EMC Isilon OneFS with Cloudera Hadoop Installation Guide

version 1.03

Abstract

This guide walks you through the process of installing EMC Isilon OneFS with the Cloudera distribution of Hadoop.
Publication History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.01</td>
<td>January 27, 2017</td>
<td>Initial version</td>
</tr>
<tr>
<td>1.02</td>
<td>July 31, 2017</td>
<td>- Updated list of users that the Isilon_hadoop_tools script creates.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Removed instructions to modify the block size that is used for reading from Isilon since the block size is 128M by default.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Removed the “—fixperm” flag when using the Isilon_create_users.sh script. That flag is intended for user on existing deployments where directory ownership is wrong on OneFS.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Updated various screen shots and the troubleshooting section.</td>
</tr>
<tr>
<td>1.03</td>
<td>October 5, 2017</td>
<td>- Added “Updates and Additional Information on Isilon Hadoop Installs” section.</td>
</tr>
</tbody>
</table>

EMC², EMC, Isilon, OneFS, and the EMC logo are registered trademarks in the United States and other countries. All other trademarks used herein are the property of their respective owners.

© Copyright 2017 EMC. All rights reserved. Published in the USA.

EMC believes the information in this document is accurate as of its publication date. The information is subject to change without notice.
## CONTENTS

Introduction ........................................................................................................................... 5
Audience ................................................................................................................................. 5
Overview ................................................................................................................................. 5
Updates and additional information on Isilon Hadoop installs .............................................. 7
Prerequisites .......................................................................................................................... 7
Install Isilon OneFS with Cloudera Manager ........................................................................ 8
Prepare OneFS ....................................................................................................................... 8
  Validate OneFS version and license activation ................................................................. 8
  Configure Isilon OneFS components ................................................................................ 9
  Create an Access Zone ........................................................................................................ 9
  Configure SmartConnect ................................................................................................. 10
  Configure DNS for Isilon .................................................................................................... 11
  Verify the SmartConnect configuration ........................................................................... 12
  Create and configure Isilon HDFS root ........................................................................... 12
  Modify the access control list (ACL) settings for OneFS .................................................. 13
  Create HDFS users and groups ....................................................................................... 13
  Create users and directories on the OneFS cluster using Isilon scripts ......................... 14
  Create users on the OneFS cluster manually .................................................................. 17
Prepare Cloudera .................................................................................................................. 17
  Deploy Cloudera Manager ............................................................................................... 18
  Configure the Hadoop cluster ......................................................................................... 22
Troubleshoot Cloudera Manager ........................................................................................ 30
Verify the deployment .......................................................................................................... 32
Contacting Isilon Technical Support ................................................................................... 34
Introduction

Hadoop is an open-source framework that enables the distributed processing of large data sets across clusters of computers. You can follow the steps in this guide to install Isilon OneFS with Hadoop for use with Cloudera.

Before you begin, you must install an Isilon OneFS cluster.

Audience

This guide is intended for systems administrators, IT program managers, IT architects, and IT managers who will be installing Isilon OneFS in conjunction with a Cloudera distribution of Hadoop.

Overview

The Isilon OneFS scale-out network-attached storage (NAS) platform provides Hadoop clients with direct access to Big Data through a Hadoop Distributed File System (HDFS) protocol interface. An Isilon cluster powered by the OneFS operating system delivers a scalable pool of storage with a global namespace.

Hadoop compute clients can access the data that is stored on an Isilon cluster by connecting to any node over the HDFS protocol. All nodes configured for HDFS provide NameNode and DataNode functionality. Each node boosts performance and expands the cluster’s capacity. For Hadoop analytics, the Isilon scale-out distributed architecture minimizes bottlenecks, rapidly serves Big Data, and optimizes performance for MapReduce jobs.

In a traditional Hadoop deployment, the Hadoop compute nodes run analytics jobs against large sets of data. A NameNode directs the compute nodes to the data stored on a series of DataNodes. The NameNode is a separate server that holds metadata for every file that is stored on the DataNodes. Often data is stored in production environments and then copied to a landing zone server to be loaded on to HDFS. This process is network intensive and exposes the NameNode as a potential single point of failure.
In an Isilon with Hadoop deployment, Isilon OneFS serves as the file system for Hadoop compute clients. On a OneFS cluster, every node in the cluster acts as a NameNode and DataNode, providing automated failover protection.

When a Hadoop client runs a job, the clients accesses the data that is stored on a OneFS cluster by connecting over HDFS. The HDFS protocol is native to the Isilon OneFS operating system, and no data migration is required.

The Cloudera distribution is stored on a separate compute cluster, and individual clients connect directly to the Isilon cluster to store and access Hadoop data. OneFS handles HDFS file data exchange as a protocol in order to store and retrieve the data to match the requirements of the client.
Updates and additional information on Isilon Hadoop installs

The rapid release of new features and versions of Hadoop projects can introduce new behaviors and requirements. We recommend that you review the latest updates on the Using Hadoop with Isilon - Isilon Info Hub for updates and known issues while deploying Isilon and Hadoop.

Prerequisites

For supported versions, see the Hadoop Distributions and Products Supported by OneFS compatibility matrix.

Cloudera Distribution with Apache Hadoop (CDH)

Ensure that the following requirements are met:

- CDH 5 parcel: 5.7.1-1.cdh5.7.1.p0.11 or later.
- Familiarity with the Cloudera documentation and the installation instructions.
  - To view the Cloudera documents, go to http://www.cloudera.com/documentation.html
  - Use the following table to record the components that you plan to install.

<table>
<thead>
<tr>
<th>Component</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloudera Manager version</td>
<td></td>
</tr>
<tr>
<td>CDH parcel version</td>
<td></td>
</tr>
<tr>
<td>Cloudera server (FQDN)</td>
<td></td>
</tr>
</tbody>
</table>

Isilon OneFS cluster configuration

Ensure that the following requirements are met:

- An Isilon OneFS cluster running OneFS 8.0.0.1 or greater. OneFS 8.0.0.1 contains a number of updates to facilitate the integration and deployment of Hadoop. It is strongly recommended to use OneFS 8.0.0.1 or greater.
- SmartConnect Advanced, a separately licensed Isilon module, is activated and SmartConnect is configured on your Isilon OneFS cluster.
- HDFS, a separately licensed Isilon module, is activated on your Isilon OneFS cluster.
- Valid Isilon SmartConnect SSIP and Domain Name System (DNS) delegation is in place to provide name resolution services for a SmartConnect zone. For more information, see Isilon External Network Connectivity Guide—Routing, Network Topologies, and Best Practices for SmartConnect.
- A dedicated Isilon Access Zone is in use. This is not the same as the System Zone.
- An Isilon HDFS root directory in the Access Zone.
- A simple access model between Hadoop and Isilon; UID and GUID, with parity.

- Use the following table to record the components that you have installed.

| Component               | Version/License |
Install Isilon OneFS with Cloudera Manager

The installation of Isilon OneFS with Cloudera can be separated into four stages as represented in the following illustration.

To complete each stage, you must perform tasks on both the Cloudera cluster and the Isilon OneFS cluster.

<table>
<thead>
<tr>
<th>Component</th>
<th>Version/License</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDFS module</td>
<td></td>
</tr>
<tr>
<td>OneFS cluster name</td>
<td></td>
</tr>
</tbody>
</table>

Prepare OneFS

Complete the following steps to configure your Isilon OneFS cluster for use with Cloudera Data Platform. Preparing OneFS requires you to configure DNS, SmartConnect, and Access Zones to allow for the Hadoop cluster to connect to the OneFS cluster. If these preparation steps are not successful, the subsequent configuration steps might fail.

Validate OneFS version and license activation

You must validate your OneFS version, check your licenses, and confirm that they are activated.

1. From a node in your OneFS cluster, confirm that your OneFS cluster is running OneFS 8.0.0.x by typing the following command:

   ```
   isi version
   ```

   If you are using an older version, the implementation steps in this guide might not work.

2. Confirm that licenses for HDFS and SmartConnect Advanced are operational. If these licenses are not active and valid, some commands in this guide will not work.

   Run the following commands to confirm that HDFS and SmartConnect Advanced are installed:

   ```
   isi license licenses list
   ```
isi license licenses view HDFS
isi license licenses view "SmartConnect Advanced"

3. If your modules are not licensed, obtain a license key from your Isilon sales representative. Type the following command to activate the license:

`isi license activate --key <key>`

Configure Isilon OneFS components
After you configure DNS for Isilon OneFS, set up and configure the following Isilon OneFS components.

- Create an access zone
- Create a SmartConnect zone
- Create and configure the HDFS root in the access zone
- Create users and groups
- Create a basic HDFS folder structure for use with HDFS

Use the following table to record the configuration information for the OneFS cluster HDP Cloudera integration:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access zone name</td>
<td></td>
</tr>
<tr>
<td>Access zone path</td>
<td></td>
</tr>
<tr>
<td>SmartConnect zone name (FQDN)</td>
<td></td>
</tr>
<tr>
<td>IP range for IP pool (ranges)</td>
<td></td>
</tr>
<tr>
<td>SmartConnect pool name (subnet pool)</td>
<td></td>
</tr>
<tr>
<td>Node and interfaces in the pool</td>
<td></td>
</tr>
<tr>
<td>HDFS root path</td>
<td></td>
</tr>
</tbody>
</table>

Create an Access Zone
On one of the Isilon OneFS nodes, you must define an access zone on the OneFS cluster and enable the Hadoop node to connect to it.

1. On a node in the OneFS cluster, you must define an access zone on the OneFS cluster and enable the Hadoop node to connect to it.

   `isi zone zones create --name=zone1-cdh --path=/ifs/zone1/cdh --create-path`

2. Confirm that you have successfully created access zones.

   `isi zone zones list --verbose`
Output similar to the following displays:

```
Name: System
  Path: /ifs
  Groupnet: groupnet0
  Map Untrusted: -
  NetBIOS Name: -
  User Mapping Rules: -
  Home Directory Umask: 0077
  Skeleton Directory: /usr/share/skel
  Cache Entry Expiry: 4H
  Zone ID: 1
```

```
Name: zone1-cdh
  Path: /ifs/zone1/cdh
  Groupnet: groupnet0
  Map Untrusted: -
  Auth Providers: lsa-local-provider:zone1-cdh
  NetBIOS Name: -
  User Mapping Rules: -
  Home Directory Umask: 0077
  Skeleton Directory: /usr/share/skel
  Cache Entry Expiry: 4H
  Zone ID: 2
```

3. Create the HDFS root directory within the access zone that you created.

```
mkdir -p /ifs/zone1/cdh/hadoop-root
```

4. List out the contents of the Hadoop access zone root directory.

```
ls -al /ifs/zone1/cdh
```

5. Modify your existing subnets and specify a service address to determine if the SC IP address already exists.

```
isi network subnets modify groupnet0.subnet0 --sc-service-addr=x.x.x.x
```

**Configure SmartConnect**

On a node in the OneFS cluster, add a static IP address pool and associate it with the access zone you created earlier.

**Steps to perform on the OneFS cluster**

1. To create an access zone, run the following command, where:

   `--name subnet: <poolname>` is the new IP pool in subnet (for example, subnet0:pool1).
--ranges is the IP range that is assigned to the IP pool
--ifaces are the node interfaces that are added to the pool
--access-zone is the access zone that the pool is assigned to
--sc-dns-zone is the SmartConnect zone name
--sc-subnet is the name of the SmartConnect service subnet that is responsible for this zone

isi network pools create --id=groupnet0:subnet0:hadoop-pool-cdh --ranges=x.x.x.x--x.x.x.x --access-zone=zonel-cdh --alloc-method=dynamic --ifaces=X-Y:<your interfaces> --sc-subnet=subnet0 --sc-dns-zone=<isilonsczone-cdh.FQDN> --description=hadoop

2. View the properties of the existing pool.

isi network pools view --id=groupnet0:subnet0:hadoop-pool-cdh

Output similar to the following displays:

ID: groupnet0.subnet0.hadoop-pool-cdh
Groupnet: groupnet0
Subnet: subnet0
    Name: hadoop-pool-cdh
    Rules: -
    Access Zone: zonel-cdh
    Allocation Method: dynamic
    Aggregation Mode: lacp
    SC Suspended Nodes: -
        Description: cdh_hadoop_access_zone
            Ifaces: 1:ext-1, 2:ext-1, 3:ext-1, 4:ext-1
            IP Ranges: x.x.x.x--x.x.x.x
        Rebalance Policy: auto
    SC Auto Unsuspend Delay: 0
    SC Connect Policy: round_robin
        SC Zone: isilonsczone-cdh.FQDN
    SC DNS Zone Aliases: -
    SC Failover Policy: round_robin
        SC Subnet: subnet0
            SC Ttl: 0
        Static Routes: -

Configure DNS for Isilon

Note: Before you begin, the OneFS cluster must be implemented according to Isilon best practices. For more information, see the HDFS Setup section of the Isilon Best Practices for Hadoop Data Storage.

Set up DNS records for a SmartConnect zone. You must create the required DNS records that are used to access your OneFS cluster from the Hadoop cluster. All hosts in your Hadoop cluster must be configured for both forward and reverse DNS lookups Hadoop relies heavily on DNS and performs many DNS lookups during normal operation.
You can set up a SmartConnect zone for the connections from Hadoop compute clients. SmartConnect is a module that specifies how the Isilon OneFS cluster handles connection requests from clients. For additional information and best practices for SmartConnect, see the Isilon External Network Connectivity Guide.

Each SmartConnect zone represents a specific pool of IP addresses. When you associate a SmartConnect zone with an access zone, OneFS allows only clients that connect through the IP addresses in the SmartConnect zone to reach the HDFS data in the access zone. A root HDFS directory is specified for each access zone. This configuration isolates data within access zones and allows you to restrict client access to the data.

A SmartConnect zone distributes NameNode requests from Hadoop compute clients across the node interfaces in the IP pool. Each node's NameNode process will reply with the IP address of the HDFS DataNode where the client can access the data. When a Hadoop compute client makes an initial DNS request to connect to the SmartConnect zone FQDN, the Hadoop client requests are delegated to the SmartConnect Service IP, which responds with a valid node to connect to. The client will then connect to an Isilon node that serves as a NameNode. When a second Hadoop client makes a DNS request to connect to the SmartConnect zone, the SmartConnect Service routes the client connection to a different node than the one used by the previous Hadoop compute client.

When you create a SmartConnect zone, you must add a Name Server (NS) record as a delegated domain to the authoritative DNS zone that contains the OneFS cluster.

For additional information and best practices, see the "DNS delegation best practices" section of the Isilon External Network Connectivity Guide.

**Verify the SmartConnect configuration**

You must validate that SmartConnect is set up correctly by pinging the SmartConnect zone FQDN from the Hadoop client.

**Step to perform on the Hadoop client**

From the Hadoop client, ping the SmartConnect zone several times.

```
ping isilonsczone-cdh.FQDN
```

When you view the output of this command, note that different IP addresses are returned for each ping command, because with each DNS response, the IP addresses are returned through rotating round-robin DNS from the list of potential IP addresses. This validates that the SmartConnect zone name FQDN is operating correctly.

**Create and configure Isilon HDFS root**

On a node in the OneFS cluster, assign the Hadoop Distributed File System (HDFS) root directory.

**Steps to perform on the OneFS cluster**

1. View the HDFS service settings.

```
isihdfssettingsview --zone=zonel-cdh
```

2. Map the HDFS user to Isilon root. Create a user mapping rule to map the HDFS user to the OneFS root account. This mapping enables the services from the Hadoop cluster to communicate with the OneFS cluster using the correct credentials.

```
isizonezonesmodify --user-mapping-rules="hdfs=>root" --zone zonel-cdh
```

3. Set the HDFS block size that is used for reading from Isilon by running the following command:

```
isihdfssettingsmodify --zone=zonel-cdh --default-block-size=128M
```
4. Create an indicator file in the Hadoop directory to view your OneFS cluster and access zone through HDFS.

```bash
touch /ifs/zone1/cdh/hadoop-root/THIS_IS_ISILON_zone1-cdh.txt
```

**Modify the access control list (ACL) settings for OneFS**

**Step to perform on the OneFS cluster**

Run the following command to modify ACL settings BEFORE you create directories or files in the next section. This creates the correct permission behavior on the cluster for HDFS.

**Note:** ACL policies are cluster-wide, so you should understand this change before performing it on production clusters.

```bash
isi auth settings acls modify --group-owner-inheritance=parent
isi auth settings view  < -- validate that it worked
```

**Create HDFS users and groups**

For each Hadoop system account that will submit HDFS jobs or access the file system, you must create local users on the OneFS cluster. You can add Hadoop users to the OneFS cluster manually or by using the script provided at: https://github.com/Isilon/isilon_hadoop_tools

**Note:** The script creates the service accounts that are required for Hadoop.

**Important**

Isilon recommends that you maintain consistent names and numeric IDs for all users and groups on the OneFS cluster and your Hadoop clients. This consistency is important in multi-protocol environments because the HDFS protocol refers to users and groups by name, and NFS refers to users and groups by their numeric IDs (UIDs and GUIDs). Maintaining this parity is critical in the behavior of Isilon OneFS multiprotocol file access.

When installing Hadoop with Cloudera Manager, the installer creates all of the required system accounts on all the clients. For example, a Hadoop system account, `yarn`, is created with the UID of 502 and the GUID of 502 on the Hadoop cluster nodes, Cloudera will only create these accounts if they don't already exist. We can ensure parity by pre-creating them on all nodes to be installed in the Hadoop cluster. We can look to enforce parity by manually managing when and how these local system accounts get created. Since the Hadoop installer cannot create the local accounts directly on Isilon OneFS, they need to be created manually. You must create the Isilon `yarn` local account user in the Isilon access zone in which `yarn` will access data. You need to create a local user `yarn` with the UID of 502 and the GUID of 502 to ensure consistency of access and permissions.

For guidance and more information about maintaining parity between OneFS and Hadoop local users and UIDs, see the following blog post:

https://community.emc.com/community/products/isilon/blog/2016/06/22/isilon-and-hadoop-user-uid-parity

There are many methods of achieving UID and GUID parity. You can leverage the create user and directory scripts, perform manual matching, or create scripts that parse users and create the equivalent users. However you choose to achieve this, the sequence will depend on your deployment methodology and management practices. Isilon highly recommends that you to achieve consistency between the Hadoop cluster and Isilon OneFS, for example, hdfs=hdfs, yarn=yarn, hbase=hbase, and so on from a UID and GUID consistency perspective.
Create users and directories on the OneFS cluster using Isilon scripts

This methodology achieves parity by executing user creation in the following sequence:

1. Create local users and groups on Isilon OneFS.
2. Collect the UIDs and GUIDs of the users.
3. Pre-create local users and groups on all CDH hosts to be deployed.

If the local Hadoop system users and groups already exist on the Linux host, then the Cloudera wizard does not create them. If you created them with the same UIDs and GUIDs as on Isilon OneFS, you will maintain parity.

You must add a user on the OneFS cluster for each user that runs Hadoop services and user that submits jobs, and you must create any additional users that may access the OneFS cluster.

In the following steps, you can run an Isilon script to create local user and group accounts on the OneFS cluster. The Isilon script adds a list of default Hadoop users to the OneFS cluster that are mapped to the services and applications on the Hadoop cluster.

**Note**

If the users and groups must be defined by your directory services, such as Active Directory or LDAP, you must create the users and groups manually as outlined later in the section entitled Creating users manually.

The `isilon_create_users.sh` script creates the following users:

accumulo, anonymous, apache, cmjobuser, cloudera-scm, flume, hbase, hdfs, hive, HTTP, httpfs, hue, impala, kafka, keytrustee, kms, kudu, llama, mapred, oozie, sentry, solr, spark, sqoop, sqoop2, yarn, zookeeper

1. **Perform the following steps on the Isilon OneFS cluster.**
   a. On a node in the OneFS cluster, create a scripts directory. You will extract the scripts to this directory.
   
   ```bash
   mkdir -p /ifs/zone1/cdh/scripts
   ```
   b. Change directories to the scripts directory that you created and change the execute permissions.
   
   ```bash
   cd /ifs/zone1/cdh/scripts
   chmod u+x *
   ```
   c. Clone or download the latest version of the Isilon Hadoop tools from [https://github.com/Isilon/isilon_hadoop_tools](https://github.com/Isilon/isilon_hadoop_tools) as a zip file and unzip the files.
   d. Use WinSCP to transfer the `isilon_create_users.sh` and `isilon_create_directories.sh` script files to the `/ifs/zone1/cdh/scripts` directory or use `wget` to get the scripts directly.
   e. OR if Isilon has direct internet access, run the following command.
   
   ```bash
   wget https://raw.githubusercontent.com/Isilon/isilon_hadoop_tools/master/isilon_create_users.sh --no-check-certificate
   wget https://raw.githubusercontent.com/Isilon/isilon_hadoop_tools/master/isilon_create_directories.sh --no-check-certificate
   ```
   f. To create all required local users and groups on your OneFS cluster for the Hadoop services and applications, run the following two scripts:

**Script usage:**
bash isilon_create_users.sh --dist <DIST> --startgid <GID> --startuid <UID> --zone <ZONE>

Where:

<table>
<thead>
<tr>
<th>dist</th>
<th>This will correspond to your Hadoop distribution - cdh, hwx, phd, phd3, or bi.</th>
</tr>
</thead>
<tbody>
<tr>
<td>startgid</td>
<td>The beginning UID range for the creation of users (the default is 1000).</td>
</tr>
<tr>
<td>startuid</td>
<td>The beginning GUID range for the creation off users (the default is 1000).</td>
</tr>
<tr>
<td>zone</td>
<td>The name of the access zone where the users should be created (useful for multi-tenant environments that will use a single KDC). The System zone is the default.</td>
</tr>
</tbody>
</table>

Example:

bash isilon_create_users.sh --dist cdh --startuid 501 --startgid 501 --zone zonel-cdh

Output similar to the following appears:

Info: Hadoop distribution: cdh
Info: users will start at UID 501
Info: groups will start at GID 501
Info: will put users in zone: zonel-cdh
Info: HDFS root: /ifs/zonel/cdh/hadoop-root
Info: passwd file: zonel-cdh.passwd
Info: group file: zonel-cdh.group

SUCCESS -- Hadoop users created successfully!
Done!

g. Run the isilon_create_directories.sh script. This script creates the required directories to map to the Hadoop users with appropriate ownership and permissions.

Script usage:

bash isilon_create_directories.sh --dist<DIST> --zone<ZONE>

Where:

<table>
<thead>
<tr>
<th>dist</th>
<th>The distribution of Hadoop for which the script should create an NDFS directory skeleton.</th>
</tr>
</thead>
<tbody>
<tr>
<td>zone</td>
<td>The name of the access zone where the directory skeleton should be created.</td>
</tr>
</tbody>
</table>

bash isilon_create_directories.sh --dist cdh --zone zonel-cdh

Output similar to the following displays, which shows the directories that the script creates:

Info: Hadoop distribution: cdh
Info: will use users in zone: zonel-cdh
Info: will fix permissions and owners on existing directories created by this script.
Info: HDFS root dir is /ifs/zonel/cdh/hadoop-root

# Creates Hadoop directory structure on Isilon system HDFS.
h. List the HDFS base directories.

```bash
ls -al /ifs/zone1/cdh/hadoop-root
ls -al /ifs/zone1/cdh/hadoop-root/user
```

2. On the OneFS cluster, collect the created local users and group UID & GID's.

   a. List out and copy the contents of the `zone1-cdh.passwd` and `zone1-cdh.group` files that are created when the user_create script is executed.

   ```bash
cat zone1-cdh.passwd
cat zone1-cdh.group
```

3. Steps to perform on the Hadoop clients

   For each Hadoop client:

   a. Edit the `/etc/passwd` file on the Hadoop client and cut and paste the contents of the `zone1-cdh.passwd` file from Isilon to the end of the existing HDFS password file.

   b. Edit the `/etc/group` file on the Hadoop client and cut and paste the contents of the `zone1-cdh.group` file from Isilon to the end of the existing HDFS group file.
We have now created all the relevant users and groups on Isilon and all our Hadoop clients with uid/gid parity. As the local accounts exist on the Hadoop clients the Cloudera installer will not create them but leverage the existed ones maintaining ID parity with the Isilon. Other methods to achieve these exist but are beyond the scope of this document.

**Create users on the OneFS cluster manually**

To add more users, in addition to the users that the Isilon script configures, on the OneFS cluster you can add a local user for each additional Hadoop user that submits MapReduce jobs. The following commands show how to manually add a single user called `hduser1`.

**Warning**

If your users and groups are defined by your directory service, such as Active Directory or LDAP, do NOT run these commands. This section addresses setting permissions of the HDFS root files or membership to run jobs. These steps create users, but will likely fail when you run jobs with this configuration.

**Steps to perform on the OneFS cluster**

1. Add a group to the OneFS cluster.
   ```bash
   isi auth groups create hduser1 --zone zone1 --provider local --gid <GUID>
   ```
2. Create the user and the user’s Hadoop home directories on the OneFS cluster.
   ```bash
   isi auth users create hduser1 --primary-group hduser1 --zone zone1 --provider local --home-directory /ifs/isiloncluster1/zone1/hadoop/user/hduser1 --uid <UID>
   ```
3. Assign permissions to the user’s home directory on the Hadoop cluster. The ID 2 in the example below is from when you previously ran the `isi zone zones view zone1` command.
   ```bash
   isi_run -z2 chown hduser1:hduser1 /ifs/isiloncluster1/hadoop/user/hduser1
   chmod 755 /ifs/isiloncluster1/hadoop/user/hduser1
   ```

**Perform the following step on the Hadoop client**

Since you created a new user on Isilon OneFS to run jobs, you need to create the same user with UID parity on any Linux hosts that the user will access to run jobs.

1. Add the user to the Hadoop cluster.
   ```bash
   adduser hduser1 -u <UID>
   ```

**Prepare Cloudera**

The steps in this stage occur on the Cloudera hosts that become your Hadoop servers and clients.

Hadoop clusters and services rely heavily on DNS. All client hosts in your system must be configured for both forward and reverse DNS lookups. Validate that all hosts can resolve each other’s hostnames and IP addresses. For additional information, see the Cloudera installation documentation.
Deploy Cloudera Manager

To prepare Cloudera Manager for configuration, follow the instructions for your version of Cloudera in the *Cloudera Installation and Upgrade Guide*. You can find a thorough overview of the installation procedure on the Cloudera site. This procedure begins with the download of the bits and installation of Cloudera Manager.

For an overview of Cloudera, see the following: [Overview of Cloudera and the Cloudera Documentation Set](#).

Install Cloudera Manager. Before you begin the installation of Cloudera, ensure that all your hosts meet the requirements needed by Cloudera to complete a successful Hadoop cluster installation. For more information and these installation guides, go to the [Overview of Cloudera and the Cloudera Documentation Set](#).

Before installing any Hadoop cluster, you should consult the [OneFS supportability matrix](#) for Cloudera Manager and CDH compatibility.

1. Run the following to install Cloudera.

   ```
   ./cloudera-manager-installer.bin
   ```

2. The following will display when running the installer.

3. Review the Cloudera license and accept the license agreement.
4. Accept the Oracle Binary Code license agreement.
5. Review and accept the Oracle Binary Code license agreement.
6. Install the Oracle JDK.
7. After you install the Oracle JDK, the following screen appears. Note the URL and the username/password for the Cloudera Manager WebUI.

8. Your installation should now be complete.
9. Validate that the Cloudera Manager Service is running.

```
service cloudera-scm-server status
```

A valid response is:

```
cloudera-scm-server (pid 10487) is running...
```

Look for problems at the end of the `cloudera-scm-server.log` file.

```
tail -f /var/log/cloudera-scm-server/cloudera-scm-server.log
```

10. Log in to the Cloudera Manager WebUI (http://<IP of CM Server>:7180 with the following credentials: `user`: admin; `password`: admin and accept the EULA.

11. Select the version of Cloudera that you want to deploy.
12. Specify the hosts for your CDH cluster installation. In this guide, we will deploy to a single Linux host, but the process is the same when multiple hosts are used in the Hadoop cluster. Hosts should be specified using the same hostname (FQDN) that they will identify themselves with.
   - Select **Use Parcels** (recommended), the CDH stack that you want to deploy, any additional parcels, and then click **Continue**.

13. Install the Oracle Java Development Kit (JDK) and install the Java Unlimited Strength Encryption Policy (JUSEP) files to secure the cluster.

14. Do NOT select **Single User Mode**.

15. Provide the SSH credentials, either root password or SSH keys, depending on how you configure management of your Linux hosts. The installation of the Cloudera Management package initiates.
16. Wait for the installation to complete and then click **Continue**.

17. Click **Finish**.

**Configure the Hadoop cluster**

1. **IMPORTANT:** Select the **Custom Services** option on the **Choose the CDH 5 services that you want to install on your cluster** screen to deploy the Hadoop cluster with Isilon. This is key to the Isilon integration. If you select anything other than **Custom Services**, you cannot install Isilon.
2. Once you select **Custom Services**, select the Hadoop services you wish to deploy.

**Important**

- **Do not select HDFS.** The Cloudera HDFS service is not needed.
- **Select Isilon** as the storage.
- **We also do not recommend MapReduce,** as MapReduce2 is deprecated and is included in Yarn. If you have a legacy application and it is not written for Yarn, MapReducev1 can be enabled.

3. After you have selected the Hadoop services you need, continue to assign roles on the **Customize Role Assignments** screen. Since this is a single host, all roles are deployed on the same host. Assign the hosts, per the Cloudera documentation and depending on your configuration.

4. Select the Isilon Gateway role on the same host that is running Cloudera Manager, and then click **Continue**.
5. View the selected hosts.

6. Continue with the cluster database setup. Select **Use Embedded Database** and select the default settings. Consult the Cloudera documentation for best practices regarding database configuration.

7. Test the database connections, and then click **Continue**.
Continuing with the Cluster Setup, assign the Isilon cluster to the following parameters. Note that the parameters for the SmartConnect zone ports are different.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>default_fs_name</td>
<td>hdfs://smartconnectzonename:8020</td>
</tr>
<tr>
<td>webhdfs_url</td>
<td><a href="http://smartconnectzonename:8082/webhdfs/v1">http://smartconnectzonename:8082/webhdfs/v1</a></td>
</tr>
</tbody>
</table>
### Cluster Setup

#### Review Changes

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HDFS Root Directory</strong></td>
<td>Cluster 1 &gt; HBase (Service-Wide)</td>
</tr>
<tr>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>Path</td>
<td>/hbase</td>
</tr>
<tr>
<td><strong>Enable Indexing</strong></td>
<td>Cluster 1 &gt; HBase (Service-Wide)</td>
</tr>
<tr>
<td><strong>Enable Replication</strong></td>
<td>Cluster 1 &gt; HBase (Service-Wide)</td>
</tr>
<tr>
<td><strong>Hive Warehouse Directory</strong></td>
<td>Cluster 1 &gt; Hive (Service-Wide)</td>
</tr>
<tr>
<td>Name</td>
<td>/user/hive/warehouse</td>
</tr>
<tr>
<td><strong>Hive Metastore Server Port</strong></td>
<td>Cluster 1 &gt; Hive Metastore Server Default Group</td>
</tr>
<tr>
<td>Port</td>
<td>9083</td>
</tr>
<tr>
<td><strong>Impala Daemon Scratch Directories</strong></td>
<td>Cluster 1 &gt; Impala Daemon Default Group</td>
</tr>
<tr>
<td><strong>HDFS Block Size</strong></td>
<td>Cluster 1 &gt; Isilon (Service-Wide)</td>
</tr>
<tr>
<td>Size</td>
<td>128 MB</td>
</tr>
</tbody>
</table>

#### Default File System URI

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>default_fs_name</strong></td>
<td></td>
</tr>
</tbody>
</table>

#### WebHDFS URL

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>webhdfs_url</strong></td>
<td></td>
</tr>
</tbody>
</table>

#### Alerts: Mail Server Hostname

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Alert Server Hostname</strong></td>
<td>Isilon Publisher Default Group</td>
</tr>
<tr>
<td>Hostname</td>
<td>localhost</td>
</tr>
</tbody>
</table>

#### Alerts: Mail Server Username

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Alert Server Username</strong></td>
<td>Isilon Publisher Default Group</td>
</tr>
</tbody>
</table>

#### Alerts: Mail Server Password

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Alert Server Password</strong></td>
<td>Isilon Publisher Default Group</td>
</tr>
</tbody>
</table>

#### Alerts: Mail Message Recipients

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Alert Message Recipients</strong></td>
<td>Isilon Publisher Default Group</td>
</tr>
<tr>
<td>Recipients</td>
<td>root@localhost</td>
</tr>
</tbody>
</table>

#### Custom Alert Script

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>alert_script_path</strong></td>
<td></td>
</tr>
</tbody>
</table>

#### Host Monitor Storage Directory

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Host Monitor Storage Directory</strong></td>
<td>Isilon Monitor Default Group</td>
</tr>
<tr>
<td>Storage Directory</td>
<td>/var/lib/cloudera-host-monitor</td>
</tr>
<tr>
<td>Service Monitor Storage Directory</td>
<td>Setenv Monitor Default Group</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td></td>
<td>/var/lib/cloudera-service-monitor</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ShareLib Root Directory oozie.service.WordflowAppService system.lib path</th>
<th>Cluster 1 &gt; Oozie (Service-Wide)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>/var/lib/oozie</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Oozie Server Data Directory</th>
<th>Cluster 1 &gt; Oozie Server Default Group</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>/var/lib/oozie/data</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Zookeeper Znode</th>
<th>Cluster 1 &gt; Sbtr (Service-Wide)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>/sbtr</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HDFS Data Directory</th>
<th>Cluster 1 &gt; Sbtr (Service-Wide)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>/sbtr</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sqoop 2 Server Metastore Directory</th>
<th>Cluster 1 &gt; Sqoop 2 Sbtr Default Group</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>/var/lib/sqoop2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sqoop Repository Database Type</th>
<th>Cluster 1 &gt; Sqoop 2 Sbtr Default Group</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Derby</td>
</tr>
<tr>
<td></td>
<td>PostgreSQL</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sqoop Repository Database Host</th>
<th>Cluster 1 &gt; Sqoop 2 Sbtr Default Group</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>localhost</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sqoop Repository Database Name</th>
<th>Cluster 1 &gt; Sqoop 2 Sbtr Default Group</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>sqoop</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sqoop Repository Database User org.apache.sqoop.repositorydb.user</th>
<th>Cluster 1 &gt; Sqoop 2 Sbtr Default Group</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>sqoop</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sqoop Repository Database Password org.apache.sqoop.repositorydb.password</th>
<th>Cluster 1 &gt; Sqoop 2 Sbtr Default Group</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NodeManager Local Directories yarn.nodemanager.local.dirs</th>
<th>Cluster 1 &gt; NodeManager Default Group</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>/yarn/nm</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Enable Container Usage Metrics Collection</th>
<th>Cluster 1 &gt; YARN (MR2 included) (Service-Wide)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cloudera Manager Container Usage Metrics Directory</th>
<th>Cluster 1 &gt; YARN (MR2 included) (Service-Wide)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><a href="http://yarn/containermetrics">http://yarn/containermetrics</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Container Usage Output Directory</th>
<th>Cluster 1 &gt; YARN (MR2 included) (Service-Wide)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><a href="http://yarn/containermetrics/aggregate">http://yarn/containermetrics/aggregate</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Container Usage MapReduce Job User</th>
<th>Cluster 1 &gt; YARN (MR2 included) (Service-Wide)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Directory dataDir</th>
<th>Cluster 1 &gt; Sbtr Default Group</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>/var/lib/oozie/data</td>
</tr>
</tbody>
</table>
9. Assign the two Isilon parameters. Leave all the default settings, and then click **Continue**.

10. Continue the setup and monitor deployment.

11. Review the cluster setup as it deploys.
12. The setup completes. You can review additional details by opening specific services.
13. Click **Continue** to complete the Hadoop cluster deployment. The services are now installed, configured, and running on your cluster.
Troubleshoot Cloudera Manager

1. Return to the main Cloudera Manager dashboard to review the status. It is common to see alarms and down issues on the dashboard. Review the alarms and services and triage as needed. Some services might simply need restarting.

Follow the standard protocols in starting these services, for example:

- Start the service
- Monitor and review logs as needed
- Review the Isilon /var/log/hdfs.log file on all nodes
- Restart services to resolve alarms following configuration changes.
It is common to see configuration issues. Address these and make the required changes as needed to resolve each issue.

**Note**

Since Isilon uses OneFS as the native file storage format and not HDFS, Isilon does not support HDFS Trash recovery so you can keep the HDFS Trash as disabled. Isilon OneFS does support other mechanisms to recover deleted files, as OneFS primarily supports Snapshots or SyncIQ.

Restart services to resolve alarms following configuration changes.
The Hadoop cluster and services are now fully operational and can be tested.

Verify the deployment

Test the basic functionality of the Isilon and Cloudera integration (without Kerberos) with the following steps.

1. Browse the Isilon HDFS root.

   ```
   hadoop fs -ls /
   ```

   Output similar to the following displays:
2. Write to the Isilon HDFS root by creating a test directory, for example, “Made_from_Cloudera”.

Output similar to the following displays:

Found 6 items

```
-rw-r-x-- 3 root hadoop 0 2017-04-24 15:04 /Made_from_Cloudera
```

3. Run some basic smoketest jobs, for example PI or Teragen, Teravalidate, or Terasort to test MapReduce.

**NOTE:** With Cloudera 5.7, you may notice that Impala service is not started fully. Some additional configuration changes are needed to get this service started. The steps to install and configure Impala are in the following document: Get Cloudera 5.7 Impala running with Isilon.
This completes the overview of deploying non-Kerberos Cloudera CDH with Isilon OneFS.

**Contacting Isilon Technical Support**

**Online Support:** [https://support.emc.com/](https://support.emc.com/)

**Telephone Support:**
United States: 800-782-4362 (800-SVC-4EMC)  
Canada: 800-543-4782  
Worldwide: +1-508-497-7901  
Additional [worldwide access numbers](https://support.emc.com/)

**Help with Online Support Tools:**

For questions specific to HDFS, contact [support@emc.com](mailto:support@emc.com)