PowerScale OneFS

HDFS Reference Guide

OneFS 8.1.2.0 - 9.1.0.0

October 2020

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.**

1

Introduction to this guide

This guide provides information for PowerScale OneFS and Hadoop Distributed File System (HDFS) administrators when implementing a PowerScale OneFS and Hadoop system integration. This guide describes how you can use the PowerScale OneFS Web administration interface (Web UI) and command-line interface (CLI) to configure and manage your PowerScale and Hadoop clusters.

Topics:

Where to go for support

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

Dell Technologies support	● Support tab on the Dell homepage: https://www.dell.com/support/incidents-online. Once you identify your product, the "How to Contact Us" gives you the option of email, chat, or telephone support. ● For questions about accessing online support, send an email to support@emc.com.
Telephone support	● United States: 1-800-SVC-4EMC (1-800-782-4362) ● Canada: 1-800-543-4782 ● Worldwide: 1-508-497-7901 ● Local phone numbers for a specific country/region are available at Dell EMC Customer Support Centers.
PowerScale OneFS Documentation Info Hubs	● OneFS Info Hubs: https://www.dell.com/support/article/sln318794

Copyright

Published October 2020

Dell believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED “AS-IS.“ DELL MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND

WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF

MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. USE, COPYING, AND DISTRIBUTION OF ANY DELL SOFTWARE DESCRIBED

IN THIS PUBLICATION REQUIRES AN APPLICABLE SOFTWARE LICENSE.

Published in the USA.

EMC Corporation

Hopkinton, Massachusetts 01748-9103

1-508-435-1000. In North America 1-866-464-7381

www.EMC.com

2

Overview of HDFS with OneFS

This chapter provides information about how the Hadoop Distributed File System (HDFS) can be implemented with PowerScale OneFS.

Topics:

How Hadoop is implemented on OneFS

In a Hadoop implementation on a PowerScale cluster, PowerScale OneFS serves as the file system for Hadoop compute clients. The Hadoop distributed file system (HDFS) is supported as a protocol, which is used by Hadoop compute clients to access data on the HDFS storage layer.

Hadoop compute clients can access the data that is stored on a PowerScale cluster by connecting to any node over the HDFS protocol, and all nodes that are configured for HDFS provide NameNode and DataNode functionality as shown in the following illustration.

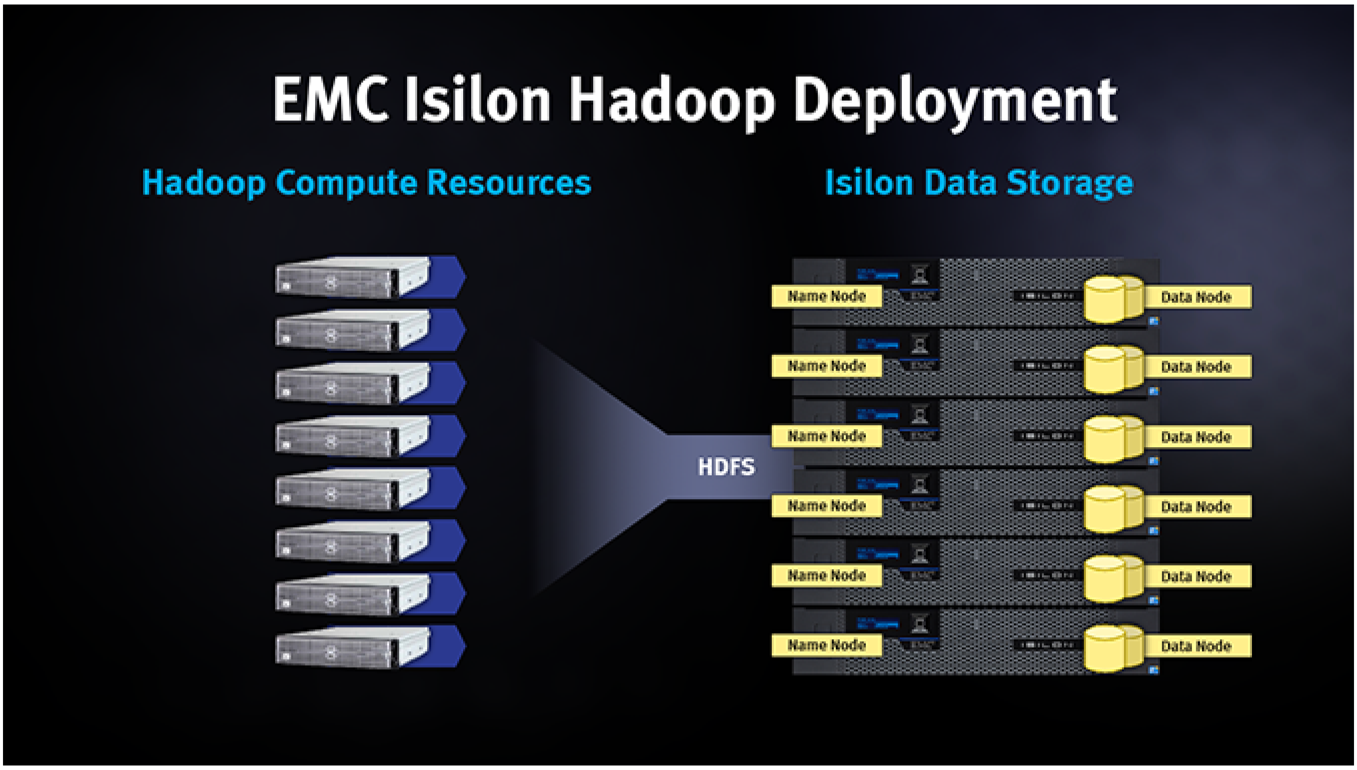

Figure 1. PowerScale OneFS Hadoop Deployment

Each node boosts performance and expands the cluster's capacity. For Hadoop analytics, the PowerScale scale-out distributed architecture minimizes bottlenecks, rapidly serves Big Data, and optimizes performance.

How a PowerScale OneFS Hadoop implementation differs from a traditional Hadoop deployment

A Hadoop implementation with OneFS differs from a typical Hadoop implementation in the following ways:

- The Hadoop compute and HDFS storage layers are on separate clusters instead of the same cluster.
- Instead of storing data within a Hadoop distributed file system, the storage layer functionality is fulfilled by OneFS on a PowerScale cluster. Nodes on the PowerScale cluster function as both a NameNode and a DataNode.
- The compute layer is established on a Hadoop compute cluster that is separate from the PowerScale cluster. The Hadoop MapReduce framework and its components are installed on the Hadoop compute cluster only.
- Instead of a storage layer, HDFS is implemented on OneFS as a native, lightweight protocol layer between the PowerScale cluster and the Hadoop compute cluster. Clients from the Hadoop compute cluster connect over HDFS to access data on the PowerScale cluster.
- In addition to HDFS, clients from the Hadoop compute cluster can connect to the PowerScale cluster over any protocol that OneFS supports such as NFS, SMB, FTP, and HTTP. PowerScale OneFS is the only non-standard implementation of HDFS offered that allows for multi-protocol access. PowerScale makes for an ideal alternative storage system to native HDFS by marrying HDFS services with enterprise-grade data management features.
- Hadoop compute clients can connect to any node on the PowerScale cluster that functions as a NameNode instead of being routed by a single NameNode.

Hadoop distributions supported by OneFS

You can run most common Hadoop distributions with the PowerScale cluster.

OneFS supports many distributions of the Hadoop Distributed File System (HDFS). These distributions are updated independently of OneFS and on their own schedules.

For the latest information about Hadoop distributions that OneFS supports, see the Hadoop Distributions and Products Supported by OneFS page.

HDFS files and directories

You must configure one HDFS root directory in each OneFS access zone that will contain data accessible to Hadoop compute clients. When a Hadoop compute client connects to the cluster, the user can access all files and sub-directories in the specified root directory. The default HDFS directory is `/ifs`.

Note the following:

- Associate each IP address pool on the cluster with an access zone. When Hadoop compute clients connect to the PowerScale cluster through a particular IP address pool, the clients can access only the HDFS data in the associated access zone. This configuration isolates data within access zones and allows you to restrict client access to the data.
- Unlike NFS mounts or SMB shares, clients connecting to the cluster through HDFS cannot be given access to individual folders within the root directory. If you have multiple Hadoop workflows that require separate sets of data, you can create multiple access zones and configure a unique HDFS root directory for each zone.
- When you set up directories and files under the root directory, make sure that they have the correct permissions so that Hadoop clients and applications can access them. Directories and permissions will vary by Hadoop distribution, environment, requirements, and security policies.

For more information about access zones, refer to the OneFS CLI Administration Guide or OneFS Web Administration Guide for your version of OneFS.

Hadoop user and group accounts

Before implementing Hadoop, ensure that the user and groups accounts that you will need to connect over HDFS are configured on the PowerScale cluster.

Additionally, ensure that the user accounts that your Hadoop distribution requires are configured on the PowerScale cluster on a per-zone basis. The user accounts that you need and the associated owner and group settings vary by distribution, requirements, and security policies. The profiles of the accounts, including UIDs and GIDS, on the PowerScale cluster should match the profiles of the accounts on your Hadoop compute clients.

OneFS must be able to look up a local Hadoop user or group by name. If there are no directory services, such as Active Directory or LDAP, that can perform a user lookup, you must create a local Hadoop user or group. If directory services are available, a local user account or user group is not required.

HDFS and SmartConnect

You can configure a SmartConnect DNS zone to manage connections from Hadoop compute clients.

SmartConnect is a module that specifies how the DNS server on a PowerScale cluster handles connection requests from clients. For each IP address pool on the PowerScale cluster, you can configure a SmartConnect DNS zone which is a fully qualified domain name (FQDN).

For more information about SmartConnect, refer to the OneFS CLI Administration Guide or OneFS Web Administration Guide for your version of OneFS.

Note the following:

- Hadoop compute clients can connect to the cluster through the SmartConnect DNS zone name, and SmartConnect evenly distributes NameNode requests across IP addresses and nodes in the pool.
- When a Hadoop compute client makes an initial DNS request to connect to the SmartConnect zone, the Hadoop client is routed to the IP address of an PowerScale node that serves as a NameNode. Subsequent requests from the Hadoop compute client go to the same node. When a second Hadoop client makes a DNS request for the SmartConnect zone, SmartConnect balances traffic and routes the client connection to a different node than that used by the previous Hadoop compute client.
- If you specify a SmartConnect DNS zone that you want Hadoop compute clients to connect through, you must add a Name Server (NS) record as a delegated domain to the authoritative DNS zone that contains the PowerScale cluster.
- On the Hadoop compute cluster, you must set the value of the `fs.defaultFS` property to the SmartConnect DNS zone name in the `core-site.xml` file.

3

Configuring OneFS with HDFS

The following sections are steps you need perform to configure OneFS with HDFS.

Topics:

Activate the HDFS and SmartConnect Advanced licenses

Before you can use OneFS with HDFS, you must confirm that licenses for HDFS and SmartConnect Advanced are active.

1. To confirm that HDFS and SmartConnect Advanced are installed, run the following commands:

```
isi license list
isi license view HDFS
isi license view SMARTCONNECT_ADVANCED
```

2. If your modules are not licensed, obtain a license key from your PowerScale sales representative. To activate the license, type the following command, where `license file path` is the location of your license file:

```
isi license add --path <license file path>
```

Configuring the HDFS service

You can configure HDFS service settings on your OneFS cluster to improve performance for HDFS workflows.

HDFS service settings overview

HDFS service settings affect the performance of HDFS workflows.

You can configure the following HDFS service settings:

Setting	Description
Block size	The HDFS block size setting on the PowerScale cluster determines how the HDFS service returns data on read requests from Hadoop compute client. You can modify the HDFS block size on the cluster to increase the block size from 4 KB up to 1 G. The default block size is 128 MB. Increasing the block size enables the PowerScale cluster nodes to read and write HDFS data in larger blocks and optimize performance for most use cases.

Setting	Description
	The Hadoop cluster maintains a different block size that determines how a Hadoop compute client writes a block of file data to the PowerScale cluster. The optimal block size depends on your data, how you process your data, and other factors. You can configure the block size on the Hadoop cluster in the `hdfs-site.xml` configuration file in the dfs.block.size property.
Checksum type	The HDFS service sends the checksum type to Hadoop compute clients, but it does not send any checksum data, regardless of the checksum type. The default checksum type is set to `None`. If your Hadoop distribution requires sending a checksum type other than `None` to the client, you can set the checksum type to `CRC32` or `CRC32C`.

Enable or disable the HDFS service globally (Web UI)

Enable or disable the HDFS service globally using the OneFS web administration interface (Web UI).

1. Click **Protocols** > **Hadoop (HDFS) > Settings**.
2. Check or uncheck the **Enable HDFS service** checkbox at the top of the page.
3. Select **Enable** or **Disable** on the **Confirm Action** window.

Enable or disable the HDFS service per-access zone (Web UI)

Enable or disable the HDFS service on a per-access zone using the OneFS web administration interface (Web UI).

1. Click **Protocols** > **Hadoop (HDFS) > Settings**.
2. From the **Current Access Zone** list, select the access zone that you want to enable or disable the HDFS service for.
3. From the **HDFS service settings** area, select or clear the **Enable HDFS service for zone <zone>** check box.
4. Click **Save Changes**.

Enable or disable the HDFS service globally (CLI)

Enable or disable the HDFS service globally using the OneFS command-line interface (CLI).

Run the `isi services` command.
The following command enables the HDFS service globally:

```
isi services hdfs enable
```

The following command disables the HDFS service globally:

```
isi services hdfs disable
```

Enable or disable the HDFS service per-access zone (CLI)

Enable or disable the HDFS service on a per-access zone basis using the OneFS command-line interface (CLI).

Run the `isi hdfs settings modify` command.
The following command enables the HDFS service in zone3:

```
isi hdfs settings modify --service=yes --zone=zone3
```

The following command disables the HDFS service in zone3:

```
isi hdfs settings modify --service=no --zone=zone3
```

Configure HDFS service settings (Web UI)

Configure HDFS service settings in each access zone using the OneFS web administration interface.

1. Click **Protocols** > **Hadoop (HDFS)** > **Settings**.
2. From the **Current Access Zone** list, select the access zone in which you want to configure service settings.
3. From the **HDFS Service Settings** area, select the HDFS block size you want from the **Default Block Size** list.

   The HDFS block size determines how the HDFS service returns data upon read requests from Hadoop compute client.
4. Select the checksum type from the **Default Checksum Type** list.

   The HDFS service does not send any checksum data, regardless of the checksum type.
5. Click **Save Changes**.

Configure HDFS service settings (CLI)

Configure HDFS service settings in each access zone using the OneFS command-line interface.

Run the `isi hdfs settings modify` command.
The following command sets the block size to 256 KB in the zone3 access zone:

```
isi hdfs settings modify --default-block-size=256K --zone=zone3
```

You must specify the block size in bytes. Suffixes K, M, and G are allowed.

The following command sets the checksum type to crc32 in the zone3 access zone:

```
isi hdfs settings modify --default-checksum-type=crc32 --zone=zone3
```

View HDFS settings (Web UI)

View the HDFS settings for an access zone using the OneFS web administration interface.

1. Click **Protocols** > **Hadoop (HDFS)** > **Settings**.
2. From the **Current Access Zone** list, select the access zone that you want to view the HDFS settings for.

   The **Settings** tab displays the current HDFS options in the following areas:

   - **HDFS Service Settings**
   - **HDFS Protocol Settings**
   - **Ambari Server Settings**

View HDFS settings (CLI)

View the HDFS settings for an access zone using the command-line interface.

1. Open a secure shell (SSH) connection to any node in the cluster and then log in.
2. Run the `isi hdfs settings view` command.
   The following command displays the HDFS settings in the zone1 access zone:

   ```
   isi hdfs settings view --zone=zone1
   ```

Modify HDFS log levels (CLI)

You can set the default logging level of HDFS service events for any node on the PowerScale cluster.

This procedure is available only through the command-line interface.

1. Open a secure shell (SSH) connection to a node in the cluster and log in.
2. Run the `isi hdfs log-level modify` command.

The following command sets the HDFS log level to trace on the node:

```
isi hdfs log-level modify --set=trace
```

View HDFS log levels (CLI)

You can view the default logging level of HDFS services events for any node in the PowerScale cluster.

This procedure is available only through the OneFS command-line interface.

1. Open a secure shell (SSH) connection to a node in the cluster and log in.
2. Run the `isi hdfs log-level view` command.

Set the HDFS root directory (Web UI)

Configure one HDFS root directory in each access zone using the OneFS web administration interface.

1. Click **Protocols** > **Hadoop (HDFS)** > **Settings**.
2. From the **Current Access Zone** list, select the access zone for which you want to specify the root directory.
3. From the **HDFS Protocol Settings** area, in the **Root Directory** field, type or browse to directory that you want to use for the HDFS root directory.

   The root directory must be within `/ifs`.
4. Click **Save Changes**.

Set the HDFS root directory (CLI)

Configure one HDFS root directory in each access zone using the command-line interface.

The directory structure that you want to set as the root path must exist first on the OneFS file system.

- Run the `isi hdfs settings modify` command.
  The following command specifies that Hadoop compute clients connecting to the zone3 access zone are provided access to the `/ifs/data/hadoop` directory:

```
isi hdfs settings modify --root-directory=/ifs/zone3/hadoop --zone=zone3
```

Configuring HDFS authentication methods

You can configure an HDFS authentication method on a per-access zone basis.

When a Hadoop compute client connects to the PowerScale cluster through an access zone, the client must authenticate with the method that is specified for that access zone.

ⓘ **NOTE:** If you want Hadoop compute clients running Hadoop 2.2 and later to connect to an access zone through Kerberos, you must configure HDFS authentication properties on the Hadoop client.

Supported HDFS authentication methods

The authentication method determines the credentials that OneFS requires to establish a Hadoop compute client connection.

An HDFS authentication method is specified for each access zone. OneFS supports the following authentication methods for HDFS:

Authentication method	Description
Simple only	Requires only a username to establish client connections.
Kerberos only	Requires Kerberos credentials to establish client connections.

Authentication method	Description
	ⓘ **NOTE:** You must configure Kerberos as an authentication provider on the PowerScale cluster, and you must modify the `core-site.xml` file on clients running Hadoop 2.2 and later.
All (default value)	Accepts both simple authentication and Kerberos credentials. If Kerberos settings and file modifications are not completed, client connections default to simple authentication. ⚠ **CAUTION: To prevent unintended access through simple authentication, set the authentication method to `Kerberos only` to enforce client access through Kerberos.**

Set the HDFS authentication method (Web UI)

Configure the HDFS authentication method in each access zone using the OneFS web administration interface.

If you want Hadoop clients to connect to an access zone through Kerberos, a Kerberos authentication provider must be configured and added to the access zone.

1. Click **Protocols** > **Hadoop (HDFS)** > **Settings**.
2. In the **Current Access Zone** list, select the access zone that you want to specify the authentication method for.
3. In the **HDFS Protocol Settings** area, in the **Authentication Type** list, select one of the following authentication methods:
   - Both Simple and Kerberos authentication (OneFS 8.1.2 - 8.2.0 only)
   - Simple authentication
   - Kerberos authentication
4. Click **Save Changes**.

Set the HDFS authentication method (CLI)

Configure the HDFS authentication method in each access zone using the command-line interface.

If you want to Hadoop clients to connect to an access zone through Kerberos, a Kerberos authentication provider must be configured and added to the access zone.

Run the `isi hdfs settings modify` command.
The following command specifies that Hadoop compute clients connecting to the zone3 must be identified through the simple authentication method:

```
isi hdfs settings modify  --authentication-mode=simple_only --zone3
```

The following command specifies that Hadoop compute clients connecting to zone3 must be identified through the Kerberos authentication method:

```
isi zone zones modify zone3 --authentication-mode=kerberos_only
```

To ensure that users can authenticate through Kerberos, you must modify the `core-site.xml` file on clients running Hadoop 2.2 and later.

Configure Kerberos authentication for Hadoop clients (CLI)

If you want Hadoop compute clients running Hadoop 2.2 and later to connect to an access zone through Kerberos, you must modify the `core-site.xml` and `hdfs-site.xml` files on the Hadoop clients.

Kerberos must be set as the HDFS authentication method in the access zone and a Kerberos authentication provider must be configured and assigned to the access zone.

Note that if you are changing the `core-site.xml` and `hdfs-site.xml` files directly with an editor per the instructions below, this will work, but those changes will likely be overwritten. This is because these two configuration files are frequently overwritten by the Ambari or Cloudera Navigator user interfaces. Therefore, if you are managing the cluster with Ambari or Cloudera Navigator, we recommend that you use their respective user interfaces to make any configuration changes.

1. Go to the `$HADOOP_CONF` directory on your Hadoop client.

2. Open the `core-site.xml` file in a text editor.
3. Set the value of the hadoop.security.token.service.use_ip property to **`false`** as shown in the following example:

```
<property>
  <name>hadoop.security.token.service.use_ip</name>
  <value>false</value>
</property>
```

4. Save and close the `core-site.xml` file.
5. Open the `hdfs-site.xml` file in a text editor.
6. Set the value of the dfs.namenode.kerberos.principal.pattern property to the Kerberos realm configured in the Kerberos authentication provider as shown in the following example:

```
<property>
  <name>dfs.namenode.kerberos.principal.pattern</name>
  <value>hdfs/*@storage.company.com</value>
</property>
```

7. Save and close the `hdfs-site.xml` file.

Creating a local Hadoop user

OneFS must be able to look up local Hadoop users by name. If there are no directory services in an access zone that can perform a user lookup, you must create a local Hadoop user that maps to a user on a Hadoop compute client for that access zone. If directory services are available, a local user account is not required. You can create a local Hadoop user using either the OneFS web administration interface (Web UI) or the command-line interface (CLI).

Create a local Hadoop user (Web UI)

Create a local Hadoop user using the OneFS web administration interface.

1. Click **Access** > **Membership & Roles** > **Users**.
2. From the **Current Access Zone** list, select the access zone that you want to create a local Hadoop user for.
3. From the **Providers** list, select **LOCAL**.
4. Click **Create User**, and then type a name for the Hadoop user in the **Username** field.
5. Click **Create User**.

Create a local Hadoop user (CLI)

Create a local Hadoop user using the command-line interface.

Run the `isi auth users create` command.
The following command creates a user who is named hadoop-user1 and assigns the user to the local authentication provider in the zone3 access zone:

```
isi auth users create --name=hadoop-user1 --provider=local --zone=zone3
```

Enabling the WebHDFS REST API

OneFS supports access to HDFS data through WebHDFS REST API client applications.

WebHDFS is a RESTful programming interface based on HTTP operations such as GET, PUT, POST, and DELETE that is available for creating client applications. WebHDFS client applications allow you to access HDFS data and perform HDFS operations through HTTP and HTTPS.

- WebHDFS is supported by OneFS on a per-access zone basis and is enabled by default.
- WebHDFS supports simple authentication or Kerberos authentication. If the HDFS authentication method for an access zone is set to `All`, OneFS uses simple authentication for WebHDFS.

- To prevent unauthorized client access through simple authentication, disable WebHDFS in each access zone that should not support it.

You can specify whether access to HDFS data through WebHDFS client applications is supported in each access zone using either the OneFS web administration interface or the command-line interface.

Enable or disable WebHDFS (Web UI)

Configure access to HDFS data through WebHDFS client applications using the OneFS web administration interface.

1. Click **Protocols** > **Hadoop (HDFS)** > **Settings**.
2. From the **Current Access Zone** list, select the access zone that you want to enable or disable WebHDFS for.
3. From the **HDFS Protocol Settings** area, select or clear the **Enable WebHDFS Access** checkbox.
4. Click **Save Changes**.

Enable or disable WebHDFS (CLI)

Configure access to HDFS data through WebHDFS client applications using the command-line interface.

Run the `isi hdfs settings modify` command.
The following command enables WebHDFS in zone3:

```
isi hdfs settings modify --webhdfs-enabled=yes --zone=zone3
```

The following command disables WebHDFS in zone3:

```
isi hdfs settings modify --webhdfs-enabled=no --zone=zone3
```

Configuring secure impersonation

Secure impersonation enables you to create proxy users that can impersonate other users to run Hadoop jobs.

You might configure secure impersonation if you use applications, such as Apache Oozie, to automatically schedule, manage, and run Hadoop jobs. For example, you can create an Oozie proxy user that securely impersonates a user called HadoopAdmin, which allows the Oozie user to request that Hadoop jobs be performed by the HadoopAdmin user.

You configure proxy users for secure impersonation on a per–zone basis, and users or groups of users that you assign as members to the proxy user must be from the same access zone. A member can be one or more of the following identity types:

- User specified by user name or UID
- Group of users specified by group name or GID
- User, group, machine, or account specified by SID
- Well-known user specified by name

If the proxy user does not present valid credentials or if a proxy user member does not exist on the cluster, access is denied. The proxy user can only access files and sub-directories located in the HDFS root directory of the access zone. It is recommended that you limit the members that the proxy user can impersonate to users that have access only to the data the proxy user needs.

ⓘ **NOTE:** Names cannot contain the following invalid characters:

" / \ [ ] : ; | = , + * ? < >

Create a proxy user (Web UI)

Create a proxy user using the OneFS web administration interface.

Add the users that you want to designate as proxy users or members to the PowerScale cluster. The proxy user and its members must belong to the same access zone.

1. Click **Protocols** > **Hadoop (HDFS)** > **Proxy Users**.
2. From the **Current Access Zone** list, select the access zone in which you want to add a proxy user.

3. Click **Create a Proxy User**.
4. In the **Name** field, type or browse for the user that you want to designate as a new proxy user.

   If you browse for a user, you can search within each authentication provider that is assigned to the current access zone in the **Select a User** dialog box.
5. Click **Add a Member**. The **Select a User, Group, or Well-known SID** dialog box appears.
6. In the **Search for** area, select the type of member that you want to search for.

   Members can be individual users or groups. You can search for a user or group by name or by well-known SID.
7. Optional: Click **Search** to display the search results based on the search criteria.
8. Select the member that you want from the **Search Results** list, and then click **Select**.
   The **Select a User, Group, or Well-known SID** dialog box closes.
9. Click **Create a Proxy User**.

Create a proxy user (CLI)

Create a proxy user using the command-line interface.

Add the users that you want to designate as proxy users or members to the PowerScale cluster. The proxy user and its members must belong to the same access zone.

Run the `isi hdfs proxyusers create` command.
The following command designates hadoop-user23 in zone1 as a new proxy user:

```
isi hdfs proxyusers create hadoop-user23 --zone=zone1
```

The following command designates hadoop-user23 in zone1 as a new proxy user and adds the group hadoop-users to the list of members that the proxy user can impersonate:

```
isi hdfs proxyusers create hadoop-user23 --zone=zone1 --add-group=hadoop-users
```

The following command designates hadoop-user23 in zone1 as a new proxy user and adds UID 2155 to the list of members that the proxy user can impersonate:

```
isi hdfs proxyusers create hadoop-user23 --zone=zone1 --add-UID=2155
```

Modify a proxy user (Web UI)

Modify the list of members that a proxy user securely impersonates using the PowerScale web administration interface.

1. Click **Protocols** > **Hadoop (HDFS)** > **Proxy Users**.
2. From the **Current Access Zone** list, select the access zone for which you want to modify a proxy user.
3. From the **Proxy Users** list, select the checkbox next to the proxy user that you want to modify, and then click **View/Edit**.
4. From the **View Proxy User Details** dialog box, click **Edit Proxy User**.
5. Add or remove members, and then click **Save Changes**.

Modify a proxy user (CLI)

Modify the list of members that a proxy user securely impersonates using the command-line interface.

Run the `isi hdfs proxyusers modify` command.
The following command removes a user with the user ID 2155 and adds a well-known user who is named LOCAL to the list of members for proxy user hadoop-user23 in zone1:

```
isi hdfs proxyusers modify hadoop-user23 --zone=zone1 --add-wellknown=LOCAL --remove-
uid=2155
```

View proxy users (Web UI)

View a list of all proxy users in an access zone and view individual proxy user details using the OneFS web administration interface.

1. Click **Protocols** > **Hadoop (HDFS)** > **Proxy Users**.
2. From the **Current Access Zone** list, select the access zone in which you want to view a proxy user.
   The **Proxy Users** list displays all proxy users who are configured in the access zone.
3. From the **Proxy Users** list, select the checkbox next to the proxy user that you want to view, and then click **View/Edit**.
   The **View Proxy User Details** dialog box appears.
4. Click **Close** when you are finished viewing proxy user details.

View proxy users (CLI)

View a list of all proxy users in an access zone and view individual proxy user details using the command-line interface.

1. Open a secure shell (SSH) connection to any node in the cluster and log in.
2. To view a list of all proxy users configure in a specific access zone, run the `isi hdfs proxyusers list` command.
   The following command displays a list of all proxy users configured in zone1:

   ```
   isi hdfs proxyusers list --zone=zone1
   ```

3. To view the configuration details for a specific proxy user, run the `isi hdfs proxyusers view`command.
   The following command displays the configuration details for the hadoop-user23 proxy user in zone1:

   ```
   isi hdfs proxyusers view hadoop-user23 --zone=zone1
   ```

View the member list of a proxy user (CLI)

Display the list of users and groups, known as members, assigned to a proxy user. The proxy user can securely impersonate any user in the member list.

This procedure is available only through the command-line interface.

Run the `isi hdfs proxyusers members list` command.
The following command displays a detailed list of the users and groups of users that are members of proxy user hadoop-user23 in zone1:

```
isi hdfs proxyusers members list hadoop-user23 --zone=zone1 -v
```

Delete a proxy user (Web UI)

Delete a proxy user from an access zone using the OneFS web administration interface.

Deleting a proxy user deletes the user from the list of users that can perform secure impersonation. The user is not deleted from the system.

1. Click **Protocols** > **Hadoop (HDFS)** > **Proxy Users**.
2. From the **Current Access Zone** list, select the access zone that has the proxy user that you want to delete.
3. From the **Proxy Users** list, select the checkbox next to the proxy user that you want to delete, and then click **Delete**.
4. In the confirmation dialog box, click **Delete**.

Delete a proxy user (CLI)

Delete a proxy user from an access zone using the command-line interface.

Deleting a proxy user deletes the user from the list of users that can perform secure impersonation. The user is not deleted from the system.

Run the `isi hdfs proxyusers delete` command.
The following command deletes the proxy user hadoop-user23 from the zone1 access zone:

```
isi hdfs proxyusers delete hadoop-user23 --zone=zone1
```

Configuring virtual HDFS racks

You can create a virtual HDFS rack of nodes on your PowerScale cluster to optimize performance and reduce latency when accessing HDFS data.

OneFS enables you to specify a group of preferred HDFS nodes on your PowerScale cluster and an associated group of Hadoop compute clients as a virtual HDFS rack. Virtual HDFS racks allow you to fine-tune client connectivity by directing Hadoop compute clients to go through quicker, less-busy switches or to faster nodes, depending on your network topology.

When a Hadoop compute client from the specified group connects to the cluster, OneFS returns at least two IP addresses from the group of preferred HDFS nodes. You specify the preferred HDFS nodes by IP address pool. Virtual HDFS racks do not support IP address pools in the IPv6 family.

Create a virtual HDFS rack (Web UI)

Create a virtual HDFS rack of nodes on your PowerScale cluster using the OneFS web administration interface.

1. Click **Protocols** > **Hadoop (HDFS)** > **Virtual Racks**.
2. From the **Current Access Zone** list, select the access zone in which you want to add a virtual HDFS rack.
3. Click **Create a Virtual Rack**.
4. In the **Name** field, type a name for the new virtual rack.

   A rack name must begin with a forward slash—for example, `/hdfs-rack2`.
5. In the **Client IP Ranges** fields, specify the IP address range of Hadoop compute clients to be associated with the virtual HDFS rack.

   You can associate multiple IP ranges.
6. From the **IP Pools** area, select the IP address pool that you want from the **Available Pools** table and click **Add**.
7. Click **Create Virtual Rack**.

Create a virtual HDFS rack (CLI)

Create a virtual HDFS rack of nodes on your PowerScale cluster using the command-line interface.

Run the `isi hdfs racks create` command.

A rack name begins with a forward slash—for example, `/hdfs-rack2`.

The following command creates a rack named `/hdfs-rack2` in the zone5 access zone:

```
isi hdfs racks create /hdfs-rack2 --zone=zone5
```

The following command creates a rack named `/hdfs-rack2` in the zone5 access zone, specifies 120.135.26.10-120.135.26.20 as the IP address range of Hadoop compute clients associated with the rack, and specifies subnet0:pool0 as the IP address pool of OneFS nodes assigned to the rack:

```
isi hdfs racks create /hdfs-rack2 --zone=zone5 --client-ip-
ranges=120.135.26.10-120.135.26.20 --ip-pools=subnet0:pool0
```

Modify a virtual HDFS rack (Web UI)

Modify the settings of a virtual HDFS rack using the OneFS web administration interface.

1. Click **Protocols** > **Hadoop (HDFS)** > **Virtual Racks**.
2. From the **Current Access Zone** list, select the access zone in which you want to modify a virtual HDFS rack.

3. From the **Virtual Racks** list, select the checkbox next to the virtual HDFS rack that you want to modify, and then click **View/Edit**.
4. From the **View Virtual Rack Settings** dialog box, click **Edit Virtual Rack**.
5. Modify virtual rack settings, and then click **Save Changes**.

Modify a virtual HDFS rack (CLI)

Modify the settings of a virtual HDFS rack using the command line interface.

Run the `isi hdfs racks modify` command.

A rack name begins with a forward slash—for example, `/hdfs-rack2`.

The following command renames a rack that is named `/hdfs-rack2` in the zone3 access zone to /hdfs-rack5:

```
isi hdfs racks modify /hdfs-rack2 --new-name=/hdfs-rack5 --zone=zone3
```

The following command adds 120.135.26.30-120.135.26.40 to the list of existing Hadoop compute client IP addresses assigned to `/hdfs-rack2` in the zone3 access zone:

```
isi hdfs racks modify /hdfs-rack2 --add-client-ip-ranges=120.135.26.30-120.135.26.40 --
zone=zone3
```

In addition to adding a range to the list of existing ranges, you can modify the client IP address ranges by replacing the current ranges, deleting a specific range or deleting all ranges.

The following command replaces the existing IP pools with subnet1:pool1 and subnet2:pool2 assigned to /hdfs-rack2 in the zone3 access zone:

```
isi hdfs racks modify /hdfs-rack2 --ip-pools=subnet1:pool1,subnet2:pool2 --zone=zone3
```

In addition to replacing the list of existing pools with new pools, you can modify the IP pools by adding pools to the list of current pools, deleting a specific pool or deleting all pools.

View virtual HDFS racks (Web UI)

View a list of all the virtual HDFS racks in an access zone and view individual virtual rack details using the OneFS web administration interface.

1. Click **Protocols** > **Hadoop (HDFS)** > **Virtual Racks**.
2. From the **Current Access Zone** list, select the access zone in which you want to view a virtual HDFS rack.
   The **Virtual Racks** list displays all virtual HDFS racks that are configured in the access zone.
3. From the **Virtual Racks** list, select the checkbox next to the virtual HDFS rack that you want to view, and then click **View/Edit**.
   The **View Virtual Rack Settings** dialog box appears.
4. Click **Close** when you are finished viewing virtual HDFS rack details.

View virtual HDFS racks (CLI)

View a list of all virtual HDFS racks in an access zone and view individual virtual rack details using the command line interface.

1. Open a secure shell (SSH) connection to any node in the cluster and log in.
2. To view a list of all virtual HDFS racks configured in an access zone, run the `isi hdfs racks list` command.
   The following command lists all HDFS racks configured in the zone1 access zone:

   ```
   isi hfds racks list --zone=zone1
   ```

   The following command displays setting details for all virtual HDFS racks configured in the zone1 access zone:

   ```
   isi hdfs racks list --zone=zone1 -v
   ```
3. To view the setting details for a specific virtual HDFS rack, run the `isi hdfs racks view` command:

Each rack name begins with a forward slash—for example `/hdfs-rack2`.

The following example command displays setting details for the virtual HDFS rack named /hdfs-rack2 that is configured in the zone1 access zone:

```
isi hdfs racks view /hdfs-rack2 --zone=zone1
```

Delete a virtual HDFS rack (Web UI)

Delete a virtual HDFS rack from an access zone using the OneFS web administration interface.

1. Click **Protocols** > **Hadoop (HDFS)** > **Virtual Racks**.
2. From the **Current Access Zone** list, select the access zone in which you want to delete a virtual HDFS rack.
3. From the **Virtual Racks** list, select the checkbox next to the virtual HDFS rack that you want to delete, and then click **Delete**.
4. In the confirmation dialog box, click **Delete**.

Delete a virtual HDFS rack (CLI)

Delete a virtual HDFS rack from an access zone using the command-line interface.

1. Run the `isi hdfs racks delete` command.

   A rack name begins with a forward slash—for example, `/hdfs-rack2`.

   The following command deletes the virtual HDFS rack that is named `/hdfs-rack2` from the zone1 access zone:

   ```
   isi hdfs racks delete /hdfs-rack2 --zone=zone1
   ```
2. At the prompt, type **`yes`**.

Configuring HDFS wire encryption

You can configure HDFS wire encryption using either the OneFS web administration interface or the command-line interface.

If you are using OneFS 8.1.2.0 or later, you can protect data that is transmitted between an HDFS client and OneFS through data-in-flight encryption, also known as HDFS wire encryption. In a Kerberos-enabled Hadoop environment, you can enable this feature on all of the HDFS clients and on OneFS. Wire encryption manages the negotiations between an HDFS client and OneFS to encrypt and decrypt data.

HDFS wire encryption enables OneFS to encrypt data that is transmitted between OneFS and HDFS to meet regulatory requirements. Wire encryption uses Advanced Encryption Standard (AES) to encrypt the data. 128-bit, 192-bit, and 256-bit key lengths are available.

HDFS wire encryption that is supported by OneFS is different than the Apache HDFS Transparent Data Encryption technology. For more information, refer to Enhanced Hadoop security with OneFS 8.0.1 and Hortonworks HDP.

ⓘ **NOTE:** When HDFS wire encryption is enabled, there is a significant impact on the HDFS protocol throughput and I/O performance.

Configure HDFS wire encryption (Web UI)

You can configure HDFS wire encryption using the OneFS web administration interface.

1. Click **Protocols** > **Hadoop (HDFS)** > **Settings**.
2. In the **Data Transfer Cipher** list box, select one of the following options.

Option	Description
To enable HDFS wire encryption	Select one of the Advanced Encryption Standard (AES) ciphers, **AES/CTR/NoPadding with 128 bit key**, **AES/CTR/NoPadding with 192 bit key**, or **AES/CTR/NoPadding with 256 bit key**.
To disable HDFS wire encryption	Select **Do not encrypt data.**

3. Click **Save Settings**.

Configure HDFS wire encryption (CLI)

You can configure HDFS wire encryption using the command-line interface.

Perform the task "Configure Ranger plugin settings" before configuring HDFS wire encryption.

To configure HDFS wire encryption, run `isi hdfs settings modify --data-transfer-cipher` *encryption_argument*.

Option	Description
To enable HDFS wire encryption	Set the *encryption_argument* to one of the Advanced Encryption Standard (AES) ciphers, `aes_128_ctr`,`aes_192_ctr` , or `aes_256_ctr`.
To disable HDFS wire encryption	Set the *encryption_argument* to `none`

```
isi hdfs settings modify --data-transfer-cipher aes_128_ctr
```

Configuring HDFS transparent data encryption

If you are using OneFS 8.2.0 or later, you can encrypt in-flight and data at-rest Hadoop datasets between HDFS clients and OneFS through transparent data encryption (TDE). You can enable this feature on all HDFS clients and on OneFS.

HDFS Transparent Data Encryption implements transparent end-to-end encryption. With TDE, the data can only be encrypted and decrypted by the client, and HDFS never stores or has access to unencrypted data or unencrypted data encryption keys.

Transparent data encryption satisfies two typical security, regulatory, and compliance requirements for encryption:

- Data encryption on persistent media, such as on-disk data
- In-transit encryption, for example when data is traveling over the network

How transparent data encryption works

Once configured, data read from and written to configured HDFS directories, called *encryption zones*, is transparently encrypted and decrypted without requiring changes to user application code. The contents of encryption zones are transparently encrypted upon write and transparently decrypted upon read.

Each encryption zone is associated with a single encryption zone key, which is specified when the zone is created. Each file within an encryption zone has its own unique *data encryption key* (DEK). Data encryption keys are never handled directly by HDFS. Instead, HDFS only handles an *encrypted data encryption key* (EDEK). HDFS clients decrypt the EDEK, and then use the subsequent DEK to read and write data. During this transfer, HDFS data nodes simply see a stream of encrypted bytes.

Limitations

- HDFS TDE for OneFS 8.2.0 is limited to HDFS client access only. Note that HDFS TDE is not a multi-protocol feature. Once the data is ingested, it can only be decrypted by the HDFS client.
- Data written to and read from HDFS encryption zones should only be accessed over the HDFS protocol.
- The OneFS web administration interface configuration is not available for Key Management Server (KMS) configuration or for encryption zone management. Encryption zones can only be created using the OneFS command-line interface (CLI). Hadoop command-line tools to create encryption zones are not currently supported in OneFS 8.2.0.
- SyncIQ will not move encryption zones for OneFS 8.2.0.

You can configure TDE using the following command-line interface (CLI) commands:

- `isi hdfs crypto encryption-zones create`
- `isi hdfs crypto settings modify`

- `isi hdfs crypto settings view`
- `isi hdfs crypto encryption-zones list`

The CLI syntax is described in the **HDFS Commands** section of this guide.

Configure HDFS transparent data encryption (CLI)

Configure HDFS TDE using the OneFS command-line interface (CLI). Read the following workflow before you begin.

1. On the Hadoop client, create an encryption zone key on the Key Management Server (KMS) responsible for generating encryption keys for the files. Note that the keyadmin user can be used to perform key creation.

   ```
   ./hadoop key create key1 -provider <provider-path>
   ```

   For example:

   ```
   hadoop key create key5 -provider kms://http@ambari100-c.west.isilon.com:9292/kms
   ```

   If you do not want to add the *-provider* option to the necessary `hadoop key <operation>` command, find your environment below and set the `hadoop.security.key.provider.path` property.

KMS	HDP < 2.6.x	HDP >= 3.0.1
Ranger KMS	When you add the Ranger KMS, you will be prompted with the recommended settings for the KMS provider. The property is set automatically in `HDFS > Configs > Advanced > Advanced core-site`. If the property is not configured automatically, add it to the `custom core-site.xml` file. Set the property in `HDFS > Configs > Advanced > Custom core-site`.	Set property in `Ambari > Services > OneFS > Configs > Advanced > Custom core-site`
Other KMS	Set the property in `HDFS > Configs > Advanced > Custom core-site`. If the property is not configured automatically, add it to the `custom core-site.xml` file.	Set property in `Ambari > Services > OneFS > Configs > Advanced > Custom core-site`

   Steps:

   HDP version < 2.6.x -- not using Ranger KMS

   a. Navigate to `HDFS > Configs > Advanced > Custom core-site`.
   b. Click **Add Property**.
   c. Enter the property as: `hadoop.security.key.provider.path=kms://<kms-url>/kms`

      For example,

      ```
      hadoop.security.key.provider.path=kms://http@m105.solarch.lab.emc.com:1688/kms
      ```

   d. Click **Add**.
   e. Save settings.

   HDP version 3.0.1 or later -- any KMS

   a. Navigate to `Ambari > Services > OneFS > Configs > Advanced > Custom core-site`.
   b. Click **Add Property**.
   c. Enter the property as: `hadoop.security.key.provider.path=kms://<kms-url>/kms`

      For example,

      ```
      hadoop.security.key.provider.path=kms://http@m105.solarch.lab.emc.com:9292/kms
      ```

   d. Click **Add**.
   e. Save settings.

   Authorization Exception Errors

The OneFS Key Management Server configuration is configured per zone. If you receive an Authorization Exception error similar to the following:

```
key1 has not been created.
org.apache.hadoop.security.authorize.AuthorizationException: User:hdfs not allowed to
do 'CREATE_KEY' on 'key1'
```

then log into Ranger as the `keyadmin` user and perform the following step. Note that the default password is `keyadmin`.

- Click on the KMS instance and edit the user you want to allow key administration privileges and then save the changes.

Note that this example uses the Ranger KMS server. Follow similar procedures for other KMS servers to fix user authorization issues.

2. On the OneFS cluster, configure the KMS URL, create a directory, and make it an encryption zone. The encryption zone must be somewhere within the HDFS root directory for that zone.

```
isi hdfs crypto settings modify --kms-url<string>

isi hdfs crypto encryption-zones create <path><keyname>
```

For example:

```
isi hdfs crypto settings modify --kms-url=http://m105.solarch.lab.emc.com:9292 --
zone=hdfs -v

isi hdfs crypto settings view --zone=hdfs

isi hdfs crypto encryption-zones create --path=/ifs/hdfs/A --key-name=keyA --
zone=hdfs -v
```

When you run the `isi hdfs crypto settings` command, note in the output that **Port 9292** is specific to the Ranger KMS server. If you are using a different KMS server, a different port may be required. Do not use Port 6080, as this port is for the Ranger UI only.

The HDFS root path in the example above is `/ifs/hdfs`. Change this to your HDFS root path followed by the empty directory corresponding to your encryption zone. For example, `/ifs/hdfs/A` as in the example above.

Important: Do not create the encryption zone from a DFS client in the Hadoop cluster. The encryption zone must be created using the OneFS CLI as shown above, otherwise you will see an error similar to the following on the console and in the OneFS hdfs.log file:

```
RemoteException: Unknown RPC: createEncryptionZone
```

3. List the encryption zones.

```
isi hdfs crypto encryption-zones list
```

With the encryption zone defined on the OneFS cluster, you will be able to list the encryption zone immediately from any DFS client in the Hadoop cluster.

4. The next step is to test the reading/writing of a file to the created encryption zone by an authorized user.
The `ambari-qa` user is the default smoke-test user that comes with HDP. For this test, the KMS server is updated to allow the `ambari-qa` user to obtain the keys and metadata as well as to generate and decrypt encryption keys. With the policy updated on the KMS server for the `ambari-qa` user, you can proceed to test writing and reading a test.txt file from the created encryption zone on OneFS (for example, to the /A encryption zone as in the previous example) from a DFS client in the Hadoop cluster as the `ambari-qa` user.

```
cat test.txt
hdfs dfs -put test.txt /A
hdfs dfs -ls /A
hdfs dfs -cat /A/test.txt
```

5. Verify that the test file is actually encrypted on the OneFS cluster by logging into OneFS as the root administrator and displaying the contents of the test file in the test directory`/ifs/hdfs/A` in our example .

   ```
   cd /ifs/hdfs/A
   ls
   cat test.txt
   ```

   Result: You should see that the contents of the test file are encrypted and the original text is not displayed even by the privileged root user on OneFS. The test file created by the `ambari-qa` user has read permissions for both the Hadoop group and everyone, since the "hive" user is defined in the KMS with decrypt privileges. The "hive" user can decrypt the file created by the `ambari-qa` user, but cannot place any files into the encryption zone, in this case /A, since the *write* permission is missing for the Hadoop group that the "hive" user is a member of.
6. If you do the same test with a user not defined in the KMS for the specified encryption zone, for example the "mapred" user, reading of the test file is denied as shown in the following example.

   ```
   hdfs dfs -cat /A/test.txt

   cat: user:mapred not allowed to do 'DECRYPT_EEK' on 'keyA'
   ```
7. If you need to delete the encryption zone, deleting the encryption zone directory on OneFS is sufficient to delete the encryption zone.

4

OneFS with HDFS command reference

You can access and configure the HDFS service through the OneFS command-line interface. These commands perform the same operations as the OneFS web administration interface. These commands in this section are provided as a reference.

Topics:

HDFS commands

The following list of OneFS commands will help you to manage your OneFS and Hadoop system integration.

isi hdfs crypto settings modify

Configures HDFS for transparent data encryption.

Syntax

```
isi hdfs crypto settings modify
  [--kms-url <string>]
```

Options

--kms-url *<string>*

Specifies the URL of the Key Management Server.

isi hdfs crypto settings view

View settings for HDFS transparent data encryption.

Syntax

```
isi hdfs crypto settings view
```

Options

There are no options for this command.

isi hdfs crypto encryption-zones create

Create and name the HDFS encryption zones for transparent data encryption.

Syntax

```
isi hdfs crypto encryption-zones create <path><keyname>
```

Options

<path> <keyname>

Specifies a directory and key name for the encryption zone.

The encryption zone must be somewhere within the HDFS root directory for that zone.

isi hdfs crypto encryption-zones list

List the HDFS encryption zones.

Syntax

```
isi hdfs crypto encryption-zones list
```

Options

There are no options for this command.

isi hdfs fsimage job settings modify

Change the interval between successive FSImages.

Syntax

```
isi hdfs fsimage job settings modify
  [--generation-interval <string>]
  [--verbose]
  [--zone <string>]
```

Options

--generation-interval *<string>*

The interval between successive FSImages.

--help *<string>*

Display help for this command.

{--verbose | -v}

Display more detailed information.

--zone *<string>*

The access zone to which the HDFS settings apply.

isi hdfs fsimage job settings view

Review the frequency of an FSImage job.

Syntax

```
isi hdfs fsimage job settings view
  [--zone <string>]
```

Options

--help *<string>*

Display help for this command.

--zone *<string>*

The access zone to which the HDFS settings apply.

isi hdfs fsimage job view

Review the status of an FSImage job.

Syntax

```
isi hdfs fsimage job view
  [--zone <string>]
```

Options

--help *<string>*

Display help for this command.

--zone *<string>*

The access zone to which the HDFS settings apply.

isi hdfs fsimage latest delete

Delete the latest FSImage.

Syntax

```
isi hdfs fsimage latest delete
  [--zone <string>]
  [{--verbose | -v}]
  [{--force | -f}]
```

Options

--zone *<string>*

The access zone to which the HDFS settings apply.

{**--verbose** | **-v**}

Display more detailed information.

{**--force** | **-f**}

Do not prompt for confirmation.

isi hdfs fsimage latest view

Review the latest FSImage.

Syntax

```
isi hdfs fsimage latest view
  [--zone <string>]
```

Options

--help *<string>*

Display help for this command.

--zone *<string>*

The access zone to which the HDFS settings apply.

isi hdfs fsimage settings modify

Enable FSImage on the HDFS access zone. For more information, see the Additional Resources - Cloudera Navigator section of this guide.

http://doc.isilon.com/onefs/hdfs/05-ifs-c-hdfs-admin-guide-overview-chapter.htm

Syntax

```
isi hdfs fsimage settings modify
  [--enabled {yes | no}]
  [--zone <string>]
  [--verbose]
```

Options

--enabled {**yes** | **no**}

Enables or disables the HDFS FSImage service. Allow access to FSImage and start FSImage generation. The HDFS FSImage service is disabled by default. This service should only be enabled on a Hadoop-enabled Access Zone that will use Cloudera Navigator.

--help *<string>*

Display help for this command.

{**--verbose** | **-v**}

Display more detailed information.

--zone *<string>*

The access zone to which the HDFS settings apply.

isi hdfs inotify settings view

Review the configuration of the INotify stream.

Syntax

```
isi hdfs inotify settings view
  [--zone <string>]
```

Options

--help *<string>*

Display help for this command.

--zone *<string>*

The access zone to which the HDFS settings apply.

isi hdfs inotify settings modify

Enable INotify on the HDFS access zone.

Syntax

```
isi hdfs inotify settings modify
  [--enabled {yes | no}]
  [--maximum-delay <string>]
  [--retention <string>]
  [--zone <string>]
  [--verbose]
```

Options

--enabled {yes | no}

Allows access to FSImage and starts FSImage generation. The HDFS FSImage service is disabled by default. This service should only be enabled on a Hadoop-enabled access zone that will use Cloudera Navigator.

--help *<string>*

Display help for this command.

--maximum-delay *<string>*

The maximum duration until an edit event is reported in INotify.

--retention *<string>*

The minimum duration edits will be retained.

{--verbose | -v}

Display more detailed information.

--zone *<string>*

The access zone to which the HDFS settings apply.

isi hdfs inotify stream reset

Reset the INotify stream by deleting collected events.

Syntax

```
isi hdfs inotify stream reset
  [--zone <string>]
  [{--verbose | -v}]
  [{--force | -f}]
```

Options

--zone *<string>*

The access zone to which the HDFS settings apply.

{--verbose | -v}

Display more detailed information.

{--force | -f}

Do not prompt for confirmation.

isi hdfs inotify stream view

Review the INotify stream.

Syntax

```
isi hdfs inotify stream view
  [--zone <string>]
```

Options

--help *<string>*

Display help for this command.

--zone *<string>*

The access zone to which the HDFS settings apply.

isi hdfs log-level modify

Modifies the log level of the HDFS service on the node.

Syntax

```
isi hdfs log-level modify
  [--set {always|error|warning|info|verbose|debug|trace|default}  ]
  [--verbose|  -v]
```

Options

--set {always | error | warning | info | verbose | debug | trace | default}

Sets the default logging level for the HDFS service on the cluster. The default value is `default`.

--verbose | -v

Displays more detailed information.

isi hdfs log-level view

Displays the current log level of the HDFS service on the node.

Syntax

```
isi hdfs log-level view
```

Options

There are no options for this command.

isi hdfs proxyusers create

Creates a proxy user that can securely impersonate another user or group.

Syntax

```
isi hdfs proxyusers create <proxyuser-name>
  [--zone <zone-name>]
  [--add-group <group-name>...]
  [--add-gid <group-identifier>...]
  [--add-user <user-name>...]
  [--add-uid <user-identifier>...]
  [--add-sid <security-identifier>...]
  [--add-wellknown <well-known-name>...]
  [--verbose]
```

Options

<proxyuser-name>

Specifies the user name of a user currently configured on the cluster to be designated as a proxy user.

--zone *<zone-name>*

Specifies the access zone the user authenticates through.

--add-group *<group-name>...*

Adds the group specified by name to the list of proxy user members. The proxy user can impersonate any user in the group. The users in the group must authenticate to the same access zone as the proxy user. You can specify multiple group names in a comma-separated list.

--add-gid *<group-identifier>...*

Adds the group by specified by UNIX GID to the list of proxy user members. The proxy user can impersonate any user in the group. The users in the group must authenticate to the same access zone as the proxy user. You can specify multiple UNIX GIDs in a comma-separated list.

--add-user *<user-name>...*

Adds the user specified by name to the list of members the proxy user can impersonate. The user must authenticate to the same access zone as the proxy user. You can specify multiple user names in a comma-separated list.

--add-uid *<user-identifier>...*

Adds the user specified by UNIX UID to the list of members the proxy user can impersonate. The user must authenticate to the same access zone as the proxy user. You can specify multiple UNIX UIDs in a comma-separated list.

`--add-sid` *<security-identifier>...*

Adds the user, group of users, machine or account specified by Windows SID to the list of proxy user members. The object must authenticate to the same access zone as the proxy user. You can specify multiple Windows SIDs in a comma-separated list.

`--add-wellknown` *<well-known-name>...*

Adds the well-known user specified by name to the list of members the proxy user can impersonate. The well-known user must authenticate to the same access zone as the proxy user. You can specify multiple well-known user names in a comma-separated list.

{ **`--verbose`** | **`-v`**}

Displays more detailed information.

Examples

The following command designates hadoop-user23 in zone1 as a new proxy user:

```
isi hdfs proxyusers create hadoop-user23 --zone=zone1
```

The following command designates hadoop-user23 in zone1 as a new proxy user and adds the group of users named hadoop-users to the list of members that the proxy user can impersonate:

```
isi hdfs proxyusers create hadoop-user23 --zone=zone1 \
--add-group=hadoop-users
```

The following command designates hadoop-user23 in zone1 as a new proxy user and adds UID 2155 to the list of members that the proxy user can impersonate:

```
isi hdfs proxyusers create hadoop-user23 --zone=zone1 --add-UID=2155
```

isi hdfs proxyusers modify

Modifies a proxy user that can securely impersonate another user or group.

Syntax

```
isi hdfs proxyusers modify <proxyuser-name>
  [--zone <zone-name>]
  [--add-group <group-name>...]
  [--add-gid <group-identifier>...]
  [--add-user <user-name>...]
  [--add-uid <user-identifier>...]
  [--add-sid <security-identifier>...]
  [--add-wellknown <well-known-name>...]
  [--remove-group <group-name>...]
  [--remove-gid <group-identifier>...]
  [--remove-user <user-name>...]
  [--remove-uid <user-identifier>...]
  [--remove-sid <security-identifier>...]
  [--remove-wellknown <well-known-name>...]
  [--verbose]
```

Options

<proxyuser-name>

Specifies the user name of the proxy user to be modified.

--zone *<zone-name>*

Specifies the access zone that the proxy user authenticates through.

--add-group *<group-name>...*

Adds the group specified by name to the list of proxy user members. The proxy user can impersonate any user in the group. The users in the group must authenticate to the same access zone as the proxy user. You can specify multiple group names in a comma-separated list.

--add-gid *<group-identifier>...*

Adds the group specified by UNIX GID to the list of proxy user members. The proxy user can impersonate any user in the group. The users in the group must authenticate to the same access zone as the proxy user. You can specify multiple UNIX GIDs in a comma-separated list.

--add-user *<user-name>...*

Adds the user specified by name to the list of members the proxy user can impersonate. The user must authenticate to the same access zone as the proxy user. You can specify multiple user names in a comma-separated list.

--add-uid *<user-identifier>...*

Adds the user specified by UNIX UID to the list of members the proxy user can impersonate. The user must authenticate to the same access zone as the proxy user. You can specify multiple UNIX UIDs in a comma-separated list.

--add-sid *<security-identifier>...*

Adds the user, group of users, machine or account specified by Windows SID to the list of proxy user members. The object must authenticate to the same access zone as the proxy user. You can specify multiple Windows SIDs in a comma-separated list.

--add-wellknown *<well-known-name>...*

Adds the well-known user specified by name to the list of members the proxy user can impersonate. The well-known user must authenticate to the same access zone as the proxy user. You can specify multiple well-known user names in a comma-separated list.

--remove-group *<group-name>...*

Removes the group specified by name from the list of proxy user members so that the proxy user can no longer impersonate any user in the group. You can specify multiple group names in a comma-separated list.

--remove-gid *<group-identifier>...*

Removes the group specified by UNIX GID from the list of proxy user members so that the proxy user can no longer impersonate any user in the group. You can specify multiple UNIX GIDs in a comma-separated list.

--remove-user *<user-name>...*

Removes the user specified by name from the list of members the proxy user can impersonate. You can specify multiple user names in a comma-separated list.

--remove-uid *<user-identifier>...*

Removes the user specified by UNIX UID from the list of members the proxy user can impersonate. You can specify multiple UNIX UIDs in a comma-separated list.

--remove-sid *<security-identifier>...*

Removes the user, group of users, machine or account specified by Windows SID from the list of proxy user members. You can specify multiple Windows SIDs in a comma-separated list.

--remove-wellknown *<well-known-name>...*

Removes the well-known user specified by name from the list of members the proxy user can impersonate. You can specify multiple well-known user names in a comma-separated list.

{**--verbose** | **-v**}

Displays more detailed information.

Examples

The following command adds the well-known local user to, and removes the user whose UID is 2155 from, the list of members for proxy user hadoop-user23 in zone1:

```
isi hdfs proxyusers modify hadoop-user23 --zone=zone1 \
--add-wellknown=local --remove-uid=2155
```

isi hdfs proxyusers delete

Deletes a proxy user.

Syntax

```
isi hdfs proxyusers delete <proxyuser-name>
  [--zone <zone-name>]
  [--force]
  [--verbose]
```

Options

<proxyuser-name>

Specifies the user name of the proxy user to be deleted.

--zone *<zone-name>*

Specifies the access zone that the proxy user authenticates through.

{ **--force** | **-f**}

Deletes the specified proxy user without requesting confirmation.

{ **--verbose** | **-v**}

Displays more detailed information.

Examples

The following command deletes hadoop-user23 in zone1 from the list of proxy users:

```
isi hdfs proxyusers delete hadoop-user23 --zone=zone1
```

isi hdfs proxyusers members list

Displays the users and groups of users, known as members, that can be impersonated by a proxy user.

Syntax

```
isi hdfs proxyusers members list <proxyuser-name>
  [--zone <zone-name>]
  [--format {table | json | csv | list}]
  [--no-header ]
  [--no-footer ]
  [--verbose]
```

Options

<proxyuser-name>

Specifies the name of the proxy user.

--zone *<zone-name>*

Specifies the access zone the proxy user authenticates through.

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

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

--no-header

Displays table and CSV output without headers.

--no-footer

Displays table output without footers.

{ **--verbose** | **-v**}

Displays more detailed information.

Examples

The following command displays a detailed list of the users and groups that are members of proxy user hadoop-user23 in zone1:

```
isi hdfs proxyusers members list hadoop-user23 --zone=zone1 -v
```

The system displays output similar to the following example:

```
Type: user
Name: krb_user_005
  ID: UID:1004
--------------------------------------------------------------------------------
Type: group
Name: krb_users
  ID: SID:S-1-22-2-1003
--------------------------------------------------------------------------------
Type: wellknown
Name: LOCAL
  ID: SID:S-1-2-0
```

isi hdfs proxyusers list

Displays all proxy users that are configured in an access zone.

Syntax

```
isi hdfs proxyusers list
  [--zone <zone-name>]
  [--format {table | json | csv | list}]
  [--no-header ]
  [--no-footer ]
  [--verbose]
```

Options

--zone *<zone-name>*

Specifies the name of the access zone.

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

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

--no-header

Displays table and CSV output without headers.

--no-footer

Displays table output without footers.

{ **--verbose** | **-v**}

Displays more detailed information.

Examples

The following command displays a list of all proxy users that are configured in zone1:

```
isi hdfs proxyusers list --zone=zone1
```

The system displays output similar to the following example:

```
Name
-------------
hadoop-user23
hadoop-user25
hadoop-user28
-------------
Total: 3
```

isi hdfs proxyusers view

Displays the configuration details of a specific proxy user.

Syntax

```
isi hdfs proxyusers view <proxyuser-name>
   [--zone <zone-name>]
```

Options

<proxyuser-name>

Specifies the user name of the proxy user.

--zone *<zone-name>*

Specifies the access zone the proxy user authenticates through.

Examples

The following command displays the configuration details for the hadoop-user23 proxy user in zone1:

```
isi hdfs proxyusers view hadoop-user23 --zone=zone1
```

The system displays output similar to the following example:

```
Name: hadoop-user23
Members: krb_users
         LOCAL
         krb_user_004
```

isi hdfs racks create

Creates a new virtual HDFS rack.

Syntax

```
isi hdfs racks create <rack-name>
  [--client-ip-ranges <low-ip-address>-<high-ip-address>]...
  [--ip-pools <subnet>:<pool>]...
  [--zone <string>]
  [--verbose]
```

Options

<rack-name>

Specifies the name of the virtual HDFS rack. The rack name must begin with a forward slash—for example, **/example-name**.

--client-ip-ranges *<low-ip-address>-<high-ip-address>...*

Specifies IP address ranges of external Hadoop compute clients assigned to the virtual rack.

--ip-pools *<subnet>:<pool>...*

Assigns a pool of OneFS cluster IP addresses to the virtual rack.

--zone *<string>*

Specifies the access zone that will contain the virtual rack.

{**--verbose** | **-v**}

Displays more detailed information.

isi hdfs racks modify

Modifies a virtual HDFS rack.

Syntax

```
isi hdfs racks modify <rack-name>
  [--name <rack-name>]
  [--client-ip-ranges <low-ip-address>-<high-ip-address>]...
  [--add-client-ip-ranges <low-ip-address>-<high-ip-address>]...
  [--remove-client-ip-ranges <low-ip-address>-<high-ip-address>]...
  [--clear-client-ip-ranges]
  [--ip-pools <subnet>:<pool>]...
  [--add-ip-pools <subnet>:<pool>]...
  [--remove-ip-pools <subnet>:<pool>]...
  [--clear-ip-pools]
  [--zone <string>]
  [--verbose]
```

Options

<rack-name>

Specifies the virtual HDFS rack to be modified. Each rack name begins with a forward slash—for example **/example-name**.

--name *<rack-name>*

Assigns a new name to the specified virtual rack. The rack name must begin with a forward slash—for example **/example-name**.

--client-ip-ranges *<low-ip-address>-<high-ip-address>*...

Specifies IP address ranges of external Hadoop compute clients assigned to the virtual rack. The value assigned through this option overwrites any existing IP address ranges. You can add a new range through the `--add-client-ip-ranges` option.

--add-client-ip-ranges *<low-ip-address>-<high-ip-address>*...

Adds a specified IP address range of external Hadoop compute clients to the virtual rack.

--remove-client-ip-ranges *<low-ip-address>-<high-ip-address>*...

Removes a specified IP address range of external Hadoop compute clients from the virtual rack. You can only remove an entire range; you cannot delete a subset of a range.

--clear-client-ip-ranges

Removes all IP address ranges of external Hadoop compute clients from the virtual rack.

--ip-pools *<subnet>*:*<pool>*...

Assigns pools of OneFS node IP addresses to the virtual rack. The value assigned through this option overwrites any existing IP address pools. You can add a new pool through the `--add-ip-pools` option.

--add-ip-pools *<subnet>*:*<pool>*...

Adds a specified pool of OneFS cluster IP addresses to the virtual rack.

--remove-ip-pools *<subnet>*:*<pool>*...

Removes a specified pool of OneFS cluster IP addresses from the virtual rack.

--clear-ip-pools

Removes all pools of OneFS cluster IP addresses from the virtual rack.

--zone *<string>*

Specifies the access zone that contains the virtual rack you want to modify.

{**--verbose** | **-v**}

Displays more detailed information.

isi hdfs racks delete

Deletes a virtual HDFS rack.

Syntax

```
isi hdfs racks delete <rack-name>
  [--zone <string>]
  [--force]
  [--verbose]
```

Options

<rack-name>

Deletes the specified virtual HDFS rack. Each rack name begins with a forward slash—for example, **/example-name**.

--zone *<string>*

Specifies the access zone that contains the virtual rack you want to delete.

{**--force** | **-f**}

Suppresses command-line prompts and messages.

{**--verbose** | **-v**}

Displays more detailed information.

isi hdfs racks list

Lists the HDFS racks in an access zone.

Syntax

```
isi hdfs racks list
  [--zone <string>]
  [--format {table | json | csv | list}]
  [--no-header]
  [--no-footer]
  [--verbose]
```

Options

--zone *<string>*

Specifies the access zone. The system displays all virtual racks in the specified zone.

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

Display HDFS racks in table, JSON, CSV, or list format.

{**--no-header** | **-a**}

Do not display headers in CSV or table output format.

{**--no-footer** | **-z**}

Do not display table summary footer information.

{**--verbose** | **-v**}

Displays more detailed information.

isi hdfs racks view

Displays information for a specific virtual HDFS rack.

Syntax

```
isi hdfs racks view <rack-name>
  [--zone <string>]
```

Options

<rack-name>

Specifies the name of the virtual HDFS rack to view. Each rack name begins with a forward slash—for example, **/example-name**.

--zone *<string>*

Specifies the access zone that contains the virtual rack you want to view.

isi hdfs ranger-plugin settings modify

Modify Apache Ranger plug-in settings for HDFS.

Syntax

```
isi hdfs ranger-plugin settings modify
  [--enabled <boolean>]
  [--policy-manager-url <string>]
  [--repository-name <string>]
  [--zone <string>]
  [--verbose]
```

Options

--enabled *<boolean>*

Enable the HDFS Ranger plug-in.

--policy-manager-url *<string>*

The scheme, host name, and port of the Apache Ranger server. For example:

`http://ranger.com:6080` or:

`https://ranger.com/6182`

--repository-name *<string>*

The HDFS repository name hosted on the Apache Ranger server.

--zone *<string>*

The access zone containing the HDFS repository.

{--verbose | -v}

Display more detailed information.

isi hdfs ranger-plugin settings view

View Apache Ranger plug-in settings for HDFS.

Syntax

```
isi hdfs ranger-plugin settings view
  [--zone <string>]
```

Options

--zone *<string>*

The access zone containing the HDFS repository.

isi hdfs settings modify

Modifies the HDFS settings for an access zone.

Syntax

```
isi hdfs settings modify
  [--service {yes | no}]
  [--default-block-size <size>]
  [--default-checksum-type {none | crc32 | crc32c}]
  [--authentication-mode {simple_only | kerberos_only}]
  [--root-directory <path>]
  [--webhdfs-enabled {yes | no]
  [--ambari-server <string>]
  [--ambari-namenode <string>]
  [--ambari-metrics-collector <string>]
  [--odp-version <string>]
  [--data-transfer-cipher {none | aes_128_ctr | aes_192_ctr | aes_256_ctr}]
  [--zone <string>]
  [--verbose]
```

Options

--service {yes | no}

Enables or disables the HDFS service in the specified access zone. The HDFS service is enabled by default.

--default-block-size *<size>*

The block size (in bytes) reported by the HDFS service. K, M, and G; for example, 64M, 512K, 1G, are valid suffixes. The default value is 128 MB.

--default-checksum-type {none | crc32 | crc32c}

The checksum type reported by the HDFS service. The default value is `none`

--authentication-mode {simple_only | kerberos_only}

The authentication method used for HDFS connections through the specified access zone. The default value for authentication-mode is `simple_only` for OneFS 8.2.1 and later versions.

--root-directory *<path>*

Root path that contains HDFS data in the access zone that can be accessed by Hadoop compute client connections. The root directory must be within the access zone base directory.

--webhdfs-enabled {yes | no}

Enables or disables the WebHDFS in the specified access zone. WebHDFS is enabled by default.

--ambari-server *<string>*

The Ambari server that receives communication from an Ambari agent. The value must be a resolvable hostname, FQDN, IPv4 or IPv6 address.

--ambari-namenode *<string>*

A point of contact in the access zone that Hadoop services managed through the Ambari interface should connect through. The value must be a resolvable IPv4 address or a SmartConnect zone name.

--ambari-metrics-collector *<string>*

The host name for the metrics collector. The value must be a resolvable hostname, FQDN, IPv4 or IPv6 address.

--odp-version *<string>*

The version of the Open Data Platform (ODP) stack repository, including build number if one exists, installed by the Ambari server. This is required to support ODP upgrades on other systems that are part of the Hadoop cluster.

--data-transfer-cipher {none | aes_128_ctr | aes_192_ctr | aes_256_ctr}

The Advanced Encryption Standard (AES) cipher to use for wire encryption.

--zone *<string>*

The access zone to which the HDFS settings apply.

{--verbose | -v}

Display more detailed information.

isi hdfs settings view

Displays the HDFS settings in an access zone.

Syntax

```
isi hdfs settings view
  [--zone <string>]
```

Options

--zone *<string>*

Specifies the access zone. The system will display the HDFS settings for the specified zone.

5

Additional resources

This chapter includes information about configuring third-party HDFS components like Ambari. Links to additional content resources about how to implement Hadoop on a PowerScale cluster are also provided.

Topics:

HDFS components

HDFS components include the Ambari Management Pack for OneFS and third-party components such as Ambari and Cloudera Manager.

Ambari

The Ambari components you use depend on your version of Ambari and the Hortonworks Data Platform (HDP).

The Ambari agent client and server framework applies through Ambari 2.6. The Ambari Management Pack for OneFS OneFS applies to Ambari 2.7.1.0 with HDP 3.0.1.0 and later on OneFS 8.1.2 and later.

Ambari agent

The Apache Ambari client and server framework, as part of the Hortonworks Data Platform (HDP), is an optional third-party tool that enables you to configure, manage, and monitor a Hadoop cluster through a browser-based interface. This section applies only to the OneFS Ambari agent through Ambari 2.6.

The OneFS Ambari agent is configured per access zone. You can configure the Ambari agent in any access zone that contains HDFS data. To start the Ambari agent in an access zone, you must specify the IPv4 address of the external Ambari server and the address of a NameNode. The NameNode acts as the point of contact for the access zone.

The Apache Ambari server receives communications from the Ambari agent. Once the Ambari agent is assigned to the access zone, it registers with the Ambari server. The agent then provides heartbeat status to the server. The Ambari server must be a resolvable hostname, FQDN, or IPv4 address and must be assigned to an access zone.

The NameNode is the designated point of contact in an access zone that Hadoop services manage through the Ambari interface. For example, if you manage services such as YARN or Oozie through the Ambari agent, the services connect to the access zone through the specified NameNode. The Ambari agent communicates the location of the designated NameNode to the Ambari server and to the Ambari agent. If you change the designated NameNode address, the Ambari agent updates the Ambari server. The NameNode must be a valid SmartConnect zone name or an IP address from the IP address pool that is associated with the access zone.

> **NOTE:** The specified NameNode value maps to the NameNode, secondary NameNode, and DataNode components on the OneFS Ambari agent.

The OneFS Ambari agent is based on the Apache Ambari framework and is compatible with multiple Ambari server versions. For a complete list of supported versions, see the Supported Hadoop Distributions and Products page on the EMC Community Network (ECN).

Configuring Ambari agent settings

You can configure Ambari agent support in each access zone that contains HDFS data using either the OneFS web administration interface or the command-line interface.

Configure Ambari agent settings (Web UI)

1. Click **Protocols** > **Hadoop (HDFS)** > **Settings**.
2. From the **Current Access Zone** list, select the access zone in which you want to enable Ambari server settings.
3. From the **Ambari Server Settings** area, in the **Ambari Server** field, type the name of the external Ambari server that communicates with the Ambari agent.

   The value must be a resolvable hostname, FQDN, IPv4, or IPv6 address.
4. In the **Ambari NameNode** field, designate the SmartConnect FQDN or IP address of the access zone where the HDFS data resides on the cluster.

   The IP address must belong to an IP address pool that shares access zone. IPv6 addresses are not supported.
5. In the **ODP Version** field, specify the version of the Open Data Platform (ODP) stack repository, including build number if one exists, installed by the Ambari server.

   The ODP version is required to support ODP upgrades on other systems that are part of the Hadoop cluster.
6. Click **Save Changes**.

Configure Ambari agent settings (CLI)

Run the `isi hdfs settings modify` command.
The following command specifies company.ambari.server.com as the external Ambari server that receives communication from the Ambari agent running in the zone3 access zone, and designates 192.168.205.5 as the point of contact in the zone3 access zone for Hadoop services that are managed through the Ambari interface.

```
isi hdfs settings modify \
--ambari-server=company.ambari.server.com \
--ambari-namenode=192.168.205.5  \
--zone=zone3
```

Ambari Management Pack for OneFS

The Ambari Management Pack for OneFS enables deploying Ambari 2.7 or later with Hadoop 3.0 or later on OneFS 8.1.2 or later.

You use the Ambari Management Pack for OneFS (the Management Pack) to add OneFS artifacts to Ambari using an installation wizard.

After you choose the storage type (HDFS or OneFS), the installation wizard gathers information to identify the OneFS hosts and automatically assigns components such as the NameNode and DataNode.

Download the Management pack installation bundle from the product download page. For installation details, see the *Dell EMC OneFS with Hadoop and HortonWorks Installation Guide* on the OneFS 8.1.2 Documentation - Isilon Info Hub.

Ambari metrics and alerts

In a Hadoop deployment with OneFS 8.1.2.0 or later releases, a node in a PowerScale cluster can monitor, collect, and push metrics data at 1 minute intervals to the Ambari Metrics Collector, which is one of the components of the Ambari Metrics System from Hortonworks.

All of the OneFS metrics and alert data that are provided to Ambari are cluster-wide. For example, for a three-node PowerScale cluster, the network NDFS traffic aggregated across all three nodes is reported to Ambari. **Note**: OneFS metrics for specific access zones that contain HDFS data sets is not currently supported.

The following command designates 192.168.205.5 as the point of contact in the zone3 access zone for Hadoop services that are managed through the Ambari interface.

```
isi hdfs settings modify  \
--ambari-namenode=192.168.205.5 \
--ambari-metrics-collector http://ambari-metrics-collector-host.com  \
--zone=zone3
```

To specify the name of the external Ambari host where the Ambari Metrics Collector component is installed using the Web UI:

1. Click **Protocols** > **Hadoop (HDFS)** > **Settings**
2. In the Ambari Metrics Collector field, specify the name of the external Ambari host where the Ambari Metrics Collector component is installed.

The value must be a resolvable hostname, FQDN, IPv4, or IPv6 address.

3. **Save Changes**

To view the Ambari metrics, follow the steps that are outlined in the *Dell EMC Isilon OneFS with Hadoop and HortonWorks Installation Guide* on the OneFS 8.1.2 Documentation - Isilon Info Hub.

Cloudera

Cloudera Navigator

OneFS provides support for Cloudera's Navigator application with the release of OneFS 8.1.2 and later versions.

OneFS supports the following data management tasks in Cloudera Navigator:

- **Browse and search data**: Find the owner, creation and modification dates, understand data origins, and history.
- **Lineage and provenance**: Track data from its source and monitor downstream dependencies.
- **Discovery and exploration**: Add, review, and update metadata on the objects contained in the Hadoop data store.
- **Custom metadata and tagging**: Add custom tags and information to data and objects in HDFS.

The Cloudera Navigator Data Management component is a comprehensive data governance and stewardship tool available to supplement Cloudera's distribution including Apache Hadoop (CDH). Navigator recognizes HDFS, Yarn, Impala, and Hive as sources of data that it can manage. It extracts information from these services to provide additional insight into how data was created and managed—and when and by whom it was changed—using metadata and job history along with HDFS data feeds.

The primary use of Navigator is data governance to monitor and track data in a HDFS workflow. One of the unique challenges with very large data sets is being able to track and monitor how data moves through the data analytics workflow. A key Navigator feature is the ability to link between input and output data through analytics jobs like Mapred, or perform data transformations on table-based data in Hive or Impala databases. Navigator then analyzes the metadata and job history and links it together to generate lineage.

Traditional HDFS metadata management

In a traditional Direct Attached Storage (DAS) Hadoop with NameNode (NN) deployment of HDFS, the NameNode's main role is to store all the metadata of the underlying data blocks: the HDFS namespace, directory structures, file permissions, and block IDs to files. While this data is held in memory for operational use, it is critical that this data is persisted to disk for recovery and fault tolerance.

In traditional HDFS, this metadata is stored in two ways:

- FSImage (a binary image file accessed through an HTTP end point)
- INotify stream (an ordered JSON edit log retrieved through HDFS RPCs)

The FSImage image file is a complete point-in-time representation of the HDFS file system metadata. The FSImage file is used on NameNode startup to load the metadata into memory. Because it is inefficient at handling incremental updates, all of the modifications to the HDFS file system are recorded in a transaction log (INotify stream) rather than frequently rewriting the FSImage file. This provides the NameNode with a number of capabilities, and modifications can be tracked without having to constantly regenerate the FSImage file. In the event of a NameNode restart, the combination of the latest FSImage and INotify log can be used to provide an accurate view of the file system at any point in time.

Eventually the HDFS cluster will need to construct a new FSImage that encompasses all INotify log file entries consolidated with the old FSImage directly into a new updated FSImage file to provide an updated point-in-time representation of the file system. This is known as *checkpointing* and is a resource-intensive operation. During checkpointing, the NameNode has to restrict user access to the system, so instead of restricting access to the active NameNode, HDFS offloads this operation to the Secondary NameNode (SN)—or to a standby NameNode—when operating in high availability (HA) mode. The secondary NameNode handles the merge of existing FSImage and INotify transaction logs and generates a new complete FSImage for the NameNode. At this time, the latest FSImage can be used in conjunction with the new INotify log files to provide the current file system. It is important that the checkpoints occur, otherwise on a NameNode restart, it has to construct the entire HDFS metadata from the available FSImage and all INotify logs. This can take a significant amount of time, and the HDFS file system will be unavailable while this occurs.

Cloudera Navigator metadata management

The Navigator metadata service accesses data in a number of ways, such as Yarn application logs, Hive and Impala applications, and HDFS metadata through polling of the FSImage file and INotify transaction logs. It collects all of this information and stores it within Apache Solr databases on the Hadoop cluster. Navigator then runs additional extractions and analytics to create the data that you can view in Navigator. The ability to collect the underlying HDFS metadata from FSImage and INotify is critical to

Navigator's ability to view the file system and is why, up until the release of OneFS 8.1.1, OneFS Hadoop clusters were unable to provide HDFS file system data to Navigator.

Navigator's primary function is to read an initial FSImage and then use the INotify logs to gain access to all file system updates that have occurred. It is possible under specific situations that Navigator is required to refresh its data from a full FSImage rather than leveraging the INotify log, but this does not occur normally.

It is important to recognize that Navigator data is not real-time; it periodically updates the data through polling and extraction to create the data reviews. This behavior is consistent with both DAS and OneFS deployments and is how Cloudera Navigator is designed to operate.

OneFS support for Cloudera Navigator

The OneFS approach to handling file system allocation, block location, and metadata management is fundamentally different than how a traditional Apache-based HDFS file system manages its data and metadata. When OneFS is integrated into a Hadoop cluster, it provides the storage file system to the Hadoop cluster that is based on OneFS and not on an HDFS-based file system. Its layout and protection scheme is fundamentally different than HDFS, and so is its management of metadata and blocks. Since OneFS is not a NameNode-based HDFS file system—and no NameNode is present in the Hadoop cluster—the OneFS file system presents NameNode and DataNode-like functionality to the remote Hadoop cluster through the HDFS service. OneFS doesn't rely on FSImage and INotify transaction log-based metadata management within OneFS with HDFS data. In order to support the native OneFS capabilities, enterprise features for Hadoop, and provide multiprotocol access, OneFS uses the underlying file system presented to the HDFS protocol for Hadoop access. Therefore, prior to OneFS 8.1.1, OneFS could not provide an FSImage and INotify log for consumption.

With the release of OneFS 8.1.1 and later versions, OneFS integrates with Cloudera Navigator by enabling an FSImage and INotify log file on OneFS in an HDFS access zone. By enabling an HDFS Hadoop access zone root for FSImage and INotiffy integration, you are, in effect, telling OneFS to create an FSImage file and start tracking HDFS file system events in an INotify log file, thereby making that data available for consumption by Navigator. Once enabled, OneFS effectively begins to mimic the behavior of a traditional NameNode deployment, and an FSImage file is generated by OneFS. All HDFS file system operations are logged into an INotify stream.

Periodically OneFS will regenerate a new FSImage, but this operation is not true checkpointing or merging of the INotify log as performed on an HDFS NameNode, because the actual file system and operations are still handled by the core OneFS file system. The FSImage and INotify logs are generated by OneFS to provide the required data to Cloudera Navigator in the required format.

The FSImage regeneration job runs daily to recreate a current FSImage which—combined with the current INotify logs—will represent the current state of data and metadata in the HDFS root from an HDFS perspective.

OneFS is a multi-protocol file system, which provides unified access to its data through many protocols, including HDFS, NFS, SMB, and others. Since only HDFS file system operations are captured by the INotify log, Navigator will only initially see this metadata; any metadata created in the HDFS data directories by NFS or SMB will not get included in the INotify stream. However, on regeneration of an FSImage, these files will be included in the current FSImage, and Navigator will see them the next time it uses a later refreshed FSImage. Since Navigator's primary method of obtaining updated metadata is based on INotify logs, it may take some time before non-HDFS-originating data is included. This is expected behavior and should be taken into account if multiprotocol workflows are in use.

Using Navigator with OneFS

In order to enable Navigator integration, both FSImage and INotify need to be enabled on the HDFS access zone within OneFS. Once enabled, they should not be disabled unless the use of Navigator is to be permanently discontinued.

You should not enable FSImage and INotify on any zones that do not use Navigator, as these add unnecessary overhead. Within OneFS, the FSImage and INotify features are access zone-aware and should only be enabled on any Hadoop-enabled access zone that will use Navigator. There is no reason to enable it on a zone that is not being monitored by Navigator, since it will add overhead to that cluster due to a feature that is not being consumed.

No additional configuration changes are required within Cloudera Manager or Navigator to enable integration. When integration is initially enabled, it will take some time for the initial HDFS data to become visible within Navigator and additional time is needed to generate linkage. As new data is added, it will show up in Navigator and will be linked based on the polling and extraction period within Navigator.

Additionally, note the following:

- You can enable FSImage and INotify either through the command line interface or through the web administration interface.
- Once FSImage and INotify are enabled, you must deploy CDH 5.12 with Cloudera Navigator. Cloudera deployments prior to CDH 5.12 will not allow Navigator installation.
- Wait approximately an hour until Navigator has gathered information from applications.
- Clusters will need to be sized to accommodate the performance impact of INotify.
- Events are logged to the `/var/log/hdfs.log` file and messages.

- You should avoid disabling INotify—or toggling INotify and FSImage off and on—as these are destructive actions in Cloudera Navigator and can cause metadata data loss.
- Do not set the FSImage *generation interval* (the interface between successive FSImages) beyond the INotify *retention period* (the minimum duration edit logs will be retained). The INotify minimum retention period must be longer than the FSImage generation interval.
- With INotify enabled there is an expected performance impact for all edit actions over HDFS.
- FSimage generation takes approximately one hour for every three million files.
- To view the data in Navigator, use Yarn, Hive, or another application.
- OneFS 8.1.1 and later releases do not support Cloudera Navigator data audit capabilities.

See HDFS commands for a list of the HDFS commands available from the OneFS command line interface.

For more information about Cloudera Navigator, see:

- OneFS and Cloudera Navigator support
- Cloudera Navigator documentation hub
- Cloudera Navigator data management

Apache Ranger support

OneFS supports Apache Ranger as part of a Hadoop deployment with a PowerScale cluster.

The Apache Ranger console provides a centralized security framework to manage access control over Hadoop data access components such as Apache Hive and Apache HBase. These policies can be set for both individual users or groups and then enforced consistently on files, folders, and databases.

Only Ranger's HDFS authorization policies with Deny conditions are supported by OneFS. Documentation for Apache JIRA RANGER-606 describes how to use Deny conditions, which were added to Apache Ranger 0.6.0.

For more information on Apache Ranger and specific HDP components, refer to the Apache Ranger pages on the Hortonworks site.

- AD, Kerberos, and local authentication are supported.
- One-way SSL with Ranger policy server is supported with MIT KDC and Active Directory (AD).
- Apache Ranger audit of HDFS access is not currently supported.
- Tag policies are not currently supported.

Editing Apache Ranger HDFS plugin settings

You can enable the Apache Ranger HDFS plugin to allow additional oversight of HDFS protocol authentication using either the OneFS web administration interface or the command-line interface (CLI).

You can enable Apache Ranger on PowerScale clusters and then check for new authorization policies, receive HDFS requests from clients, and apply authorization policies to the HDFS requests, which can be one of DENY, ALLOW, or UNDETERMINED. Enable the Apache Ranger HDFS plugin using the steps that are outlined in the Hortonworks Security Guide.

Enabling the Apache Ranger plugin allows the authorization policies that are defined in the Ranger HDFS service instance, also called a repository, prior to Apache Ranger 0.6.0. The policies must first allow users or groups access to resources and then deny specific users or groups from access. If a user is not included in the allow list, they are denied access by default. For more information about creating a DENY policy, see Apache Ranger deny policies with OneFS 8.0.1.0

NOTE: A poorly formed policy can have an unintended impact, for example, blocking access.

The repository name is a setting within Apache Ranger. The minimum supported version of Apache Ranger is 0.6.0 because the Ranger DENY policy is supported only in 0.6.0 and later versions. In version 0.6.0, Apache Ranger changed the name of this feature to service instance. The service instance is the name of the HDFS service instance within the Apache Ranger Admin UI used as the repository name.

If you have a Kerberos-enabled cluster, follow the instructions in the Hortonworks Security Guide to enable the Ranger HDFS plugin on the cluster.

Edit Apache Ranger HDFS plugin settings (Web UI)

The policy manager URL is found on the Ambari server at **Ambari** > **Ranger** > **Configs** as the **policymgr_external_url**. This URL is created by combining `http://`, followed by the host name where Ranger Admin is installed, followed by the *ranger.service.http.port*, which is usually 6080, followed by /

1. Click **Protocols** > **Hadoop (HDFS)** > **Ranger Plugin Settings**.
2. In the **Ranger Plugin settings** area, select **Enable Ranger Plugin**
3. In the **Policy manager URL** field, type the URL that points to the location of the Policy Manager.
4. In the **Repository name** field, type the name of the HDFS repository.
5. Click **Save Changes**.

Edit Apache Ranger HDFS plugin settings (CLI)

The policy manager URL is found on the Ambari server at **Ambari** > **Ranger** > **Configs** as the **policymgr_external_url**. This URL is created by combining `http://`, followed by the hostname where Ranger Admin is installed, followed by the *ranger.service.http.port*, which is usually 6080, followed by /

To configure Ranger plugin settings, run the `isi hdfs ranger-plugin settings modify` command.

The *--policy-manager-url* is created by combining `http://`, followed by the hostname where Ranger Admin is installed, followed by the *ranger.service.http.port*, which is usually 6080, followed by `/`.

The following command configures the Ranger plugin settings.

```
isi hdfs ranger-plugin settings modify --policy-manager-url http://resolvable_name:6080/
--repository-name repository_name --enabled true --zone zone_name
```

Using Hadoop with PowerScale

In addition to this HDFS Reference Guide (this guide), use the following resources to implement your OneFS and HDFS system integration.

Compatibility information

- Hadoop Distributions and Products Supported by OneFS

Information specific to PowerScale

- Using Hadoop with Powerscale OneFS - Powerscale Info Hub
- Overview of Powerscale OneFS and Hadoop (video)
- Hadoop Distributions and Products Supported by Powerscale OneFS
- Prepare Powerscale OneFS for Hadoop Cheat Sheet
- Powerscale OneFS and Hadoop known issues
- Powerscale OneFS CLI Commands for Managing Hadoop Integrations
- Powerscale Permissions Model and Adding HDFS
- Powerscale OneFS with Hadoop and Overlapping HDFS Directories
- Hadoop Tiered Storage with Powerscale and ECS
- HDFS Tiering with Powerscale and ECS
- Using MariaDB and MySQL for Ambari and and Ranger KMS on Powerscale OneFS with Hadoop
- Running Spark on Powerscale OneFS with Hadoop
- Increasing Hadoop Resiliency and Operational Efficiency with Powerscale OneFS
- Using HDFS TDE with Powerscale OneFS
- Powerscale OneFS and Hadoop Local UID Parity
- Getting Powerscale OneFS - Hadoop UID/GID parity
- Powerscale OneFS and Hadoop Proxy Users

- Considerations for Active Directory based on Kerberos with Hadoop
- RFC2307 and Active Directory Identity Management with Powerscale
- Backing Up Hadoop To Powerscale OneFS
- Troubleshooting a Permissions Issue between Hadoop and Powerscale OneFS
- Creating a Bi-Directional HDFS Mirror Across HDP and Powerscale Clusters with Falcon
- Powerscale OneFS ACLs with Hadoop
- Change in Default SMB Directory ACLs Setting in Powerscale OneFS 8.0.1
- Powerscale Telemetry for the Hadoop Admin
- Simple LLAP Demo on Powerscale OneFS
- SmartConnect with Network Pools and HDFS Racks - Part 1
- SmartConnect with Network Pools and HDFS Racks - Part 2

Hortonworks and Ambari

- Powerscale OneFS with Hadoop and Hortonworks Installation Guide (PDF)
- Powerscale OneFS 8.2 Certification with HDP 3.1
- Upgrade Hortonworks HDP 2.6.5 to HDP 3.0.1
- Automating Creation of Valid SAMAccount Names with Ambari for Powerscale OneFS
- Configuring Ambari Hive View with Powerscale OneFS
- Apache Ranger deny policies with OneFS 8.0.1.0
- Ambari Metrics and Alerts with Powerscale OneFS
- Enhanced Hadoop Security with OneFS 8.0.1 and Hortonworks HDP
- Ever better HDP upgrades with Powerscale OneFS
- Powerscale OneFS, Ambari, and Accumulo Tracer
- Powerscale OneFS and Ambari 2.5 with HDP 2.6
- Powerscale OneFS and Ambari 2.4 with HDP 2.5

Hortonworks and Ambari with Kerberos

- Powerscale OneFS with Ambari Multitenant Active Directory Integration Guide (PDF)
- Powerscale OneFS with Hadoop and Hortonworks Kerberos Installation Guide (PDF)
- Ambari Automated Kerberos Configuration with Powerscale OneFS
- Ambari HDP with Powerscale OneFS 8.0.0.1 and Active Directory Kerberos Implementation
- Duplicate SPNs with Powerscale AD Kerberos and Hortonworks prevent services from starting
- KDC Kerberized Yarn Service Fail to Start on 8.0.1 with Ambari using WebHDFS curl calls
- The infamous '401 Authorization Required' error when starting Kerberized services
- Troubleshooting first-tIme services start with Kerberized services
- HDP and Kerberos Authentication Troubleshooting
- Troubleshooting checklist - Hadoop Kerberization and services

Cloudera

- Powerscale OneFS with Hadoop and Cloudera Installation Guide (PDF)
- Powerscale OneFS and Cloudera Navigator Support
- Powerscale OneFS 8.1.1 and Cloudera 5.13+ for Cloudera Navigator
- Cloudera and Powerscale OneFS Implementation - Part 1
- Cloudera and Powerscale OneFS Implementation - Part 2
- Powerscale OneFS and Cloudera Backup and Disaster Recovery Integration - Part1
- Powerscale OneFS and Cloudera Backup and Disaster Recovery Integration - Part 2
- Get Cloudera 5.7 Impala starting with Powerscale OneFS
- CDH 5.14.2 DFSIO Testing with Powerscale F800

Cloudera with Kerberos

- Powerscale OneFS with Hadoop and Cloudera Kerberos Installation Guide (PDF)
- Cloudera 5.7 with Powerscale OneFS 8.0.0.1 and Active Directory Kerberos Implementation
- Getting the Hue Service Started with Kerberized Cloudera with Powerscale OneFS

Known issues and troubleshooting

- Powerscale OneFS and Hadoop known issues
- Troubleshooting checklist - Hadoop Kerberization and services
- OneFS Customer Troubleshooting Guides Info Hub
- Customer Troubleshooting Guide - HDFS and Ambari (PDF)
- Customer Troubleshooting Guide - HDFS and Ambari with Kerberos (PDF)
- Customer Troubleshooting Guide - HDFS and Cloudera (PDF)
- Customer Troubleshooting Guide - HDFS and Cloudera with Kerberos (PDF)