Tutorials

Cloud Sync

NetApp
February 12, 2024

This PDF was generated from https://docs.netapp.com/us-en/cloudsync/task_copying_acls.html on February 12, 2024. Always check docs.netapp.com for the latest.

Table of Contents

Tutorials 1
Copying ACLs 1
Syncing NFS data using data-in-flight encryption 3
Setting up the data broker to use an external HashiCorp Vault 8

Tutorials

Copying ACLs

Cloud Sync can copy access control lists (ACLs) between a source SMB share and a target SMB share, or between a source NFS server and target NFS server. If needed, you can manually preserve ACLs for SMB shares yourself by using robocopy.

Choices

- Set up Cloud Sync to automatically copy ACLs
- Manually copy ACLs between SMB shares

Setting up Cloud Sync to copy ACLs

Copy ACLs between SMB servers or between NFS servers by enabling a setting when you create a relationship or after you create a relationship.

What you'll need

- A new sync relationship or an existing sync relationship.

  For SMB shares, note that this feature is available for new sync relationships created after the 23 Feb 2020 release. If you'd like to use this feature with existing relationships created prior to that date, then you'll need to recreate the relationship.

- Any type of data broker.

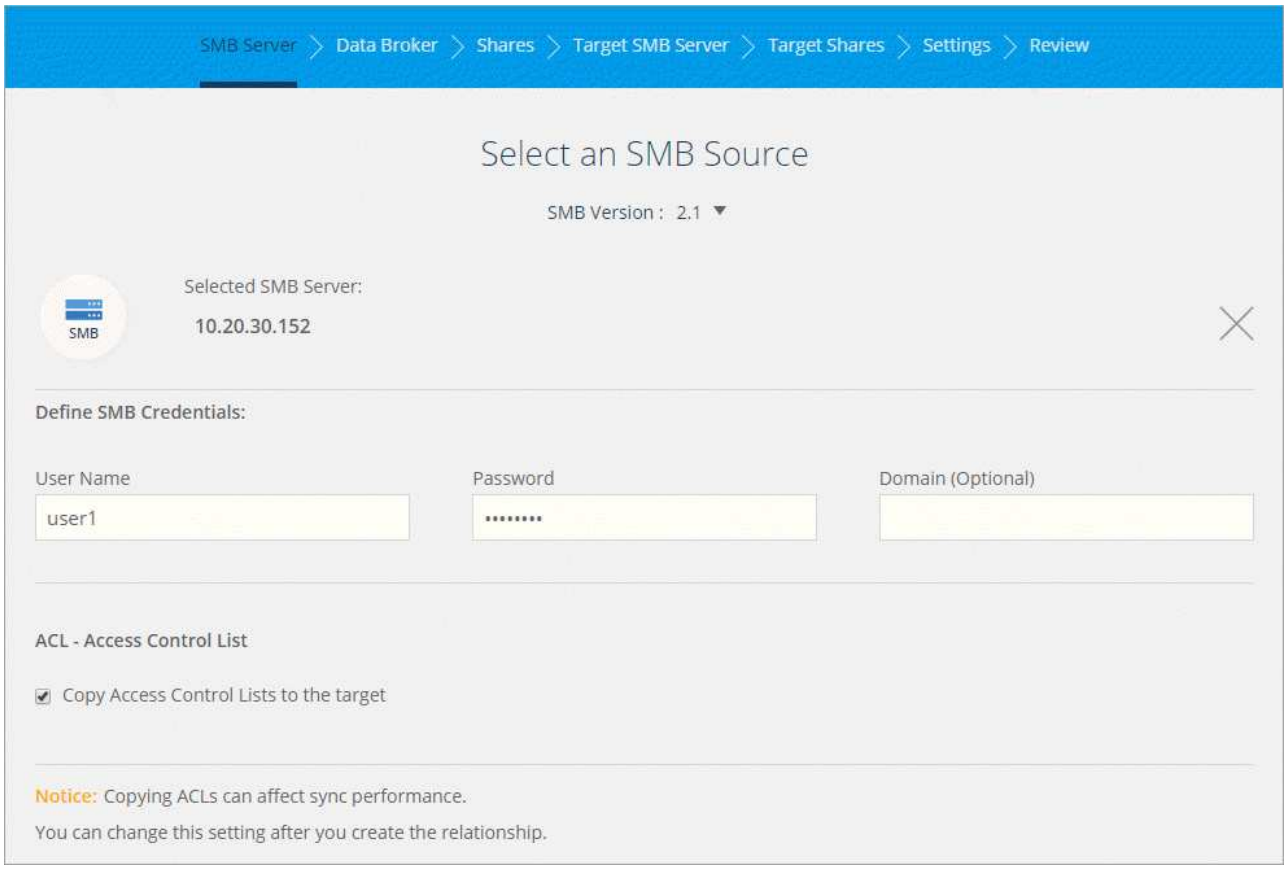
  This feature works with *any* type of data broker: the AWS, Azure, Google Cloud Platform, or on-prem data broker. The on-prem data broker can run any supported operating system.

- For NFS, you'll need to use version 4 or later.

  Copying ACLs isn't supported with NFS version 3.

Steps for a new relationship

1. From Cloud Sync, click **Create New Sync Relationship**.
2. Drag and drop **SMB Server** to the source and target or **NFS Server** to the source and target and click **Continue**.
3. On the **SMB Server** or **NFS Server** page:
   a. Enter a new server or select an existing server and click **Continue**.
   b. Select **Copy Access Control Lists to the target** and click **Continue**.



4. Follow the remaining prompts to create the sync relationship.

Steps for an existing relationship

1. Hover over the sync relationship and click the action menu.
2. Click **Settings**.
3. Select **Copy Access Control Lists to the target**.
4. Click **Save Settings**.

Result

When syncing data, Cloud Sync preserves the ACLs between the source and target servers.

Manually copying ACLs between SMB shares

You can manually preserve ACLs between SMB shares by using the Windows robocopy command.

Steps

1. Identify a Windows host that has full access to both SMB shares.
2. If either of the endpoints require authentication, use the **net use** command to connect to the endpoints from the Windows host.

   You must perform this step before you use robocopy.

3. From Cloud Sync, create a new relationship between the source and target SMB shares or sync an existing relationship.
4. After the data sync is complete, run the following command from the Windows host to sync the ACLs and

ownership:

```
robocopy /E /COPY:SOU /secfix [source] [target] /w:0 /r:0 /XD ~snapshots
/UNILOG:"[logfilepath]
```

Both *source* and *target* should be specified using the UNC format. For example: \\<server>\<share>\<path>

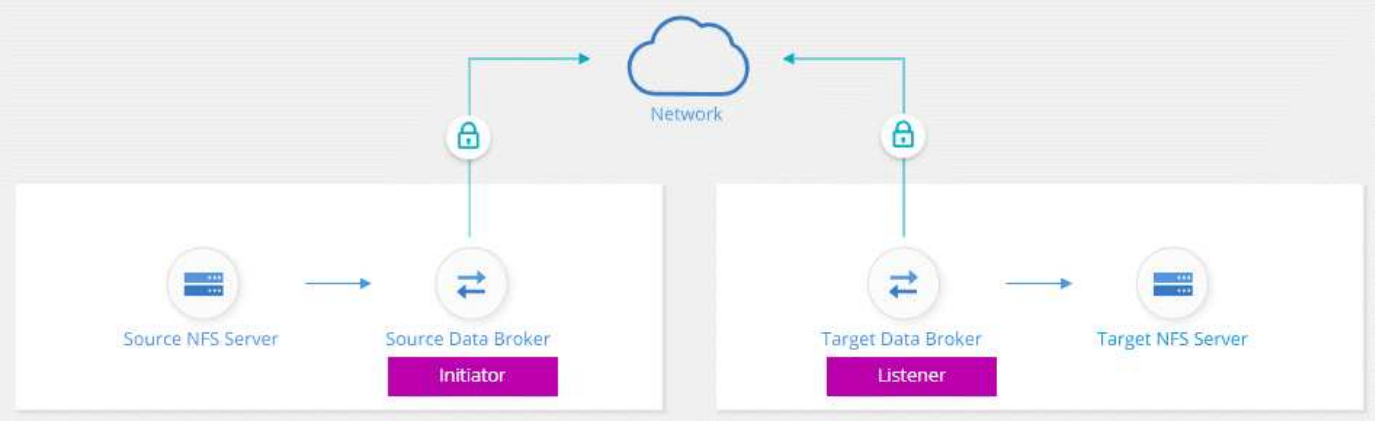
Syncing NFS data using data-in-flight encryption

If your business has strict security policies, you can sync NFS data using data-in-flight encryption. This feature is supported from an NFS server to another NFS server and from Azure NetApp Files to Azure NetApp Files.

For example, you might want to sync data between two NFS servers that are in different networks. Or you might need to securely transfer data on Azure NetApp Files across subnets or regions.

How data-in-flight encryption works

Data-in-flight encryption encrypts NFS data when it's sent over the network between two data brokers. The following image shows a relationship between two NFS servers and two data brokers:



One data broker functions as the *initiator*. When it's time to sync data, it sends a connection request to the other data broker, which is the *listener*. That data broker listens for requests on port 443. You can use a different port, if needed, but be sure to check that the port is not in use by another service.

For example, if you sync data from an on-premises NFS server to a cloud-based NFS server, you can choose which data broker listens for the connection requests and which sends them.

Here's how in-flight encryption works:

1. After you create the sync relationship, the initiator starts an encrypted connection with the other data broker.
2. The source data broker encrypts data from the source using TLS 1.3.
3. It then sends the data over the network to the target data broker.
4. The target data broker decrypts the data before sending it to the target.
5. After the initial copy, the service syncs any changed data every 24 hours. If there is data to sync, the process starts with the initiator opening an encrypted connection with the other data broker.

If you prefer to sync data more frequently, you can change the schedule after you create the relationship.

Supported NFS versions

- For NFS servers, data-in-flight encryption is supported with NFS versions 3, 4.0, 4.1, and 4.2.
- For Azure NetApp Files, data-in-flight encryption is supported with NFS versions 3 and 4.1.

Proxy server limitation

If you create an encrypted sync relationship, the encrypted data is sent over HTTPS and isn't routable through a proxy server.

What you'll need to get started

Be sure to have the following:

- Two NFS servers that meet source and target requirements or Azure NetApp Files in two subnets or regions.
- The IP addresses or fully qualified domain names of the servers.
- Network locations for two data brokers.

  You can select an existing data broker but it must function as the initiator. The listener data broker must be a *new* data broker.

  If you have not yet deployed a data broker, review the data broker requirements. Because you have strict security policies, be sure to review the networking requirements, which includes outbound traffic from port 443 and the internet endpoints that the data broker contacts.

  - Review AWS installation
  - Review Azure installation
  - Review GCP installation
  - Review Linux host installation

Syncing NFS data using data-in-flight encryption

Create a new sync relationship between two NFS servers or between Azure NetApp Files, enable the in-flight encryption option, and follow the prompts.

Steps

1. Log in to NetApp Cloud Central and select Cloud Sync.
2. Click **Create New Sync Relationship**.
3. Drag and drop **NFS Server** to the source and target locations or **Azure NetApp Files** to the source and target locations and select **Yes** to enable data-in-flight encryption.

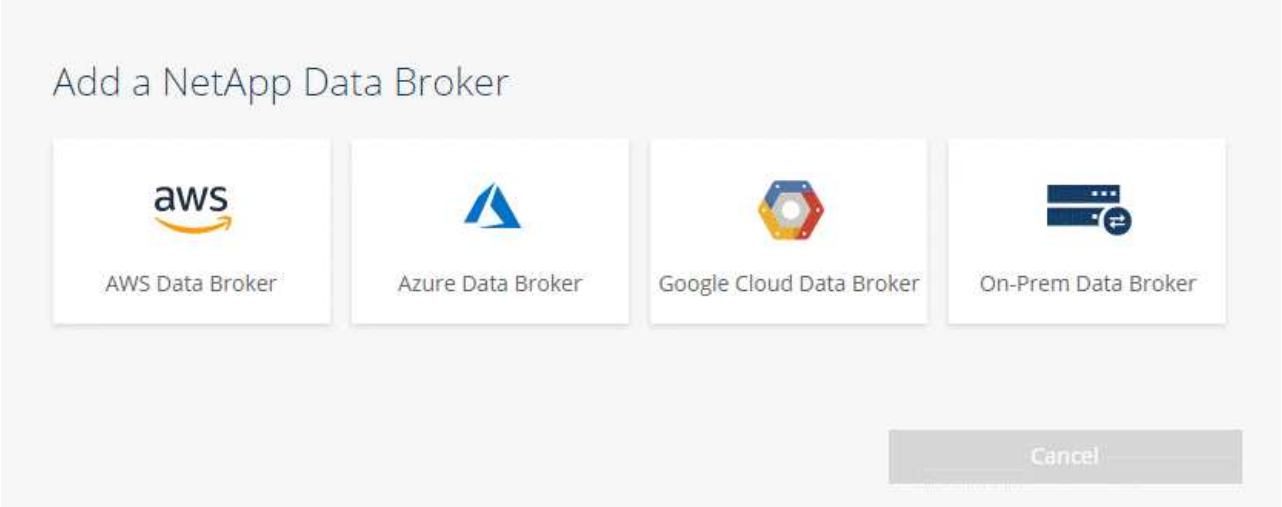
   The following image shows what you'd select to sync data between two NFS servers:

The following image shows what you'd select to sync data between Azure NetApp Files:

4. Follow the prompts to create the relationship:
   a. **NFS Server/Azure NetApp Files**: Choose the NFS version and then specify a new NFS source or select an existing server.
   b. **Define Data Broker Functionality**: Define which data broker *listens* for connection requests on a port and which one *initiates* the connection. Make your choice based on your networking requirements.
   c. **Data Broker**: Follow the prompts to add a new source data broker or select an existing data broker.

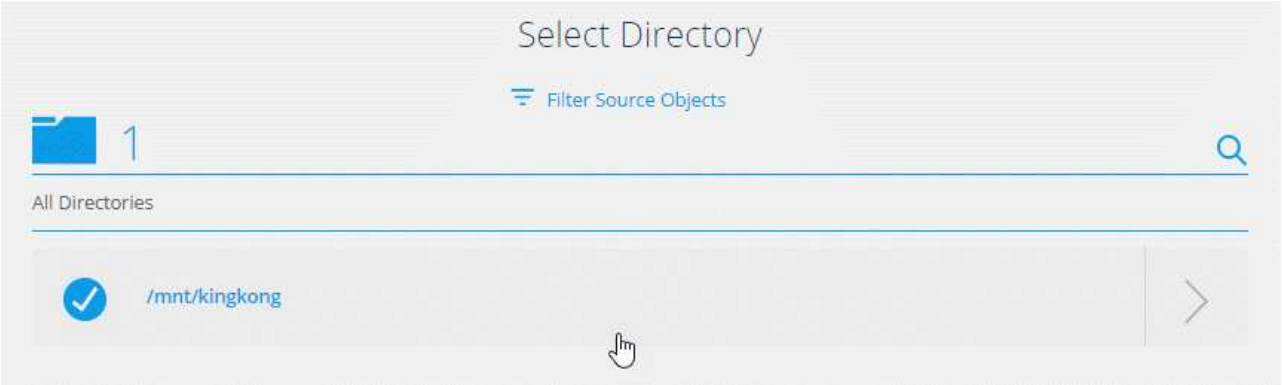
      If the source data broker acts as the listener, then it must be a new data broker.

      If you need a new data broker, Cloud Sync prompts you with the installation instructions. You can deploy the data broker in the cloud or download an installation script for your own Linux host.



d. **Directories**: Choose the directories that you want to sync by selecting all directories, or by drilling down and selecting a subdirectory.

   Click **Filter Source Objects** to modify settings that define how source files and folders are synced and maintained in the target location.

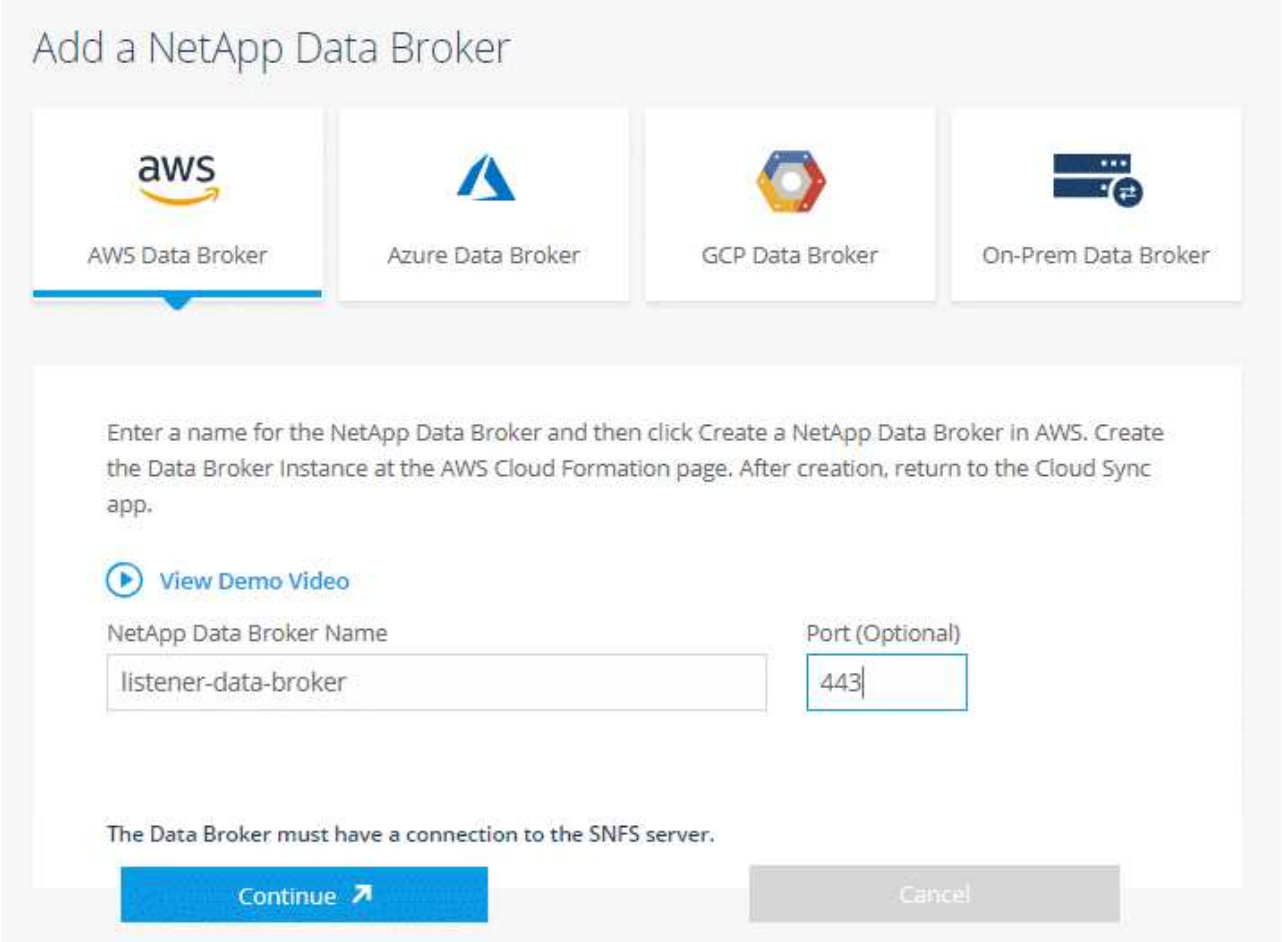


e. **Target NFS Server/Target Azure NetApp Files**: Choose the NFS version and then enter a new NFS target or select an existing server.

f. **Target Data Broker**: Follow the prompts to add a new source data broker or select an existing data broker.

   If the target data broker acts as the listener, then it must be a new data broker.

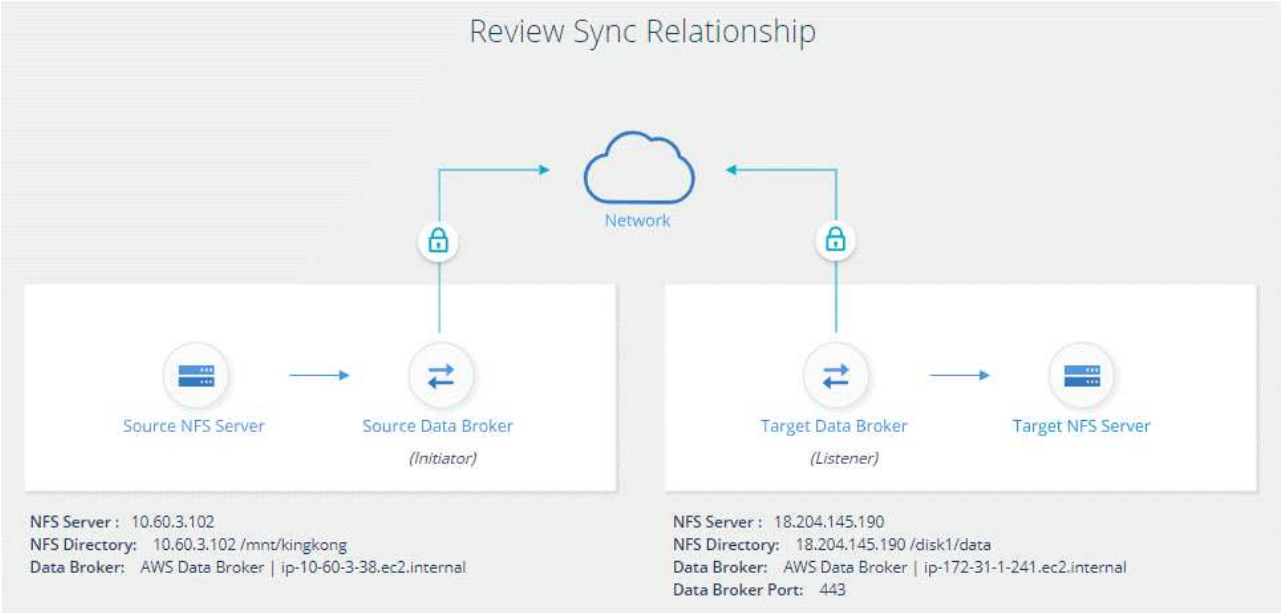
   Here's an example of the prompt when the target data broker functions as the listener. Notice the option to specify the port.



g. **Target Directories**: Select a top-level directory, or drill down to select an existing subdirectory or to create a new folder inside an export.

h. **Settings**: Define how source files and folders are synced and maintained in the target location.

i. **Review**: Review the details of the sync relationship and then click **Create Relationship**.



Result

Cloud Sync starts creating the new sync relationship. When it's done, click **View in Dashboard** to view details about the new relationship.

Setting up the data broker to use an external HashiCorp Vault

When you create a sync relationship that requires Amazon S3, Azure, or Google Cloud credentials, you need to specify those credentials through the Cloud Sync user interface or API. An alternative is to set up the data broker to access the credentials (or *secrets*) directly from an external HashiCorp Vault.

This feature is supported through the Cloud Sync API with sync relationships that require Amazon S3, Azure, or Google Cloud credentials.

Prepare the vault

Prepare the vault to supply credentials to the data broker by setting up the URLs. The URLs to the secrets in the vault must end with *Creds*.

Prepare the data broker

Prepare the data broker to fetch credentials from the external vault by modifying the local config file for the data broker.

Create a sync relationship using the API

Now that everything is set up, you can send an API call to create a sync relationship that uses your vault to get the secrets.

Preparing the vault

You'll need to provide Cloud Sync with the URL to the secrets in your vault. Prepare the vault by setting up those URLs. You need to set up URLs to the credentials for each source and target in the sync relationships that you plan to create.

The URL must be set up as follows:

```
/<path>/<requestid>/<endpoint-protocol>Creds
```

Path

The prefix path to the secret. This can be any value that's unique to you.

Request ID

A request ID that you need to generate. You'll need to provide the ID in one of the headers in the API POST request when you create the sync relationship.

Endpoint protocol

One of the following protocols, as defined in the post relationship v2 documentation: S3, AZURE, or GCP (each must be in uppercase).

Creds

The URL must end with *Creds*.

Examples

The following examples show URLs to secrets.

Example for the full URL and path for source credentials

http://example.vault.com:8200/my-path/all-secrets/hb312vdasr2/S3Creds

As you can see in the example, the prefix path is */my-path/all-secrets/*, the request ID is *hb312vdasr2* and the source endpoint is S3.

Example for the full URL and path for target credentials

http://example.vault.com:8200/my-path/all-secrets/n32hcbnejk2/AZURECreds

The prefix path is */my-path/all-secrets/*, the request ID is *n32hcbnejk2*, and the target endpoint is Azure.

Preparing the data broker

Prepare the data broker to fetch credentials from the external vault by modifying the local config file for the data broker.

Steps

1. SSH to the data broker.
2. Edit the local.json file that resides in /opt/netapp/databroker/config.
3. Set enable to **true** and set the config parameter fields under *external-integrations.hashicorp* as follows:

   enabled
   - Valid values: true/false
   - Type: Boolean
   - Default value: false
   - True: The data broker gets secrets from your own external HashiCorp Vault
   - False: The data broker stores credentials in its local vault

   url
   - Type: string
   - Value: The URL to your external vault

   path
   - Type: string
   - Value: Prefix path to the secret with your credentials

   Reject-unauthorized
   - Determines if you want the data broker to reject unauthorized external vault
   - Type: Boolean
   - Default: false

auth-method

- Your authentication method to the external vault
- Type: string
- Valid values: “aws-iam” / “role-app”

role-name

- Type: string
- Your role name (in case you use aws-iam)

Secretid & rootid

- Type: string (in case you use app-role)

Namespace

- Type: string
- Your namespace (X-Vault-Namespace header if needed)

Example

```
{
        “external-integrations”: {
                “hashicorp”: {
                        “enabled”: true,
                        “url”: “https://example.vault.com:8200”,
                        “path”: ““my-path/all-secrets”,
                        “reject-unauthorized”: false,
                        “auth-method”: “aws-role”,
                        “aws-role”: {
                              “role-name”: “my-role”
                        }
                }
        }
}
```

Creating a new sync relationship using secrets form the vault

Now that everything is set up, you can send an API call to create a sync relationship that uses your vault to get the secrets.

Post the relationship using the Cloud Sync REST API.

```
Headers:
Authorization: Bearer <user-token>
Content-Type: application/json
x-account-id: <accountid>
x-netapp-external-request-id-src: request ID as part of path for source
credentials
x-netapp-external-request-id-trg: request ID as part of path for target
credentials
Body: post relationship v2 body
```

- To obtain a user token and your Cloud Central account ID, refer to this page in the documentation.
- To build a body for your post relationship, refer to the relationships-v2 API call.

Example

Example for the POST request:

```
url: https://api.cloudsync.netapp.com/api/relationships-v2
headers:
"x-account-id": "CS-SasdW"
"x-netapp-external-request-id-src": "hb312vdasr2"
"Content-Type": "application/json"
"Authorization": "Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6Ik…"
Body:
{
"dataBrokerId": "5e6e111d578dtyuu1555sa60",
"source": {
        "protocol": "s3",
        "s3": {
                "provider": "sgws",
                "host": "1.1.1.1",
                "port": "443",
                "bucket": "my-source"
     },
"target": {
        "protocol": "s3",
        "s3": {
                "bucket": "my-target-bucket"
        }
    }
}
```

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.