



# **Manage SnapMirror SVM replication**

## **ONTAP 9**

NetApp  
June 04, 2024

# Table of Contents

- Manage SnapMirror SVM replication . . . . . 1
  - About SnapMirror SVM replication . . . . . 1
  - Replicate SVM configurations . . . . . 8
  - Serve data from an SVM DR destination . . . . . 19
  - Reactivate the source SVM . . . . . 22
  - Convert volume replication relationships to an SVM replication relationship . . . . . 32
  - Delete an SVM replication relationship . . . . . 33

# Manage SnapMirror SVM replication

## 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 disaster 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.

For more information about SnapMirror unified replication, see [SnapMirror unified replication basics](#).

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 disaster recovery 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 disaster recovery 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
---------------------	-----------------

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>
Relationship types	<ul style="list-style-type: none"> <li>• SnapMirror disaster recovery</li> <li>• Beginning with ONTAP 9.2, SnapMirror unified replication</li> </ul>
Replication scope	Intercluster only. You cannot replicate SVMs in the same cluster.
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>
Consistency groups asynchronous support	Beginning with ONTAP 9.14.1, a maximum of 32 SVM disaster recovery relationships are supported when consistency groups exist. See <a href="#">Protect a consistency group</a> and <a href="#">Consistency group limits</a> for more information.
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 disaster recovery relationship within a MetroCluster configuration can act as a source for additional SVM disaster recovery configurations.</p> <p>Beginning with ONTAP 9.5, SnapMirror SVM replication is supported on MetroCluster configurations.</p> <ul style="list-style-type: none"> <li>• In releases earlier than ONTAP 9.10.X, a MetroCluster configuration cannot be the destination of an SVM disaster recovery relationship.</li> <li>• In ONTAP 9.10.1 and later releases, a MetroCluster configuration can be the destination of an SVM disaster recovery relationship for migration purposes only, and it must meet all necessary requirements described in <a href="#">TR-4966: Migrating a SVM into a MetroCluster solution</a>.</li> <li>• Only an active SVM within a MetroCluster configuration can be the source of an SVM disaster recovery 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 disaster recovery relationship, since the volumes are not online.</li> <li>• When the sync-source SVM is the source of an SVM disaster recovery relationship, the source SVM disaster recovery relationship information is replicated to the MetroCluster partner.</li> <li>• During the switchover and switchback processes, replication to the SVM disaster recovery destination might fail.</li> </ul> <p>However, after the switchover or switchback process completes, the next SVM disaster recovery scheduled updates will succeed.</p>
Consistency group	Supported beginning with ONTAP 9.14.1. For more information, see <a href="#">Protect a consistency group</a> .
ONTAP S3	Not supported with SVM disaster recovery.
SnapMirror Synchronous	Not supported with SVM disaster recovery.

Version-independence	Not supported.
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>

## Configurations replicated in SVM disaster recovery relationships

The following table shows the interaction of the `snapmirror create -identity-preserve` option and the `snapmirror policy create -discard-configs network` 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
	Service 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 and Snapshot 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 disaster recovery storage limits

The following table shows the recommended maximum number of volumes and SVM disaster recovery 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 disaster relationships

## 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.

You should determine which replication workflow best suits your needs:

- [Replicate an entire SVM configuration](#)
- [Exclude LIFs and related network settings from SVM replication](#)
- [Exclude network, name service, and other settings from SVM configuration](#)

## 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 minimum supported schedule (RPO) for FlexVol volumes in an SVM SnapMirror relationship is 15 minutes. The minimum supported schedule (RPO) for FlexGroup volumes in an SVM SnapMirror relationship is 30 minutes.

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 minimum supported schedule (RPO) for FlexVol volumes in an SVM SnapMirror relationship is 15 minutes. The minimum supported schedule (RPO) for FlexGroup volumes in an SVM SnapMirror relationship is 30 minutes.

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 svml -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 svml -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 svml: -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 svml: -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.

### 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.

### 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](#).

### 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 minimum supported schedule (RPO) for FlexVol volumes in an SVM SnapMirror relationship is 15 minutes. The minimum supported schedule (RPO) for FlexGroup volumes in an SVM SnapMirror relationship is 30 minutes.

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
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
```

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.

### Before you begin

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 -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  
-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:
```

## Copyright information

Copyright © 2024 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

## Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.