protection relationship:

```
snapmirror resync -source-path SVM: -destination-path SVM:
```



You must enter a colon (:) after the SVM name in the -source-path and -destination -path options. See the example below.

The following example reestablishes the relationship between the original source SVM, svm1, and the original destination SVM, svm\_backup:

```
cluster_dst:> snapmirror resync -source-path svm1: -destination-path  
svm_backup:
```

## Convert volume replication relationships to an SVM replication relationship

You can convert replication relationships between volumes to a replication relationship between the storage virtual machines (SVMs) that own the volumes, provided that each volume on the source (except the root volume) is being replicated, and each volume on the source (including the root volume) has the same name as the volume on the destination.

### About this task

Use the `volume rename` command when the SnapMirror relationship is idle to rename destination volumes if necessary.

### Steps

1. From the destination SVM or the destination cluster, run the following command to resync the source and destination volumes:

```
snapmirror resync -source-path SVM:volume -destination-path SVM:volume -type  
DP|XDP -schedule schedule -policy policy
```

For complete command syntax, see the man page.



Although `resync` does not require a baseline transfer, it can be time-consuming. You might want to run the `resync` in off-peak hours.

The following example resyncs the relationship between the source volume `volA` on `svm1` and the destination volume `volA` on `svm_backup`:

```
cluster_dst:> snapmirror resync -source-path svm1:volA -destination  
-path svm_backup:volA
```

2. Create an SVM replication relationship between the source and destination SVMs, as described in [Replicating SVM configurations](#).

You must use the `-identity-preserve true` option of the `snapmirror create` command when you create your replication relationship.

3. Stop the destination SVM:

```
vserver stop -vserver SVM
```

For complete command syntax, see the man page.

The following example stops the destination SVM `svm_backup`:

```
cluster_dst:> vserver stop svm_backup
```

4. From the destination SVM or the destination cluster, run the following command to resync the source and destination SVMs:

```
snapmirror resync -source-path SVM: -destination-path SVM: -type DP|XDP  
-schedule schedule -policy policy
```

For complete command syntax, see the man page.



You must enter a colon (:) after the SVM name in the `-source-path` and `-destination-path` options. See the example below.

Although resync does not require a baseline transfer, it can be time-consuming. You might want to run the resync in off-peak hours.

The following example resyncs the relationship between the source SVM `svm1` and the destination SVM `svm_backup`:

```
cluster_dst:> snapmirror resync -source-path svm1: -destination-path  
svm_backup:
```

## Delete an SVM replication relationship

You can use the `snapmirror delete` and `snapmirror release` commands to delete an SVM replication relationship. You can then delete unneeded destination volumes manually.

### About this task

The `snapmirror release` command deletes any SnapMirror-created Snapshot copies from the source. You can use the `-relationship-info-only` option to preserve the Snapshot copies.

For complete command syntax on commands, see the man page.

### Steps

1. Run the following command from the destination SVM or the destination cluster to break the replication relationship:

```
snapmirror break -source-path SVM: -destination-path SVM:
```



You must enter a colon (:) after the SVM name in the `-source-path` and `-destination-path` options. See the example below.

The following example breaks the relationship between the source SVM `svm1` and the destination SVM `svm_backup`:

```
cluster_dst::> snapmirror break -source-path svm1: -destination-path  
svm_backup:
```

2. Run the following command from the destination SVM or the destination cluster to delete the replication relationship:

```
snapmirror delete -source-path SVM: -destination-path SVM:
```



You must enter a colon (:) after the SVM name in the `-source-path` and `-destination-path` options. See the example below.

The following example deletes the relationship between the source SVM `svm1` and the destination SVM `svm_backup`:

```
cluster_dst::> snapmirror delete -source-path svm1: -destination-path  
svm_backup:
```

3. Run the following command from the source cluster or source SVM to release the replication relationship information from the source SVM:

```
snapmirror release -source-path SVM: -destination-path SVM:
```



You must enter a colon (:) after the SVM name in the `-source-path` and `-destination-path` options. See the example below.

The following example releases information for the specified replication relationship from the source SVM `svm1`:

```
cluster_src::> snapmirror release -source-path svm1: -destination-path  
svm_backup:
```

## Manage SnapMirror root volume replication

### Manage SnapMirror root volume replication overview

Every SVM in a NAS environment has a unique namespace. The SVM *root volume*, containing operating system and related information, is the entry point to the namespace hierarchy. To ensure that data remains accessible to clients in the event of a node outage or failover, you should create a load-sharing mirror copy of the SVM root volume.

The main purpose of load-sharing mirrors for SVM root volumes is no longer for load sharing; instead, their purpose is for disaster recovery.

- If the root volume is temporarily unavailable, the load-sharing mirror automatically provides read-only access to root volume data.



- If the root volume is permanently unavailable, you can promote one of the load-sharing volumes to provide write access to root volume data.

## Create and initializing load-sharing mirror relationships

You should create a load-sharing mirror (LSM) for each SVM root volume that serves NAS data in the cluster. You can create the LSM on any node other than the one containing the root volume, such as the partner node in an HA pair, or preferably in a different HA pair. For a two-node cluster, you should create the LSM on the partner of the node with the SVM root volume.

### About this task

If you create an LSM on the same node, and the node is unavailable, you have a single point of failure, and you do not have a second copy to ensure the data remains accessible to clients. But when you create the LSM on a node other than the one containing the root volume, or on a different HA pair, your data is still accessible in the event of an outage.

For example, in a four-node cluster with a root volume on three nodes:

- For the root volume on HA 1 node 1, create the LSM on HA 2 node 1 or HA 2 node 2.
- For the root volume on HA 1 node 2, create the LSM on HA 2 node 1 or HA 2 node 2.
- For the root volume on HA 2 node 1, create the LSM on HA 1 node 1 or HA 1 node 2.

### Steps

1. Create a destination volume for the LSM:

```
volume create -vserver SVM -volume volume -aggregate aggregate -type DP -size size
```

The destination volume should be the same or greater in size than the root volume.

It is a best practice to name the root and destination volume with suffixes, such as `_root` and `_m1`.

For complete command syntax, see the man page.

The following example creates a load-sharing mirror volume for the root volume `svm1_root` in `cluster_src`:

```
cluster_src:> volume create -vserver svm1 -volume svm1_m1 -aggregate aggr_1 -size 1gb -state online -type DP
```

2. Create a replication job schedule, as described in [Creating a replication job schedule](#).
3. Create a load-sharing mirror relationship between the SVM root volume and the destination volume for the LSM:

```
snapmirror create -source-path SVM:volume|cluster://SVM/volume -destination -path SVM:volume|cluster://SVM/volume -type LS -schedule schedule
```

For complete command syntax, see the man page.

The following example creates a load-sharing mirror relationship between the root volume `svm1_root` and the load-sharing mirror volume `svm1_m1`:

```
cluster_src:> snapmirror create -source-path svm1:svm1_root  
-destination-path svm1:svm1_m1 -type LS -schedule hourly
```

The type attribute of the load-sharing mirror changes from `DP` to `LS`.

#### 4. Initialize the load-sharing mirror:

```
snapmirror initialize-ls-set -source-path SVM:volume|cluster://SVM/volume
```

Initialization can be time-consuming. You might want to run the baseline transfer in off-peak hours.

For complete command syntax, see the man page.

The following example initializes the load-sharing mirror for the root volume `svm1_root`:

```
cluster_src:> snapmirror initialize-ls-set -source-path svm1:svm1_root
```

### Update a load-sharing mirror relationship

Load-sharing mirror (LSM) relationships are updated automatically for SVM root volumes after a volume in the SVM is mounted or unmounted, and during `volume create` operations that include the ``junction-path`` option. You can manually update a LSM relationship if you want it updated before the next scheduled update.

Load-sharing mirror relationships update automatically in the following circumstances:

- It's time for a scheduled update
- A mount or unmount operation is performed on a volume in the SVM root volume
- A `volume create` command is issued that includes the `junction-path` option

#### Step

##### 1. Update a load-sharing mirror relationship manually:

```
snapmirror update-ls-set -source-path SVM:volume|cluster://SVM/volume
```

The following example updates the load-sharing mirror relationship for the root volume `svm1_root`:

```
cluster_src:> snapmirror update-ls-set -source-path svm1:svm1_root
```

### Promote a load-sharing mirror

If a root volume is permanently unavailable, you can promote the load-sharing mirror

(LSM) volume to provide write access to root volume data.

### What you'll need

You must use advanced privilege level commands for this task.

### Steps

1. Change to advanced privilege level:

```
set -privilege advanced
```

2. Promote an LSM volume:

```
snapmirror promote -destination-path SVM:volume|cluster://SVM/volume
```

For complete command syntax, see the man page.

The following example promotes the volume `svm1_m2` as the new SVM root volume:

```
cluster_src::*> snapmirror promote -destination-path svm1:svm1_m2

Warning: Promote will delete the offline read-write volume
cluster_src://svm1/svm1_root and replace it with
cluster_src://svm1/svm1_m2. Because the volume is offline,
it is not possible to determine whether this promote will
affect other relationships associated with this source.
Do you want to continue? {y|n}: y
```

Enter `y`. ONTAP makes the LSM volume a read/write volume, and deletes the original root volume if it is accessible.



The promoted root volume might not have all of the data that was in the original root volume if the last update did not occur recently.

3. Return to admin privilege level:

```
set -privilege admin
```

4. Rename the promoted volume following the naming convention you used for the root volume:

```
volume rename -vserver SVM -volume volume -newname new_name
```

The following example renames the promoted volume `svm1_m2` with the name `svm1_root`:

```
cluster_src::> volume rename -vserver svm11 -volume svm1_m2 -newname
svm1_root
```

5. Protect the renamed root volume, as described in step 3 through step 4 in [Creating and initializing load-sharing mirror relationships](#).

# SnapMirror technical details

## Use path name pattern matching

You can use pattern matching to specify the source and destination paths in `snapmirror` commands.

`snapmirror` commands use fully qualified path names in the following format: `vserver:volume`. You can abbreviate the path name by not entering the SVM name. If you do this, the `snapmirror` command assumes the local SVM context of the user.

Assuming that the SVM is called “vserver1” and the volume is called “vol1”, the fully qualified path name is `vserver1:vol1`.

You can use the asterisk (\*) in paths as a wildcard to select matching, fully qualified path names. The following table provides examples of using the wildcard to select a range of volumes.

*	Matches all paths.
vs*	Matches all SVMs and volumes with SVM names beginning with <code>vs</code> .
:*src	Matches all SVMs with volume names containing the <code>src</code> text.
:vol	Matches all SVMs with volume names beginning with <code>vol</code> .

```
vs1::> snapmirror show -destination-path *:*dest*

Progress
Source          Destination  Mirror          Relationship  Total
Last
Path            Type   Path            State          Status          Progress
Healthy Updated
-----
vs1:sm_src2
DP   vs2:sm_dest1
Snapmirrored  Idle
true  -
```

## Use extended queries to act on many SnapMirror relationships

You can use *extended queries* to perform SnapMirror operations on many SnapMirror relationships at one time. For example, you might have multiple uninitialized SnapMirror

relationships that you want to initialize using one command.

### About this task

You can apply extended queries to the following SnapMirror operations:

- Initializing uninitialized relationships
- Resuming quiesced relationships
- Resynchronizing broken relationships
- Updating idle relationships
- Aborting relationship data transfers

### Step

1. Perform a SnapMirror operation on many relationships:

```
snapmirror command {-state state } *
```

The following command initializes SnapMirror relationships that are in an Uninitialized state:

```
vs1::> snapmirror initialize {-state Uninitialized} *
```

### Ensure a common Snapshot copy in a mirror-vault deployment

You can use the `snapmirror snapshot-owner create` command to preserve a labeled Snapshot copy on the secondary in a mirror-vault deployment. Doing so ensures that a common Snapshot copy exists for the update of the vault relationship.

### About this task

If you use a combination mirror-vault fan-out or cascade deployment, you should keep in mind that updates will fail if a common Snapshot copy does not exist on the source and destination volumes.

This is never an issue for the mirror relationship in a mirror-vault fan-out or cascade deployment, since SnapMirror always creates a Snapshot copy of the source volume before it performs the update.

It might be an issue for the vault relationship, however, since SnapMirror does not create a Snapshot copy of the source volume when it updates a vault relationship. You need to use the `snapmirror snapshot-owner create` to ensure that there is at least one common Snapshot copy on both the source and destination of the vault relationship.

### Steps

1. On the source volume, assign an owner to the labeled Snapshot copy you want to preserve:

```
snapmirror snapshot-owner create -vserver SVM -volume volume -snapshot  
snapshot -owner owner
```

The following example assigns ApplicationA as the owner of the snap1 Snapshot copy:

```
clust1::> snapmirror snapshot-owner create -vserver vs1 -volume vol1  
-snapshot snap1 -owner ApplicationA
```

2. Update the mirror relationship, as described in [Updating a replication relationship manually](#).

Alternatively, you can wait for the scheduled update of the mirror relationship.

3. Transfer the labeled Snapshot copy to the vault destination:

```
snapmirror update -source-path SVM:volume|cluster://SVM/volume, ... -destination  
-path SVM:volume|cluster://SVM/volume, ... -source-snapshot snapshot
```

For complete command syntax, see the man page.

**The following example transfers the `snap1` Snapshot copy**

```
clust1::> snapmirror update -vserver vs1 -volume vol1  
-source-snapshot snap1
```

The labeled Snapshot copy will be preserved when the vault relationship is updated.

4. On the source volume, remove the owner from the labeled Snapshot copy:

```
snapmirror snapshot-owner delete -vserver SVM -volume volume -snapshot  
snapshot -owner owner
```

The following examples removes `ApplicationA` as the owner of the `snap1` Snapshot copy:

```
clust1::> snapmirror snapshot-owner delete -vserver vs1 -volume vol1  
-snapshot snap1 -owner ApplicationA
```

## Compatible ONTAP versions for SnapMirror relationships

You should verify that the source and destination volumes are running compatible ONTAP versions before creating a SnapMirror data protection relationship.



Version-independence is not supported for SVM replication.

## Unified replication relationships

For SnapMirror relationships of type “XDP”, using on premises or Cloud Volumes ONTAP releases:

Beginning with ONTAP 9.9.0:



- ONTAP 9.x.0 releases are cloud-only releases and support Cloud Volumes ONTAP (CVO) systems. The asterisk (\*) after the release version indicates a cloud-only release.
- ONTAP 9.x.1 releases are general releases and support both on-premises and CVO systems.



Locate the higher, more recent ONTAP version in the left column, and in the top row locate the lower ONTAP version to determine interoperability. Interoperability is bidirectional.

### Interoperability for ONTAP version 9.3 and later

ONTAP version...															
	9.13.0*	9.12.1	9.12.0*	9.11.1	9.11.0*	9.10.1	9.10.0*	9.9.1	9.9.0*	9.8	9.7	9.6	9.5	9.4	9.3
9.13.0*	Yes	Yes	No	Yes	No	Yes	No	Yes	No	Yes	No	No	No	No	No
9.12.1	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	No
9.12.0*	No	Yes	Yes	Yes	No	Yes	No	Yes	No	Yes	Yes	No	No	No	No
9.11.1	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
9.11.0*	No	Yes	No	Yes	Yes	Yes	No	Yes	No	Yes	Yes	Yes	No	No	No
9.10.1	Yes	Yes	Yes	n/a	n/a	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
9.10.0*	No	Yes	No	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No	No
9.9.1	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
9.9.0*	No	Yes	No	Yes	No	Yes	n/a	Yes	Yes	Yes	Yes	Yes	Yes	No	No
9.8	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
9.7	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
9.6	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
9.5	No	No	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
9.4	No	No	No	No	No	No	No	No	No	No	No	No	Yes	Yes	Yes
9.3	No	No	No	No	No	No	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes

## SnapMirror Synchronous relationships



SnapMirror Synchronous is not supported for ONTAP cloud instances.

ONTAP version...	Interoperates with these previous ONTAP versions...							
	9.12.1	9.11.1	9.10.1	9.9.1	9.8	9.7	9.6	9.5
9.12.1	Yes	Yes	Yes	Yes	Yes	Yes	No	No
9.11.1	Yes	Yes	Yes	Yes	No	No	No	No
9.10.1	Yes	Yes	Yes	Yes	Yes	No	No	No
9.9.1	Yes	Yes	Yes	Yes	Yes	Yes	No	No
9.8	Yes	No	Yes	Yes	Yes	Yes	Yes	No
9.7	Yes	No	No	Yes	Yes	Yes	Yes	Yes
9.6	No	No	No	No	Yes	Yes	Yes	Yes
9.5	No	No	No	No	No	Yes	Yes	Yes

## SnapMirror DR relationships

For SnapMirror relationships of type “DP” and policy type “async-mirror”:



DP-type mirrors cannot be initialized beginning with ONTAP 9.11.1 and are completely deprecated in ONTAP 9.12.1. For more information, see [Deprecation of data protection SnapMirror relationships](#).



In the following table, the column on the left indicates the ONTAP version on the source volume, and the top row indicates the ONTAP versions you can have on your destination volume.

Source	Destination											
	9.11.1	9.10.1	9.9.1	9.8	9.7	9.6	9.5	9.4	9.3	9.2	9.1	9.0
9.11.1	Yes	No	No	No	No	No	No	No	No	No	No	No
9.10.1	Yes	Yes	No	No	No	No	No	No	No	No	No	No
9.9.1	Yes	Yes	Yes	No	No	No	No	No	No	No	No	No
9.8	No	Yes	Yes	Yes	No	No	No	No	No	No	No	No
9.7	No	No	Yes	Yes	Yes	No	No	No	No	No	No	No
9.6	No	No	No	Yes	Yes	Yes	No	No	No	No	No	No
9.5	No	No	No	No	Yes	Yes	Yes	No	No	No	No	No
9.4	No	No	No	No	No	Yes	Yes	Yes	No	No	No	No
9.3	No	No	No	No	No	No	Yes	Yes	Yes	No	No	No
9.2	No	No	No	No	No	No	No	Yes	Yes	Yes	No	No



9.1	No	No	No	No	No	No	No	No	No	Yes	Yes	Yes	No
9.0	No	No	No	No	No	No	No	No	No	No	Yes	Yes	Yes



Interoperability is not bidirectional.

#### SnapMirror SVM DR relationships

- For SVM DR data and SVM protection:

SVM DR is only supported between clusters running the same version of ONTAP.

- For SVM DR for SVM migration:
  - Replication is supported in a single direction from an earlier version of ONTAP to a later version of ONTAP; for example, from ONTAP 9.11.1 to ONTAP 9.12.1.
  - The ONTAP version on the target cluster must be no more than 2 versions newer, as shown in the table below.
  - Replication is not supported for long-term data protection use cases.

The asterisk (\*) after the release version indicates a cloud-only release.

Source	Destination														
	9.3	9.4	9.5	9.6	9.7	9.8	9.9.0*	9.9.1	9.10.0*	9.10.1	9.11.0*	9.11.1	9.12.0*	9.12.1	9.13.0*
9.3	Yes	Yes	Yes												
9.4		Yes	Yes	Yes											
9.5			Yes	Yes	Yes										
9.6				Yes	Yes	Yes									
9.7					Yes	Yes	Yes								
9.8						Yes	Yes	Yes							
9.9.0							Yes	Yes	Yes						
9.9.1								Yes	Yes	Yes					
9.10.0									Yes	Yes	Yes				
9.10.1										Yes	Yes	Yes			
9.11.0											Yes	Yes	Yes		
9.11.1												Yes	Yes	Yes	
9.12.0													Yes	Yes	Yes

9.12.1														Yes	Yes
9.13.0															Yes

h| SnapLock mode h| Protection level h| WORM file deleting during retention

a|  
Compliance mode  
a|  
At the file level  
a|  
Cannot be deleted

a|  
Enterprise mode  
a|  
At the disk level  
a|  
Can be deleted by the compliance administrator using an audited “privileged delete” procedure

After the retention period has elapsed, you are responsible for deleting any files you no longer need. Once a file has been committed to WORM, whether under Compliance or Enterprise mode, it cannot be modified, even after the retention period has expired.

You cannot move a WORM file during or after the retention period. You can copy a WORM file, but the copy will not retain its WORM characteristics.

The following table shows the differences in capabilities supported by SnapLock Compliance and Enterprise modes:

h| Capability h| SnapLock Compliance h| SnapLock Enterprise

a|  
Enable and delete files using privileged delete  
a|  
No  
a|  
Yes  
a|  
Reinitialize disks  
a|  
No  
a|  
Yes  
a|  
Destroy SnapLock aggregates and volumes during retention period  
a|  
No  
a|  
Yes, with the exception of the SnapLock audit log volume  
a|  
Rename aggregates or volumes

a|  
No  
a|  
Yes  
a|  
Use non-NetApp disks

a|  
No  
a|  
Yes (with [FlexArray Virtualization](#))  
a|  
Use the SnapLock volume for audit logging  
a|  
Yes  
a|  
Yes, beginning with ONTAP 9.5

== Supported and unsupported features with SnapLock

The following table shows the features that are supported with SnapLock Compliance mode, SnapLock Enterprise mode, or both:

h| Feature h| Supported with SnapLock Compliance h| Supported with SnapLock Enterprise

a|  
Consistency Groups  
a|  
No  
a|  
No

a|  
Encrypted volumes  
a|  
Yes, beginning with ONTAP 9.2. Learn more about [Encryption and SnapLock](#).  
a|  
Yes, beginning with ONTAP 9.2. Learn more about [Encryption and SnapLock](#).

a|  
FabricPools on SnapLock aggregates  
a|  
No  
a|  
Yes, beginning with ONTAP 9.8. Learn more about [FabricPool on SnapLock Enterprise aggregates](#).  
a|  
Flash Pool aggregates  
a|  
Yes, beginning with ONTAP 9.1.  
a|  
Yes, beginning with ONTAP 9.1.

a|  
FlexClone

a|  
You can clone SnapLock volumes, but you cannot clone files on a SnapLock volume.

a|  
You can clone SnapLock volumes, but you cannot clone files on a SnapLock volume.

a|  
FlexGroup volumes

a|  
Yes, beginning with ONTAP 9.11.1. Learn more about [SnapLock support on FlexGroup volumes](#).

a|  
Yes, beginning with ONTAP 9.11.1. Learn more about [SnapLock support on FlexGroup volumes](#).

a|  
LUNs

a|  
No

a|  
No

a|  
MetroCluster configurations

a|  
Yes, beginning with ONTAP 9.3. Learn more about [MetroCluster support](#).

a|  
Yes, beginning with ONTAP 9.3. Learn more about [MetroCluster support](#).

a|  
SAN

a|  
No

a|  
No

a|  
Single-file SnapRestore

a|  
No

a|  
Yes

a|  
SnapMirror Business Continuity

a|  
No

a|  
No

a|  
SnapRestore

a|  
No

a|  
Yes

a|  
SMTape

a|  
No  
a|  
No

a|  
SnapMirror Synchronous

a|  
No  
a|  
No

a|  
SSDs

a|  
Yes, beginning with ONTAP 9.1.  
a|  
Yes, beginning with ONTAP 9.1.

a|  
Storage efficiency features

a|  
Yes, beginning with ONTAP 9.9.1. Learn more about [storage efficiency support](#).  
a|  
Yes, beginning with ONTAP 9.9.1. Learn more about [storage efficiency support](#).

<p>== FabricPool on SnapLock Enterprise aggregates</p> <p>FabricPools are supported on SnapLock Enterprise aggregates beginning with ONTAP 9.8. However, your account team needs to open a product variance request documenting that you understand that FabricPool data tiered to a public or private cloud is no longer protected by SnapLock because a cloud admin can delete that data. [NOTE] ====</p> <p>Any data that FabricPool tiers to a public or private cloud is no longer protected by SnapLock because that data can be deleted by a cloud administrator. ====</p> <p>== FlexGroup volumes</p> <p>SnapLock supports FlexGroup volumes beginning with ONTAP 9.11.1; however, the following features are not supported:</p> <ul style="list-style-type: none"> <li>* Legal-hold</li> <li>* Event-based retention</li> <li>* SnapLock for SnapVault (supported beginning with ONTAP 9.12.1)</li> </ul> <p>You should also be aware of the following behaviors:</p> <ul style="list-style-type: none"> <li>* The volume compliance clock (VCC) of a FlexGroup volume is determined by the VCC of the root constituent. All non-root constituents will have their VCC closely</li> </ul>	<p>n): y ----</p> <p>. Repeat this procedure for each node that hosts a SnapLock aggregate. —====</p> <p>== Enable ComplianceClock resynchronization for an NTP-configured system</p> <p>You can enable the SnapLock ComplianceClock time synchronization feature when an NTP server is configured.</p> <p>.What you'll need</p> <ul style="list-style-type: none"> <li>* This feature is available only at the advanced privilege level.</li> <li>* You must be a cluster administrator to perform this task.</li> <li>* The SnapLock license must be installed on the node.</li> <li>* This feature is available only for Cloud Volumes ONTAP, ONTAP Select, and VSIM platforms.</li> </ul> <p>.About this task</p> <p>When the SnapLock secure clock daemon detects a skew beyond the threshold, ONTAP uses the system time to reset both the system and volume ComplianceClocks. A period of 24 hours is set as the skew threshold. This means that the system ComplianceClock is synchronized to the system clock only if the skew is more than a day</p>	<p>enterprise*` + The man page for the command contains a complete list of options. + The following command creates a SnapLock Compliance aggregate named aggr1 with three disks on node1: + ---- cluster1::&gt; storage aggregate create -aggregate aggr1 -node node1 -diskcount 3 -snaplock-type compliance ----</p> <p>= Create and mount SnapLock volumes :icons: font :relative_path: ./snaplock/ :imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/ [.lead] You must create a SnapLock volume for the files or Snapshot copies that you want to commit to the WORM state. Beginning with ONTAP 9.10.1, any volume you create, regardless of the aggregate type, is created by default as a non-SnapLock volume. You must use the <code>-snaplock -type</code> option to explicitly create a SnapLock volume by specifying either Compliance or Enterprise as the SnapLock type. By default, the SnapLock</p>	<p>enterprise` + For a complete list of options, see the man page for the command. The following options are not available for SnapLock volumes: <code>-nvfail</code>, <code>-atime-update</code>, <code>-is</code>, <code>-autobalance</code>, <code>-eligible</code>, <code>-space</code>, <code>-mgmt-try-first</code>, and <code>vmalign</code>. + The following command creates a SnapLock Compliance volume named vol1 on aggr1 on vs1: + ---- cluster1::&gt; volume create -vserver vs1 -volume vol1 -aggregate aggr1 -snaplock-type compliance ---- —====</p> <p>== Mount a SnapLock volume</p> <p>You can mount a SnapLock volume to a junction path in the SVM namespace for NAS client access.</p> <p>.What you'll need</p> <p>The SnapLock volume must be online.</p> <p>.About this task</p> <ul style="list-style-type: none"> <li>* You can mount a SnapLock volume only under the root of the SVM.</li> <li>* You cannot mount a regular volume under a SnapLock volume.</li> </ul> <p>.Steps</p>
--	--	--	---

| Value| Unit| Notes

a|

0 - 65535

a|

seconds

a|

a|

0 - 24

a|

hours

a|

a|

0 - 365

a|

days

a|

a|

0 - 12

a|

months

a|

a|

0 - 100

a|

years

a|

Beginning with ONTAP 9.10.1. For earlier ONTAP releases, the value is 0 - 70.

a|

max

a|

-

a|

Use the maximum retention period.

a|

min

a|

-

a|

Use the minimum retention period.

a|

infinite

a|

-

a|

Retain the files forever.

a|

unspecified

a|

-

a|

Retain the files until an absolute retention period is set.



The values and ranges for the maximum and minimum retention periods are identical, except for `max` and `min`, which are not applicable. For more information about this task, see [Set the retention time overview](#).

You can use the `volume snaplock show` command to view the retention period settings for the volume. For more information, see the man page for the command.

[NOTE]

$$\frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} f(x) e^{-x^2} dx = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} f(x) e^{-x^2} dx$$

After a file has been committed to the WORM state, you can extend but not shorten the retention period.

=====	=====	=====	=====
=====	=====	=====	=====

## .Steps

. Set the default retention period for files on a SnapLock volume:

+

```
volume snaplock modify -vserver SVM_name -volume volume_name -default-retention
-period default_retention_period -minimum-retention-period min_retention_period
-maximum-retention-period max_retention_period
```

+

For a complete list of options, see the man page for the command.

+

[NOTE]

=====

The following examples assume that the minimum and maximum retention periods have not been modified previously.

_____	_____	_____	_____
_____	_____	_____	_____

+

The following command sets the default retention period for a Compliance or Enterprise volume to 20 days:

+

\_\_\_\_\_

```
cluster1::> volume snaplock modify -vserver vs1 -volume vol1 -default-retention-period 20days
```

— — — —

+

The following command sets the default retention period for a Compliance volume to 70 years:

+

■■■■

```
cluster1::> volume snaplock modify -vserver vs1 -volume vol1 -maximum-retention-period 70years
```

— — —

+

The following command sets the default retention period for an Enterprise volume to 10 years:

+

\_\_\_\_\_

```
cluster1::> volume snaplock modify -vserver vs1 -volume vol1 -default-retention-period max -maximum-retention-period 10years
```

■■■■

+

The following commands set the default retention period for an Enterprise volume to 10 days:

+

■■■■

```
cluster1::> volume snaplock modify -vserver vs1 -volume vol1 -minimum-retention-period 10days
```

```
cluster1::> volume snaplock modify -vserver vs1 -volume vol1 -default-retention-period min
```

■■■■

+

The following command sets the default retention period for a Compliance volume to infinite:

h|

***Using EBR to extend the retention period of already existing WORM files***

a|

EBR is convenient when you want to extend the retention period of already existing WORM files. For example, it might be your firm's policy to retain employee W-4 records in unmodified form for three years after the employee changes a withholding election. Another company policy might require that W-4 records be retained for five years after the employee is terminated.

In this situation, you could create an EBR policy with a five-year retention period. After the employee is terminated (the "event"), you would apply the EBR policy to the employee's W-4 record, causing its retention period to be extended. That will usually be easier than extending the retention period manually, particularly when a large number of files is involved.

## .Steps

. Create an EBR policy:

+

```
snaplock event-retention policy create -vserver SVM_name -name policy_name  
-retention-period retention_period
```

+

The following command creates the EBR policy `employee_exit` on `vs1` with a retention period of ten years:

+

----

```
cluster1::>snaplock event-retention policy create -vserver vs1 -name employee_exit -retention-period 10years
```

----

. Apply an EBR policy:

+

```
snaplock event-retention apply -vserver SVM_name -name policy_name -volume  
volume_name -path path_name
```

+

The following command applies the EBR policy `employee_exit` on `vs1` to all the files in the directory `d1`:

+

----

```
cluster1::>snaplock event-retention apply -vserver vs1 -name employee_exit -volume vol1 -path /d1
```

----

= Create an audit log

:icons: font

:relative\_path: ./snaplock/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

You must create a SnapLock-protected audit log before performing a privileged delete or SnapLock volume move. The audit log records the creation and deletion of SnapLock administrator accounts, modifications to the log volume, whether privileged delete is enabled, privileged delete operations, and SnapLock volume move operations.

.What you'll need

You must be a cluster administrator to create a SnapLock aggregate.

.About this task

You cannot delete an audit log until the log file retention period has elapsed. You cannot modify an audit log even after the retention period has elapsed. This is true for both SnapLock Compliance and Enterprise modes.

[NOTE]

=====

In ONTAP 9.4 and earlier, you cannot use a SnapLock Enterprise volume for audit logging. You must use a SnapLock Compliance volume. In ONTAP 9.5 and later, you can use either a SnapLock Enterprise volume or a SnapLock Compliance volume for audit logging. In all cases, the audit log volume must be mounted at the

	Value	Unit	Notes
a			
	none		
a			
-			
a			
	The default.		
a			
	5 - 5256000		
a			
	minutes		
a			
-			
a			
	1 - 87600		
a			
	hours		
a			
-			
a			
	1 - 3650		
a			
	days		
a			
-			
a			
	1 - 120		
a			
	months		
a			
-			
a			
	1 - 10		
a			
	years		
a			
-			

<p>[NOTE]</p> <p>====</p> <p>The minimum value is 5 minutes and the maximum value is 10 years.</p> <p>====</p> <p>.Steps</p> <p>.</p> <p>Autocommit files on a SnapLock volume to WORM:</p> <p>+</p> <p><b>volume</b></p> <p><b>snaplock</b></p> <p><b>modify</b></p> <p><b>-vserver</b></p> <p><b>SVM_name</b></p> <p><b>-volume</b></p> <p><b>volume_name</b></p> <p><b>-autocommit-period</b></p> <p><b>autocommit_period</b></p> <p>+</p> <p>For a complete list of options, see the man page for the command.</p> <p>+</p> <p>The following command autocommits the files on volume vol1 of SVM vs1, as long as the files remain unchanged for 5 hours:</p> <p>+</p>	<p>false*</p> <p>+</p> <p>For a complete list of options, see the man page for the command.</p> <p>+</p> <p>The following command enables VAM on volume vol1 of SVM vs1:</p> <p>+</p> <p>----</p> <p>cluster1::&gt;volume snaplock modify -vserver vs1 -volume vol1 -is -volume -append -mode -enabled true</p> <p>----</p> <p>. Use a suitable command or program to create files with write permissions.</p> <p>+</p> <p>The files are WORM-appendable by default.</p> <p></p> <p>= Commit Snapshot copies to WORM on a</p>	<p>enterprise</p> <p>-type DP</p> <p>-size size*</p> <p>+</p> <p>[NOTE]</p> <p>====</p> <p>Beginning with ONTAP 9.10.1, SnapLock and non-SnapLock volumes can exist on the same aggregate; therefore, you are no longer required to create a separate SnapLock aggregate if you are using ONTAP 9.10.1. You use the volume -snaplock -type option to specify a Compliance or Enterprise SnapLock volume type. In ONTAP releases earlier than ONTAP 9.10.1, the SnapLock mode, Compliance or Enterprise, is inherited from the aggregate. Version-flexible destination</p>	<p>enterprise</p> <p>-type DP</p> <p>-size size*</p> <p>+</p> <p>[NOTE]</p> <p>Beginning with ONTAP 9.10.1, SnapLock and non-SnapLock volumes can exist on the same aggregate; therefore, you are no longer required to create a separate SnapLock aggregate if you are using ONTAP 9.10.1. You use the volume -snaplock -type option to specify a Compliance or Enterprise SnapLock volume type. In ONTAP releases earlier than ONTAP 9.10.1, the SnapLock mode—Compliance or Enterprise—is inherited from the aggregate. Version-flexible destination volumes are</p>	<p>DP -policy</p> <p><i>policy_name</i></p> <p>-schedule</p> <p><i>schedule_name</i>*</p> <p>+</p> <p>The following command creates a SnapMirror relationship between the source volume srcvolA on SVM1 and the destination volume dstvolB on SVM2, and assigns the policy SVM1-mirror and the schedule weekendcron:</p> <p>+</p> <p>----</p> <p>SVM2::&gt; snapmirror create -source-path SVM1:srcvolA -destination-path SVM2:dstvolB -type XDP -policy SVM1-mirror -schedule weekendcron</p> <p>----</p> <p>+</p> <p>[NOTE]</p> <p>The XDP type is available in</p>	<p>enabled</p>	<p>permanently</p> <p>-disabled*</p> <p>+</p> <p>The following command enables the privileged delete feature for the Enterprise volume dataVol on SVM1:</p> <p>+</p> <p>----</p> <p>SVM1::&gt; volume snaplock modify -vserver SVM1 -volume dataVol -privileged -delete enabled</p> <p>----</p> <p>== Delete Enterprise-mode WORM files</p> <p></p> <p>You can use the privileged delete feature to delete Enterprise-mode WORM files during the retention period.</p> <p>.What you'll need</p> <p>* You must be a SnapLock administrator</p>	<p>no}: y</p> <p>[Job 32] Job succeeded: Successful</p> <p>----</p> <p>== Enable Snapshot copy locking on an existing volume</p> <p>.</p> <p>To modify an existing volume to enable Snapshot copy locking, enter the following command:</p> <p>+</p> <p>volume modify -vserver <i>vserver_name</i> -volume <i>volume_name</i> -snapshot -locking -enabled true</p> <p>== Create a locked Snapshot copy</p> <p></p> <p>Beginning with ONTAP 9.12.1, you can create Snapshot copy policies to apply a Snapshot copy retention period and</p>
---	---	---	---	---	----------------	---	--

h	Limit h	Scope h	Minimum h	Maximum
	Number of consistency groups			
	Cluster			
	0			
	Same as maximum volume count in cluster			
	Number of parent consistency groups			
	Cluster			
	0			
	Same as maximum volume count in cluster			
	Number of individual and parent consistency groups			
	Cluster			
	0			
	Same as maximum volume count in cluster			
	Consistency group	Same as maximum volume count in cluster		
	1			
	80			
	Number of volumes in the child of a parent consistency group			
	Parent consistency group			
	1			
	80			
	Number of volumes in a child consistency group			
	Child consistency group			
	1			
	80			
	Number of child consistency groups in a parent consistency group			
	Parent consistency group			
	1			
	5			

= Configure a single consistency group

:icons: font

:relative\_path: ./consistency-groups/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

Consistency groups can be created with existing volumes or new LUNs. Once a volume is associated with a consistency group, it cannot be added with another consistency group.

Beginning in ONTAP 9.12.1, you can modify the member volumes of a consistency group. For more information on this process, refer to [Modify a consistency group](#). In ONTAP 9.10.1 through 9.11.1, modifying the member volumes of a consistency group is not supported.

== Create a consistency group with new LUNs

Creating a consistency group with new LUNs allows you to create and configure new LUNs in a new consistency group and set protection policies.

.Steps

- . Select **Storage > Consistency groups**.
- . Select **+Add** then **Using New LUNs**.
- . Name the consistency group. Designate the number of LUNs and capacity per LUN.
- .. **Application Type**: If you are using ONTAP 9.12.1 or later, select an application type. If no value is selected, the consistency group will be assigned the type of **Other** by default. Learn more about tagging consistency in [Application and component tags](#). If you plan to create an SM-BC relationship, you must use **Other**.
- . Select the host operating system and LUN format. Enter the host initiator information.
- . To configure protection policies, add a child consistency group, or show more options about host initiators, select **More options**.
- . Select **Save**.
- . Confirm your consistency group has been created by returning to the main consistency group menu where it will appear once the job completes. If you set a protection policy, you will know it has been applied when you see a green shield under look under the appropriate policy, remote or local.

== Create a consistency group with existing volumes

You can use existing volumes to create a consistency group.

.Steps

- . Select **Storage > Consistency groups**.
- . Select **+Add** then **Using existing volumes**.
- . Name the consistency group and select the storage VM.
- .. **Application Type**: If you are using ONTAP 9.12.1 or later, select an application type. If no value is selected, the consistency group will be assigned the type of **Other** by default. Learn more about tagging consistency in [Application and component tags](#). If the consistency group has an SM-BC relationship, you must use **Other**.
- . Select the existing volumes to include. Only volumes that are not already part of a consistency group will be available for selection.

+

[NOTE]

If creating a consistency group with existing volumes, the consistency group supports FlexVol volumes. Volumes with Asynchronous or Synchronous SnapMirror relationships can be added to consistency groups, but they are not consistency group-aware. Consistency groups do not support S3 buckets, MetroCluster, or storage VMs with SVMMDR relationships.

+

h| Mediator version h| Supported Linux versions

a| 1.5

a|

\* Red Hat Enterprise Linux: 7.6, 7.7, 7.8, 7.9, 8.1, 8.2, 8.3, 8.4, 8.5

\* CentOS: 7.6, 7.7, 7.8, 7.9

a| 1.4

a|

\* Red Hat Enterprise Linux: 7.6, 7.7, 7.8, 7.9, 8.1, 8.2, 8.3, 8.4, 8.5

\* CentOS: 7.6, 7.7, 7.8, 7.9

a| 1.3

a|

\* Red Hat Enterprise Linux: 7.6, 7.7, 7.8, 7.9, 8.1, 8.2, 8.3

\* CentOS: 7.6, 7.7, 7.8, 7.9

a| 1.2

a|

\* Red Hat Enterprise Linux: 7.6, 7.7, 7.8, 8.1

\* CentOS: 7.6, 7.7, 7.8



## == Licensing

- \* SnapMirror synchronous (SM-S) license must be applied on both clusters
- \* SnapMirror license must be applied on both clusters

+

### [NOTE]

If your ONTAP storage systems were purchased before June 2019, click [NetApp ONTAP Master License Keys](#) to get the required SM-S license.

## == Networking environment

- \* Inter-cluster latency round trip time (RTT) must be less than 10 milliseconds
- \* SCSI-3 persistent reservations are **not** supported with SM-BC

## == Supported protocols

- \* Only SAN protocols are supported (not NFS/SMB)
- \* Only Fibre Channel and iSCSI protocols are supported
- \* The default IPspace is required by SM-BC for cluster peer relationships. Custom IPspace is not supported.

## == NTFS Security Style

NTFS security style is **not** supported on SM-BC volumes.

## == ONTAP Mediator

- \* Must be provisioned externally and attached to ONTAP for transparent application failover.
- \* For more information about the ONTAP Mediator, see [Prepare to install the ONTAP Mediator service](#).

## == Read-write destination volumes

\* SM-BC relationships are not supported on read-write destination volumes. Before you can use a read-write volume, you must convert it to a DP volume by creating a volume-level SnapMirror relationship and then deleting the relationship. For details, see [Converting existing relationships to SM-BC relationships](#)

## == Large LUNs and large volumes

- \* Large LUNs and large volumes greater than 100TB are supported only on All SAN Arrays

### [NOTE]

You must ensure that both the primary and secondary cluster are All SAN Arrays, and that they both have ONTAP 9.8 or later installed. If the secondary cluster is running a version earlier than ONTAP 9.8 or if it is not an All SAN Array, the synchronous relationship can go out of sync if the primary volume grows larger than 100 TB.

## = Considerations and limits

:hardbreaks:

:icons: font

:linkattrs:

| ONTAP version | Maximum number of relationships  
 | ONTAP 9.8-9.9.1 | 5  
 | ONTAP 9.10.1 | 20  
 | ONTAP 9.11.1 | 50

#### === Volumes per consistency group

From ONTAP 9.8 to 9.9.1, the maximum number of volumes supported per SM-BC consistency group relationship is twelve, a limit which is platform-independent. Beginning with ONTAP 9.10.1, the maximum number of volumes supported per SM-BC relationship is sixteen.

#### === Volumes

Limits in SM-BC are calculated based on the number of endpoints, not the number of relationships. A consistency group with 12 volumes contributes 12 endpoints on both the source and destination. Both SM-BC and SnapMirror Synchronous relationships contribute to the total number of endpoints.

The maximum endpoints per platform are included in the following table.

[options="header"]

S. No	Platform 3+	Endpoints per HA for SM-BC 3+	Overall sync and SM-BC endpoints per HA
ONTAP 9.8-9.9.1	ONTAP 9.10.1	ONTAP 9.11.1	ONTAP 9.8-9.9.1   ONTAP 9.10.1   ONTAP 9.11.1
1			
AFF			
60			
200			
400			
80			
200			
400			
2			
ASA			
60			
200			
400			
80			
200			
400			

#### === SAN object limits

The following SAN object limits are included in the following table and apply regardless of the platform.

Limits of objects in an SM-BC relationship	Count
LUNs per volume	
256	
LUN maps per node	
2048	
LUN maps per cluster	

|4096

|LIFs per VServer (with at least one volume in an SM-BC relationship)

|256

|Inter-cluster LIFs per node

|4

|Inter-cluster LIFs per cluster

|8

<p>== Supported configurations and features</p> <p>=== Partial file restore</p> <p>Beginning in ONTAP 9.12.1, partial LUN restore is supported for SM-BC volumes. For information on this process, refer to <a href="#">Restore part of a file from a Snapshot copy</a>.</p> <p>=== Fan-out configurations</p> <p>SM-BC supports <a href="#">fan-out configurations</a> with the MirrorAllSnapshots policy and, beginning in ONTAP 9.11.1, the the MirrorAndVault policy. Fan-out configurations are not supported in SM-BC with the XDPDefault policy.</p> <p>If you experience a failover on the SM-BC destination in a fan-out configuration, you will have to manually <a href="#">resume protection in the fan-out configuration</a>.</p>	<pre>::list struct dev_info devi_sibling</pre>	<pre>::print struct dev_info devi_mdi_client</pre>	<pre>::print mdi_client_t ct_vprivate</pre>	<pre>::print struct scsi_vhci_lun svl_lun_wwn svl_fops_name"</pre>	<pre>mdb -k' ---- + ---- svl_lun_wwn = 0xa002a1c8960 "600a098038313 477543f5245397 87938" svl_fops_name = 0xa00298d69e0 "conf f_tpgs" ----</pre> <p>NOTE: <code>conf</code> will be added to the <code>svl_fops_name</code> when a <code>scsi-vhci-failover-override</code> has been applied. For additional information and recommended changes to default settings, refer to NetApp KB article <a href="#">Solaris Host support recommended settings in SnapMirror Business Continuity (SM-BC) configuration</a>.</p> <p>=== HP-UX Known issues and limitations for SM-BC configuration</p> <p>Beginning in ONTAP 9.10.1, SM-BC for HP-UX is supported. If an automatic unplanned failover (AUFO) event occurs on the isolated master cluster in</p>
--	--	--	---	--	--

Command	Description
lun igroup create	Create an igroup on a cluster
lun map	Map a LUN to an igroup
lun show	Display a list of LUNs
snapmirror create	Create a new SnapMirror relationship
snapmirror initialize	Initialize an SM-BC consistency group
snapmirror update	Initiates a common snapshot creation operation
snapmirror show	Display a list of SnapMirror relationships
snapmirror failover	Start a planned failover operation
snapmirror resync	Start a resynchronization operation
snapmirror delete	Delete a SnapMirror relationship
snapmirror release	Remove source information for a SnapMirror relationship
volume snapshot restore-file	Available with SM-BC beginning in ONTAP 9.11.1, <a href="#">restore a single file or LUN</a>

<p>= Prepare to use the ONTAP Mediator</p> <p>:hardbreaks:</p> <p>:icons: font</p> <p>:linkattrs:</p> <p>:relative_path: ./smbc/</p> <p>:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/</p> <p>[.lead]</p> <p>The ONTAP Mediator establishes a quorum for the ONTAP clusters in an SM-BC relationship. It coordinates automated failover when a failure is detected and helps to avoid split-brain scenarios when each cluster simultaneously tries to establish control as the primary cluster.</p> <p>== Prerequisites for the ONTAP Mediator</p> <p>The ONTAP Mediator includes its own set of prerequisites. You must meet these prerequisites before installing the mediator. For more information, see <a href="#">Prepare to install the ONTAP Mediator service</a>.</p> <p>== Network configuration</p> <p>By default, the ONTAP Mediator provides service through TCP port 31784. You should make sure that port 31784 is open and available between the ONTAP clusters and the mediator.</p> <p>= Summary of deployment best practices</p> <p>:hardbreaks:</p> <p>:icons: font</p> <p>:linkattrs:</p> <p>:relative_path: ./smbc/</p> <p>:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/</p>	<pre>iscsi -ostype os -initiator initiator_name`&lt;br&gt; +&lt;br&gt; &lt;strong&gt;Example&lt;/strong&gt;&lt;br&gt; +&lt;br&gt; &amp;#8230;&amp;#8203;.&lt;br&gt; lun igroup create -igroup ig1 -protocol iscsi -ostype linux -initiator -initiator iqn.2001- 04.com.example:abc123&lt;br&gt; &amp;#8230;&amp;#8203;.&lt;br&gt; &lt;br&gt; . Map LUNs to the igroup:&lt;br&gt; +&lt;br&gt; &lt;code&gt;lun map -path path_name -igroup igroup_name&lt;/code&gt;&lt;br&gt; +&lt;br&gt; &lt;strong&gt;Example:&lt;/strong&gt;&lt;br&gt; +&lt;br&gt; &amp;#8230;&amp;#8203;.&lt;br&gt; lun map -path /vol/src1/11 -group ig1&lt;br&gt; &amp;#8230;&amp;#8203;.&lt;br&gt; &lt;br&gt; . Verify the LUNs are mapped:&lt;br&gt; +&lt;br&gt; &lt;code&gt;lun show&lt;/code&gt;&lt;br&gt; &lt;br&gt; . On the application host, discover the new LUNs.&lt;br&gt; &lt;br&gt; &lt;br&gt; &lt;br&gt; :leveloffset: -1&lt;br&gt; &lt;br&gt; &lt;br&gt; = Administration&lt;br&gt; &lt;br&gt; :leveloffset: +1&lt;br&gt; &lt;br&gt; &lt;br&gt; &lt;a id="ID42651b48016229d45d51b2fe feec5e4e"&gt;↗&lt;/a&gt;&lt;br&gt; = Create a common Snapshot copy&lt;br&gt; :hardbreaks:&lt;br&gt; :icons: font&lt;br&gt; :linkattrs:&lt;br&gt; :relative_path: ./smbc/&lt;br&gt; :imagesdir: /tmp/d20230427-16143- 1lwdbyr/source/./data- protection/./media/&lt;br&gt; &lt;br&gt; &lt;br&gt; [.lead]&lt;br&gt; In addition to the regularly scheduled Snapshot copy operations, you can manually create a common Snapshot copy between the volumes in the primary SnapMirror consistency group and the volumes in the secondary SnapMirror consistency group.&lt;br&gt; &lt;br&gt; In ONTAP 9.8, the scheduled snapshot creation interval is one hour. Beginning with ONTAP 9.9.1, that interval is 12 hours.&lt;br&gt; &lt;br&gt; .Before you begin&lt;br&gt; &lt;br&gt; The SnapMirror group relationship must be in sync.&lt;br&gt; &lt;br&gt; .Steps&lt;br&gt; &lt;br&gt; . Create a common Snapshot copy:&lt;br&gt; +&lt;br&gt; &lt;code&gt;destination::&amp;gt;snapmirror update -destination-path vs1_dst:/cg/cg_dst&lt;/code&gt;&lt;br&gt; &lt;br&gt; . Monitor the progress of the update:&lt;br&gt; +&lt;br&gt;</pre>	<p>n}: y</p> <p>Operation succeeded: snapmirror delete for the relationship with destination "vs1:/cg/dd".</p> <p>....</p> <p>= Failure creating a SnapMirror relationship and initializing consistency group</p> <p>:hardbreaks:</p> <p>:icons: font</p> <p>:linkattrs:</p> <p>:relative_path: ./smbc/</p> <p>:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/</p> <p>[.lead]</p> <p>.Issue:</p> <p>Creation of SnapMirror relationship and consistency group initialization fails.</p> <p>.Solution:</p> <p>Ensure that you have not exceeded the limit of consistency groups per cluster. Consistency group limits in SM-BC are platform independent and differ based on the version of ONTAP. See <a href="#">Additional restrictions and limitations</a> for limitations based on ONTAP version.</p> <p>.Error:</p> <p>If the consistency group is stuck initializing, check the status of your consistency group initializations with the ONTAP REST API, System Manager or the command <code>sn show -expand</code>.</p> <p>.Solution:</p> <p>If consistency groups fail to initialize, remove the SM-BC relationship, delete the consistency group, then recreate the relationship and initialize it. This workflow differs depending on the</p>
---	---	--

h| If you are using ONTAP 9.8-9.9.1 h| If you are using ONTAP 9.10.1 or later

a|

- . [Remove the SM-BC configuration](#)
- . [Create a consistency group relationship](#)
- . [Initialize the consistency group relationship](#)

a|

. Under **Protection > Relationships**, find the SM-BC relationship on the consistency group. Select , then **Delete** to remove the SM-BC relationship.

- . [Delete the consistency group](#)
- . [Configure the consistency group](#)

= Planned failover unsuccessful

:hardbreaks:

:icons: font

:linkattrs:

:relative\_path: ./smbc/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

.Issue:

After executing the `snapmirror failover start` command, the output for the `snapmirror failover show` command displays a message indicates that a nondisruptive operation is in progress.

.Example:

....

Cluster1::> snapmirror failover show

Source Destination Error

Path Path Type Status start-time end-time Reason

-----  
vs1:/cg/cg vs0:/cg/cg planned failed 10/1/2020 10/1/2020 SnapMirror Failover cannot start because a volume move is running. Retry the command once volume move has finished.

08:35:04 08:35:04

....

.Cause:

Planned failover cannot begin when a nondisruptive operation is in progress, including volume move, aggregate relocation, and storage failover.

.Solution:

Wait for the nondisruptive operation to complete and try the failover operation again.

= Mediator not reachable or Mediator quorum status is false

:hardbreaks:

:icons: font

:linkattrs:

:relative\_path: ./smbc/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

.Issue:

After executing the `snapmirror failover start` command, the output for the `snapmirror failover show` command displays a message indicating that Mediator is not configured.

See [Initialize the ONTAP Mediator](#).



What to check	CLI command	Indicator
Mediator from Site A	<code>snapmirror mediator show</code>	The connection status will be unreachable
Site B connectivity	<code>cluster peer show</code>	Availability will be unavailable
Consensus status of the SM-BC volume	<code>volume show volume_name -fields smbc-consensus</code>	The sm-bc consensus field will read Awaiting-consensus

For additional information about diagnosing and resolving this issue, refer to the Knowledge Base article [Link between Site A and Mediator down and Site B down when using SM-BC](#).

= SM-BC SnapMirror delete operation fails when fence is set on destination volume

:hardbreaks:

:icons: font

:linkattrs:

:relative\_path: ./smbc/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

.Issue:

SnapMirror delete operation fails when any of the destination volumes have redirection fence set.

.Solution

Performing the following operations to retry the redirection and remove the fence from the destination volume.

- \* SnapMirror resync

- \* SnapMirror update

= Volume move operation stuck when primary is down

:hardbreaks:

:icons: font

:linkattrs:

:relative\_path: ./smbc/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

.Issue:

A volume move operation is stuck indefinitely in cutover deferred state when the primary site is down in an SM-BC relationship.

When the primary site is down, the secondary site performs an automatic unplanned failover (AUFO). When a volume move operation is in progress when the AUFO is triggered the volume move becomes stuck.

.Solution:

Abort the volume move instance that is stuck and restart the volume move operation.

= SnapMirror release fails when unable to delete Snapshot copy

:hardbreaks:

h| Mediator version h| Supported Linux versions

a| 1.5

a|

\* Red Hat Enterprise Linux: 7.6, 7.7, 7.8, 7.9, 8.1, 8.2, 8.3, 8.4, 8.5

\* CentOS: 7.6, 7.7, 7.8, 7.9

a| 1.4

a|

\* Red Hat Enterprise Linux: 7.6, 7.7, 7.8, 7.9, 8.1, 8.2, 8.3, 8.4, 8.5

\* CentOS: 7.6, 7.7, 7.8, 7.9

a| 1.3

a|

\* Red Hat Enterprise Linux: 7.6, 7.7, 7.8, 7.9, 8.1, 8.2, 8.3

\* CentOS: 7.6, 7.7, 7.8, 7.9

a| 1.2

a|

\* Red Hat Enterprise Linux: 7.6, 7.7, 7.8, 8.1

\* CentOS: 7.6, 7.7, 7.8

NOTE: The kernel version must match the operating system version.

\* 64-bit physical installation or virtual machine

\* 8 GB RAM

\* User: Root access

== Upgrade the host operating system and then the Mediator

The following table provides the upgrade guidelines if you are upgrading from RHEL/CentOS 7.6 to a later RHEL/CentOS release in addition upgrading the Mediator version.

[cols="20,20,60"]

h| Target Linux version h| Target Mediator version h| Upgrade notes

a|

\* Red Hat Enterprise Linux: 7.6, 7.7, 7.8, 8.1

\* CentOS: 7.6, 7.7, 7.8

a|

1.2

a|

- The upgrade must be performed in the following order:
  - a. Upgrade the operating system from RHEL/CentOS version.
  - b. Reboot the host to apply the kernel module changes.
  - c. Upgrade the Mediator from the immediately prior version to the current version.
- For MetroCluster:
  1. The storage iscsi-initiator show command will report that the connection to the Mediator service is down during the upgrade.
  2. The ONTAP operating system will generate the following EMS events:
    - a. cf.mccip.med.auso.stDisabled during the upgrade

b. cf.mccip.med.auso.stEnabled when automatic unplanned switchover is re-enabled

a|

\* Red Hat Enterprise Linux: 7.6, 7.7, 7.8, 7.9, 8.1, 8.2, 8.3

\* CentOS: 7.6, 7.7, 7.8, 7.9

a|

1.3

a|

.. Upgrade the operating system from RHEL/CentOS version.

.. Reboot the host to apply the kernel module changes.

.. Upgrade the Mediator from the immediately prior version to the current version.

a|

\* Red Hat Enterprise Linux: 7.6, 7.7, 7.8, 7.9, 8.1, 8.2, 8.3, 8.4, 8.5

\* CentOS: 7.6, 7.7, 7.8, 7.9

a|

1.4

a|

.. Upgrade the operating system from RHEL/CentOS version.

.. Reboot the host to apply the kernel module changes.

.. Upgrade the Mediator from the immediately prior version to the current version.

a|

\* Red Hat Enterprise Linux: 7.6, 7.7, 7.8, 7.9, 8.1, 8.2, 8.3, 8.4, 8.5

\* CentOS: 7.6, 7.7, 7.8, 7.9

a|

1.5

a|

.. Upgrade the operating system from RHEL/CentOS version.

.. Reboot the host to apply the kernel module changes.

If you do not reboot the host, an error message appears prompting you to perform a reboot.

.. Upgrade the Mediator from the immediately prior version to the current version.

a|

The best practices for installing Red Hat Enterprise Linux or CentOS and the associated repositories on your system are listed below. Systems installed or configured differently might require additional steps.

\* You must install Red Hat Enterprise Linux or CentOS according to Red Hat best practices. Due to end-of-life support for CentOS 8.x versions, compatible versions of CentOS 8.x are not recommended.

\* While installing the ONTAP Mediator service on Red Hat Enterprise Linux or CentOS, the system must have access to the appropriate repository so that the installation program can access and install all the required software dependencies.

\* For the yum installer to find dependent software in the Red Hat Enterprise Linux repositories, you must have registered the system during the Red Hat Enterprise Linux installation or afterwards by using a valid Red Hat subscription.

+

See the Red Hat documentation for information about the Red Hat Subscription Manager.

\* The following ports must be unused and available for the Mediator:

**31784**

**3260**

\* If using a third-party firewall: refer to [Firewall requirements for ONTAP Mediator](#)

\* If the Linux host is in a location without access to the internet, you must ensure that the required packages are available in a local repository.

+

If you are using Link Aggregation Control Protocol (LACP) in a Linux environment, you must correctly configure the kernel and make sure the `sysctl net.ipv4.conf.all.arp_ignore` is set to "2".

+

The following packages are required by the ONTAP Mediator service:

+

[cols="25,35,40"]

h| All RHEL/CentOS versions h| Additional packages for RHEL/CentOS 7.x h| Additional packages for RHEL 8.x

a|

- openssl
- openssl-devel
- kernel-devel
- gcc
- libselinux-utils
- make
- redhat-lsb-core
- patch
- bzip2
- python36
- python36-devel
- perl-Data-Dumper
- perl-ExtUtils-MakeMaker
- python3-pip

a|

- policycoreutils-python
- python36-pip

a|

- elfutils-libelf-devel
- policycoreutils-python-utils

The Mediator installation package is a self-extracting compressed tar file that includes:

\* An RPM file containing all dependencies that cannot be obtained from the supported release's repository.

\* An install script.

A valid SSL certification is recommended, as documented in this procedure.

== Enable access to the repositories

h| If your operating system is... h| You must provide access to these repositories...

a|

RHEL 7.x

a|

rhel-7-server-optional-rpms

a|

CentOS 7.x

a|

C7.6.1810 - Base repository

a|

RHEL 8.x

a|

- rhel-8-for-x86\_64-baseos-rpms
- rhel-8-for-x86\_64-appstream-rpms

[illegible]

| If you want to...| Use this command...

```
a|
Bring a tape drive online
a|
storage tape online
a|
Clear an alias name for tape drive or media changer
a|
storage tape alias clear
a|
Enable or disable a tape trace operation for a tape drive
a|
storage tape trace
a|
Modify the tape drive cartridge position
a|
storage tape position
a|
Reset a tape drive
a|
storage tape reset
```



This command is available only at the advanced privilege level.

```
a|
Set an alias name for tape drive or media changer
a|
storage tape alias set
a|
Take a tape drive offline
a|
storage tape offline
a|
View information about all tape drives and media changers
a|
storage tape show
a|
View information about tape drives attached to the cluster
a|
    • storage tape show-tape-drive
    • system node hardware tape drive show

a|
View information about media changers attached to the cluster
a|
storage tape show-media-changer
a|
View error information about tape drives attached to the cluster
a|
storage tape show-errors
a|
```



View all ONTAP qualified and supported tape drives attached to each node in the cluster

a|

```
storage tape show-supported-status
```

a|

View aliases of all tape drives and media changers attached to each node in the cluster

a|

```
storage tape alias show
```

a|

Reset the statistics reading of a tape drive to zero

a|

```
storage stats tape zero tape_name
```

You must use this command at the nodeshell.

a|

View tape drives supported by ONTAP

a|

```
storage show tape supported [-v]
```

You must use this command at the nodeshell. You can use the `-v` option to view more details about each tape drive.

a|

View tape device statistics to understand tape performance and check usage pattern

a|

```
storage stats tape tape_name
```

You must use this command at the nodeshell.

<p>For more information about these commands, see the man pages.</p> <p>= Use a nonqualified tape drive</p> <pre>:icons: font :relative_path: ./tape-backup/ :imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/</pre> <p>[.lead]</p> <p>You can use a nonqualified tape drive on a storage system if it can emulate a qualified tape drive. It is then treated like a qualified tape drive. To use a nonqualified tape drive, you must first determine whether it emulates any of the qualified tape drives.</p> <p>.About this task</p> <p>A nonqualified tape drive is one that is attached to the storage system, but not supported or recognized by ONTAP.</p> <p>.Steps</p> <p>. View the nonqualified tape drives attached to a storage system by using the <code>storage tape show-supported-status</code> command.</p> <p>+ The following command displays tape drives attached to the storage system and the support and qualification status of each tape drive. The nonqualified tape drives are also listed.</p> <p><code>tape_drive_vendor_name</code> is a nonqualified tape drive attached to the storage system, but not supported by ONTAP.</p> <p>+ ----</p> <pre>cluster1::&gt; storage tape show-supported-status -node Node1</pre> <p>Node: Node1</p>	<p>persistent</p>	<pre>off}*` + scsi selects the SCSI Reserve/Release mechanism. + persistent selects SCSI Persistent Reservations. + off disables tape reservations.</pre> <p>.Related information</p> <p><a href="#">What tape reservations are</a></p> <p>= Commands for verifying tape library connections</p> <pre>:icons: font :relative_path: ./tape-backup/ :imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/</pre> <p>[.lead]</p> <p>You can view information about the connection path between a storage system and a tape library configuration attached to the storage system. You can use this information to verify the connection path to the tape library configuration or for troubleshooting issues related to the connection paths.</p> <p>You can view the following tape library details to verify the tape library connections after adding or creating a new tape library, or after restoring a failed path in a single-path or multipath access to a tape library. You can also use this information while troubleshooting path-related errors or if access to a tape library fails.</p> <ul style="list-style-type: none"> <li>* Node to which the tape library is attached</li> <li>* Device ID</li> <li>* NDMP path</li> <li>* Tape library name</li> <li>* Target port and initiator port IDs</li> </ul>
--	-------------------	---

| If you want to...| Use this command...

a|

View information about a tape library in a cluster

a|

system node hardware tape library show

a|

View path information for a tape library

a|

storage tape library path show

a|

View path information for a tape library for every initiator port

a|

storage tape library path show-by-initiator

a|

View connectivity information between a storage tape library and cluster

a|

storage tape library config show

For more information about these commands, see the man pages.

:leveloffset: -1

= About tape drives

:leveloffset: +1

= Qualified tape drives overview

:icons: font

:relative\_path: ./tape-backup/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

You must use a qualified tape drive that has been tested and found to work properly on a storage system. You can follow tape aliasing and also enable tape reservations to ensure that only one storage system accesses a tape drive at any particular time.

A qualified tape drive is a tape drive that has been tested and found to work properly on storage systems. You can qualify tape drives for existing ONTAP releases by using the tape configuration file.

= Format of the tape configuration file

:icons: font

:relative\_path: ./tape-backup/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

The tape configuration file format consists of fields such as vendor ID, product ID, and details of compression types for a tape drive. This file also consists of optional fields for enabling the autoload feature of a tape drive and changing the command timeout values of a tape drive.

The following table displays the format of the tape configuration file:

[options="header"]

Item	Size	Description
a		
vendor_id	(string)	
a		
	up to 8 bytes	
a		
		The vendor ID as reported by the <code>SCSI Inquiry</code> command.
a		
product_id	(string)	

a|  
up to 16 bytes  
a|  
The product ID as reported by the SCSI Inquiry command.  
a|  
id\_match\_size(number)  
a|

a|  
The number of bytes of the product ID to be used for matching to detect the tape drive to be identified, beginning with the first character of the product ID in the Inquiry data.

a|  
vendor\_pretty (string)

a|  
up to 16 bytes

a|  
If this parameter is present, it is specified by the string displayed by the command, storage tape show -device-names; otherwise, INQ\_VENDOR\_ID is displayed.

a|  
product\_pretty(string)

a|  
up to 16 bytes

a|  
If this parameter is present, it is specified by the string displayed by the command, storage tape show -device-names; otherwise, INQ\_PRODUCT\_ID is displayed.

#### [NOTE]

====

The vendor\_pretty and product\_pretty fields are optional, but if one of these fields has a value, the other must also have a value.

====

The following table explains the description, density code, and compression algorithm for the various compression types, such as l, m, h, and a:

[options="header"]

Item	Size	Description
------	------	-------------

a|  
{l \ | m \ | h \ | a}\_description=(string)

a|  
up to 24 bytes

a|  
The string to print for the nodeshell command, sysconfig -t, that describes characteristics of the particular density setting.

a|  
{l \ | m \ | h \ | a}\_density=(hex codes)

a|  
The density code to be set in the SCSI mode page block descriptor corresponding to the desired density code for l, m, h, or a.

a|

```
{l \ | m \ | h \ | a}_algorithm=(hex codes)
a|
```

a|  
The compression algorithm to be set in the SCSI Compression Mode Page corresponding to the density code and the desired density characteristic.

The following table describes the optional fields available in the tape configuration file:  
  
[options="header"]


| Field| Description

```
a|
autoload=(Boolean yes/no)
```

a|  
This field is set to *yes* if the tape drive has an automatic loading feature; that is, after tape cartridge is inserted, the tape drive becomes ready without the need to execute a SCSI *load* (start/stop unit) command. The default for this field is *no*.

```
a|
cmd_timeout_0x
```

a|  
Individual timeout value. You must use this field only if you want to specify a different timeout value from the one being used as a default by the tape driver. The sample file lists the default SCSI command timeout values used by the tape drive. The timeout value can be expressed in minutes (m), seconds (s), or milliseconds (ms).

 You should not change this field.

You can download and view the tape configuration file from the NetApp Support Site.

.Example of a tape configuration file format

The tape configuration file format for the HP LTO5 ULTRIUM tape drive is as follows:

```
vendor_id="HP"

product_id="Ultrium 5-SCSI"

id_match_size=9

vendor_pretty="Hewlett-Packard"

product_pretty="LTO-5"

l_description="LTO-3(ro)/4 4/800GB"

l_density=0x00

l_algorithm=0x00

m_description="LTO-3(ro)/4 8/1600GB cmp"

m_density=0x00

m_algorithm=0x01

h_description="LTO-5 1600GB"

h_density=0x58

h_algorithm=0x00

a_description="LTO-5 3200GB cmp"

a_density=0x58

a_algorithm=0x01

autoload="yes"
```

.Related information

[NetApp Tools: Tape Device Configuration Files](#)

= How the storage system qualifies a new tape drive dynamically

:icons: font

:relative\_path: ./tape-backup/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

| Scenario| Reassigning of the alias

a|

When the system reboots

a|

The tape drive is automatically reassigned its previous alias.

a|

When a tape device moves to another port

a|

The alias can be adjusted to point to the new address.

a|

When more than one system uses a particular tape device

a|

The user can set the alias to be the same for all the systems.



[NOTE]

====

When you upgrade from Data ONTAP 8.1.x to Data ONTAP 8.2.x, the tape alias feature of Data ONTAP 8.2.x modifies the existing tape alias names. In such a case you might have to update the tape alias names in the backup application.

====

Assigning tape aliases provides a correspondence between the logical names of backup devices (for example, st0 or mc1) and a name permanently assigned to a port, a tape drive, or a medium changer.

[NOTE]

====

st0 and st00 are different logical names.

====

[NOTE]

====

Logical names and serial numbers are used only to access a device. After the device is accessed, it returns all error messages by using the physical path name.

====

There are two types of names available for aliasing: physical path name and serial number.

= What physical path names are

:icons: font

:relative\_path: ./tape-backup/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

Physical path names (PPNs) are the numerical address sequences that ONTAP assigns to tape drives and tape libraries based on the SCSI-2/3 adapter or switch (specific location) they are connected to the storage system. PPNs are also known as electrical names.

PPNs of direct-attached devices use the following format: `host_adapter.device_id_lun`

[NOTE]

====

The LUN value is displayed only for tape and medium changer devices whose LUN values are not zero; that is, if the LUN value is zero the `lun` part of the PPN is not displayed.

====

For example, the PPN 8.6 indicates that the host adapter number is 8, the device ID is 6, and the logical unit number (LUN) is 0.

SAS tape devices are also direct-attached devices. For example, the PPN 5c.4 indicates that in a storage system, the SAS HBA is connected in slot 5, SAS tape is connected to port C of the SAS HBA, and the device ID is 4.

PPNs of Fibre Channel switch-attached devices use the following format: `switch:port_id.device_id_lun`

h| If you are performing data transfer at the source or destination in... h| Use the following command...

a|  
SVM-scoped NDMP mode

a|  
`vserver services ndmp on`



For NDMP authentication in the admin SVM, the user account is `admin` and the user role is `admin` or `backup`. In the data SVM, the user account is `vsadmin` and the user role is `vsadmin` or `vsadmin-backup` role.

a|  
Node-scoped NDMP mode

a|  
`system services ndmp on`

<p>. Transfer data within a storage system or between storage systems using the <code>ndmpcopy</code> command at the nodeshell:</p> <pre>+ `*::&gt; system node run -node &lt;node_name&gt; &lt; ndmpcopy [options] source_IP:source_path destination_IP:destination _path [-mcs {inet</pre>	<pre>inet6}} [-mcd {inet</pre>	<pre>inet6}} [-md {inet</pre>	<pre>inet6}}` + [NOTE] ==== DNS names are not supported in ndmpcopy. You must provide the IP address of the source and the destination. The loopback address (127.0.0.1) is not supported for the source IP address or the destination IP address. ====  The ndmpcopy command determines the address mode for control connections as follows: The address mode for control connection corresponds to the IP address provided. * You can override these rules by using the -mcs and -mcd options. If the source or the destination is the ONTAP system, then depending on the NDMP mode (node-scoped or SVM-scoped), use an IP address that allows access to the target volume. source_path and destination_path are the absolute path names till the granular level of volume, qtree, directory or file. -mcs specifies the preferred addressing mode for the control connection to the source storage system. + inet indicates an IPv4 address mode and inet6 indicates an IPv6 address mode.</pre>
--	--------------------------------	-------------------------------	--

| Option| Description

a|

-sa username:[password]

a|

This option sets the source authentication user name and password for connecting to the source storage system. This is a mandatory option.

For a user without admin privilege, you must specify the user's system-generated NDMP-specific password. The system-generated password is mandatory for both admin and non-admin users.

a|

-da username:[password]

a|

This option sets the destination authentication user name and password for connecting to the destination storage system. This is a mandatory option.

a|

-st {md5|text}

a|

This option sets the source authentication type to be used when connecting to the source storage system. This is a mandatory option and therefore the user should provide either the `text` or `md5` option.

a|

-dt {md5|text}

a|

This option sets the destination authentication type to be used when connecting to the destination storage system.

a|

-l

a|

This option sets the dump level used for the transfer to the specified value of level. Valid values are 0, 1, to 9, where 0 indicates a full transfer and 1 to 9 specifies an incremental transfer. The default is 0.

a|

-d

a|

This option enables generation of ndmcopy debug log messages. The ndmcopy debug log files are located in the `/mroot/etc/log` root volume. The ndmcopy debug log file names are in the `ndmcopy.yyyymmdd` format.

a|

-f

a|

This option enables the forced mode. This mode enables system files to be overwritten in the `/etc` directory on the root of the 7-Mode volume.

a|

-h

a|

This option prints the help message.

a|

-p

a|

This option prompts you to enter the password for source and destination authorization. This password overrides the password specified for `-sa` and `-da` options.



You can use this option only when the command is running in an interactive console.

a|  
`-exclude`  
a|  
This option excludes specified files or directories from the path specified for data transfer. The value can be a comma-separated list of directory or file names such as **.pst** or `.txt`.

:leveloffset: -1

= NDMP for FlexVol volumes

:leveloffset: +1

= About NDMP for FlexVol volumes

:icons: font

:relative\_path: ./tape-backup/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

The Network Data Management Protocol (NDMP) is a standardized protocol for controlling backup, recovery, and other types of data transfer between primary and secondary storage devices, such as storage systems and tape libraries.

By enabling NDMP support on a storage system, you enable that storage system to communicate with NDMP-enabled network-attached backup applications (also called *Data Management Applications* or *DMAs*), data servers, and tape servers participating in backup or recovery operations. All network communications occur over TCPIP or TCP/IPv6 network. NDMP also provides low-level control of tape drives and medium changers.

You can perform tape backup and restore operations in either node-scoped NDMP mode or storage virtual machine (SVM) scoped NDMP mode.

You must be aware of the considerations that you have to take into account while using NDMP, list of environment variables, and supported NDMP tape backup topologies. You can also enable or disable the enhanced DAR functionality. The two authentication methods supported by ONTAP for authenticating NDMP access to a storage system are: plaintext and challenge.

.Related information

[Environment variables supported by ONTAP](#)

= About NDMP modes of operation

:icons: font

:relative\_path: ./tape-backup/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

You can choose to perform tape backup and restore operations either at the node level as you have been doing until now or at the storage virtual machine (SVM) level. To perform these operations successfully at the SVM level, NDMP service must be enabled on the SVM.

If you upgrade from Data ONTAP 8.2 to Data ONTAP 8.3, the NDMP mode of operation used in 8.2 will continue to be retained post the upgrade from 8.2 to 8.3.

Environment variable	Valid values	Default	Description
----------------------	--------------	---------	-------------

a  DEBUG			
-------------	--	--	--

a  Y or N			
--------------	--	--	--

a  N			
---------	--	--	--

a  Specifies that debugging information is printed.			
--	--	--	--

a  FILESYSTEM			
------------------	--	--	--

a  string			
--------------	--	--	--

a  none			
------------	--	--	--

a  Specifies the path name of the root of the data that is being backed up.			
--	--	--	--

a  NDMP_VERSION			
--------------------	--	--	--

a  return_only			
-------------------	--	--	--

a  none			
------------	--	--	--

a  You should not modify the NDMP_VERSION variable. Created by the backup operation, the NDMP_VERSION variable returns the NDMP version.			
---	--	--	--

ONTAP sets the NDMP\_VERSION variable during a backup for internal use and to pass to a backup application for informational purposes. The NDMP version of an NDMP session is not set with this variable.

a  PATHNAME_SEPARATOR			
--------------------------	--	--	--

a  return_value			
--------------------	--	--	--

a  none			
------------	--	--	--

a  Specifies the path name separator character.			
--	--	--	--

This character depends on the file system being backed up. For ONTAP, the character “/” is assigned to this variable. The NDMP server sets this variable before starting a tape backup operation.

a  TYPE			
------------	--	--	--

a  dump or smtape			
----------------------	--	--	--

a  dump			
------------	--	--	--

a  Specifies the type of backup supported to perform tape backup and restore operations.			
---	--	--	--

a			
---	--	--	--

VERBOSE

a|

Y or N

a|

N

a|

Increases the log messages while performing a tape backup or restore operation.

== Environment variables supported for dump

[options="header"]

| Environment variable| Valid values| Default| Description

a|

ACL\_START

a|

return\_only

a|

none

a|

Created by the backup operation, the ACL\_START variable is an offset value used by a direct access restore or restartable NDMP backup operation.

The offset value is the byte offset in the dump file where the ACL data (Pass V) begins and is returned at the end of a backup. For a direct access restore operation to correctly restore backed-up data, the ACL\_START value must be passed to the restore operation when it begins. An NDMP restartable backup operation uses the ACL\_START value to communicate to the backup application where the nonrestartable portion of the backup stream begins.

a|

BASE\_DATE

a|

0, -1, or DUMP\_DATE value

a|

-1

a|

Specifies the start date for incremental backups.

When set to -1, the BASE\_DATE incremental specifier is disabled. When set to 0 on a level 0 backup, incremental backups are enabled. After the initial backup, the value of the DUMP\_DATE variable from the previous incremental backup is assigned to the BASE\_DATE variable.

These variables are an alternative to the LEVEL/UPDATE based incremental backups.

a|

DIRECT

a|

Y or N

a|

N

a|

Specifies that a restore should fast-forward directly to the location on the tape where the file data resides instead of scanning the entire tape.



For direct access recovery to work, the backup application must provide positioning information. If this variable is set to `Y`, the backup application specifies the file or directory names and the positioning information.

a|  
DMP\_NAME

a|  
string

a|  
none

a|  
Specifies the name for a multiple subtree backup.

This variable is mandatory for multiple subtree backups.

a|  
DUMP\_DATE

a|  
return\_value

a|  
none

a|  
You do not change this variable directly. It is created by the backup if the `BASE_DATE` variable is set to a value other than `-1`.

The `DUMP_DATE` variable is derived by prepending the 32-bit level value to a 32-bit time value computed by the dump software. The level is incremented from the last level value passed into the `BASE_DATE` variable. The resulting value is used as the `BASE_DATE` value on a subsequent incremental backup.

a|  
ENHANCED\_DAR\_ENABLED

a|  
Y or N

a|  
N  
a|  
Specifies whether enhanced DAR functionality is enabled. Enhanced DAR functionality supports directory DAR and DAR of files with NT Streams. It provides performance improvements.

Enhanced DAR during restore is possible only if the following conditions are met:

- ONTAP supports enhanced DAR.
- File history is enabled (`HIST=Y`) during the backup.
- The `ndmpd.offset_map.enable` option is set to `on`.
- `ENHANCED_DAR_ENABLED` variable is set to `Y` during restore.

a|  
EXCLUDE

a|  
pattern\_string

a|  
none

a|

Specifies files or directories that are excluded when backing up data.

The exclude list is a comma-separated list of file or directory names. If the name of a file or directory matches one of the names in the list, it is excluded from the backup.

The following rules apply while specifying names in the exclude list:

- The exact name of the file or directory must be used.
- The asterisk (\*), a wildcard character, must be either the first or the last character of the string.

Each string can have up to two asterisks.

- A comma in a file or directory name must be preceded with a backslash.
- The exclude list can contain up to 32 names.



Files or directories specified to be excluded for backup are not excluded if you set `NON_QUOTA_TREE` to `Y` simultaneously.

a|  
EXTRACT  
a|  
Y, N, or E  
a|  
N  
a|

Specifies that subtrees of a backed-up data set are to be restored.

The backup application specifies the names of the subtrees to be extracted. If a file specified matches a directory whose contents were backed up, the directory is recursively extracted.

To rename a file, directory, or qtree during restore without using DAR, you must set the `EXTRACT` environment variable to `E`.

a|  
EXTRACT\_ACL  
a|  
Y or N  
a|  
Y  
a|

Specifies that ACLs from the backed up file are restored on a restore operation.

The default is to restore ACLs when restoring data, except for DARs (`DIRECT=Y`).

a|  
FORCE  
a|  
Y or N  
a|  
N  
a|

Determines if the restore operation must check for volume space and inode availability on the destination volume.

Setting this variable to `Y` causes the restore operation to skip checks for volume space and inode availability on the destination path.

If enough volume space or inodes are not available on the destination volume, the restore operation recovers as much data allowed by the destination volume space and inode availability. The restore operation stops when volume space or inodes are not available.

```
a|
HIST
a|
Y or N
a|
N
a|
Specifies that file history information is sent to the backup application.
```

Most commercial backup applications set the HIST variable to `Y`. If you want to increase the speed of a backup operation, or you want to troubleshoot a problem with the file history collection, you can set this variable to `N`.



You should not set the HIST variable to `Y` if the backup application does not support file history.

```
a|
IGNORE_CTIME
a|
Y or N
a|
N
a|
Specifies that a file is not incrementally backed up if only its ctime value has changed since the previous incremental backup.
```

Some applications, such as virus scanning software, change the ctime value of a file within the inode, even though the file or its attributes have not changed. As a result, an incremental backup might back up files that have not changed. The `IGNORE_CTIME` variable should be specified only if incremental backups are taking an unacceptable amount of time or space because the ctime value was modified.

The `NDMP dump` command sets `IGNORE_CTIME` to `false` by default. Setting it to `true` can result in the following data loss:



1. If `IGNORE_CTIME` is set to `true` with a volume level incremental `ndmpcopy`, it results in the deleting of files, which are moved across qtrees on source.
2. If `IGNORE_CTIME` is set to `true` during a volume level incremental dump, it results in the deleting of files, which are moved across qtrees on source during incremental restore.

To avoid this problem, `IGNORE_CTIME` must be set to `false` during volume level `NDMP dumps` or `ndmpcopy`.

```
a|
IGNORE_QTREES
a|
Y or N
a|
```

N

a|

Specifies that the restore operation does not restore qtree information from backed-up qtrees.

a|

LEVEL

a|

0-31

a|

0

a|

Specifies the backup level.

Level 0 copies the entire data set. Incremental backup levels, specified by values above 0, copy all files (new or modified) since the last incremental backup. For example, a level 1 backs up new or modified files since the level 0 backup, a level 2 backs up new or modified files since the level 1 backup, and so on.

a|

LIST

a|

Y or N

a|

N

a|

Lists the backed-up file names and inode numbers without actually restoring the data.

a|

LIST\_QTREES

a|

Y or N

a|

N

a|

Lists the backed-up qtrees without actually restoring the data.

a|

MULTI\_SUBTREE\_NAMES

a|

string

a|

none

a|

Specifies that the backup is a multiple subtree backup.

Multiple subtrees are specified in the string, which is a newline-separated, null-terminated list of subtree names. Subtrees are specified by path names relative to their common root directory, which must be specified as the last element of the list.

If you use this variable, you must also use the DMP\_NAME variable.

a|

NDMP\_UNICODE\_FH

a|

Y or N

a|  
N  
a|  
Specifies that a Unicode name is included in addition to the NFS name of the file in the file history information.

This option is not used by most backup applications and should not be set unless the backup application is designed to receive these additional file names. The HIST variable must also be set.

a|  
NO\_ACLS  
a|  
Y or N  
a|  
N  
a|  
Specifies that ACLs must not be copied when backing up data.

a|  
NON\_QUOTA\_TREE  
a|  
Y or N  
a|  
N  
a|  
Specifies that files and directories in qtrees must be ignored when backing up data.

When set to Y, items in qtrees in the data set specified by the FILESYSTEM variable are not backed up. This variable has an effect only if the FILESYSTEM variable specifies an entire volume. The NON\_QUOTA\_TREE variable only works on a level 0 backup and does not work if the MULTI\_SUBTREE\_NAMES variable is specified.



Files or directories specified to be excluded for backup are not excluded if you set NON\_QUOTA\_TREE to Y simultaneously.

a|  
NOWRITE  
a|  
Y or N  
a|  
N  
a|  
Specifies that the restore operation must not write data to the disk.

This variable is used for debugging.

a|  
RECURSIVE  
a|  
Y or N  
a|  
Y  
a|  
Specifies that directory entries during a DAR restore be expanded.

The `DIRECT` and `ENHANCED_DAR_ENABLED` environment variables must be enabled (set to `Y`) as well. If the `RECURSIVE` variable is disabled (set to `N`), only the permissions and ACLs for all the directories in the original source path are restored from tape, not the contents of the directories. If the `RECURSIVE` variable is set to `N` or the `RECOVER_FULL_PATHS` variable is set to `Y`, the recovery path must end with the original path.



If the `RECURSIVE` variable is disabled and if there is more than one recovery path, all of the recovery paths must be contained within the longest of the recovery paths. Otherwise, an error message is displayed.

For example, the following are valid recovery paths because all of the recovery paths are within `foo/dir1/deepdir/myfile`:

- `/foo`
- `/foo/dir`
- `/foo/dir1/deepdir`
- `/foo/dir1/deepdir/myfile`

The following are invalid recovery paths:

- `/foo`
- `/foo/dir`
- `/foo/dir1/myfile`
- `/foo/dir2`
- `/foo/dir2/myfile`

a|  
`RECOVER_FULL_PATHS`

a|  
`Y` or `N`

a|  
`N`

a|  
Specifies that the full recovery path will have their permissions and ACLs restored after the DAR.

`DIRECT` and `ENHANCED_DAR_ENABLED` must be enabled (set to `Y`) as well. If `RECOVER_FULL_PATHS` is set to `Y`, the recovery path must end with the original path. If directories already exist on the destination volume, their permissions and ACLs will not be restored from tape.

a|  
`UPDATE`

a|  
`Y` or `N`

a|  
`Y`

a|  
Updates the metadata information to enable `LEVEL` based incremental backup.

== Environment variables supported for SMTape

[options="header"]

Environment variable	Valid values	Default	Description
----------------------	--------------	---------	-------------

a			
---	--	--	--

BASE_DATE			
-----------	--	--	--

a			
---	--	--	--

DUMP_DATE			
-----------	--	--	--

a			
---	--	--	--

-1			
----	--	--	--

a			
---	--	--	--

Specifies the start date for incremental backups.			
---	--	--	--

BASE\_DATE is a string representation of the reference Snapshot identifiers. Using the BASE\_DATE string, SMTape locates the reference Snapshot copy.

BASE\_DATE is not required for baseline backups. For an incremental backup, the value of the DUMP\_DATE variable from the previous baseline or incremental backup is assigned to the BASE\_DATE variable.

The backup application assigns the DUMP\_DATE value from a previous SMTape baseline or incremental backup.

a			
---	--	--	--

DUMP_DATE			
-----------	--	--	--

a			
---	--	--	--

return_value			
--------------	--	--	--

a			
---	--	--	--

none			
------	--	--	--

a			
---	--	--	--

At the end of an SMTape backup, DUMP_DATE contains a string identifier that identifies the Snapshot copy used for that backup. This Snapshot copy could be used as the reference Snapshot copy for a subsequent incremental backup.			
---	--	--	--

The resulting value of DUMP\_DATE is used as the BASE\_DATE value for subsequent incremental backups.

a			
---	--	--	--

SMTAPE_BACKUP_SET_ID			
----------------------	--	--	--

a			
---	--	--	--

string			
--------	--	--	--

a			
---	--	--	--

none			
------	--	--	--

a			
---	--	--	--

Identifies the sequence of incremental backups associated with the baseline backup.			
---	--	--	--

Backup set ID is a 128-bit unique ID that is generated during a baseline backup. The backup application assigns this ID as the input to the SMTAPE\_BACKUP\_SET\_ID variable during an incremental backup.

a			
---	--	--	--

SMTAPE_SNAPSHOT_NAME			
----------------------	--	--	--

a			
---	--	--	--

Any valid Snapshot copy that is available in the volume			
---	--	--	--

a			
---	--	--	--

Invalid

a|  
When the SMTAPE\_SNAPSHOT\_NAME variable is set to a Snapshot copy, that Snapshot copy and its older Snapshot copies are backed up to tape.

For incremental backup, this variable specifies incremental Snapshot copy. The BASE\_DATE variable provides the baseline Snapshot copy.

a|  
SMTAPE\_DELETE\_SNAPSHOT

a|  
Y or N

a|  
N  
a|  
For a Snapshot copy created automatically by SMTape, when the SMTAPE\_DELETE\_SNAPSHOT variable is set to Y, then after the backup operation is complete, SMTape deletes this Snapshot copy. However, a Snapshot copy created by the backup application will not be deleted.

a|  
SMTAPE\_BREAK\_MIRROR

a|  
Y or N

a|  
N  
a|  
When the SMTAPE\_BREAK\_MIRROR variable is set to Y, the volume of type DP is changed to a RW volume after a successful restore.



:leveloffset: -1

= Common NDMP tape backup topologies

:icons: font

:relative\_path: ./tape-backup/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

NDMP supports a number of topologies and configurations between backup applications and storage systems or other NDMP servers providing data (file systems) and tape services.

== Storage system-to-local-tape

In the simplest configuration, a backup application backs up data from a storage system to a tape subsystem attached to the storage system. The NDMP control connection exists across the network boundary. The NDMP data connection that exists within the storage system between the data and tape services is called an NDMP local configuration.

== Storage system-to-tape attached to another storage system

A backup application can also back up data from a storage system to a tape library (a medium changer with one or more tape drives) attached to another storage system. In this case, the NDMP data connection between the data and tape services is provided by a TCP or TCP/IPv6 network connection. This is called an NDMP three-way storage system-to-storage system configuration.

== Storage system-to-network-attached tape library

NDMP-enabled tape libraries provide a variation of the three-way configuration. In this case, the tape library attaches directly to the TCP/IP network and communicates with the backup application and the storage system through an internal NDMP server.

== Storage system-to-data server-to-tape or data server-to-storage system-to-tape

NDMP also supports storage system-to-data-server and data-server-to-storage system three-way configurations, although these variants are less widely deployed. Storage system-to-server allows storage system data to be backed up to a tape library attached to the backup application host or to another data server system. The server-to-storage system configuration allows server data to be backed up to a storage system-attached tape library.

= Supported NDMP authentication methods

:icons: font

:relative\_path: ./tape-backup/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

You can specify an authentication method to allow NDMP connection requests. ONTAP supports two methods for authenticating NDMP access to a storage system: plaintext and challenge.

In node-scoped NDMP mode, both challenge and plaintext are enabled by default. However, you cannot

| System memory of a storage system| Maximum number of NDMP sessions

a|

Less than 16 GB

a|

8

a|

Greater than or equal to 16 GB but less than 24 GB

a|

20

a|

Greater than or equal to 24 GB

a|

36

You can obtain the system memory of your storage system by using the `sysconfig -a` command (available through the nodeshell). For more information about using this command, see the man pages.

:leveloffset: -1

= About NDMP for FlexGroup volumes

:icons: font

:relative\_path: ./tape-backup/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

Beginning with ONTAP 9.7, NDMP is supported on FlexGroup volumes.

Beginning with ONTAP 9.7, the `ndmpcopy` command is supported for data transfer between FlexVol and FlexGroup volumes.

If you revert from ONTAP 9.7 to an earlier version, the incremental transfer information of the previous transfers is not retained and therefore, you must perform a baseline copy after reverting.

Beginning with ONTAP 9.8, the following NDMP features are supported on FlexGroup volumes:

- \* The NDMP\_SNAP\_RECOVER message in the extension class 0x2050 can be used for recovering individual files in a FlexGroup volume.
- \* NDMP restartable backup extension (RBE) is supported for FlexGroup volumes.
- \* Environment variables EXCLUDE and MULTI\_SUBTREE\_NAMES are supported for FlexGroup volumes.

= About NDMP with SnapLock volumes

:icons: font

:relative\_path: ./tape-backup/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

Creating multiple copies of regulated data provides you with redundant recovery scenarios, and by using NDMP dump and restore, it's possible to preserve the write once, read many (WORM) characteristics of source files on a SnapLock volume.

WORM attributes on the files in a SnapLock volume are preserved when backing up, restoring and copying data; however, WORM attributes are enforced only when restoring to a SnapLock volume. If a backup from a SnapLock volume is restored to a volume other than a SnapLock volume, the WORM attributes are preserved but are ignored and are not enforced by ONTAP.

= Manage node-scoped NDMP mode for FlexVol volumes

:leveloffset: +1

| If you want to...| Use this command...

a|

Enable NDMP service

a|

system services ndmp on\*

a|

Disable NDMP service

a|

system services ndmp off\*

a|

Display NDMP configuration

a|

system services ndmp show\*

a|

Modify NDMP configuration

a|

system services ndmp modify\*

a|

Display the default NDMP version

a|

system services ndmp version\*

a|

Display NDMP service configuration

a|

system services ndmp service show

a|

Modify NDMP service configuration

a|

system services ndmp service modify

a|

Display all NDMP sessions

a|

system services ndmp status

a|

Display detailed information about all NDMP sessions

a|

system services ndmp probe

a|

Terminate the specified NDMP session

a|

system services ndmp kill

a|

Terminate all NDMP sessions

a|

system services ndmp kill-all

a|

Change the NDMP password

a|

system services ndmp password\*

a|

Enable node-scoped NDMP mode

a|

system services ndmp node-scope-mode on\*

a|

Disable node-scoped NDMP mode

a|

```
system services ndmp node-scope-mode off*
```

a|

Display the node-scoped NDMP mode status

a|

```
system services ndmp node-scope-mode status*
```

a|

Forcefully terminate all NDMP sessions

a|

```
system services ndmp service terminate
```

a|

Start the NDMP service daemon

a|

```
system services ndmp service start
```

a|

Stop the NDMP service daemon

a|

```
system services ndmp service stop
```

a|

Start logging for the specified NDMP session

a|

```
system services ndmp log start*
```

a|

Stop logging for the specified NDMP session

a|

```
system services ndmp log stop*
```

\* These commands are deprecated and will be removed in a future major release.

For more information about these commands, see the man pages for the `system services ndmp` commands.

= User authentication in a node-scoped NDMP mode

:icons: font

:relative\_path: ./tape-backup/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

In the node-scoped NDMP mode, you must use NDMP specific credentials to access a storage system in order to perform tape backup and restore operations.

The default user ID is “root”. Before using NDMP on a node, you must ensure that you change the default NDMP password associated with the NDMP user. You can also change the default NDMP user ID.

.Related information

[Commands for managing node-scoped NDMP mode](#)

:leveloffset: -1

= Manage SVM-scoped NDMP mode for FlexVol volumes

:leveloffset: +1

= Manage SVM-scoped NDMP mode for FlexVol volumes overview

:icons: font

:relative\_path: ./tape-backup/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

You can manage NDMP on a per SVM basis by using the NDMP options and commands. You can modify the NDMP options by using the `vserver services ndmp modify` command. In the SVM-scoped NDMP mode, user authentication is integrated with the role-based access control mechanism.

You can add NDMP in the allowed or disallowed protocols list by using the `vserver modify` command. By default, NDMP is in the allowed protocols list. If NDMP is added to the disallowed protocols list, NDMP sessions cannot be established.

You can control the LIF type on which an NDMP data connection is established by using the `-preferred -interface-role` option. During an NDMP data connection establishment, NDMP chooses an IP address that belongs to the LIF type as specified by this option. If the IP addresses do not belong to any of these LIF types, then the NDMP data connection cannot be established. For more information about the `-preferred -interface-role` option, see the man pages.

| If you want to...| Use this command...

a|

Enable NDMP service

a|

```
vserver services ndmp on
```



NDMP service must always be enabled on all nodes in a cluster. You can enable NDMP service on a node by using the `system services ndmp on` command. By default, NDMP service is always enabled on a node.

a|

Disable NDMP service

a|

```
vserver services ndmp off
```

a|

Display NDMP configuration

a|

```
vserver services ndmp show
```

a|

Modify NDMP configuration

a|

```
vserver services ndmp modify
```

a|

Display default NDMP version

a|

```
vserver services ndmp version
```

a|

Display all NDMP sessions

a|

```
vserver services ndmp status
```

a|

Display detailed information about all NDMP sessions

a|

```
vserver services ndmp probe
```

a|

Terminate a specified NDMP session

a|

```
vserver services ndmp kill
```

a|

Terminate all NDMP sessions

a|

```
vserver services ndmp kill-all
```

a|

Generate the NDMP password

a|

```
vserver services ndmp generate-password
```

a|

Display NDMP extension status

a|

```
vserver services ndmp extensions show
```

This command is available at the advanced privilege level.

a|

Modify (enable or disable) NDMP extension status

a|

```
vserver services ndmp extensions modify
```

This command is available at the advanced privilege level.

a|

Start logging for the specified NDMP session

a|

```
vserver services ndmp log start
```

This command is available at the advanced privilege level.

a|

Stop logging for the specified NDMP session

a|

```
vserver services ndmp log stop
```

This command is available at the advanced privilege level.



For more information about these commands, see the man pages for the `vserver services ndmp` commands.

= What Cluster Aware Backup extension does

:icons: font

:relative\_path: ./tape-backup/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

CAB (Cluster Aware Backup) is an NDMP v4 protocol extension. This extension enables the NDMP server to establish a data connection on a node that owns a volume. This also enables the backup application to determine if volumes and tape devices are located on the same node in a cluster.

To enable the NDMP server to identify the node that owns a volume and to establish a data connection on such a node, the backup application must support the CAB extension. CAB extension requires the backup application to inform the NDMP server about the volume to be backed up or restored prior to establishing the data connection. This allows the NDMP server to determine the node that hosts the volume and appropriately establish the data connection.

With the CAB extension supported by the backup application, the NDMP server provides affinity information about volumes and tape devices. Using this affinity information, the backup application can perform a local backup instead of a three-way backup if a volume and tape device are located on the same node in a cluster.

= Availability of volumes and tape devices for backup and restore on different LIF types

:icons: font

:relative\_path: ./tape-backup/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

You can configure a backup application to establish an NDMP control connection on any of the LIF types in a cluster. In the storage virtual machine (SVM)-scoped NDMP mode, you can determine the availability of volumes and tape devices for backup and restore operations depending upon these LIF types and the status of the CAB extension.

The following tables show the availability of volumes and tape devices for NDMP control connection LIF types and the status of the CAB extension:

== Availability of volumes and tape devices when CAB extension is not supported by the backup application

[options="header"]

NDMP control connection LIF type  Volumes available for backup or restore  Tape devices available for backup or restore		
a		
Node-management LIF		
a		
All volumes hosted by a node		
a		

Tape devices connected to the node hosting the node-management LIF

a|

Data LIF

a|

Only volumes that belong to the SVM hosted by a node that hosts the data LIF

a|

None

a|

Cluster-management LIF

a|

All volumes hosted by a node that hosts the cluster-management LIF

a|

None

a|

Intercluster LIF

a|

All volumes hosted by a node that hosts the intercluster LIF

a|

Tape devices connected to the node hosting the intercluster LIF

== Availability of volumes and tape devices when CAB extension is supported by the backup application

[options="header"]

| NDMP control connection LIF type| Volumes available for backup or restore| Tape devices available for backup or restore

a|

Node-management LIF

a|

All volumes hosted by a node

a|

Tape devices connected to the node hosting the node-management LIF

a|

Data LIF

a|

All volumes that belong to the SVM that hosts the data LIF

a|

None

a|

Cluster-management LIF

a|

All volumes in the cluster

a|

All tape devices in the cluster

a|

Intercluster LIF

a|

All volumes in the cluster

a|

All tape devices in the cluster

= What affinity information is

:icons: font

:relative\_path: ./tape-backup/

:imagesdir: /tmp/d20230427-16143-

1lwdbyr/source/./data-protection/./media/

[.lead]

With the backup application being CAB aware, the NDMP server provides unique location information about volumes and tape devices. Using this affinity information, the backup application can perform a local backup instead of a three-way backup if a volume and a tape device share the same affinity.

If the NDMP control connection is established on a node management LIF, cluster management LIF, or an intercluster LIF, the backup application can use the affinity information to determine if a volume and tape device are located on the same node and then perform either a local or a three-way backup or restore operation. If the NDMP control connection is established on a data LIF, then the backup application always performs a three-way backup.

== Local NDMP backup and Three-way NDMP backup

image::.../media/local\_and\_three-

way\_backup\_in\_vserver\_aware\_ndmp\_mode.png[local and three way back up diagrams]

Using the affinity information about volumes and tape devices, the DMA (backup application) performs a local NDMP backup on the volume and tape device located on Node 1 in the cluster. If the volume moves from Node 1 to Node 2, affinity information about the volume and tape device changes. Hence, for a subsequent backup the DMA performs a three-way NDMP backup operation. This ensures continuity of the backup policy for the volume irrespective of the node to which the volume is moved to.

.Related information

[What Cluster Aware Backup extension does](#)

= NDMP server supports secure control connections in SVM-scoped mode

:icons: font

:relative\_path: ./tape-backup/

false]` command.

= NDMP data connection types

:icons: font

:relative\_path: ./tape-backup/

:imagesdir: /tmp/d20230427-16143-

1lwdbyr/source/./data-protection/./media/

[.lead]

In the storage virtual machine (SVM)-scoped NDMP mode, the supported NDMP data connection types depend on the NDMP control connection LIF type and the status of the CAB extension. This NDMP data connection type indicates whether you can perform a local or a three-way NDMP backup or restore operation.

You can perform a three-way NDMP backup or restore operation over a TCP or TCP/IPv6 network. The following tables show the NDMP data connection types based on the NDMP control connection LIF type and the status of the CAB extension.

== NDMP data connection type when CAB extension is supported by the backup application

[options="header"]

| NDMP control connection LIF type| NDMP data connection type

a|

Node-management LIF

a|

LOCAL, TCP, TCP/IPv6

a|

Data LIF

a|

TCP, TCP/IPv6

a|

Cluster-management LIF

a|

LOCAL, TCP, TCP/IPv6

a|

Intercluster LIF

a|

LOCAL, TCP, TCP/IPv6

== NDMP data connection type when CAB extension is not supported by the backup application

[options="header"]

| NDMP control connection LIF type| NDMP data connection type

a|

Node-management LIF

a|

LOCAL, TCP, TCP/IPv6

a|

Data LIF

a|

TCP, TCP/IPv6

a|

Cluster-management LIF

a|

TCP, TCP/IPv6

a|

Intercluster LIF

a|

LOCAL, TCP, TCP/IPv6

.Related information

[What Cluster Aware Backup extension does](#)

[Network management](#)

= User authentication in the SVM-scoped NDMP mode

:icons: font

:relative\_path: ./tape-backup/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

In the storage virtual machine (SVM)-scoped NDMP mode, NDMP user authentication is integrated with role-based access control. In the SVM context, the NDMP user must have either the “vsadmin” or “vsadmin-backup” role. In a cluster context, the NDMP user must have either the “admin” or “backup” role.

Apart from these pre-defined roles, a user account associated with a custom role can also be used for NDMP authentication provided that the custom role has the “vserver services ndmp” folder in its command directory and the access level of the folder is not “none”. In this mode, you must generate an NDMP password for a given user account, which is created through role-based access control. Cluster users in an admin or backup role can access a node-management LIF, a cluster-management LIF, or an intercluster LIF. Users in a vsadmin-backup or vsadmin role can access only the data LIF for that SVM. Therefore, depending on the role of a user, the availability of volumes and tape devices for backup and restore operations vary.

This mode also supports user authentication for NIS and LDAP users. Therefore, NIS and LDAP users can access multiple SVMs with a common user ID and password. However, NDMP authentication does not support Active Directory users.

In this mode, a user account must be associated with the SSH application and the “User password” authentication method.

.Related information

[Commands for managing SVM-scoped NDMP mode](#)

[System administration](#)

[ONTAP concepts](#)

= Generate an NDMP-specific password for NDMP users

:icons: font

:relative\_path: ./tape-backup/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

In the storage virtual machine (SVM)-scoped NDMP mode, you must generate a password for a specific user ID. The generated password is based on the actual login password for the NDMP user. If the actual login password changes, you must generate the NDMP-specific password again.

| Stage| Action

a|

1

a|

For less than full volume or full qtree backups, ONTAP traverses directories to identify the files to be backed up. If you are backing up an entire volume or qtree, ONTAP combines this stage with Stage 2.

a|

2

a|

For a full volume or full qtree backup, ONTAP identifies the directories in the volumes or qtrees to be backed up.

a|

3

a|

ONTAP writes the directories to tape.

a|

4

a|

ONTAP writes the files to tape.

a|

5

a|

ONTAP writes the ACL information (if applicable) to tape.

The dump backup uses a Snapshot copy of your data for the backup. Therefore, you do not have to take the volume offline before initiating the backup.

The dump backup names each Snapshot copy it creates as `snapshot_for_backup.n`, where `n` is an integer starting at 0. Each time the dump backup creates a Snapshot copy, it increments the integer by 1. The integer is reset to 0 after the storage system is rebooted. After the backup operation is completed, the dump engine deletes this Snapshot copy.

When ONTAP performs multiple dump backups simultaneously, the dump engine creates multiple Snapshot copies. For example, if ONTAP is running two dump backups simultaneously, you find the following Snapshot copies in the volumes from which data is being backed up: `snapshot_for_backup.0` and `snapshot_for_backup.1`.

#### [NOTE]

=====

When you are backing up from a Snapshot copy, the dump engine does not create an additional Snapshot copy.

=====

= Types of data that the dump engine backs up

:icons: font

:relative\_path: ./tape-backup/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

#### [.lead]

The dump engine enables you to back up data to tape to guard against disasters or controller disruptions. In addition to backing up data objects such as a files, directories, qtrees, or entire volumes, the dump engine can back up many types of information about each file. Knowing the types of data that the dump engine can back up and the restrictions to take into consideration can help you plan your approach to disaster recovery.

In addition to backing up data in files, the dump engine can back up the following information about each file, as applicable:

- \* UNIX GID, owner UID, and file permissions
- \* UNIX access, creation, and modification time
- \* File type
- \* File size
- \* DOS name, DOS attributes, and creation time
- \* Access control lists (ACLs) with 1,024 access control entries (ACEs)
- \* Qtree information
- \* Junction paths

Junction paths are backed up as symbolic links.

- \* LUN and LUN clones

+

You can back up an entire LUN object; however, you cannot back up a single file within the LUN object. Similarly, you can restore an entire LUN object but not a single file within the LUN.

+

#### [NOTE]

=====

| Backup order| Increment level| Increment chain| Base| Files backed up

a|  
1  
a|  
0  
a|  
Both  
a|  
Files on the storage system  
a|  
All files in the backup path  
a|  
2  
a|  
2  
a|  
0, 2, 3  
a|  
Level-0 backup  
a|  
Files in the backup path created since the level-0 backup  
a|  
3  
a|  
3  
a|  
0, 2, 3  
a|  
Level-2 backup  
a|  
Files in the backup path created since the level-2 backup  
a|  
4  
a|  
1  
a|  
0, 1, 4  
a|  
Level-0 backup, because this is the most recent level that is lower than the level-1 backup  
a|  
Files in the backup path created since the level-0 backup, including files that are in the level-2 and level-3  
backups  
a|  
5  
a|  
4  
a|  
0, 1, 4  
a|  
The level-1 backup, because it is a lower level and is more recent than the level-0, level-2, or level-3 backups  
a|  
Files created since the level-1 backup



= What the blocking factor is

:icons: font

:relative\_path: ./tape-backup/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

A tape block is 1,024 bytes of data. During a tape backup or restore, you can specify the number of tape blocks that are transferred in each read/write operation. This number is called the *blocking factor*.

You can use a blocking factor from 4 to 256. If you plan to restore a backup to a system other than the system that did the backup, the restore system must support the blocking factor that you used for the backup. For example, if you use a blocking factor of 128, the system on which you restore that backup must support a blocking factor of 128.

During an NDMP backup, the `MOVER_RECORD_SIZE` determines the blocking factor. ONTAP allows a maximum value of 256 KB for `MOVER_RECORD_SIZE`.

= When to restart a dump backup

:icons: font

:relative\_path: ./tape-backup/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

A dump backup sometimes does not finish because of internal or external errors, such as tape write errors, power outages, accidental user interruptions, or internal inconsistency on the storage system. If your backup fails for one of these reasons, you can restart it.

You can choose to interrupt and restart a backup to avoid periods of heavy traffic on the storage system or to avoid competition for other limited resources on the storage system, such as a tape drive. You can interrupt a long backup and restart it later if a more urgent restore (or backup) requires the same tape drive. Restartable backups persist across reboots. You can restart an aborted backup to tape only if the following conditions are true:

- \* The aborted backup is in phase IV.
- \* All of the associated Snapshot copies that were locked by the dump command are available.
- \* The file history must be enabled.

When such a dump operation is aborted and left in a restartable state, the associated Snapshot copies are locked. These Snapshot copies are released after the backup context is deleted. You can view the list of backup contexts by using the `vserver services ndmp restartable backup show` command.

----

```
cluster::> vserver services ndmpd restartable-backup show
Vserver Context Identifier Is Cleanup Pending?
```

```
-----
vserver1 330e6739-0179-11e6-a299-005056bb4bc9 false
vserver1 481025c1-0179-11e6-a299-005056bb4bc9 false
vserver2 5cf10132-0179-11e6-a299-005056bb4bc9 false
3 entries were displayed.
```

	Stage	Action
--	-------	--------

a	1	
---	---	--

a	2	
---	---	--

a	3	ONTAP catalogs the files that need to be extracted from the tape.
---	---	---

a	4	
---	---	--

a	5	
---	---	--

a	6	ONTAP creates directories and empty files.
---	---	--

a	7	
---	---	--

a	8	
---	---	--

a	9	ONTAP reads a file from tape, writes it to disk, and sets the permissions (including ACLs) on it.
---	---	---

a	10	
---	----	--

a	11	
---	----	--

a	12	ONTAP repeats stages 2 and 3 until all the specified files are copied from the tape.
---	----	--

= Types of data that the dump engine restores

:icons: font

:relative\_path: ./tape-backup/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

When a disaster or controller disruption occurs, the dump engine provides multiple methods for you to recover all of the data that you backed up, from single files, to file attributes, to entire directories. Knowing the types of data that dump engine can restore and when to use which method of recovery can help minimize downtime.

You can restore data to an online mapped LUN. However, host applications cannot access this LUN until the restore operation is complete. After the restore operation is complete, the host cache of the LUN data should be flushed to provide coherency with the restored data.

The dump engine can recover the following data:

- \* Contents of files and directories

- \* UNIX file permissions

- \* ACLs

+

If you restore a file that has only UNIX file permissions to an NTFS qtree or volume, the file has no Windows NT ACLs. The storage system uses only the UNIX file permissions on this file until you create a Windows NT ACL on it.

+

[NOTE]

====

If you restore ACLs backed up from storage systems running Data ONTAP 8.2 to storage systems running Data ONTAP 8.1.x and earlier that have an ACE limit lower than 1,024, a default ACL is restored.

====

- \* Qtree information

+

Qtree information is used only if a qtree is restored to the root of a volume. Qtree information is not used if a qtree is restored to a lower directory, such as /vs1/vol1/subdir/lowerdir, and it ceases to be a qtree.

- \* All other file and directory attributes

- \* Windows NT streams

- \* LUNs

**A LUN must be restored to a volume level or a qtree level for it to remain as a LUN.**

+

**If it is restored to a directory, it is restored as a file because it does not contain any valid metadata.**

A 7-Mode LUN is restored as a LUN on an ONTAP volume.

- \* A 7-Mode volume can be restored to an ONTAP volume.

- \* VM-aligned files restored to a destination volume inherit the VM-align properties of the destination volume.

- \* The destination volume for a restore operation might have files with mandatory or advisory locks.

+

While performing restore operation to such a destination volume, the dump engine ignores these locks.

= Considerations before restoring data

| System memory of a storage system| Total number of dump backup and restore sessions

a|

Less than 16 GB

a|

4

a|

Greater than or equal to 16 GB but less than 24 GB

a|

16

a|

Greater than or equal to 24 GB

a|

32

[NOTE]

=====

If you use `ndmpcopy` command to copy data within storage systems, two NDMP sessions are established, one for dump backup and the other for dump restore.

=====

You can obtain the system memory of your storage system by using the `sysconfig -a` command (available through the nodeshell). For more information about using this command, see the man pages.

.Related information

[Scalability limits for NDMP sessions](#)

= Tape backup and restore support between Data ONTAP operating in 7-Mode and ONTAP

:icons: font

:relative\_path: ./tape-backup/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

You can restore data backed up from a storage system operating in 7-Mode or running ONTAP to a storage system either operating in 7-Mode or running ONTAP.

The following tape backup and restore operations are supported between Data ONTAP operating in 7-Mode and ONTAP:

- \* Backing up a 7-Mode volume to a tape drive connected to a storage system running ONTAP
- \* Backing up an ONTAP volume to a tape drive connected to a 7-Mode system
- \* Restoring backed-up data of a 7-Mode volume from a tape drive connected to a storage system running ONTAP
- \* Restoring backed-up data of an ONTAP volume from a tape drive connected to a 7-Mode system
- \* Restoring a 7-Mode volume to an ONTAP volume

+

[NOTE]

=====

....

- A 7-Mode LUN is restored as a LUN on an ONTAP volume.

- You should retain the ONTAP LUN identifiers when restoring a 7-Mode LUN to an existing ONTAP LUN.

....

=====

- \* Restoring an ONTAP volume to a 7-Mode volume

+

[NOTE]

=====

An ONTAP LUN is restored as a regular file on a 7-Mode volume.

=====

= Delete restartable contexts

| If you are performing tape backup and restore operations in the...| Then...

a|

storage virtual machine (SVM) scoped NDMP mode when CAB extension is supported by the backup application

a|

You can continue performing incremental tape backup and restore operations on read/write and read-only volumes without reconfiguring backup policies.

a|

SVM-scoped NDMP mode when CAB extension is not supported by the backup application

a|

You can continue performing incremental tape backup and restore operations on read/write and read-only volumes if you migrate the LIF configured in the backup policy to the node that hosts the destination aggregate. Otherwise, after the volume move, you must perform a baseline backup before performing the incremental backup operation.

a|

Node-scoped NDMP mode

## [NOTE]

=====

When a volume move occurs, if the volume belonging to a different SVM on the destination node has the same name as that of the moved volume, then you cannot perform incremental backup operations of the moved volume.

=====

.Related information

[ONTAP concepts](#)

= How dump works when a FlexVol volume is full

:icons: font

:relative\_path: ./tape-backup/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

Before performing an incremental dump backup operation, you must ensure that there is sufficient free space in the FlexVol volume.

If the operation fails, you must increase the free space in the Flex Vol volume either by increasing its size or by deleting the Snapshot copies. Then perform the incremental backup operation again.

= How dump works when volume access type changes

:icons: font

:relative\_path: ./tape-backup/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

When a SnapMirror destination volume or a SnapVault secondary volume changes state from read/write to read-only or from read-only to read/write, you must perform a baseline tape backup or restore operation.

SnapMirror destination and SnapVault secondary volumes are read-only volumes. If you perform tape backup and restore operations on such volumes, you must perform a baseline backup or restore operation whenever the volume changes state from read-only to read/write or from read/write to read-only.

.Related information

[ONTAP concepts](#)

= How dump works with SnapMirror single file or LUN restore

:icons: font

:relative\_path: ./tape-backup/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

| System memory of the storage system| Total number of SMTape backup and restore sessions

a|

Less than 16 GB

a|

6

a|

Greater than or equal to 16 GB but less than 24 GB

a|

16

a|

Greater than or equal to 24 GB

a|

32



You can obtain the system memory of your storage system by using the `sysconfig -a` command (available through the nodeshell). For more information about using this command, see the man pages.

.Related information

[Scalability limits for NDMP sessions](#)

[Scalability limits for dump backup and restore sessions](#)

= What tape seeding is

:icons: font

:relative\_path: ./tape-backup/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

Tape seeding is an SMTape functionality that helps you initialize a destination FlexVol volume in a data protection mirror relationship.

Tape seeding enables you to establish a data protection mirror relationship between a source system and a destination system over a low-bandwidth connection.

Incremental mirroring of Snapshot copies from the source to the destination is feasible over a low bandwidth connection. However, an initial mirroring of the base Snapshot copy takes a long time over a low-bandwidth connection. In such cases, you can perform an SMTape backup of the source volume to a tape and use the tape to transfer the initial base Snapshot copy to the destination. You can then set up incremental SnapMirror updates to the destination system using the low-bandwidth connection.

.Related information

[ONTAP concepts](#)

= How SMTape works with storage failover and ARL operations

:icons: font

:relative\_path: ./tape-backup/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

Before you perform SMTape backup or restore operations, you should understand how these operations work with storage failover (takeover and giveback) or aggregate relocation (ARL) operation. The `-override-vetoes` option determines the behavior of SMTape engine during a storage failover or ARL operation.

When an SMTape backup or restore operation is running and the `-override-vetoes` option is set to `false`, a user-initiated storage failover or ARL operation is stopped and the backup or restore operation complete. If the backup application supports CAB extension, then you can continue performing incremental SMTape backup and restore operations without reconfiguring backup policies. However, if the `-override-vetoes` option is set to `true`, then the storage failover or ARL operation is continued and the SMTape backup or restore operation is aborted.

Type	Description
log	Logging event
dmp	Dump event
rst	Restore event

\* `timestamp` shows the date and time of the event.

\* The `identifier` field for a dump event includes the dump path and the unique ID for the dump. The `identifier` field for a restore event uses only the restore destination path name as a unique identifier. Logging-related event messages do not include an `identifier` field.

= What logging events are

:icons: font

:relative\_path: ./tape-backup/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

The event field of a message that begins with a log specifies the beginning of a logging or the end of a logging.

It contains one of the events shown in the following table:

[options="header"]

Event	Description
Start_Logging	Indicates the beginning of logging or that logging has been turned back on after being disabled.
Stop_Logging	Indicates that logging has been turned off.

= What dump events are

:icons: font

:relative\_path: ./tape-backup/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

The event field for a dump event contains an event type followed by event-specific information within parentheses.

The following table describes the events, their descriptions, and the related event information that might be recorded for a dump operation:

[options="header"]

Event	Description	Event information
-------	-------------	-------------------

a		
---	--	--

Start		
-------	--	--

a		
---	--	--

NDMP dump is started		
----------------------	--	--

a		
---	--	--

Dump level and the type of dump		
---------------------------------	--	--

a		
---	--	--

End		
-----	--	--

a		
---	--	--

Dumps completed successfully		
------------------------------	--	--

a		
---	--	--

Amount of data processed		
--------------------------	--	--

a		
---	--	--

Abort		
-------	--	--

a		
---	--	--

The operation is cancelled		
----------------------------	--	--

a		
---	--	--

Amount of data processed		
--------------------------	--	--

a		
---	--	--

Options		
---------	--	--

a		
---	--	--

Specified options are listed		
------------------------------	--	--

a		
---	--	--

All options and their associated values, including NDMP options		
---	--	--

a		
---	--	--

Tape_open		
-----------	--	--

a		
---	--	--

The tape is open for read/write		
---------------------------------	--	--

a		
---	--	--

The new tape device name		
--------------------------	--	--

a		
---	--	--

Tape_close		
------------	--	--

a		
---	--	--

The tape is closed for read/write		
-----------------------------------	--	--

a		
---	--	--

The tape device name		
----------------------	--	--

a		
---	--	--

Phase-change		
--------------	--	--

a|  
A dump is entering a new processing phase  
a|  
The new phase name  
a|  
Error  
a|  
A dump has encountered an unexpected event  
a|  
Error message  
a|  
Snapshot  
a|  
A Snapshot copy is created or located  
a|  
The name and time of the Snapshot copy  
a|  
Base\_dump  
a|  
A base dump entry in the internal metafile has been located  
a|  
The level and time of the base dump (for incremental dumps only)

= What restore events are  
:icons: font  
:relative\_path: ./tape-backup/  
:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/  
  
[.lead]  
The event field for a restore event contains an event type followed by event-specific information in parentheses.  
  
The following table provides information about the events, their descriptions, and the related event information that can be recorded for a restore operation:  
  
[options="header"]

Event	Description	Event information
a	Start	
a	NDMP restore is started	
a	Restore level and the type of restore	
a	End	
a	Restores completed successfully	
a	Number of files and amount of data processed	
a	Abort	
a		

The operation is cancelled  
a|  
Number of files and amount of data processed  
a|  
Options  
a|  
Specified options are listed  
a|  
All options and their associated values, including NDMP options  
a|  
Tape\_open  
a|  
The tape is open for read/write  
a|  
The new tape device name  
a|  
Tape\_close  
a|  
The tape is closed for read/write  
a|  
The tape device name  
a|  
Phase-change  
a|  
Restore is entering a new processing phase  
a|  
The new phase name  
a|  
Error  
a|  
Restore encounters an unexpected event  
a|  
Error message

:leveloffset: -1  = Enabling or disabling event logging :icons: font :relative_path: ./tape-backup/ :imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/  [.lead] You can turn the event logging on or off.  .Steps  . To enable or disable event logging, enter the following command at the clustershell: + `*options -option_name backup.log.enable -option-value {on	off}` + on turns event logging on. + off turns event logging off. + [NOTE] ==== Event logging is turned on by default. ====  :leveloffset: -1  = Error messages for tape backup and restore of FlexVol volumes  :leveloffset: +1  = Backup and restore error messages  :leveloffset: +1  = Resource limitation: no available thread :icons: font :relative_path: ./tape-backup/ :imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/  * <b>Message</b> + Resource	"NDMP4_ADDR_TCP_IPv6"] not supported for ["IPv6"	"IPv4"] control connections :icons: font :relative_path: ./tape-backup/ :imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/  * <b>Message</b> + `Data connection type ["NDMP4_ADDR_TCP"	"NDMP4_ADDR_TCP_IPv6"] not supported for ["IPv6"	"IPv4"] control connections`  * <b>Cause</b> + In node-scoped NDMP mode, the NDMP data connection established must be of the same network address type (IPv4 or IPv6) as the NDMP control connection.  * <b>Corrective action</b> + Contact your backup application vendor.  = DATA LISTEN: CAB data connection prepare precondition error :icons: font :relative_path: ./tape-backup/ :imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/  * <b>Message</b> + DATA LISTEN: CAB data connection prepare precondition error
---	---	--	---	--	--

h| Policy h| Policy Type h| Update behavior

a|

MirrorLatest

a|

async-mirror

a|

Transfer the Snapshot copy created by SnapMirror.

a|

MirrorAndVault

a|

mirror-vault

a|

Transfer the Snapshot copy created by SnapMirror and any less recent Snapshot copies made since the last update, provided they have SnapMirror labels “daily” or “weekly”.

a|

Unified7year

a|

mirror-vault

a|

Transfer the Snapshot copy created by SnapMirror and any less recent Snapshot copies made since the last update, provided they have SnapMirror labels “daily”, “weekly”, or “monthly”.

## [NOTE]

=====

For complete background information on SnapMirror policies, including guidance on which policy to use, see [Data Protection](#).

=====

### === Understanding SnapMirror labels

Every policy with the “mirror-vault” policy type must have a rule that specifies which Snapshot copies to replicate. The rule “daily”, for example, indicates that only Snapshot copies assigned the SnapMirror label “daily” should be replicated. You assign the SnapMirror label when you configure Element Snapshot copies.

### === Replication from an Element source cluster to an ONTAP destination cluster

You can use SnapMirror to replicate Snapshot copies of an Element volume to an ONTAP destination system. In the event of a disaster at the Element site, you can serve data to clients from the ONTAP system, then reactivate the Element source volume when service is restored.

An Element volume is roughly equivalent to an ONTAP LUN. SnapMirror creates a LUN with the name of the Element volume when a data protection relationship between Element software and ONTAP is initialized. SnapMirror replicates data to an existing LUN if the LUN meets the requirements for Element to ONTAP replication.

Replication rules are as follows:

- \* An ONTAP volume can contain data from one Element volume only.
- \* You cannot replicate data from an ONTAP volume to multiple Element volumes.

### === Replication from an ONTAP source cluster to an Element destination cluster

Beginning with ONTAP 9.4, you can replicate Snapshot copies of a LUN created on an ONTAP system back to an Element volume:

- \* If a SnapMirror relationship already exists between an Element source and an ONTAP destination, a LUN created while you are serving data from the destination is automatically replicated when the source is reactivated.
- \* Otherwise, you must create and initialize a SnapMirror relationship between the ONTAP source cluster and the Element destination cluster.

Replication rules are as follows:

- \* The replication relationship must have a policy of type “async-mirror”.
- +
- Policies of type “mirror-vault” are not supported.
- \* Only iSCSI LUNs are supported.
- \* You cannot replicate more than one LUN from an ONTAP volume to an Element volume.
- \* You cannot replicate a LUN from an ONTAP volume to multiple Element volumes.

### === Prerequisites

You must have completed the following tasks before configuring a data protection relationship between Element and ONTAP:



h| Resource or feature h| Support details

a|  
SnapMirror  
a|

- The SnapMirror restore feature is not supported.
- The `MirrorAllSnapshots` and `XDPDefault` policies are not supported.
- The “vault” policy type is not supported.
- The system-defined rule “all\_source\_snapshots” is not supported.
- The “mirror-vault” policy type is supported only for replication from Element software to ONTAP. Use “async-mirror” for replication from ONTAP to Element software.
- The `-schedule` and `-prefix` options for `snapmirror policy add-rule` are not supported.
- The `-preserve` and `-quick-resync` options for `snapmirror resync` are not supported.
- Storage efficiency is not preserved.
- Fan-out and cascade data protection deployments are not supported.

a|  
ONTAP  
a|

- ONTAP Select is supported beginning with ONTAP 9.4 and Element 10.3.
- Cloud Volumes ONTAP is supported beginning with ONTAP 9.5 and Element 11.0.

a|  
Element  
a|

- Volume size limit is 8 TiB.
- Volume block size must be 512 bytes. A 4K byte block size is not supported.
- Volume size must be a multiple of 1 MiB.
- Volume attributes are not preserved.
- Maximum number of Snapshot copies to be replicated is 30.

a|  
Network  
a|

- A single TCP connection is allowed per transfer.
- The Element node must be specified as an IP address. DNS hostname lookup is not supported.
- IPspaces are not supported.

a|  
SnapLock  
a|  
SnapLock volumes are not supported.  
a|

FlexGroup

a|

FlexGroup volumes are not supported.

a|

SVM DR

a|

ONTAP volumes in an SVM DR configuration are not supported.

a|

MetroCluster

a|

ONTAP volumes in a MetroCluster configuration are not supported.

= Workflow for replication between Element and ONTAP

:icons: font

:relative\_path: ./element-replication/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

Whether you are replicating data from Element to ONTAP or from ONTAP to Element, you need to configure a job schedule, specify a policy, and create and initialize the relationship. You can use a default or custom policy.

The workflow assumes that you have completed the prerequisite tasks listed in [Prerequisites](#). For complete background information on SnapMirror policies, including guidance on which policy to use, see [Data protection](#).

image::.../media/solidfire-to-ontap-backup-workflow.gif[]

= Enable SnapMirror in Element software

:leveloffset: +1

= Enable SnapMirror on the Element cluster

:icons: font

:relative\_path: ./element-replication/

:imagesdir: /tmp/d20230427-16143-1lwdbyr/source/./data-protection/./media/

[.lead]

You must enable SnapMirror on the Element cluster before you can create a replication relationship. You can perform this task in the Element software web UI only.

.Before you begin

\* The Element cluster must be running NetApp Element software version 10.1 or later.

\* SnapMirror can only be enabled for Element clusters used with NetApp ONTAP volumes.

.About this task

The Element system comes with SnapMirror disabled by default. SnapMirror is not automatically enabled as part of a new installation or upgrade.

[NOTE]

====

Once enabled, SnapMirror cannot be disabled. You can only disable the SnapMirror feature and restore the default settings by returning the cluster to the factory image.

====

.Steps

. Click **Clusters > Settings**.

. Find the cluster-specific settings for SnapMirror.

h| Policy type h| Relationship type

a|  
async-mirror  
a|  
SnapMirror DR  
a|  
mirror-vault  
a|  
Unified replication

<p>.Step</p> <p>. Create a custom replication policy:</p> <p>+</p> <pre>`snapmirror policy create -vserver SVM -policy policy -type async-mirror</pre>	<pre>mirror-vault -comment comment -tries transfer_tries -transfer -priority low</pre>	<pre>normal -is-network -compression-enabled true</pre>	<pre>false` + For complete command syntax, see the man page. + Beginning with ONTAP 9.5, you can specify the schedule for creating a common Snapshot copy schedule for SnapMirror Synchronous relationships by using the -common -snapshot-schedule parameter. By default, the common Snapshot copy schedule for SnapMirror Synchronous relationships is one hour. You can specify a value from 30 minutes to two hours for the Snapshot copy schedule for SnapMirror Synchronous relationships. + The following example creates a custom replication policy for SnapMirror DR that enables network compression for data transfers: + ----- cluster_dst::&gt; snapmirror policy create -vserver svm1 -policy DR_compressed -type async-mirror -comment "DR with network compression enabled" -is -network-compression -enabled true ----- + The following example creates a custom replication policy for unified replication: + ----- cluster_dst::&gt; snapmirror policy create -vserver svm1 -policy my_unified</pre>
--	--	---	--

h| System-defined rule h| Used in policy types h| Result

a|  
sm\_created

a|  
async-mirror, mirror-vault

a|  
A Snapshot copy created by SnapMirror is transferred on initialization and update.

a|  
daily

a|  
mirror-vault

a|  
New Snapshot copies on the source with the SnapMirror label “daily” are transferred on initialization and update.

a|  
weekly

a|  
mirror-vault

a|  
New Snapshot copies on the source with the SnapMirror label “weekly” are transferred on initialization and update.

a|  
monthly

a|  
mirror-vault

a|  
New Snapshot copies on the source with the SnapMirror label “monthly” are transferred on initialization and update.

