

POWERSCALE ONEFS WITH HADOOP AND CLUDERA FOR KERBEROS INSTALLATION GUIDE

8.1.2 – 9.0.0

Abstract

This guide walks you through the process of installing PowerScale OneFS with the Cludera for Kerberos distribution of Hadoop.



Copyright © 2020 Dell Inc. or its subsidiaries. All rights reserved.

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

THE INFORMATION IN THIS PUBLICATION IS PROVIDED “AS-IS.” DELL MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. USE, COPYING, AND DISTRIBUTION OF ANY DELL SOFTWARE DESCRIBED IN THIS PUBLICATION REQUIRES AN APPLICABLE SOFTWARE LICENSE.

Dell, EMC, and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be the property of their respective owners.

Published in the USA.

EMC Corporation
Hopkinton, Massachusetts 01748-9103
1-508-435-1000 In North America 1-866-464-7381
www.EMC.com

Publication History

Version	Date	Description
1.00	April 28, 2017	Initial version
1.01	July 31, 2017	<ul style="list-style-type: none"> Updated list of users that the <code>lsilon_hadoop_tools</code> 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 “<code>--fixperm</code>” flag when using the <code>lsilon_create_users.sh</code> script. That flag is intended for user on existing deployments where directory ownership is wrong on OneFS.
1.03	October 5, 2017	<ul style="list-style-type: none"> Added “Updates and Additional Information on Isilon Hadoop Installs” section.
2.00	August 19, 2019	<ul style="list-style-type: none"> Added OneFS 8.1.2 related content
3.00	June 16, 2020	<ul style="list-style-type: none"> Added OneFS 8.2 - 9.0 related content.

CONTENTS

Introduction	6
Audience.....	6
Overview.....	6
Updates and additional information about OneFS Hadoop installs.....	8
Prerequisites.....	8
Cloudera distribution with Apache Hadoop (CDH)	8
OneFS cluster configuration	8
Install OneFS with Cloudera Manager	9
Preparing OneFS.....	9
Validate OneFS version and license activation	10
Configure OneFS components	10
Create an Access Zone	11
Configure SmartConnect	12
Configure DNS for OneFS.....	13
Verify the SmartConnect configuration.....	13
Create HDFS users and groups.....	14
Create users and directories on the OneFS cluster using Tools for Using Hadoop with OneFS	14
Create users on the OneFS cluster manually	14
Create and configure HDFS for OneFS 8.1.2 and previous versions.....	15
Configure HDFS user for OneFS 8.2 and later versions	16
Preparing Cloudera.....	17
Deploy Cloudera Manager	17
Configure the Hadoop cluster	23
Troubleshoot Cloudera Manager.....	31
Verify the deployment	33
Configure Kerberos with OneFS and Cloudera Manager	35
Prerequisites	35
OneFS.....	35
Cloudera Manager.....	36
Enable Kerberos with OneFS on Cloudera using Active Directory authentication	36
How Kerberos is implemented on the OneFS and Hadoop clusters	37
Preconfigure Cloudera Manager	37
Prepare hosts for Kerberization	38

Configure OneFS for Kerberos authentication	38
Review the OneFS SPNs	40
Create proxy users.....	40
Enable Kerberos authentication in Cloudera Manager.....	41
Enable Kerberos with OneFS on Cloudera using MIT KDC	54
Prepare hosts for Kerberization	54
Enable CDH Kerberos using MIT	55
Create the KDC as a OneFS authorization provider	60
Configure ports in Cloudera Manager	66
Test and validate Hadoop services	68
Troubleshoot services	70
Contacting Dell EMC PowerScale Technical Support.....	75

Introduction

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

Before you begin, you must install a PowerScale OneFS cluster.

Audience

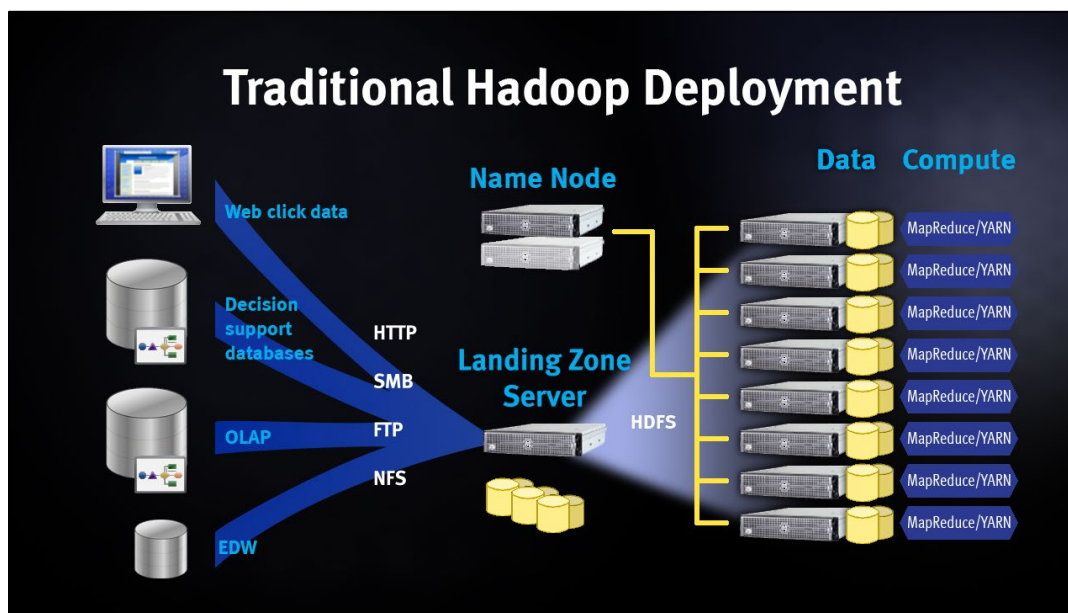
This guide is intended for systems administrators, IT program managers, IT architects, and IT managers who are installing Isilon OneFS with a Cloudera distribution of Hadoop.

Overview

The PowerScale 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. A PowerScale cluster powered by the OneFS operating system delivers a scalable pool of storage with a global namespace.

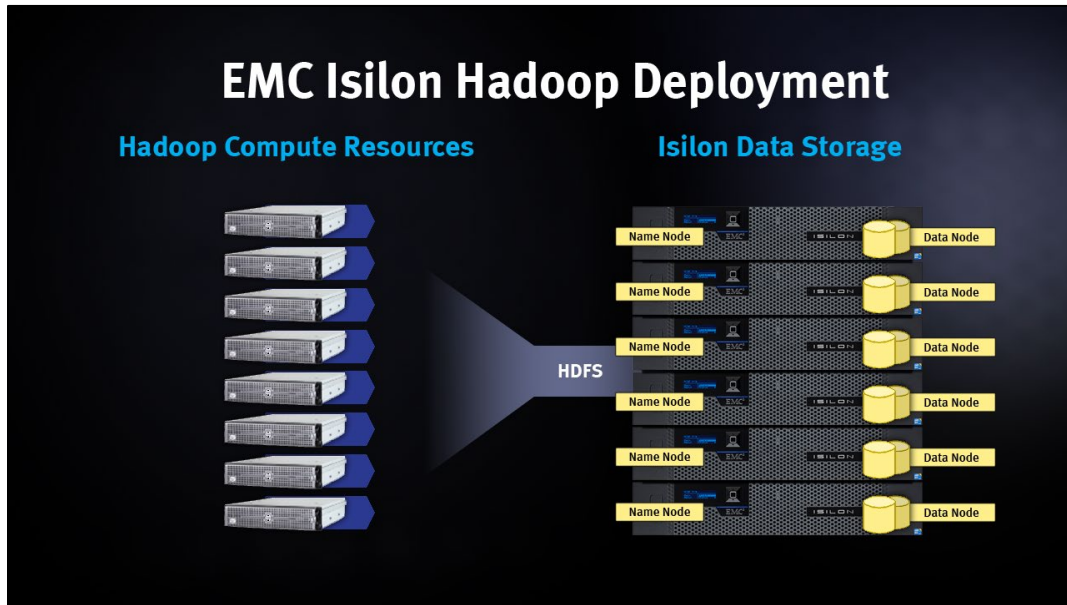
Hadoop compute clients can access the data that is stored on a PowerScale OneFS cluster by connecting to any node over the HDFS protocol. All nodes that are configured for HDFS provide NameNode and DataNode functionality. Each node boosts performance and expands the cluster capacity. For Hadoop analytics, the PowerScale 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 before it is loaded on to HDFS. This process is network intensive and exposes the NameNode as a potential single point of failure.



In a PowerScale OneFS with Hadoop deployment, OneFS serves as the file system for Hadoop compute clients. On a PowerScale 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 access the data that is stored on a OneFS cluster by connecting over HDFS. The HDFS protocol is native to the 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 OneFS 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 about OneFS Hadoop installs

The rapid release of new features and versions of Hadoop projects can introduce new behaviors and requirements. It is recommended that you review the latest updates on the [Using Hadoop with Isilon - Isilon Info Hub](#) for updates and known issues while deploying OneFS and Hadoop.

Prerequisites

For supported versions, see [Hadoop Distributions and Products Supported by OneFS](#).

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

Component	Version
Cloudera Manager version	
CDH parcel version	
Cloudera server (FQDN)	

OneFS cluster configuration

Ensure that the following requirements are met:

- A OneFS cluster running OneFS 8.1.2.0 or later.
- SmartConnect Advanced, a separately licensed OneFS module, is activated and SmartConnect is configured on your OneFS cluster.
- HDFS, a separately licensed OneFS module, is activated on your OneFS cluster. Contact your Dell EMC PowerScale sales representative for more information about receiving your license keys.
- A valid OneFS 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](#).
- A dedicated OneFS Access Zone is in use; this is not the same as the System Zone.
- A OneFS HDFS root directory in the Access Zone.
- A simple access model between Hadoop and OneFS; UID and GUID, with parity.
- Use the following table to record the components that you have installed.

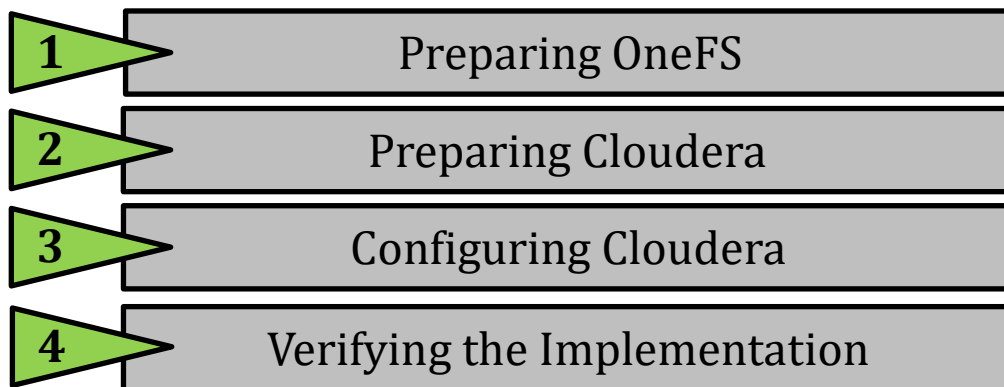
Component	Version or License
PowerScale OneFS	

Component	Version or License
SmartConnect module	
HDFS module	
OneFS cluster name	

Install OneFS with Cloudera Manager

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

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



Preparing OneFS

Complete the following steps to configure your 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.

Review the current [Isilon OneFS and Hadoop Known Issues](#) for any changes or updates to OneFS and Hadoop configuration.

Validate OneFS version and license activation

Validate your OneFS version, check your licenses, and confirm that they are activated. Other OneFS licenses may be needed for additional OneFS functionality to be interoperable with HDFS, they are not addressed in this installation guide.

1. From a node in your OneFS cluster, confirm that your OneFS cluster is running OneFS 8.1.2 or later by typing the following command:

```
isi version
```

2. Add the licenses for HDFS using the following command:

```
isi license add --evaluation=HDFS
```

3. Confirm that license for HDFS is operational. If this license is not active and valid, some commands in this guide will not work.

Run the following commands to confirm that HDFS is installed:

```
isi license licenses list  
isi license licenses view HDFS
```

4. If your modules are not licensed, obtain a license key from your Dell EMC PowerScale sales representative. Type the following command to activate the license:

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

5. Enable HDFS by running the following command:

```
isi services hdfs enable
```

6. Install the latest rollup patches for your version of OneFS. See [Current Isilon OneFS Patches](#) for the latest rollup patches and run the following:

```
isi upgrade patches list  
isi upgrade patches install patch-<patch-ID>.pkg --rolling=false
```

Example:

```
isi upgrade patches install patch-240163.pkg --rolling=false
```

Configure OneFS components

After you configure DNS for OneFS, set up and configure the following 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 with Cloudera integration:

Parameter	Value
Access zone name	
Access zone path	
SmartConnect zone name (FQDN)	
IP range for IP pool (ranges)	
SmartConnect pool name (subnet pool)	
Node and interfaces in the pool	
HDFS root path	

Create an Access Zone

On one of the 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, create your Hadoop access zone:

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

2. Verify that the access zones are set up correctly:

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

Output similar to the following displays:

```
Name: System
Path: /ifs
Groupnet: groupnet0
Map Untrusted: -
Auth Providers: lsa-local-provider:System, lsa-file-provider:System
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/data/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/data/zone1/cdh/hadoop-root
isi hdfs settings modify --zone=zone1-hdp --root-
directory=/ifs/data/zone1/cdh/hadoop-root
```

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

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

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.

1. Modify your existing subnets and specify a service address:

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

2. Create an access network pool, run the following command, where:

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

```
isi network pools create --id=<groupnet>:<subnet>:<name> --ranges=<IP-IP> --
access-zone=<access-zone> --alloc-method=static --ifaces=<interfaces> --sc-
subnet=<subnet> --sc-dns-zone=<smartconnectzone> --description=hadoop
```

For example:

```
isi network pools create groupnet0:subnet0:hadoop-pool-cdh --
ranges=10.120.130.30-10.120.140.40 --access-zone=zone1-cdh --alloc-
method=static --ifaces=1-4:40gige-1 --sc-subnet=subnet0 --sc-dns-
zone=cdh.zone1.emc.com --description=hadoop"
```

3. View the properties of the access network pool:

```
isi network pools view --id=groupnet0:subnet0:pool2
```

Output similar to the following displays:

```
ID: groupnet0.subnet0.hadoop-pool-cdh
Groupnet: groupnet0
Subnet: subnet0
Name: hadoop-pool-cdh
Rules: -
Access Zone: zone1-cdh
Allocation Method: static
```

```

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: 10.120.130.30-10.120.140.40
  Rebalance Policy: auto
SC Auto Unsuspend Delay: 0
  SC Connect Policy: round_robin
    SC Zone: cdh.zone1.emc.com
SC DNS Zone Aliases: -
  SC Failover Policy: round_robin
    SC Subnet: subnet0
    SC Ttl: 0
  Static Routes: -

```

Configure DNS for OneFS

Before you begin, the OneFS cluster must already be implemented according to Dell EMC PowerScale 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. 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 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 replies 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 connects to a OneFS 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 node that is 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.

Verify the SmartConnect configuration

Validate that SmartConnect is set up correctly by pinging the SmartConnect zone FQDN several times from the Hadoop client.

```
ping cdh.zone1.emc.com
```

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 HDFS users and groups

For each Hadoop system account that submits HDFS jobs or access the file system, you must create local users and groups on the OneFS cluster. You can add Hadoop users and groups to the OneFS cluster manually or by following the process at: https://github.com/Isilon/isilon_hadoop_tools

Important

Dell EMC PowerScale 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 multiprotocol 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 OneFS multiprotocol file access.

During installation of Hadoop with Cloudera Manager, the installer creates all 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 creates these accounts if they do not exist. You can ensure parity by precreating them on all nodes that to be installed in the Hadoop cluster. You 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 OneFS, they must be created manually. Create the OneFS *yarn* local account user in the OneFS access zone in which *yarn* accesses data. 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 article: [Isilon and Hadoop Local User UID Parity](#)

There are many methods of achieving UID and GUID parity. You can leverage [Tools for Using Hadoop with OneFS](#), 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. It is highly recommended that you maintain consistency between the Hadoop cluster and 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 Tools for Using Hadoop with OneFS

Go to Tools for Using Hadoop with OneFS to set up the users and directories on the cluster.

Create users on the OneFS cluster manually

You can add a user for each additional Hadoop user that submits MapReduce jobs in addition to the users that the OneFS script configures on the OneFS cluster. The following procedures 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.

Manual 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/data/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/data/hadoop/user/hduser1
```

Manual steps to perform on the Hadoop client

Since you created a user on OneFS to run jobs, you must create the same user with UID parity on any Linux hosts that the user accesses to run jobs.

1. Add the user to the Hadoop cluster.

```
adduser hduser1 -u <UID>
```

Create and configure HDFS for OneFS 8.1.2 and previous versions

In OneFS 8.1.2 and previous versions, the HDFS user must be mapped to root and you must modify the access control list (ACL).

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

1. View the HDFS service settings.

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

2. Set the HDFS root directory for the access zone. **Note:** It is recommended that the directory for the access zone is not set to the root of /ifs.

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

3. Map the HDFS user to 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.

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

5. Modify the access control list (ACL) settings for OneFS.

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.

```
isi auth settings acls modify --group-owner-inheritance=parent
isi auth settings view
```

Configure HDFS user for OneFS 8.2 and later versions

In OneFS 8.2.0, the HDFS user no longer must be mapped to root. Instead a new role with backup and restore privileges must be assigned as follows:

On a node in the OneFS 8.2 cluster, create a role and configure the backup and restore privileges to the HDFS user.

1. View the HDFS service settings.

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

2. Set the HDFS root directory for the access zone. **Note:** It is recommended that the directory for the access zone is not set to the root of /ifs.

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

3. Create a role for the Hadoop access zone.

```
isi auth roles create --name=<role_name> --description=<role_description> --
zone=<access_zone>
```

For example:

```
isi auth roles create --name=HdfsAccess --description="Bypass FS permissions" --
zone=zone1-cdh
```

4. Add restore privileges to the new “HdfsAccess” role.

```
isi auth roles modify <role_name> --add-priv=ISI_PRIV_IFS_RESTORE --
zone=<access_zone>
```

For example:

```
isi auth roles modify HdfsAccess --add-priv=ISI_PRIV_IFS_RESTORE --zone=zone1-cdh
```

5. Add backup privileges to the new “HdfsAccess” role.

```
isi auth roles modify <role_name> --add-priv=ISI_PRIV_IFS_BACKUP --
zone=<access_zone>
```

For example:

```
isi auth roles modify HdfsAccess --add-priv=ISI_PRIV_IFS_BACKUP --zone=zone1-cdh
```

6. Add user hdfs to the new “HdfsAccess” role.

```
isi auth roles modify <role_name> --add-user=hdfs --zone=<access_zone>
```

For example:

```
isi auth roles modify HdfsAccess --add-user=hdfs --zone=zone1-cdh
```


7. Verify the role setup, backup and restore privileges, and HDFS user setup.

```
isi auth roles view <role_name> --zone=<access_zone>
```

For example:

```
isi auth roles view HdfsAccess --zone=zone1-cdh
Name: HdfsAccess
Description: Bypass FS permissions
Members: - hdfs
Privileges
ID: ISI_PRIV_IFS_BACKUP
Read Only: True
ID: ISI_PRIV_IFS_RESTORE
Read Only: True
```

8. (Optional) Flush auth mapping and auth cache to make the HDFS user take immediate effect as the “HdfsAccess” role that you created above.

```
isi_for_array "isi auth mapping flush --all"
isi_for_array "isi auth cache flush --all"
```

Note

ACL Policies no longer must be modified for OneFS 8.2 and later as the hdfs protocols act the same as non-OneFS HDFS for File System Group Owner inheritance.

Preparing 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 the other 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 Cloudera requirements 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 [Hadoop Distributions and Products Supported by OneFS](#) for Cloudera Manager and CDH compatibility.

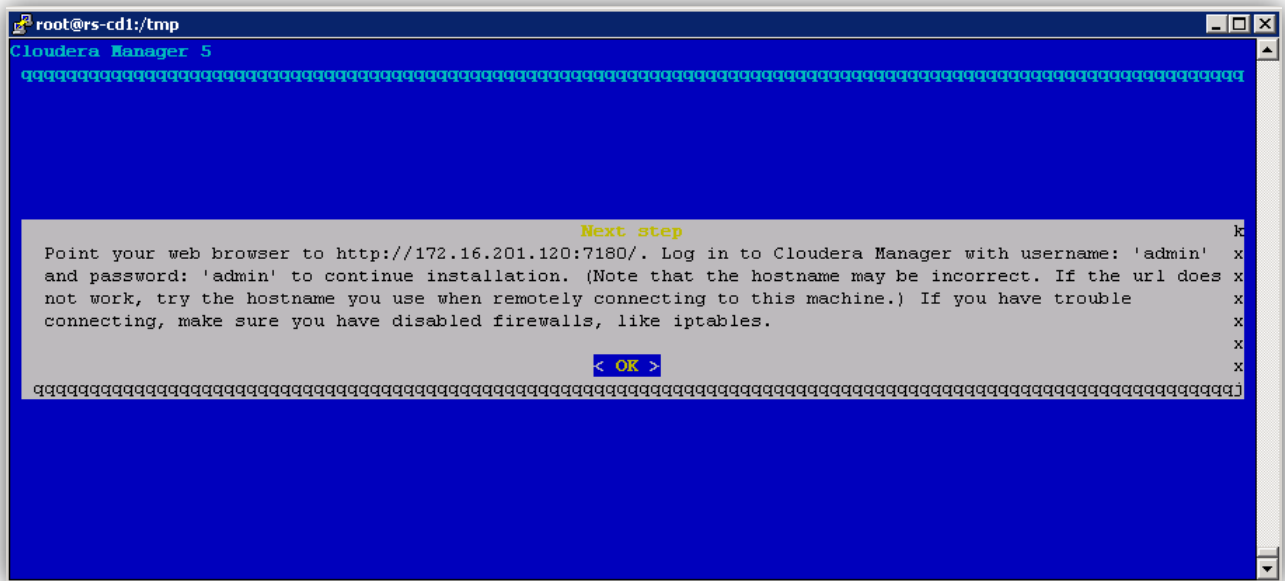
1. Run the following to install Cloudera.

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

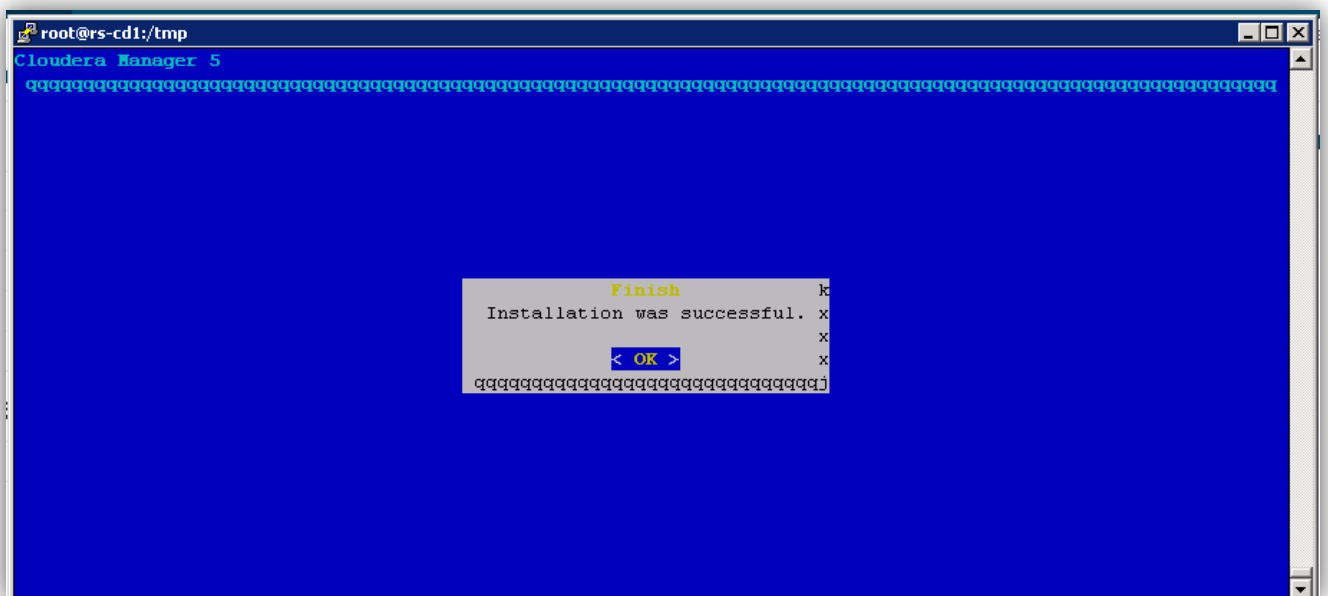
- The following displays when running the installer:

- Review the Cloudera license and accept the license agreement.
- Accept the Oracle Binary Code license agreement.
- Review and accept the Oracle Binary Code license agreement.
- Install the Oracle JDK.

7. After you install the Oracle JDK, the following screen appears. Note the URL and the username and 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>:7180>) with the following credentials: **user:** admin; **password:** admin and accept the EULA.

11. Select the version of Cloudera that you want to deploy.

Welcome to Cloudera Manager

Which edition do you want to deploy?

Upgrading to **Cloudera Enterprise Data Hub Edition** provides important features that help you manage and monitor your Hadoop clusters in mission-critical environments.

	Cloudera Express	Cloudera Enterprise Data Hub Edition Trial	Cloudera Enterprise
License	Free	60 Days After the trial period, the product will continue to function as Cloudera Express . Your cluster and your data will remain unaffected.	Annual Subscription <input type="button" value="Select License File"/> <input type="button" value="Upload"/>
Node Limit	Unlimited	Unlimited	Unlimited
CDH	✓	✓	✓
Core Cloudera Manager Features	✓	✓	✓
Advanced Cloudera Manager Features		✓	✓

Cloudera Enterprise is available in three editions:

- Basic Edition
- Flex Edition
- Data Hub Edition

1 2

12. Specify the hosts for your CDH cluster installation. In this guide, you 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 identify themselves with.
 - Select **Use Parcels (Recommended)**, the CDH stack that you want to deploy, any additional parcels, and then click **Continue**.

Cluster Installation

Select Repository

Cloudera recommends the use of parcels for installation over packages, because parcels enable Cloudera Manager to easily manage the software on your cluster, automating the deployment and upgrade of service binaries. Electing not to use parcels will require you to manually upgrade packages on all hosts in your cluster when software updates are available, and will prevent you from using Cloudera Manager's rolling upgrade capabilities.

Choose Method

☐ Use Packages
 ☒ Use Parcels (Recommended)
 More Options

Select the version of CDH

☒ CDH-5.10.1-1.cdh5.10.1.p0.10
☐ CDH-4.7.1-1.cdh4.7.1.p0.47
Versions of CDH that are too new for this version of Cloudera Manager (5.10.1) will not be shown.

Additional Parcels

☐ ACCUMULO-1.7.2-5.5.0.ACCUMULO5.5.0.p0.8
☐ ACCUMULO-1.4.4-1.cdh4.5.0.p0.65
☒ None
☐ KAFKA-2.1.1-1.2.1.1.p0.18
☒ None
☐ SQOOP_NETEZZA_CONNECTOR-1.5c5
☒ None

Back
1 2 3 4 5 6 7
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.

Provide SSH login credentials.

Root access to your hosts is required to install the Cloudera packages. This installer will connect to your hosts via SSH and log in either directly as root or as another user with password-less sudo/pbrun privileges to become root.

Login To All Hosts As: ☒ root
☐ Another user

You may connect via password or public-key authentication for the user selected above.

Authentication Method: ☒ All hosts accept same password
☐ All hosts accept same private key

Enter Password:

Confirm Password:

SSH Port:

Number of Simultaneous Installations: (Running a large number of installations at once can consume large amounts of network bandwidth and other system resources)

16. Wait for the installation to complete and then click **Continue**.

Cluster Installation

Installation completed successfully.

1 of 1 host(s) completed successfully.

Hostname	IP Address	Progress	Status
RDUVNODE00004.vlab.local	10.09.37.150	<div style="width: 100%; height: 10px; background-color: green;"></div>	Installation completed successfully. Details

Note

The installer checks and validates hosts. If the validation check fails, follow the recommendations to resolve and then retry the validation. Common errors are the following, which are all related to Linux (not OneFS).

- Transparent_hugepage
- Swappiness

On successful completion of the host inspector, 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 OneFS. This is key to the OneFS integration. If you select anything other than **Custom Services**, you cannot install OneFS.

Choose the CDH 5 services that you want to install on your cluster.

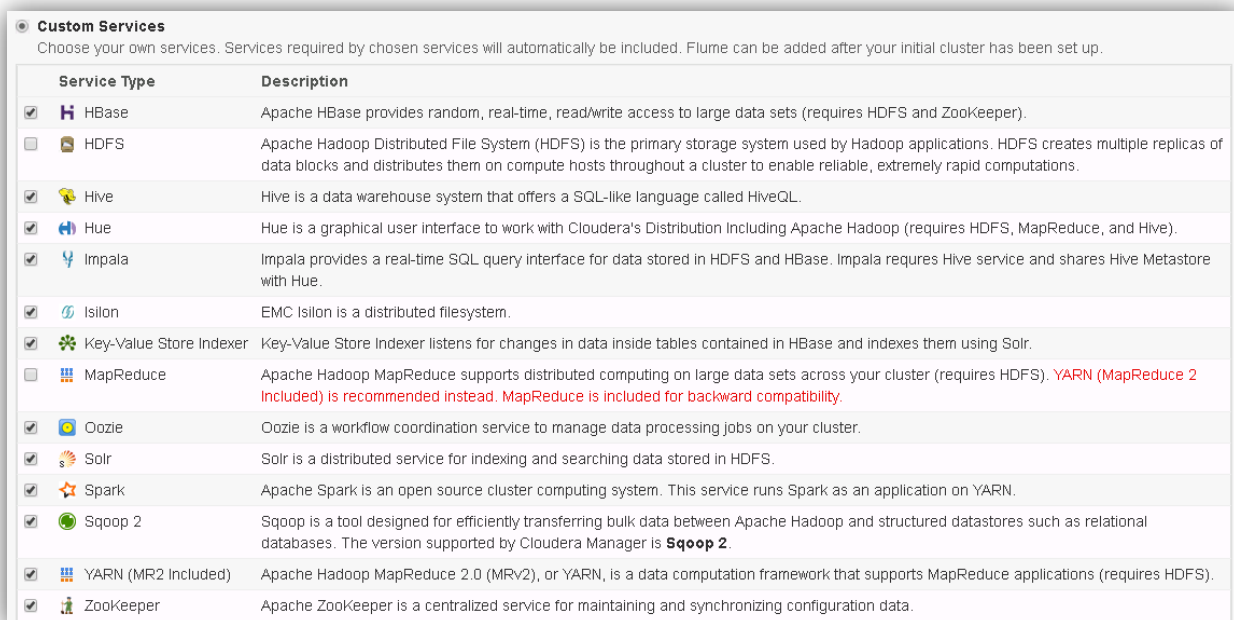
Choose a combination of services to install.

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

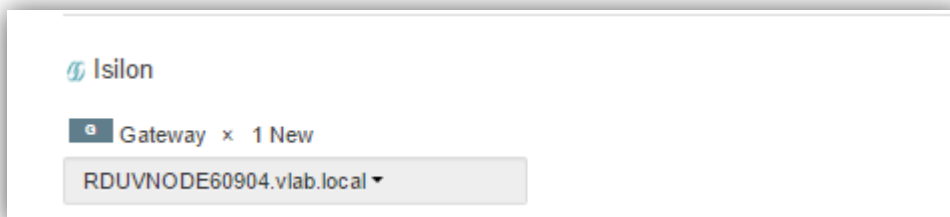
- Once you select **Custom Services**, select the Hadoop services that you want to deploy.

Important

- Do not select HDFS.** The Cloudera HDFS service is not needed.
- Select **Isilon** as the storage.
- It is not recommended to select MapReduce, as MapReduce2 is depreciated and is in Yarn. If you have a legacy application and it is not written for Yarn, MapReducev1 can be enabled.



- After you have selected the Hadoop services that 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.
- Select the Isilon Gateway role on the same host that is running Cloudera Manager, and then click **Continue**.



8. Continuing with the **Cluster Setup**, assign the OneFS cluster to the following parameters. The parameters for the SmartConnect zone ports are different:

```
default_fs_name      hdfs://smartconnectzonename:8020
webhdfs_url          http://smartconnectzonename:8082/webhdfs/v1
```

Cluster Setup

Review Changes

HDFS Root Directory hbase.rootdir	Cluster 1 > HBase (Service-Wide) /hbase	?
Enable Indexing	Cluster 1 > HBase (Service-Wide) <input checked="" type="checkbox"/> ↕	?
Enable Replication hbase.replication	Cluster 1 > HBase (Service-Wide) <input checked="" type="checkbox"/> ↕	?
Hive Warehouse Directory hive.metastore.warehouse.dir	Cluster 1 > Hive (Service-Wide) /user/hive/warehouse	?
Hive Metastore Server Port hive.metastore.port	Cluster 1 > Hive Metastore Server Default Group 9083	?
Impala Daemon Scratch Directories scratch_dirs	Cluster 1 > Impala Daemon Default Group /impala/impalad + -	?
HDFS Block Size dfs.block.size, dfs.blocksize	Cluster 1 > Isilon (Service-Wide) 128 MiB ▼	?

Default File System URI default_fs_name	Cluster 1 > Isilon (Service-Wide) <div style="border: 1px solid red; height: 15px; width: 100%;"></div> Missing required value: Default File System URI	?
WebHDFS URL webhdfs_url	Cluster 1 > Isilon (Service-Wide) <div style="border: 1px solid red; height: 15px; width: 100%;"></div>	?
Alerts: Mail Server Hostname	Alert Publisher Default Group localhost	?
Alerts: Mail Server Username	Alert Publisher Default Group <div style="border: 1px solid red; height: 15px; width: 100%;"></div>	?
Alerts: Mail Server Password	Alert Publisher Default Group <div style="border: 1px solid red; height: 15px; width: 100%;"></div>	?
Alerts: Mail Message Recipients	Alert Publisher Default Group root@localhost	?
Custom Alert Script alert.script.path	Alert Publisher Default Group <div style="border: 1px solid red; height: 15px; width: 100%;"></div>	?
Host Monitor Storage Directory firehose.storage.base.directory	Host Monitor Default Group /var/lib/cloudera-host-monitor	?

Service Monitor Storage Directory firehose.storage.base.directory	Service Monitor Default Group /var/lib/cloudera-service-monitor	?
ShareLib Root Directory oozie.service.WorkflowAppService.system.lib path	Cluster 1 > Oozie (Service-Wide) /user/oozie	?
Oozie Server Data Directory	Cluster 1 > Oozie Server Default Group /var/lib/oozie/data	?
ZooKeeper Znode	Cluster 1 > Solr (Service-Wide) /solr	?
HDFS Data Directory	Cluster 1 > Solr (Service-Wide) /solr	?
Sqoop 2 Server Metastore Directory	Cluster 1 > Sqoop 2 Server Default Group /var/lib/sqoop2	?
Sqoop Repository Database Type	Cluster 1 > Sqoop 2 Server Default Group <input checked="" type="radio"/> Derby <input type="radio"/> PostgreSQL	?
Sqoop Repository Database Host	Cluster 1 > Sqoop 2 Server Default Group localhost	?

Sqoop Repository Database Name	Cluster 1 > Sqoop 2 Server Default Group sqoop	?
Sqoop Repository Database User org.apache.sqoop.repository.jdbc.user	Cluster 1 > Sqoop 2 Server Default Group sa	?
Sqoop Repository Database Password org.apache.sqoop.repository.jdbc.password	Cluster 1 > Sqoop 2 Server Default Group	?
NodeManager Local Directories yarn.nodemanager.local-dirs	Cluster 1 > NodeManager Default Group /yarn/nm + -	?
Enable Container Usage Metrics Collection	Cluster 1 > YARN (MR2 Included) (Service-Wide) <input type="checkbox"/>	?
Cloudera Manager Container Usage Metrics Directory	Cluster 1 > YARN (MR2 Included) (Service-Wide) /tmp/cmYarnContainerMetrics	?
Container Usage Output Directory	Cluster 1 > YARN (MR2 Included) (Service-Wide) /tmp/cmYarnContainerMetricsAggregate	?
Container Usage MapReduce Job User	Cluster 1 > YARN (MR2 Included) (Service-Wide)	?
Data Directory dataDir	Cluster 1 > Server Default Group /var/lib/zookeeper	?

Transaction Log Directory
dataLogDir

Cluster 1 > Server Default Group

?

Back

1 2 3 4 5 6

Continue

- Assign the two Isilon parameters. Leave all the default settings, and then click **Continue**.

Default File System URI
default_fs_name

Isilon (Service-Wide)

WebHDFS URL
webhdfs_url

Isilon (Service-Wide)

- Continue the setup and monitor deployment.

Cluster Setup

First Run Command

Status: **Running** Start Time: Jun 20, 4:58:08 PM

Abort

Details Completed 0 of 8 step(s)

☒ All
☐ Failed Only
☐ Running Only

Step	Context	Start Time	Duration	Actions
Deploy Client Configuration	Cluster 1	Jun 20, 4:58:08 PM		<div>Abort</div>
Start Cloudera Management Service, ZooKeeper, Isilon				
Start HBase, Solr				
Start YARN (MR2 Included), Key-Value Store Indexer				
Start Sqoop 2, Spark				
Start Hive				
Start Oozie, Impala				
Start Hue				

11. Review the cluster setup as it deploys.

Cluster Setup

First Run Command

Status: **Running** Start Time: Jun 20, 4:58:08 PM Abort

Details Completed 5 of 8 step(s). All Failed Only Running Only


Step	Context	Start Time	Duration	Actions
<div> <div> </div> <div> <div>✓</div> <div>Deploy Client Configuration</div> <div>Successfully deployed all client configurations.</div> </div> </div>	<div> <div>Cluster 1</div> </div>	Jun 20, 4:58:08 PM	16.79s	
<div> <div> </div> <div> <div>✓</div> <div>Start Cloudera Management Service, Zookeeper, Isilon</div> <div>Successfully completed 3 steps.</div> </div> </div>		Jun 20, 4:58:25 PM	42.4s	
<div> <div> </div> <div> <div>✓</div> <div>Start HBase, Solr</div> <div>Successfully completed 2 steps.</div> </div> </div>		Jun 20, 4:59:07 PM	2.5m	
<div> <div> </div> <div> <div>✓</div> <div>Start YARN (MR2 Included), Key-Value Store Indexer</div> <div>Successfully completed 2 steps.</div> </div> </div>		Jun 20, 5:01:35 PM	79.54s	
<div> <div> </div> <div> <div>✓</div> <div>Start Sqoop 2, Spark</div> <div>Successfully completed 2 steps.</div> </div> </div>		Jun 20, 5:02:55 PM	2.5m	
<div> <div> </div> <div> <div>Start Hive</div> <div>0/1 steps completed.</div> </div> </div>		Jun 20, 5:05:25 PM		
<div> <div> </div> <div> <div>Start Oozie, Impala</div> </div> </div>				
<div> <div> </div> <div> <div>Start Hue</div> </div> </div>				

Details Completed 6 of 8 step(s). All Failed Only Running Only

Step	Context	Start Time	Duration	Actions
<div> <div> </div> <div> <div>✓</div> <div>Deploy Client Configuration</div> <div>Successfully deployed all client configurations.</div> </div> </div>	<div> <div>Cluster 1</div> </div>	Jun 20, 4:58:08 PM	16.79s	
<div> <div> </div> <div> <div>✓</div> <div>Execute DeployClusterClientConfig for (spark_on_yarn,hive,hbase,solr,yarn,isilon) in parallel.</div> <div>Successfully completed 6 steps.</div> </div> </div>		Jun 20, 4:58:08 PM	16.79s	
<div> <div> </div> <div> <div>✓</div> <div>Deploy Client Configuration</div> <div>Successfully deployed client configuration.</div> </div> </div>	<div> <div>Spark</div> </div>	Jun 20, 4:58:08 PM	16.71s	
<div> <div> </div> <div> <div>✓</div> <div>Deploy Client Configuration</div> <div>Successfully deployed client configuration.</div> </div> </div>	<div> <div>Hive</div> </div>	Jun 20, 4:58:09 PM	16.34s	
<div> <div> </div> <div> <div>✓</div> <div>Deploy Client Configuration</div> <div>Successfully deployed client configuration.</div> </div> </div>	<div> <div>HBase</div> </div>	Jun 20, 4:58:09 PM	16.05s	
<div> <div> </div> <div> <div>✓</div> <div>Deploy Client Configuration</div> <div>Successfully deployed client configuration.</div> </div> </div>	<div> <div>Solr</div> </div>	Jun 20, 4:58:09 PM	15.72s	
<div> <div> </div> <div> <div>✓</div> <div>Deploy Client Configuration</div> <div>Successfully deployed client configuration.</div> </div> </div>	<div> <div>YARN (MR2 Included)</div> </div>	Jun 20, 4:58:09 PM	15.69s	
<div> <div> </div> <div> <div>✓</div> <div>Deploy Client Configuration</div> <div>The service has no roles.</div> </div> </div>	<div> <div>Isilon</div> </div>	Jun 20, 4:58:09 PM	0ms	

12. The setup will complete. 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.


First Run Command

Status: **Finished** Start Time: Jun 20, 4:58:08 PM Duration: 13.7m

Finished First Run of the following services successfully: Isilon, ZooKeeper, HBase, Solr, YARN (MR2 Included), Key-Value Store Indexer, Spark, Sqoop 2, Hive, Impala, Oozie, Hue, Cloudera Management Service.

Details Completed 8 of 8 step(s). All Failed Only Running Only

Step	Context	Start Time	Duration	Actions
  Deploy Client Configuration Successfully deployed all client configurations.	 Cluster 1 	Jun 20, 4:58:08 PM	16.79s	
  Start Cloudera Management Service, ZooKeeper, Isilon Successfully completed 3 steps.		Jun 20, 4:58:25 PM	42.4s	
  Start HBase, Solr Successfully completed 2 steps.		Jun 20, 4:59:07 PM	2.5m	
  Start YARN (MR2 Included), Key-Value Store Indexer Successfully completed 2 steps.		Jun 20, 5:01:35 PM	79.54s	
  Start Sqoop 2, Spark Successfully completed 2 steps.		Jun 20, 5:02:55 PM	2.5m	
  Start Hive Successfully completed 1 steps.		Jun 20, 5:05:25 PM	81.43s	
  Start Oozie, Impala Successfully completed 2 steps.		Jun 20, 5:06:47 PM	4.4m	
  Start Hue Successfully completed 1 steps.		Jun 20, 5:11:12 PM	35.95s	

Back

1 2 3 4 **5** 6

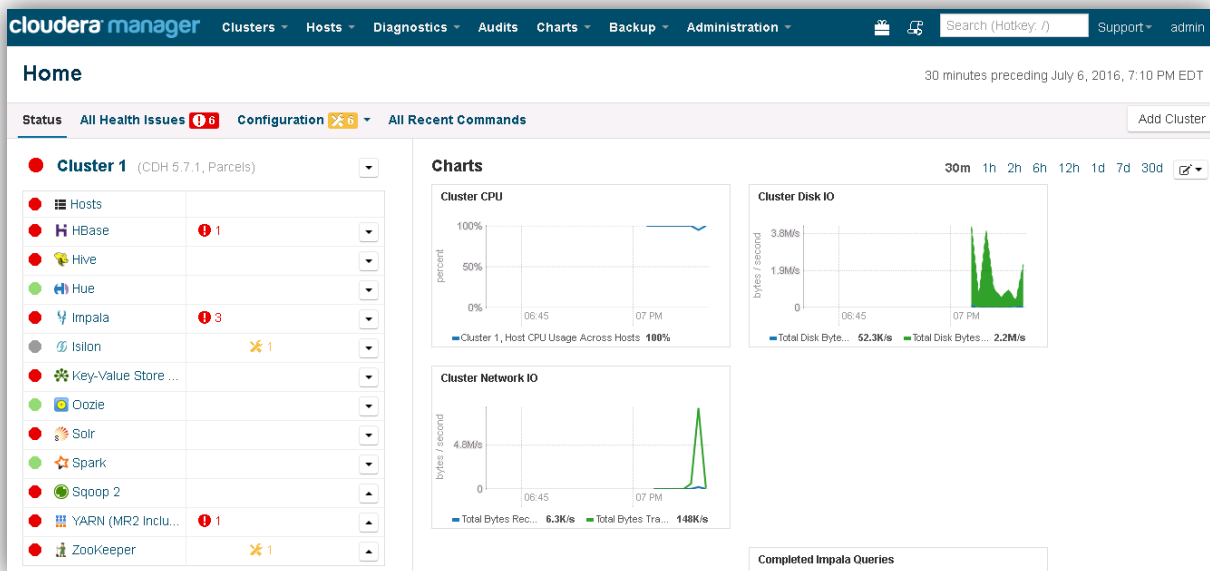
Continue

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 need restarting.

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

- Start the service.
- Monitor and review logs as needed.
- Review the `/var/log/hdfs.log` file on all nodes.
- Restart services to resolve alarms following configuration changes.



It is common to see configuration issues. Address the issues and make the required changes that are needed to resolve each issue.

Note

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

Home

Status All Health Issues **8** Configuration **6** All Recent Commands

All Configuration Issues

▼ **Cluster 1**

- ZooKeeper: [Service zookeeper has 1 Server. Cloudera suggests at least 3 Servers for Zookeeper.](#) Suppress...
- Isilon: [Use Trash](#) Suppress...
HDFS Trash is disabled. Deleted files will be unrecoverable.

▼ **Other**

- Cloudera Management Service: [Java Heap Size of Service Monitor in Bytes](#) Suppress...
The recommended heap size is 1.0 GiB bytes, 692.0 MiB more than is configured.
- Cloudera Management Service: [Maximum Non-Java Memory of Service Monitor](#) Suppress...
The recommended non-Java memory size is 1.5 GiB, 768.0 MiB more than is configured.
- Cloudera Management Service: [Java Heap Size of Host Monitor in Bytes](#) Suppress...
The recommended heap size is 1.0 GiB bytes, 692.0 MiB more than is configured.
- Cloudera Management Service: [Maximum Non-Java Memory of Host Monitor](#) Suppress...
The recommended non-Java memory size is 1.5 GiB, 768.0 MiB more than is configured.

Restart services as needed to resolve alarms following configuration changes.

Restart Command

Status: **Running** Context: [YARN \(MR2 Included\)](#) Start Time: Jul 6, 7:12:56 PM **Abort**

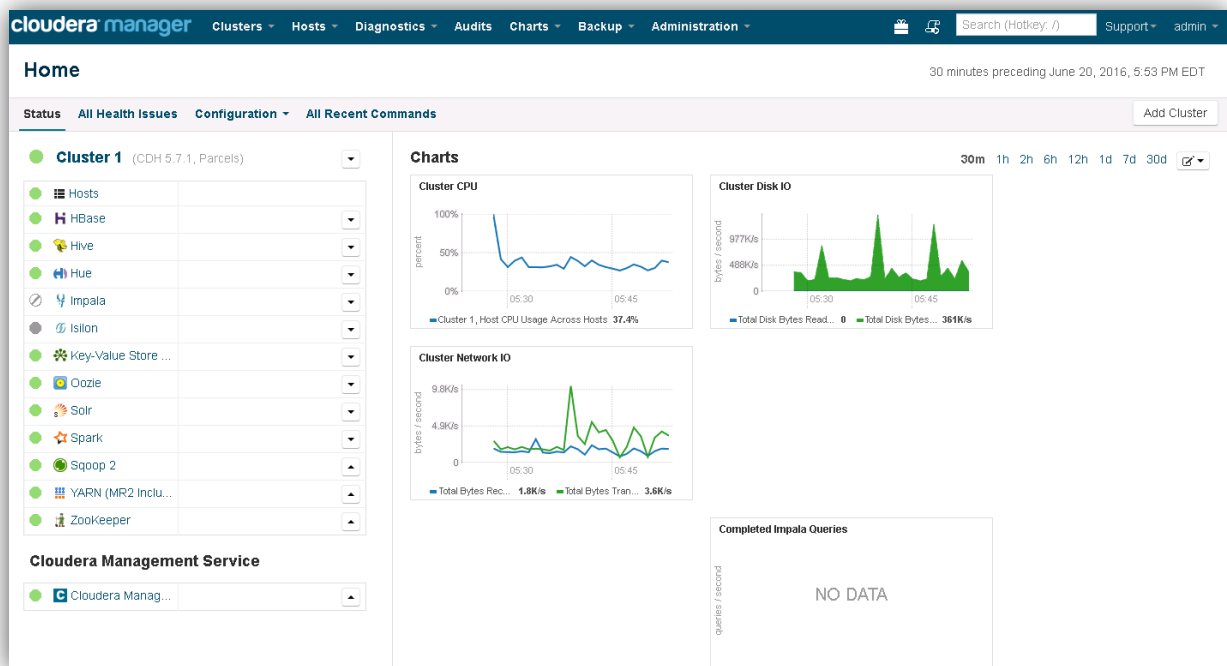
Details Completed 1 of 2 step(s).

☒ All ☐ Failed Only ☐ Running Only

Step	Context	Start Time	Duration	Actions
Stop Successfully stopped service.	YARN (MR2 Included)	Jul 6, 7:12:56 PM	8.53s	
Start	YARN (MR2 Included)	Jul 6, 7:13:04 PM		Abort
Execute command Start on service YARN (MR2 Included) Waiting for command (129) to finish Execute command Start on service YARN (MR2 Included)	YARN (MR2 Included)	Jul 6, 7:13:04 PM		Abort

Close

The Hadoop cluster and services are now fully operational and can be tested.



Verify the deployment

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

1. Browse the HDFS root.

```
hadoop fs -ls /
```

Output similar to the following displays:

```
Found 5 items
-rw-r--r--  3 root  wheel                0 2017-04-21 13:55 /THIS_IS_ISILON_zone1-
cdh.txt
drwxr-xr-x  - hbase hbase                0 2017-04-21 14:04 /hbase
drwxrwxr-x  - solr  solr                 0 2017-04-21 13:56 /solr
drwxrwxrwt  - hdfs  supergroup           0 2017-04-21 14:04 /tmp
drwxr-xr-x  - hdfs  supergroup           0 2017-04-21 15:19 /user
```

2. Write to the HDFS root by creating a test directory, for example, "Made_from_Cloudera."

Output similar to the following displays:

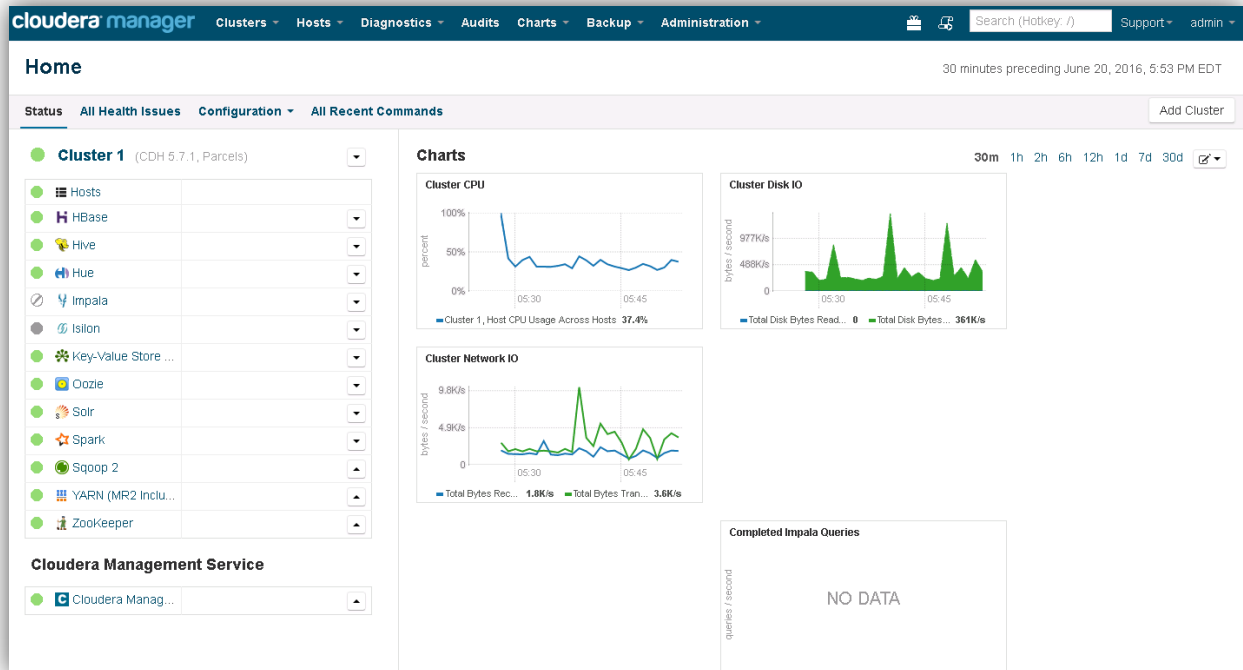
```
Found 6 items
drwxr-xr-x  - root  hadoop                0 2017-04-24 15:04 /Made_from_Cloudera
-rw-r--r--  3 root  wheel                0 2017-04-21 13:55 /THIS_IS_ISILON_zone1-
cdh.txt
drwxr-xr-x  - hbase hbase                0 2017-04-21 14:04 /hbase
drwxrwxr-x  - solr  solr                 0 2017-04-21 13:56 /solr
drwxrwxrwt  - hdfs  supergroup           0 2017-04-21 14:04 /tmp
drwxr-xr-x  - hdfs  supergroup           0 2017-04-21 15:19 /user
```

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

```
[root@rs-cd1 tmp]# yarn jar /opt/cloudera/parcels/CDH-5.7.1-1.cdh5.7.1.p0.11/lib/hadoop-mapreduce/hadoop-mapreduce-examples-2.6.0-cdh5.7.1.jar teragen 1000 /teragenOUT
16/07/07 17:25:34 INFO client.RMProxy: Connecting to ResourceManager at rs-cd1.foo.com/172.16.201.120:8032
16/07/07 17:25:35 INFO terasort.TeraSort: Generating 1000 using 2
16/07/07 17:25:35 INFO mapreduce.JobSubmitter: number of splits:2
16/07/07 17:25:36 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1467926691021_0001
16/07/07 17:25:37 INFO impl.YarnClientImpl: Submitted application application_1467926691021_0001
16/07/07 17:25:37 INFO mapreduce.Job: The url to track the job: http://rs-cd1.foo.com:8088/proxy/application_1467926691021_0001/
16/07/07 17:25:37 INFO mapreduce.Job: Running job: job_1467926691021_0001
16/07/07 17:26:05 INFO mapreduce.Job: Job job_1467926691021_0001 running in uber mode : false
16/07/07 17:26:05 INFO mapreduce.Job: map 0% reduce 0%
16/07/07 17:26:36 INFO mapreduce.Job: map 100% reduce 0%
16/07/07 17:26:37 INFO mapreduce.Job: Job job_1467926691021_0001 completed successfully
16/07/07 17:26:39 INFO mapreduce.Job: Counters: 31
  File System Counters
    FILE: Number of bytes read=0
    FILE: Number of bytes written=233210
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=164
    HDFS: Number of bytes written=100000
    HDFS: Number of read operations=8
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=4
  Job Counters
    Launched map tasks=2
    Other local map tasks=2
    Total time spent by all maps in occupied slots (ms)=59843
    Total time spent by all reduces in occupied slots (ms)=0
    Total time spent by all map tasks (ms)=59843
    Total vcore-seconds taken by all map tasks=59843
    Total megabyte-seconds taken by all map tasks=61279232
  Map-Reduce Framework
    Map input records=1000
    Map output records=1000
    Input split bytes=164
    Spilled Records=0
    Failed Shuffles=0
    Merged Map outputs=0
    GC time elapsed (ms)=402
    CPU time spent (ms)=1480
    Physical memory (bytes) snapshot=361488384
    Virtual memory (bytes) snapshot=3003490304
    Total committed heap usage (bytes)=505413632
  org.apache.hadoop.examples.terasort.TeraGen$Counters
    CHECKSUM=2173251765740
  File Input Format Counters
    Bytes Read=0
  File Output Format Counters
    Bytes Written=100000
```

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: [Start Cloudera 5.7 Impala with Isilon](#).



This completes the overview of deploying non-Kerberos Cloudera CDH with 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 OneFS cluster, ensure that the prerequisites in the following sections are met.

OneFS

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

- You must be running OneFS 8.1.2 or later.
- A dedicated OneFS access zone is in use; this access zone is not in the system zone.
- A OneFS SmartConnect zone is correctly configured for HDFS access.
- A simple access model exists between Hadoop and OneFS. User UIDs and GIDs are correctly implemented and allow HDFS access to the OneFS 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) providers. See the [Microsoft Active Directory documentation](#) for a high-level technical review regarding using Active Directory as a KDC.

Cloudera Manager

Ensure that the following prerequisites are met:

- You must be running CDH 5.7.1 or later.
- You will integrate an existing Cloudera CDH cluster and a OneFS 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.
 - OneFS SmartConnect zone name lookups must resolve correctly.
 - Reverse PTR records for all IP addresses in the SmartConnect pool must exist.
 - OneFS must be able to resolve all hosts, KDCs, and Active Directory servers as needed.
- Cloudera Manager must be configured correctly for OneFS 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 OneFS 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 OneFS permissioning model for multiprotocol access following Kerberization. This guide does not address all configurations or requirements. Other [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). See the [Cloudera documentation](#) for additional information about how to enable Kerberos authentication.

Since you are integrating an existing Cloudera CDH cluster and 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 OneFS and Hadoop integration.
2. Prepare the Cloudera cluster and Linux hosts for Kerberization.
3. All services must be running on the Cloudera dashboard before Kerberization.

4. Integrate the OneFS 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 OneFS 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:

- OneFS, 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 OneFS-integrated Hadoop cluster is a blend between Linux hosts running compute services and OneFS running data services, Cloudera cannot 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 OneFS completely. The Kerberization of a OneFS-integrated Hadoop cluster should be deployed as follows:

- The 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, see the Cloudera security documents: [Enabling Kerberos Authentication Using the Cloudera Wizard](#).

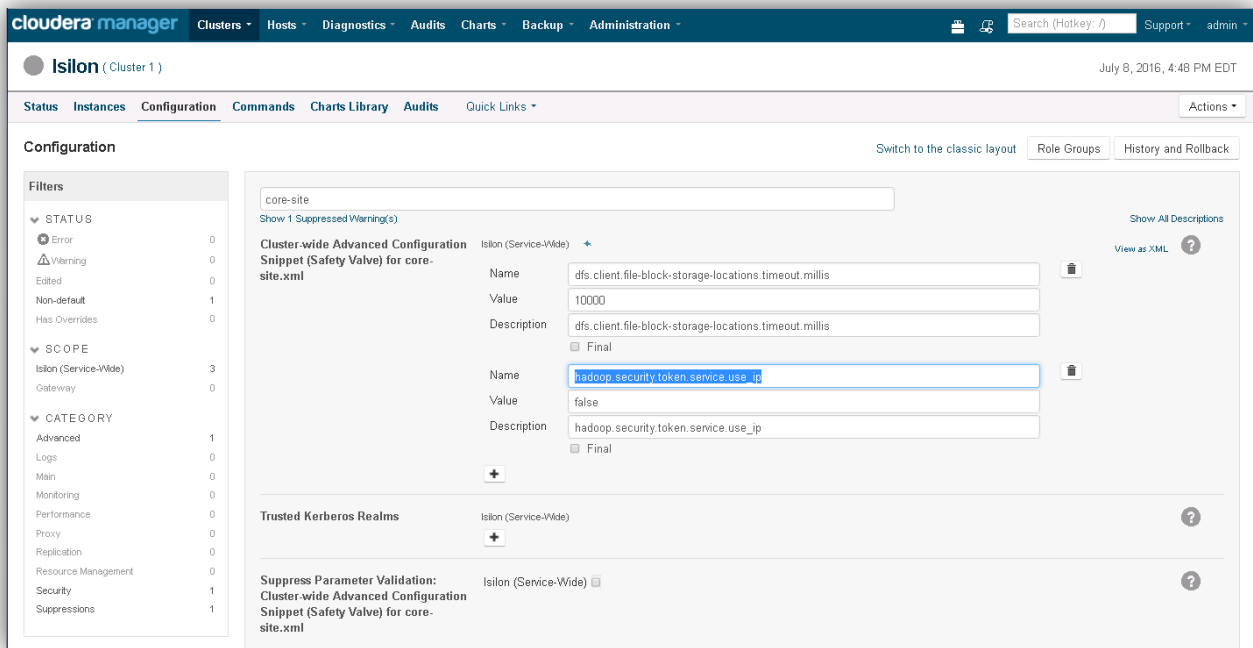
Preconfigure Cloudera Manager

Review the following configuration settings before proceeding with the Cloudera installation. Ensure 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 OneFS cluster, click the **Configuration** tab. For the OneFS 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**. 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. See 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:

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

Configure 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 go 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 who perform Hadoop jobs and the OneFS cluster, and also enable a standard OneFS 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. Ensure that you have enabled Active Directory GC indexing and replication as described in OneFS: How to configure OneFS and Active Directory for RFC2307 compliance, [article 335338](#) 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.
- OneFS 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 identify 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=zone1-hdp
```

where:

hdfs=>root	Maps the HDFS user to root
vlab* &= *[]	Maps all AD users to the local user, for example, AD\bob = bob, AD\jane = jane
vlab* += *[group]	(optional) Maps the users' primary group to AD; defines the GID group and not domain users.
vlab* += *[groups]	(optional) Maps the users' primary group to AD; defines GID group and not domain users.

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.

```
isi auth mapping token --zone=<zone-name> --user=hdfs
isi auth mapping token --zone=<zone-name> --user=hdfs@domain.com
```

Review the OneFS SPNs

Since OneFS is a clustered file system running on multiple nodes that are 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:

SPN	Name	Role
hdfs/clustername.fqdn	Clustername that is joined to AD	hdfs authentication to AD
hdfs/namenode.smartconnectname.fqdn	NN FQDN used by Ambari	hdfs authentication to AD per Smartconnect Zone
HTTP/namenode.smartconnectname.fqdn	NN FQDN used by Ambari	WebHDFS authentication to AD per Smartconnect Zone

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:

```
isi 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 - SPNs: hdfs/rip2-cd1.foo.com & HTTP/rip2-cd1.foo.com
```

For additional information about adding or modifying OneFS SPNs in Active Directory, see the [Isilon OneFS CLI Administration Guide](#).

Create proxy users

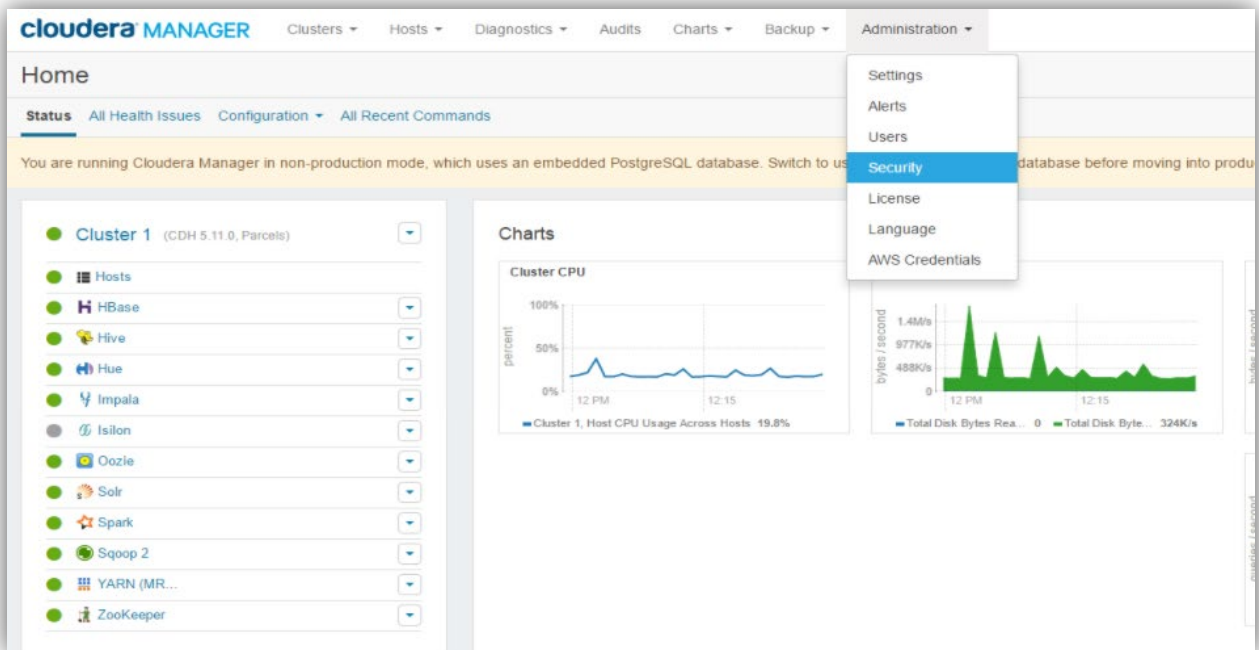
Create the required proxy users. Proxy users are required for service account impersonation for specific Hadoop services to run jobs and to add the required proxy users as needed. For more information about creating proxy users, see the [Isilon OneFS CLI Administration Guide](#).

This completes the OneFS Hadoop Active Directory setup.

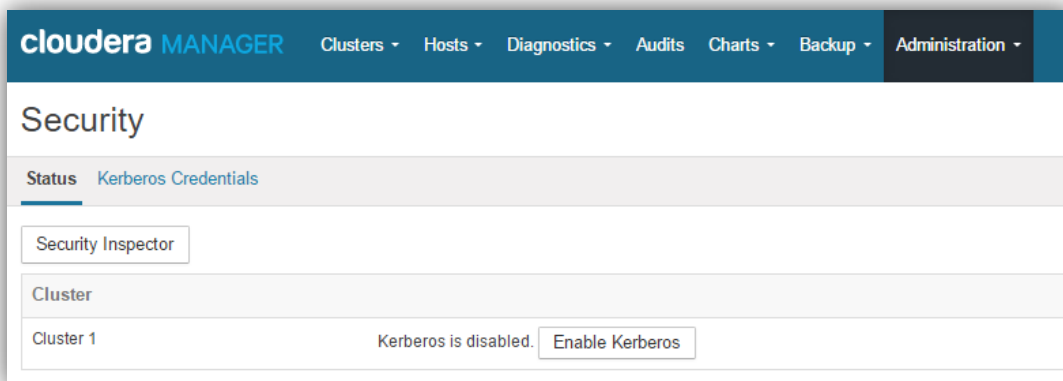
Enable Kerberos authentication in Cloudera Manager

After meeting the preceding prerequisites, you can Kerberize the Cloudera cluster. It is recommended that you suspend all client and user activity on the Hadoop cluster before starting 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 the boxes when you have completed the steps and then click **Continue**.

Enable Kerberos for Cluster 1

Welcome

This wizard walks you through the steps to configure Cloudera Manager and CDH to use Kerberos for authentication. All services in the cluster, as well as the Cloudera Management Service, are restarted as part of the wizard. Before proceeding with the wizard, read the [documentation](#) about enabling Kerberos.

Before using the wizard, ensure that you have performed the following steps:

Set up a working KDC. Cloudera Manager supports MIT KDC and Active Directory.

☒ Yes, I've set up a working KDC.

The KDC should be configured to have non-zero ticket lifetime and renewal lifetime. CDH will not work properly if tickets are not renewable.

☒ Yes, I've checked that the KDC allows renewable tickets.

OpenLdap client libraries should be installed on the Cloudera Manager Server host if you want to use Active Directory. Also, Kerberos client libraries should be installed on ALL hosts.

☒ Yes, I've installed the client libraries.

Cloudera Manager needs an account that has permissions to create other accounts in the KDC.

☒ Yes, I've created a proper account for Cloudera Manager.

Back
1 2 3 4 5 6 7 8 9
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 that will be used for principals.

Enable Kerberos for Cluster 1

KDC Information

Specify information about the KDC. The properties below are used by Cloudera Manager to generate principals for CDH daemons running on the cluster.

KDC Type: ☐ MIT KDC ☒ Active Directory

KDC Server Host: RDUVNODE60909 vlab.local

Kerberos Security Realm: VLAB.LOCAL

Kerberos Encryption Types: rc4-hmac, aes256-cts

Active Directory Suffix: ou=CD1,DC=vlab,DC=com

If you are using a version of CDH that supports the following, select the **Active Directory Delete Accounts on Credential Regeneration** checkbox.

Active Directory Delete Accounts on Credential Regeneration ☒

Set this option to true if regeneration of credentials should automatically delete the associated Active Directory accounts. Used only if Active Directory KDC is used for authentication.

Active Directory Set Encryption Types ☐

Set this option to true if creation of Active Directory accounts should automatically turn on the associated encryption types represented by the msDS-EncryptionTypes field. Used only if Active Directory KDC is used for authentication.

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

Active Directory Password Properties: length=12,minLowerCaseLetters=2,minUpperCaseLetters=2,minDigits=2,minSpaces=0,minSpecialChars=0,speci

Active Directory Account Properties: accountExpires=0,objectClass=top,objectClass=person,objectClass=organizationalPerson,objectClass=user

Active Directory Account Prefix

Active Directory Domain Controller Override

- Continuing in the Cloudera Manager wizard, manage the host `krb5.conf` host files by selecting the **Manage krb5.conf through Cloudera Manager** check box.

Enable Kerberos for Cluster 1

KRB5 Configuration

Specify the properties needed for generating `krb5.conf` for the cluster. You can use the safety valve fields to specify configuration of an advanced KDC setup; for example, with cross-realm authentication.

Manage `krb5.conf` through Cloudera Manager ☒

Kerberos Ticket Lifetime: 1 day(s)

Kerberos Renewable Lifetime: 7 day(s)

DNS Lookup KDC ☐

Forwardable Tickets ☒

KDC Timeout: 3 second(s)

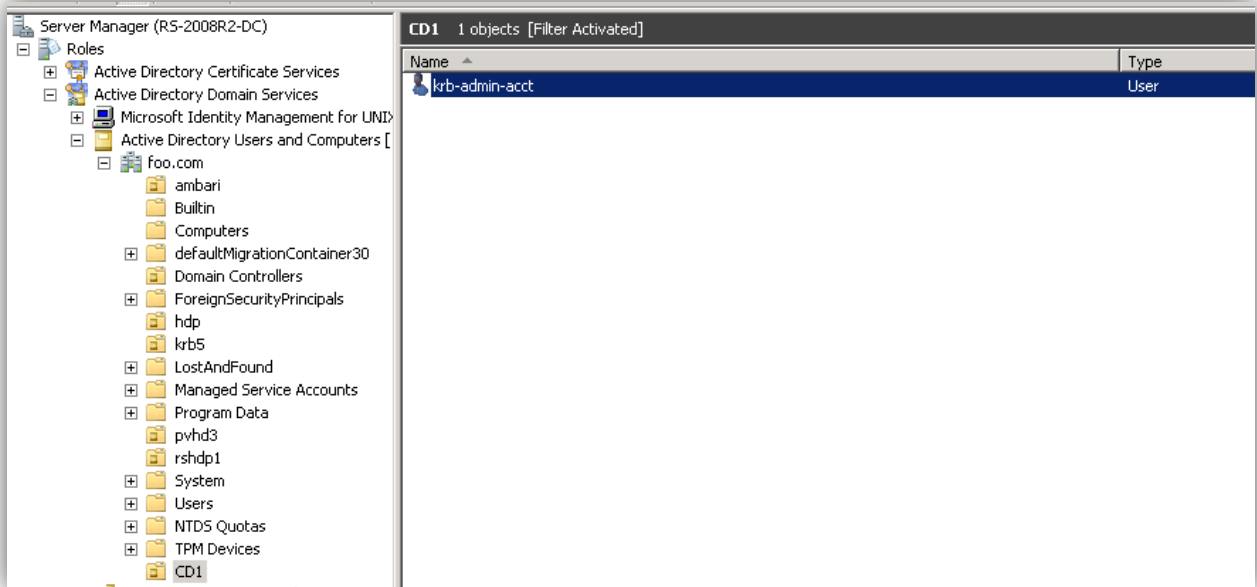
Advanced Configuration Snippet (Safety Valve) for [libdefaults] section of `krb5.conf`

Advanced Configuration Snippet (Safety Valve) for the Default Realm in `krb5.conf`

Back 1 2 3 4 5 6 7 8 9 Continue

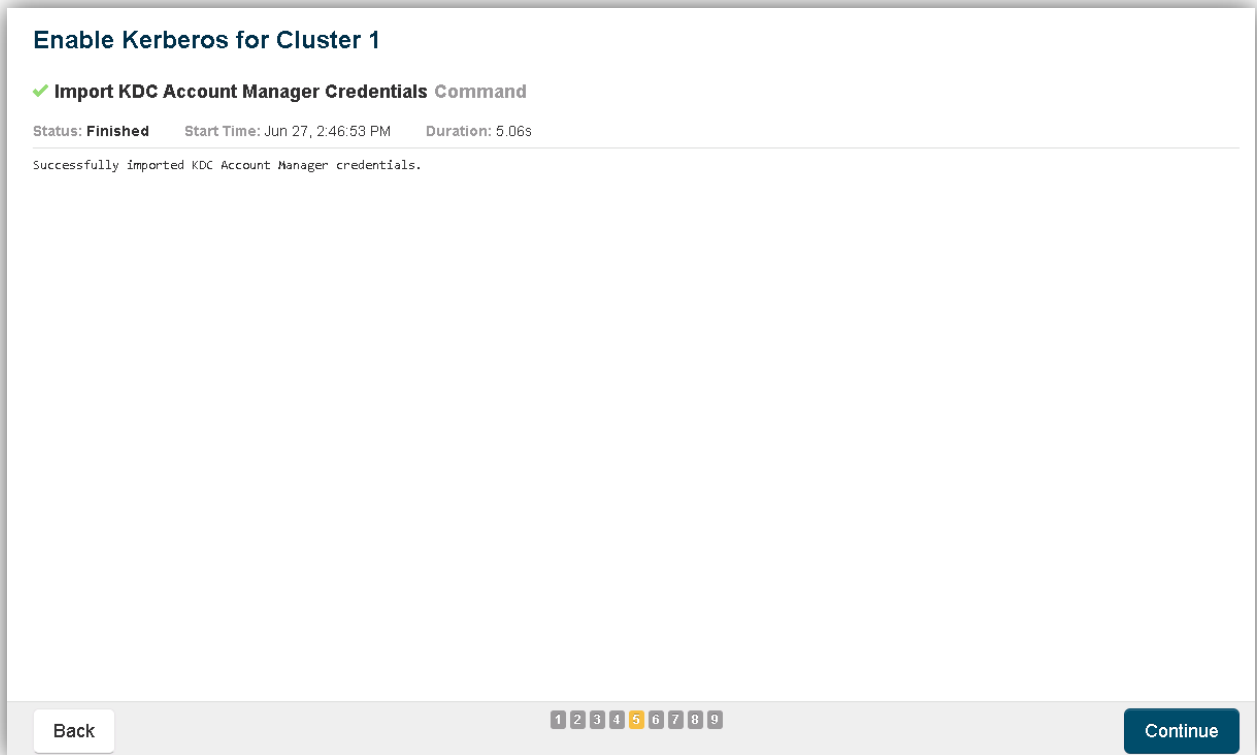
- Accept the defaults on the next screen and then click **Continue**. Since Cloudera Manager creates and manages all the principals, an AD OU with a delegated administrative account is used.

7. Look up the credentials for the AD user with delegated access to the OU in the AD domain. In the 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.



Note

If Active Directory is not configured for LDAPS, errors may display, which you must fix to continue.

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

Kerberos Principal

Specify the kerberos principal used by each service in the cluster. Additional steps maybe required if you decide to change these principals from their default values. Please read the [documentation](#) about custom principals before making changes on this page.

Kerberos Principal	
HBase (Service-Wide)	<input type="text" value="hbase"/>
Hive (Service-Wide)	<input type="text" value="hive"/>
Hue (Service-Wide)	<input type="text" value="hue"/>
Impala (Service-Wide)	<input type="text" value="impala"/>
Oozie (Service-Wide)	<input type="text" value="oozie"/>
Solr (Service-Wide)	<input type="text" value="solr"/>
Spark (Service-Wide)	<input type="text" value="spark"/>
Sqoop 2 (Service-Wide)	<input type="text" value="sqoop2"/>
YARN (MR2 Included) (Service-Wide)	<input type="text" value="yarn"/>
ZooKeeper (Service-Wide)	<input type="text" value="zookeeper"/>

Back
1 2 3 4 5 6 7 8 9
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**.

Enable Kerberos for Cluster 1

Configure Ports

Configure the privileged ports required by DataNodes in a secure HDFS service.

DataNode Transceiver	1004	Port for DataNode's Xceiver Protocol. Combined with the DataNode's hostname to build its address.
DataNode HTTP Web UI	1006	Port for the DataNode HTTP web UI. Combined with the DataNode's hostname to build its HTTP address.

The cluster needs to be restarted for the changes to take effect.

☒ Yes, I am ready to restart the cluster now.

Back
1 2 3 4 5 6 7 8 9
Continue

- 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.

```
isi hdfs settings modify --authentication-mode=kerberos_only --zone=rip2-cd1
```

Hadoop (HDFS) Current Access Zone: cd1-zone

Settings Proxy Users Virtual Racks

Edit HDFS Settings

HDFS Service Settings

☒ **Enable HDFS service**

Default Block Size
128 MB

Default Checksum Type
None

HDFS Protocol Settings

☒ **Enable WebHDFS access**

Authentication Type
Kerberos authentication

Root Directory
/ifs/cd1-zone/hdfs Browse...

Root directory must be within /ifs/cd1-zone

- The Kerberization process is automatically initialized. The Cloudera services are Kerberized, user principals are created in Active Directory, and keytabs are distributed.

Enable Kerberos Command

Status: **Running** Context: [Cluster 1](#) Start Time: Jun 27, 5:53:46 PM Abort

Details Completed 4 of 8 step(s). All Failed Only Running Only

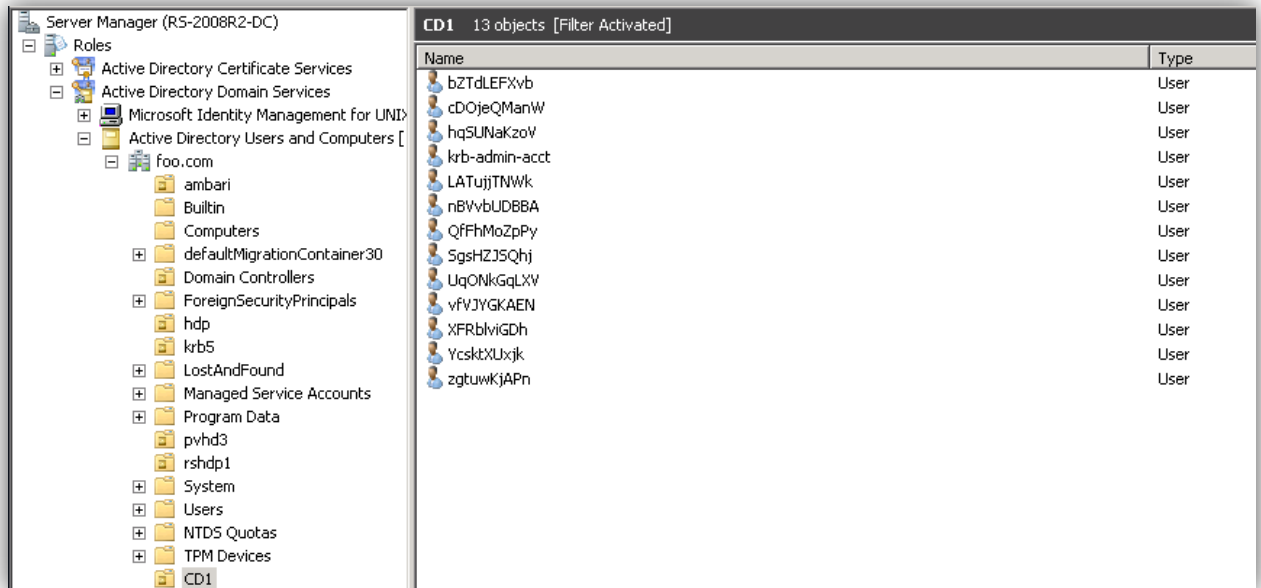
Step	Context	Start Time	Duration	Actions
➤ <input checked="" type="checkbox"/> Stop All services successfully stopped.	Cluster 1	Jun 27, 5:53:46 PM	82.08s	
➤ <input checked="" type="checkbox"/> Stop Successfully stopped service.	Cloudera Management Service	Jun 27, 5:55:08 PM	10.83s	
➤ <input checked="" type="checkbox"/> Deploy Kerberos Client Configuration Successfully deployed krb5.conf.	Cluster 1	Jun 27, 5:55:19 PM	15.75s	
➤ <input checked="" type="checkbox"/> Configure the Cluster for Kerberos Successfully configured the cluster for Kerberos.	Cluster 1	Jun 27, 5:55:35 PM	10ms	
➤ <input checked="" type="checkbox"/> Wait for credentials to be generated Waiting for command (881) to finish		Jun 27, 5:55:35 PM		

Deploy client configuration

Start Cloudera Management Services

Start cluster

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 try to restart.

Enable Kerberos for Cluster 1

✓ Enable Kerberos Command

Status: **Finished** Context: [Cluster 1](#) Start Time: Apr 27, 4:51:09 PM Duration: 9.3m

Successfully enabled Kerberos.

Details [Completed 8 of 8 step\(s\).](#) ● All ● Failed Only ● Running Only

Step	Context	Start Time	Duration	Actions
➤ ✓ Stop cluster All services successfully stopped.	Cluster 1	Apr 27, 4:51:09 PM	56.75s	
➤ ✓ Stop Cloudera Management Services Successfully stopped service.	Cloudera Management Service	Apr 27, 4:52:06 PM	10.27s	
➤ ✓ Deploy krb5.conf Successfully deployed krb5.conf.	Cluster 1	Apr 27, 4:52:16 PM	15.67s	
➤ ✓ Configure all services to use Kerberos Successfully configured the cluster for Kerberos.	Cluster 1	Apr 27, 4:52:32 PM	15ms	
➤ ✓ Wait for credentials to be generated Command (1168) has completed successfully		Apr 27, 4:52:32 PM	7.6s	
➤ ✓ Deploy client configuration Successfully deployed all client configurations.	Cluster 1	Apr 27, 4:52:40 PM	16.56s	
➤ ✓ Start Cloudera Management Services Successfully started service.	Cloudera Management Service	Apr 27, 4:52:57 PM	28.89s	
➤ ✓ Start cluster All services successfully started.	Cluster 1	Apr 27, 4:53:28 PM	7m	

[Back](#) 1 2 3 4 5 6 7 8 9 [Continue](#)

The following screen displays if the Kerberization was successful.

Enable Kerberos for Cluster 1

Congratulations!

You have enabled Kerberos for all your cluster(s).

Cluster	Status
Cluster 1	Successfully enabled Kerberos.

The Hue service may fail and stop the wizard. This is a known issue in some versions of Cloudera that you must work around if Hue is in use.

<p>✓ Execute command Start concurrently on 2 services Successfully completed 2 steps.</p>		Jul 11, 1:05:41 PM	75.49s
<p>✗ Start Failed to start service.</p>	Hue	Jul 11, 1:07:10 PM	1.46s
<p>✗ Starting 2 roles on service Service did not start successfully; not all of the required roles started; only 0/2 roles started. Reasons : null</p>		Jul 11, 1:07:10 PM	1.46s
<p>✗ Start this Kerberos Ticket Renewer Failed to start role.</p>	Kerberos Ticket Renewer (rs-cd1)	Jul 11, 1:07:11 PM	151ms
<p>✗ Start a role Role failed to start due to error com.cloudera.cmf.service.config.ConfigGenException: Unable to generate config file hue.ini for role with type KT_RENEWER: Unable to resolve owner of StringParamSpec{key=kerberos_princ_name}.</p>	Kerberos Ticket Renewer (rs-cd1)	Jul 11, 1:07:11 PM	111ms
<p>Role Log</p> <p>Full log file</p>			
<p>✗ Start this Hue Server Failed to start role.</p>	Hue Server (rs-cd1)	Jul 11, 1:07:11 PM	342ms
<p>✗ Start a role Role failed to start due to error com.cloudera.cmf.service.config.ConfigGenException: Unable to generate config file hue.ini for role with type HUE_SERVER: Unable to resolve owner of StringParamSpec{key=kerberos_princ_name}.</p>	Hue Server (rs-cd1)	Jul 11, 1:07:11 PM	292ms
<p>Role Log</p> <p>Full log file</p>			

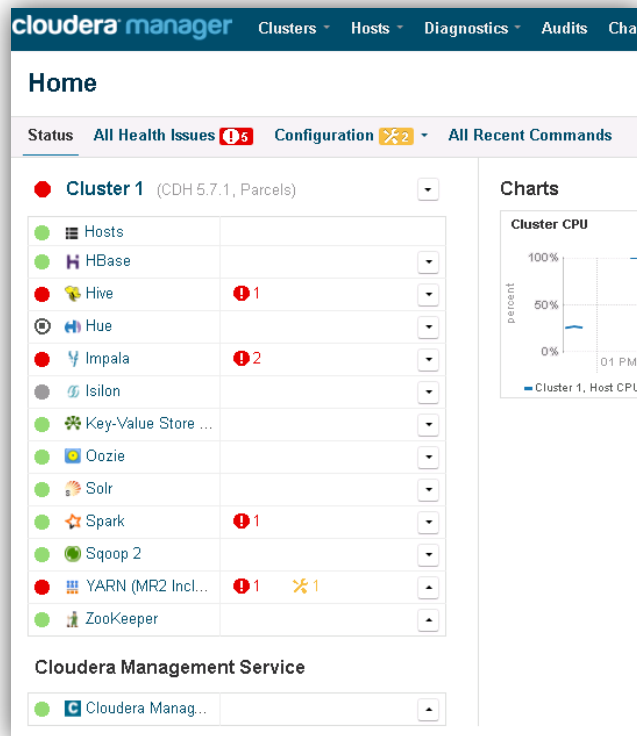
Back

1 2 3 4 5 6 7 8 9

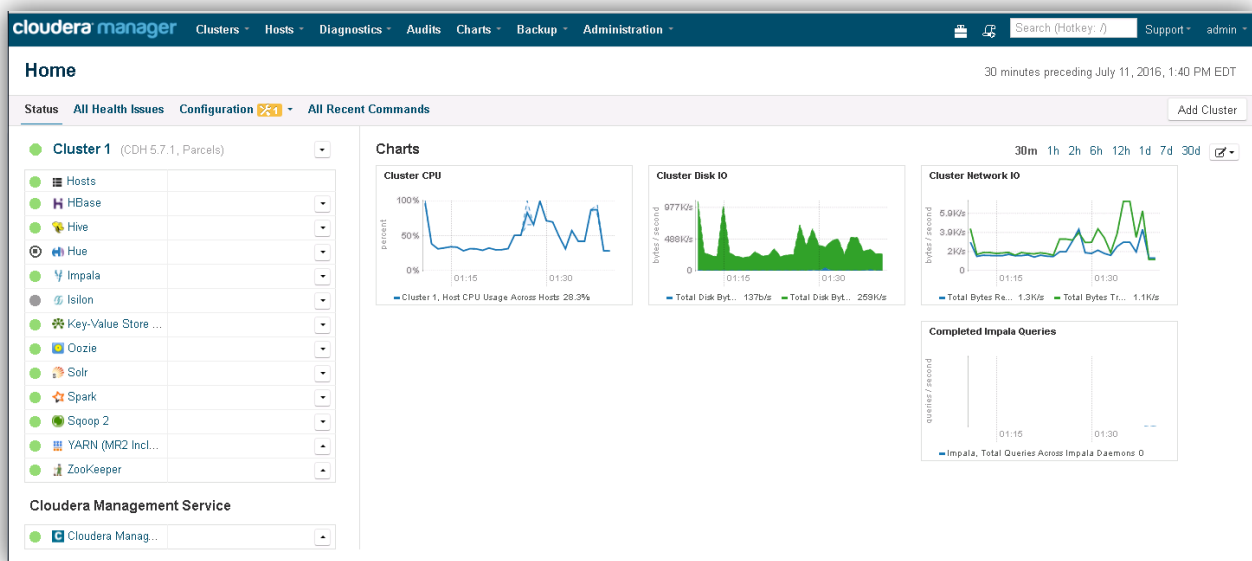
Continue

15. If the failure of the Hue service prevents the wizard from completing, you must 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 might see some services in an unhealthy state. You can address each of these services individually by starting or restarting them as required. Monitor the log files to get them started. You may have 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 about starting the Hue service, see the [Getting the Hue Service Started on Kerberized Cloudera with Isilon](#) article.

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

Go 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. See 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

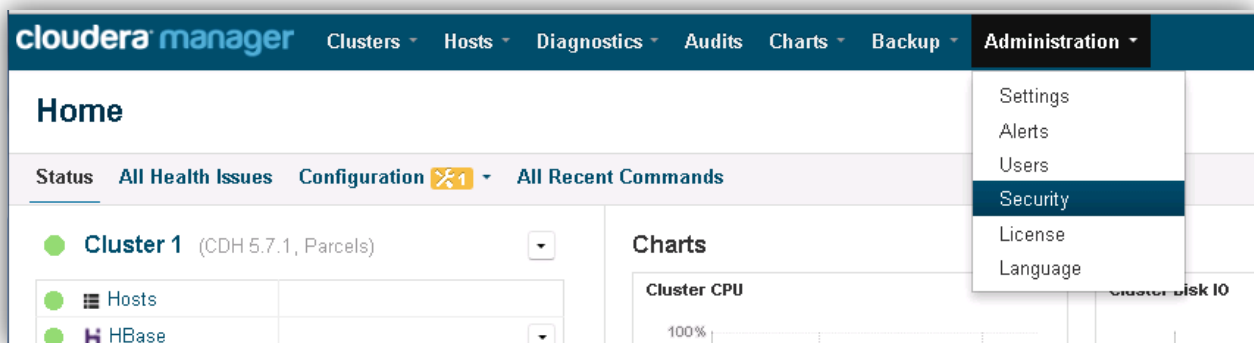
The KDC and Kerberos realm should be set up and configured per the Cloudera requirements and your realm setup.

Enable CDH Kerberos using MIT

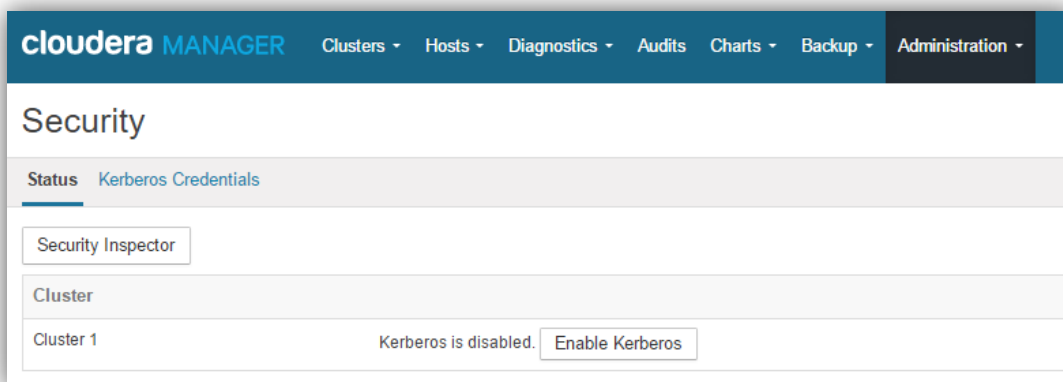
Use this procedure to enable Kerberos on a CDH cluster using MIT KDC. See the [Cloudera documentation](#) for additional information about how to enable Kerberos authentication. See [Get or Create a Kerberos Principal for the Cloudera Manager](#) for an example of creating principals for MIT KDC.

It is recommended that you suspend all client and user activity on the Hadoop cluster before starting 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 the boxes when you have completed the steps and then click **Continue**.

Enable Kerberos for Cluster 1

Welcome

This wizard walks you through the steps to configure Cloudera Manager and CDH to use Kerberos for authentication. All services in the cluster, as well as the Cloudera Management Service, are restarted as part of the wizard. Before proceeding with the wizard, read the [documentation](#) about enabling Kerberos.

Before using the wizard, ensure that you have performed the following steps:

Set up a working KDC. Cloudera Manager supports MIT KDC and Active Directory.

☒ Yes, I've set up a working KDC.

The KDC should be configured to have non-zero ticket lifetime and renewal lifetime. CDH will not work properly if tickets are not renewable.

☒ Yes, I've checked that the KDC allows renewable tickets.

OpenLdap client libraries should be installed on the Cloudera Manager Server host if you want to use Active Directory. Also, Kerberos client libraries should be installed on ALL hosts.

☒ Yes, I've installed the client libraries.

Cloudera Manager needs an account that has permissions to create other accounts in the KDC.

☒ Yes, I've created a proper account for Cloudera Manager.

Back
1 2 3 4 5 6 7 8 9
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).

Enable Kerberos for Cluster 1

KDC Information

Specify information about the KDC. The properties below are used by Cloudera Manager to generate principals for CDH daemons running on the cluster.

KDC Type

☒ MIT KDC
 ☐ Active Directory

KDC Server Host
kdc

RDUVNODE60909.vlab.local

Kerberos Security Realm
default_realm

VLAB.LOCAL

Kerberos Encryption Types

rc4-hmac

+

-

aes256-cts-hmac-sha1-96

+

-

Maximum Renewable Life for
Principals

5

day(s)

Back

1 2 3 4 5 6 7 8 9

Continue

- On the **KRB5 Configuration** screen, select the **Manage krb5.conf through Cloudera Manager** check box. After specifying the configuration settings, click **Continue**.

Enable Kerberos for Cluster 1

KRB5 Configuration

Specify the properties needed for generating krb5.conf for the cluster. You can use the safety valve fields to specify configuration of an advanced KDC setup; for example, with cross-realm authentication.

Manage krb5.conf through Cloudera Manager	<input checked="" type="checkbox"/> C	?
Kerberos Ticket Lifetime ticket_lifetime	1 day(s)	?
Kerberos Renewable Lifetime renew_lifetime	7 day(s)	?
DNS Lookup KDC dns_lookup_kdc	<input type="checkbox"/>	?
Forwardable Tickets forwardable	<input checked="" type="checkbox"/>	?
KDC Timeout kdc_timeout	3 second(s)	?
Advanced Configuration Snippet (Safety Valve) for [libdefaults] section of krb5.conf		?

Back 1 2 3 4 5 6 7 8 9 Continue

- 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 must be set up per Cloudera requirements within the KDC realm to facilitate principal management.

Enable Kerberos for Cluster 1

KDC Account Manager Credentials

Enter the credentials for the account that has permissions to create other users. Cloudera Manager will store it in encrypted form and use it whenever new principals need to be generated.

Username	cloudera-scm/admin	@	VLAB.LOCAL
Password	*****		

The **Enable Kerberos for Cluster 1** wizard imports your existing KDC account manager credentials.

Enable Kerberos for Cluster 1

✓ Import KDC Account Manager Credentials Command

Status: **Finished** Start Time: Apr 7, 11:20:58 AM Duration: 5.02s

Successfully imported KDC Account Manager credentials.

- No changes are required on the **Kerberos Principal** screen. These are the services that you will Kerberizing. Click **Continue**.

Enable Kerberos for Cluster 1

Kerberos Principal

Specify the Kerberos principal used by each service in the cluster. Additional steps maybe required if you decide to change these principals from their default values. Please read the [documentation](#) about custom principals before making changes on this page.

Kerberos Principal	
HBase (Service-Wide)	<input type="text" value="hbase"/>
Hive (Service-Wide)	<input type="text" value="hive"/>
Hue (Service-Wide)	<input type="text" value="hue"/>
Impala (Service-Wide)	<input type="text" value="impala"/>
Oozie (Service-Wide)	<input type="text" value="oozie"/>
Spark (Service-Wide)	<input type="text" value="spark"/>
YARN (MR2 Included) (Service-Wide)	<input type="text" value="yarn"/>
ZooKeeper (Service-Wide)	<input type="text" value="zookeeper"/>

Back
1 2 3 4 5 6 7 8 9
Continue

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

Caution

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

Enable Kerberos for Cluster 1

Configure Ports

Configure the privileged ports required by DataNodes in a secure HDFS service.

DataNode Transceiver Port	<input type="text" value="1004"/>	Port for DataNode's Xceiver Protocol. Combined with the DataNode's hostname to build its address.
DataNode HTTP Web UI Port	<input type="text" value="1006"/>	Port for the DataNode HTTP web UI. Combined with the DataNode's hostname to build its HTTP address.

The cluster needs to be restarted for the changes to take effect.

☒ Yes, I am ready to restart the cluster now.

Create the KDC as a OneFS authorization provider

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

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

- List the realm.

```
isi auth krb5 realm list
```

For example:

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

3. Create the Kerberos domains.

```
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:

```
rsteven-45uwrnw-1# isi auth krb5 domain create --domain=vlab.local --realm=VLAB.LOCAL
rsteven-45uwrnw-1# isi auth krb5 domain create --domain=.vlab.local --realm=VLAB.LOCAL
rsteven-45uwrnw-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 created in the OneFS web administration interface under the **Kerberos Provider** tab as shown in the following screen. (Since the OneFS SPNs have not been added yet, you will see the **Requires Additional Configuration** warning here):

Kerberos Realms [+ Create a Kerberos Realm](#)

Realm	Key Distribution Centers (KDCs)	Admin Server	Actions
VLAB.LOCAL Default Realm	RDUVNODE60909.vlab.local	RDUVNODE60909.vlab.local	View/Edit More

Kerberos Domains [+ Create a Kerberos Domain](#)

Domain	Kerberos Realm	Actions
.vlab.local	VLAB.LOCAL	View/Edit More
vlab.local	VLAB.LOCAL	View/Edit More

Kerberos Providers [+ Create a Kerberos Provider](#)

Realm	Service Principal Name (SPN) Management	Actions
VLAB.LOCAL	Recommended Requires Additional Configuration	View/Edit More

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

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

For example:

```
rsteven-45uwrnw-1# isi zone zones modify --zone=zone2-cdh --add-auth-provider=krb5:VLAB.LOCAL
rsteven-45uwrnw-1# isi zone zones view --zone=zone2-cdh
      Name: zone2-cdh
      Path: /ifs/zone2/cdh
      Groupnet: groupnet0
      Map Untrusted: -
      Auth Providers: lsa-local-provider:zone2-cdh, lsa-krb5-provider:VLAB.LOCAL
      NetBIOS Name: -
      User Mapping Rules: hdfs=>root
      Home Directory Umask: 0077
      Skeleton Directory: /usr/share/skel
      Cache Entry Expiry: 4H
      Negative Cache Entry Expiry: 1m
      Zone ID: 3
```

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

```
isi auth krb5 spn create --provider-name=VLAB.LOCAL --spn=hdfs/isilonsczone-
cdh2.vlab.local --user=cloudera-scm/admin@VLAB.LOCAL
isi auth krb5 spn create --provider-name=VLAB.LOCAL --spn=HTTP/isilonsczone-
cdh2.vlab.local --user=cloudera-scm/admin@VLAB.LOCAL
```

6. List the Kerberos realms by running the following command.

```
isi auth krb5 spn list
```

For example:

```
rsteven-45uwrnw-1# isi auth krb5 spn list VLAB.LOCAL
SPN                               Kvno
-----
HTTP/isilonsczone-cdh2.vlab.local@VLAB.LOCAL 2
hdfs/isilonsczone-cdh2.vlab.local@VLAB.LOCAL 2
-----
Total: 2
Note that this Kerberos realm has SPNs and keys managed manually.
```

You can also view the principals in the OneFS web administration interface. The previous warnings are gone.

Active Directory

LDAP

NIS

Local Provider

File Provider

Kerberos Provider

Kerberos Settings

Create a Kerberos Provider in One Step

Use the "Get Started" button to create a complete Kerberos provider including new realm, domain(s), and provider in one step.

Get Started

Kerberos Realms

+ Create a Kerberos Realm

Bulk actions

Realms	Key Distribution Centers (KDCs)	Admin Server	Actions
VLAB.LOCAL Default Realm	RDUVNODE80909.vlab.local	RDUVNODE80909.vlab.local	View / Edit / More

Kerberos Domains

+ Create a Kerberos Domain

Bulk actions

Domain	Kerberos Realm	Actions
.vlab.local	VLAB.LOCAL	View / Edit / More
vlab.local	VLAB.LOCAL	View / Edit / More

Kerberos Providers

+ Create a Kerberos Provider

Bulk actions

Realms	Service Principal Name (SPN) Management	Actions
VLAB.LOCAL	Manual	View / Edit / More

- List the OneFS principals that are created by Isilon on the KDC by running the following command after logging into the KDC:

```
listprincs
```

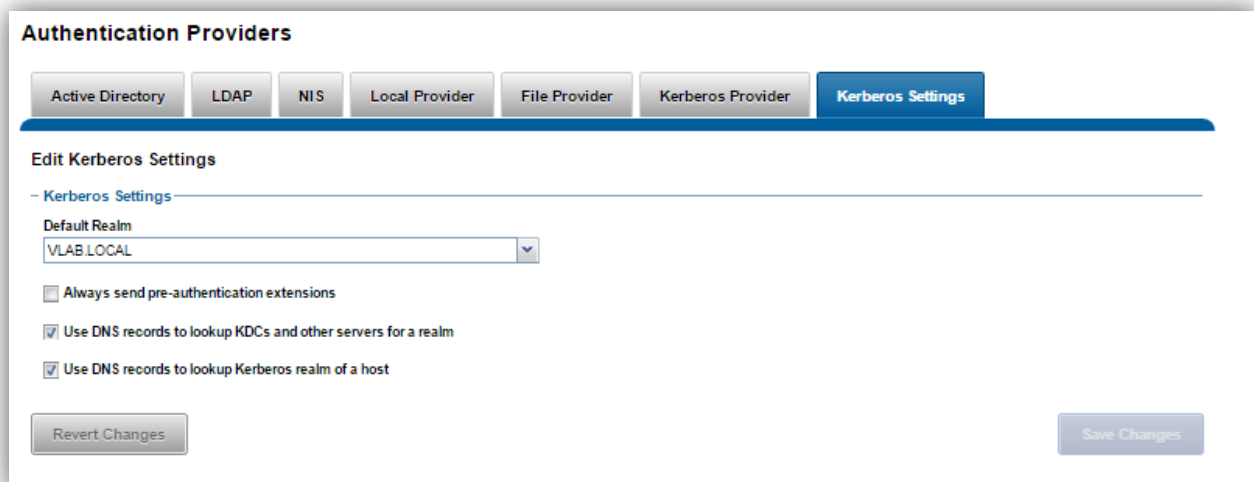
For example:

```
kadmin.local: listprincs
HTTP/RDUVNODE60904.vlab.local@VLAB.LOCAL
HTTP/isilonzone-cdh2.vlab.local@VLAB.LOCAL
K/M@VLAB.LOCAL
cloudera-scm/admin@VLAB.LOCAL
hbase/RDUVNODE60904.vlab.local@VLAB.LOCAL
hdfs/isilonzone-cdh2.vlab.local@VLAB.LOCAL
hive/RDUVNODE60904.vlab.local@VLAB.LOCAL
hue/RDUVNODE60904.vlab.local@VLAB.LOCAL
impala/RDUVNODE60904.vlab.local@VLAB.LOCAL
kadmin/admin@VLAB.LOCAL
kadmin/changepw@VLAB.LOCAL
kadmin/rduvnode60909.vlab.local@VLAB.LOCAL
kdcuser1@VLAB.LOCAL
krbtgt/VLAB.LOCAL@VLAB.LOCAL
mapred/RDUVNODE60904.vlab.local@VLAB.LOCAL
oozie/RDUVNODE60904.vlab.local@VLAB.LOCAL
spark/RDUVNODE60904.vlab.local@VLAB.LOCAL
yarn/RDUVNODE60904.vlab.local@VLAB.LOCAL
zookeeper/RDUVNODE60904.vlab.local@VLAB.LOCAL
```

The OneFS cluster should now be 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 article [Ambari Automated Kerberos Configuration with Isilon](#) 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=$isilon_zone --add-user=ambari-qa
isi hdfs proxyusers create hive --zone=$isilon_zone --add-user=ambari-qa
```

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

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

10. View the HDFS settings.

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

For example:

```
rsteven-45uwrnw-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
      Ambari Server: -
    Ambari Namenode: -
      Odp Version: -
    Data Transfer Cipher: none
  Ambari 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**.

Enable Kerberos for Cluster 1

Configure Ports

Configure the privileged ports required by DataNodes in a secure HDFS service.

DataNode Transceiver Port	<input type="text" value="1004"/>	Port for DataNode's Xceiver Protocol. Combined with the DataNode's hostname to build its address.
DataNode HTTP Web UI Port	<input type="text" value="1006"/>	Port for the DataNode HTTP web UI. Combined with the DataNode's hostname to build its HTTP address.

The cluster needs to be restarted for the changes to take effect.

☒ Yes, I am ready to restart the cluster now.

2. The following screen displays during Kerberization.

Enable Kerberos for Cluster 1

Enable Kerberos Command

Status: **Running** Context: [Cluster 1](#) Start Time: Apr 7, 11:23:03 AM [Abort](#)

Details [Completed 0 of 8 step\(s\).](#) ☒ All ☐ Failed Only ☐ Running Only

Step	Context	Start Time	Duration	Actions
➤ Stop cluster	Cluster 1	Apr 7, 11:23:04 AM		Abort
Stop Cloudera Management Services				
Deploy krb5.conf				
Configure all services to use Kerberos				
➤ Wait for credentials to be generated				
Deploy client configuration				
Start Cloudera Management Services				
Start cluster				

Back 1 2 3 4 5 6 7 8 9 [Continue](#)

- Click **Continue** when the status displays as **Finished**.

Enable Kerberos for Cluster 1

✓ Enable Kerberos Command

Status: **Finished** Context: [Cluster 1](#) ⌵ Start Time: Apr 10, 8:37:23 AM Duration: 4.4m

Successfully enabled Kerberos.

Details [Completed 8 of 8 step\(s\).](#) ⊙ All ⊙ Failed Only ⊙ Running Only

Step	Context	Start Time	Duration	Actions
➤ ✓ Stop cluster All services successfully stopped.	⌵ Cluster 1 ⌵	Apr 10, 8:37:23 AM	35.88s	
➤ ✓ Stop Cloudera Management Services Successfully stopped service.	⌵ Cloudera Management Service ⌵	Apr 10, 8:37:59 AM	8.44s	
➤ ✓ Deploy krb5.conf Successfully deployed krb5.conf.	⌵ Cluster 1 ⌵	Apr 10, 8:38:08 AM	15.5s	
➤ ✓ Configure all services to use Kerberos Successfully configured the cluster for Kerberos.	⌵ Cluster 1 ⌵	Apr 10, 8:38:23 AM	5ms	
➤ ✓ Wait for credentials to be generated Command (266) has completed successfully		Apr 10, 8:38:23 AM	5.01s	
➤ ✓ Deploy client configuration Successfully deployed all client configurations.	⌵ Cluster 1 ⌵	Apr 10, 8:38:28 AM	15.66s	
➤ ✓ Start Cloudera Management Services Successfully started service.	⌵ Cloudera Management Service ⌵	Apr 10, 8:38:44 AM	24.44s	
➤ ✓ Start cluster All services successfully started.	⌵ Cluster 1 ⌵	Apr 10, 8:39:10 AM	2.6m	

Back 1 2 3 4 5 6 7 8 9 [Continue](#)

The Kerberization is now complete.

Enable Kerberos for Cluster 1

Congratulations!

You have enabled Kerberos for all your cluster(s).

Cluster	Status
Cluster 1	Successfully enabled Kerberos.

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 OneFS enforces Kerberos-only access to HDFS root. This test also validates that simple authentication is not supported.

Sample output is shown:

```
kdcuser1@rduvnode61384:~ $ hadoop fs -ls /
17/04/10 10:14:14 WARN security.UserGroupInformation: PrivilegedActionException as:kdcuser1 (auth:KERBEROS) cause:java.
security.sasl.SaslException: GSS initiate failed [Caused by GSSException: No valid credentials provided (Mechanism level:
Failed to find any Kerberos tgt)]
17/04/10 10:14:14 WARN ipc.Client: Exception encountered while connecting to the server : javax.security.sasl.SaslExcepti
on: GSS initiate failed [Caused by GSSException: No valid credentials provided (Mechanism level: Failed to find any Kerbe
ros tgt)]
17/04/10 10:14:14 WARN security.UserGroupInformation: PrivilegedActionException as:kdcuser1 (auth:KERBEROS) cause:java.i
o.IOException: javax.security.sasl.SaslException: GSS initiate failed [Caused by GSSException: No valid credentials provi
ded (Mechanism level: Failed to find any Kerberos tgt)]
ls: Failed on local exception: java.io.IOException: javax.security.sasl.SaslException: GSS initiate failed [Caused by GSS
Exception: No valid credentials provided (Mechanism level: Failed to find any Kerberos tgt)]; Host Details : local host i
s: "rduvnode61384.vlab.local/10.99.36.251"; destination host is: "isilonzone-cdh2.vlab.local":8020;
```

2. Authenticate and obtain a valid Kerberos ticket.

```
kinit
klist -e
```

```
kdcuser1@rduvnode61384:~ $ kinit
Password for kdcuser1@VLAB.LOCAL:
kdcuser1@rduvnode61384:~ $ klist -e
Ticket cache: FILE:/tmp/krb5cc_50000
Default principal: kdcuser1@VLAB.LOCAL

Valid starting    Expires          Service principal
04/10/17 10:14:21  04/11/17 10:14:21  krbtgt/VLAB.LOCAL@VLAB.LOCAL
renew until 04/10/17 10:14:21, Etype (skey, tkt): arcfour-hmac, aes256-cts-hmac-sha1-96
```

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:

```
kdcuser1@rduvnode61384:~ $ hadoop fs -ls /
Found 5 items
-rw-r--r--   3 root   wheel           0 2017-04-06 09:37 /THIS_IS_ISILON_zone2-cdh.txt
drwx----- - hbase  hbase           0 2017-04-10 08:39 /hbase
drwxrwxr-x - solr   solr           0 2017-04-06 09:39 /solr
drwxrwxrwt - hdfs   supergroup     0 2017-04-06 15:36 /tmp
drwxr-xr-x - hdfs   supergroup     0 2017-04-06 09:39 /user
```

- 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/kdcuser1/gen01
```

Output similar to the following displays:

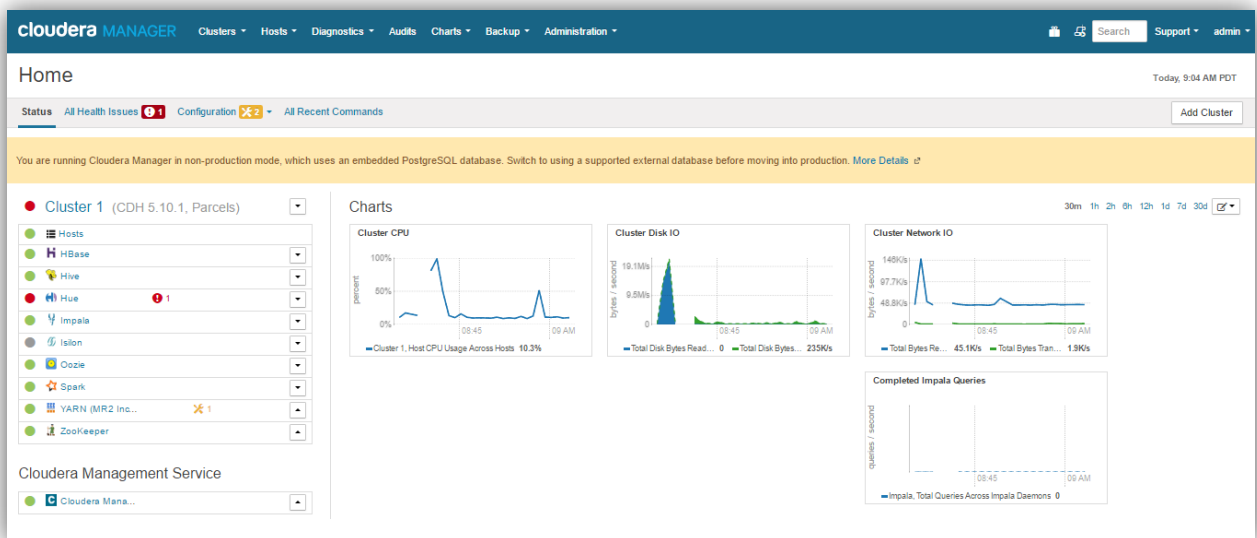
```
kdcuser1@rduvnode61384:~$ hadoop jar /opt/cloudera/parcels/CDH/jars/hadoop-mapreduce-examples-2.6.0-cdh5.10.1.jar teragen 10000 /user/kdcuser1/gen01
17/04/11 08:11:23 INFO client.RMProxy: Connecting to ResourceManager at rduvnode61384.vlab.local/10.99.36.251:8032
17/04/11 08:11:23 INFO hdfs.DFSClient: Created token for kdcuser1: HDFS_DELEGATION_TOKEN owner=kdcuser1@VLAB.LOCAL, renewer=yarn/rduvnode61384.vlab.local@VLAB.LOCAL, realUser=, issueDate=1491923454532, maxDate=1492528254532, sequenceNumber=0, masterKeyId=0 on 10.99.36.242:8020
17/04/11 08:11:23 INFO security.TokenCache: Got dt for hdfs://isilonsczone-cdh2.vlab.local:8020; Kind: HDFS_DELEGATION_TOKEN, Service: 10.99.36.242:8020, Ident: (token for kdcuser1: HDFS_DELEGATION_TOKEN owner=kdcuser1@VLAB.LOCAL, renewer=yarn/rduvnode61384.vlab.local@VLAB.LOCAL, realUser=, issueDate=1491923454532, maxDate=1492528254532, sequenceNumber=0, masterKeyId=0)
17/04/11 08:11:24 INFO terasort.TeraGen: Generating 10000 using 2
17/04/11 08:11:24 INFO mapreduce.JobSubmitter: number of splits:2
17/04/11 08:11:24 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1491855457974_0004
17/04/11 08:11:24 INFO mapreduce.JobSubmitter: Kind: HDFS_DELEGATION_TOKEN, Service: 10.99.36.242:8020, Ident: (token for kdcuser1: HDFS_DELEGATION_TOKEN owner=kdcuser1@VLAB.LOCAL, renewer=yarn/rduvnode61384.vlab.local@VLAB.LOCAL, realUser=, issueDate=1491923454532, maxDate=1492528254532, sequenceNumber=0, masterKeyId=0)
17/04/11 08:11:24 INFO impl.YarnClientImpl: Submitted application application_1491855457974_0004
17/04/11 08:11:24 INFO mapreduce.Job: The url to track the job: http://rduvnode61384.vlab.local:8088/proxy/application_1491855457974_0004/
17/04/11 08:11:24 INFO mapreduce.Job: Running job: job_1491855457974_0004
17/04/11 08:11:35 INFO mapreduce.Job: Job job_1491855457974_0004 running in uber mode : false
17/04/11 08:11:35 INFO mapreduce.Job: map 0% reduce 0%
17/04/11 08:11:41 INFO mapreduce.Job: map 50% reduce 0%
17/04/11 08:11:45 INFO mapreduce.Job: map 100% reduce 0%
17/04/11 08:11:46 INFO mapreduce.Job: Job job_1491855457974_0004 completed successfully
17/04/11 08:11:46 INFO mapreduce.Job: Counters: 31
  File System Counters
    FILE: Number of bytes read=0
    FILE: Number of bytes written=247374
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=164
    HDFS: Number of bytes written=1000000
    HDFS: Number of read operations=8
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=4
  Job Counters
    Launched map tasks=2
    Other local map tasks=2
    Total time spent by all maps in occupied slots (ms)=8264
    Total time spent by all reduces in occupied slots (ms)=0
    Total time spent by all map tasks (ms)=8264
    Total vcore-seconds taken by all map tasks=8264
    Total megabyte-seconds taken by all map tasks=33849344
  Map-Reduce Framework
    Map input records=10000
    Map output records=10000
    Input split bytes=164
    Spilled Records=0
    Failed Shuffles=0
    Merged Map outputs=0
    GC time elapsed (ms)=66
    CPU time spent (ms)=1460
    Physical memory (bytes) snapshot=373968896
    Virtual memory (bytes) snapshot=3133206528
    Total committed heap usage (bytes)=502267904
```

```
org.apache.hadoop.examples.terasort.TeraGen$Counters
CHECKSUM=21555350172850
File Input Format Counters
  Bytes Read=0
File Output Format Counters
  Bytes Written=1000000
kdcuser1@rduvnode61384:~$
```

Troubleshoot services

Troubleshoot each of the failed services individually. Review 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:

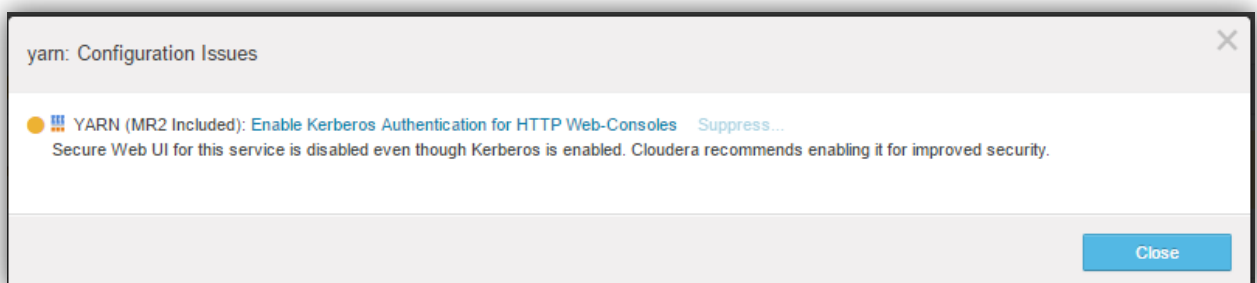


You may also see exception errors similar to the following if services have failed:

```
rsteven-45uwrnw-1: 2017-04-10T09:00:31-07:00 <30.6> rsteven-45uwrnw-1 hdfs[2969]: [hdfs] RPC V9 user: hue/rduvnode61384.vlab.local@VLAB.LOCAL exception: java.lang.SecurityException cause: User hive/rduvnode61384.vlab.local cannot impersonate User hue/rduvnode61384.vlab.local@VLAB.LOCAL Reason: Proxyuser "hive" Members do not intersect
```

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.



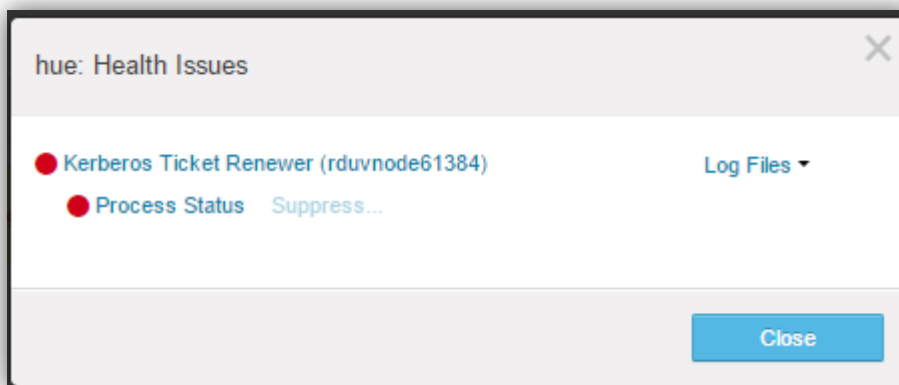
Example 2—Hive Canary service

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.

```
2017-04-27T17:06:15-04:00 <30.6> isilon01-1 hdfs[2553]: [hdfs] READ block: blk_4302570300_1000 offset: 1024 len: 931
2017-04-27T17:07:04-04:00 <30.6> isilon01-1 hdfs[2553]: [hdfs] READ block: blk_4302570300_1000 offset: 0 len: 1950
2017-04-27T17:08:10-04:00 <30.6> isilon01-1 hdfs[2553]: [hdfs] RPD V3 user: hue/centos-06.foo.com@FOO.COM exception: java.lang.SecurityException cause: User hive/centos-06.foo.com@FOO.COM cannot impersonate User hue/centos-06.foo.com@FOO.COM Reason: Proxyuser "hive" Members do not intersect
```

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>.



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-sha1, no salt
Key: vno 1, arcfour-hmac, no salt
Key: vno 1, des-hmac-sha1, 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:

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

Example output

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

Example: Run the `get_principal` command again:

```
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: 90 days 00:00:00
Last modified: Mon Apr 10 10:06:58 PDT 2017 (cloudera-scm/admin@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-sha1, no salt
Key: vno 1, arcfour-hmac, no salt
Key: vno 1, des-hmac-sha1, 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:

```
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: 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-sha1-96, no salt
Key: vno 2, aes128-cts-hmac-sha1-96, no salt
Key: vno 2, des3-cbc-sha1, no salt
Key: vno 2, arcfour-hmac, no salt
Key: vno 2, des-hmac-sha1, 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.
```

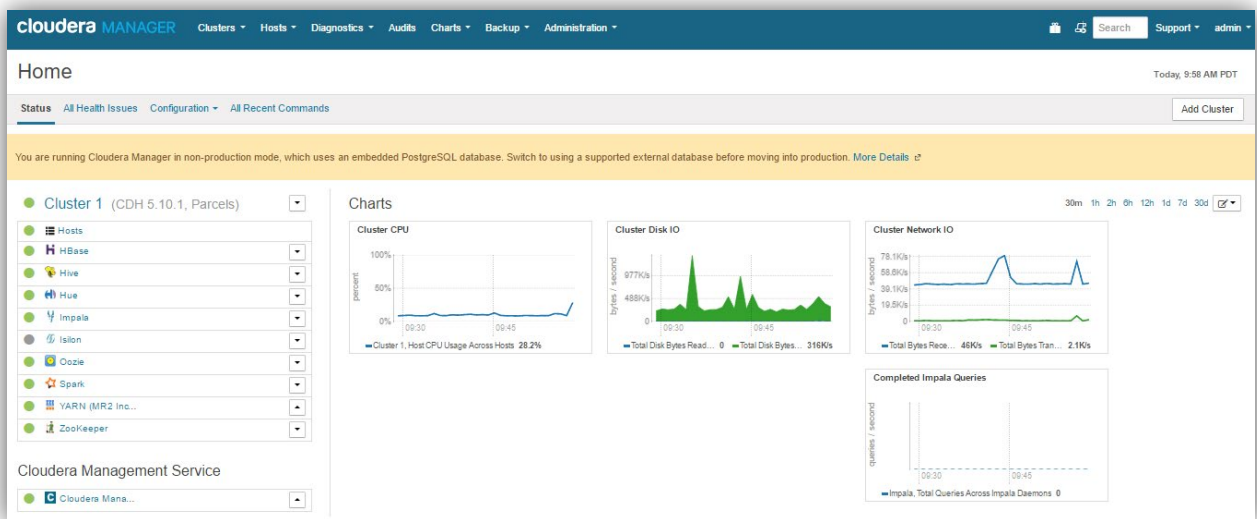
Example: Run the `get_principal` command on the Hue principal:

```
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-sha1-96, no salt
Key: vno 2, aes128-cts-hmac-sha1-96, no salt
Key: vno 2, des3-cbc-sha1, no salt
Key: vno 2, arcfour-hmac, no salt
Key: vno 2, des-hmac-sha1, no salt
Key: vno 2, des-cbc-md5, no salt
MKey: vno 1
Attributes:
Policy: [none]
```

- 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:



- Review the `krb5.com` file.

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

```
root@RDUVNODE60904:~ # 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 = RDUVNODE60909.vlab.local
admin_server = RDUVNODE60909.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 Dell EMC PowerScale Technical Support.



Contacting Dell EMC PowerScale Technical Support

Online Support: <https://support.dell.com/>

Telephone Support:

United States: 800-782-4362 (800-SVC-4EMC)

Canada: 800-543-4782

Worldwide: +1-508-497-7901

Other [worldwide access numbers](#)