Backup protection with cloud targets

ONTAP 9

NetApp
February 27, 2023

Table of Contents

Backup protection with cloud targets

Requirements for cloud target relationships

Make sure that your source and target environments meet the requirements for S3 SnapMirror backup protection to cloud targets.

You must have valid account credentials with the object store provider to access the data bucket.

Intercluster network interfaces and an IPspace should be configured on the cluster before the cluster can connect to a cloud object store. You should create enter cluster network interfaces on each node to seamlessly transfer data from the local storage to the cloud object store.

For StorageGRID targets, you need to know the following information:

- server name, expressed as a fully-qualified domain name (FQDN) or IP address
- bucket name; the bucket must already exist
- access key
- secret key

In addition, the CA certificate used to sign the StorageGRID server certificate needs to be installed on the ONTAP S3 cluster's admin storage VM using the `security certificate install command`. For more information, see Installing a CA certificate if you use StorageGRID.

For AWS S3 targets, you need to know the following information:

- server name, expressed as a fully-qualified domain name (FQDN) or IP address
- bucket name; the bucket must already exist
- access key
- secret key

The DNS server for the ONTAP cluster's admin storage VM must be able to resolve FQDNs (if used) to IP addresses.

Create a backup relationship for a new bucket (cloud target)

When you create new S3 buckets, you can back them up immediately to an S3 SnapMirror target bucket on an object store provider, which can be a StorageGRID system or an AWS S3 deployment.

What you'll need

- You have valid account credentials and configuration information for the object store provider.
- Intercluster network interfaces and an IPspace have been configured on the source system.
- • The DNS configuration for the source storage VM must be able to resolve the target's FQDN.

System Manager procedure

1. Edit the storage VM to add users, and to add users to groups:
   a. Click **Storage > storage VMs**, click the storage VM, click **Settings** and then click ✎ under **S3**.

      See Add S3 users and groups for more information.

2. Add a Cloud Object Store on the source system:
   a. Click **Protection > Overview**, then select **Cloud Object Stores**.
   b. Click **Add**, then select **Amazon S3** or **StorageGRID**.
   c. Enter the following values:
      - Cloud object store name
      - URL style (path or virtual-hosted)
      - storage VM (enabled for S3)
      - Object store server name (FQDN)
      - Object store certificate
      - Access key
      - Secret key
      - Container (bucket) name
3. Create an S3 SnapMirror policy if you don't have an existing one and you don't want to use the default policy:
   a. Click **Protection > Overview**, and then click **Local Policy Settings**.
   b. Click → next to **Protection Policies**, then click **Add**.
      - Enter the policy name and description.
      - Select the policy scope, cluster or SVM
      - Select **Continuous** for S3 SnapMirror relationships.
      - Enter your **Throttle** and **Recovery Point Objective** values.
4. Create a bucket with SnapMirror protection:
   a. Click **Storage > Buckets**, then click **Add**.
   b. Enter a name, select the storage VM, enter a size, then click **More Options**.
   c. Under **Permissions**, click **Add**. Verifying permissions is optional but recommended.
      - **Principal** and **Effect** - select values corresponding to your user group settings or accept the defaults.
      - **Actions** - make sure the following values are shown: `GetObject,PutObject,DeleteObject,ListBucket,GetBucketAcl,GetObjectAcl,ListBucketMultipartUploads,ListMultipartUploadParts`
      - **Resources** - use the defaults `_(bucketname, bucketname/*)` or other values you need.

        See Manage user access to buckets for more information about these fields.

   d. Under **Protection**, check **Enable SnapMirror (ONTAP or Cloud)**, select **Cloud Storage**, then select the **Cloud Object Store**.

When you click **Save**, a new bucket is created in the source storage VM, and it is backed up to the cloud object store.

CLI procedure

1. If this is the first S3 SnapMirror relationship for this SVM, verify that root user keys exist for both source and destination SVMs and regenerate them if they do not:
   `vserver object-store-server user show`

   Confirm that there is an access key for the root user. If there is not, enter:
   `vserver object-store-server user regenerate-keys -vserver svm_name -user root`

   Do not regenerate the key if one already exists.

2. Create a bucket in the source SVM:
   `vserver object-store-server bucket create -vserver svm_name -bucket bucket_name [-size integer[KB|MB|GB|TB|PB]] [-comment text] [additional_options]`

3. Add access rules to the default bucket policy:
   `vserver object-store-server bucket policy add-statement -vserver svm_name -bucket bucket_name -effect {allow|deny} -action object_store_actions -principal user_and_group_names -resource object_store_resources [-sid text] [-index integer]`

   Example

   ```
   clusterA::> vserver object-store-server bucket policy add-statement
   -bucket test-bucket -effect allow -action
   GetObject,PutObject,DeleteObject,ListBucket,GetBucketAcl,GetObjectAcl,Li
   stBucketMultipartUploads,ListMultipartUploadParts -principal - -resource
   test-bucket, test-bucket /*
   ```

4. Create an S3 SnapMirror policy if you don’t have an existing one and you don’t want to use the default policy:
   `snapmirror policy create -vserver svm_name -policy policy_name -type continuous [-rpo integer] [-throttle throttle_type] [-comment text] [additional_options]`

   Parameters:
   * `type continuous` – the only policy type for S3 SnapMirror relationships (required).
   * `-rpo` – specifies the time for recovery point objective, in seconds (optional).
   * `-throttle` – specifies the upper limit on throughput/bandwidth, in kilobytes/seconds (optional).

   Example

   ```
   clusterA::> snapmirror policy create -vserver vs0 -type continuous -rpo
   0 -policy test-policy
   ```

5. If the target is a StorageGRID system, install the StorageGRID CA server certificate on the admin SVM of the source cluster:

`security certificate install -type server-ca -vserver src_admin_svm -cert-name storage_grid_server_certificate`

See the `security certificate install` man page for details.

6. Define the S3 SnapMirror destination object store:
`snapmirror object-store config create -vserver svm_name -object-store-name target_store_name -usage data -provider-type {AWS_S3|SGWS} -server target_FQDN -container-name remote_bucket_name -is-ssl-enabled true -port port_number -access-key target_access_key -secret-password target_secret_key`

   Parameters:
   * `-object-store-name` – the name of the object store target on the local ONTAP system.
   * `-usage` – use `data` for this workflow.
   * `-provider-type` – `AWS_S3` and `SGWS` (StorageGRID) targets are supported.
   * `-server` – the target server's FQDN or IP address.
   * `-is-ssl-enabled` –enabling SSL is optional but recommended.

   See the `snapmirror object-store config create` man page for details.

   Example

   ```
   src_cluster::> snapmirror object-store config create -vserver vs0
   -object-store-name sgws-store -usage data -provider-type SGWS -server
   sgws.example.com -container-name target-test-bucket -is-ssl-enabled true
   -port 443 -access-key abc123 -secret-password xyz890
   ```

7. Create an S3 SnapMirror relationship:
`snapmirror create -source-path svm_name:/bucket/bucket_name -destination-path object_store_name:/objstore -policy policy_name`

   Parameters:
   * `-destination-path` – the object store name you created in the previous step and the fixed value `objstore`.

   You can use a policy you created or accept the default.

   Example

   ```
   src_cluster::> snapmirror create -source-path vs0:/bucket/test-bucket
   -destination-path sgws-store:/objstore -policy test-policy
   ```

8. Verify that mirroring is active:
`snapmirror show -policy-type continuous -fields status`

Create a backup relationship for an existing bucket (cloud target)

You can begin backing up existing S3 buckets at any time; for example, if you upgraded

an S3 configuration from a release earlier than ONTAP 9.10.1.

What you'll need

- You have valid account credentials and configuration information for the object store provider.
- Intercluster network interfaces and an IPspace have been configured on the source system.
- The DNS configuration for the source storage VM must be able to resolve the target's FQDN.

System Manager procedure

1. Verify that the users and groups are correctly defined:
   Click **Storage > storage VMs**, click the storage VM, click **Settings** and then click ✎ under S3.

   See Add S3 users and groups for more information.

2. Create an S3 SnapMirror policy if you don't have an existing one and you don't want to use the default policy:
   a. Click **Protection > Overview**, and then click **Local Policy Settings**.
   b. Click → next to **Protection Policies**, then click **Add**.
   c. Enter the policy name and description.
   d. Select the policy scope, cluster or SVM
   e. Select **Continuous** for S3 SnapMirror relationships.
   f. Enter your **Throttle** and **Recovery Point Objective values**.
3. Add a Cloud Object Store on the source system:
   a. Click **Protection > Overview**, then select **Cloud Object Store**.
   b. Click **Add**, then select **Amazon S3** or **Others** for StorageGRID Webscale.
   c. Enter the following values:
      - Cloud object store name
      - URL style (path or virtual-hosted)
      - storage VM (enabled for S3)
      - Object store server name (FQDN)
      - Object store certificate
      - Access key
      - Secret key
      - Container (bucket) name
4. Verify that the bucket access policy of the existing bucket still meets your needs:
   a. Click **Storage** > **Buckets** and then select the bucket you want to protect.
   b. In the **Permissions** tab, click ✎ **Edit**, then click **Add** under **Permissions**.
      - **Principal** and **Effect** - select values corresponding to your user group settings or accept the defaults.
      - **Actions** - make sure the following values are shown:
        `GetObject,PutObject,DeleteObject,ListBucket,GetBucketAcl,GetObjectAcl,ListBucketMultipartUploads,ListMultipartUploadParts`

- **Resources** - use the defaults *(bucketname, bucketname/*)* or other values you need.

  See Manage user access to buckets for more information about these fields.

5. Back up the bucket using S3 SnapMirror:
   a. Click **Storage** > **Buckets** and then select the bucket you want to back up.
   b. Click **Protect**, select **Cloud Storage** under **Target**, then select the **Cloud Object Store**.

When you click **Save**, the existing bucket is backed up to the cloud object store.

CLI procedure

1. Verify that the access rules in the default bucket policy are correct:
   `vserver object-store-server bucket policy add-statement -vserver svm_name -bucket bucket_name -effect {allow|deny} -action object_store_actions -principal user_and_group_names -resource object_store_resources [-sid text] [-index integer]`

   Example

   ```
   clusterA::> vserver object-store-server bucket policy add-statement
   -bucket test-bucket -effect allow -action
   GetObject,PutObject,DeleteObject,ListBucket,GetBucketAcl,GetObjectAcl,Li
   stBucketMultipartUploads,ListMultipartUploadParts -principal - -resource
   test-bucket, test-bucket /*
   ```

2. Create an S3 SnapMirror policy if you don't have an existing one and you don't want to use the default policy:
   `snapmirror policy create -vserver svm_name -policy policy_name -type continuous [-rpo integer] [-throttle throttle_type] [-comment text] [additional_options]`

   Parameters:
   * `type continuous` – the only policy type for S3 SnapMirror relationships (required).
   * `-rpo` – specifies the time for recovery point objective, in seconds (optional).
   * `-throttle` – specifies the upper limit on throughput/bandwidth, in kilobytes/seconds (optional).

   Example

   ```
   clusterA::> snapmirror policy create -vserver vs0 -type continuous -rpo
   0 -policy test-policy
   ```

3. If the target is a StorageGRID system, install the StorageGRID CA certificate on the admin SVM of the source cluster:
   `security certificate install -type server-ca -vserver src_admin_svm -cert-name storage_grid_server_certificate`

   See the `security certificate install` man page for details.

4. Define the S3 SnapMirror destination object store:
`snapmirror object-store config create -vserver svm_name -object-store-name target_store_name -usage data -provider-type {AWS_S3|SGWS} -server target_FQDN -container-name remote_bucket_name -is-ssl-enabled true -port port_number -access-key target_access_key -secret-password target_secret_key`

   Parameters:
   * `-object-store-name` – the name of the object store target on the local ONTAP system.
   * `-usage` – use `data` for this workflow.
   * `-provider-type` – `AWS_S3` and `SGWS` (StorageGRID) targets are supported.
   * `-server` – the target server's FQDN or IP address.
   * `-is-ssl-enabled` –enabling SSL is optional but recommended.

   See the `snapmirror object-store config create` man page for details.

   Example

   ```
   src_cluster::> snapmirror object-store config create -vserver vs0
   -object-store-name sgws-store -usage data -provider-type SGWS -server
   sgws.example.com -container-name target-test-bucket -is-ssl-enabled true
   -port 443 -access-key abc123 -secret-password xyz890
   ```

5. Create an S3 SnapMirror relationship:
`snapmirror create -source-path svm_name:/bucket/bucket_name -destination-path object_store_name:/objstore -policy policy_name`

   Parameters:
   * `-destination-path` – the object store name you created in the previous step and the fixed value `objstore`.

   You can use a policy you created or accept the default.

   Example

   ```
   src_cluster::> snapmirror create -source-path vs0:/bucket/buck-evp
   -destination-path sgws-store:/objstore -policy test-policy
   ```

6. Verify that mirroring is active:
`snapmirror show -policy-type continuous -fields status`

Restore a bucket from a cloud target

When data in a source bucket is lost or corrupted, you repopulate your data by restoring from a destination bucket.

About this task

You can restore the destination bucket to an existing bucket or a new bucket. The target bucket for the restore operation must be larger than the destination bucket's logical used space.

If you use an existing bucket, it must be empty when starting a restore operation. Restore does not "roll back" a bucket in time; rather, it populates an empty bucket with its previous contents.

System Manager procedure

Restore the back-up data:

1. Click **Protection > Relationships**, then select **S3 SnapMirror**.
2. Click ⋮ and then select **Restore**.
3. Under **Source**, select **Existing Bucket** (the default) or **New Bucket**.
   - To restore to an **Existing Bucket** (the default), complete these actions:
     - Select the cluster and storage VM to search for the existing bucket.
     - Select the existing bucket.
     - Copy and paste the contents of the *destination* S3 server CA certificate.
   - To restore to a **New Bucket**, enter the following values:
     - The cluster and storage VM to host the new bucket.
     - The new bucket's name, capacity, and performance service level.
       See Storage service levels for more information.
     - The contents of the destination S3 server CA certificate.
4. Under **Destination**, copy and paste the contents of the *source* S3 server CA certificate.
5. Click **Protection > Relationships** to monitor the restore progress.

CLI procedure

1. If you are restoring to a new bucket, create the new bucket. For more information, see Create a backup relationship for a bucket (cloud target).
2. Initiate a restore operation for the destination bucket:
   `snapmirror restore -source-path` *`object_store_name`*`:/objstore -destination-path` *`svm_name`*`:/bucket/`*`bucket_name`*

   Example

   The following example restores a destination bucket to an existing bucket.

   ```
   clusterA::> snapmirror restore -source-path sgws.store:/objstore -destination
   -path vs0:/bucket/test-bucket
   ```

Copyright information

Trademark information