Abstract

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

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>April 28, 2017</td>
<td>Initial version</td>
</tr>
</tbody>
</table>
| 1.01    | July 31, 2017 | - Updated list of users that the Isilon_hadoop_tools script creates.  
- Removed instructions to modify the block size that is used for reading from Isilon since the block size is 128M by default.  
- 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. |
| 1.03    | October 5, 2017 | - Added “Updates and Additional Information on Isilon Hadoop Installs” section.                                                      |
CONTENTS

Introduction ................................................................................................................................. 5
Audience ....................................................................................................................................... 5
Overview ....................................................................................................................................... 5
Updates and additional information on Isilon Hadoop installs .................................................. 7
Prerequisites .................................................................................................................................. 7
  Cloudera distribution with Apache Hadoop (CDH) ................................................................. 7
  Isilon OneFS cluster configuration ......................................................................................... 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
  Prepare Cloudera ....................................................................................................................... 17
    Deploy Cloudera Manager ....................................................................................................... 18
    Configure the Hadoop cluster ................................................................................................. 23
    Troubleshoot Cloudera Manager ............................................................................................ 30
    Verify the deployment ............................................................................................................. 32
Configure Kerberos with OneFS and Cloudera Manager ............................................................. 34
  Prerequisites ............................................................................................................................. 34
    Isilon OneFS ............................................................................................................................ 34
    Cloudera Manager .................................................................................................................. 35
Enable Kerberos with OneFS on Cloudera using Active Directory authentication ........................ 35
  How Kerberos is implemented on the OneFS and Hadoop clusters ........................................ 36
  Pre-configure Cloudera Manager ............................................................................................. 36
  Prepare hosts for Kerberization ............................................................................................... 37
  Configure Isilon OneFS for Kerberos authentication ............................................................... 37
  Review the OneFS SPNs in Isilon OneFS ................................................................................ 39
  Create proxy users ................................................................................................................... 39
Enable Kerberos authentication in Cloudera Manager .............................................................. 39
Enable Kerberos with OneFS on Cloudera using MIT KDC................................................................. 51
Prepare hosts for Kerberization........................................................................................................... 51
Enable CDH Kerberos using MIT........................................................................................................ 51
Create the KDC as an Isilon authorization provider.......................................................................... 57
Configure ports in Cloudera Manager............................................................................................... 61
Test and validate Hadoop services.................................................................................................... 63
Troubleshoot services........................................................................................................................ 65
Contact Isilon Technical Support....................................................................................................... 71
**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](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.

<table>
<thead>
<tr>
<th>Component</th>
<th>Version/License</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isilon OneFS</td>
<td></td>
</tr>
<tr>
<td>SmartConnect module</td>
<td></td>
</tr>
</tbody>
</table>
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
   ```
3. If your modules are not licensed, obtain a license key from your Isilon sales representative. Type the following command to activate the license:

```bash
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.

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

2. Confirm that you have successfully created access zones.

```bash
isi zone zones list --verbose
```

Output similar to the following displays:
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

```bash
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.

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

Output similar to the following displays:

```
ID: groupnet0subnet0hadoop-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.

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

2. Set the HDFS root directory for the access zone by running the following command:

   ```
isi hdfs settings modify --zone=zone1-cdh --root-directory=/ifs/zone1/cdh/hadoop-root
   ```

3. 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.

   ```
isi zone zones modify --user-mapping-rules="hdfs=>root" --zone zone1-cdh
   ```
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 GIDs). 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 exists. 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 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.
```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
```
   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
```
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
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 of 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
bash isilon_create_users.sh --dist cdh --startuid 501 --startgid 501 --zone zone1-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: zone1-cdh
Info: HDFS root: /ifs/zone1/cdh/hadoop-root
Info: passwd file: zone1-cdh.passwd
Info: group file: zone1-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
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
bash isilon_create_directories.sh --dist cdh --zone zone1-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: zone1-cdh
Info: will fix permissions and owners on existing directories created by this script.
Info: HDFS root dir is /ifs/zone1/cdh/hadoop-root
```
h. List the HDFS base directories.

```
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.

```
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 MIT KDC/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.
   
   ```
   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.
   
   ```
   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.
   
   ```
   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.
   
   ```
   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.
   ```bash
   ./cloudera-manager-installer.bin
   ```

2. The following will display when running the installer.

   ![Cloudera Manager Installer](image)

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.

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

   A valid response is

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

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

   ```bash
   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**.

**Note**

The installer will check and validate hosts. If the validation check fails, follow the recommendations to resolve and then re-try the validation. Common errors are the following, which are all related to Linux (not Isilon).

- Transparent_hugepage
- Swappiness

On successful completion of the host inspector, click **Continue**.
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.

   ![Choose the CDH 5 services that you want to install on your cluster](image)

   - Choose a combination of services to install.
   - **Core Hadoop**
     - HDFS, YARN (MapReduce 2 included), ZooKeeper, Oozie, Hive, and Hue
   - **Core with HBase**
     - HDFS, YARN (MapReduce 2 included), ZooKeeper, Oozie, Hive, HBase, and Impala
   - **Core with Impala**
     - HDFS, YARN (MapReduce 2 included), ZooKeeper, Oozie, Hive, Hue, and Impala
   - **Core with Search**
     - HDFS, YARN (MapReduce 2 included), ZooKeeper, Oozie, Hive, Hue, and Solr
   - **Core with Spark**
     - HDFS, YARN (MapReduce 2 included), ZooKeeper, Oozie, Hive, Hue, and Spark
   - **All Services**
     - HDFS, YARN (MapReduce 2 included), ZooKeeper, Oozie, Hive, HBase, Impala, Solr, Spark, and Key-Value Store Indexer
   - **Custom Services**
     - Choose your own services. Services required by chosen services will automatically be included. Plume can be added after your initial cluster has been set up.

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**.

8. 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><code>default_fs_name</code></td>
<td>hdfs://smartconnectzonename:8020</td>
</tr>
<tr>
<td><code>webhdfs_url</code></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>HDFS Root Directory</td>
<td>/hbase</td>
</tr>
<tr>
<td>Enable Indexing</td>
<td>Cluster 1 &gt; HBase (Service-Wide)</td>
</tr>
<tr>
<td>Enable Replication</td>
<td>Cluster 1 &gt; HBase (Service-Wide)</td>
</tr>
<tr>
<td>Hive Warehouse Directory</td>
<td>/user/hive/warehouse</td>
</tr>
<tr>
<td>Hive Metastore Server Port</td>
<td>9003</td>
</tr>
<tr>
<td>Impala Daemon Scratch Directories</td>
<td>/impala/impalad</td>
</tr>
<tr>
<td>HDFS Block Size</td>
<td>128 MB</td>
</tr>
<tr>
<td>Default File System URI</td>
<td>/hadoop</td>
</tr>
<tr>
<td>WebHDFS URL</td>
<td></td>
</tr>
<tr>
<td>Alerts: Mail Server Hostname</td>
<td>localhost</td>
</tr>
<tr>
<td>Alerts: Mail Server Username</td>
<td></td>
</tr>
<tr>
<td>Alerts: Mail Server Password</td>
<td></td>
</tr>
<tr>
<td>Alerts: Mail Message Recipients</td>
<td>root@localhost</td>
</tr>
<tr>
<td>Custom Alert Script</td>
<td></td>
</tr>
<tr>
<td>Host Monitor Storage Directory</td>
<td>/var/lib/cloudera-host-monitor</td>
</tr>
<tr>
<td>Service Monitor Storage Directory</td>
<td>Storage Monitor Default Group</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-----------------------------</td>
</tr>
</tbody>
</table>
| Hazelcast storage base directory | Hazelcast Hadoop service

<table>
<thead>
<tr>
<th>ShareLib Root Directory</th>
<th>Cluster 1 &gt; Oozie (Service-Wide)</th>
</tr>
</thead>
<tbody>
<tr>
<td>sharelib/oozie</td>
<td>/usr/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>/var/lib/oozie/data</td>
<td></td>
</tr>
</tbody>
</table>

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

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

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

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

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

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

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

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

<table>
<thead>
<tr>
<th>NodeManager Local Directories</th>
<th>Cluster 1 &gt; NodeManager Default Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>yarn. nodemanager.localdirs</td>
<td></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>/hadoop/yarn/containermetrics</td>
<td></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>/hadoop/yarn/containermetricsaggregate</td>
<td></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</th>
<th>Cluster 1 &gt; Stack Default Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>dataDir</td>
<td>/var/lib/oozie</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 as needed 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:

   ```
   Found 5 items
   ```
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--r-- 3 root wheel 0 2017-04-24 15:04 /Made_from_Cloudera
```

3. Run some basic smoketest jobs, for example PI or Teragen, Teraload, 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.

**Configure Kerberos with OneFS and Cloudera Manager**

You can configure Kerberos security with OneFS 8.0.0.1 and later versions using existing Microsoft Active Directory or MIT KDC installations.

**Prerequisites**

Before you configure Kerberos on your Isilon OneFS cluster, make sure that the prerequisites described in the following sections are met.

**Isilon OneFS**

This guide assumes the following Isilon Hadoop environment is configured and operational.

- You must be running Isilon OneFS 8.0.0.1 or later versions.
- A dedicated Isilon access zone is in use (not the system zone).
- An Isilon SmartConnect zone is correctly configured for Isilon HDFS access.
- A simple access model currently exists between Hadoop and Isilon. User UIDs and GIDs are correctly implemented and allow HDFS access to the Isilon HDFS root with UID and GID parity.
- Hadoop jobs and services are fully operational.
- DNS for SmartConnect is correctly configured, including forward and reverse lookups.

Also, ensure that OneFS is preconfigured appropriately to respond to requests related to secure Kerberized HDFS that is authenticated by MIT Kerberos key distribution center (KDC) or by Microsoft Active Directory (AD).
Cloudera Manager

Make sure that the following prerequisites are met:

- You must be running CDH 5.7.1 or later.
- You will be integrating an existing Cloudera CDH cluster and an Isilon cluster into a pre-existing Microsoft Active Directory (AD) environment for Kerberos user authentication. The following must be configured:
  - Forward and reverse DNS lookups must be enabled on all hosts.
    - All compute hosts must have forward DNS lookup resolved correctly for all hosts.
    - Isilon SmartConnect zone name lookups must resolve correctly.
    - Reverse PTR records for all IP addresses in the SmartConnect pool must exist.
    - Isilon OneFS must be able to resolve all hosts, KDCs, and Active Directory servers as needed.
  - Cloudera Manager must be configured correctly for Isilon integration.
  - Cloudera Manager must be able to manage and deploy keytab and krb5.conf files.
  - If you are using an existing MIT KDC installation, MIT KDC must be running.

Note:

If your environment deviates from any of the above Isilon and Cloudera prerequisites, an alternative approach to Kerberization may be required, especially with the management of keytab and krb5.conf files.

This guide does not address Linux host Kerberization, Directory Service integration, or the Isilon OneFS permissioning model for multiprotocol access following Kerberization. This guide will not address all configurations or requirements. Additional EMC Isilon support services should be engaged when required.

Enable Kerberos with OneFS on Cloudera using Active Directory authentication

Use this procedure to enable Kerberos on a CDH cluster using Active Directory (AD). Refer to the Cloudera documentation for additional information on how to enable Kerberos authentication.

Since you will be integrating an existing Cloudera CDH cluster and Isilon OneFS cluster into a pre-existing Microsoft Active Directory environment, the high-level approach is the following:

1. Prepare and configure Active Directory (AD) for the Isilon and Hadoop integration.
2. Prepare the Cloudera cluster and Linux hosts for Kerberization.
3. All services must be running on the Cloudera dashboard prior to Kerberization.
4. Integrate the Isilon cluster into Active Directory.
5. Kerberize the CDH cluster using the Cloudera enable Kerberos wizard.
6. Cloudera must be able to manage and deploy keytab and krb5.conf files.
7. Complete the integration of Isilon and Cloudera.

8. Test and validate the Kerberized services.

To use an existing Active Directory domain for the cluster with the Cloudera Kerberos wizard, you must prepare the following:

- Isilon, Cloudera Manager, and the compute cluster hosts must have all required network access to Active Directory and AD services.
- All DNS name resolutions of the required Active Directory services are valid.
- The Active Directory supports secure LDAPS connectivity has been configured.
- The active OU user container for principals has been correctly created. For example, “OU=Hadoop-Cluster,OU=People,dc=domain,dc=com”.
- Active Directory administrative credentials with delegated control of “Create, delete, and manage user accounts” on the OU user container are implemented.

How Kerberos is implemented on the OneFS and Hadoop clusters

Since the Isilon-integrated Hadoop cluster is a blend between Linux hosts running compute services and Isilon OneFS running data services, Cloudera cannot effectively complete the Kerberization end-to-end. Since OneFS is a clustered operating system, you cannot use SSH-based remote management to configure and manage the Kerberization of Isilon OneFS completely. The Kerberization of an Isilon OneFS-integrated Hadoop cluster should be deployed as follows:

- The Isilon OneFS cluster is Kerberized.
- The Cloudera Kerberization wizard deploys Kerberization to the Linux and Hadoop services.

When both the OneFS and Hadoop cluster are fully Kerberized within the same Active Directory domain, Kerberized user access can occur between both systems seamlessly.

For additional information, refer to the Cloudera security documents: [Enabling Kerberos Authentication Using the Cloudera Wizard](#).

Pre-configure Cloudera Manager

Review the following configuration settings before proceeding with the Cloudera installation. Make sure that:

- Cloudera 5.x or a later version is running.
- Services are running (all green) on the Cloudera Manager Dashboard.
- All Cloudera-specific Kerberos requirements have been met, for example, NTP, DNS, and packages.

Before you launch the Cloudera Kerberization wizard, you must make the following configuration customizations and restart all services.

1. Click the Clusters tab on the Cloudera Manager dashboard. For the Isilon cluster, click the Configuration tab. For the Isilon service, select the Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml property, and then set the value of the hadoop.security.token.service.use_ip property to FALSE. Note that you may need to create this key.
Prepare hosts for Kerberization

Install the required client libraries in order for Kerberization to be operational on all Hadoop hosts. The OpenLDAP client libraries must be installed on the Cloudera Manager server, and all Kerberos client libraries must be installed on all hosts. Refer to the Cloudera documentation for more information: Enabling Kerberos Authentication Using the Wizard.

On Red Hat Enterprise Linux (RHEL) or Community Enterprise Operating System (CentOS), install the appropriate packages using one of the following `yum` commands:

**On all compute hosts:**

```bash
yum -y install krb5-workstation krb5-libs openldap-clients
```

Configure Isilon OneFS for Kerberos authentication

This section covers the configuration requirements for OneFS to respond to requests for secure Kerberized HDFS authenticated by Active Directory.

The following must be configured correctly before you can proceed to the next section:

- The cluster must be joined correctly to the target Active Directory as a provider. Configure the following advanced settings in OneFS web administration interface. These settings maintain user and identity mappings between users executing Hadoop jobs and the Isilon OneFS cluster, and also enable a standard Isilon permission model.
  - Click Access > Authentication Providers > Active Directory.
  - In the Active Directory Providers table, click View details for the provider whose settings you want to modify.
  - Click Advanced Active Directory Settings.
- Specify RFC 2307 for the Services For Unix Setting. Make sure that you have enabled Active Directory GC indexing and replication as described in article 000335338 for OneFS versions 8.x.x.x. and Windows Server 2012. This is a required configuration to support Active Directory that provides UIDs and GIDs to OneFS.

- The access zone that contains the HDFS root must be configured for this Active Directory provider, and the HDFS access zones service must be configured for Kerberos only.

- Isilon Service Principal Names (SPNs) must be correctly configured. Users running Hadoop jobs must have Active Directory user principals with Unix attributes allocated. OneFS leverages the Active Schema extension that support Unix identities. These schema attributes extend Active Directory objects to provide UIDs and GIDs to a user account in Active Directory. Depending on your setup, your Linux hosts might need to be integrated into AD for identity management.

- All IP addresses within the required SmartConnect zone must be added to the reverse DNS with the same fully qualified domain name (FQDN) for the cluster delegation. All IPs should resolve back to the SmartConnect zone. This is required for Kerberos authentication.

- Add the mapping rules to map the local HDFS to root, the Active Directory HDFS principal to root, the domain\hdfs to root, and all domain users to the local user, if applicable, by running the following command. In this example, “vlab” is the domain name and “zone1-hdp” is the access zone:

```
isi zone zones modify --user-mapping-rules="hdfs=>root, vlab\hdfs=>root, vlab\* &= *[], vlab\* += *[group], vlab\* += *[groups]" --zone=zonel-hdp
```

where:

<table>
<thead>
<tr>
<th>Mapping</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hdfs=&gt;root</td>
<td>Maps the HDFS user to root</td>
</tr>
<tr>
<td>vlab* &amp;= *[]</td>
<td>Maps all AD users to the local user, for example, AD\bob = bob, AD\jane = jane</td>
</tr>
<tr>
<td>vlab* += *[group]</td>
<td>(optional) Maps the users' primary group to AD; defines the GID group and not domain users</td>
</tr>
<tr>
<td>vlab* += *[groups]</td>
<td>(optional) Maps the users' primary group to AD; defines GID group and not domain users</td>
</tr>
</tbody>
</table>

Mapping rules should be made with the short NetBIOS (Network Basic Input/Output System) name of the domain only, not the fully-qualified domain name.

Output the mapping results:

```
isi zone zones list -v
```

The tokens for hdfs and hdfs@domain must be the same and must map to root using the following commands. Refer to the Isilon Troubleshooting Guide: Multiprotocol Authentication Identity Management Authorization AIMA if you encounter any problems.

```
isi auth mapping token --zone=<zone-name> --user=hdfs
isi auth mapping token --zone=<zone-name> --user=hdfs@domain.com
```
Review the OneFS SPNs in Isilon OneFS

Since OneFS is a clustered file system running on multiple nodes joined to Active Directory as a single Computer Object, the Service Principal Name (SPN) requirements for Kerberized Hadoop access are unique. OneFS requires additional SPNs for the access zone to which the HDFS NameNode access is provided when Active Directory is used, as summarized in the following table:

<table>
<thead>
<tr>
<th>SPN</th>
<th>Name</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>hdfs/clustename.fdn</td>
<td>Clustename that is joined to AD</td>
<td>hdfs authentication to AD</td>
</tr>
<tr>
<td>hdfs/namenode.smartconnectname.fdn</td>
<td>NN FQDN used by Ambari</td>
<td>hdfs authentication to AD per Smartconnect Zone</td>
</tr>
<tr>
<td>HTTP/namenode.smartconnectname.fdn</td>
<td>NN FQDN used by Ambari</td>
<td>WebHDFS authentication to AD per Smartconnect Zone</td>
</tr>
</tbody>
</table>

Review the registered SPNs on the OneFS cluster and add the required SPNs for the SmartConnect zone name if needed by running the following command:

```
is_auth ads spn list --provider-name=<AD_PROVIDER_NAME>
```

The following example illustrates the required OneFS SPNs:

Isilon Cluster Name - rip2.foo.com - SPN: hdfs/rip2.foo.com
Access Zone NN SmartConnect FQDN - rip2-cd1.foo.com - SPN's: hdfs/rip2-cd1.foo.com & HTTP/rip2-cd1.foo.com

For additional information on adding or modifying OneFS SPNs in Active Directory, refer to the Isilon OneFS CLI Administration Guide.

Create proxy users

Create the required proxy users. Proxy users are required for service account impersonation for specific Hadoop services to run jobs and to add the required proxy users as needed. For more information on creating proxy users, refer to the Isilon OneFS CLI Administration Guide. If the Isilon create users script was used, you will need to create all proxy users.

This completes the Isilon OneFS Hadoop Active Directory setup.

Enable Kerberos authentication in Cloudera Manager

After meeting the preceding prerequisites, you can Kerberize the Cloudera cluster. Isilon recommends that you suspend all client and user activity on the Hadoop cluster prior to executing any Kerberization tasks.

1. On the Cloudera Manager dashboard, click Administration and then select Security.
2. On the Security screen, click **Enable Kerberos**.

3. The Cloudera Manager wizard walks you through the steps to configure Cloudera Manager and CDH to use Kerberos for authentication. Check all of the boxes when you have completed the steps and then click **Continue**.
4. Using the **KDC Information** screen, configure the following settings:

- **KDC Type**—Select **Active Directory**.
- **KDC Server Host**—FQDN of the KDC server host.
- **Kerberos Security Realm**—Name of the Kerberos realm that you are joining. The Kerberos security realm (the AD domain) must contain uppercase characters. For example, VLAB.LOCAL.
- **Kerberos Encryption Types**—Any additional Kerberos encryption types (OneFS 8.0.x supports aes-256)
- **Active Directory Suffix**—Modify the OU for the delegated Cloudera OU to be used for principals.
If you are using a version of CDH that supports the following, select the **Active Directory Delete Accounts on Credential Regeneration** checkbox.

Complete the relevant remaining fields on this screen and then click **Continue**.

5. Continuing in the Cloudera Manager wizard, manage the host `krb5.conf` host files by selecting the **Manage krb5.conf through Cloudera Manager** check box.
6. Accept the defaults on the next screen and then click **Continue**. Since Cloudera Manager will create and manage all of the principals, an AD OU with a delegated administrative account is used.

7. Look up the credentials for the AD user with delegated access to the OU in the AD domain. In our example, **krb-admin-acct** is the AD user.
8. Enter the KDC account manager credentials for the AD user in Cloudera Manager and then click Continue.

9. The following **Import KDC Account Manager Credentials** screen displays.
If Active Directory is not configured for LDAPS, errors may display, which you will need to fix to continue.

10. On the **Kerberos Principal** screen, accept the defaults and then click **Continue**.

11. On the **Configure Ports** screen, accept the default ports. Select **Yes, I am ready to restart the cluster now** and then click **Continue**.
12. Disable simple authentication on the OneFS cluster by running the following command or by editing the HDFS settings using the OneFS web administration interface. This enforces Kerberos-only or delegation token authentication access.

   ```bash
   isi hdfs settings modify --authentication-mode=kerberos_only --zone=rip2-cd1
   ```
13. The Kerberization process is automatically initialized. The Cloudera services are Kerberized, user principals are created in Active Directory, and keytabs are distributed.

After the process is complete, you can view the principals in the AD Server Manager.

14. The Kerberos enablement will continue and the service will attempt to restart.
The following screen displays if the Kerberization was successful.

The Hue service may fail and halt the wizard. This is a known issue in some versions of Cloudera that you will need to work around if Hue is in use.
15. If the failure of the Hue service prevents the wizard from completing, you need to perform the following:

- Open another browser session to the Cloudera Manager dashboard and review the state of the services. Enter the following URL: http://<Cloudera Manager URL>:7180 in your browser.

- You will most likely see some services in an unhealthy state. You can address each of these individually by starting or restarting them as required. Monitor the log files to get them started. You may need to restart some services manually.
Once you have restarted services, they should all be fully Kerberized. Address any configuration issues or alarms as required.

On completion of restarting all services except Hue, you can close the other Kerberization wizard browser. All services are now Kerberized, and the cluster is operational (except the Hue service) as displayed below:
For more information on starting the Hue service, refer to the Getting the Hue Service Started on Kerberized Cloudera with Isilon ECN blog post.

This completes the procedure for Kerberizing Cloudera with Isilon OneFS using Active Directory.

Proceed to the Troubleshoot services and Test and validate Hadoop services sections of this document.

Enable Kerberos with OneFS on Cloudera using MIT KDC

Use the following procedures to enable Kerberos on a CDH cluster using the MIT key distribution center (KDC).

Prepare hosts for Kerberization

Install the required client libraries in order for Kerberization to be operational on all Hadoop hosts. The OpenLDAP client libraries must be installed on the Cloudera Manager server, and all Kerberos client libraries must be installed on all hosts. Refer to the Cloudera documentation for more information: Enabling Kerberos Authentication Using the Wizard.

Install packages

On Red Hat Enterprise Linux (RHEL) or Community Enterprise Operating System (CentOS), install the appropriate packages using one of the following `yum` commands:

- **On the KDC** (only required if you are setting up a new KDC):
  
  `yum -y install krb5-server krb5-libs krb5-workstation openldap-clients`

- **On Cloudera Manager** (only required if you are setting up a new KDC):
  
  `yum -y install krb5-libs krb5-workstation openldap-clients`

- **On all compute hosts**:
  
  `yum -y install krb5-workstation krb5-libs openldap-clients`

KDC setup and configuration

The KDC and Kerberos realm should be set up and configured per the Cloudera requirements and your realm setup.

Enable CDH Kerberos using MIT

Use this procedure to enable Kerberos on a CDH cluster using MIT KDC. Refer to the Cloudera documentation for additional information on how to enable Kerberos authentication. See Get or Create a Kerberos Principal for the Cloudera Manager for an example of creating principals for MIT KDC.

Isilon recommends that you suspend all client and user activity on the Hadoop cluster prior to executing any Kerberization tasks.

1. On the Cloudera Manager dashboard, click Administration and then select Security.

3. The Cloudera Manager wizard walks you through the steps to configure Cloudera Manager and CDH to use Kerberos for authentication. Check all of the boxes when you have completed the steps and then click Continue.
4. Using the **KDC Information** screen on the **Enable Kerberos for Cluster 1** wizard, configure the following settings and then click **Continue**:

- **KDC Type**—Select MIT KDC.
- **KDC Server Host**—FQDN of the KDC server host.
- **Kerberos Security Realm**—Name of the Kerberos realm that you are joining. The realm name must contain uppercase characters. For example, VLAB.LOCAL.
- **Kerberos Encryption Types**—Specify additional Kerberos encryption types (OneFS 8.0.x and later versions support aes-256).
5. On the **KRB5 Configuration** screen, select the **Manage krb5.conf through Cloudera Manager** check box. After specifying the configuration settings, click **Continue**.
6. Enter the credentials for the account that has permissions to create other users on the **KDC Account Manager Credentials** screen. The Cloudera admin account “cloudera-scm/admin” in the VLAB.LOCAL realm is used in the example below. This account needs to be set up per Cloudera requirements within the KDC realm to facilitate principal management.

The **Enable Kerberos for Cluster 1** wizard will automatically import your existing KDC account manager credentials.
7. No changes are required on the **Kerberos Principal** screen. These are the services you will be Kerberizing. Click **Continue**.

8. On the **Configure Ports** screen, configure the privileged ports required by DataNodes in a secure HDFS service.

**Caution**

**STOP and do not proceed** on the **Configure Ports** screen until you Kerberize the Isilon OneFS cluster. Leave the ports as the default, and then do not proceed until you configure Isilon in the next section.
Create the KDC as an Isilon authorization provider

1. Run a command similar to the following (using your parameters) on your Isilon OneFS cluster to create the realm, for example:

   ```bash
   isi auth krb5 create --realm=VLAB.LOCAL --admin-server=RDUVNODE60909.vlab.local -kdc=RDUVNODE60909.vlab.local --user=cloudera-scm/admin@VLAB.LOCAL
   ```

2. List the realm.

   ```bash
   isi auth krb5 realm list
   ```

   For example:

   ```bash
   steven-45uornw-1# isi auth krb5 realm list
   Realm                      Is Default Realm? KDC                  Admin Server
   VLAB.LOCAL                  Yes                      RDUVNODE60909.vlab.local RDUVNODE60909.vlab.local
   Total: 1
   steven-45uornw-1# 
   ```

3. Create the Kerberos domains.

   ```bash
   isi auth krb5 domain create --domain=<domain-name> --realm=<realm-name>
   isi auth krb5 domain create --domain=.<domain-name> --realm=<realm-name>
   isi auth krb5 domain list -verbose
   ```

   For example:

   ```bash
   steven-45uornw-1# isi auth krb5 domain create --domain=vlab.local --realm=VLAB.LOCAL
   steven-45uornw-1# isi auth krb5 domain create --domain=.vlab.local --realm=VLAB.LOCAL
   steven-45uornw-1# isi auth krb5 domain list --verbose
   Domain: vlab.local
   Realm: VLAB.LOCAL
   -----------------------------------------------
   Domain: vlab.local
   Realm: VLAB.LOCAL
   ```
You can also view the two Kerberos domains you just created in the OneFS web administration interface under the **Kerberos Provider** tab as shown in the following screen. (Note that since we have not added Isilon SPNs yet, we see the **Requires Additional Configuration** warning here):

![Kerberos Provider Interface]

4. Add the Kerberos provider to the access zone and view the zones.

   ```bash
   isi zone zones modify --zone=<zone-name> --add-auth-provider=<provider-type>:<provider-name>
   isi zone zones view --zone=<zone-name>
   ```

   For example:

   ```bash
   stefana45surya-16 isi zone zones modify --zone=zone2-cdh --add-auth-provider=krb5:VLAB.LOCAL
   stefana45surya-16 isi zone zones view --zone=zone2-cdh
   Name: zone2-cdh
   Realm: /is/zone2/cdh
   Groupmap: groupmap6
   Enckty Untrusted: -
   Auth Providers: lsam-local-provider:zone2-cdh, lsam-krb5-provider:VLAB.LOCAL
   NetBIOS Name: -
   User Mapping Rule: hdfs->root
   UNIX Directory Template: /var/zone2/cdh
   Cache Entry Expiry: 1m
   Zone ID 9
   ```

5. Create the service principal names (SPNs) (using your Kerberos provider names) by running the following command. Note that MIT KDC requires two SPNs: hdfs/smartconnectzone-name and HTTP/smartconnectzone-name.
6. List the Kerberos realms by running the following command.

   ```bash
   isi auth krb5 spn list
   ```

   For example:

   ![Kerberos realms list](image)

   You can also view the principals in the OneFS web administration interface. Note that the warnings you saw previously are now gone.

7. List the Isilon principals created by Isilon on the KDC by running the following command after logging into the KDC:
For example:

```
listprincs

kadmin.local: listprincs
HTTP/rdunode60904.vlab.local@VLAB.LOCAL
HTTP/isilonzone-cdh2.vlab.local@VLAB.LOCAL
E/IVL vãoI@VLAB.LOCAL
clusers.scm/admin8@VLAB.LOCAL
krb5e/rdunode60904.vlab.local@VLAB.LOCAL
kdfs/isilonzone-cdh2.vlab.local@VLAB.LOCAL
kine/rdunode60904.vlab.local@VLAB.LOCAL
kine/rdunode60904.vlab.local@VLAB.LOCAL
kneie/rdunode60904.vlab.local@VLAB.LOCAL
kendr/admin9@VLAB.LOCAL
kadmin/changewe@VLAB.LOCAL
kadmin/rdunode60904.vlab.local@VLAB.LOCAL
kocuser10@VLAB.LOCAL
kdfi7t@VLAB.LOCAL@VLAB.LOCAL
mapred/rdunode60904.vlab.local@VLAB.LOCAL
cozie/rdunode60904.vlab.local@VLAB.LOCAL
spark/rdunode60904.vlab.local@VLAB.LOCAL
yarn/rdunode60904.vlab.local@VLAB.LOCAL
zookeeper/rdunode60904.vlab.local@VLAB.LOCAL
```

The Isilon OneFS cluster should now be completely Kerberized.

**Note**

You can view and edit environment-specific Kerberos settings in the OneFS web administration interface under the Kerberos Settings tab as shown in the following screen.

8. Create any necessary proxy users using the instructions in the following blog article as shown:
c) Create any necessary proxy users

In unsecured clusters, any user can impersonate any other user. In secured clusters, proxy users need to be explicitly specified. If you have Hive or Oozie, add the appropriate proxy users.

isi hdfs proxyusers create oozie --zone=<isilon_zone> --add-user=ambri-qa
isi hdfs proxyusers create hive --zone=<isilon_zone> --add-user=ambri-qa

---

**Note**

If you used the create users script described earlier in this document, you do not need to add proxy users.

9. Enable Kerberos on the HDFS zone. Change the HDFS access to KRB-only by running the following command on the Isilon OneFS cluster:

```bash
isi hdfs settings modify --zone=<zone-name> --authentication-mode=kerberos_only
```

10. View the HDFS settings.

```bash
isi hdfs settings view --zone=<zone-name>
```

For example:

```
# steven-45wenv-1# isi hdfs settings view --zone=zone2-cdh
Service: Yes
Default Block Size: 128M
Default Checksum Type: none
Authentication Mode: kerberos_only
Root Directory: /ifs/zone2/cdh/hadoop-root
WebHDFS Enabled: Yes
Ambi Server: -
Ambi NameNode: -
Oup Version: -
Data Transfer Cipher: none
Ambi Metrics Collector: -
```

11. Return to the **Configure Ports** screen in Cloudera Manager.

**Configure ports in Cloudera Manager**

1. On the **Configure Ports** screen, specify the default privileged ports and select **Yes, I am ready to restart the cluster now**. Click **Continue**.
2. The following screen displays during Kerberization.
3. Click **Continue** when the status displays as **Finished**.

The Kerberization is now complete.

**Test and validate Hadoop services**

In order to validate the Kerberized cluster, run the tests that are described in this section.

1. Run the following command.

```
hadoop fs -ls /
```

This command should fail since you do not have a valid Kerberos ticket. This is because the cluster is Kerberized, and Isilon enforces Kerberos-only access to HDFS root. This test also validates that simple authentication is not supported.
Sample output is shown:

```
2. Authenticate and obtain a valid Kerberos ticket.

   kinit
   klist -e
```

```
kuser@rdvnode61384:~ $ kinit
Password for kuser@VLAB.LOCAL:
KDC has rejected your request; try again...
```

```
kuser@rdvnode61384:~ $ klist -e
Valid starting     Expires         Service_principal
                  04/10/17 10:14:11  krbtgt/VLAB.LOCAL@VLAB.LOCAL
```

```
3. Create a test file to test file writes, and then list the contents of the HDFS directory. For example:

   hadoop fs -touchz THIS_IS_ISILON_zone2-cdh.txt
   hadoop fs -ls /

   Output similar to the following displays:
```

```
kuser@rdvnode61384:~ $ hadoop fs -ls /
Found 5 items:
-rwxr-xr-x  3 root wheel    0 2017-04-06 09:37 /THIS_IS_ISILON_zone2-cdh.txt
```

```
4. Run a simple Hadoop job to test the file system and Mapreduce framework. For example:

   hadoop jar /opt/cloudera/parcels/CDH/jars/Hadoop-mapreduce-examples-2.6.0-cdh5.10.1.jar teragen 10000 /user/kduser1/gen01

   Output similar to the following displays:
```
Troubleshoot services

Troubleshoot each of the failed services individually. View and drilldown into errors and configuration issues on the Cloudera Manager dashboard. Search through the Cloudera community forum to troubleshoot any service configuration issues.

In this example, Cloudera Manager detected one Hue error and one Yarn configuration issue as shown:
You may also see exception errors similar to the following if services have failed:

```
```

**Example 1—Yarn issues**

In this example, the Yarn configuration issue requires a Cloudera configuration change. Note that “Enable Kerberos Authentication for HTTP Web-Consoles” issues should be modified or suppressed depending on your system integration.

**Example 2—Hive Canary service**

If the Hive Canary service reports an issue, review the OneFS `hdfs.log` file for the following error. This is a missing proxy user issue. To resolve it, add Hue to the Hive proxy user group.
**Example 3—Hue issues**

In this example, Hue had an issue with the Hue ticket renewer service. See the following Cloudera community forum for details on this issue: [https://community.cloudera.com/t5/Web-UI-Hue-Beeswax/Kerberos-ticket-renewer-failed-to-start-Below-is-the-log-file/td-p/48822](https://community.cloudera.com/t5/Web-UI-Hue-Beeswax/Kerberos-ticket-renewer-failed-to-start-Below-is-the-log-file/td-p/48822).

1. To troubleshoot, run the following `modprinc` commands on the KDC to enable renewable tickets and resolve the two principals.

**Example:** Run the `get principal` command:

```
kadmin.local: get_principal krbtgt/VLAB.LOCAL
```

**Example output**

```
Principal: krbtgt/VLAB.LOCAL@VLAB.LOCAL
Expiration date: [never]
Last password change: [never]
Password expiration date: [none]
Maximum ticket life: 1 day 00:00:00
Maximum renewable life: 0 days 00:00:00
Last modified: Thu Apr 06 12:24:15 PDT 2017 (db_creation@VLAB.LOCAL)
Last successful authentication: [never]
Last failed authentication: [never]
Failed password attempts: 0
Number of keys: 7
Key: vno 1, aes256-cts-hmac-sha1-96, no salt
Key: vno 1, aes128-cts-hmac-sha1-96, no salt
Key: vno 1, des3-cbc-shal, no salt
Key: vno 1, arcfour-hmac, no salt
Key: vno 1, des-hmac-shal, no salt
Key: vno 1, des-cbc-md5, no salt
Key: vno 1, des-cbc-crc, no salt
MKey: vno 1
Attributes:
Policy: [none]
```
Example: Modify the principal with the modprinc command:

```
admin.local: modprinc -maxrenewlife 90day krbtgt/VLAB.LOCAL
```

Example output

```
Principal "krbtgt/VLAB.LOCAL@VLAB.LOCAL" modified.
```

Example: Run the get principal command again:

```
admin.local: get_principal krbtgt/VLAB.LOCAL
```

Example output

```
Principal: krbtgt/VLAB.LOCAL@VLAB.LOCAL
Expiration date: [never]
Last password change: [never]
Password expiration date: [none]
Maximum ticket life: 1 day 00:00:00
Maximum renewable life: 90 days 00:00:00
Last modified: Mon Apr 10 10:06:58 PDT 2017 (cloudera-scm/admin@VLAB.LOCAL)
Last successful authentication: [never]
Failed password attempts: 0
Number of keys: 7
  Key: vno 1, aes256-cts-hmac-shal-96, no salt
  Key: vno 1, aes128-cts-hmac-shal-96, no salt
  Key: vno 1, des3-cbc-shal, no salt
  Key: vno 1, arcfour-hmac, no salt
  Key: vno 1, des-hmac-shal, no salt
  Key: vno 1, des-cbc-md5, no salt
  Key: vno 1, des-cbc-crc, no salt
MKey: vno 1
Attributes:
Policy: [none]
```

Example: Run the get principal command on the Hue principal:

```
admin.local: get_principal hue/rduvnode61384.vlab.local@VLAB.LOCAL
```

Example output

```
Principal: hue/rduvnode61384.vlab.local@VLAB.LOCAL
Expiration date: [never]
Last password change: Mon Apr 10 08:38:25 PDT 2017
Password expiration date: [none]
Maximum ticket life: 1 day 00:00:00
Maximum renewable life: 5 days 00:00:00
Last modified: Mon Apr 10 08:38:25 PDT 2017 (cloudera-scm/admin@VLAB.LOCAL)
Last successful authentication: [never]
```
Last failed authentication: [never]
Failed password attempts: 0
Number of keys: 6
Key: vno 2, aes256-cts-hmac-shal-96, no salt
Key: vno 2, aes128-cts-hmac-shal-96, no salt
Key: vno 2, des3-cbc-shal, no salt
Key: vno 2, arcfour-hmac, no salt
Key: vno 2, des-hmac-shal, no salt
Key: vno 2, des-cbc-md5, no salt
MKey: vno 1
Attributes:
Policy: [none]

Example: Run the `modprinc` command on the Hue principal:
```
$ kadmin.local: modprinc -maxrenewlife 90day +allow_renewable hue/rduvnode61384.vlab.local@VLAB.LOCAL
```

Example output
```
Principal "hue/rduvnode61384.vlab.local@VLAB.LOCAL" modified.
```

Example: Run the `get principal` command on the Hue principal:
```
$ kadmin.local: get_principal hue/rduvnode61384.vlab.local@VLAB.LOCAL
```

Example output
```
Principal: hue/rduvnode61384.vlab.local@VLAB.LOCAL
Expiration date: [never]
Last password change: Mon Apr 10 08:38:25 PDT 2017
Password expiration date: [none]
Maximum ticket life: 1 day 00:00:00
Maximum renewable life: 90 days 00:00:00
Last modified: Mon Apr 10 09:53:26 PDT 2017 (cloudera-scm/admin@VLAB.LOCAL)
Last successful authentication: [never]
Last failed authentication: [never]
Failed password attempts: 0
Number of keys: 6
Key: vno 2, aes256-cts-hmac-shal-96, no salt
Key: vno 2, aes128-cts-hmac-shal-96, no salt
Key: vno 2, des3-cbc-shal, no salt
Key: vno 2, arcfour-hmac, no salt
Key: vno 2, des-hmac-shal, no salt
Key: vno 2, des-cbc-md5, no salt
MKey: vno 1
Attributes:
Policy: [none]
2. Restart any failed services after troubleshooting. Upon restarting the failed services, the cluster and all HDFS services start running and the dashboard appears as shown:

![Dashboard Image]


```
cat /etc/krb5.conf
```

```
root@RUUVNDE60909:~ # cat /etc/krb5.conf
[libdefaults]
default_realm = VLAB.LOCAL
dns_lookup_kdc = false
dns_lookup_realm = false
ticket_lifetime = 86400
renew_lifetime = 604800
forwardable = true
default_tgs_enctypes = rc4-hmac aes256-cts-hmac-sha1-96
default_tkt_enctypes = rc4-hmac aes256-cts-hmac-sha1-96
permitted_enctypes = rc4-hmac aes256-cts-hmac-sha1-96
udp_preference_limit = 1
kdc_timeout = 3000
[realms]
VLAB.LOCAL = {
    kdc = RUUVNDE60909.vlab.local
    admin_server = RUUVNDE60909.vlab.local
}
```

If you see issues when running Kerberized jobs, increase Kerberos logging to show further details.

This completes the installation and Kerberization of OneFS with Cloudera. For further assistance, contact Isilon Technical Support.
Contact Isilon Technical Support

Online Support: 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

Help with Online Support Tools:
For questions specific to HDFS, contact support@emc.com