PowerScale OneFS 9.7.0.0 CloudPools Administration Guide

9.7.0.0

Notes, cautions, and warnings

NOTE: A NOTE indicates important information that helps you make better use of your product.

CAUTION: A CAUTION indicates either potential damage to hardware or loss of data and tells you how to avoid the problem.

WARNING: A WARNING indicates a potential for property damage, personal injury, or death.

Contents

Tables

1

Introduction to this guide

Topics:

* About this guide
* Where to get help

About this guide

This guide describes Dell Technologies PowerScale CloudPools, a licensed software module that works with the Dell Technologies PowerScale OneFS operating system. This guide describes how the CloudPools interface provides access to OneFS cloud configuration, operation, and management.

Where to get help

The Dell Technologies Support site contains important information about products and services including drivers, installation packages, product documentation, knowledge base articles, and advisories.

A valid support contract and account might be required to access all the available information about a specific Dell Technologies product or service.

Additional options for getting help

This section contains resources for getting answers to questions about PowerScale products.

Dell Technologies Support	• Contact Technical Support
Telephone support	• United States: 1-800-SVC-4EMC (1-800-782-4362) • Canada: 1-800-543-4782 • Worldwide: 1-312-725-5401 • Local phone numbers for a specific country or region are available at Contact Technical Support .
PowerScale OneFS Documentation Info Hubs	• PowerScale OneFS Info Hubs

CloudPools overview and setup

Topics:

Migration from previous versions

If you have existing CloudPool accounts from versions earlier than OneFS 8.2, there are migration considerations.

● For information about migrating from OneFS 8.x to either 8.2.2.x, 9.0.0.0, or 9.1.0.0. see Cloudpools - Upgrading 8.x to 8.2.2.x or later.

● If you are migrating from OneFS 8.2.x or 9.0.0.0 to 9.1.0.0, there are no additional migration considerations.

CloudPools overview

CloudPools extends the capabilities of OneFS by moving data to lower-cost cloud storage. CloudPools can seamlessly connect to various cloud storage systems.

CloudPools is a licensed module that is built on the SmartPools file pool policy framework. CloudPools lets you move older or seldom-used data to cloud storage and free up space on your cluster. See Supported cloud providers and storage types for the list of cloud providers.

SmartPools provides granular control of file storage on a cluster. SmartPools uses a policy framework to separate files into logical groups called file pools, and to store those file pools in specific storage tiers.

CloudPools expands the SmartPools framework by treating a cloud repository as an additional storage tier.

CloudPools uses a similar workflow to SmartPools. To store files in the cloud, you must define at least one account with a cloud storage provider. In addition, you must configure OneFS for cloud storage, and create file pool policies that identify and move files to the cloud. You can configure CloudPools to move files to the cloud automatically, based on file pool policies. Alternatively, you can use OneFS commands to archive individual files to, or recall files from, the cloud.

A SmartPools job runs the file pool policies and sends matched files to the cloud storage target. The job typically runs once a day. For each file that is stored on the cloud, a small SmartLink file replaces the original file on the cluster. To access file data stored in the cloud, users open the related SmartLink file through any supported protocol (SMB, NFS, Swift, or HDFS). Accessing data through a SmartLink file is called inline access. To fully recall a file from the cloud, use the `isi cloud recall` command.

(i) **NOTE:** Changing the file policy will not recall already archived files from the cloud.

File pool policies define the data to archive	CloudPools uses the same mechanism as SmartPools uses to define data to archive. You define files to store in the cloud by creating file pool policies. These policies use file matching criteria to determine which files to move to the cloud. File pool policies are applied when the SmartPools system job runs. This job runs daily by default. When files match a file pool policy that contains CloudPools actions, OneFS moves the file data of matched files to the cloud. Only metadata and a SmartLink file remain on the cluster, freeing up storage space.
Files remain visible in OneFS	Although file data is moved to remote storage, the files remain visible in the OneFS file system. CloudPools accomplishes this access by retaining a local SmartLink file. The SmartLink file is a pointer to the location of data in the cloud. You can read, write, archive, and recall files from the cloud as needed. When a user views the OneFS file system through a supported protocol (SMB, NFS, Swift, or HDFS), SmartLink files appear to be the original files. When the user opens a SmartLink file, OneFS

automatically retrieves and caches data as needed from the cloud. This operation is called inline access. Any modifications the user makes to a file during inline access are applied to the data stored in the cloud. CloudPools provides a CLI command that fully recalls files from the cloud. Recall replaces the SmartLink files with the recalled files.

CloudPools concepts

CloudPools moves file data from PowerScale to the cloud. Users can access or recall these files as needed.

Following are descriptions of key CloudPools concepts that affect the end user:

Archive
The CloudPools process of moving file data to the cloud. This process extracts the data from the file and places it in one or more cloud objects. CloudPools then moves these objects to cloud storage, and leaves in place on the local cluster a representative file. This file is called a SmartLink file.

SmartLink file
For every file that is archived to the cloud, OneFS maintains an associated SmartLink file on the cluster. A SmartLink file contains metadata and map information which allows the data in the cloud to be accessed or fully recalled. If the SmartLink file archiving policy permits it, users can automatically retrieve and cache data from the cloud by accessing the SmarkLink file. Like other files, SmartLink files can be backed up through NDMP or synchronized to other clusters with SyncIQ. When SmartLink files are retrieved from a backup or SyncIQ operation, CloudPools maintains their links to related file data in the cloud.

Inline access
CloudPools enables users connecting to a cluster through supported protocols to access cloud data by opening associated SmartLink files. This process is called inline access. To the user connecting to OneFS through a supported protocol, a SmartLink file appears to be the original file. When the user opens a SmartLink file, CloudPools retrieves and caches cloud data locally. The user can view and edit the file as usual. CloudPools automatically retrieves and sends any updated file data back to the cloud so that the cloud contains the latest version.

> (i) **NOTE:** CloudPools offers inline access as a user convenience. However, CloudPools is designed mainly as an archival solution, and is not intended for storing data that is frequently updated. Such data should be left on the local cluster until it stabilizes and is ready for archival.

Recall
The CloudPools process of reversing the archival process. CloudPools replaces the SmartLink file with the original file data, and removes the cloud objects from cloud storage.

Following are the key concepts related to CloudPools configuration:

Cloud provider accounts
CloudPools requires you to set up one or more accounts with a cloud provider. See Supported cloud providers and storage types. You use the account information from the cloud provider to configure cloud accounts on the PowerScale cluster.

Cloud storage accounts
A cloud storage account is a OneFS entity that defines access to a specific cloud provider account. These accounts are used to enable and track local use of a cloud provider account. The cloud storage account configuration includes the cloud provider account credentials.

CloudPool
A CloudPool is a OneFS entity that contains a single cloud storage account and provides a conduit between OneFS and the cloud storage repository. Creating a CloudPool requires the availability of at least one cloud storage account. The cloud storage account must be of the same type as the CloudPool.

File pool policies
File pool policies are the essential control mechanism for both SmartPools and CloudPools. OneFS runs all file pool policies regularly. Each file pool policy specifies the files to manage, actions to take on the files, protection levels, and I/O optimization settings. The file pool policy also specifies the remote cloud account to archive files to, and how to handle files before archiving them. A policy can also specify where to store the locally retained SmartLink files.

Quickstart

To get started with CloudPools, follow these steps.
1. Perform initial setup tasks described in CloudPools setup tasks . Topics include licensing, setting up an optional proxy server, and reviewing and modifying the default configuration settings.
2. Choose your cloud provider, set up provider accounts, and define cloud storage pools. See Providers, CloudPool accounts, and storage pools.

3. Create file pool policies that define the files to archive and monitor the resulting file pool policy jobs. See File pool policies for CloudPools.
4. Access archived files as described in Retrieving file data from the cloud.

CloudPools setup tasks

The following tasks prepare your system for CloudPools.
- Obtain and activate a CloudPools software license
- Configure network proxy servers with CloudPools
- Configure CloudPools

Obtain and activate a CloudPools software license

The CloudPools software module requires a license.

Running CloudPools requires the activation of these software module licenses:

- SmartPools
- CloudPools

If your current OneFS license does not include CloudPools and SmartPools, contact Customer Support or submit a service request (SR). See the *PowerScale OneFS Web Administration Guide* for procedures to add software modules to your license, generate a new license activation file, and submit the activation file to the Dell EMC Software License Center (SLC).

A signed license file is returned to a responsible person in your organization through email. To activate the licenses, upload the signed license file to OneFS.. You can upload the file from either the web interface or the CLI, as described in the following topics.

Upload the updated license file (Web UI)

After you receive an updated license file from Dell Technologies Software Licensing Central (SLC), upload the updated file to the cluster.

Steps

1. Click **Cluster Management** > **Licensing**.
2. In the **Upload and activate a signed license file** area, click **Browse** and select the signed license file.
3. Click **Upload and Activate**.

Upload the updated license file (CLI)

After you receive a signed license file from Dell Technologies Software Licensing Central (SLC), upload the file to your cluster.

Steps

Run the `isi license add` command.

```
# isi license add --path <file-path-on-your-local-machine>
```

For example, the following command uploads the signed license file that you had previously copied into a directory named `/ifs` on your local machine:

```
isi license add --path /ifs/ISLN_15002_13-Feb-2019.xml
```

Configure network proxy servers with CloudPools

You can configure CloudPools so that data that is archived to, or recalled from, a public cloud provider is routed through a proxy server.

By default, CloudPools communicates directly with the designated cloud provider. If the cloud provider is private, the default communication protocol might be acceptable. Example private providers are another Dell Technologies PowerScale cluster or an ECS appliance running on the same corporate network.

However, when CloudPools archives data to a public cloud provider, such as Amazon S3, communication occurs through the public Internet. This protocol might violate your organizational security policies.

In a typical configuration, a PowerScale cluster is installed in a data center behind one or more firewalls. Ports that enable communication to the public Internet are closed. To enable CloudPools to archive data to a public cloud provider, you can configure it to work with a proxy server.

CloudPools works with proxy servers running the following protocols:

- SOCKS v4
- SOCKS v5
- HTTP

Configuration on the CloudPools side includes creating a network proxy entry and connecting the network proxy to a cloud storage account. Both SOCKS v5 and HTTP can be configured with or without authentication. SOCKS v4 does not support authentication.

From OneFS, you can also list network proxies, view network proxy properties, modify proxy settings, and delete proxies. Except for connecting the network proxy to a cloud storage account, you must use the CLI to run all other proxy server commands.

For details about creating a network proxy, see the *OneFS CLI Administration Guide* or the *OneFS Web Administration Guide*.

Create a network proxy (CLI)

You can create a network proxy to redirect CloudPools traffic to and from a public cloud provider. CloudPools supports proxy servers running the SOCKS v4, SOCKS v5, and HTTP protocols.

Prerequisites

The proxy server should be online and ready to accept a connection from the PowerScale cluster.

Steps

Run the `isi cloud proxies create` command.
The following command creates the proxy object `myproxy1`, and links it to a specific proxy server URL, proxy type, and port:

```
isi cloud proxies create myproxy1 10.99.58.250 socks_5 1080
```

Results

When you later create or modify a cloud storage account, the `myproxy1` network proxy is available. When you select the proxy, and save the changes, CloudPools verifies the proxy server connection.

View a list of network proxies (CLI)

You can view a list of existing network proxies in CloudPools.

Prerequisites

You or someone in your organization must first have created network proxies using the `isi cloud proxies create` command.

Steps

Run the `isi cloud proxies list` command.

The command displays a list of proxy names, hosts, and types.

View network proxy properties (CLI)

You can view the properties of a network proxy.

Prerequisites

You or someone in your organization must have created a network proxy using the `isi cloud proxies create` command.

Steps

Run the `isi cloud proxies view` command.
The following command displays the properties of a proxy named `myproxy1`:

```
isi cloud proxies view myproxy1
```

Properties shown include ID, name, host, type, and port.

Modify a network proxy (CLI)

You can modify the properties of an existing network proxy in CloudPools.

Prerequisites

You or someone in your organization must have created the network proxy using the `isi cloud proxies create` command.

Steps

Run the `isi cloud proxies modify` command.
The following command adds a user name and password necessary to connect to a network proxy:

```
isi cloud proxies modify myproxy1 --username cloud1 --password @xy16+RZ20
```

Results

You can now add the network proxy to a cloud storage account.

Delete a network proxy (CLI)

You can delete an existing network proxy in CloudPools. However, if the proxy is connected to a cloud storage account, you cannot delete the proxy.

Prerequisites

You or someone in your organization must have created the network proxy using the `isi cloud proxies create` command.

Steps

1. Run the `isi cloud proxies delete` command.
   The following command deletes the proxy named myproxy1:

   ```
   isi cloud proxies delete myproxy1
   ```

   OneFS asks you to confirm the deletion:

```
Are you sure? (yes/no):
```

2. Type yes and press ENTER.

Results

If the proxy is already connected to a cloud storage account in CloudPools, OneFS prevents you from deleting the proxy. Otherwise, the proxy is deleted.

Configure CloudPools

You can manage CloudPools default settings, including snapshot archival, encryption, compression, cache settings, data retention settings, and the ability to regenerate an encryption key. Regenerate an encryption key when you suspect that the existing key was compromised.

View CloudPools settings (CLI)

You can view the top-level configuration settings for CloudPools.

Steps

Run the `isi cloud settings view` command.

The command displays CloudPools settings such as accessibility, cache expiration, whether compression and encryption is enabled, and so on.

For explanations of the properties, see the isi cloud settings modify CLI reference page.

Modify default cloud settings (CLI)

You can modify default CloudPools settings.

About this task

Use the `isi cloud settings view` command to display current settings. Then change the settings with `isi cloud settings modify`, and verify the new setting with `isi cloud settings view`.

Steps

1. Run the `isi cloud settings modify` command.

    For example, the following command enables both encryption and compression of cloud data:

    ```
    isi cloud settings modify --default-encryption-enabled=yes
    --default-compression-enabled=yes
    ```

2. Verify the change.

    ```
    isi cloud settings modify
    ```

Generate a new master encryption key (CLI)

You can generate a new master encryption key. The key is used to encrypt data and is stored with cloud data objects.

Prerequisites

Only generate a new master encryption key if you believe the existing key has been compromised.

Steps

Run the `isi cloud settings regenerate-encryption-key` command.

The following command generates a new encryption key in verbose mode.

```
isi cloud settings regenerate-encryption-key --verbose
```

In verbose mode, the system confirms the process:

```
Encryption key has been regenerated
```

Providers, CloudPool accounts, and storage pools

Topics:

Supported cloud providers and storage types

CloudPools supports the following cloud providers and associated storage types.

Table 1. Supported Cloud Providers

Cloud Provider	Supported Storage Types
Dell Technologies PowerScale	All ⓘ **NOTE:** CloudPools only supports RAN. OneFS S3 is not supported.
Dell Technologies ECS Appliance	All
Amazon S3	S3 Standard storage classes ⓘ **NOTE:** CloudPools does not support Glacier.
Amazon C2S S3	S3 Standard storage classes
Microsoft Azure	Blob storage, hot access tiers ⓘ **NOTE:** CloudPools does not support cold blobs.
Google Cloud Platform	Standard, Nearline, and Coldline ⓘ **NOTE:** CloudPools does not support the Archive storage class.
Alibaba Cloud	Standard OSS

Dell Technologies PowerScale

CloudPools enables a Dell Technologies PowerScale cluster to function as a cloud storage provider.

In this scenario, a secondary PowerScale cluster provides a private cloud solution. The primary cluster archives files to the secondary cluster. Both clusters are managed in your corporate data center. The secondary cluster must be running a compatible version of OneFS.

To act as a cloud storage provider, the PowerScale cluster uses APIs that configure CloudPools policies, define cloud storage accounts, and retrieve cloud storage usage reports. These APIs are known collectively as the PowerScale Platform API, and are described in the *OneFS API Reference*.

To configure a secondary PowerScale cluster as the cloud storage repository, you must complete several tasks:

- On the secondary cluster:
  - Log in with system administrator privileges, and create a user.

- Create a role with access to Console, Platform API, License, Namespace Traverse, and Namespace Access privileges. Make the new user a member of this role.
- Log in as the new user, and create the directory where cloud data should be stored. For example: `/ifs/data/HQ-Archive`.
- On the primary cluster:
  - Create the cloud storage account with a provider type of `isilon`. Specify the new user credentials and the appropriate URI for the secondary cluster. Because the secondary cluster is within your corporate network, the URI looks similar to the following:

    ```
    https://10.1.210.310:8080/namespace/ifs/data/HQ-Archive
    ```
  - Create a CloudPool that refers to the new cloud storage account.

Dell EMC ECS Appliance

CloudPools supports ECS appliance as a cloud provider.

ECS is a complete software-defined cloud storage platform deployed on a turn-key appliance from Dell EMC. It supports the storage, manipulation, and analysis of unstructured data on a massive scale.

The ECS appliance is specifically designed to support mobile, cloud, big data, and next-generation applications. As an appliance, it is simple to install and deploy with support for multi-tenancy, self-service access, usage metering, on-demand cloud storage-as-a-service, and dynamic application provisioning.

Amazon S3

CloudPools can be configured to store data on Amazon Simple Storage Service (Amazon S3), a public cloud provider. CloudPools supports only S3 Standard storage classes.

CloudPools supports only S3 Standard storage classes on Amazon S3.

When you configure CloudPools to use Amazon S3 for cloud storage, you must specify the following attributes in the connection settings:

- URI, username, and passkey
- S3 Account ID
- S3 Storage Region
- S3 Telemetry Reporting Bucket

When you first establish an account with Amazon S3, the cloud provider gives you an account ID and allows you to choose a storage region. Amazon S3 offers multiple storage regions in the U.S. and other regions of the world.

(i) **NOTE:** CloudPools supports Amazon Web Services Signature V2 and V4 to authenticate queries to its cloud storage.

An S3 telemetry reporting bucket is required. The telemetry reporting bucket is where Amazon S3 stores billing reports. This bucket must be accessible to CloudPools.

To set up an S3 telemetry reporting bucket:

1. On the S3 console, go to **Billing & Cost Management** preferences.
2. Indicate that you want to receive billing reports.
3. Specify the bucket for storing the billing reports.

Amazon C2S S3

CloudPools can be configured to store data on Amazon C2S (Commercial Cloud Services) S3 (Simple Storage System).

When you configure CloudPools to use Amazon C2S S3 for cloud storage, in addition to URI, username, and passkey, you must specify the S3 Storage Region in the connection settings.

When you first establish an account with Amazon C2S, the cloud provider gives you an account ID and allows you to choose a storage region. Amazon C2S offers multiple storage regions in the U.S. and other regions of the world.

Additionally, connectivity to Amazon C2S S3 accounts requires that the credential server information is entered into the system.

Microsoft Azure

You can configure CloudPools to store data on Microsoft Azure, a public cloud provider.

CloudPools supports Blob storage, Hot access tiers on Microsoft Azure. Cold blobs are not supported.

When you establish an account with Microsoft Azure, you create a username. Microsoft provides you with a URI and a passkey. When you configure CloudPools to use Azure, you must specify the same URI, username, and passkey.

Google Cloud Platform

CloudPools can store data on Google Cloud Platform, a public cloud provider.

CloudPools supports Standard, Nearline, and Coldline storage types on Google Cloud Platform. Google Cloud Platform must be set to interoperability mode.

To configure Google Cloud Platform in interoperability mode, use the Google Cloud Platform interface as follows:

1. Log in to Google Cloud Platform.
2. On the main dashboard, choose **Storage** > **Settings** > **Interoperability**.
3. Follow the prompts to create an interoperable storage access key and secret.

You can now configure Google Cloud Platform as the provider in OneFS CloudPools. The following information is required:

- Google Cloud URI
- For username, use the Google Cloud interoperable storage access key.
- For key, use the Google Cloud secret for the access key.

You may also specify the storage region. Google offers multiple storage regions in the U.S. and other areas of the world. If you do not choose a storage region, the default storage region for Google Cloud Platform becomes the storage region for CloudPools.

Alibaba Cloud

CloudPools can store data on Alibaba Cloud, a public cloud provider.

CloudPools supports Standard OSS storage on Alibaba Cloud.

When configuring Alibaba Cloud as the provider, you must provide the Alibaba URI, username, and passkey. Alibaba offers multiple sites in the U.S. and other areas of the world. The URI indicates your chosen connection site.

Create and manage cloud storage accounts

A cloud storage account provides OneFS with the information it requires to connect to a remote cloud storage provider. You can create and edit one or more cloud storage accounts in OneFS.

Before creating a cloud storage account, you must establish an account with one of the supported cloud providers. OneFS attempts to connect to the cloud provider using the credentials you provide in the cloud storage account. To specify a proxy server with the cloud storage account, first create the proxy server with the `isi cloud proxies create` command.

After you create a cloud storage account, you create a CloudPool that uses the cloud storage account. A CloudPool is a storage container that OneFS can use to store data. OneFS enforces the following relationships between cloud storage accounts and CloudPools.

- Each cloud storage account can only belong to a single CloudPool storage container.
- A cloud storage account and the CloudPool must have matching types. The type values identify the cloud provider, such as `azure`.

Create a cloud storage account (Web UI)

You define cloud storage accounts in OneFS as an essential part of CloudPools configuration. The account username, password, and URI that you used to establish an account with your cloud provider is required. You can also specify a proxy server to redirect CloudPools archive and retrieval traffic to and from a public cloud provider.

Prerequisites

If you are creating an Amazon C2S S3 account, you must perform the following steps using the OneFS CLI before creating the account:

1. Import the CA certificate.

```
isi certificate authority import <certificate-path>
        [--name certificate_name]
        [--description  certificate_description]
```

2. Import the CAP Client Certificate and Private Key.

```
isi cloud certificates import <certificate-path> <certificate-key-path>
        [--name certificate_name]
        [--certificate-key-password <enter certificate password string>]
```

Steps

1. Click **File System** > **Storage Pools** > **CloudPools**.
2. Click **+ Create a Cloud Storage Account**.
3. In the **Create a Cloud Storage Account** dialog box, **Connection Settings**:
   a. Enter In the **Name or Alias** field, enter a name for the account.
   b. In the **Type** drop-down menu, select a type of cloud account. Choices are **Dell EMC PowerScale**, **Dell EMC ECS Appliance**, **Microsoft Azure**, **Amazon S3**, **Amazon C2S S3**, **Google Cloud Platform**, and **Alibaba Cloud**.
4. In the **Create a Cloud Storage Account** dialog box, complete the **Cloud account information**:

Table 2. Cloud account information

Field	Action	Required for
URI	Enter the fully qualified URI for the account. The URI must use the HTTPS protocol, and match the URI used to set up the account with your cloud provider.	All account types
Username	Enter the cloud provider account username. This username should have been set up with the cloud provider.	Dell EMC PowerScale Dell EMC ECS Appliance Microsoft Azure Google Cloud Platform Alibaba Cloud
Key	Enter the password or secret key that is associated with the cloud provider account username.	Dell EMC PowerScale Dell EMC ECS Appliance Microsoft Azure Google Cloud Platform Alibaba Cloud
Proxy	If you have defined one or more network proxies, and want to use one for this cloud account, select the name from the proxy.	All account types
Skip SSL certificate validation (not recommended).	Disable or prevent certificate validation.	All account types

Table 2. Cloud account information (continued)

Field	Action	Required for
Account ID	The account ID provided when the Amazon S3 account was created.	Amazon S3
Telemetry reporting bucket	The bucket on Amazon S3 that stores billing reports. Ensure that the bucket is configured to allow OneFS CloudPools to access it.	Amazon S3
Storage region	The region that stores the data. The provider assigns this value when creating the account.	Optional for: Amazon S3, Google Cloud Platform, Alibaba Cloud

5. For Amazon C2S S3 accounts only, complete the **Credential server information**:

Option	Description
URI	Enter a fully qualified URI for Amazon C2S S3 account credential server.
Agency	Agency name required to connect to an Amazon C2S S3 Access Portal (CAP or Token Vending Machine (TVM)).
Mission	Mission name required to connect to an Amazon C2S S3 Access Portal.
Role	Role name required to connect to an Amazon C2S S3 Access Portal.
Certificate	Name or id of a certificate to connect to an Amazon C2S S3 Access Portal. Import the certificate using the `isi certificate authority` and `isi cloud certificate import` commands.
Proxy	Name of id of a proxy to connect to an Amazon C2S S3 Access Portal. The proxy is created using `isi proxy create` CLI.

6. Click the **Connect Account** button.
   The **Create a Cloud Storage Account** dialog box closes, and the new cloud account appears in the **Cloud Storage Accounts** list. The Name, Type, State, Username, and URI associated with the account is displayed.

Create a cloud storage account (CLI)

You create cloud storage accounts to enable CloudPools to archive files to cloud storage. The account username, password, and URI that you used to establish an account with your cloud provider is required. You can also specify a proxy server to redirect CloudPools archive and retrieval traffic to and from a public cloud provider.

Steps

1. If you are creating an account for Amazon C2S S3 accounts, complete the following steps before creating the account, otherwise continue to step 2.
   a. Import the CA certificate.

   ```
   isi certificate authority import <certificate-path>
         [--name certificate_name]
         [--description  certificate_description]
   ```

   b. Import the CAP Client Certificate and Private Key.

   ```
   isi cloud certificates import <certificate-path> <certificate-key-path>
         [--name certificate_name]
         [--certificate-key-password <enter certificate password string>]
   ```

2. Run the `isi cloud accounts create` command.
   The following command creates a Microsoft Azure cloud storage account.

> (i) **NOTE:** This type of account requires a key from the cloud provider.

```
isi cloud accounts create --name=c-acct1 --type=azure
--uri=https://admin2.blob.core.windows.net --account-username=adm1
--key=!$P@$$c0de998==
```

The following command creates the same account, while specifying a proxy server.

```
isi cloud accounts create --name=c-acct1 --type=azure
--uri=https://admin2.blob.core.windows.net --account-username=adm1
--key=!$P@$$c0de998== --proxy myproxy1
```

The following command creates an Amazon C2S S3 account.

```
isi cloud accounts create --name=C2S3
 --credential-provider-uri=<your-info>
 --credential-provider-agency=<your-info>
 --credential-provider-certificate=<your-info>
 --credential-provider-mission=<your-info>
 --credential-provider-proxy=<your-info>
 --credential-provider-role=<your-info>
 --storage-region=<your-info>
```

Next steps

After the cloud storage account successfully connects to the cloud provider, you must add the cloud storage account to a CloudPool in OneFS. OneFS is then able to archive files to the cloud.

Edit a cloud storage account (Web UI)

You can edit some of the settings of an existing cloud storage account.

Steps

1. Click **File System** > **Storage Pools** > **CloudPools**.
2. In the **Cloud Storage Accounts** list, click the **View/Edit** button to the right of the account that you want to edit.
3. In the **View Cloud Storage Account Details** dialog box, click the **Edit Account** button.
4. In the **Edit Cloud Storage Account Details** dialog box, perform any of these actions:
   a. In the **Name or Alias** field, enter a new name for the account. You cannot change the type of account.
   b. In the **URI** field, enter a fully qualified URI for the account. The URI must use the HTTPS protocol, and match the URI used to set up the account with your cloud provider.
   c. In the **User Name** field, enter the account user name, which must be the same as the user name provided to the cloud provider.
   d. In the **Key** field, enter the account password. The password must be the same as the password that you provided to the cloud provider, or the key that the cloud provider issued to you.
   e. If you want to use a different proxy server for this cloud account, select the name of the new proxy from the **Proxy** drop-down box.
   f. If you are editing an Amazon S3 account, you can also specify a new `Account ID` and `Telemetry Reporting Bucket`. You cannot change the `Storage Region`.
5. Click the **Save Changes** button.
   CloudPools validates that your cloud data is still accessible. Otherwise, it alerts you and does not save the changes.

Modify a cloud storage account (CLI)

You can modify the information associated with a cloud storage account.

About this task

To modify a cloud storage account, you must specify the account name. You can run the `isi cloud accounts list` command to list cloud storage accounts.

Steps

Run the `isi cloud accounts modify` command.

This sample command changes the name of the cloud storage account **CloudAcct3** to **CloudAcct5** and specifies a proxy server through which communications with the public cloud provider are to be managed.

```
isi cloud accounts modify CloudAcct3 --name=CloudAcct5 --proxy myproxy1
```

Delete a cloud storage account (CLI)

You can delete a cloud storage account. However, proceed with extreme caution, as deleting an account results in loss of cloud data.

Prerequisites

⚠ **CAUTION: Deleting an account results in the permanent loss of access to the data. In effect, you are deleting the data.**

Rather than deleting the cloud storage account, you can stop archiving data to a cloud storage account without deleting it by running the `isi cloud pools modify` command and removing the account from its parent CloudPools. Previously archived files remain in cloud storage, and SmartLink files on the local cluster still point to the cloud data.

Consider the following before deleting a cloud storage account:

- It is recommended that you contact Dell Technologies Customer Support prior to deleting a cloud storage account.
- NDMP, SyncIQ, and Snapshots may be referencing the SmartLink files, which will not function if you delete the associated cloud storage account.
- Cloud objects are not cleaned up when an account is deleted using this command. The cloud objects must be manually removed after deleting the account.
- To delete an account, there must not be any storage pools using that account. Because of dependencies, this usually means that you must first:
  - Delete (or modify) all policies that see storage pools referencing the account
  - Delete (or modify) all storage pools that see the account

Steps

1. Run the `isi cloud accounts delete` command.

   The following command deletes the cloud storage account **OldRecords**.

   ```
   isi cloud accounts delete OldRecords --acknowledge yes
   ```

   In this case, OneFS responds with the following message:

   ```
   ****************************************************
   WARNING: Deleting an account is extremely dangerous.
   Continuing with this operation will result in a permanent loss of data.
   Type 'confirm delete data' to proceed.  Press enter to cancel:
   ```

2. Type the confirmation string **confirm delete data**, then press ENTER.
   The cloud storage account is deleted. Although cloud data remains with your cloud provider, it is not in a format that anyone can access. It cannot be used to reconstruct the files.

List cloud storage accounts (CLI)

You can list all cloud storage accounts created on your cluster, in various formats and sorted order.

About this task

The `isi cloud accounts list` command creates a report of cloud storage accounts and related information. The report includes account name, type of account, account username, URI, status, and bucket, if applicable. You can specify the output in table, json, csv, or list form. You can also request the output to sort by any of the information fields, in ascending or descending order.

Steps

Run the `isi cloud accounts list` command.
The command results appear on the command line in the requested format.

Example

The following command generates a table of cloud accounts sorted by account type in descending order.

```
# isi cloud accounts list --sort type --descending --format table
```

The following command generates json output that lists cloud accounts sorted by account name in the default ascending order.

```
# isi cloud accounts list --sort name --format json
```

View a cloud storage account (CLI)

You can view detailed information about a cloud storage account.

Steps

Run the `isi cloud accounts view` command.
The following command displays account information for the **CloudAcct3** account.

```
isi cloud accounts view CloudAcct3
```

Output from the command displays the properties of the cloud storage account, including account name, type, and more.

Create and manage CloudPools

A CloudPool is a storage pool that enables OneFS to use cloud storage as another tier of storage available to the cluster. Each CloudPool services one cloud storage account.

You can create, view, edit, and monitor CloudPools.

Create a CloudPool (Web UI)

You can create a CloudPool and add a cloud storage account.

Steps

1. Click **File System** > **Storage Pools** > **CloudPools**.
2. Click the **+ Create a CloudPool** button.
3. In the **Create a CloudPool** dialog box, in the **Name** field, enter a name for the CloudPool. The name must be unique on your cluster.
4. From the **Type** drop-down menu, select a type of CloudPool account.
5. Enter a vendor and description for the CloudPool.

6. From the **Account in CloudPool** list, select the cloud storage account that this CloudPool should service. The list is empty until you select a value from the **Type** . The **Account in CloudPool** list then shows only those cloud storage accounts that match that type, for example, Microsoft Azure.

7. Click **Create a CloudPool** .

    The dialog box closes and, in the **CloudPools** list, the new CloudPool is displayed along with its type, state, vendor, and description.

Create a CloudPool (CLI)

You can create a CloudPool and add a cloud storage account.

About this task

A CloudPool is the mechanism that connects a cloud storage account to OneFS. When you create a CloudPool, OneFS enforces two requirements:

- The CloudPool may service only one cloud storage account.
- The cloud storage account must be of the same type as the CloudPool. For example, an Azure CloudPool may only service an Azure cloud storage account.

Steps

Run the `isi cloud pools create` command.

Provide the following parameters in the command:

- A unique name
- The CloudPool type—The values are: `isilon` (for Dell EMC PowerScale), `ecs`, `azure`, `s3`, `c2s-s3`, `google`, or `alibaba-cloud`
- A cloud account name

The following command creates an Azure-based CloudPool named `cp_az`, using the Azure account named `csa_azure1`:

```
isi cloud pools create cp_az azure csa_azure1 --vendor Microsoft
```

Results

The following example shows how to view the result of this operation. Use the ID (name) of the CloudPool in the command.

```
isi cloud pools view cp_az
```

The output displays the CloudPool ID, name, type, and other properties.

View information about a CloudPool (Web UI)

You can view information about a CloudPool, including the cloud storage account, vendor, type, and description.

Steps

1. Click **File System** > **Storage Pools** > **CloudPools**.

    In the **CloudPools** list, each CloudPool is represented by a blue cloud. Associated cloud accounts are listed below each CloudPool, and represented by an orange user icon. The type, state, vendor, and description associated with each CloudPool is displayed.

2. To further view the settings of a CloudPool, click **View/Edit** to the right of the CloudPool.
    The **View Cloud Storage Pool Details** dialog box displays information about the CloudPool.

3. Click **Close** to close the dialog box.

View information about a CloudPool (CLI)

You can view information about a CloudPool, including the cloud storage account, vendor, type, and description.

Prerequisites

The CloudPool must already have been created.

Steps

Run the `isi cloud pools view` command.

The following command provides information on the CloudPool named **cah_s3_cp**.

```
isi cloud pools view cah_s3_cp
```

The output of this command displays the ID, name, type, and other CloudPool properties.

Modify a CloudPool (Web UI)

You can modify a CloudPool, changing the name, the account it contains, the cloud vendor, and the description.

Steps

1. Click **File System** > **Storage Pools** > **CloudPools**.

   In the **CloudPools** list, each CloudPool is represented by a blue cloud icon. The cloud account associated with each CloudPool is listed and represented by an orange user icon. The type, vendor, and description are also displayed.
2. Click **View/Edit** to the right of the CloudPool that you want to modify.
   The **View Cloud Storage Pool Details** dialog box appears.
3. Click **Edit CloudPool**.
   The **Edit CloudPool Details** dialog box appears.
4. Modify the name, vendor, or description fields, as intended.
5. From the **Account in CloudPool** drop-down list, select a different account of the same type.
6. Click the **Save changes** button.
7. In the **View Cloud Storage Pool Details** dialog box, click **Close**.

Results

Any changes that you made to the CloudPool are reflected in the **CloudPools** list.

Modify a CloudPool (CLI)

You can modify a CloudPool, changing the name, the account it contains, the cloud vendor, and the description.

Prerequisites

To determine the available CloudPools on your system, run the `isi cloud pools view` command.

Steps

Run the `isi cloud pools modify` command.

The following command modifies a CloudPool named **c_pool_azure**, removing its cloud storage account

```
isi cloud pool modify c_pool_azure --remove-accounts c_acct2
--description "Secondary archive"
```

Delete a CloudPool (CLI)

You can delete a CloudPool. However, you should proceed with caution. CloudPools provide the mechanism to connect OneFS to your cloud storage accounts. If you delete a CloudPool, the associated cloud storage account is no longer accessible.

Prerequisites

Run the `isi cloud pools list` command to display the names of the CloudPools on your cluster. Run the `isi cloud pools view` command, along with the name to get information about a CloudPool.

Steps

1. Run the `isi cloud pools delete` command.

   The following command deletes the CloudPool named **c_pool_azure**.

   ```
   isi cloud pools delete c_pool_azure
   ```

   OneFS asks you to confirm the deletion, as follows:

   ```
   Are you sure? (yes/[no]):
   ```

2. Type **yes** and press ENTER.
   The CloudPool is deleted.

Monitoring CloudPools (Web UI)

You can monitor the health of CloudPools configured on your cluster.

Steps

1. Click **File System** > **Storage Pools** > **Summary**.
2. In the **Status** list, check the status for CloudPools.

   Status conditions for CloudPools are `Good` or `Needs Attention`. A status of `Needs Attention` appears when a CloudPool cannot connect to the remote cloud provider. This could indicate issues with the Internet connection or with the cloud provider. If you confirm that your Internet connection is good, contact your cloud provider for help.

Manage CloudPools license

You can track the usage of licenses of CloudPools storage providers.
- Run the following command to track the licenses usages.

  ```
  isi_fsa_pools_usage [-h] [--debug] [--fsa-jobid FSA_JOBID] directory
  ```

  - Directory = \ifs would be normal usage to get the total for the cluster
  - Keep handy your FSA details.

- After running the above command, if the result shows you have consistently exceeded 100% over purchased capacity for > 1 year, buy a supplemental license.
- There is no capacity downsize option. Capacity can be upgraded by purchasing a second capacity license for the delta between the current and actual consumed capacity. This license does not need to be activated/applied, add the two license capacities when checking against a supplemental license.
- Dell Technologies recommends you to monitor your license consumption and adhere to EULA.
- Contact your Dell sales representative for an additional capacity license.

File pool policies for CloudPools

This section describes how to define and maintain the file pool policies that archive files to cloud storage.

Topics:

Overview of CloudPools file processing

CloudPools archives file data to the cloud. You can access the archived data whenever needed, for reading or writing. You can also fully recall the data from the cloud, essentially reversing the archive.

File pool policies	You create file pool policies to identify the files to archive to the cloud. When a file pool policy that contains cloud actions runs, CloudPools moves file data to the cloud. The data is stored in specialized cloud data objects, collectively called cloud data. File data can be encrypted and compressed before it is archived to the cloud. For general information about how file pool policies are run and the resulting policy jobs, see the *OneFS Administration Guide*.
SmartLink files	For each file that is archived, CloudPools retains a local proxy file. The proxy is called a SmartLink file. SmartLink files include special metadata and maps to the file data in the cloud.
Inline access	When a user browses OneFS, typically through an SMB connection or NFS export, SmartLink files appear in place of the datafiles. When a user opens a SmartLink file, CloudPools manages the data access. This process is called inline access. A read request retrieves data from the cloud and caches it locally. For subsequent reads, if the requested data is not yet cached locally, it is retrieved then. A write to an uncached area causes a read of a block of data from the cloud. The data is cached and then modified. As the user views the file, CloudPools continues to cache data as required by the application. If the user modifies and saves the file, the changes are also held in the cache. Periodically, CloudPools scans SmartLink files for pending data changes and writes them to the appropriate objects in the cloud. In this way, the archived data is kept up to date.
Recall	You can recall archived files from the cloud. CloudPools fully replaces the SmartLink files with the recalled files.
File pool policies and SmartLink files	Like any file in OneFS, SmartLink files are controlled either by the default file pool policy or by parameters in a custom file pool policy. If you configure additional file pool policies, these policies have priority over the default file pool policy. File pool policies contain instructions that determine how OneFS manages files across a cluster and in the cloud. Because SmartLink files produced by CloudPools are retained on the cluster, OneFS applies file pool policies to these files, as well. When file pool policies run, the system compares each file on the system with each file pool policy. A file matches the file matching criteria of a single file pool policy and is subject to all of the settings in that file pool policy; anything not specified uses default settings. Note that the examples of SSD strategy and snapshot configuration are NOT file matching criteria, they are actions that are taken on matching files.

Archiving files with file pool policies

You can configure a file pool policy to identify the files you want to archive to the cloud and the CloudPools actions to apply to these files.

Specifying a file pool policy, you can archive files using either the OneFS web administration interface or the command-line interface. A file pool policy that archives files to the cloud must specify the following information:

- Files to manage: These can be files of a certain type, files in a specified path, or files that match specified criteria, such as size, creation date, or last modified date.
- CloudPools actions: The cloud storage pool to send file data to, and whether the data should be compressed or encrypted.

Sample policies with CloudPools actions

Each file pool policy identifies a set of files and the CloudPools actions to apply to the file pool. You can identify files to be archived based on multiple criteria, including file type, size, directory path, time of file creation, time of last file access, and time of last file modification.

File-matching criteria in a file pool policy enable you to define a logical group of files referred to as a file pool. After defining a file pool, you specify CloudPools actions to perform on the files, including the cloud storage target, compression, and encryption.

For example, you might define file pool policies that specify files to be archived based on criteria similar to the following:

- Files of <type>, last accessed before <date>
- Files older than <date>, last accessed after <date>, and of <type>
- Files in <directory> that are older than <date>
- Files marked with <custom attribute>, that are older than <date>

You can specify file-matching criteria on a per-policy basis. Each file pool policy allows you to combine multiple criteria using AND statements and OR statements, providing significant flexibility and control for your workflow.

Combining cloud and local storage policy actions

You can specify both cloud and a local storage actions in the same file pool policy. The cloud actions are applied to the data of matching files, while the local actions apply to the SmartLink files that are created in place.

SmartPools settings can determine the target storage pool or tier, file protection level, I/O optimization, and data access optimization. The SmartLink files are processed according to the specified SmartPools parameters. If some settings are not specified in the custom file pool policy, the default file pool policy settings are applied to the SmartLink files.

About file pool policy order

OneFS compares all files to file pool policies in order. The first custom policy that matches a file controls how that file is handled. All other custom file pool policies in the ordered list are ignored. For any of the attributes that the matching custom policy does not specify, the value from the default policy is applied.

This makes the order of file pool policies important. If two or more file pool policies would match the same file, you must ensure that the policy order delivers your preferred file handling instructions.

After a file match with a file pool policy occurs, the system uses the settings in the matching policy to store and protect the file. However, a matching policy might not specify all settings for the match file. In this case, the default policy is used for those settings not specified in the custom policy. For each file stored on the OneFS cluster, the system needs to determine the following:

- Requested protection level
- Data storage target for local data cache
- SSD strategy for metadata and data
- Protection level for local data cache
- Configuration for snapshots
- SmartCache setting
- L3 cache setting
- Data access pattern
- CloudPools actions (if any)

If no custom policy matches a file, the default policy specifies all storage settings for the file. The default policy, in effect, matches all files not matched by any other SmartPools policy. For this reason, the default policy is the last in the file pool policy list, and always the last policy the system applies.

Files that have been archived to the cloud are always governed by the original policy.

File pool policy cloud archive parameters

CloudPools provides a specific set of file pool parameters that support archiving files to the cloud. The following table describes these parameters.

The parameters are available in the Web UI and in the CLI command. The first two columns in the following table show the corresponding names in the two interfaces.

Table 3. File pool policy parameter descriptions

Web UI parameter name	CLI parameter name	Description	Usage notes
CloudPool Storage Target	`cloud-pool`	The PowerScale administrative container for a cloud storage account.	Each CloudPool can refer to only one cloud storage account. You must create a cloud storage account before creating and configuring a CloudPool. A CloudPools and its cloud storage account must be the same type: Dell Technologies PowerScale, Dell Technologies ECS Appliance, Amazon S3, Amazon C2S, Microsoft Azure, Google Cloud, or Alibaba Cloud.
Encrypt data before transfer	`cloud-encryption-enabled`	Specifies whether CloudPools encrypts data before archiving it. The default value is `disabled`	Cloud data is decrypted when it is accessed or recalled.
Compress data before transfer	`cloud-compression-enabled`	Specifies whether CloudPools compresses data before archiving it. The default value is `disabled`.	Cloud data is decompressed when it is accessed or recalled.
Cloud Data Retention Period	`cloud-data-retention`	The length of time that cloud files are retained after the files are fully recalled. The default value is `1 week`.	CloudPools cleans up local resources that were allocated for the SmartLink files. CloudPools also removes the associated cloud objects. This work is performed frequently, affecting only objects whose retention period expired. ⓘ **NOTE:** The system removes (garbage-collects) cloud objects when their SmartLink files and all local references including snapshots to them are removed. If a SmartLink file is backed up and the original SmartLink file is later deleted, associated cloud objects are deleted only after the retention time of the backed-up SmartLink file expires.
Incremental Backup Retention Period for NDMP Incremental Backup and SyncIQ	`cloud-incremental-backup-retention`	Specifies the length of time that OneFS retains cloud data that is referenced by a SmartLink file that was last replicated by SyncIQ or an incremental NDMP backup. The default value is `5 years`.	If a SmartLink file is backed up and the original SmartLink file is later deleted, associated cloud objects are deleted only after the retention time of the backed-up SmartLink file expires.

Table 3. File pool policy parameter descriptions (continued)

Web UI parameter name	CLI parameter name	Description	Usage notes
Full Backup Retention Period for NDMP Only	`cloud-full-backup-retention`	Specifies the length of time that OneFS retains cloud data that is referenced by a SmartLink file that was backed up by a full NDMP backup. The default value is `5 years`.	If a SmartLink file is backed up and the original SmartLink file is later deleted, associated cloud objects are deleted only after the applicable retention period expires.
Writeback Frequency	`cloud-writeback-frequency`	Specified the interval at which the system writes the data stored in the cache of SmartLink files to the cloud. The default value is `9 hours`.	Specifies how often SmartLink files modified on the cluster are written to their associated cloud data objects.
Accessibility	`cloud-accessibility`	Specifies how data is cached in SmartLink files when a user or application accesses a SmartLink file on the cluster. Values are cached and no-cache. The default value is `cached`.	Determines whether cloud data is cached when a file is accessed on the local cluster. **cached** When `cached` is selected, accessed cloud data is cached to the SmartLink file on read or write access. **no-cache** When `no-cache` is selected, the system does not cache data in the SmartLink files on read access, but passes it through to the local accessing application. If you write to data accessed when this setting applies, the system caches your changes. Choose `no-cache` if you want to limit the use of cluster resources.
Cache Read Ahead	`cloud-readahead`	Specifies the cache read ahead strategy for cloud files (one of partial or full) The default value is `partial`	Specifies whether cloud data is fully or partially retrieved when you access a SmartLink file on the cluster. If `partial` is specified, the system only retrieves the file blocks needed

Table 3. File pool policy parameter descriptions (continued)

Web UI parameter name	CLI parameter name	Description	Usage notes
			when a SmartLink file is accessed. If `full` is specified, all cloud data is fully cached when the SmartLink file is accessed.
Cache Expiration	`cloud-cache-expiration`	Specifies the number of days until the system purges expired cache information in SmartLink files. The default value is `1 day`.	Specifies how long the system retains cloud data that has been retrieved in the cache of associated SmartLink files. The system purges the SmartLink file cache of data that has not been accessed for the number of days specified.

File matching options for cloud archival policies

Each file pool policy must provide match criteria to identify the files to archive and the cloud target where the files should be stored.

The following table describes the match criteria to use when creating file pool policies.

Table 4. Match criteria

Match criteria		Description
Web admin interface	**Command line interface**	
Filename	--name	Includes or excludes files based on the file name. You can specify whether to include or exclude full or partial names that contain specific text. Wildcard characters are supported.
Path	--path	Includes or excludes files based on the file path. You can specify whether to include or exclude full or partial paths that contain specified text. You can also include the wildcard characters *, **?**, and **[ ]**.
File Type	--file-type	Includes or excludes files based on one of the following file-system object types: ● Regular file ● Directory ● Other
File Attribute	--custom-attribute	Includes or excludes files based on a custom user-defined attribute.
Modified	--changed-time	Includes or excludes files based on when the file was last modified. You can specify a relative date and time, such as "older than 2 weeks," or a specific date and time, such as "before January 1, 2012." Time settings are based on a 24-hour clock. For more information see *Use time stamps for cloud data archival and recall*.
Accessed	--accessed-time	Includes or excludes files based on when the file was last accessed. You can specify a relative date and time, such as "older than 2 weeks," or a specific date and time, such as "before January 1, 2012." Time settings are based on a 24-hour clock.

Table 4. Match criteria (continued)

Match criteria		Description
Web admin interface	**Command line interface**	
		ⓘ **NOTE:** Because it affects performance, access time tracking as a file pool policy criterion is disabled by default.
Metadata Changed	--metadata-changed-time	Includes or excludes files based on when the file metadata was last modified. This option is available only if the global access-time-tracking option of the cluster is enabled.
		You can specify a relative date and time, such as "older than 2 weeks," or a specific date and time, such as "before January 1, 2012." Time settings are based on a 24-hour clock.
Created	--birth-time	Includes or excludes files based on when the file was created.
		You can specify a relative date and time, such as "older than 2 weeks," or a specific date and time, such as "before January 1, 2012." Time settings are based on a 24-hour clock.
Size	--size	Includes or excludes files based on their size.
		ⓘ **NOTE:** File sizes are represented in multiples of 1024, not 1000.

Managing cloud policies

CloudPools takes advantage of the SmartPools infrastructure, and applies file pool policies to determine which files are to be archived to the cloud.

Therefore, you must activate both a SmartPools and a CloudPools license to store data in the cloud.

By defining file pool policies, you can have OneFS automatically archive files to the cloud when they match certain characteristics, such as age, size, type, or location.

File pool policies are both for SmartPools and CloudPools purposes. A file pool policy can specify a local storage target, a cloud storage target, or both. If you create a policy that specifies both local and cloud targets, the policy moves file data to the cloud, and applies the local settings to the SmartLink files retained on the local cluster.

If the purpose of a file pool policy is to move files to a local node pool or tier, do not configure a cloud target. Conversely, if the purpose of a policy is to archive files to the cloud, configuring a local target. In this case, the system uses the settings of the default file pool policy to store the local SmartLink files which may or may not meet the needs for the local parameters.

Create a file pool policy for cloud storage (Web UI)

You can create file pool policies that specify CloudPools actions to be applied to selected files.

Steps

1. Click **File System** > **Storage Pools** > **File Pool Policies**.
2. Click the **+ Create a File Pool Policy** button.

   The **Create a File Pool Policy** dialog box displays.
3. Enter a policy name and, optionally, a description.
4. In the **Select Files to Manage** area, use the pull-down menus to specify the file selection criteria for cloud storage. The criteria you specify are used by OneFS to determine the files to archive. The criteria you specify for file selection can include the following attributes, combined with Boolean operators:
   - Filename
   - Path
   - File Type
   - File Attribute
   - Modified

- Accessed
- Metadata Changed
- Created
- Size

5. In the **Apply CloudPools Actions to Selected Files** area, select **Move to cloud storage**.

6. In the **CloudPool Storage Target** drop-down menu, select an existing CloudPool, and specify whether to encrypt and compress data before it is archived to the cloud.

7. Click **Show Advanced CloudPool Settings** to specify additional cloud storage options. See the table in File pool policy cloud archive parameters for explanations of each parameter.

Section	Parameter
Data Retention Settings	**Cloud Data Retention Period** **Incremental Backup Retention Period for NDMP Incremental Backup and SyncIQ**: **Full Backup Retention Period for NDMP Only**
Accessibility and Cache Settings	**Writeback Frequency** **Accessibility** **Cache Read Ahead** **Cache Expiration**

8. Click **Create Policy**.

The file pool policy appears under File Pool Policies in the **File Pool Policies** window.

Results

The next time the SmartPools system job is run, the file pool policy performs the specified actions.

Create a file pool policy for cloud storage (CLI)

You can create file pool policies that specify CloudPools actions to be applied to selected files.

About this task

To create CloudPool policies, use the `isi filepool policies create` command. Use the `--cloud-pool` option to specify the CloudPool storage pool for which you are creating the policy.

Steps

Run the `isi filepool policies create` command.

```
isi filepool policies create <policy-name> --cloud-pool <cloudpool-name> ....
```

In the above syntax, the ellipses (…) indicates that policy options need to be specified. There are many options for defining a policy:

- You may define a filter that is applied to a set of files.
  - Often policies are defined based on the amount of time that has elapsed since the file was last modified. This requires you to state the elapsed time in years, months, weeks, days, hours, minutes, or seconds in a *<duration>* parameter. The format for a duration parameter is `integer{Y|M|W|D|H|m|s}`. For example, 2D indicates 2 days; 2W indicates 2 weeks.
  - Other options use a definitive time.

  For information about how to create a filter and filter options, see the `isi filepool policies create` reference page.

- You may specify optional parameters that control aspects of data retention, data accessibility, and cache settings. For information about the parameters, see the table in File pool policy cloud archive parameters.

The following example creates a file pool policy with the name `archive` and the CloudPool storage target `S3_pool`. The command specifies a file-matching pattern to archive all files in a specified directory path that have not been accessed after November 30, 2019.

```
isi filepool policies create archive --cloud-pool S3_pool
--begin-filter --name="*.*" --and --path="/ifs/home/users"
--and --accessed-time=2019-11-30 --operator=lt --end-filter
```

Modify cloud attributes in a file pool policy (Web UI)

You can modify a file pool policy. Each file pool policy for cloud archival specifies a file-matching pattern and the actions to perform on the matched files (file pool).

Steps

1. Click **File System** > **Storage Pools** > **File Pool Policies**.

   The **File Pool Policies** page appears.
2. In the File Pool Policies list, next to the file pool policy you intend to modify, click **View/Edit**.

   The **View File Pool Policy Details** dialog box appears.
3. Click **Edit Policy**.
   The **Edit File Pool Policy Details** dialog box appears.
4. Make your changes in the appropriate areas and click **Save Changes**.

Results

Changes to the file pool policy are applied the next time the SmartPools system job runs.

Modify cloud attributes in a file pool policy (CLI)

You can modify a file pool policy. Each file pool policy for cloud archival specifies a file-matching pattern and the actions to perform on the matched files (file pool).

About this task

You can run the `isi filepool policies list` command to list available file pool policies.

Steps

Run the `isi filepool policies modify` command.

The following example modifies the file-matching pattern in a file pool policy named **my_policy**.

```
isi filepool policies modify my_policy --begin-filter
--name="*.jpg" --and --accessed-time=2013-08-01 --operator=lt
--end-filter
```

List file pool policies (CLI)

You can list all file pool policies stored in OneFS.

Steps

Run the `isi filepool policies list` command.

View details of a file pool policy (CLI)

You can display detailed information about a file pool policy.

About this task

To list all available file pool policies, you can run the `isi filepool policies list` command.

Steps

Run the `isi filepool policies view` command.

The following command displays information about the policy `my_policy`, including status, associated CloudPool, whether encryption and compression are enabled, and more.

```
isi filepool policies view my_policy
```

Apply a file pool policy to a specified file or path (CLI)

You can apply a file pool policy to specified files or directories manually, rather than waiting for the SmartPools job to run.

About this task

For `isi filepool apply` to execute, the file or directory specified must match one of the defined file pool policies.

Steps

Run the `isi filepool apply` command.

The following command applies the appropriate file pool policy to all files and subdirectories in a given path.

```
isi filepool apply --path=/ifs/data/images --recurse
```

Archive files directly to the cloud (CLI)

You can archive specific files directly to the cloud. To enable this, CloudPools must match these files to an existing file pool policy.

Prerequisites

A custom file pool policy that matches the specified file or files and points to cloud storage must already exist on your system.

Steps

Run the `isi cloud archive` command.

The following command specifies a directory and all of its subdirectories and files to be archived if they match the specified file pool policy:

```
isi cloud archive /ifs/data/shared/images/*.* --recursive yes --policy  mypolicy
```

Managing cloud jobs

You can monitor and manage two types of cloud jobs: system jobs that are always running in the background, and manual jobs that are created with the `isi cloud jobs archive` and `isi cloud jobs recall` commands. OneFS enables you to monitor the status of both job types, and to monitor and manage your manual archive and recall jobs, as needed.

View a list of cloud jobs (CLI)

You can list all CloudPools jobs. Both CloudPools system jobs and manual jobs are listed.

About this task

CloudPools system jobs are always running to service caching and clean-up (garbage collection) processes. CloudPools manual jobs include archive jobs specified in file pool policies, and recall jobs started from the OneFS command-line interface. Each job is listed by ID, description, state, and type.

Steps

Run the `isi cloud jobs list` command.

Output from the command lists CloudPools job ID, description, status, and type.

View a cloud job (CLI)

You can view information about a CloudPools job.

Prerequisites

You need to know the ID of the job you want to view. You can run the `isi cloud jobs list` command to see the IDs for all cloud jobs.

Steps

Run the `isi cloud jobs view` command.

The following command views information about a job with the ID of 63.

```
isi cloud jobs view 63
```

Pause a cloud job (CLI)

You can pause a running CloudPools job, or pause all CloudPools jobs. This operation is typically done only for troubleshooting purposes.

Prerequisites

To pause a specific job, you need to know the ID of the job. Run the `isi cloud jobs list` command to see a list of all cloud job IDs.

Steps

Run the `isi cloud jobs pause` command.

The following command pauses a job with the ID of 63.

```
isi cloud jobs pause 63
```

This command pauses all running archive jobs and also pauses any subsequently started archive jobs:

```
isi cloud jobs pause archive
```

Resume a paused cloud job (CLI)

You can resume a cloud job that was paused.

Prerequisites

To resume a job, you must know the ID of the job. Run the `isi cloud jobs list` command to see a list of all cloud job IDs.

Steps

Run the `isi cloud jobs resume` command.
The following command resumes a job with the ID of 63.

```
isi cloud jobs resume 63
```

If you paused all cloud jobs using `isi cloud jobs pause archive`, you must resume all cloud jobs using the following command:

```
isi cloud jobs resume archive
```

Cancel a cloud job (CLI)

You can cancel a running CloudPools job.

Prerequisites

To cancel a job, you need to know the ID of the job. Run the `isi cloud jobs list` command to see a list of all cloud job IDs.

Steps

Run the `isi cloud jobs cancel` command.
The following command cancels a job with an ID of 63.

```
isi cloud jobs cancel 63
```

Retrieving file data from the cloud

You can retrieve file data from the cloud either by inline access through a supported protocol (SMB, NFS, Swift, or HDFS), or by fully recalling files.

Inline access of cloud data

Users can retrieve file data in the cloud by accessing a SmartLink file on the local cluster through a supported protocol. This method is referred to as inline access.

When the user reads or writes data by accessing a SmartLink file, for example, through an SMB share, CloudPools retrieves and locally caches file data from the cloud. The amount of data that is cached is determined by the CloudPools **Cache Read Ahead** setting.

If the user makes changes to the file, CloudPools maintains those changes in cache and periodically updates file data to the cloud so that the latest version is always archived.

Recalling files from the cloud

You can fully recall files from cloud storage.

CloudPools restores the full file to the cluster and overwrites its associated SmartLink file. CloudPools garbage collection removes related objects from the cloud according to parameter settings in the policy.

> (i) **NOTE:** The full file is restored to its original directory. If the file pool policy that originally archived the file to the cloud is still in effect, the next time the SmartPools job runs, the recalled file is archived to the cloud again. If you do not want the recalled file to be re-archived, you can move the file to a different directory that would not be affected by the file pool policy, or you can modify or delete the policy.

Use the CLI command `isi cloud recall` to recall files from cloud storage. You can recall files individually by name or you can specify a directory path. A recursive option specifies whether to recall files in nested subdirectories.

See the isi cloud recall reference page for syntax and usage information.

CloudPools with other OneFS functions

Topics:

Compression and encryption of cloud data

You can specify compression and encryption of data that is moved to the cloud.

With CloudPools, you can enable compression and encryption on a per-policy basis. Both encryption and compression are disabled by default.

Files encrypted or compressed when stored in the cloud are automatically decrypted and decompressed when data is cached (inline access) or the file is recalled from the cloud to local storage.

CloudPools uses a master encryption key to encrypt the data encryption keys. Encryption applies to both the SmartLink file and the file data archived to the cloud. Both the SmartLink file and the archived data include encrypted copies of the data encryption keys. After a file is encrypted, it can only be decrypted by recalling it.

CloudPools keeps track of the encryption status of SmartLink files in snapshots and referenced data in the cloud. If SmartLink files in snapshots are unencrypted and refer to unencrypted cloud objects, the SmartLink files in the snapshots remain unencrypted even if you create a new CloudPools policy that encrypts the latest version of the file.

OneFS stores the master encryption key in the local key management system. You can generate a new version of the key if you believe the key has been compromised. If regenerated, the new master key secures new data written to the cloud. Previously written data is secured by the old data encryption keys, resident in the local SmartLink files.

Self-encrypting drives

CloudPools works with nodes that are equipped with self-encrypting drives (SEDs). Any SmartLink files left on SEDs are handled like any other file. However, note the following about archived data and whether it remains encrypted:

- The process of archiving a SED file decrypts the data on the SED.

  Any read of SED data decrypts the data. Because CloudPools reads the data to archive it, the data is decrypted.

- The data archived to the cloud is not encrypted unless the CloudPools policy includes `encrypt=True`.

CloudPools protocol support

CloudPools supports inline access of cloud data through protocols such as SMB, NFS, Swift, and HDFS. Consequently, users who access files from other systems can also access files stored in the cloud.

When a user connects to the cluster with a given protocol (SMB, NFS, Swift, and HDFS) and browses files, Smartlink files appear to be original (normal) files.

In those scenarios, when the user opens a SmartLink file, CloudPools retrieves and caches the original file data from the cloud. Depending on the **Cache Read Ahead** setting, either a portion of the file data, or the entire file, is cached.

If the user modifies the file, CloudPools caches the changes and periodically writes the changes back to the cloud. In this way, the cloud data is kept fully up to date.

SyncIQ interoperability

SyncIQ enables you to synchronize data from your PowerScale primary (source) cluster to a secondary (target) cluster. If the primary cluster becomes unavailable, you can fail over to the secondary cluster. Users can continue to access data, including data stored in the cloud.

During SyncIQ replication, all files, including SmartLink files, are copied from the source cluster to the target cluster. Users given access to the target cluster through supported protocols can retrieve cloud data or recalls the full file exactly as it would from the original source cluster. In these cases, CloudPools retrieves and caches data (inline access) or fully recall files from the cloud.

Unless you grant cloud write access (which should not be done, except in the case of long term failover) to the secondary cluster, CloudPools stores any changes to SmartLink files in the local cache. This cache is limited only by available space on the cluster.

SyncIQ policies

CloudPools supports SyncIQ replication of SmartLink files to one or more target clusters. SyncIQ can also be used to restore backed up SmartLink files to their original (source) cluster.

The two types of SyncIQ policies are synchronization policies and copy policies. These policies can be run manually, or configured to run automatically, based on policy settings.

CloudPools supports both types of SyncIQ policy. When SyncIQ replicates SmartLink files to a target cluster, secondary information associated with a SmartLink file, such as local cache state and unsynchronized cache data, is also replicated.

If your source (primary) cluster goes down or is unavailable for any reason, and you fail over to the secondary cluster, users can continue to access SmartLink files and, therefore, cloud data, as they would normally.

If the failover is temporary and you plan to restore your source cluster to full operation, you do not need to enable cloud write access on the secondary cluster. Any changes that users make to SmartLink files are stored in the local cache, which is limited only by the amount of free space on your cluster. When you fail back to your source cluster, and restore updated SmartLink files, only then will CloudPools write the cached modifications back to the cloud.

If the failover is long-term or permanent, see Configuring access to cloud data from a secondary cluster for information about providing the secondary cluster with write access to the cloud data.

CloudPools cloud data retention time

Retention parameters define an amount of time for cloud data to remain in cloud storage after the related SmartLink file is deleted.

When CloudPools archives a file from your cluster to cloud storage, it creates a SmartLink file on the PowerScale cluster in place of the archived file. If the archived data remains in the cloud, the SmartLink file remains in place to represent and access the cloud data.

If a user deletes or recalls a SmartLink file, and all other references to that cloud data (such as snapshots taken while the file was archived) are removed, the cloud data that is associated with that SmartLink file becomes eligible for garbage collection. CloudPools deletes the data from cloud storage, but not immediately. A calculated deletion date is associated with each SmartLink file. The deletion date determines the time that must pass after the SmartLink file is deleted or recalled before the cloud data is garbage-collected. Retention parameters determine the deletion date.

Retention parameters and the deletion date are important when you are using SyncIQ or NDMP backups.

A SmartLink file might be backed up to tape through NDMP or replicated to another PowerScale cluster through SyncIQ. More than one SmartLink file can point to the same cloud data. Users might use a restored or replicated version of a deleted or recalled SmartLink file to access the cloud data. Depending on the deletion date, the supporting cloud data might be deleted. In these situations, the data is unreachable from the backup or replicated version of the SmartLink file.

Retention parameters are configurable. Short retention times are likely to cause the cloud garbage collection to occur before a user attempts to access the data using a restored or replicated SmartLink file. Longer retention times give your organization more time to ensure that a restore would work as expected. In general, if you are backing up or replicating SmartLink files, do not set small values in the retention parameters.

The following sections describe the retention parameters and how CloudPools uses them to calculate the deletion date.

Retention parameters

The following archive policy retention periods affect the deletion date:

- **Cloud Data Retention Period** specifies the retention time of cloud data beyond the time when an associated local SmartLink file is deleted or recalled, leaving no file references from the local cluster. The default setting is one week.
- **Backup Retention Period for NDMP Incremental Backup and SyncIQ** specifies the retention time of cloud data whose SmartLink file is backed up by an incremental NDMP backup, or replicated by a SyncIQ operation. If a local SmartLink file is deleted, you can restore the SmartLink file copy, and access the cloud data. The default setting is five years.
- **Full Backup Retention Period for NDMP Only** is the retention time of cloud data whose SmartLink file is backed up by a full NDMP backup. If a local SmartLink file is deleted, the SmartLink file copy is restored from the backup, and users can access the cloud data. The default setting is five years. See *Deletion date calculations* for more information.

When a SmartLink file is replicated to a secondary cluster and is then deleted from the primary cluster, CloudPools considers all those settings to determine when to delete the associated cloud objects.

- CloudPools first determines which workflows affect the deleted file. CloudPools considers only the settings that apply to the relevant workflows. For example, if an NDMP file was never fully backed up, CloudPools does not consider the **Full Backup Retention Period for NDMP Only** parameter to determine when to delete associated cloud objects for that file.
- CloudPools uses the longer of all the appropriate durations to determine when to delete cloud objects.

If you delete a SmartLink file on the secondary cluster (because the primary cluster is temporarily unavailable), the deleted state remains in cache. When you fail back to the primary cluster, CloudPools deletes the SmartLink file. CloudPools uses the retention settings to determine when to delete the associated cloud data.

You can view retention values using the following command:

```
isi filepool policies view <policy-name>
```

For example:

```
# isi filepool policies view my-policy
   .
   .
   .
              Cloud Data Retention: 1W
Cloud Incremental Backup Retention: 5Y
       Cloud Full Backup Retention: 5Y
```

Deletion date calculations

Each retention parameter is a delta time in seconds. When certain events happen, one of the retention values is added to the current time to create an absolute future time. If the new time is farther in the future, it becomes the new deletion date.

The new deletion date is calculated as follows:

- When SyncIQ or NDMP copies a SmartLink file, either the **Backup Retention Period for NDMP Incremental Backup and SyncIQ** or **Full Backup Retention Period for NDMP Only** value is used to calculate the new deletion date. This calculation only applies to when a file is part of a SyncIQ job or NDMP backup. If the file has not changed, and thus is not backed up, the deletion date will not be updated.
- When the file is recalled, the **Cloud Data Retention Period** value is used to calculate the new deletion date. This occurs regardless of whether the file was previously backed up using SyncIQ or NDMP. In other words, the later of the current object deletion or garbage collection time for the file or the current time plus the Cloud Data Retention Period are used for the garbage collection time. For example, if a SmartLinked file was last backed up over 5 years ago (default) and is then recalled, the new garbage collection time will be 1 week (default Cloud Data Retention Time) beyond the time of the recall.
- When a file is deleted (and all references to that file are removed - i.e. snapshots that contain changes to that file), then the greater of the garbage collection time or the current time + Cloud Data Retention Time is used to determine the new deletion date.

Garbage collection occurs after the deletion date. The cluster that has cloud access to the cloud account that archived the file performs the garbage collection. See the isi cloud access add command.

Attempts to read a SmartLink file whose cloud data was garbage collected fail. For example, a SmartLink file on a SyncIQ target cluster does not work if the cloud data was deleted. Similarly, a SmartLink file that is restored from an NDMP backup does not work. This situation typically occurs only if the policy on the SmartLink file uses small values for the retention parameters.

Replicated SmartLink files

If you modify or delete a SmartLink file that has been replicated in a SyncIQ operation, CloudPools manages the associated cloud objects.

Here are the scenarios and how they are handled.

If you modify a SmartLink file on the primary cluster, the changes are cached and, depending on the **Writeback Frequency** setting, are periodically written back to the cloud. In this way, cloud data is always kept up to date.

If you modify a SmartLink file on a secondary cluster (because the primary cluster is temporarily unavailable), changes remain in cache. When you fail back to the primary cluster, only then are changes written back to the cloud according to the **Writeback Frequency** setting.

If you delete a SmartLink file that was replicated in a SyncIQ operation, CloudPools appropriately manages the deletion of the associated cloud data. Two retention periods can affect the cloud objects associated with a SmartLink file that has been replicated: the **Cloud Data Retention Period** and the **Incremental Backup Retention Period for NDMP Incremental Backup and SyncIQ**. See CloudPools cloud data retention time

If you delete a SmartLink file on the secondary cluster (because the primary cluster is temporarily unavailable), the deleted state will remain in cache. When you fail back to the primary cluster, CloudPools deletes the SmartLink file, and uses the retention settings to determine when to delete the associated cloud data.

SyncIQ deep copy

You can create a SyncIQ policy that replicates full files rather than SmartLink files when copying data from the primary (source) cluster to a secondary (target) cluster.

When you create a SyncIQ policy, you can modify the **Deep Copy for CloudPools** setting. The default setting is `Deny`, which means that, during a SyncIQ operation, SmartLink files are replicated to the target cluster.

Alternatively, you can select either the `Allow` or `Force` option for deep copy.

- `Allow` replicates SmartLink files to the target cluster, unless there is a SmartLink version mismatch. In that case, SyncIQ retrieves the full file data from the cloud and then replicates the data.
- `Force` retrieves and copies full file data from the cloud for all SmartLink files in the SyncIQ policy. This option replicates the full files to the target cluster.

ⓘ **NOTE:** A SyncIQ operation that forces deep copy can take more time, consume more system resources, and increase costs to download the data. It is recommended not to use deep copy unless you have a specific reason to do so. For example, use deep copy to back up data from the primary cluster to a secondary cluster that is running a pre-8.2 version of OneFS. If you are unsure whether to use deep copy, contact Technical Support for guidance.

Configuring access to cloud data from a secondary cluster

You can make cloud data available on a secondary cluster if your primary cluster becomes unavailable.

To configure such access, you must have replicated the primary cluster's data onto a secondary cluster using SyncIQ. Alternatively, you can restore an NDMP backup of the data to a secondary cluster.

The secondary cluster must have active SyncIQ, SmartPools, and CloudPools licenses.

With SyncIQ, when failover to a secondary cluster is required, two use cases are supported: short-term failover versus long-term failover.

In the short-term failover use case, the intention is to restore and failback to the primary cluster as quickly as possible. The secondary cluster is a temporary solution, enabling users to open SmartLink files from supported protocols and access cloud data as usual. Instead of writing any changes back to the cloud, however, CloudPools caches these changes locally in the SmartLink files on the secondary cluster. After the primary cluster is restored to service, CloudPools writes back any changes on the secondary cluster to the primary cluster. Cached data in SmartLink files will then be written back to cloud storage.

In a long-term failover situation, in which the primary cluster will be out of service for an extended period or decommissioned entirely, other considerations become important. In this scenario, because only one cluster can have write access to cloud storage, you need to transfer write access to the failover cluster. From a CloudPools perspective in this scenario, the failover cluster effectively becomes the primary cluster. See Configure write access to cloud pool data in long-term failover situations

With the NDMP approach, however, the short-term failover scenario is less practical. The secondary cluster should be given cloud write access to enable any cached modifications to SmartLink files to be written back to cloud storage. The alternative would be to somehow write modified SmartLink files back to the primary cluster after it is restored to service, but this might be more time-consuming.

> △ **CAUTION: Never allow write access to cloud data from more than one cluster at a time because it can result in data corruption.**

Configure write access to cloud pool data in long-term failover situations

In the case of a long-term failover situation, you can provide write access to data in the cloud to the secondary (failover) cluster.

Prerequisites

Prerequisites are:

- Data from the primary (source) cluster must have been replicated to or restored on the secondary (target) cluster by a SyncIQ or NDMP process.
- The secondary (target) cluster must have both a SmartPools and CloudPools license.
- You must know or be able to obtain the GUID associated with the cloud data. Best practice would be to obtain and save this information before you actually need to use it, when the cloud data is configured. Otherwise, see Step 1 in the procedure below.

About this task

By default, write access to cloud data can occur only from the OneFS cluster that originally archived the data to the cloud. In a short-term failover scenario, the secondary cluster reads the data from the cloud and, if the user makes any modifications, the secondary cluster collects modifications in cache. When the original cluster is available again and failover is complete, the original cluster takes over and writes the cached modifications to the cloud.

In a long-term failover situation, dependence on cached modifications is risky. In that case, you might choose to provide the secondary cluster with write access to the cloud data.

> △ **CAUTION: This capability is offered to work around cases where the primary cluster will be unavailable for an extended period. Never allow write access to cloud data from more than one cluster at a time because it can result in data corruption. Before allowing another cluster to have cloud write access, make sure that cloud write access is removed from the primary cluster, or that the primary cluster is offline and remains offline. If the primary cluster becomes available again, continue to ensure that only one cluster has write access to the cloud data. Do this by removing write access from the secondary cluster before allowing the primary cluster to regain write access.**

The following procedure describes how to remove write access to cloud data from one cluster and provide that access to another cluster. Follow the steps in the order shown.

> △ **CAUTION: If the primary cluster is not operational and cannot be made operational, you are forced to skip step 3. In that case, you must be sure to remove the write access from the secondary cluster before attempting to restart the primary cluster. Data corruption could result if two clusters have write access to the cloud data.**

Steps

1. Obtain the GUID that is associated with the cloud data.

   The GUID of the cluster that originally archived the cloud data is permanently associated with the cloud data. In most scenarios, this is the GUID of the primary cluster. If you have reconfigured clusters, it is possible that the primary cluster is not the one that originally archived to the cloud.

   On the cluster that originally archived to the cloud, run this command:

   ```
   isi cloud access list
   ```

The GUID of the cluster on which you are running the command is identified with the phrase `(current)`.

Other GUIDs, if any, identify other clusters on which data was archived (using another account) and from which data has been replicated with SyncIQ or restored with NDMP.

2. Failover to the secondary cluster.
3. On the primary cluster, remove write access to the cloud data.

```
# isi cloud access remove <GUID>
```

where *<GUID>* is the GUID of the cluster that originally archived the cloud data. For example:

```
# isi cloud access remove ab9dd991261e11e382240800200c9a66
```

4. On the secondary cluster, give write access to the cloud data.

```
# isi cloud access add <GUID>
```

where *<GUID>* is the GUID of the cluster that originally archived the data. For example:

```
# isi cloud access add ab9dd991261e11e382240800200c9a66
```

Results

The secondary cluster can write modifications to the cloud, rather than storing the modifications in cache.

Return write access to the primary cluster

When the primary cluster returns to service, you can return write access to data in the cloud to the primary cluster.

About this task

This procedure describes how to fail back to the primary cluster after a long-term failover. All of the steps in a fail back scenario are listed here for context, but only the steps specific to CloudPools are described in detail. For more information about data failover and failback with SyncIQ, see the *OneFS CLI Administration Guide*.

Steps

1. Perform SIQ resync-prep <policy> on the original primary cluster.
2. Wait until the <policy>_mirror policy exists on the secondary cluster.
3. Perform SIQ sync of <policy>_mirror on the secondary cluster.
4. Perform SIQ allow-write of <policy>_mirror on the original primary cluster.
5. On the secondary cluster, remove write access to the cloud data.

```
# isi cloud access remove <GUID>
```

where *<GUID>* is the GUID of the cluster that originally archived the cloud data. For example:

```
# isi cloud access remove ab9dd991261e11e382240800200c9a66
```

6. On the primary cluster, give write access to the cloud data.

```
# isi cloud access add <GUID>
```

where *<GUID>* is the GUID of the cluster that originally archived the data. For example:

```
# isi cloud access add ab9dd991261e11e382240800200c9a66
```

7. Wait until source directory on primary cluster becomes writable.
8. Perform SIQ recovery resync-prep of <policy>_mirror on the secondary cluster.

Results

The primary cluster can write modifications to the cloud, whereas the secondary cluster can not.

NDMP backup and restore of SmartLink files

You can perform NDMP backup and restore operations on data that has been archived to the cloud.

Backup and restore capabilities with CloudPools data include:

- Archiving SmartLink files when backing up from a cluster
- Restoring data, including SmartLink files, to the same cluster
- Restoring data, including SmartLink files, to another cluster
- Backing up version information with each SmartLink file, and restoring the Smartlink file after verifying the version compatibility on the target cluster

(i) **NOTE:** SmartLink files that are backed up with OneFS 8.2.0 and later releases cannot be restored to releases earlier than 8.2.0.

You specify how files are backed up and restored by setting the NDMP environment variables `BACKUP_OPTIONS` and `RESTORE_OPTIONS`. See Administering NDMP in the *PowerScale OneFS CLI Administration Guide* for details about configuring the backup settings and managing NDMP environment variables.

(i) **NOTE:** DeepCopy and ComboCopy backups recall file data from the cloud. The data is not stored on disks. Recall of file data may incur charges from cloud vendors.

With NDMP backup, by default, CloudPools supports the backup of SmartLink files that contain cloud metadata such as location of the object. Other details such as version information, account information, local cache state, and unsynchronized cache data that are associated with the SmartLink file are also backed up.

To prevent data loss when recovering SmartLink files with incompatible versions, use the NDMP combo copy backup option. Use this option to back up SmartLink files with full data. Full data includes metadata and user data. Use the NDMP combo copy option by setting the `BACKUP_OPTIONS` environment variable.

When the combo copy option is used for backup, you can use the combo copy, shallow copy, or deep copy restore options to recover SmartLink files. You can specify these options by setting appropriate values to the `RESTORE_OPTIONS` environment variable:

- The combo copy restore option restores SmartLink files from the backup stream only if their version is compatible with the OneFS version on the target cluster. If the SmartLink file version is incompatible with the OneFS version on the target cluster, a regular file is restored.
- If the version check operation on the target cluster is successful, the shallow copy restore operation restores the backed-up SmartLink file as a SmartLink file on the target cluster.
- If the version check operation on the target cluster fails, the deep copy restore operation forces the recovery of the SmartLink files as regular files on the target cluster .
- If you do not specify any restore operation, NDMP restores SmartLink files using the combo copy restore operation by default.
- When you specify multiple restore options, the combo copy restore operation has the highest priority. The shallow copy restore operation has the next highest priority. The deep copy restore operation has the lowest priority.

In CloudPools settings, you can set three retention periods that affect backed up SmartLink files and their associated cloud data:

- Full Backup Retention Period for NDMP takes effect when the SmartLink file is backed up as part of a full backup. The default is five years.
- Incremental Backup Retention Period for Incremental NDMP Backup and SyncIQ takes effect when a SmartLink file is backed up as part of an incremental backup. The default is five years.
- Cloud Data Retention Period defines the duration that data in the cloud is kept when its related SmartLink file is deleted. The default is one week.

CloudPools ensures the validity of a backed-up SmartLink file within the cloud data retention period. Set the retention periods appropriately to ensure that when the SmartLink file is restored from tape, it remains valid. CloudPools disallows restoring invalid SmartLink files.

CloudPools ensures that a backed-up SmartLink file is still valid by checking the retention periods that are stored for the file. If the retention time is past the restore time, CloudPools prevents NDMP from restoring the SmartLink file.

CloudPools ensures that the account under which the SmartLink files were originally created is not deleted. If it is deleted, both NDMP backup and restore of SmartLink files fail.

Checking the version of SmartLink files

During an NDMP backup session, version data for CloudPools SmartLink files is included in the backup stream.

When restoring data, a version check is performed on the SmartLink files. If the version check determines that the SmartLink files are incompatible with the operating system version running on the target cluster, the NDMP restore session does not restore the SmartLink files to the target cluster and reports the version incompatibilities in the NDMP log.

CloudPools, snapshots, and SnapRevert

The SnapshotIQ, SyncIQ, FSAnalyze, and NDMP Backup functions create point-in-time snapshots of directories in OneFS. Even as files are modified, the snapshot versions are maintained. As part of file matching, CloudPools can include files that have snapshot versions.

CloudPools archives the latest versions of those files to the cloud, and creates local SmartLink files in place of the archived files.

The default CloudPools setting is to allow files with snapshot versions to be archived, but you can change the default setting.

CloudPools also supports SnapRevert for SmartLink files. For example, suppose that CloudPools archived a directory named `/ifs/data/images` to the cloud. The files in the `images` directory would be replaced with SmartLink files.

If you create a SnapRevert domain for the directory, and run the SnapRevert job, the CloudPools archival process is reversed, and the original files are restored to the directory. CloudPools removes any cloud data that was created as part of the original archive process.

In OneFS 8.2 and later, files in SnapRevert (and WORM) domains cannot be SmartLinked. Do not configure a SnapRevert domain in a path that is being archived. Make sure your archive policies do not include SnapRevert directories. Archive jobs that include SnapRevert paths do not complete.

(i) **NOTE:** If you upgrade to OneFS 8.2.x or later from a release earlier than 8.2.X, SnapRevert domains still work the same for already archived files. However, you must not have SnapRevert domains in paths that are being newly archived. Before starting an archive job, manually remove SnapRevert domains (if any exist) from the cloud pool policies in the job. Otherwise, the archive job stalls.

CloudPools and SmartLock

The OneFS SmartLock feature is a software implementation of write once read many (WORM) files. CloudPools and SmartLock are compatible where feasible for WORM file support.

SmartLock supports two types of directories: a Compliance domain and an Enterprise domain.

WORM files in a Compliance domain

There is nothing to gain by configuring a CloudPool policy to archive files in a SmartLock Compliance domain.
- When you create a Compliance domain, the target directories must be empty. This requirement prevents CloudPools SmartLink files from being present in the domain.
- You cannot move existing CloudPools SmartLink files into a Compliance domain. The request is denied.
- You cannot archive existing files in a Compliance domain to the cloud. The CloudPools SmartLink file creation attempt generates an error.

WORM files in an Enterprise domain

In an Enterprise domain, a file is a normal file until it is committed. To become a WORM file, a file must have retention configured and be committed. For CloudPools support, this means the following:
- You can archive a committed file. The CloudPools SmartLink file is successfully created in the Enterprise domain.
- You can read the committed, archived file via the SmartLink file. You cannot edit, rename, move, or delete the file.

- You can recall the committed, archived file, and you can archive it again. However, you cannot edit, rename, delete, or move the recalled file out of the Enterprise domain.

CloudPools and SmartQuotas

With SmartQuotas, an administrator can enforce storage limits for users. Recalling data from the cloud could cause users to exceed assigned storage limits.

When CloudPools archives files to cloud storage, it creates SmartLink files on local storage in place of the archived files. SmartLink files typically take up considerably less storage space than the archived files they replace. When users recall archived files from the cloud, the full files replace the SmartLink files in local storage. Depending on the type of quota that was configured in SmartQuotas, a recall could cause a user to exceed their quota.

There are several types of quota thresholds that can apply to the quota accounting metric.

Physical quotas   A physical quota is implemented by configuring SmartQuota with the `--thresholds-on physicalsize` option. With physical quotas, when cloud data is recalled, the recalled file's physical size is used in the quota accounting metric. When users are close to their assigned quotas, a recall can exceed the quota. For example, suppose a user's quota is 500 MB, and files older than six months are archived to the cloud. This saves the user 250 MB of space, as the SmartLink files take up relatively little local storage space. In the meantime, the user has added more files and now has 400 MB of data in local storage. If the user recalls files from the cloud that would take up more than 100 MB of storage, the user would exceed the quota.

Logical quotas   A logical quota is implemented by configuring SmartQuota with the `--thresholds-on fslogicalsize` option. With logical quotas, when cloud data is recalled, the SmartLink file's logical size is used in the quota accounting metric rather than the recalled physical size. Using the logical size effectively mitigates the risk of exceeding quotas that exists with physical quotas.

As a storage administrator, you might want to use logical quotas where CloudPools is in effect. Otherwise, if physical quotas are used, ensure that your users are aware of the possibility of exceeding the quota and how best to mitigate the issue.

CloudPools and SmartDedupe

SmartDedupe scans the OneFS file system for files that contain identical blocks of data. If SmartDedupe finds duplicate blocks, SmartDedupe moves a single copy of the blocks to a hidden file called a shadow store. SmartDedupe then deletes the duplicate blocks from the original files and replaces the blocks with pointers to the shadow store.

CloudPools interacts as follows with SmartDedupe:

- If a file pool policy specifies that de-duplicated files should be archived to cloud storage, CloudPools archives those de-duplicated files and leaves SmartLink files in their place in local storage.
- When an archived file that had been de-duplicated is recalled from the cloud, the SmartLink file is replaced and the recalled file placed back in local storage is no longer de-duplicated.
- SmartDedupe does not de-duplicate SmartLink files.

CloudPools tips and troubleshooting

Topics:

CloudPools best practices

For best results using CloudPools, follow these best practices.

Use time stamps for cloud data archival and recall

Use time matching patterns (creation, modification, last access) when you archive data to and recall data from the cloud. This enables more efficient archival and recall operations, and therefore better performance.

When you create a file pool policy for archiving data to the cloud, several of the file-matching criteria involve time:

- Created
- Accessed
- Modified

Therefore, you can specify file-matching criteria that specify when the files were created, when files were last accessed, or when they were last modified. Note that some files (program files, libraries, scripts and so forth) are not changed after they are created; if modified time is used to archive, these files might be pushed to cold storage. This can result in slow application load, for example, for often used files. Access time would be a better criterion for these types of files.

CloudPools archiving and file size

You can gain the most benefit from CloudPools, in terms of freeing up storage space on your cluster, by archiving larger files. Archiving small files provides less, if any, benefit.

(i) **NOTE:** Do not archive files of 32 kibibytes (KiB) or less. Doing so does not save any space on the cluster.

One of the benefits of archiving files to the cloud with CloudPools is how quickly you can recall these files when needed.

To enable fast recall, CloudPools creates a SmartLink file for every file whose data is archived to the cloud. SmartLink files each contain a map to the data in the cloud, metadata, and cache space. SmartLink files are generally small in size, but can grow if data is cached through inline access.

Therefore, if you archive small files to the cloud, SmartLink files are left in their place on the cluster, and could approach, or even exceed, the size of the original file.

Create exclusive accounts for CloudPools purposes

You should create an account with your cloud provider that is exclusively for CloudPools use. This prevents conflicts that might lead to data corruption or loss.

If your organization accesses cloud provider accounts outside of OneFS CloudPools operation, users must be careful not to in any way access or change data archived by CloudPools. Any such data access or modification would likely corrupt the data and compromise data retrieval and recall from CloudPools.

To prevent this, create an account in CloudPools that is exclusively for CloudPools use. Use entirely separate accounts for other cloud applications with your cloud provider.

CloudPools limitations and expected behaviors

During normal CloudPools operation, you should be aware of the following limitations and expected behaviors.

Rolling upgrade before CloudPools usage	If you are performing a rolling upgrade to a OneFS 8.2.x version, and intend to use CloudPools for the first time, wait until the upgrade and commit complete. By waiting for the upgrade to complete, you start using CloudPools with the most recent CloudPools upgrades. Upgrades to newer versions of the CloudPools stub files (SmartLink files) are not required.
Cloud storage account deletion	⚠ **CAUTION: Do not delete a cloud storage account that is in use by archived files.**
	Doing so can lead to data loss or unavailability for the archived files that use the deleted account. Any attempt to open SmartLink files that are associated with a deleted account fails with I/O error messages. In addition, NDMP backup and restore and SyncIQ failover and failback fail when a related cloud storage account is deleted. If an NFS or SMB user attempts to open a SmartLink file for inline access and receives an I/O error, it may indicate that the related cloud storage account was deleted. Dell recommends trying inline access of other SmartLink files in the same CloudPool. If the same error is generated for those files, it means that the cloud storage account is deleted and data is lost. If the other SmartLink files are accessible, the SmartLink file that generated the error might be corrupted. Either way, contact Technical Support for assistance.
Accessing SmartLink files	You can view and modify cloud data by accessing SmartLink files.
SmartLink file timestamps	Opening a SmartLink file through a supported protocol can change the timestamp data. When a file is first archived and the SmartLink file is created, the ctime timestamp stays the same as the timestamp on the original file. However, the first time the SmartLink file is opened (inline access), the ctime timestamp changes because a cache component is added to the file.
Inline access can appear to convert a SmartLink file to a regular file	When a user accesses a SmartLink file on the PowerScale cluster using a supported protocol, the file opens in an application on the client system. During this process, called inline access, most applications support the creation of a CloudPools cache from which users can view and modify archived data. With inline access, the SmartLink file remains intact on the cluster. Modifications that the user makes to file data are stored in the cache and updated to the cloud. However, some applications do not support inline access. Instead, these applications create a copy of the original file apart from the SmartLink file. The new file, containing all original file data, is given a new logical I-node (LIN) number and timestamps that differ from the file that was originally archived. This behavior has been observed in only a few programs, including Microsoft Office applications. In these cases, an entirely new file is created. The original SmartLink file and its associated data in the cloud is tagged for removal (garbage collection). If the new file meets the criteria of the file pool policy that archived the original file to the cloud, the new file is archived the next time the SmartPools job runs. A new SmartLink file is created in its place on the local cluster. If the new file does not meet the policy criteria, the full file remains on the cluster.
	For best results using CloudPools, Dell recommends that you avoid archiving files that users are still actively modifying.
Client-based tools and SmartLink files	If you run a client-based tool such as AVScan (anti-virus scan) or a backup application, file data in the cloud is fully cached back to the SmartLink files. This activity can result in heavy network usage and increased service provider costs, and negates space saving on your cluster.
Expired SmartLink files	Expired SmartLink files are not restored using NDMP and do not synch back using SyncIQ. A SmartLink file on an NDMP backup or on a SyncIQ secondary (target) cluster is expired when either of the following is true: ● The original SmartLink file is deleted from the primary (source) cluster. ● The original file data in the cloud is fully recalled.
Recall interrupted	When a full cache is in process (that is, someone performed an inline access of a SmartLink file), recall of the same file can fail. The full cache is allowed to complete first, and the user should retry the recall after caching is completed.
ADS files	CloudPools does not archive and recall ADS (alternate data stream) files.
SMB Oplock	SMB Oplock (lease/notification) does not work in cases where you create a file with the SUPERCEDE flag, the file already exists, and is archived.

CloudPools logs

You can access CloudPools logs to view activity and troubleshoot problems.

The following logs are available in OneFS for CloudPools operation.

Type	Name	Path
Client cluster-side logs	Cpool daemon	`/var/log/isi_cpool_d.log`
	Job Engine	`/var/log/isi_job_d.log`
	I/O	`/var/log/isi_cpool_io_d.log` and `/var/log/lwiod.log` `isi_cpool_io_d.log` also applies to cluster-local access
	Provisioning	`/var/log/isi_papi_d.log`
	NDMP	`/var/log/isi_ndmp_d.log`
	SyncIQ	`/var/log/isi_migrate.log`
	Messages	`/var/log/messages`
Platform API cloud-side logs	Platform API (RAN)	`/var/log/isi_object_d.log`
	HTTPd apache	`/var/log/apache2/webui_httpd_error.log` and `/var/log/apache2/webui_httpd_access.log`
	Session authentication Messages	`/var/log/messages`

> (i) **NOTE:** Make sure that the client cluster-side time is accurate to within 15 minutes of the cloud provider. We recommend configuring NTP on the cluster to maintain the correct time.

Troubleshooting CloudPools

This section describes other troubleshooting items for CloudPools administration and operation.

Cloud storage account cannot connect to the cloud

In the OneFS, if a cloud storage account is shown in the web administration interface with a red `Needs Attention` icon, or in the CLI interface with an `Unreachable` state, this usually indicates that the cluster has lost Internet connectivity or the service provider's cloud storage facility is offline. Ensure that the cluster has Internet connectivity. If it does, contact your service provider for help.

Determining if a file is a SmartLink file

To determine if a file is archived to the cloud, you can check whether the local version of the file is a SmartLink file. A quick way to check is by using the `ls -lo` command.
1. Run `ls -lo <file-name>`
2. Look for `ssmartlinked` in the output.

```
server1c-1# ls -lo /ifs/test_20200116_090317
-rw-r--r--  1 root  wheel  inherit,writecache,wcinherit,ssmartlinked
3145728 Jan 16 09:03 /ifs/test_20200116_090317
```

An alternate way to check is with the `isi get -D` command.
1. Run `isi get -D <file-name>`
2. Look for the value in the `SmartLinked` property:

```
server1c-1# isi get -D koala.jpg | grep SmartLinked:
* SmartLinked:   True
```

CloudPools CLI commands

This section is a reference for all commands related to CloudPools in the OneFS command line interface.

For any command, use the `--help` option to show a list of command options.

Topics:

isi antivirus icap settings modify

Sets and displays global configuration settings for ICAP antivirus scanning.

Syntax

```
isi antivirus icap settings modify
   [--fail-open {true | false}]
   [{--glob-filters <string>... | --clear-glob-filters
   | --add-glob-filters <string> | --remove-glob-filters <string>}]
   [--glob-filters-enabled {true | false}]
   [--glob-filters-include {true | false}]
   [--path-prefixes <path>... | --clear-path-prefixes
   | --add-path-prefixes <path> | --remove-path-prefixes <path>}]
   [--repair {true | false}]
   [--report-expiry <integer><time>]
   [--scan-cloudpool-files{true | false}]
   [--scan-on-close {true | false}]
   [--scan-on-open {true | false}]
   [--scan-size-maximum <integer>{k | M | G | T | P}]
   [--service {true | false}]
   [--quarantine {true | false}]
   [--truncate {true | false}]
   [--verbose]
```

Options

--fail-open {true | false}

> If `--scan-on-open` is set to `true`, determines whether users can access files that cannot be scanned. If this option is set to `false`, users cannot access a file until the file is scanned by an ICAP server.
>
> If `--scan-on-open` is set to `true`, this option has no effect.

--glob-filter *<string>*

> Specifies a file name or extension. To specify multiple filters, you must include multiple `--glob-filter` options within the same command. Specifying this option will remove any existing glob filters.
>
> All strings that include wildcards must be enclosed in quotes, for example, "`*`.jpg".

Table 5. Wildcards

Wildcard character	Description
*	Matches any string in place of the asterisk. For example, specifying "`m*`" would match "movies" and "m123"
[ ]	Matches any characters contained in the brackets, or a range of characters separated by a dash. For example, specifying "`b[aei]t`" would match "bat", "bet", and "bit" For example, specifying "`1[4-7]2`" would match "142", "152", "162", and "172" You can exclude characters within brackets by following the first bracket with an exclamation mark. For example, specifying "`b[!ie]`" would match "bat" but not "bit" or "bet" You can match a bracket within a bracket if it is either the first or last character. For example, specifying "`[[c]at`" would match "cat", and "[at"

Table 5. Wildcards (continued)

Wildcard character	Description
	You can match a dash within a bracket if it is either the first or last character. For example, specifying `"car[-s]"` would match "cars", and "car-"
?	Matches any character in place of the question mark. For example, specifying `"t?p"` would match "tap", "tip", and "top"

(i) **NOTE:** If you specify this option, the specified filters will replace all previously specified filters in the list.

--clear-glob-filters

Clears the list of filters.

--add-glob-filters <string>

Adds the specified filters to the list of filters.

--remove-glob-filters <string>

Removes the specified filters to the list of filters.

--glob-filters-enabled {true | false}

Determines whether glob filters are enabled. If no glob filters are specified, glob filters will remain disabled even if this option is set to `true`.

--glob-filters-include {true | false}

Determines how glob filters are interpreted by OneFS. If set to `true`, OneFS will scan only files that match a glob filter. If set to `false`, OneFS will scan only files that do not match any glob filters.

--path-prefix *<path>*

If specified, only files contained in the specified directory path will be scanned. This option affects only on-access scans. To specify multiple directories, you must include multiple `--path-prefix` options within the same command. Specifying this option will remove any existing path prefixes.

(i) **NOTE:** If you specify this option, the specified filters will replace all previously specified filters in the list.

--clear-path-prefixes

Clears the list of paths.

--add-path-prefixes <path>

Adds the specified paths to the list of paths.

--remove-path-prefixes <path>

Removes the specified paths to the list of paths.

--repair {true | false}

Determines whether OneFS attempts to repair files that threats are detected in.

--report-expiry *<integer>* *<time>*

Determines how long OneFS will retain antivirus scan reports before deleting them.

The following *<time>* values are valid:

Y	Specifies years
M	Specifies months
W	Specifies weeks
D	Specifies days
H	Specifies hours
m	Specifies minutes

s	Specifies seconds

--scan-cloudpool-files {true | false}

    Determines whether cloudpool files are scanned for antiviruses..

--scan-on-close {true | false}

    Determines whether files are scanned after the files are closed.

--scan-on-open {true | false}

    Determines whether files are scanned before the files are sent to users.

--scan-size-maximum *<integer>*{k | M | G | T | P}

    If specified, OneFS will not send files larger than the specified size to an ICAP server to be scanned.

    ⓘ **NOTE:** Although the parameter accepts values larger than `2GB`, OneFS does not scan files larger than `2GB`.

--service {true | false}

    Determines whether the antivirus service is running.

--quarantine {true | false}

    Determines whether OneFS quarantines files that threats are detected in. If `--repair` is set to `true`, OneFS will attempt to repair the files before quarantining them. If both `--truncate` and `--quarantine` are set to `true`, the `--truncate` option is ignored.

--truncate {true | false}

    Determines whether OneFS truncates files that threats are detected in. If `--repair` is set to `true`, OneFS will attempt to repair the files before truncating them. If both `--truncate` and `--quarantine` are set to `true`, the `--truncate` option is ignored.

{--verbose | -v}

    Displays a message confirming that the settings have been modified.

isi cloud access add

Adds cloud write access to the cluster for all accounts that are associated with the cluster GUID that was supplied.

Syntax

```
isi cloud access add <guid>
  [--expiration-date]<timestamp>
  [--force
  [--verbose]
```

Options

<guid>

    The GUID of the cluster that will be granted access to buckets for all cloud accounts.

--expiration-date *<timestamp>*

    The date and time when unreferenced (stale) data will be removed from the cloud. The timestamp format is *MMDDYY:hh:mm*. For example, `022016:12:00` specifies an expiration date and time of February 20, 2016 at 12:00 PM.

    Use this command to set the date when all currently unreferenced objects or files in the cloud will be deleted from the cloud.

{-- force | -f}

    Do not ask for confirmation.

{--verbose | -v}

    Displays more detailed information.

Examples

The following example adds cloud write access to a cluster by specifying the cluster GUID and an expiration date:

```
isi cloud access add 000556bf1e82059801563f1ad44a8c155acf
--expiration-date 022016:12:00
```

OneFS displays a message indicating which cloud accounts and file pool policies the secondary cluster gain access to, and requires confirmation. Type **yes**, and press ENTER to complete the process.

ⓘ **NOTE:** Only a single cluster should have access to the cloud accounts and pools for a specific GUID at a given time.

isi cloud access list

Displays a list of cluster GUIDs and associated cloud accounts and buckets that the cluster has, or is eligible for write access to cloud data. Available clusters are the primary cluster and any other clusters to which data has been replicated with SyncIQ or restored with NDMP. Only one cluster at a time should have write access to cloud data for a particular cluster or GUID.

Syntax

```
isi cloud access list
  [--limit | -l]<integer>
  [--sort {name | guid | state}]
  [--descending | -d]
  [--format {table | json | csv | list}]
  [--no-header | -a]
  [--no-footer | -z]
  [--verbose | -v]
```

Options

{--limit | -l} *<integer>*

  Limits the number of eligible clusters displayed in the list.

--sort {name | guid | state}

  Sorts the list of eligible clusters according to the specified category.

--format {table | json | csv | list}

  Displays output in table (default), JavaScript Object Notation (JSON), comma-separated value (CSV), or list format.

{--descending | -d}

  Outputs the list of eligible clusters in descending order according to the specified sort option.

{--no-header | -a}

  Displays table and CSV output without headers.

{--no-footer | -z}

  Displays table output without footers.

{--verbose | -v}

  Displays more detailed information.

isi cloud access remove

Removes cloud resource access from the specified cluster.

Syntax

```
isi cloud access remove <guid>
 [--force]
 [--verbose]
```

Options

<guid>

> The reference number, or globally unique identifier (GUID), of the cluster where accounts or policies have been created.

{--force|-f}

> Execute the command without requiring confirmation.

{--verbose|-v }

> Displays more detailed information.

Examples

The following example removes cloud write access from a cluster identified by a specified GUID:

```
isi cloud access remove 000556bf1e82059801563f1ad44a8c155acf
```

OneFS displays a message indicating the cloud accounts and file pool policies to which the cluster will no longer have access, and requires confirmation. Type **yes**, and press ENTER to complete the process.

isi cloud access view

Displays the details of a cluster that either has or is eligible for write access to cloud data.

Syntax

```
isi cloud access view <guid>
```

Options

<guid>

> The reference number, or globally unique identifier (GUID), of the cluster.

isi cloud accounts create

Creates a cloud storage account that connects CloudPools to your cloud storage provider.

Syntax

```
isi cloud accounts create <name> <type> <uri>
  [--account-username <string>]
  [--key <string>]
  [--enabled {yes | no}]
  [--account-id <string>]
  [--telemetry-bucket <string>]
  [--storage-region <string>]
  [--skip-ssl-validation {yes | no}]
  [--enable-ocsp {yes | no}]
  [--ocsp-responder-url-required {yes | no}]
  [--proxy <string>]
  [--credential-provider-uri <string>]
  [--credential-provider-agency <string>]
  [--credential-provider-certificate <string>]
  [--credential-provider-mission <string>]
  [--credential-provider-proxy <string>]
  [--credential-provider-role <string>]
  [--force]
  [--verbose]
```

Options

<name>

The name of the cloud storage account.

<type>

The type of cloud storage account. Use one of the following values:

Table 6. *<type>*

Value	Description
isilon	Dell Technologies PowerScale
ecs	Dell Technologies ECS Appliance
azure	Microsoft Azure
s3	Amazon S3
c2s-s3	Amazon Commercial Cloud Services S3
google	Google Cloud Platform (using interoperability access)
alibaba-cloud	Alibaba Cloud

<uri>

The cloud account URI. This URI must match that provided to the cloud vendor.

{--account-username | -u} *<string>*

The username for the cloud account. This name must be identical to the user name provided by the cloud vendor.

--key *<string>*

The cloud account access key or password. This information is provided by the cloud vendor.

--enabled {yes | no}

Enables or disables the account at creation time. By default, a cloud storage account is enabled on creation. To disable an account on creation, use this setting with the `no` option.

`--account-id` *<string>*

A required Amazon S3-only setting. The account ID number provided by Amazon when you first establish an account with the vendor.

`--telemetry-bucket` *<string>*

A required Amazon S3-only setting. The telemetry bucket name that you specified when you first established an account with the vendor. This is where usage reports are stored.

`--storage-region` *<string>*

An optional parameter for Amazon S3 or Google Storage Platform cloud types. The region value must match the storage region that you specified when you first established an account with the cloud provider. For example, `us-west-1`. If you do not specify a region, the cloud provider chooses its default region.

`--skip-ssl-validation` `{yes | no}`

Specifies whether to circumvent SSL certificate validation when connecting to a cloud provider's storage repository. Unless you specify this setting with a `yes` instruction, OneFS will attempt to perform SSL certificate validation when connecting. For security purposes, we recommend not enabling this setting. If you are connecting to a cloud provider that is within your corporate network (for example, PowerScale or ECS), and you are having trouble connecting, you can skip SSL validation.

`--enable-ocsp` `{yes | no}`

Applies only to the C2S-S3 cloud type. It indicates whether to use OCSP to check the revocation status of the authentication certificate.

`--ocsp-responder-url-required` `{yes | no}`

Applies only to the C2S-S3 cloud type. It indicates whether a certificate without an OCSP responder URL is considered valid or not.

`--proxy` *<string>*

The network proxy through which CloudPools traffic to and from a public cloud provider should be redirected. The specified network proxy must already have been created with the `isi cloud proxies create` command.

`--credential-provider-uri` *<string>*

Applies only to the C2S-S3 cloud type. The URI to connect to a credential provider.

`--credential-provider-agency` *<string>*

Applies only to the C2S-S3 cloud type. The agency name required to connect to the credential provider.

`--credential-provider-certificate` *<string>*

Applies only to the C2S-S3 cloud type. The name or id of a certificate to connect to the credential provider.

`--credential-provider-mission` *<string>*

Applies only to the C2S-S3 cloud type. The Mission name required to connect to the credential provider.

`--credential-provider-proxy` *<string>*

Applies only to the C2S-S3 cloud type. The name or id of a proxy to connect to the credential provider.

`--credential-provider-role` *<string>*

Applies only to the C2S-S3 cloud type. The role name required to connect to the credential provider.

`{--force | -f }`

Execute the command without requiring confirmation.

`{--verbose | -v}`

Displays more detailed information.

Examples

The following example creates a Microsoft Azure cloud account:

```
isi cloud accounts create my_azure azure https://myazure.windows.net myuser
dhgXJ9OAIahXvYmL
```

isi cloud accounts delete

Deletes a cloud storage account.

⚠️ **CAUTION: Deleting an account results in the permanent loss of access to the data. In effect, you are deleting the data.**

Syntax

```
isi cloud accounts delete <id>
[--acknowledge <string>]
[--verbose]
```

Options

<id>

> The name of the cloud account. You can use the `isi cloud accounts list` command to display the names of cloud accounts.

--acknowledge *<string>*

> Enables the account deletion to proceed. This parameter is required. You must include a text string with the parameter, such as **yes**, **proceed**, or other string.

{--verbose | -v}

> Displays more detailed information.

Example

The following example deletes a Microsoft Azure cloud account:

```
isi cloud accounts delete my_azure --acknowledge yes
```

When you run the command, OneFS displays the following message and requires confirmation:

```
***********************************************************************
WARNING: Deleting an account is extremely dangerous.
Continuing with this operation will result in a permanent loss of data.
Type 'confirm delete data' to proceed.  Press enter to cancel:
```

To proceed, type **confirm delete data**, and press ENTER.

isi cloud accounts list

Lists cloud accounts.

Syntax

```
isi cloud accounts list
  [--limit <integer>]
  [--sort {id | name | type | account_username | uri | state | bucket}]
  [--descending]
  [--format {table | json | csv | list}]
  [--no-header]
  [--no-footer]
  [--verbose]
```

Options

{--limit | -l} *<integer>*

> Limits the number of cloud accounts displayed in the list.

--sort {id | name | type | account_username | uri | state | bucket}

> Sorts the list of cloud accounts according to the specified category.

--format {table | json | csv | list}

> Displays output in table (default), JavaScript Object Notation (JSON), comma-separated value (CSV), or list format.

{--descending | -d}

> Outputs the list of cloud accounts in descending order according to the specified sort option.

{--no-header | -a}

> Displays table and CSV output without headers.

{--no-footer | -z}

> Displays table output without footers.

{--verbose | -v}

> Displays more detailed information.

isi cloud accounts modify

Modifies a cloud account.

Syntax

```
isi cloud accounts modify <id>
  [--name <string>]
  [--account-username <string>]
  [--key <string>]
  [--uri <string>]
  [--enabled {yes | no}]
  [--account-id <string>]
  [--telemetry-bucket <string>]
  [--storage-region <string>]
  [--skip-ssl-validation {yes | no}]
  [--enable-ocsp {yes | no}]
  [--ocsp-responder-url-required {yes | no}]
```

```
[{--proxy <string> | --clear-proxy}]
[--skip-account-check {yes | no}]
[--credential-provider-uri <string>]
[--credential-provider-agency <string>]
[--credential-provider-certificate <string>]
[--credential-provider-mission <string>]
[--credential-provider-proxy <string>]
[--credential-provider-role <string>]
[--verbose]
```

Options

<id>

The ID of the cloud account. In this case, the ID is the same as the cloud account name.

{--name | -n} <string>

The name of the cloud account. In this case, the name is the same as the ID.

{--account-username | -u} <string>

The username for the cloud account. This name must be identical to the user name provided by the cloud vendor.

--key <string>

The cloud account access key or password. This information is provided by the cloud vendor.

--uri <string>

The cloud account URI. This URI must match that provided to the cloud vendor.

--enabled {yes | no}

By default, when you create a cloud storage account, it is enabled. To disable the account on creation, you can use this setting with the `no` option.

--account-id <string>

This is a required Amazon S3-only setting. The account ID number provided by Amazon when you first establish an account with the vendor.

--telemetry-bucket <string>

This is a required Amazon S3-only setting. The telemetry bucket name that you specified when you first established an account with the vendor.

--storage-region <string>

This is a required Amazon S3, Google Cloud Platform, Alibaba Cloud setting. The storage region that you specified when you first established an account with the vendor. For example, `us-west-1`.

--skip-ssl-validation {yes | no}

Specifies whether to circumvent SSL certificate validation when connecting to a cloud provider's storage repository. Unless you specify this setting with a `yes` instruction, OneFS will attempt to perform SSL certificate validation when connecting. For security purposes, we recommend not enabling this setting. If you are connecting to a cloud provider (for example, RAN or ECS) that is inside your corporate network, and you are having trouble connecting, you can skip SSL validation.

--enable-ocsp {yes | no}

Applies only to the C2S-S3 cloud type. It indicates whether to use OCSP to check the revocation status of the authentication certificate.

--ocsp-responder-url-required {yes | no}

Applies only to the C2S-S3 cloud type. It indicates whether a certificate without an OCSP responder URL is considered valid or not.

{--proxy <string> | --clear-proxy}

Use `--proxy` to set or change a network proxy through which CloudPools traffic is redirected, on its way to and from a public cloud provider. The specified network proxy must already have been created with the `isi cloud proxies create` command.

Use `--clear-proxy` to remove a previously set proxy. When you remove a proxy, CloudPools traffic flows directly to the cloud provider.

--skip-account-check {yes | no}

If set to yes, CloudPools skips the validation step to determine if the cloud storage account is still accessible. We do not recommend skipping this check.

`{--verbose|-v}`

Displays more detailed information.

Example

The following example modifies a Microsoft Azure cloud account:

```
isi cloud accounts modify my_azure
--uri https://myazure.windows.net
--account-username myuser --key dhgXJ9OAIahXvYmL
```

isi cloud accounts view

Displays the details of a cloud account.

Syntax

```
isi cloud accounts view <id>
```

Options

<id>

Specifies the id of the cloud account to view. You can use the `isi cloud accounts list` command to obtain ids of available cloud accounts.

isi cloud archive

Queues one or more files to be archived to the cloud. For files to be archived, they must match the specified file pool policy, or any file pool policy with a cloud target.

Syntax

```
isi cloud archive <files>
  [--recursive {yes | no}]
  [--policy <string>]
  [--verbose]
  [--help]
```

Options

<files>

Specifies the files to archive. Specify `--files` for each additional file to process. Alternatively, you can specify a file matching pattern such as `/ifs/data/archive/images/*.jpg`.

`{--recursive|-r}{yes|no}`

Specifies whether the operation should apply recursively to nested directories in the file string.

`--policy <string>`

Specifies the file pool policy to appy to the specified files. If you specify one or more files to be archived and do not specify a policy, OneFS will compare the files with each configured file pool policy.

{--verbose | -v}

Displays more detailed information.

Examples

The following example archives multiple files to the cloud according to a specific file pool policy:

```
isi cloud archive /ifs/data/images/big.jpg --file /ifs/data/huge.jpg
--policy my_policy
```

The following example archives an entire directory to the cloud. The operation must match an existing file pool policy to be successful.

```
isi cloud archive /ifs/data/images/ --recursive yes
```

isi cloud jobs cancel

Cancels a CloudPools job initiated manually with `isi cloud archive` or `isi cloud recall`. You cannot cancel CloudPools system jobs (such as cache-writeback).

Syntax

```
isi cloud jobs cancel <id>
  [--verbose]
```

Options

<id>

The ID for the cloud job. Run `isi cloud jobs list` to see a list of all manual and system jobs and their associated IDs.

{--verbose | -v}

Displays more detailed information.

Example

This following example cancels a CloudPools job with the ID of 21.

```
isi cloud job cancel 21
```

isi cloud jobs create

Creates a CloudPool job to archive or recall files.

Syntax

```
isi cloud jobs create {archive | recall | restore-coi}
[--files <string>]
```

```
[{--recursive | -r} {yes | no}]
[--accounts <string>]
[--expiration-date <timestamp>]
[--policy <string>]
[{--verbose | -v}]
```

Options

{archive | recall | restore-coi}

> The type of job. Valid entries are `archive`, `recall`, and `restore-coi`.

--files *<string>...*

> Specifies files or directory names to which the job applies. Use `--files` for each additional specification. Directory names are valid only for archive jobs.

{--recursive | -r} {yes | no}

> Specifies whether the job should apply recursively to nested directories.

--accounts *<string>*

> Specifies the cloud accounts to be used for restore-coi jobs. To specify additional cloud accounts, use the `--accounts` option.

--expiration-date *<timestamp>*

> The expiration date for orphan objects. Use one of the following formats for *<timestamp>*:
> - A date string matching the pattern `'YYYY-MM-DD'`
> - A date/time string matching the pattern `'YYYY-MM-DD[Thh:mm[:ss]]'`

--policy *<string>*

> The policy to use in an archive job.

{--verbose | -v}

> Displays more detailed information.

Examples

For `archive` jobs, you can specify one or more directories to archive. The following command archives a single directory:

```
isi cloud jobs create archive --files /ifs/shares/dir1
```

The following example archives multiple directories:

```
isi cloud jobs create archive --files /ifs/shares/dir1 --files /ifs/shares/dir2
```

isi cloud jobs files list

Displays the list of files matched by the specified CloudPools job.

Syntax

```
isi cloud jobs files list <job-id>
  [--offset <integer>]
  [--page <integer>]
  [--id <boolean>]
  [--limit <integer>]
  [--format {table | json | csv | list}]
  [--no-header]
  [--no-footer]
  [--verbose]
```

Options

<job-id>

The ID of the job. To find the list of job IDs in CloudPools, run the `isi cloud jobs list` command.

--offset *<integer>*

The starting file ID number to display.

--page *<integer>*

Used with limit option. If present, specifies the starting page number to display where page size is specified by limit. This option will be deprecated; please use offset option instead.

--id *<boolean>*

Adds an ID number in the display before each file.

{--limit | -l} *<integer>*

Display no more than the specified number of items.

--format {table | json | csv | list}

Displays output in table (default), JavaScript Object Notation (JSON), comma-separated value (CSV), or list format.

{--no-header | -a}

Displays table and CSV output without headers.

{--no-footer | -z}

Displays table output without footers.

{--verbose | -v}

Displays more detailed information.

Example

The following example displays a list of files associated with a specific cloud job:

```
isi cloud jobs files list 21
```

isi cloud jobs list

Lists the status of CloudPools jobs, including system, archive, and recall jobs.

Syntax

```
isi cloud jobs list
[--limit <integer>]
[--sort {id | job_state | operation_state |effective_state | type
| state_change_time | completion_time | create_time | description}]
[--descending ]
[--format {table | json | csv | list}]
[--no-header]
[--no-footer]
[--verbose]
```

Options

{--limit | -d} *<integer>*

Displays no more than the specified number of items.

`--sort {id | job_state | operation_state | effective_state | type | state_change_time |`
`completion_time | create_time | description}`

Orders results by this field. The default value is `id`. Note that, to sort on other than ID, description, effective state, and type, use the `--verbose` parameter with the command.

`{--descending | -d}`

Sorts and presents data in descending order.

`--format {table | json | csv | list}`

Display output in table (default), JavaScript Object Notation (JSON), comma-separated value (CSV), or list format.

`{--no-header | -a}`

Displays table and CSV output without headers.

`{--no-footer | -z}`

Displays table output without footers.

`{--verbose | -v}`

Displays more detailed information.

isi cloud jobs pause

Pause a cloud job. You can resume a paused job with the `isi cloud jobs resume` command.

Syntax

```
isi cloud jobs pause <id>
  [--verbose | -v]
```

Options

id

The ID of the cloud job to pause.

To find job IDs, use the `isi cloud jobs list` command to view the IDs of all cloud jobs. Although possible, we recommend that you not pause any of the CloudPools system jobs that run in the background and are critical for proper operation. These include:

Table 7. *id*

ID	Description	Effective State	Type
1	Write updated data to the cloud	running	cache-writeback
2	Expire CloudPools cache	running	cache-invalidation
4	Clean up unreferenced data in the cloud	running	cloud-garbage-collection
5	Write updated snapshot data to the cloud	running	snapshot-writeback
6	Update SmartLink file formats	running	smartlink-upgrade
7	Add data to CloudPools cache	running	cache-pre-populate

`{--verbose | -v}`

Displays more detailed information.

Example

The following example pauses a cloud job with ID 19.

```
isi cloud jobs pause 19
```

isi cloud jobs resume

Resumes a paused cloud job, or resumes all paused cloud jobs.

Syntax

```
isi cloud jobs resume <id>
  [--verbose | -v]
```

Options

<id>

The ID for the cloud job to resume. Use the `isi cloud jobs list` command to view a list of jobs and their associated IDs.

--verbose | -v

Displays more detailed information.

Example

The following command resumes a paused job with an ID of 26:

```
isi cloud jobs resume 26
```

isi cloud jobs view

Shows the details of a cloud job.

Syntax

```
isi cloud jobs view <id>
```

Options

<id>

Specify the ID of the cloud job. Use the `isi cloud jobs list` command to view all jobs and their associated IDs.

Example

The following command shows the details of a job with the ID of 27:

```
isi cloud jobs view 27
```

isi cloud pools create

Creates a CloudPool, which provides the connection between OneFS and a cloud storage account.

Syntax

```
isi cloud pools create <name> <type> <account>
  [{--description | -d} <string>]
  [--vendor <string>]
  [{--verbose | -v}]
```

Options

<name>

The name of the CloudPool.

<type>

The type of account. Valid values are: `isilon` (for PowerScale), `ecs`, `azure`, `s3`, `c2s-s3` , `google`, or `alibaba-cloud`.

<account>

The name of the cloud storage account to which the CloudPool connects. The cloud storage account is required and must match the CloudPool type. Only one cloud storage account can be specified.

--description | -d *<string>*

A description of the CloudPool.

--vendor *<string>*

The name of the vendor hosting the cloud storage account.

--verbose | -v

Displays more detailed information.

Example

This following command creates a CloudPool containing a Microsoft Azure cloud storage account:

```
isi cloud pools create my_cp azure http://myazure.microsoft.com
--description="Financial records 2013" --vendor=Microsoft
```

isi cloud pools delete

Deletes a CloudPool. Proceed with caution. If you delete a CloudPool, OneFS is no longer able to access objects or files that are stored in that pool using the associated account. If the CloudPool is referenced by a file pool policy, OneFS does not allow you to delete the CloudPool.

Syntax

```
isi cloud pools delete <id>
 [--force | -f]
 [--verbose | -v]
```

Options

<id>

The name of the CloudPool. You can use the `isi cloud pools list` command to list existing CloudPools and their associated IDs.

--force | -f

Deletes the pool without asking for confirmation.

--verbose | -v

Displays more detailed information.

Example

The following command deletes a CloudPool:

```
isi cloud pool delete my_azure_pool
```

When you press ENTER to run the command, OneFS asks for confirmation. Type **yes**, and then press ENTER.

isi cloud pools list

Displays a list of CloudPools.

Syntax

```
isi cloud pools list
[--limit <integer>]
[--sort {id | name | type | description | vendor}]
[--descending]
[--format {table | json | csv | list}]
[--no-header]
[--no-footer]
[--verbose]
```

Options

{--limit | -l} <integer>

Displays no more than the specified number of items.

--sort {id | name | type | description | vendor}

Orders results by this field. The default value is `id`, which, in this case, is the same as `name`. Unless you use the `--verbose` option, you can only sort on `name` or `type`.

`{--descending|-d}`

Sorts and presents data in descending order.

`--format {table|json|csv|list}`

Displays output in table (default), JavaScript Object Notation (JSON), comma-separated value (CSV), or list format.

`{--no-header|-a}`

Displays table and CSV output without headers.

`{--no-footer|-z}`

Displays table output without footers.

`{--verbose|-v}`

Displays more detailed information.

isi cloud pools modify

Modifies a CloudPool.

Syntax

```
isi cloud pools modify <id>
  [--name <string>]
  [--accounts <string> | --clear-accounts | --add-accounts <string> | --remove-accounts
<string>]
  [--description <string>]
  [--vendor <string>]
  [--verbose]
```

Options

`<id>`

The ID of the CloudPool. Run `isi cloud pools list` to view the IDs of all CloudPools.

`{--name | -n} <string>`

Specifies a new name for the CloudPool.

`{--accounts <string> | --clear-accounts | --add-accounts <string> | --remove-accounts <string>}`

Adds or removes accounts associated with this cloudpool.

Only one account per CloudPool is allowed. To change the account associated with a CloudPool, we recommend the following:
- Create a new CloudPool using the `isi cloud pools create` command, specifying the correct account.
- Delete the old CloudPool using the `isi cloud pools delete` command. Proceed with caution. If you delete a CloudPool, OneFS is no longer able to access the associated cloud storage account.

`{--description | -d} <string>`

Provides a description of this cloud pool.

`---vendor <string>`

Provides the name of the vendor hosting the cloud pool account.

`{--verbose | -v}`

Displays more detailed information.

Examples

The following command adds a vendor name and description to an existing CloudPool:

```
 isi cloud pools modify my_azure --vendor Microsoft
--description "preferred azure account"
```

The following command removes a cloud account from the CloudPool:

```
isi cloud pools modify my_s3 --remove-accounts s3_acct_1
```

isi cloud pools view

View detailed information about a CloudPool.

Syntax

```
isi cloud pools view <id>
```

Options

<id>

The ID of the cloud pool. Run the `isi cloud pool list` command to view all CloudPools and their associated IDs.

Example

The following command displays information about a CloudPool named `my_azure_pool`.

```
isi cloud pools view my_azure_pool
```

isi cloud proxies create

Creates a network proxy through which a cloud storage account can connect to a cloud storage provider.

Syntax

```
isi cloud proxies create <name> <host> <type> <port>
 [{--username | -u} <string>]
 [{--password | -p} <string>]
 [{--verbose | -v}]
```

Options

<name>

The name of the network proxy. This can be any alphanumeric string, but should be a simple, recognizable name.

<host>

The DNS name or IP address of the proxy server. For example, `myproxy1.example.com` or `192.168.107.107`.

<type>

The proxy protocol type, one of `socks_4`, `socks_5`, or `http`.

<port>

The port number to communicate with the proxy server. The correct port number depends on the port opened up on the proxy server for communication with CloudPools.

{--username | -u} *<string>*

The user name to authenticate with the SOCKS v5 or HTTP proxy server. Note that SOCKS v4 does not support authentication.

{--password | -p} *<string>*

The password to authenticate with the SOCKS v5 or HTTP proxy server.

{--verbose | -v}

Displays more detailed information.

Examples

The following example creates a network proxy to use with CloudPools:

```
isi cloud proxies create myproxy1 myprox1.example.com socks_5 1080
--username mycloudpools --password dhgXJ9OAIahXvYmL
```

isi cloud proxies delete

Deletes a network proxy in CloudPools. Note that CloudPools prevents deletion of a proxy that is attached to a cloud storage account.

Syntax

```
isi cloud proxies delete <name>
[--force]
[--verbose]
```

Options

<name>

The unique id or name of the network proxy. You can use the `isi cloud proxies list` command to display the names of proxies.

{--force | -f}

Enables the proxy deletion to proceed without confirmation.

{--verbose | -v}

Displays more detailed information.

Example

The following example deletes a network proxy named `myproxy1`:

```
isi cloud accounts delete myproxy1
```

When you run the command, OneFS displays the following message and requires confirmation:

```
Are you sure? (yes/[no]):
```

To proceed, type **yes**, and press ENTER. If the proxy is attached to a cloud storage account, OneFS displays the following message:

```
Cannot delete proxy while used by accounts
```

isi cloud proxies list

Displays a list of network proxies created in CloudPools.

Syntax

```
isi cloud proxies list
 [--limit <integer>]
 [--sort {id | name | host | type | port}]
 [--descending]
 [--format {table | json | csv | list}]
 [--no-header]
 [--no-footer]
 [--verbose]
```

Options

{--limit | -l} *<integer>*

>   Limits the number of network proxies displayed in the list.

--sort {id | name | host | type | port}

>   Sorts the list of cloud proxies according to the specified category.

--format {table | json | csv | list}

>   Displays output in table (default), JavaScript Object Notation (JSON), comma-separated value (CSV), or list format.

{--descending | -d}

>   Outputs the list of network proxies in descending order according to the specified sort option.

{--no-header | -a}

>   Displays table and CSV output without headers.

{--no-footer | -z}

>   Displays table output without footers.

{--verbose | -v}

>   Displays more detailed information.

Example

The following example creates a network proxy to use with CloudPools:

```
isi cloud proxies create myproxy1 myprox1.example.com socks_5 1080
--username mycloudpools --password dhgXJ9OAIahXvYmL
```

isi cloud proxies modify

Modifies the properties of a network proxy.

Syntax

```
isi cloud proxies modify <name>
 [--name <string>]
 [--host <string>]
 [--type {socks_4 | socks_5 | http}]
 [--port <integer>]
 [--username <string>]
 [--clear-username]
 [--password <string>]
 [--clear-password]
 [--verbose]
```

Options

<name>

> The current name of the network proxy.

{--name | -n} <string>

> The new name of the network proxy. This can be any alphanumeric string, but should be a simple, recognizable name.

--host <string>

> The DNS name or IP address of the proxy server. For example, `myproxy1.example.com` or `192.168.107.107`.

{--type | -t} {socks_4 | socks_5 | http}

> The network proxy protocol , one of `socks_4`, `socks_5`, or `http`.

--port <integer>

> The port number to communicate with the proxy server. The correct port number depends on the port opened up on the proxy server for communication with CloudPools.

{--username | -u} <string>

> The user name to authenticate with the SOCKS v5 or HTTP proxy server. Note that SOCKS v4 does not support authentication.

--clear-username

> Clear the user name that was previously specified for proxy server authentication.

{--password | -p} <string>

> The password to authenticate with the SOCKS v5 or HTTP proxy server.

--clear-password

> Clear the password that was previously specified for proxy server authentication.

{--verbose | -v}

> Displays more detailed information.

Examples

The following example modifies a network proxy in CloudPools:

```
isi cloud proxies modify myproxy1 --type socks_4 --clear-username --clear-password
```

isi cloud proxies view

View the details of a network proxy created for CloudPools.

Syntax

```
isi cloud proxies view <name>
```

Options

<name>

>Specifies the id or name of the network proxy to view. You can use the `isi cloud proxies list` command to display a list of the available proxies.

Example

The following example displays the details of a network proxy named `myproxy1`:

```
isi cloud proxies view myproxy1
```

isi cloud recall

Recalls files from the cloud, restoring the full files to their original directories. To make sure that the specified files are present in the cloud, OneFS scans the cluster for SmartLink files prior to performing the recall.

Syntax

```
isi cloud recall <files>
  [--recursive | -r {yes | no}]
  [--verbose]
```

You can also provide a file matching filter to specify a set of files to act on. The basic syntax follows. For a full description on file matching criteria, use `man isi-file-matching` on the command line.

Options

<files>

>The files to recall. For multiple specifications, use `--files` for each additional file name entry.

[--begin-filter] *<criteria>* [--end-filter]

>A file matching filter that defines a set of files to act on. For a description of *<criteria>* and valid operators to use in the filter, enter `man isi-file-matching` on the command line.

{--recursive | -r} {yes | no}

>Specifies whether the recall should apply recursively to nested subdirectories.

{--verbose | -v}

>Displays more detailed information about the operation.

Usage

The `isi cloud recall` command restores the full file to its original directory, and overwrites the associated SmartLink file. If the file pool policy that originally archived the file to the cloud is still in effect, the next time the SmartPools job runs, the recalled file is archived to the cloud again. If you do not want the recalled file to be re-archived, you can move the file to a different directory that would not be affected by the file pool policy, or you can modify or delete the policy.

Examples

The following example recalls all files from the cloud for a directory and its subdirectories:

```
isi cloud recall /ifs/data/archives/archives2014/projects/ --recursive yes
```

The command starts a cloud job. If you use the `--verbose` parameter, OneFS reports the job number, as in the following example:

```
Created job [29]
```

You can use the `isi cloud jobs view` command with the job number to see information about the job.

isi cloud restore_coi

Restores the cloud object index (COI) for a cloud storage account on the cluster. The `isi cloud access add` command also restores the COI for a cloud storage account.

Usage

⚠️ **CAUTION: Do not execute this command unless instructed to do so by Dell Technologies Technical Support.**

A cloud object index (COI) is a persistent mapping between cloud objects, their retention periods, and optionally, the files that use the cloud objects. The cluster uses the COI when performing cleanup (garbage collection), to ensure it considers all versions of files and objects correctly.

The `isi cloud restore coi` command allows a cluster to complete a COI to include all versions of all objects. The command might be used in the following situations:
- To handle COI corruption in cases where COI entries are corrupted or deleted. This command can restore the COI for a specified cloud account.
- To increase the retention time on the cluster where the command is run for objects in the specified cloud account.

Syntax

```
isi cloud restore_coi
  [--accounts <string>]
  [--expiration-date <timestamp>]
  [--verbose]
```

Options

`--accounts <string>...`

> The name of the cloud storage account whose COI you intend to restore. By restoring the COI, you enable OneFS to not only read data from the cloud, but also to write data to the cloud.
>
> Use an additional `--accounts` parameter for each additional cloud account.

`--expiration-date <timestamp>`

> The expiration date for orphaned cloud data objects.

{--verbose | -v}

> Displays more detailed information about the operation.

Example

The following example restores the COI for a cloud storage account:

```
isi cloud restore_coi --account my_azure_acct
```

isi cloud settings modify

Controls settings for Smartlinks across various workflows.

Use isi cloud settings view to see the current settings.

Syntax

```
isi cloud settings modify
  [--default-accessibility {cached | no-cache}]
  [--default-cache-expiration <duration>]
  [--default-compression-enabled {yes | no}]
  [--default-data-retention <duration>]
  [--default-encryption-enabled {yes | no}]
  [--default-full-backup-retention <duration>]
  [--default-incremental-backup-retention <duration>]
  [--default-read-ahead <string>]
  [--default-writeback-frequency <duration>]
  [--verbose]
```

Options

These options specify the default behavior for accessing a SmartLink (archived) file, if there is no file pool policy for the archived file. If there is a file pool policy for the archived file, that policy and its settings take precedence over any of these settings.

--default-accessibility {cached | no-cache}

> Specifies `cache` to ensure that only the data blocks requested are cached and flushed or invalidated based on the cache retention settings. Specify `no-cache` to ensure that the data blocks are not cached and do not consume file system blocks long term.

--default-cache-expiration *<duration>*

> The minimum amount of time until the cache expires. A number followed by a unit of time is accepted. For example, a setting of 9H would specify a nine-hour duration. Similarly, a setting of 2D would specify a two-day duration.

--default-compression-enabled {yes | no}

> Specifies whether data is to be compressed when archived to the cloud.

--default-data-retention *<duration>*

> The minimum amount of time that cloud objects associated with a SmartLink file will be retained in the cloud after the SmartLink file is deleted from the cluster. A number followed by a unit of time is accepted. For example, a setting of 9H would specify a nine-hour duration. Similarly, a setting of 2D would specify a two-day duration.

--default-encryption-enabled {yes | no}

> Specifies whether data is to be encrypted when archived to the cloud.

--default-full-backup-retention *<duration>*

> The length of time that OneFS retains cloud data referenced by a SmartLink file that has been backed up by a full NDMP backup and is subsequently deleted. A number followed by a unit of time is accepted.

For example, a setting of 9H would specify a nine-hour duration. Similarly, a setting of 2D would specify a two-day duration.

--default-incremental-backup-retention *<duration>*

The length of time that OneFS retains cloud data referenced by a SmartLink file that has been backed up by an incremental NDMP backup, or replicated by a SyncIQ operation, and is subsequently deleted. A number followed by a unit of time is accepted. For example, a setting of 5Y would specify a five-year duration.

--default-read-ahead {partial | full}

Specifies the cache readahead strategy when SmartLink files are accessed. A partial strategy means that only the amount of data needed by the user is cached. A full strategy means that all file data will be cached when the user accesses a SmartLink file.

--default-writeback-frequency *<duration>*

The minimum amount of time to wait before OneFS updates cloud data with local changes. A number followed by a unit of time is accepted. For example, a setting of 9H would specify a nine-hour duration. Similarly, a setting of 2D would specify a two-day duration.

{--verbose | -v}

Displays more information about the operation.

Example

The following example modifies several of the default CloudPools settings:

```
isi cloud settings modify --default-writeback-frequency 12H
--default-cache-expiration 9H --default-accessability no-cache
--default-encryption-enabled yes
```

isi cloud settings regenerate-encryption-key

Generates a new master encryption key for new data that will be archived to the cloud. Previously encrypted archived data continues to require previously generated encryption keys. All previous encryption keys are preserved for use with the existing archived data.

Syntax

```
isi cloud settings regenerate-encryption-key
[--verbose]
```

Option

{--verbose | -v}

Displays more detailed information.

isi cloud settings view

Displays the current default settings in CloudPools. You can use the `isi cloud settings modify` command to change default settings.

Syntax

```
isi cloud settings view
```

Options

There are no options for this command.

Example

The following example shows sample output. Explanations of the displayed properties are included in the descriptions for isi cloud settings modify.

```
B248930-PSL-1# isi cloud settings view
                Default Accessibility: cached
              Default Cache Expiration: 1D
          Default Compression Enabled: No
                Default Data Retention: 1W
            Default Encryption Enabled: No
        Default Full Backup Retention: 5Y
Default Incremental Backup Retention: 5Y
                    Default Read Ahead: partial
            Default Writeback Frequency: 9H
```

isi filepool policies create

Create a custom file pool policy to identify a specific storage target and perform other actions on matched files and directories.

Syntax

```
isi filepool policies create <name>
 [--description <string>]
 [--begin-filter{<predicate> <operator> <link>}...--end-filter]
 [--apply-order <integer>]
 [--data-access-pattern  {random | concurrency | streaming}]
 [--set-requested-protection {default | +1 | +2:1 | +2 | +3:1 | +3 | +4 | 2x | 3x | 4x |
5x | 6x | 7x | 8x}]
 [--data-storage-target <string>]
 [--data-ssd-strategy  {metadata | metadata-write | data | avoid}]
 [--snapshot-storage-target <string>]
 [--snapshot-ssd-strategy {metadata | metadata-write | data | avoid}]
 [--enable-coalescer {Yes | No}]
 [--enable-packing {Yes | No}]
 [--cloud-pool <string>]
 [--cloud-accessibility {cached | no-cache}]
 [--cloud-cache-expiration <duration>]
 [--cloud-compression-enabled {yes | no}]
 [--cloud-data-retention <duration>]
 [--cloud-encryption-enabled {yes | no}]
 [--cloud-full-backup-retention <duration>]
 [--cloud-incremental-backup-retention <duration>]
 [--cloud-read-ahead <string>]
 [--cloud-writeback-frequency <duration>]
 [--verbose | -v]
 [--help | -h]
```

Options

<name>

Specifies the name of the file pool policy to create.

--begin-filter {*<predicate>* *<operator>* *<link>*}… --end-filter

Specifies the file-matching criteria that determine the files to be managed by the filepool policy. Each file matching criterion consists of three parts:

- Predicate. Specifies what attribute(s) to filter on. You can filter by path, name, file type, timestamp, or custom attribute, or use a combination of these attributes.
- Operator. Qualifies an attribute (for example, birth time) to describe a relationship to that attribute (for example, before).
- Link - Combines attributes using AND and OR statements.

The following predicates are valid:

--size=<*nn*>[{B | KB | MB | GB | TB | PB}]

> Selects files according to the specified size.

--path=<*pathname*>

> Selects files relative to the specified pathname.

--file-type= <*value*>

> Selects only the specified file-system object type.
>
> The following values are valid:

file	Specifies regular files.
directory	Specifies directories.
other	Specifies links.

--name= <*value*> [--case-sensitive= {true | false}]

> Selects only files whose names match the specified string. Use `--case-sensitive=true` to enable case-sensitivity.
>
> When forming the name, you can include the following wildcards:
>
> - *
> - [ ]
> - ?

--birth-time=<*timestamp*>

> Selects files that were created relative to the specified date and time. Timestamp arguments are formed as **YYYY-MM-DDTHH:MM:SS**. For example, **2013-09-01T08:00:00** specifies a timestamp of September 1, 2013 at 8:00 A.M. You can use `--operator=` with an argument of `gt` to mean after the timestamp or `lt` to mean before the timestamp.

--changed-time=<*timestamp*>

> Selects files that were modified relative to the specified date and time.

--metadata-changed-time=<*timestamp*>

> Selects files whose metadata was modified relative to the specified date and time.

--accessed-time=<*timestamp*>

> Selects files that were accessed at the specified time interval.

--custom-attribute=<*value*>

> Selects files based on a custom attribute.

You can use the `operator=` option to specify a qualifier for the file-matching criterion. Specify operators in the following form:

```
--operator=<value>
```

The following operator values are valid:

Table 8. --operator=<value>

Value	Description
eq	Equal. This is the default value.
ne	Not equal
lt	Less than
le	Less than or equal to
gt	Greater than
ge	Greater than or equal to
not	Not

Link arguments can be used to specify multiple file-matching criteria. The following links are valid:

--and

> Connects two file-matching criteria where files must match both criteria.

--or

> Connects two file-matching criteria where files must match one or the other criteria.

--description *<string>*

> Specifies a description of the filepool policy

--apply-order *<integer>*

> Specifies the order index for execution of this policy.

--data-access-pattern *<string>*

> Data access pattern random, streaming or concurrent.

--set-requested-protection *<string>*

> Specifies a protection level for files that match this filepool policy (e.g., +3, +2:3, 8x).

--data-storage-target *<string>*

> The name of the node pool or tier to which the policy moves files on the local cluster. If you do not specify a data storage target, the default is **anywhere**.

--data-ssd-strategy *<string>*

> Specifies how to use SSDs to store local data.

avoid	Writes all associated file data and metadata to HDDs only.
metadata	Writes both file data and metadata to HDDs. This is the default setting. An extra mirror of the file metadata is written to SSDs, if SSDs are available. The SSD mirror is in addition to the number required to satisfy the requested protection. Enabling GNA makes read acceleration available to files in node pools that do not contain SSDs.
metadata-write	Writes file data to HDDs and metadata to SSDs, when available. This strategy accelerates metadata writes in addition to reads but requires about four to five times more SSD storage than the **Metadata** setting. Enabling GNA does not affect read/write acceleration.
data	Uses SSD node pools for both data and metadata, regardless of whether global namespace acceleration is enabled. This SSD strategy does not result in the creation of additional mirrors beyond the normal requested protection but requires significantly increases storage requirements compared with the other SSD strategy options.

--snapshot-storage-target *<string>*

The name of the node pool or tier chosen for storage of snapshots. If you do not specify a snapshot storage target, the default is **anywhere**.

--snapshot-ssd-strategy *<string>*

Specifies how to use SSDs to store snapshots. Valid options are `metadata`, `metadata-write`, `data`, `avoid`. The default is `metadata`.

--enable-coalescer {Yes | No}

Enables the coalescer.

--enable-packing {Yes | No}

Enables packing.

--cloud-pool *<string>*

Specifies the default CloudPool and, therefore, the cloud storage account where cloud data is to be archived.

--cloud-accessibility {cached | no-cache}

Specifies the default behavior for accessing a SmartLink (archived) file if there is no file pool policy for the archived file. If there is a file pool policy for the archived file, that policy and its settings take precedence over a `--cloud-accessibility` setting. Specify `cache` to ensure that only the data blocks requested are cached and flushed or invalidated based on the cache retention settings. Specify `no-cache` to ensure that the data blocks are not cached and do not consume file system blocks long term.

--cloud-cache-expiration *<duration>*

Specifies the minimum amount of time until the cache expires. A number followed by a unit of time is accepted. For example, a setting of 9H would specify a nine-hour duration. Similarly, a setting of 2D would specify a two-day duration.

--cloud-compression-enabled {yes | no}

Specifies whether data is to be compressed when archived to the cloud.

--cloud-data-retention *<duration>*

Specifies the minimum amount of time that archived data will be retained in the cloud after a SmartLink file is deleted from the cluster. A number followed by a unit of time is accepted. For example, a setting of 9H would specify a nine-hour duration. Similarly, a setting of 2D would specify a two-day duration.

--cloud-encryption-enabled {yes | no}

Specifies whether data is to be encrypted when archived to the cloud.

--cloud-full-backup-retention *<duration>*

Specifies the minimum amount of time that cloud files will be retained after the creation of a full backup. A number followed by a unit of time is accepted. For example, a setting of 9H would specify a nine-hour duration. Similarly, a setting of 2D would specify a two-day duration.

--cloud-incremental-backup-retention *<duration>*

Specifies the minimum amount of time that cloud files will be retained after the creation of an incremental backup. A number followed by a unit of time is accepted. For example, a setting of 9H would specify a nine-hour duration. Similarly, a setting of 2D would specify a two-day duration.

--cloud-read-ahead {partial | full}

Specifies the cache readahead strategy when SmartLink files are accessed. A partial strategy means that only the amount of data needed by the user is cached. A full strategy means that all file data will be cached when the user accesses a SmartLink file.

--cloud-writeback-frequency *<duration>*

Specifies the minimum amount of time to wait before OneFS updates cloud data with local changes. A number followed by a unit of time is accepted. For example, a setting of 9H would specify a nine-hour duration. Similarly, a setting of 2D would specify a two-day duration.

--verbose

Displays more detailed information.

--help | -h

Display help for this command.

Examples

The following example creates a file pool policy that moves all files in directory `/ifs/data/chemical/arco/finance` to the local storage target named `Archive_2`.

```
 isi filepool policies create Save_Fin_Data --begin-filter
--path=/ifs/data/chemical/arco/finance --end-filter
--data-storage-target Archive_2 --data-ssd-strategy=metadata
```

The following example matches older files that have not been accessed or modified later than specified dates, and moves the files to an archival tier of storage.

```
isi filepool policies create archive_old
--data-storage-target ARCHIVE_1 --data-ssd-strategy avoid
--begin-filter --file-type=file --and --birth-time=2013-09-01
--operator=lt --and --accessed-time=2013-12-01 --operator=lt
--and --changed-time=2013-12-01 --operator=lt --end-filter
```

isi statistics cloud list

Displays a list of CloudPools statistics.

Syntax

```
isi statistics cloud list
        [--account <str>]
        [--policy <str>]
        [{--nodes | -n} <NODE>]
        [{--degraded | -d}]
        [--nohumanize]
        [{--intervalacct | -i} <float>]
        [{--repeat | -r} <integer>]
        [{--limit | -l} <integer>]
        [--long]
        [--output(Timestamp|time) | (Account|) | (Policy|pol) |
          (Cluster-GUID|guid|cluster) | In | Out | Reads | Writes |
          (Deletions|deletes|del) | (Cloud|vendor) | (A-Key|account_key|key) |
          (P-ID|policy_id|id) | Node)]
        [--sort ((Timestamp|time) | (Account|acct) | (Policy|pol) |
          (Cluster-GUID|guid|cluster) | In | Out | Reads | Writes |
          (Deletions|deletes|del) | (Cloud|vendor) | (A-Key|account_key|key) |
          (P-ID|policy_id|id) | Node)]
        [--format (table | json | csv | list | top)]
        [{--no-header | -a}]
        [{--no-footer | -z}]
        [{--verbose | -v}]
```

Options

[--account <str>]

> Identifies the account to view. Specifies the account name or a phrase to match. Default is 'all' which will select all accounts.

[--policy <str>]

> Identifies the policy to view. Specifies the policy name or a phrase to match. Default is 'none' which will select no policies.

[{--nodes | -n} <NODE>]

> Specifies node(s) for which statistics should be reported.

[{--degraded | -d}]

Continues to report if some nodes do not respond.

`--nohumanize`

Outputs raw numbers without conversion to units.

`[{--interval`*acct*` | -i} <`*float*`>]`

Waits <INTERVAL> seconds before refreshing the display.

`[{--repeat | -r} <`*integer*`>]`

Prints the requested data <REPEAT> times (-1 for infinite).

`[{--limit | -l} <`*integer*`>]`

Displays the number of statistics clouds to display.

`--long`

Displays all possible columns.

`[--output (Timestamp|time)|(Account|)|(Policy|pol)|(Cluster-GUID|guid|cluster)|In|Out | Reads|Writes|(Deletions|deletes|del)|(Cloud|vendor)|(A-Key|account_key|key)|(P-ID| policy_id|id)|Node)]`

Displays output specified column(s).

`[--sort (Timestamp|time)|(Account|acct)|(Policy|pol)|(Cluster-GUID|guid|cluster)|In|Out | Reads|Writes|(Deletions|deletes|del)|(Cloud|vendor)|(A-Key|account_key|key)|(P-ID| policy_id|id)|Node)]`

Sorts data by the specified comma-separated field(s). Prepend 'asc:' or 'desc:' to a field to change the sort order.

`[--format (table | json | csv | list | top)]`

Displays statistics clouds in table, JSON, CSV, list or top format.

`{--noheader | -a}`

Displays data without column headings.

`{ --no-footer | -z}`

Displays data without footers.

`{--verbose | -v}`

Displays more detailed information.