



# **Configure a replication relationship one step at a time**

ONTAP 9

NetApp  
May 16, 2022

# Table of Contents

- Configure a replication relationship one step at a time ..... 1
  - Create a destination volume ..... 1
  - Create a replication job schedule ..... 1
  - Customize a replication policy ..... 2
  - Create a replication relationship ..... 6
  - Initialize a replication relationship ..... 9
  - Example: Configure a vault-vault cascade ..... 10

# 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 svm1 -policy
my_snapvault -type vault
```

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

```
cluster_dst::> snapmirror policy create -vserver svm1 -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 svm1 -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 svm1 -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 svm1 -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 svm1 -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 svml -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
```

## Copyright Information

Copyright © 2022 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.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

## 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.