PowerScale OneFS with Hadoop and Hortonworks for Kerberos Installation Guide

8.1.2 – 9.0.0 with Ambari 2.7.x and later versions

Abstract

This guide walks you through the process of installing PowerScale OneFS with Hadoop for use with the Hortonworks Data Platform (HDP) 3.0.1 and later, and the Apache Ambari manager 2.7.1 and later. It also discusses the process of installing and configuring Kerberos that is enabled through Apache Ambari to work with OneFS clusters.

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	December 6, 2018	New version of the guide for OneFS 8.1.2.0 with Ambari 2.7.1.0 and HDP 3.0.1.0
2.00	August 19, 2019	Added OneFS 8.1.2 related content
3.00	October 15, 2019	HDP 3.1 certification has been completed
4.00	June 16, 2020	Added OneFS 8.2 - 9.0 related content

Contents

Introduction

Hadoop is an open-source framework that enables the distributed processing of large sets of data across clusters of systems. Hadoop clusters use the Hadoop Distributed File System (HDFS) to provide high-throughput access to application data. You can follow the steps in this guide to install PowerScale OneFS with Hadoop for use with the Hortonworks Data Platform (HDP) and the Apache Ambari manager. You can also use this guide to configure Apache Ambari-automated Kerberos with OneFS.

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

Audience

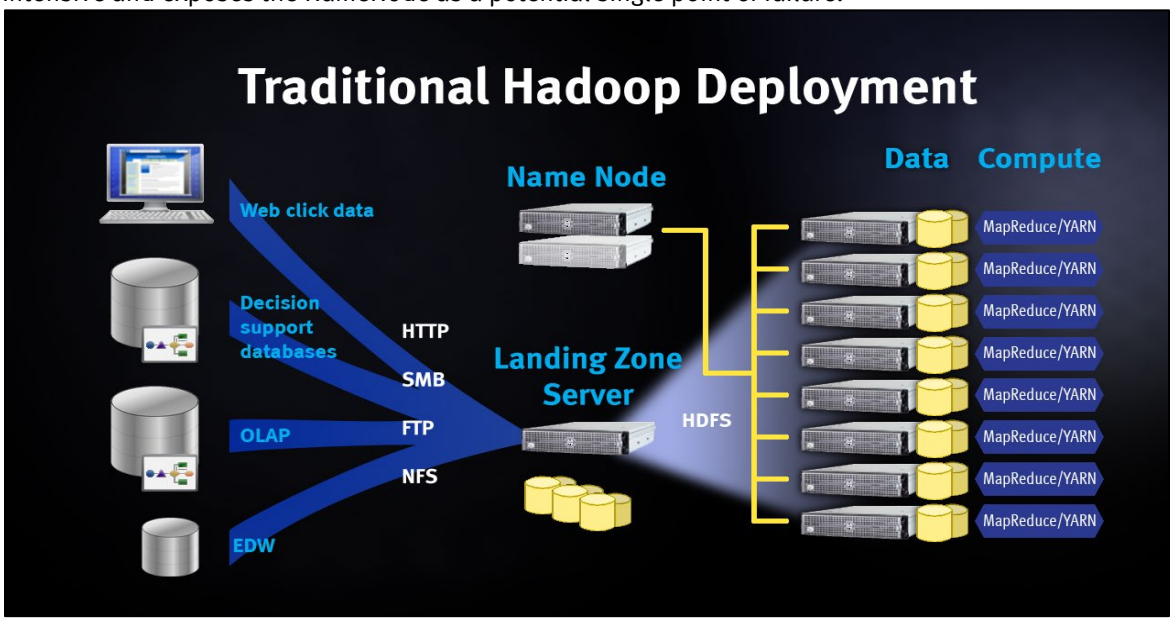
This guide is intended for systems administrators, IT program managers, IT architects, and IT managers who are installing OneFS 8.1.2.0 or later with Ambari 2.7.1 and later and HDP 3.0.1 and later.

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) interface. A PowerScale cluster that is powered by the OneFS operating system delivers a scalable pool of storage with a global namespace.

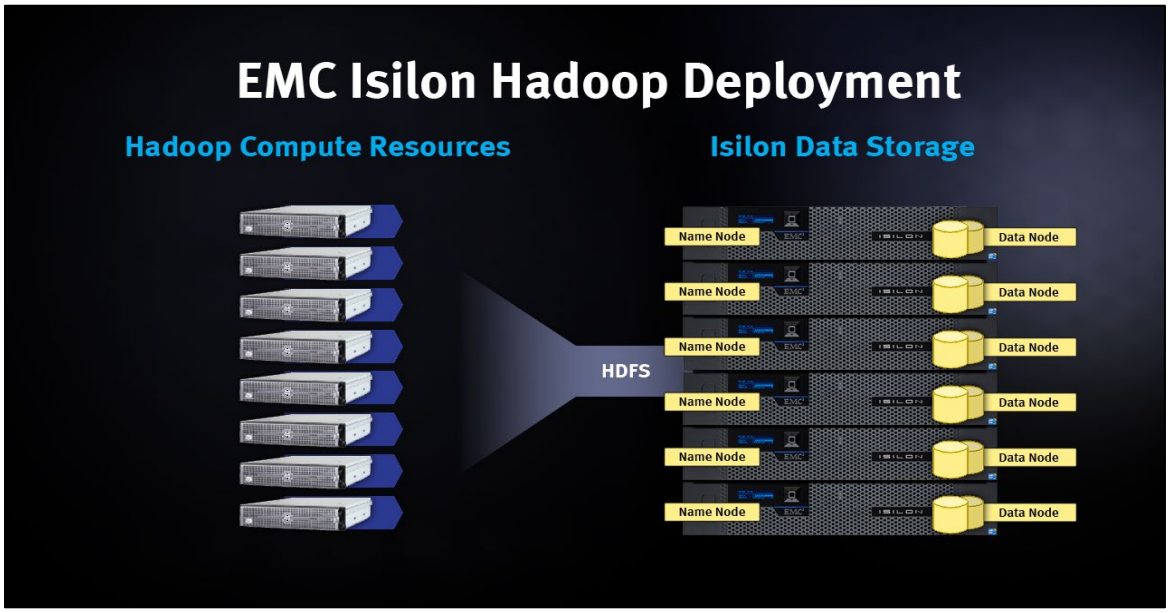
Hadoop compute clients can access the data that is stored on a PowerScale OneFS cluster by connecting to any node over the HDFS protocol. All nodes that are configured for HDFS provide NameNode and DataNode functionality. Each node boosts performance and expands the cluster capacity. For Hadoop analytics, 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 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 DataNode. 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.

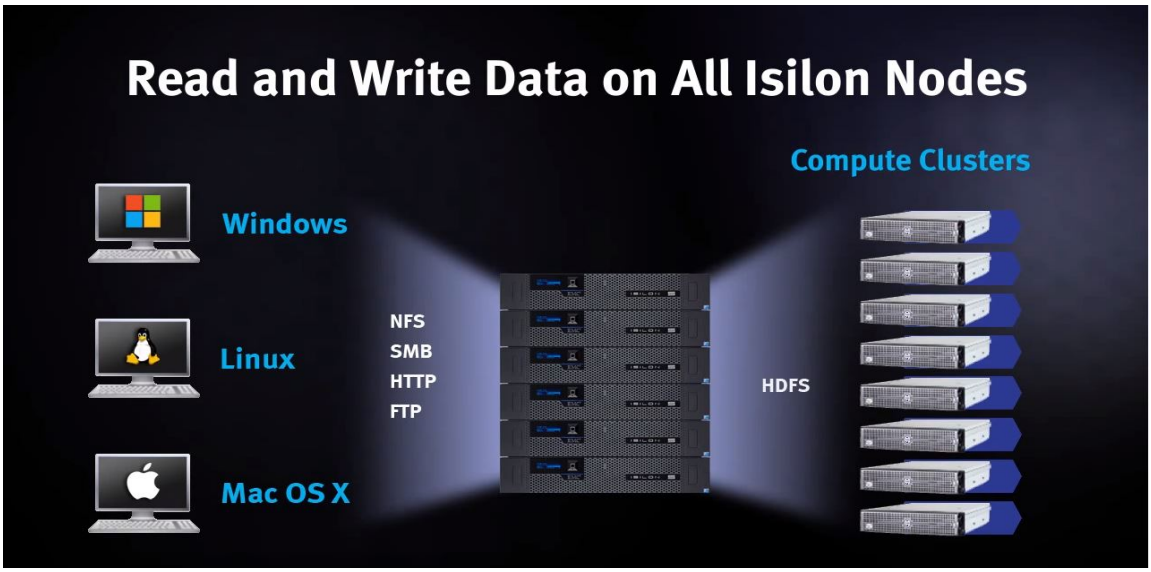


In a PowerScale OneFS cluster with Hadoop deployment, OneFS serves as the file system for Hadoop compute clients. On a PowerScale OneFS cluster, every node in the cluster acts as a NameNode and DataNode, providing automated failover protection.

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



The Hortonworks distribution is stored on the compute cluster, and the clients connect to the PowerScale OneFS cluster over the HDFS protocol to store and access Hadoop data.



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.

Ambari and Hortonworks Data Platform

OneFS Multi-tenant Installations

If the Hadoop cluster will be installed as a multi-tenant OneFS implementation, leveraging multiple Access Zones using the same Active Directory provider, additional configurations and considerations must be addressed before installing HDP and Ambari. It is recommended that you first consult the Isilon and Hadoop Multitenant Installation and Integration Guide, and then if required, engage Dell EMC Professional Services to understand the requirements and deployment strategies that are available to deploy Ambari.

Ensure that the following requirements are met:

- Hortonworks Data Platform (HDP) 3.0.1or later with Ambari 2.7.1or later.
- Password-less SSH configured
  - See the Hortonworks documentation for configuring Password-less SSH.
- Familiarity with the Ambari and Hortonworks documentation and the installation instructions
  - To view the Ambari and the Hortonworks Data Platform (HDP) documents, go to http://docs.hortonworks.com/index.html
  - Use the following table to record the components that you have installed.

Component	Version
Ambari version	
HDP stack version	
OneFS cluster name	
Isilon OneFS Ambari Management Pack	

OneFS cluster configuration

Ensure that the following requirements are met:

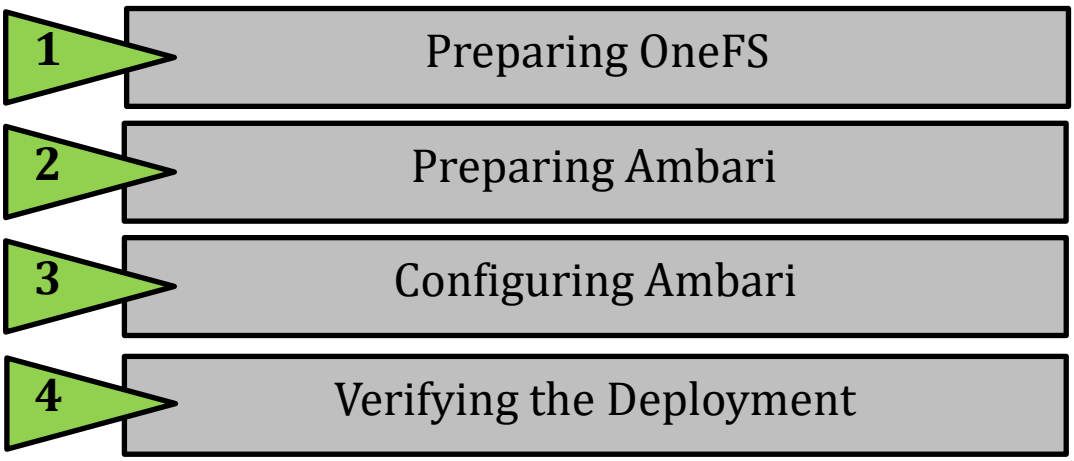
- A OneFS cluster running OneFS 8.1.2.0 or later.
- The OneFS cluster has access to your network and your network has access to the Internet. Internet access is required to download components.
- SmartConnect Advanced, a separately licensed OneFS module, is activated and SmartConnect is configured on your OneFS cluster.
- HDFS, a separately licensed 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 the 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	
SmartConnect module	
HDFS module	
OneFS cluster name	

Installing OneFS with Ambari

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

To complete the stages, you must perform tasks on both the Ambari cluster and the OneFS cluster.



Preparing OneFS

Complete the following steps to configure your OneFS cluster for use with Ambari and Hortonworks 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 the 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 licenses for HDFS are operational. If these licenses are not active and valid, some commands in this guide might 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 and configure the HDFS root in the access zone
- Create users and groups
- (Optional) Create an access zone
- (Optional) Create a SmartConnect zone

Use the following table to record the configuration information for the OneFS cluster with Hortonworks Ambari 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	
Ambari NameNode	

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-hdp --path=/ifs/data/zone1/hdp --create-path
```

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

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

Output similar to the following appears:

```
                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-hdp
                Path: /ifs/data/zone1/hdp
           Groupnet: groupnet0
      Map Untrusted: -
     Auth Providers: lsa-local-provider:zone1-hdp
       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/hdp/Hadoop

isi hdfs settings modify --zone=zone1-hdp --root-
directory=/ifs/data/zone1/hdp/Hadoop
```

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

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

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

3. View the properties of the existing pool.

```
isi network pools view --id=groupnet0:subnet0:Hadoop-pool-hdp
```

Output similar to the following appears:

```
                     ID: groupnet0.subnet0.hadoop-pool-hdp
              Groupnet: groupnet0
                Subnet: subnet0
                  Name: hadoop-pool-hdp
                 Rules: -
           Access Zone: zone1-hdp
     Allocation Method: static
     Aggregation Mode: lacp
   SC Suspended Nodes: -
         Description: hdp_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: hdp.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 Dell EMC Isilon Best Practices Guide 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 nodes NameNode process replies with the IP address of any OneFS node 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 hdp.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 cluster system account that will submit HDFS jobs or access the file system, you must create local users and groups on the OneFS cluster as Ambari cannot do this. 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 GIDs). Maintaining this parity is critical in the behavior of OneFS multiprotocol file access.

During installation the Hadoop installer creates all the required system accounts. For example, a Hadoop system account, *yarn,* is created with the UID of 502 and the GID of 500 on the Hadoop cluster nodes. 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 GID of 500 to ensure consistency of access and permissions.

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

There are many methods of achieving UID and GID 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 depends 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 GID 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 script configures on the OneFS cluster. The following procedures show how to manually add a single test user called *hduser1*.

Warning

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

Manual Steps to perform on the OneFS cluster

1. Add a group to the OneFS cluster.

```
isi auth groups create hduser1 --zone=zone1-hdp --provider local --gid <GID>
```

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-hdp --provider
local --home-directory /ifs/data/zone1/hdp/user/hduser1 --uid <UID> --enabled=true
```

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/data/zone1/hdp/hadoop/user/hduser1

chmod 755 /ifs/data/zone1/hdp/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 will access to run jobs.

1. Add the user to the Hadoop cluster.

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

Configure HDFS user for OneFS 8.1.2 and previous versions

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

On a node in the OneFS 8.1.2 cluster, create and configure the HDFS root directory.

1. View the HDFS service settings.

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

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-hdp --root-
directory=/ifs/data/zone1/hdp/hadoop
```

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 modify –add-user-mapping-rules=”hdfs=>root[]” --zone=zone1-hdp

isi zone modify --add-user-mapping-rules="yarn-ats-hbase=>yarn-ats" --zone=zone1-
hdp
```

Note: You can skip yarn-ats-hbase to yarn-ats user mapping in two cases as follows:

   a. By renaming yarn-ats-hbase principals to yarn-ats during Kerberization if the Timeline Service3 V2.0s is deployed as Embedded or System Service mode.

   b. You do not need to set the user mapping on OneFS if TLS v2.0 is configured on external HBase. For more details, see: https://docs.hortonworks.com/HDPDocuments/HDP3/HDP-3.0.1/data-operating-system/content/dosg_timeline_service_2.0_installation.html

4. Assign the Ambari NameNode in the access zone and associate the SmartConnect name with it.

```
isi hdfs settings modify --zone=<zone> --ambari-namenode=<my-smartconnectzone-name>
```

   For example:

```
isi hdfs settings modify --zone=zone1-hdp --ambari-namenode=hdfs.hop-isi-
m.solarch.lab.emc.com
```

5. Assign the Ambari Server to the HDFS zone

```
isi hdfs settings modify --zone=<zone> --ambari-server=<ambari-server.fqdn>
```

   For example:

```
isi hdfs settings modify --zone=zone1-hdp --ambari-server=amb-svr.hop-isi-
m.solarch.lab.emc.com
```

6. Create an indicator file in the Hadoop directory to view your OneFS cluster and access zone through HDFS.

```
touch /ifs/data/zone1/hdp/hadoop/THIS_IS_ISILON-hdp.txt
```

7. Modify the access control list (ACL) setting for OneFS.

   Run the following command on a node in the OneFS cluster 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 acls view
```

Configure HDFS for OneFS 8.2.0 and later versions

In OneFS 8.2.0, the HDFS user no longer needs to 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 new role and configure the backup and restore privileges to the HDFS user.

1. View the HDFS service settings.

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

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-hdp --root-
   directory=/ifs/data/zone1/hdp/Hadoop
   ```

3. Assign the Ambari NameNode in the access zone and associate the SmartConnect name with it.

   ```
   isi hdfs settings modify --zone=zone1-hdp --ambari-namenode=<my-smartconnectzone-
   name>
   ```

   For example:

   ```
   isi hdfs settings modify --zone=krbhdfs --ambari-namenode=hdfs.hop-isi-
   m.solarch.lab.emc.com
   ```

4. 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-hdp
   ```

5. 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-hdp
   ```

6. 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=zone-hdp
   ```

7. 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-hdp
```

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

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

For example:

```
isi auth roles view HdfsAccess --zone=zone1-hdp
       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
```

9. (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 do not need to 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 Ambari

Perform the steps that are discussed in this section on the Ambari hosts which become your Hadoop servers and clients.

Hadoop clusters and services rely heavily on DNS. All client hosts in your system must be configured for both forward and reverse DNS lookups. Validate that all hosts can resolve each other's hostnames and IP addresses.

Before you begin the installation of Ambari:

- Ensure that all your hosts meet the requirements that are determined by Ambari and Hortonworks to complete a successful Hadoop cluster installation. For more information and these installation guides, go to the Hortonworks website: http://docs.hortonworks.com/index.html.
- Ensure that you have access to the Isilon OneFS Ambari Management Pack . If not, contact your Dell EMC sales representative for more information about accessing the management pack. Steps to perform on the Hadoop client

  The Isilon OneFS Ambari Management Pack is a software component that can be installed in Ambari to define OneFS as a service in a Hadoop cluster. The management pack allows an Ambari administrator to start, stop, and configure OneFS as an HDFS storage service. This provides native namenode and datanode capabilities similar to traditional HDFS.

Steps to perform on the Hadoop client

1. To prepare Ambari for implementation, follow the instructions for your version of Ambari in the Hortonworks guide. See the **Installing Ambari** section.

   The guide provides the steps that you must perform to prepare the Ambari environment install Ambari and installing and configure HDP.

Important

Complete the steps in the Hortonworks guide in section 1, "Getting Ready," and section 4, "Installing Ambari." After you start the Ambari server, do not continue to section 6 of the *Hortonworks Guide* until after you have completed the instructions that are described in the Preparing OneFS section of this guide.

Complete the following steps that are described in the *Hortonworks Guide*:

1. Download the Ambari repository for the operating system that runs your installation host.
2. Set up the Ambari server.
3. Download the Isilon OneFS Ambari Management Pack installation bundle from the product download page and extract the contents on to the Ambari server.
4. Install the latest management pack on the Ambari server by running the following command:

   ```
   ambari-server install-mpack --mpack=NAME_OF_MPACK_TAR.tar.gz –verbose
   ```

   For example:

   ```
   ambari-server install-mpack --mpack=isilon-onefs-mpack-1.0.0.0-SNAPSHOT.tar.gz --verbose
   ```

   Output similar to the following displays:

```
Using python  /usr/bin/python
Installing management pack
INFO: Loading properties from /etc/ambari-server/conf/ambari.properties
INFO: Installing management pack isilon-onefs-mpack-1.0.0.0-SNAPSHOT.tar.gz
INFO: Loading properties from /etc/ambari-server/conf/ambari.properties
INFO: Download management pack to temp location /var/lib/ambari-
server/data/tmp/isilon-onefs-mpack-1.0.0.0-SNAPSHOT.tar.gz
INFO: Loading properties from /etc/ambari-server/conf/ambari.properties
INFO: Expand management pack at temp location /var/lib/ambari-
server/data/tmp/isilon-onefs-mpack-1.0.0.0-SNAPSHOT/
2018-11-07 06:36:39,137 - Execute[('tar', '-xf', '/var/lib/ambari-
server/data/tmp/isilon-onefs-mpack-1.0.0.0-SNAPSHOT.tar.gz', '-C',
'/var/lib/ambari-server/data/tmp/')] {'tries': 3, 'sudo': True, 'try_sleep': 1}
INFO: Loading properties from /etc/ambari-server/conf/ambari.properties
INFO: Loading properties from /etc/ambari-server/conf/ambari.properties
INFO: Stage management pack onefs-ambari-mpack-1.0 to staging location
/var/lib/ambari-server/resources/mpacks/onefs-ambari-mpack-1.0
INFO: Processing artifact ONEFS-addon-services of type stack-addon-service-
definitions in /var/lib/ambari-server/resources/mpacks/onefs-ambari-mpack-
0.1/addon-services
INFO: Loading properties from /etc/ambari-server/conf/ambari.properties
INFO: Loading properties from /etc/ambari-server/conf/ambari.properties
INFO: Adjusting file permissions and ownerships
INFO: about to run command: chmod  -R  0755 /var/lib/ambari-server/resources/stacks
INFO: process_pid=28352
INFO: about to run command: chown  -R -L root /var/lib/ambari-
server/resources/stacks
INFO: process_pid=28353
INFO: about to run command: chmod  -R  0755 /var/lib/ambari-
server/resources/extensions
INFO: process_pid=28354
INFO: about to run command: chown  -R -L root /var/lib/ambari-
server/resources/extensions
INFO: process_pid=28355
INFO: about to run command: chmod  -R  0755 /var/lib/ambari-
server/resources/common-services
INFO: process_pid=28356
INFO: about to run command: chown  -R -L root /var/lib/ambari-
server/resources/common-services
INFO: process_pid=28357
INFO: about to run command: chmod  -R  0755 /var/lib/ambari-server/resources/mpacks
INFO: process_pid=28358
INFO: about to run command: chown  -R -L root /var/lib/ambari-
server/resources/mpacks
INFO: process_pid=28359
INFO: about to run command: chmod  -R  0755 /var/lib/ambari-
server/resources/mpacks/cache
INFO: process_pid=28360
INFO: about to run command: chown  -R -L root /var/lib/ambari-
server/resources/mpacks/cache
INFO: process_pid=28361
INFO: about to run command: chmod  -R  0755 /var/lib/ambari-
server/resources/dashboards
INFO: process_pid=28362
```

```
INFO: about to run command: chown  -R -L root /var/lib/ambari-
server/resources/dashboards
INFO: process_pid=28363
INFO: about to run command: chown  -R -L root /var/lib/ambari-
server/resources/stacks
INFO: process_pid=28364
INFO: about to run command: chown  -R -L root /var/lib/ambari-
server/resources/extensions
INFO: process_pid=28365
INFO: about to run command: chown  -R -L root /var/lib/ambari-
server/resources/common-services
INFO: process_pid=28366
INFO: about to run command: chown  -R -L root /var/lib/ambari-
server/resources/mpacks
INFO: process_pid=28367
INFO: about to run command: chown  -R -L root /var/lib/ambari-
server/resources/mpacks/cache
INFO: process_pid=28368
INFO: about to run command: chown  -R -L root /var/lib/ambari-
server/resources/dashboards
INFO: process_pid=28369
INFO: Management pack onefs-ambari-mpack-1.0 successfully installed! Please restart
ambari-server.
INFO: Loading properties from /etc/ambari-server/conf/ambari.properties
Ambari Server 'install-mpack' completed successfully.
```

Note: The Isilon OneFS Ambari Management Pack includes a setting for Yarn that you may need to change. The Yarn Timeline Service 2.0 relies on Apache HBase for backend storage. As PowerScale is a single storage tier from Yarn's perspective, the storage policy for HBase is set to NONE in the Yarn-HBase-site. If your Yarn deployment uses an external HBase for Timeline Service 2.0, then the storage policy settings should be changed to the HBase default, HOT, or whatever is appropriate for your environment.

5. Start the Ambari server.

6. Logging in to the Ambari server opens the create cluster page.

   Important: Do not continue to section 6 of the *Hortonworks Guide* until the OneFS cluster is prepared as described in the following steps and is ready to be integrated into Ambari during the installation.

Configuring Ambari 2.7.1and later

Perform the steps in section 6 as described in the Hortonworks Data Platform Apache Ambari installation guide, "Installing, Configuring, and Deploying a Cluster." An outline of the steps that you must perform to configure Ambari and Hortonworks for OneFS is as follows:

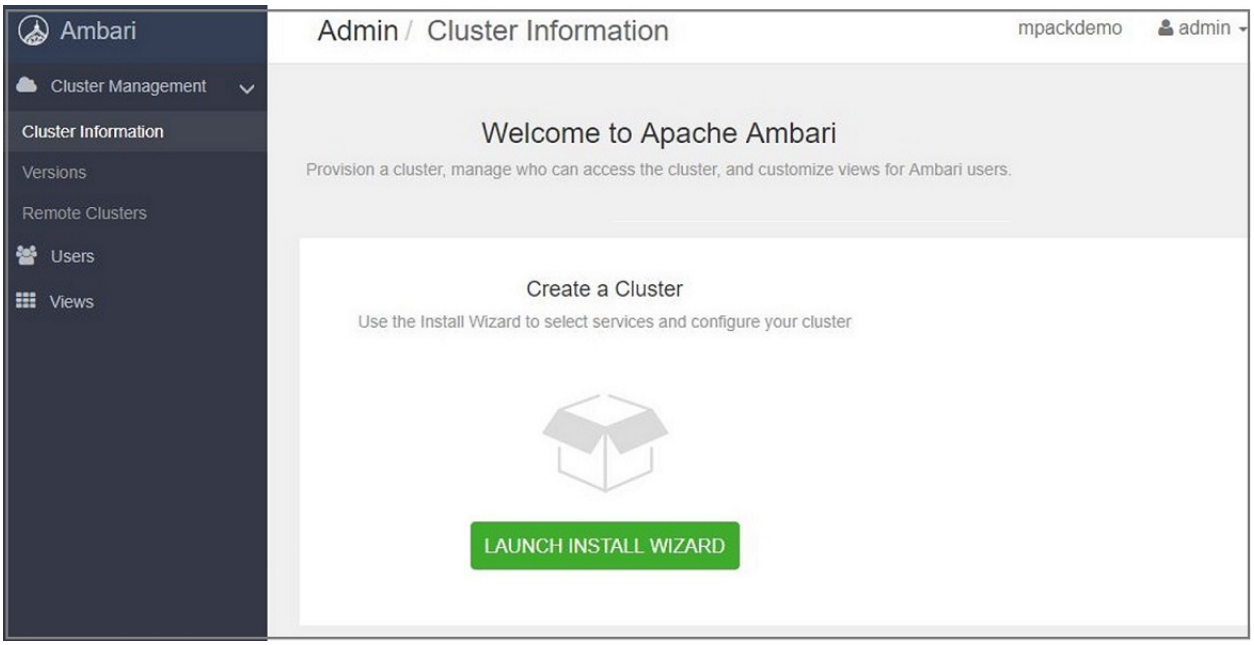
1. Choose **OneFS** instead of the **HDFS** deployment option.

2. In the **OneFS Service Settings** section, specify your SmartConnect FQDN and any other HDFS configuration settings that you want to change.

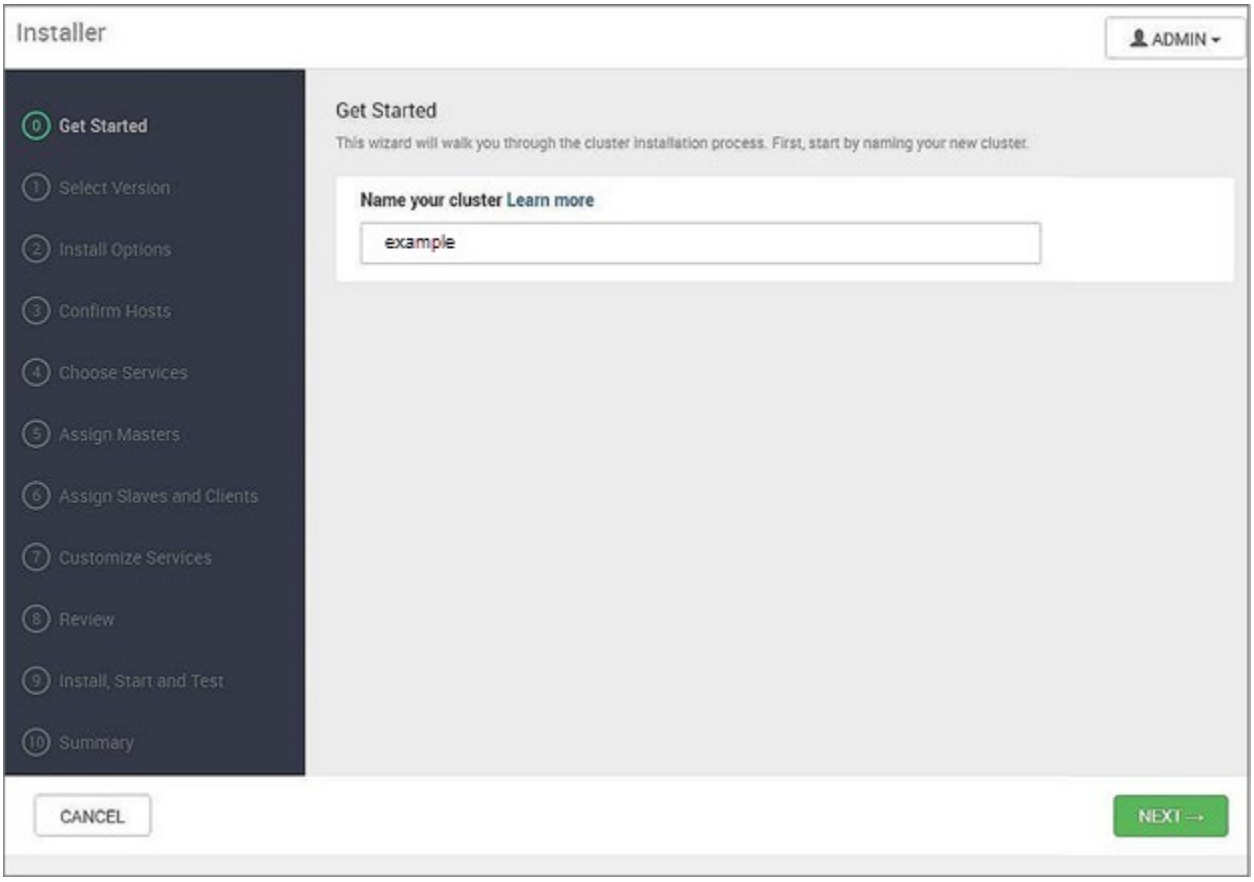
Steps to perform in your browser from your client

1. After starting the Ambari service, open Ambari Web using a web browser.

2. Point your browser to http://<your.ambari.server>:8080, where <your.ambari.server> is the name of your Ambari server host.
3. Log in to the Ambari server using admin both as the default username and password. You can change these credentials later.
4. For a new cluster, the Ambari install wizard displays a Welcome page through which you can launch the Ambari Install wizard.

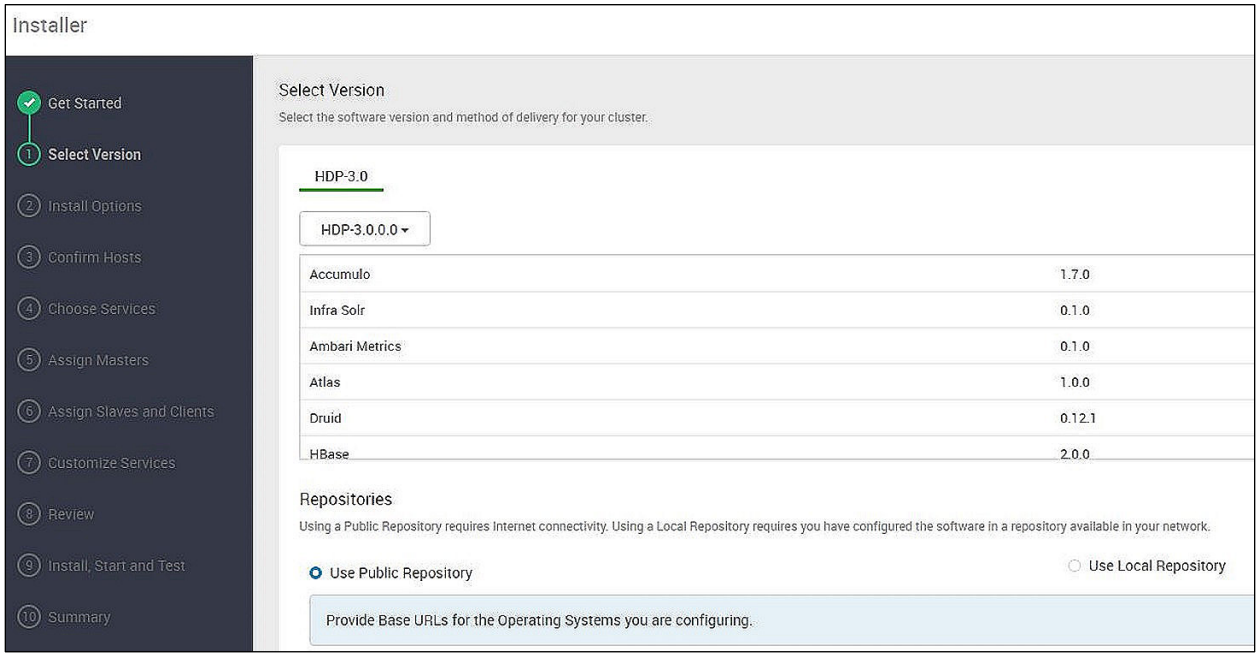


5. Select **Launch Install Wizard** and complete the installation steps in the Hortonworks guide, including the following steps that are specific to OneFS.

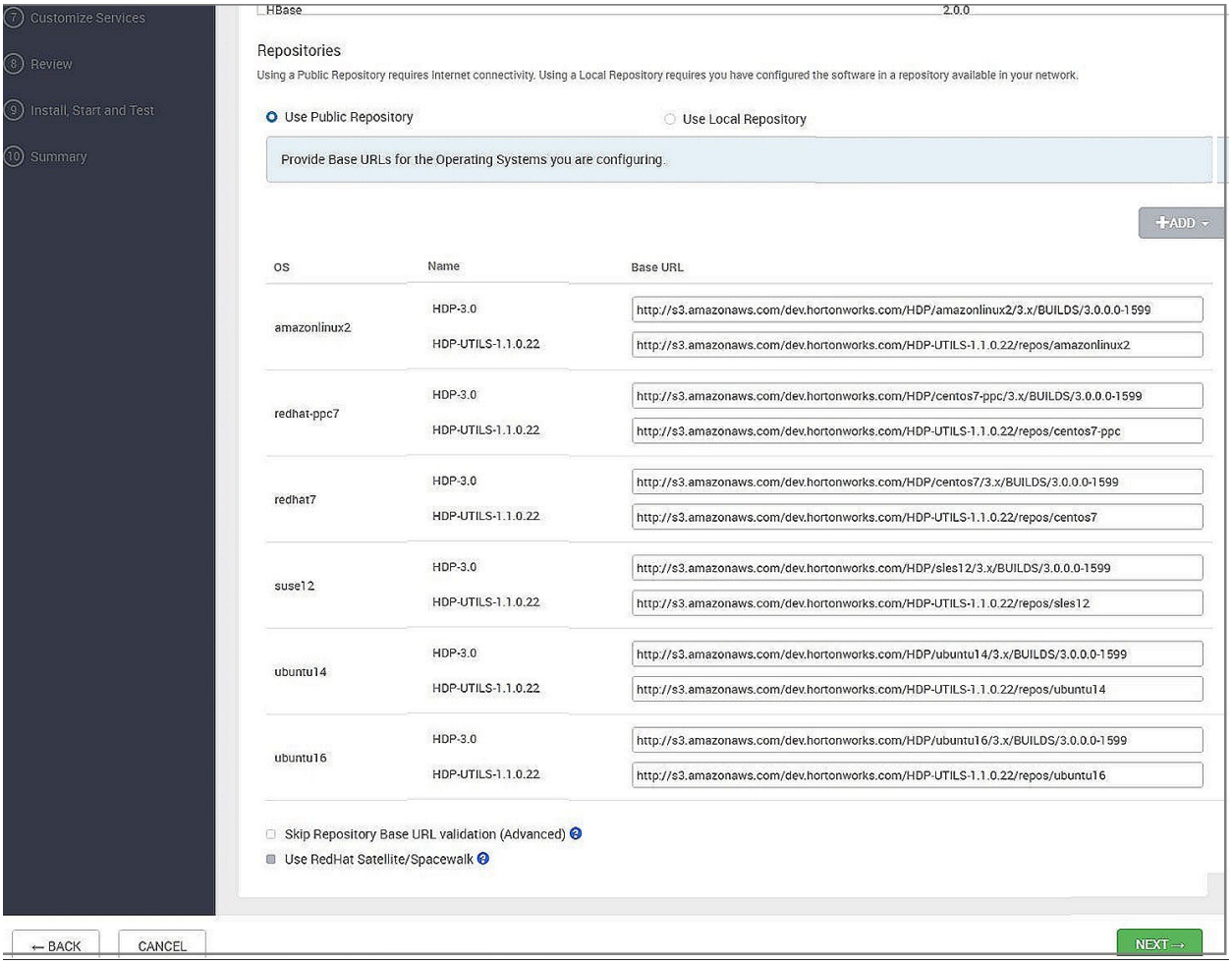


6. In the **Name your cluster** field, type a unique name for the cluster.
Note: As a best practice, Ambari Cluster names should be fewer than seven characters. Longer cluster names require additional configuration for multitenant AD installation due to OneFS specific requirements. Use abbreviated cluster names where possible to facilitate integration with OneFS. For example: hdp1, h-prod1, or similar.

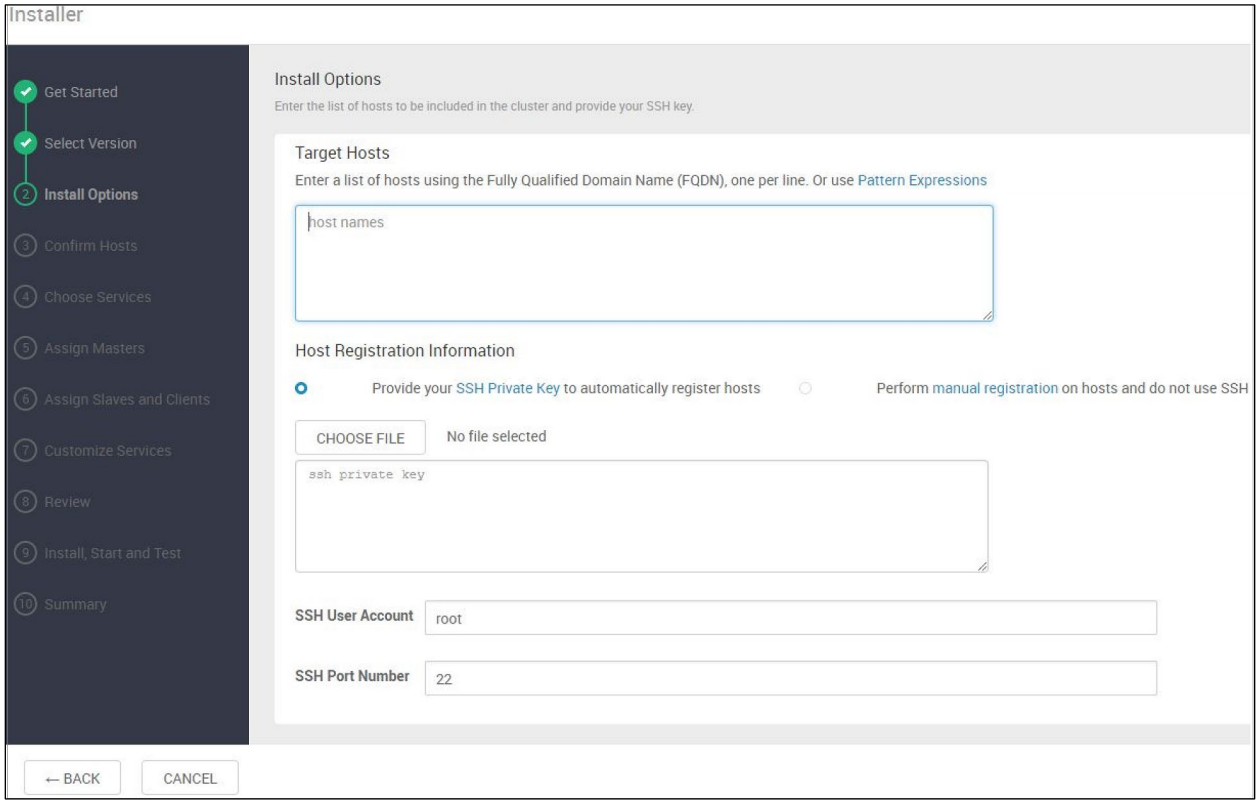
7. On the **Select Version** screen, select the version of the HDP stack and the repository to configure it. Consult Hadoop Distributions and Products Supported by OneFS for the version that you must configure with your OneFS cluster.



8. Click one of **Use Public Repository** or **Use Local Repository** based on the availability of Internet access.



9. On the **Install Options** screen, under **Target Hosts**, type the fully qualified name of all the Linux hosts that participate in the Hadoop cluster.



10. Under **Host Registration Information** attach the SSH private key.

    The creation of this host and key was performed before running the Ambari wizard. See the Hortonworks Ambari documentation for additional details.

Important

Do not include the OneFS cluster in the target hosts.

To obtain the private key:

a. Log in to your Ambari host server.

b. Generate the private key as shown:

```
ssh-keygen
```

c. Copy the key to all the hosts as shown:

```
ssh-copy-id root@XXXXX
```

d. Change to the `/root/.ssh` directory.

e. Output the contents of the `id_rsa` file.

```
cat /root/.ssh/id_rsa
```

f. Copy the output to a file and save the file on your desktop.

g. Copy the file to the machine on which you are running the web-based Ambari Install Wizard.

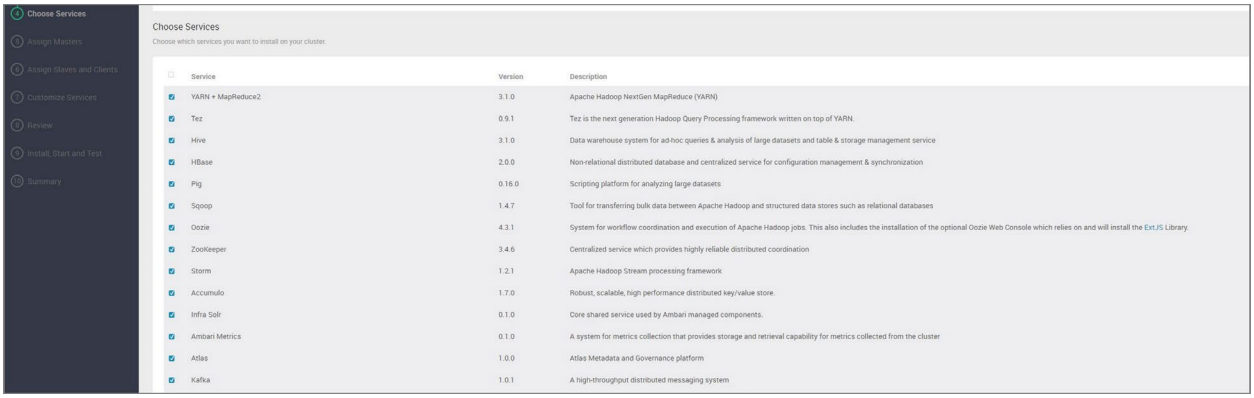
h. Copy the entire contents of the `id_rsa` file, including the leading dash characters, into the **Host Registration Information** box in the **Install Options** screen of the Ambari wizard.

   If no keys exist, see the Ambari documentation on generating a private key and setting up keyless SSH access on hosts in the cluster.

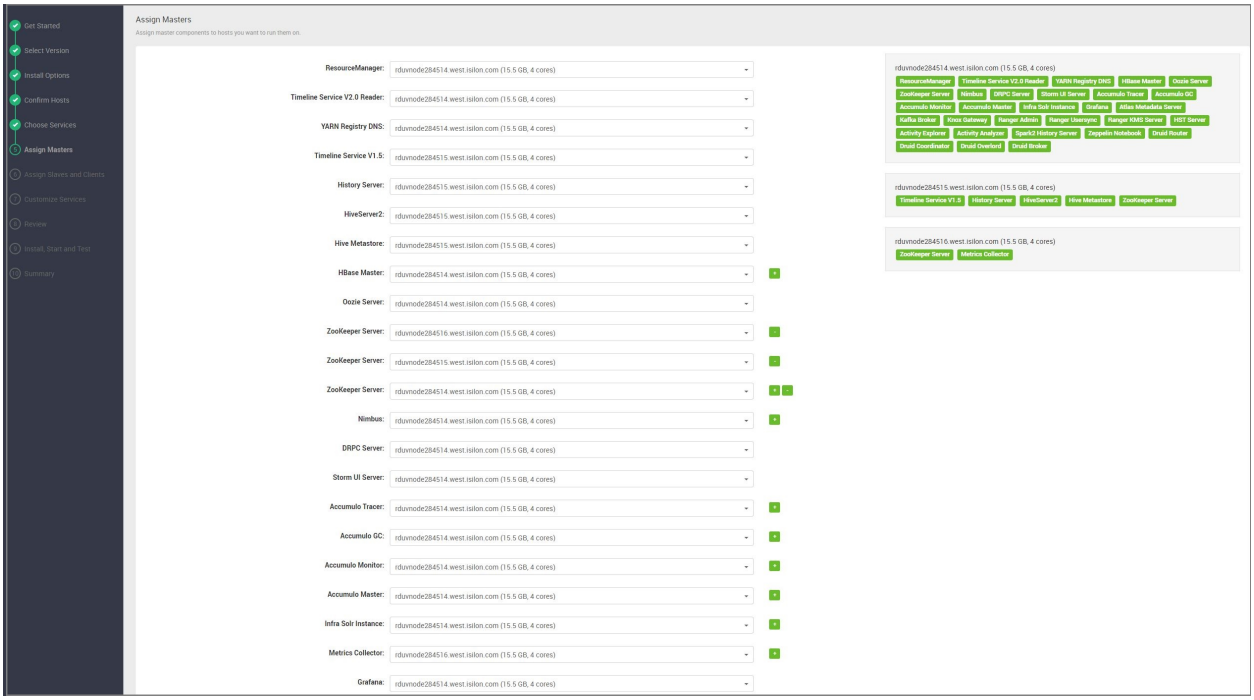
i. Select **Register and Confirm**.

11. On the **Choose File System** screen, select **OneFS**.

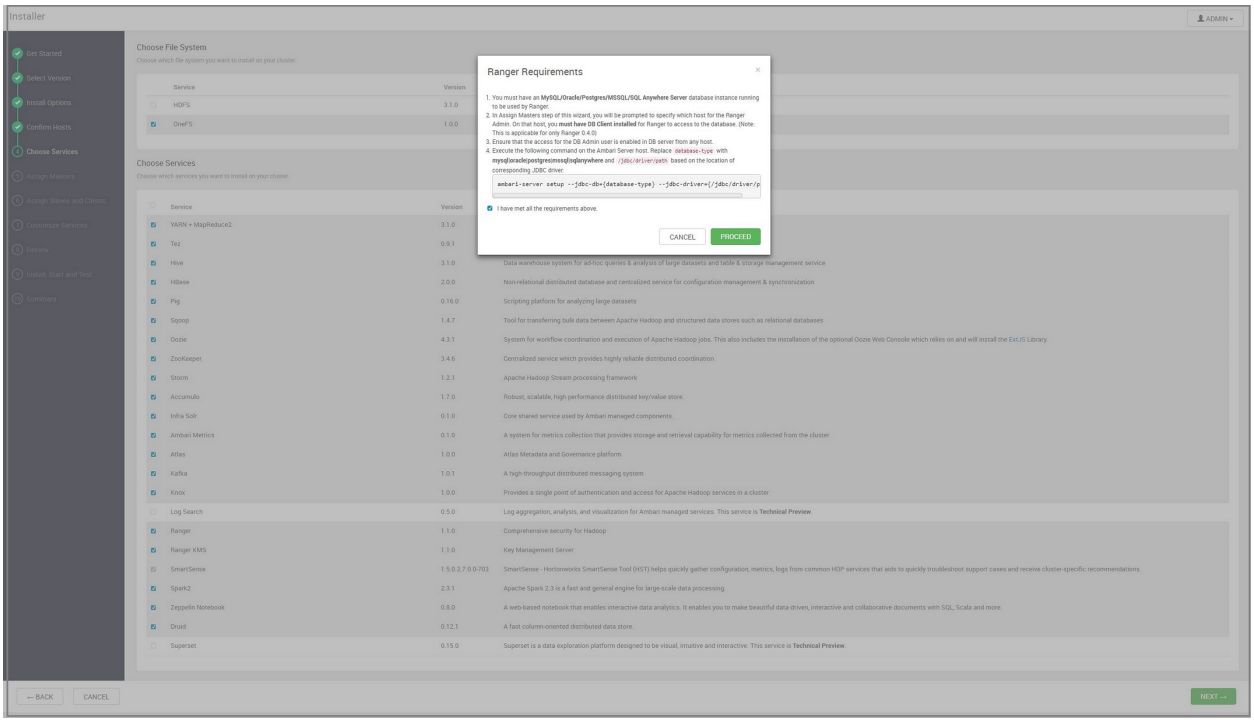
12. On the **Choose Services** screen, retain the default settings.



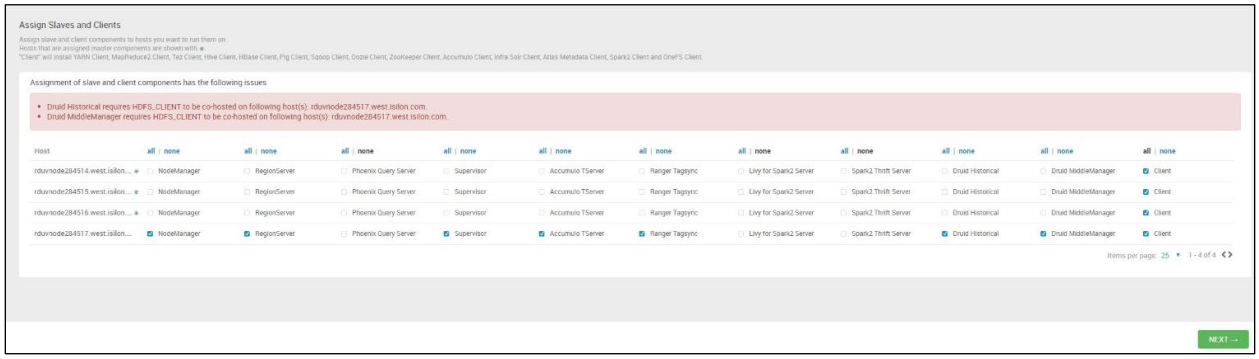
13. On the **Assign Masters** screen, retain the default settings.



14. Ensure that all the ranger requirements are met before you click **Proceed** on the **Ranger Requirements** screen.

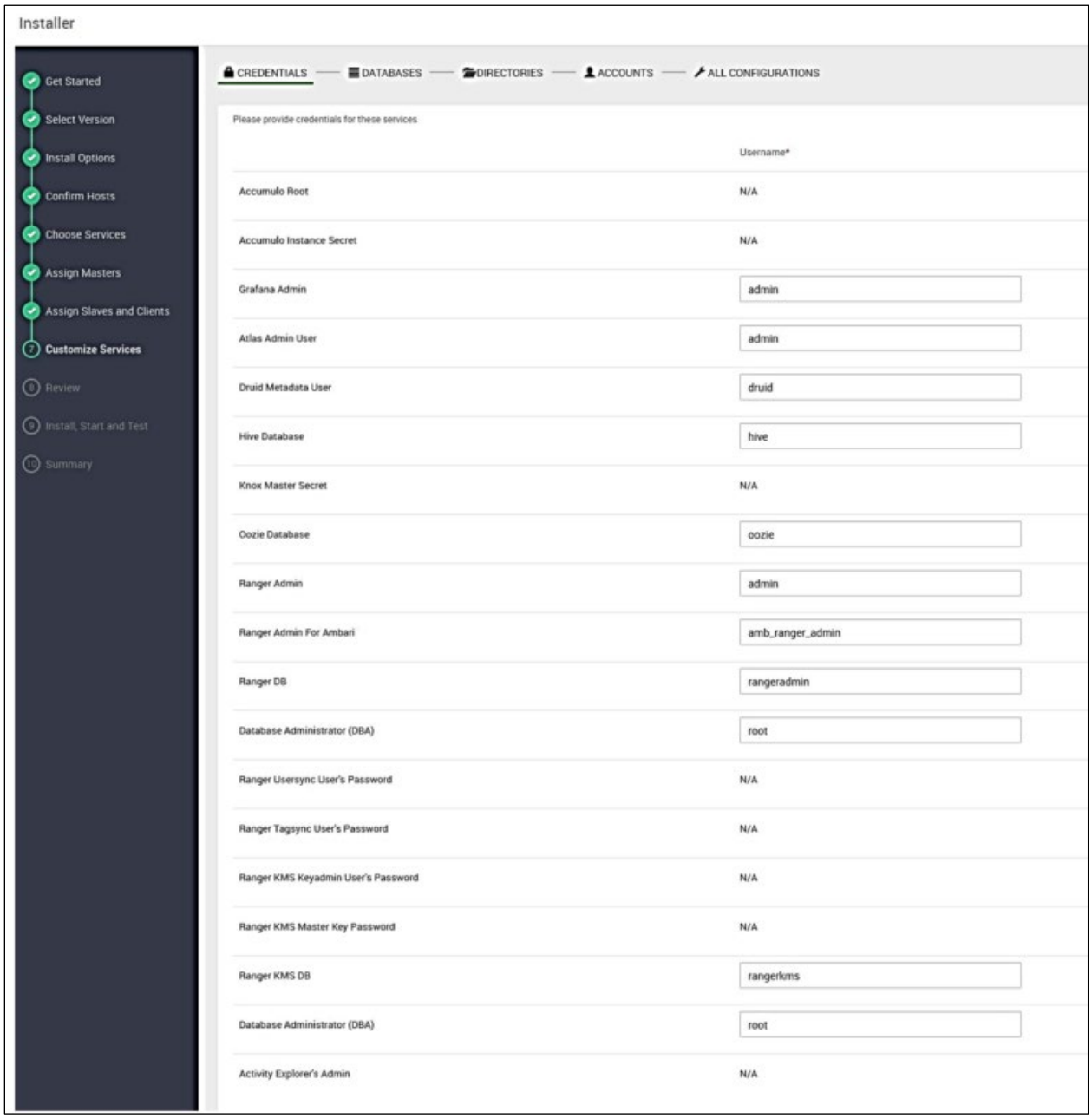


15. On the **Assign Slaves and Clients** screen, ensure that all the clients are selected.

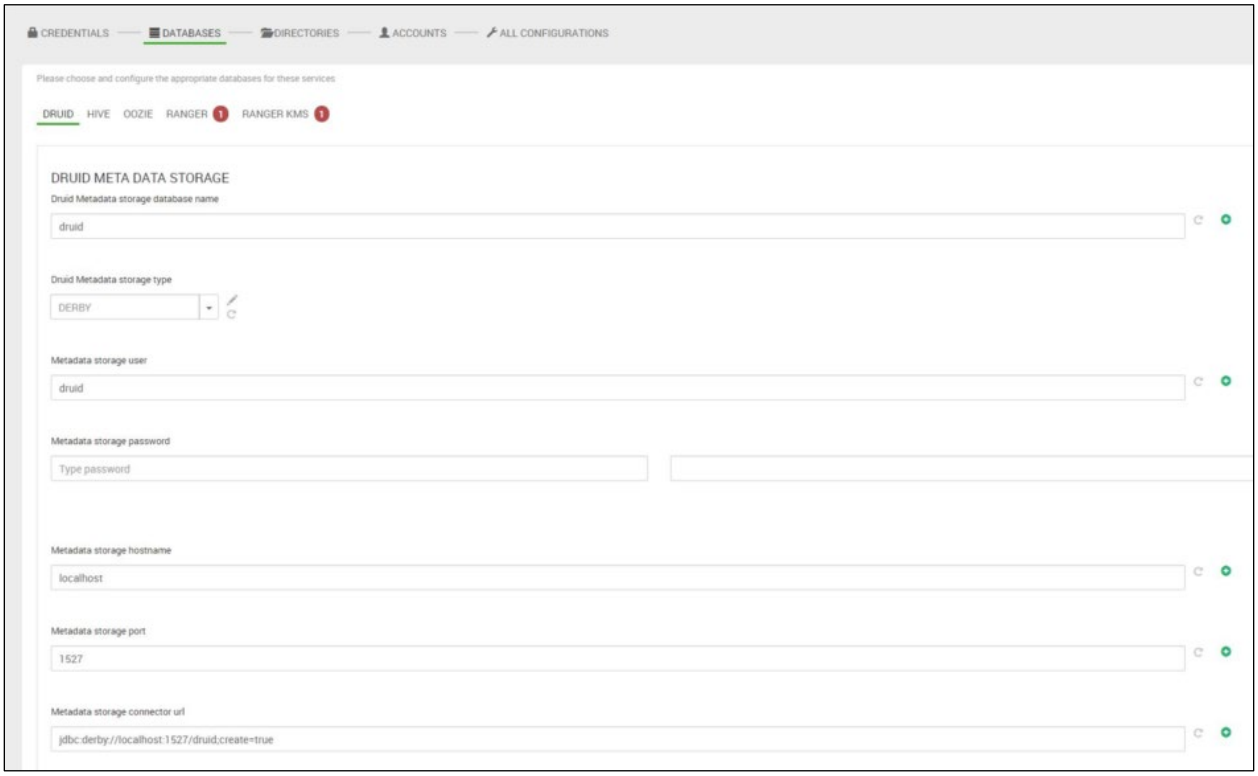


16. On the **Customize Services** screen, specify the settings on the different tabbed pages as shown:

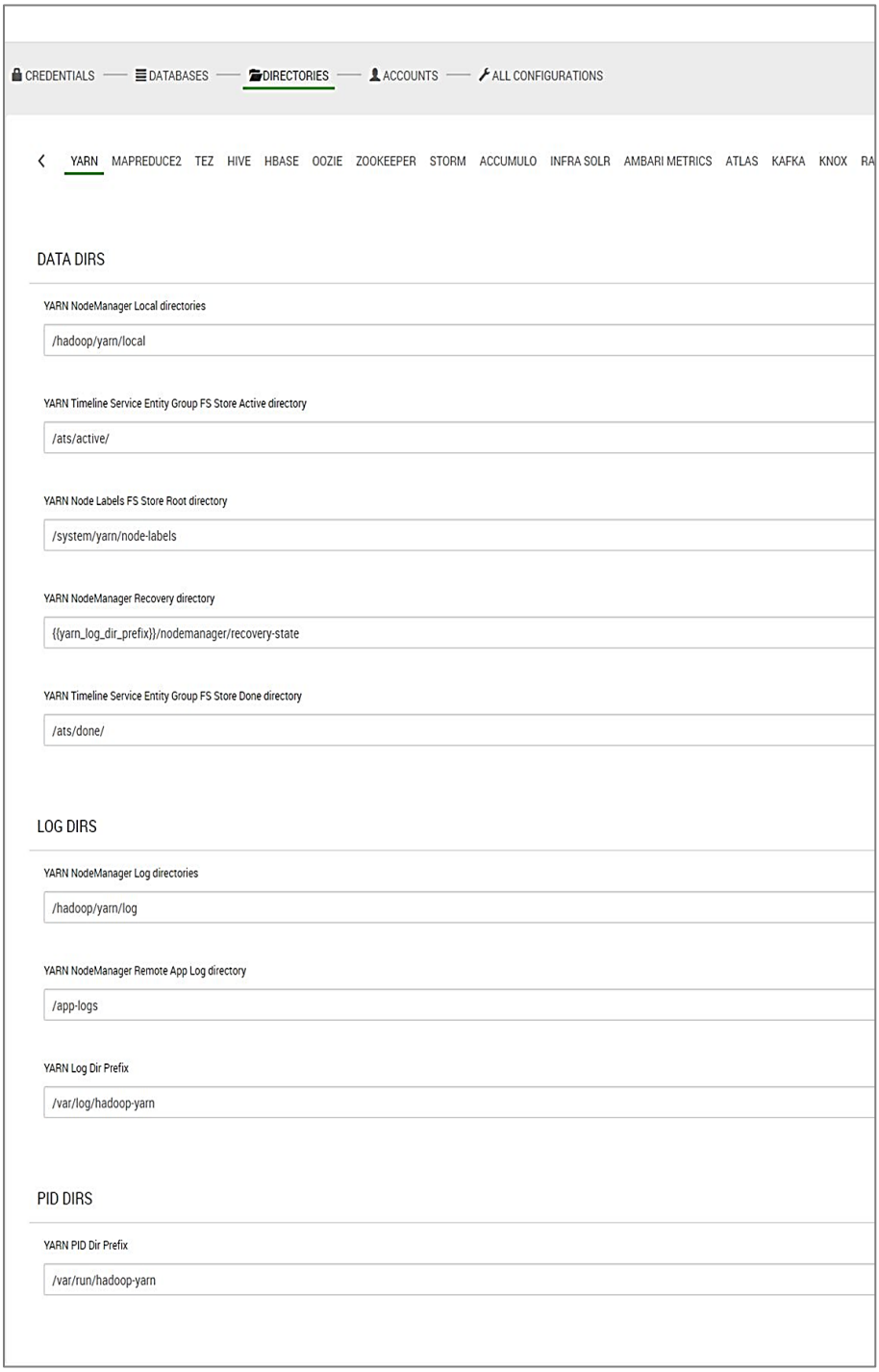
    a. On the CREDENTIALS tabbed page, specify values as shown in the following screen:



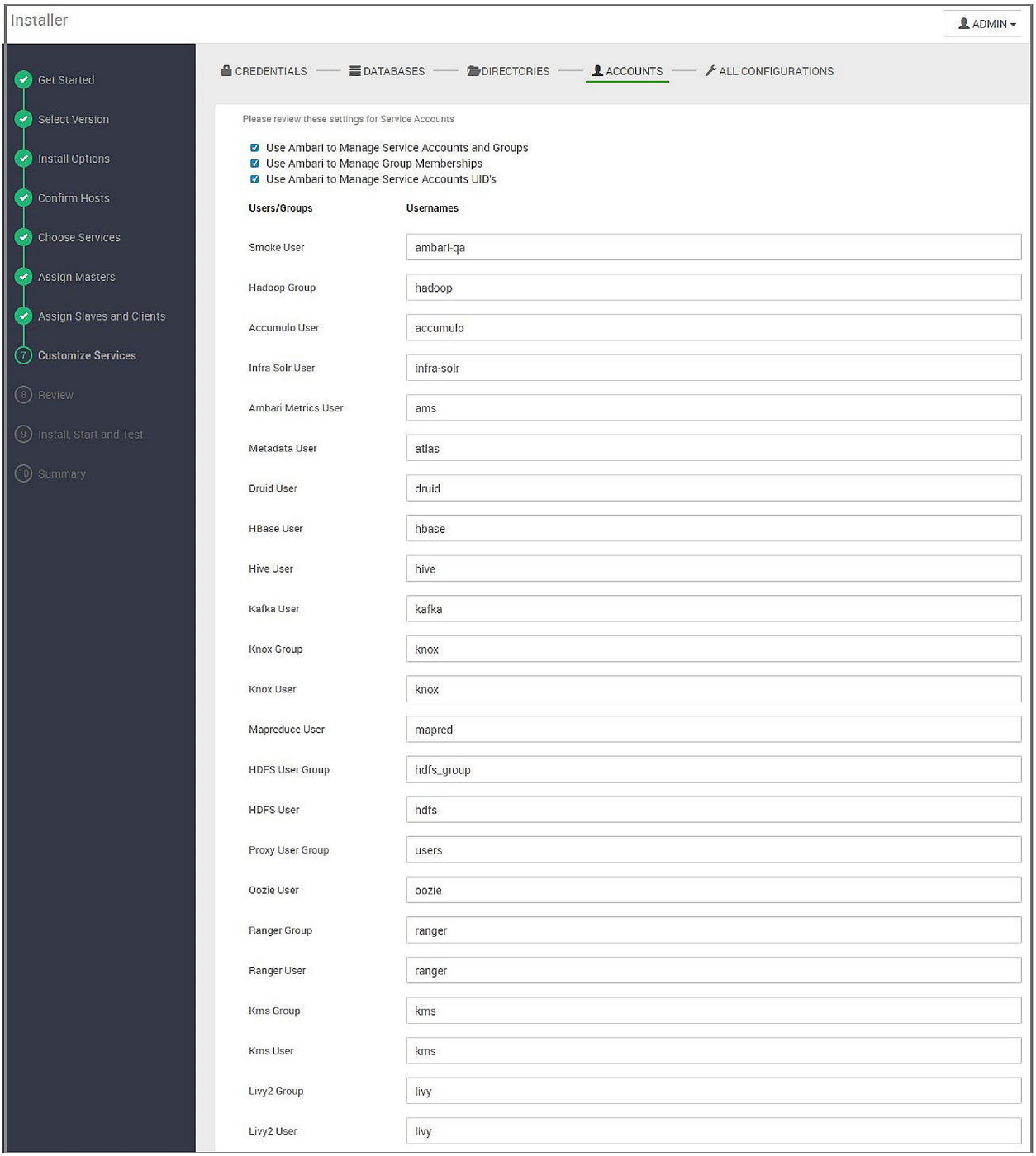
b. On the DATABASES tabbed page, configure your settings as shown in the following screen:



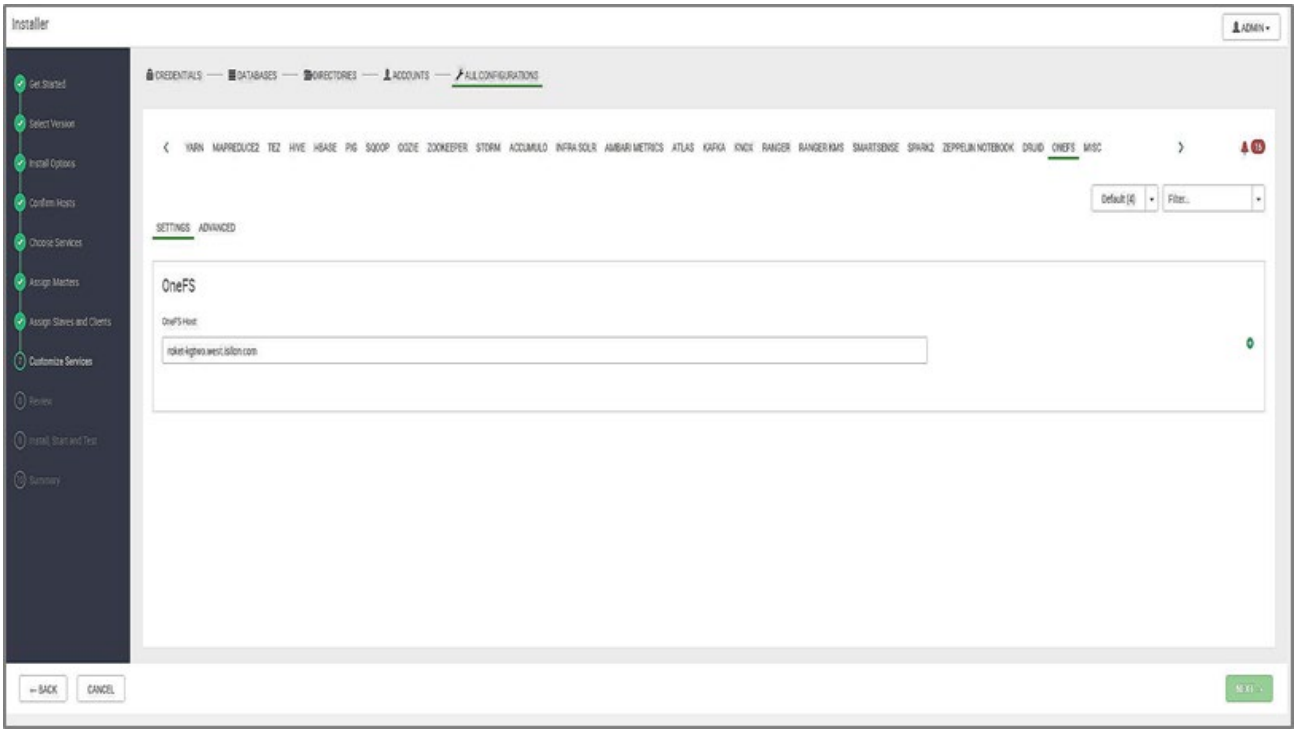
c. On the DIRECTORIES tabbed page, configure your settings as shown in the following screen:



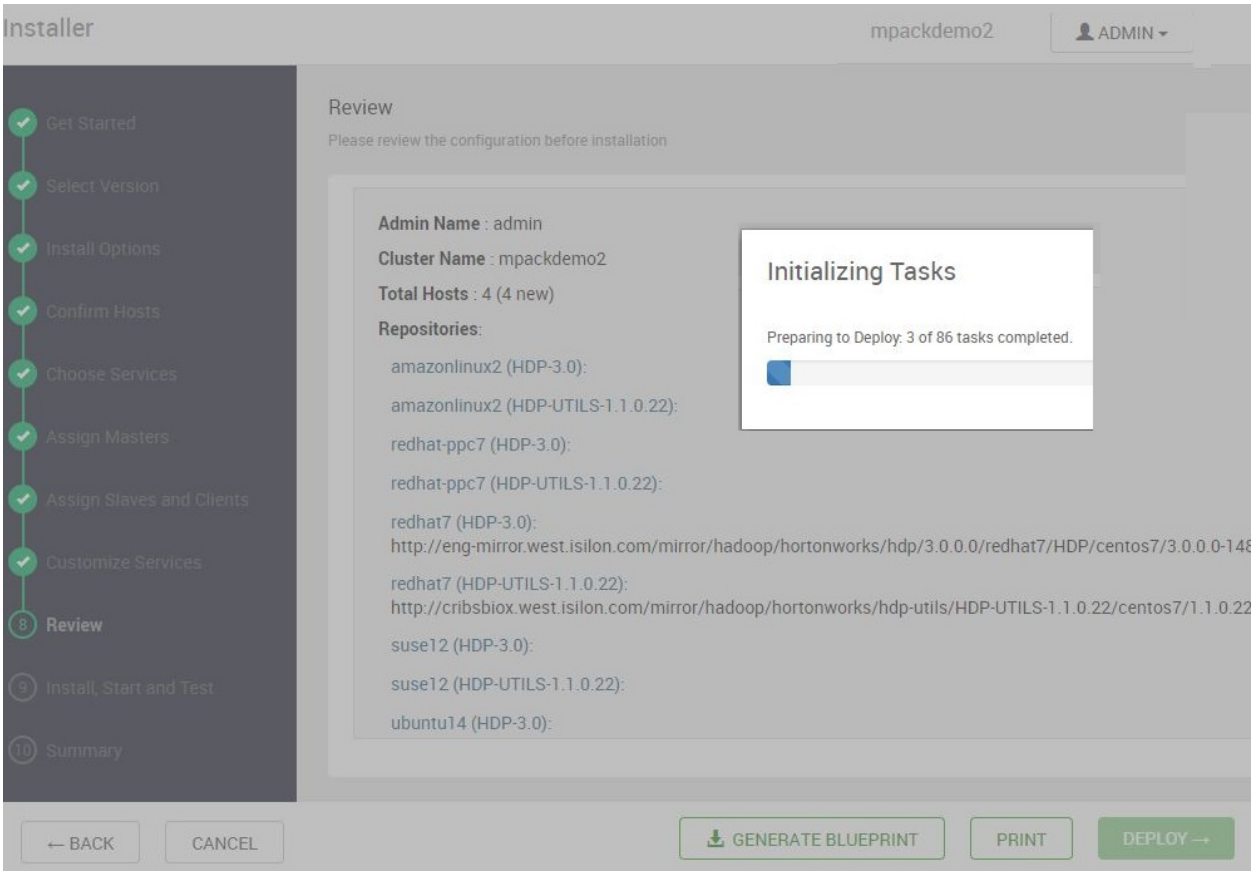
d. On the ACCOUNTS tabbed page, configure your settings as shown in the following screen:



e. On the ALL CONFIGURATIONS tabbed page, click **OneFS** and configure your settings as shown in the following screen:



17. On the **Review** screen, review the configuration for accuracy. Click **Deploy**.

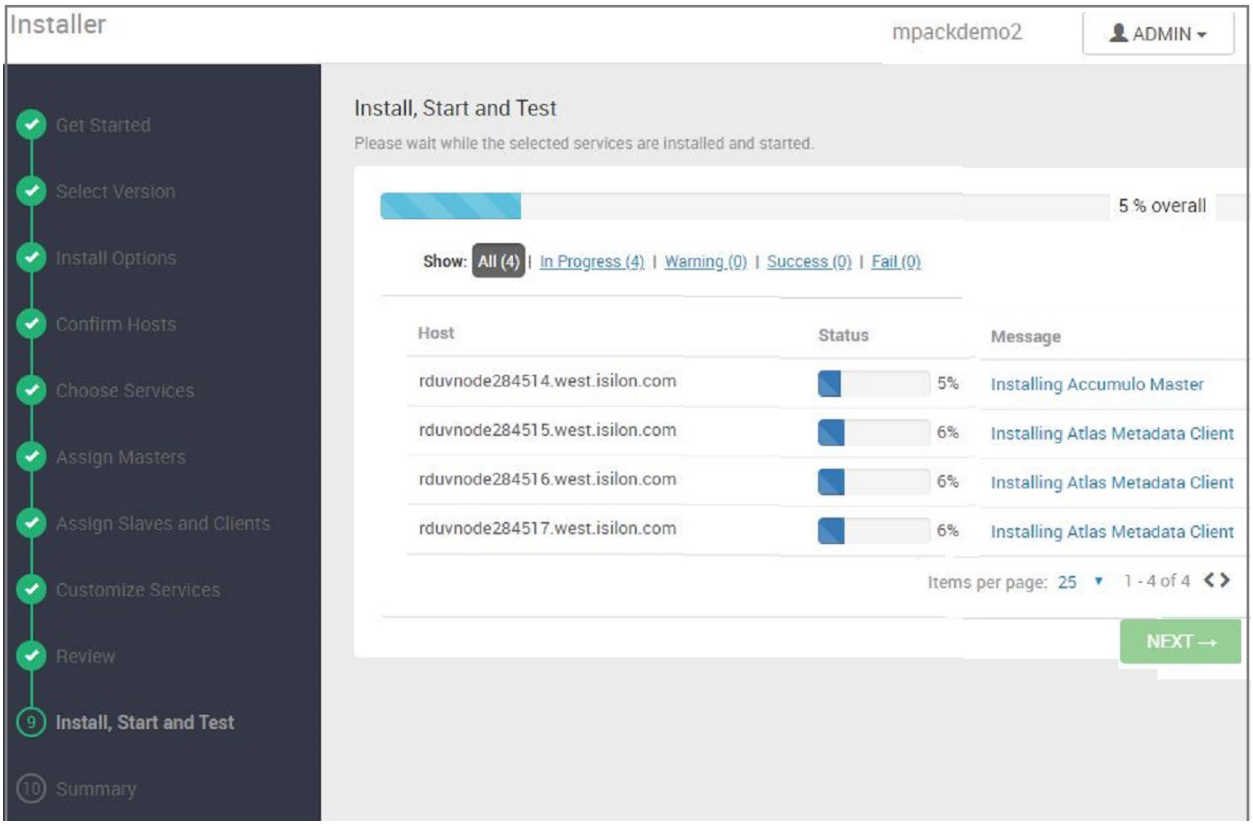


18. On the **Install, Start and Test** screen, wait for the test results. The tests should indicate success for both the OneFS and compute nodes.

    A progress indicator appears.

19. Wait until the status is 100% complete and then click **Next**.

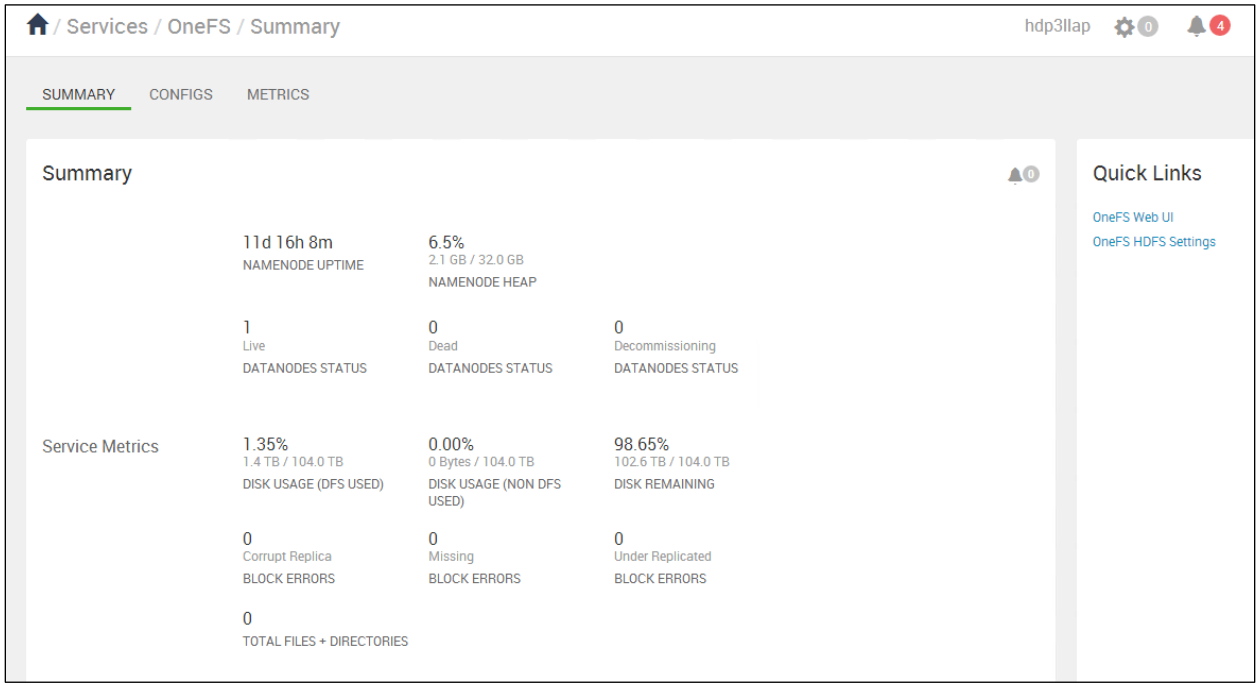
    See the Hortonworks documentation for guidance on how to manually restart the services in case some services fail to install and start up.



20. On the **Summary** screen, click **Complete**.
    Ambari starts automatically. All the services on the left pane should appear with a green checkmark.
21. Run the command as shown in the following example to configure the Ambari server for collecting OneFS metrics:

```
isi hdfs settings modify --zone=zone_name --ambari-metrics-collector=<FQDN_of_HDP
client_running_Ambari_metrics_collector_service>
```

22. Verify that Ambari can collect metrics from the OneFS SmartConnect zone as shown in the following screen:



You are now configured to use a OneFS cluster with Hortonworks for Apache Hadoop.

Configuring Ambari-Automated KDC-Based Kerberos with OneFS

You can configure Ambari-automated Kerberos using existing Microsoft Active Directory or MIT KDC installations.

Prerequisites

Before you configure Ambari-automated Kerberos on your OneFS cluster, ensure that the prerequisites that are described in the following sections are met.

PowerScale OneFS

Ensure that OneFS is preconfigured appropriately to respond to requests related to secure Kerberized HDFS, authenticated by MIT KDC or Active Directory providers.

Ambari and Hortonworks Data Platform

Ensure that the following prerequisites are met:

- You must be running Ambari 2.7.1or later and HDP 3.0.1or later.
- If you are using an existing MIT KDC installation, ensure that MIT KDC is running. If not, follow the steps that are recommended in the Ambari Security Guide to set up your Kerberos infrastructure.
- Forward and reverse DNS lookups must be enabled on all the hosts.
    - All the compute hosts must have forward DNS lookup resolved correctly for all the hosts.
    - OneFS SmartConnectZonename lookups must resolve correctly.
    - Reverse PTR records for all IP addresses in the SmartConnect pool must exist.
    - OneFS must be able to resolve all the hosts, KDCs, and Active Directory servers as needed.
- Test and validate all the host names and IP lookups before Kerberization:
    - Ambari must be able to manage and deploy `keytab` and `krb5.conf` files.
    - All the services must be running on the Ambari dashboard.
    - Before you migrate between secure and nonsecure clusters, you must remove `ats-hbase` from the source cluster. Otherwise, Timeline Service 2.0 might stop working. Read the Hortonworks Documentation for more information. Use the `yarn app -destroy` command to destroy the `ats-hbase` service.

      ```
      su - yarn-ats
      yarn app -destroy ats-hbase
      ```

    - Remove the `ats-hbase` configuration from HDFS.

      ```
      su - yarn-ats
      hadoop fs -rm -R ./{stack_version}/*
      ```

    - Delete the specification file for `ats-hbase` from HDFS.

```
su - hdfs
hadoop fs -rm -R /services/sync/yarn-ats/hbase.yarnfile
```

Enable Ambari-automated KDC Kerberos with OneFS on Hortonworks using MIT KDC

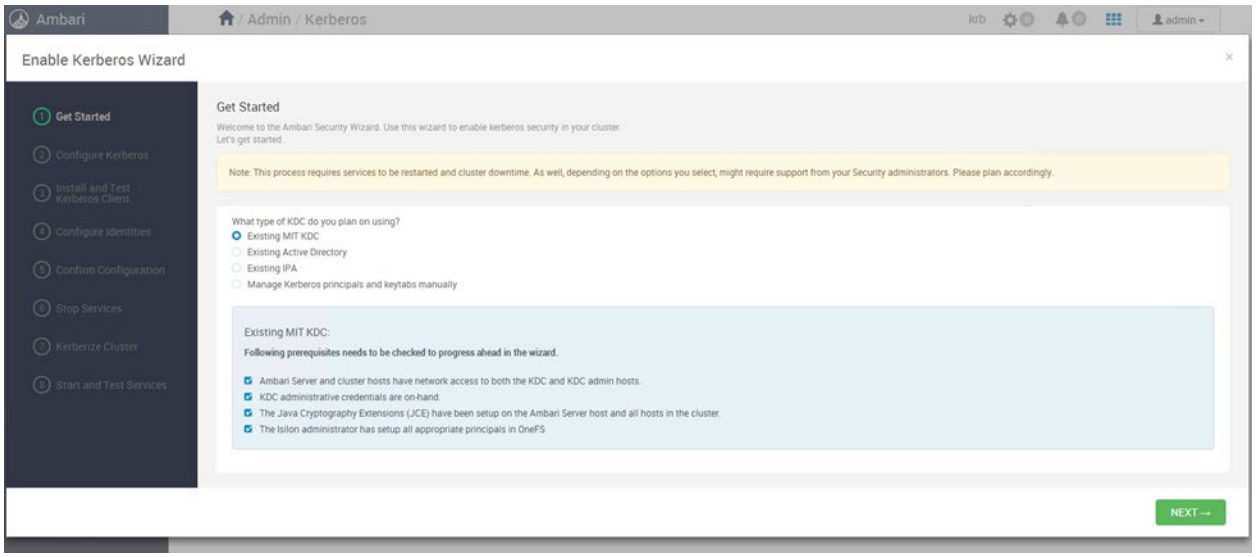
Use this procedure to enable Ambari-automated Kerberos on an HDP cluster using MIT KDC.

1. Log in to Ambari Web and click **Admin** › **Kerberos**.
2. Click **Enable Kerberos**. The following warning message might appear, depending on your configuration settings:

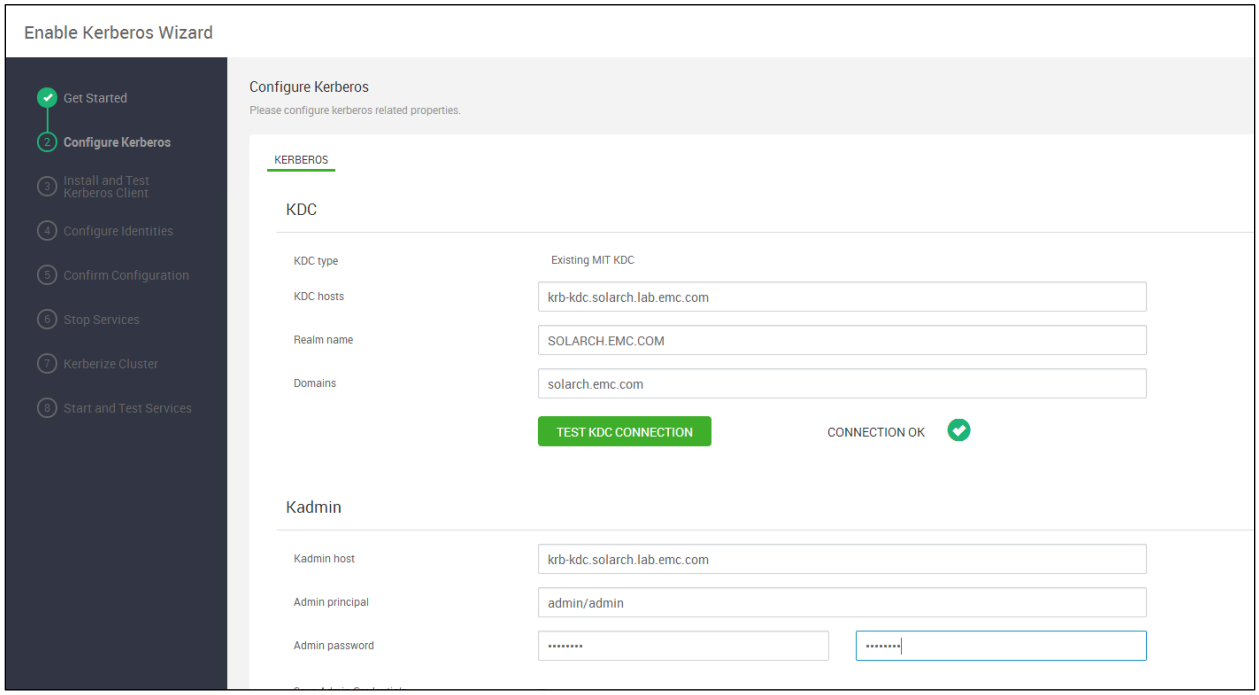
   ```
   YARN log and local dir will be deleted and ResourceManager will be formatted as
   part of Enabling/Disabling Kerberos.
   ```

   Ignore this message and go to the next step.
3. On the **Get Started** screen of the Ambari Security Wizard, select the following options:
   a. Select **Existing MIT KDC.**
   b. In **the Existing MIT KDC** area, select all the check boxes to confirm that you have checked and completed all the prerequisites. For more information about setting up and meeting the requirements and prerequisites, see the Ambari Security Guide.

      The box for “The Isilon administrator has setup all appropriate principals in OneFS” can remain checked, this is completed in a later step.



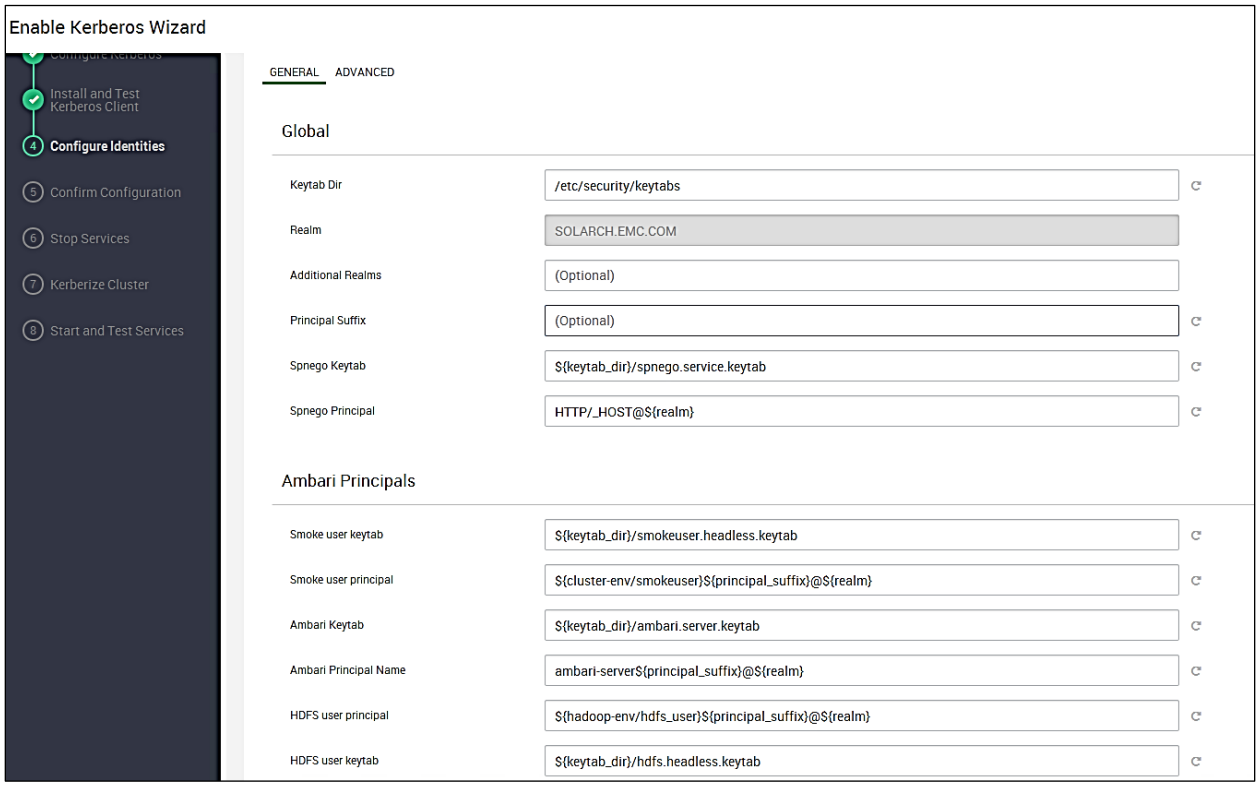
4. On the **Configure Kerberos** screen, specify the KDC and admin server information.



5. On the **Install and Test Kerberos Client** screen, run a test that deploys and configures Kerberos clients on all the hosts. In this step, the Ambari server performs a smoke test to ensure that you have configured Kerberos correctly.

6. On the **Configure Identities** page, specify mapping rules that are specific to configuring Kerberos for a OneFS cluster:

    a. Click the **General** tab and configure the Apache Ambari user principals as shown in the following screen. Remove Principal Suffix content from the **Global** option.



    b. On the Advanced tab, update the properties as specified below in Table 1.

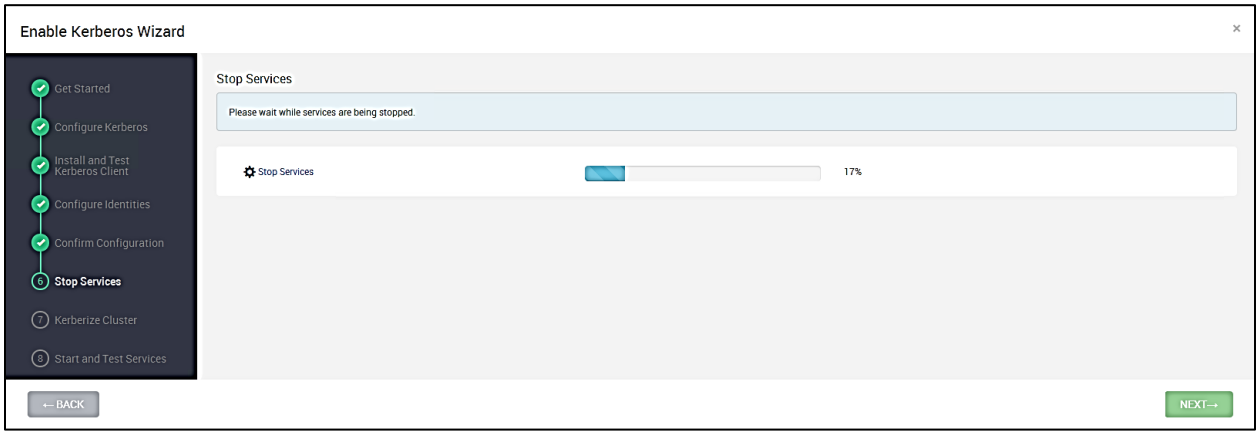
Table 1. Ambari configuration properties and required values

Hadoop service component	Ambari configuration property name	Default value	Required value
YARN	yarn.nodemanager. keytab	${keytab_dir}/nm. service.keytab	${keytab_dir}/yarn.service. keytab
YARN	yarn.nodemanager. principal	nm/_HOST@${realm}	yarn/_HOST@${realm}
YARN	yarn.resourcemanager. keytab	${keytab_dir}/rm. service.keytab	${keytab_dir}/yarn.service. keytab
YARN	yarn.resourcemanager. principal	rm/_HOST@${realm}	yarn/_HOST@$ {realm}
MapReduce2	mapreduce.jobhistory. keytab	${keytab_dir}/jhs. service.keytab	${keytab_dir}/mapred.service. keytab
MapReduce2	mapreduce.jobhistory. principal	jhs/_HOST@$ {realm}	mapred/_HOST@$ {realm}

7. On the **Confirm Configuration** screen, review the settings before you proceed.

   Note: If you exit the **Enable Kerberos Wizard** at this stage, all the configurations and customizations that you have specified so far will be lost and you must restart the configuration process again.

   On the **Stop Services** screen, all the Hadoop services in Ambari, user activities, and services on all the hosts are stopped automatically.



8. Click **Next** when all the services are stopped successfully to move to the **Kerberize Cluster** screen. The Kerberization process is automatically initialized. The Ambari services are kerberized, user principals are created, and keytabs are distributed.

   Warning: Do not click **Next** on the **Kerberize Cluster** screen until you have successfully completed the next step.

9. Connect to a OneFS cluster and perform the following steps:

   a. Ensure that your access zone is configured to use MIT KDC. If not, connect to a OneFS cluster and specify MIT KDC as a OneFS authentication provider and then configure your access zone to use MIT KDC either using the OneFS web administration interface or by running the following commands through an SSH client:

```
isi auth krb5 create --realm=$REALM --admin-server=$admin_server --
kdc=$kdc_server --user=$admin_principal --password=$admin_password
```

```
isi zone zones modify --zone=$isilon_zone --add-auth-provider=krb5:$REALM
```

   Examples on the usage of the above commands are shown:

```
Cluster1-1# isi auth krb5 create --realm=EXAMPLE.COM --admin-
server=hdfs01.vlab.local --kdc=hdfs01.vlab.local --
user=admin/admin@EXAMPLE.COM --password=Password123!
```

```
Cluster1-1# isi zone zones modify --zone=zone1-hdp --add-auth-
provider=krb5:EXAMPLE.COM
```

   b. Create service principal names for hdfs and HTTP (for WebHDFS) either using the OneFS web administration interface or by running the following commands through an SSH client:

```
isi auth krb5 spn create --provider-name=$REALM --
spn=hdfs/$isilon_smartconnect@$REALM --user=$admin_principal --
password=$admin_password
isi auth krb5 spn create --provider-name=$REALM --
spn=HTTP/$isilon_smartconnect@$REALM --user=$admin_principal --
password=$admin_password
```

   Examples on the usage of the above commands are shown:

```
cluster1-1# isi auth krb5 spn create --provider-name=EXAMPLE.COM --
spn=hdfs/isilonsczone-hdp@EXAMPLE.COM --user=admin/admin@EXAMPLE.COM --
password=Password123!
```

```
cluster1-1# isi auth krb5 spn create --provider-name=EXAMPLE.COM --
spn=HTTP/isilonsczone-hdp@EXAMPLE.COM --user=admin/admin@EXAMPLE.COM --
password=Password123!
```

   c. Run the `isi zone zones list` command. A sample output of the command is shown:

```
Cluster1-1# isi zone zones list --verbose
                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-hdp
                Path: /ifs/data/zone1/hdp
           Groupnet: groupnet0
       Map Untrusted: -
     Auth Providers: lsa-local-provider:zone1-hdp, lsa-krb5-
provider:EXAMPLE.COM
        NetBIOS Name: -
  User Mapping Rules: hdfs=>root
Home Directory Umask: 0077
  Skeleton Directory: /usr/share/skel
  Cache Entry Expiry: 4H
             Zone ID: 2
```

d. List the service principal names (SPNs) that you created in the previous step by running the following command through an SSH client:

```
isi auth krb5 spn list --provider-name=$REALM
```

An example on the usage of the above command and the sample output is shown:

```
isi auth krb5 spn list --provider-name=EXAMPLE.COM
SPN                                 Kvno
--------------------------------------
HTTP/isilonsczone-hdp@EXAMPLE.COM 2
hdfs/isilonsczone-hdp@EXAMPLE.COM 2
--------------------------------------
Total: 2
Note that this Kerberos realm has SPNs and keys managed manually.
```

e. Disable simple authentication either using the OneFS web administration interface or by running the following command through an SSH client. This action also ensures that WebHDFS uses only Kerberos for authentication:

```
isi hdfs settings modify --zone=$isilon_zone --authentication-
mode=kerberos_only
```

10. Return to the **Enable Kerberos Wizard** and click **Next** on the **Kerberize Cluster** screen.

    Ambari attempts to restart services. If some of the services fail to start, you can always restart them. Review the startup logs of the service and monitor the `isi_for_array tail -f /var/log/hdfs.log` file when the services are starting to review the progress.

Note: You can overwrite the Spark2 properties for Kerberos. For directions, see "Scenario #5, Environment 5.1: MIT KDC environment" section in *Isilon: Spark2 fails to start after Kerberization with HDP 3 and OneFS due to missing configurations*, article 534442.

Enable Ambari-automated Kerberos with OneFS on Hortonworks using Active Directory authentication

Configure OneFS to respond to requests for secure Kerberized HDFS, authenticated by Active Directory as described in this section.

- The OneFS cluster must be joined correctly to the target Active Directory. Configure the following advanced settings. In the following steps, the settings are defined using the OneFS Web Administration interface. These settings maintain user and identity mappings between users running Hadoop jobs and the OneFS cluster, and also enable a standard OneFS permission model:

    a. Click **Access** › **Authentication Providers** › **Active Directory**.

    b. In the **Active Directory Providers** table, click **View details** for the provider whose settings you want to modify.

    c. 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.
- Add the mapping rules. See step 15 in the Enable Ambari-automated Kerberos with OneFS using Active Directory section for details.
- All the IP addresses within the required SmartConnect Zone must be added to the reverse DNS with the same FQDN for the cluster delegation.
- Users running Hadoop jobs must have Active Directory user principals with UNIX attributes allocated. OneFS leverages the Active Directory Schema extension that supports UNIX identities. These schema attributes extend Active Directory objects to provide UIDs and GIDs to a user account in Active Directory. Depending on your setup, your LINUX hosts might need to be integrated into AD for identity management.

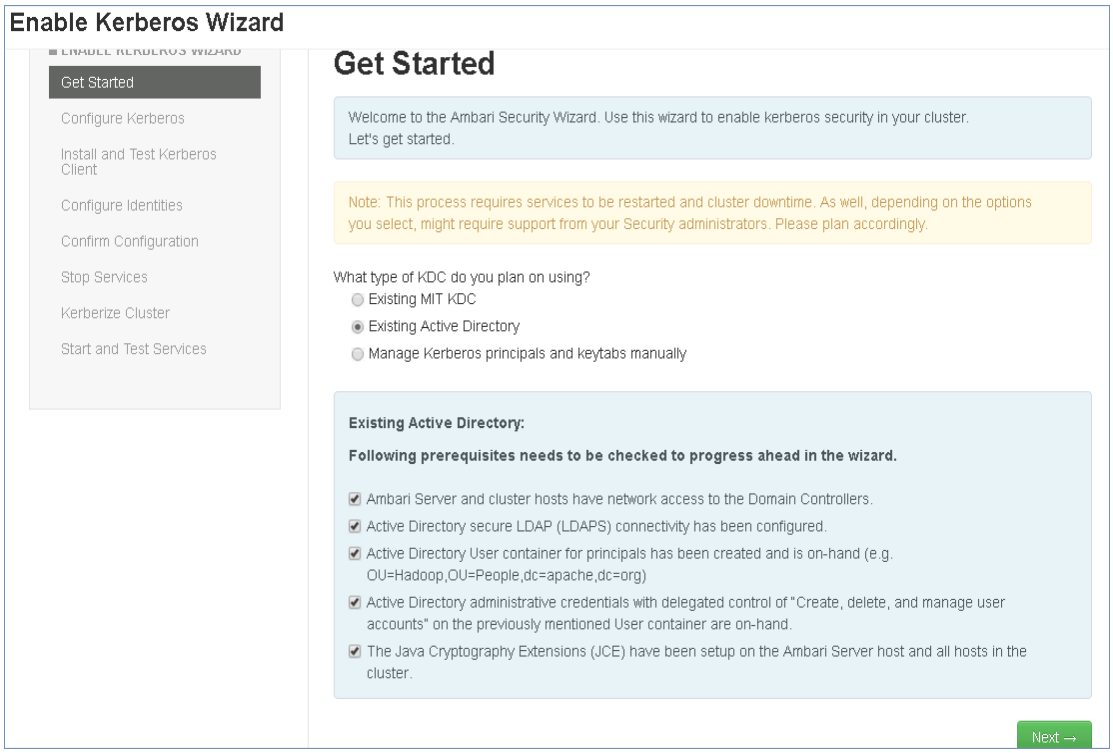
Enable Ambari-automated Kerberos with OneFS using Active Directory

This procedure outlines the steps to enable Ambari-automated Kerberos on your OneFS cluster using Active Directory. It is recommended that you stop all the user activities on the Hadoop cluster before performing a Kerberization task. If some of the services do not start up after you complete all the Kerberization tasks, see the troubleshooting section that provides some useful references to address your specific issue.

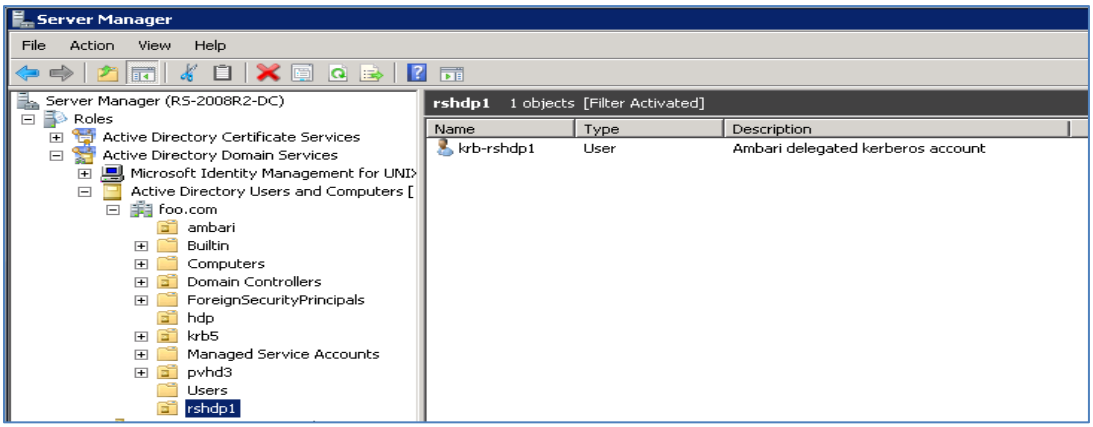
1. Log in to Ambari Web and click **Admin** › **Kerberos**.
2. Click **Enable Kerberos**.
3. On the **Get Started** screen of the Ambari Security Wizard, select the following options:

    a. Select **Existing Active Directory** as the type of KDC to use.

b. In the **Existing Active Directory** area, select all the check boxes to confirm that you have checked and completed all the prerequisites. For more information about setting up and meeting the requirements and prerequisites, see the Ambari Security Guide and Microsoft documentation for Active Directory information and configuration.



On the **Enable Kerberos** screen, specify information that is related to the KDC, the KDC Admin Account and the Service, and the Ambari principals that need to be created. The Active Directory OU and the delegated Ambari user account are shown in the following screen. This Account is used to bind to Active Directory and to create all the required Ambari principals in Active Directory as shown in the following example:



4. On the **Configure Kerberos** screen, configure the following settings:

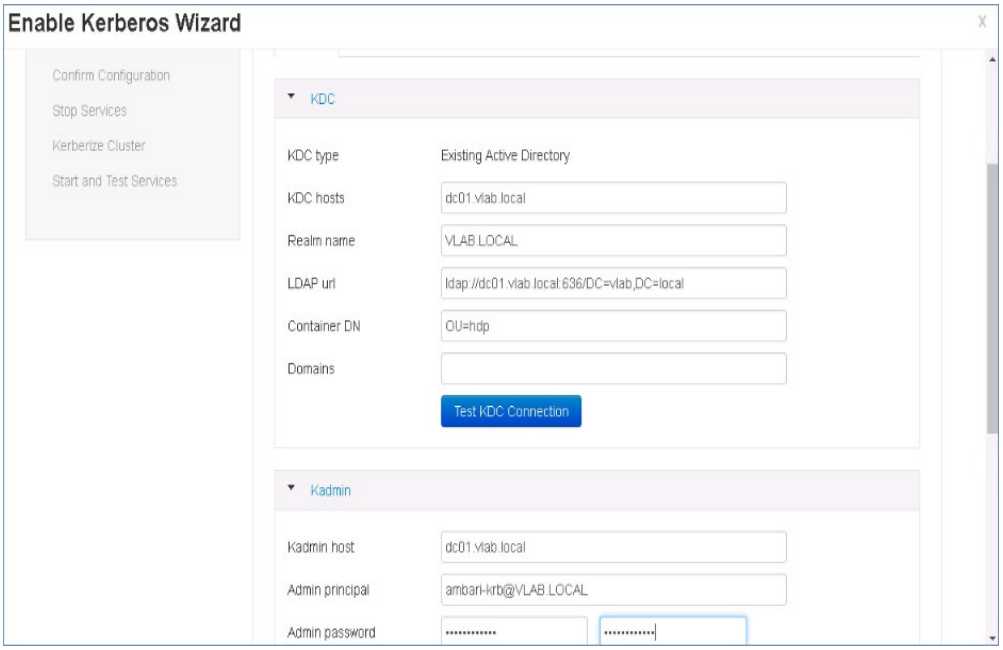
    a. **KDC Host**—Active Directory domain controller, for example, DC01.vlab.local

    b. **Realm name**—Name of the Kerberos realm that you are joining. The realm name must contain uppercase characters. For example, VLAB.LOCAL.

    c. **LDAP url**—LDAPS URL of the directory domain controller, for example, ldap://DC01.vlab.local:636/DC=vlab,DC=local

    d. **Container DN**—OU on which the delegated access was granted, for example, ou=hadoop

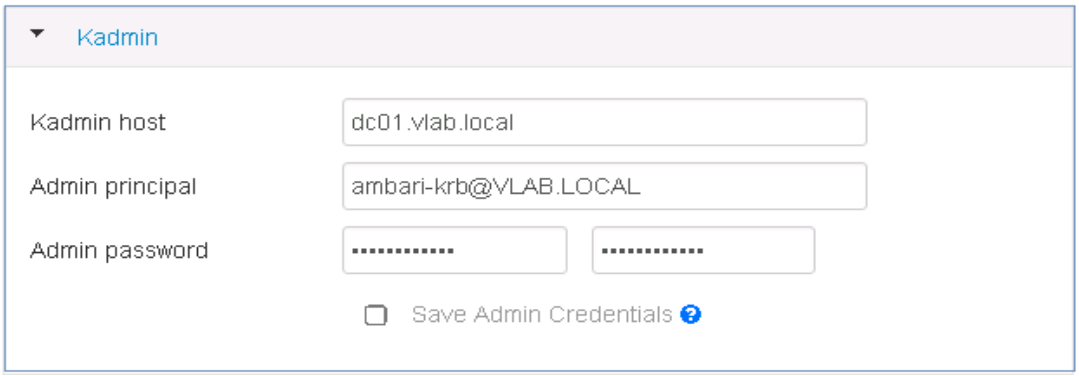
    e. **Domains**—(optional) Comma-separated list of domain names to map server host names to realm names.



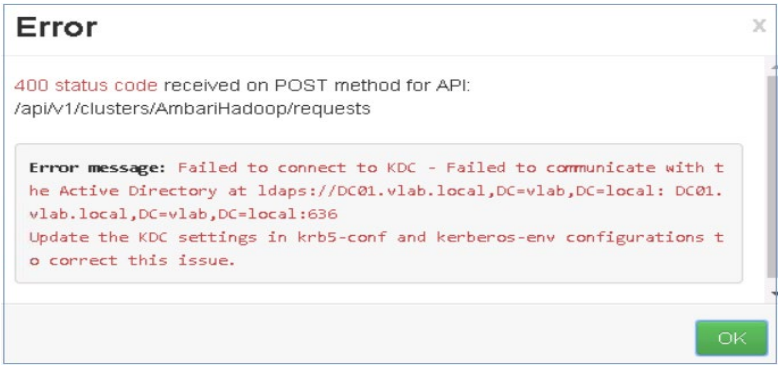
    f. **Kadmin host**—Active Directory domain controller, for example, DC01.vlab.local

    g. **Admin principal**—Active Directory user account with delegated rights, for example, krb5admin.

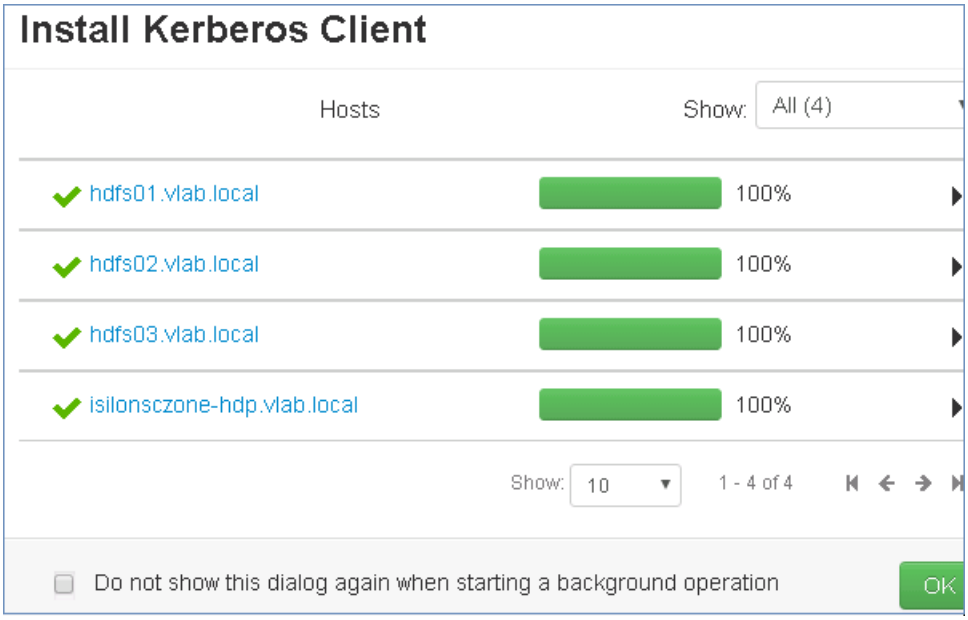
    h. **Admin password**—Password for the Admin Principal.

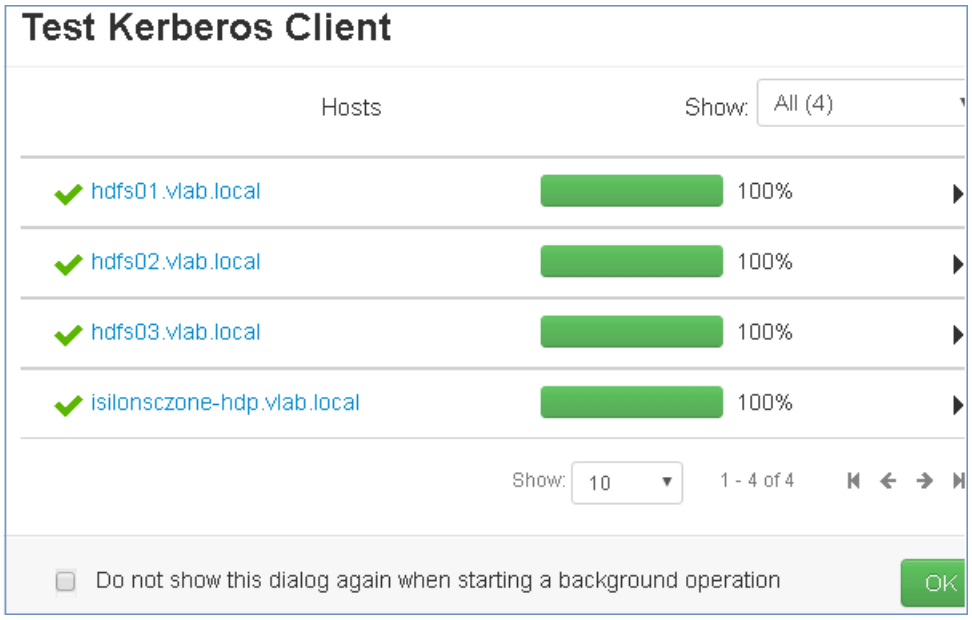


5. After specifying all the **configuration** settings, click **Next**. If clients fail to install, authentication to Active Directory or the LDAPS connection might fail. The following error message appears in such cases. Fix the error to proceed. Useful link here: https://community.hortonworks.com/content/supportkb/148572/failed-to-connect-to-kdc-make-sure-the-servers-ssl.html

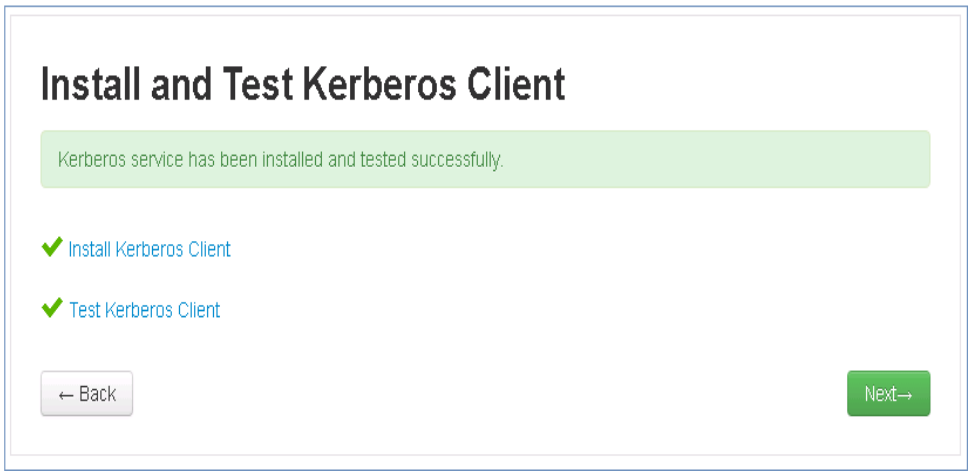


6. Using the **Install Kerberos Client** and **Test Kerberos Client** screens, deploy, configure, and test the Kerberos clients on all the hosts. Even though the Kerberos client and configuration is not pushed to a OneFS cluster now, a successful installation should be reported.



7. 

Ensure that a successful test and deployment of the Kerberos clients is reported.



8. Click **Next** on the **Install and Test Kerberos** screen.
9. On the **Configure Identities** screen, take the following actions:
    a. Click the **General** tab and configure the Apache Ambari user principals as shown in the next table. Remove `-${cluster-name}` from the default value and change to a value as shown in the **Required value** column so that it matches the service account names (users) that you created during the initial configuration of your OneFS cluster for use with Ambari and Hortonworks.

Note: Ambari creates user principals in the format `${username}-${clustername}@${realm}` and then uses the parameter `hadoop.security.auth_to_local in core-site.xml` file to map the principals to `${username}` on the file system. OneFS does not recognize these mapping rules, and therefore, you must perform this step.

- For versions of **Apache Ambari prior to Ambari 2.5** specify the following mapping rules:

Ambari user principal	Default value	Required value
Smoke user Principal	`${cluster-env/smokeuser}-${cluster_name|toLower()}@${realm}`	`${cluster-env/smokeuser}@${realm}`
HDFS user principal	`${hadoop-env/hdfs_user}-${cluster_name|toLower()}@${realm}`	`${hadoop-env/hdfs_user}@${realm}`
spark.history.kerberos.principal	`${spark-env/spark_user}-${cluster_name}@${realm}`	`${spark-env/spark_user}@${realm}`
Accumulo user principal	`${accumulo-env/accumulo_user}-${cluster_name|toLower()}@${realm}`	`${accumulo-env/accumulo_user}@${realm}`
trace.user	`tracer-${cluster_name\|toLower()}@${realm}`	`${accumulo-env/accumulo_user}@${realm}` **Note**: You must change `tracer@{realm}` to `${accumulo-env/accumulo_user}@${realm}` because there is no `tracer` user on OneFS prior to installing and configuring Kerberos. Therefore, we recommend that you change it to match the Accumulo principal. Refer to the OneFS, Ambari, and Accumulo Tracer ECN blog post for more information.
HBase user principal	`${hbase-env/hbase_user}-${cluster_name|toLower()}@${realm}`	`${hbase-env/hbase_user}@${realm}`
Storm user principal	`${storm-env/storm_user}-${cluster_name|toLower()}@${realm}`	`${storm-env/storm_user}-@${realm}`

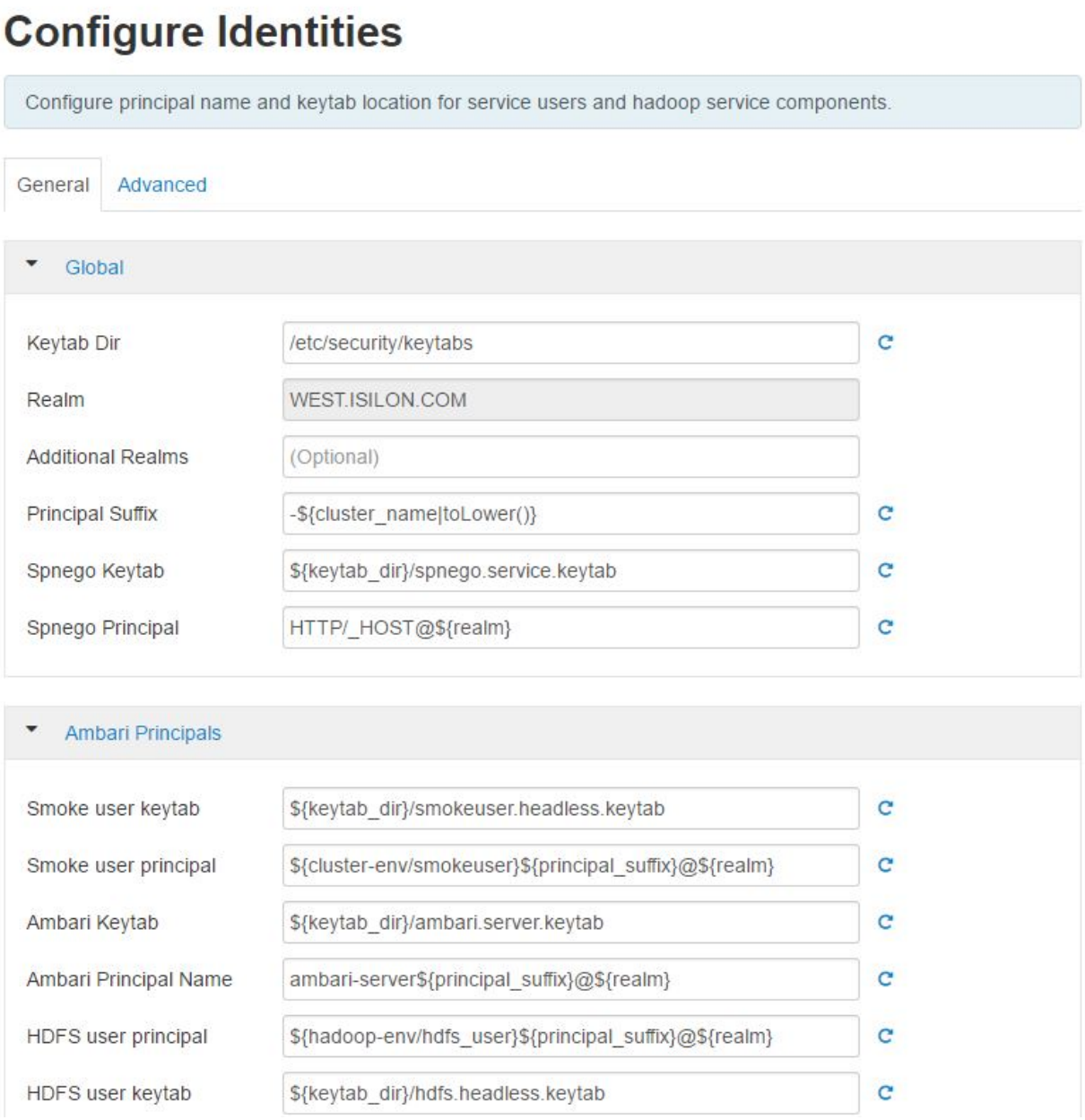
- In **Apache Ambari 2.5**, a new common property named *Principal Suffix* was added that enables Ambari administrators to easily make the same change to many principals' user names. It references the HDP cluster name through the `${cluster_name}` property by default, and each principal includes it with the `${principal_suffix}` property. Therefore, depending on what the service account names are, you may need to remove the `cluster_name` and `cluster_name|toLower()` parameters from the Principal Suffix field but not from the Ambari principals, as only the user name needs to be updated in the Ambari principals fields. For example:

- If the service account names are simply “yarn”, “hdfs”, “druid”, then delete the contents of the Principal Suffix field.
- If the cluster name is “rhubarb” and the service account names are “yarn-rhubarb”, “hdfs-rhubarb”, “druid-rhubarb”, then leave the Principal Suffix field as-is.

See the following blog post for more details regarding OneFS and Ambari 2.5/HDP 2.6.

Ambari user principal	Default value	Required value
trace.user	`tracer-${principal_suffix}@${realm}`	`${accumulo-env/accumulo_user}@${realm}` Note: You must change `tracer@{realm}` to `${accumulo-env/accumulo_user}@${realm}` because there is no `tracer` user on OneFS prior to installing and configuring Kerberos. Therefore, we recommend that you change it to match the Accumulo principal. Refer to the OneFS, Ambari, and Accumulo Tracer ECN blog post for more information.

Ambari 2.5, the `principal_suffix` parameter has been added to the Ambari principal fields as shown in the following screen. With Ambari 2.5, you may need to modify each principal to remove `${principal_suffix}` if you do not want it.

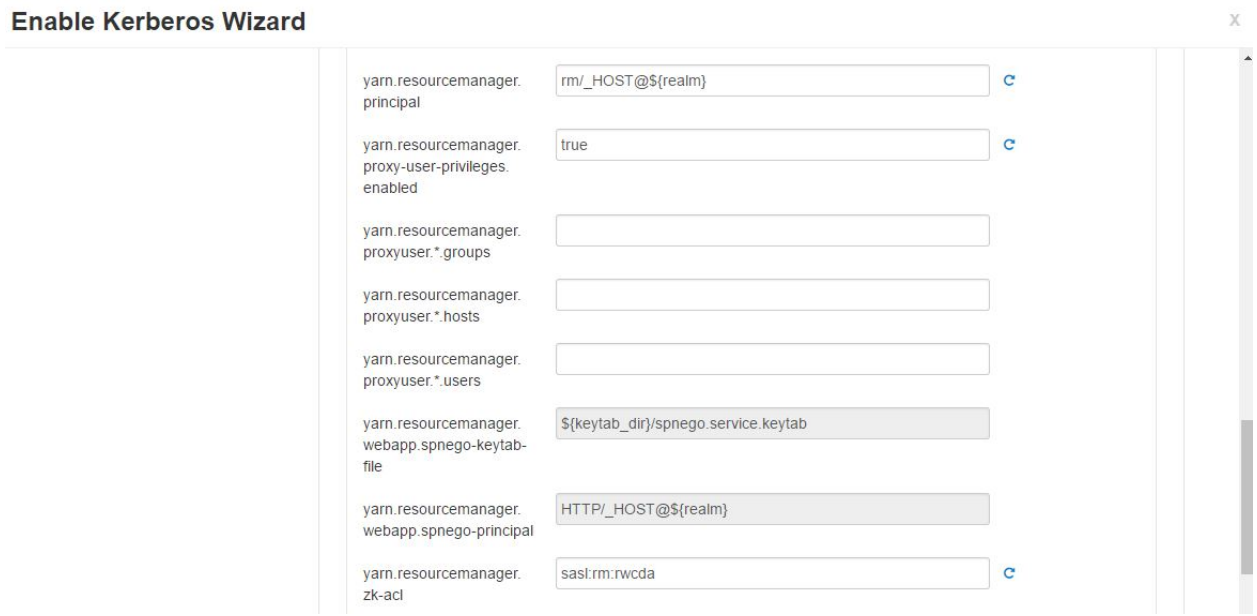


Also note that prior to Ambari 2.5, the yarn.resourcemanager.zk-acl setting must **not** be created if it does not exist. In 2.5, Ambari now configures an ACL-controlled ZooKeeper table to store the state of Yarn Resource Manager for High Availability. It records which host is the current active resource manager. Only the resource manager process has read or write access to this table. This setting is controlled with a user name in the `sasl:rm:rwcda` string value. The middle value between the colons (:) is the user name, which must be changed to match the yarn.resourcemanager.principal root when integrating with OneFS, as shown in the following table.

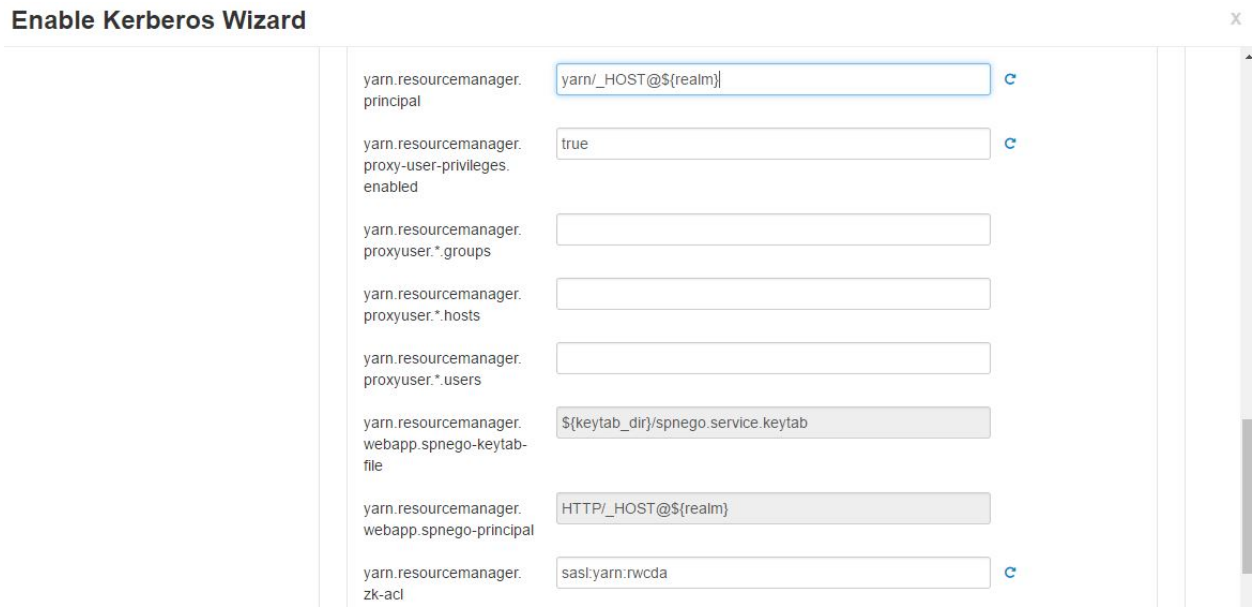
Note: This must be set properly before completing the Kerberos wizard. If not, the resource manager principal will not be able to access the state store. It will traceback and stop immediately after starting and there is no documented work-around for this. Yarn may need to be reinstalled. A change to an upcoming version of Ambari should include a usability change for zk-acl to avoid this.

Hadoop service component	Ambari setting	Default value	Required value
YARN	yarn.resourcemanager.zk-acl	`sasl:rm:rwcda`	`sasl:yarn:rwcda` **Note:** The user name between the colons must match the first principal component.

The default settings for the yarn principal, which you can find under the **Advanced** tab › Yarn drop down in Ambari 2.5, are as shown:



In this example, the yarn service account name is “yarn”, so change the yarn.resource.manager and yarn.resourcemanager.zk-acl settings as shown, where the middle value, *yarn*, matches the root of the top property, *yarn.resourcemanager.principal.*



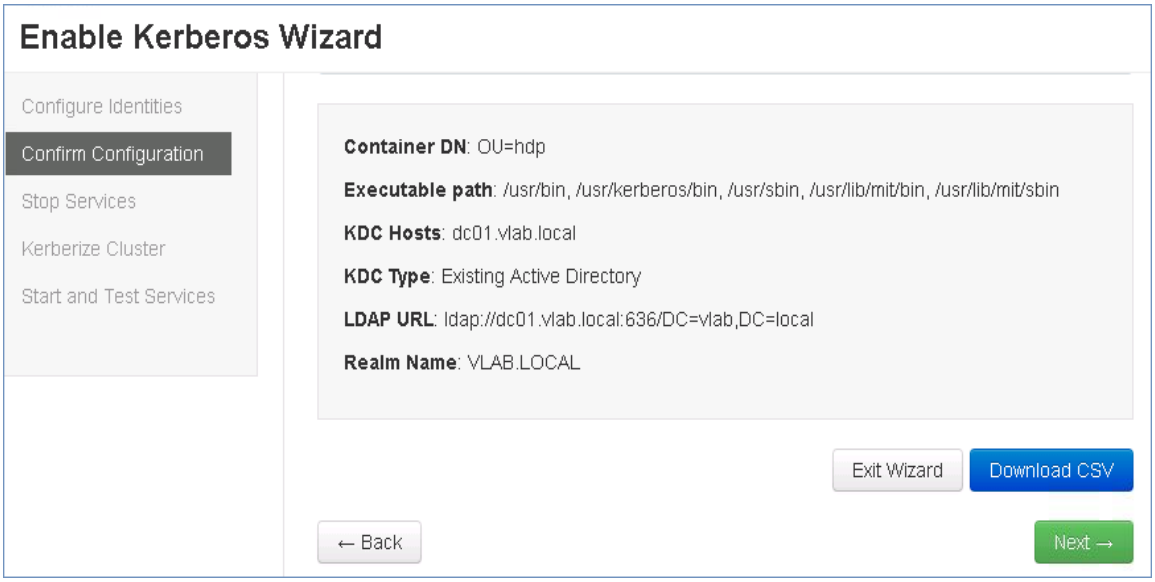
a. Click the **Advanced** tab and configure the principal name for service users and Hadoop service components so that the principal name matches the user names that are recognized by OneFS. Since Apache Hadoop has principal names per component that are mapped to per-service users or service accounts, you must change all of the principal names to match their service accounts. See the following table for details:

Note: If you are running OneFS versions 8.0.0.0 through 8.0.0.3 or OneFS 8.0.1.0, set `yarn.resourcemanager.principal` to `yarn/$rm_hostname@$REALM` by clicking **YARN** › **Custom yarn-site**. The **_HOST** syntax does not work if you have enabled Kerberos.

Hadoop service component	Ambari configuration property names	Default value	Required value
HDFS	dfs.secondary.namenode.kerberos.principal	nn/_HOST@${realm}	hdfs/_HOST@${realm}
HDFS	dfs.secondary.namenode.keytab.file	${keytab_dir}/nn.service.keytab	${keytab_dir}/hdfs.service.keytab
HDFS	dfs.datanode.kerberos.principal	dn/_HOST@${realm}	hdfs/_HOST@${realm}
HDFS	dfs.datanode.keytab.file	${keytab_dir}/dn.service.keytab	${keytab_dir}/hdfs.service.keytab
HDFS	dfs.namenode.kerberos.principal	nn/_HOST@${realm}	hdfs/_HOST@${realm}
HDFS	dfs.namenode.keytab.file	${keytab_dir}/nn.service.keytab	${keytab_dir}/hdfs.service.keytab

Hadoop service component	Ambari configuration property names	Default value	Required value
HDFS	dfs.secondary.namenode.kerberos.principal	nn/_HOST@${realm}	hdfs/_HOST@${realm}
YARN	yarn.nodemanager.principal	nm/_HOST@${realm}	yarn/_HOST@${realm}
YARN	yarn.nodemanager.keytab	${keytab_dir}/nm.service.keytab	${keytab_dir}/yarn.service.keytab
YARN	yarn.resourcemanager.principal	rm/_HOST@${realm}	yarn/_HOST@${realm}
YARN	yarn.resourcemanager.keytab	${keytab_dir}/rm.service.keytab	${keytab_dir}/yarn.service.keytab
MapReduce2	mapreduce.jobhistory.principal	jhs/_HOST@${realm}	mapred/_HOST@${realm}
MapReduce2	mapreduce.jobhistory.keytab	${keytab_dir}/jhs.service.keytab	${keytab_dir}/mapred.service.keytab
Falcon	*.dfs.namenode.kerberos.principal	nn/_HOST@${realm}	hdfs/_HOST@${realm}

10. Click **Next** on the **Configure Identities** screen.
11.
12. On the **Confirm Configuration** screen, review the settings that appear as shown:



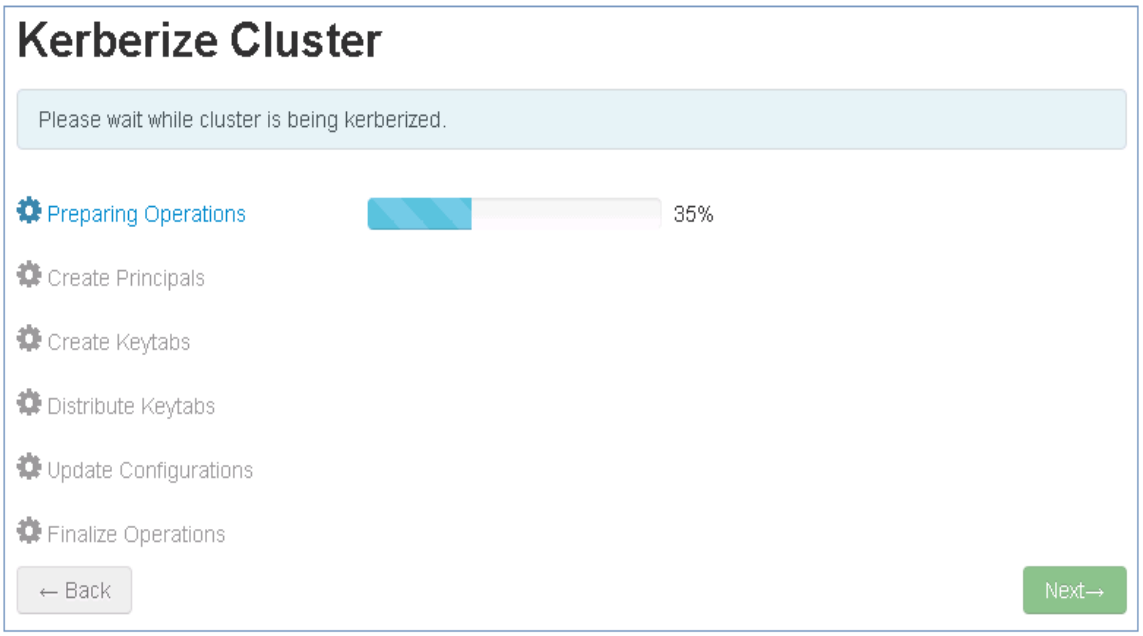
A CSV file contains all the principals and keytabs that Ambari will create in Active Directory. The file also contains principals and keytabs for OneFS but those keytabs will not be distributed to the OneFS cluster.

13. Click **Next** to accept all the configuration settings and open the **Stop Services** screen.

    All the Hadoop services in Ambari, user activities, and services on all the hosts are stopped automatically.

Note: If you exit the Ambari Security Wizard at this stage, all the configurations and customizations that you have specified will be lost and you will have to restart the configuration process again.

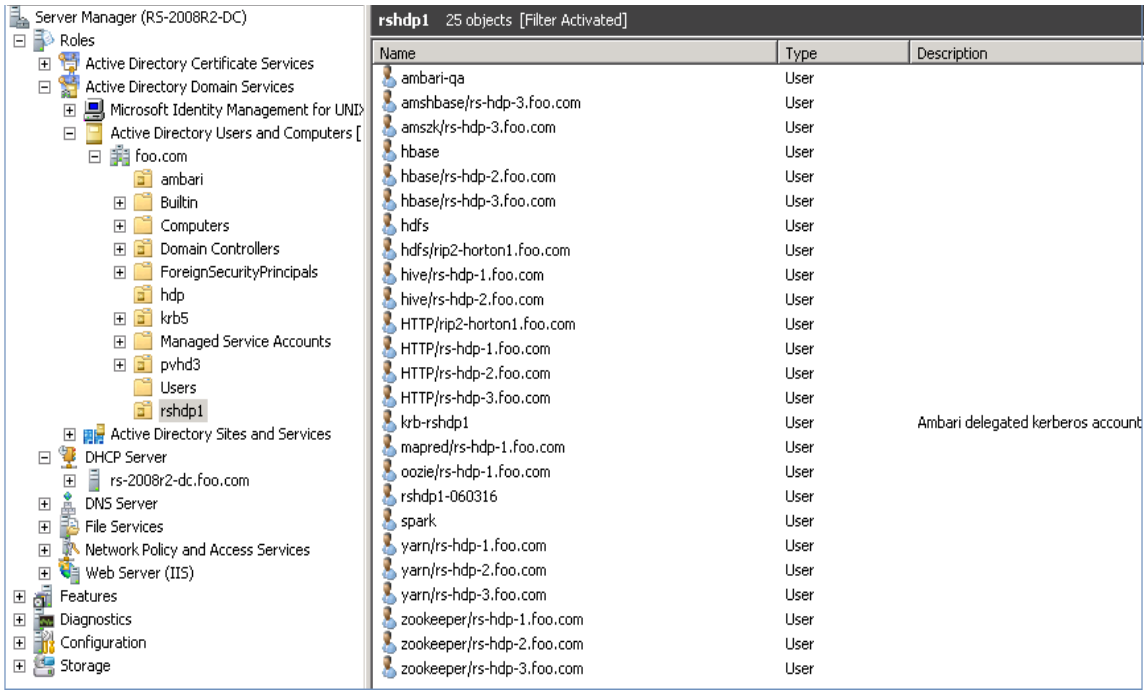
14. Click **Next** when all the services are stopped successfully to move to the **Kerberize Cluster** screen. The Kerberization process is automatically initialized. The Ambari services are Kerberized, user principals, service principals are created, and keytabs are distributed.3

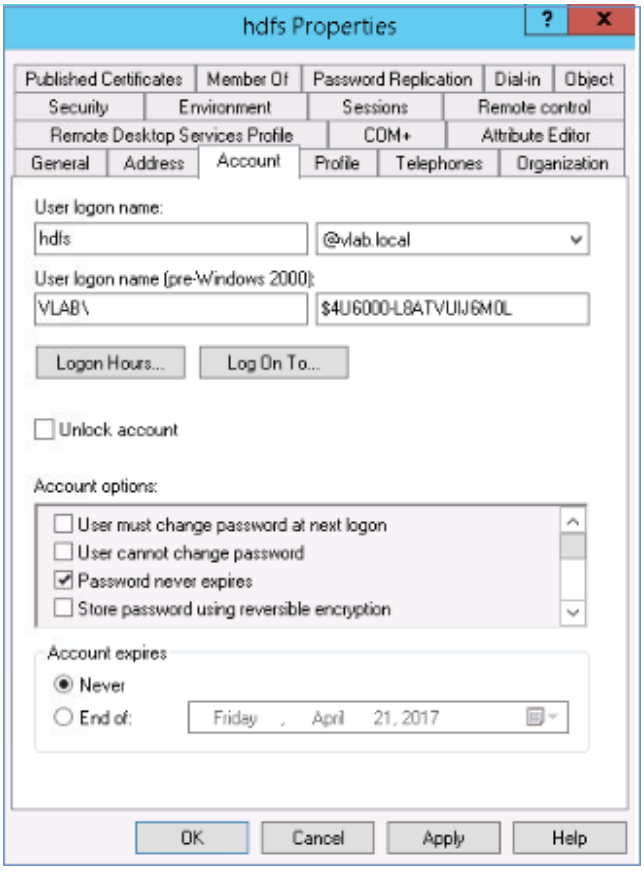


Warning

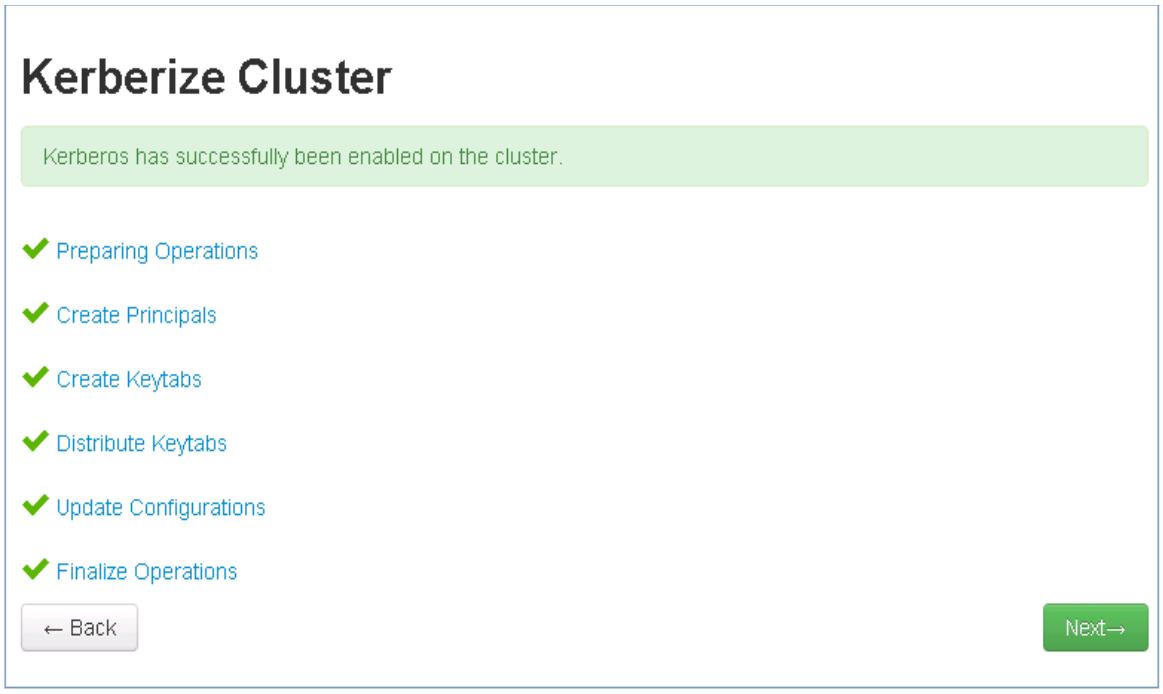
Do not click **Next** on the **Kerberize Cluster** screen until you have completed steps 13 through 17 successfully.

After the principals are created, you can view all the Active Directory principals that you designated in the installation wizard in the Hadoop OU screen as shown:





When Kerberos is successfully enabled on the cluster, the following screen appears:

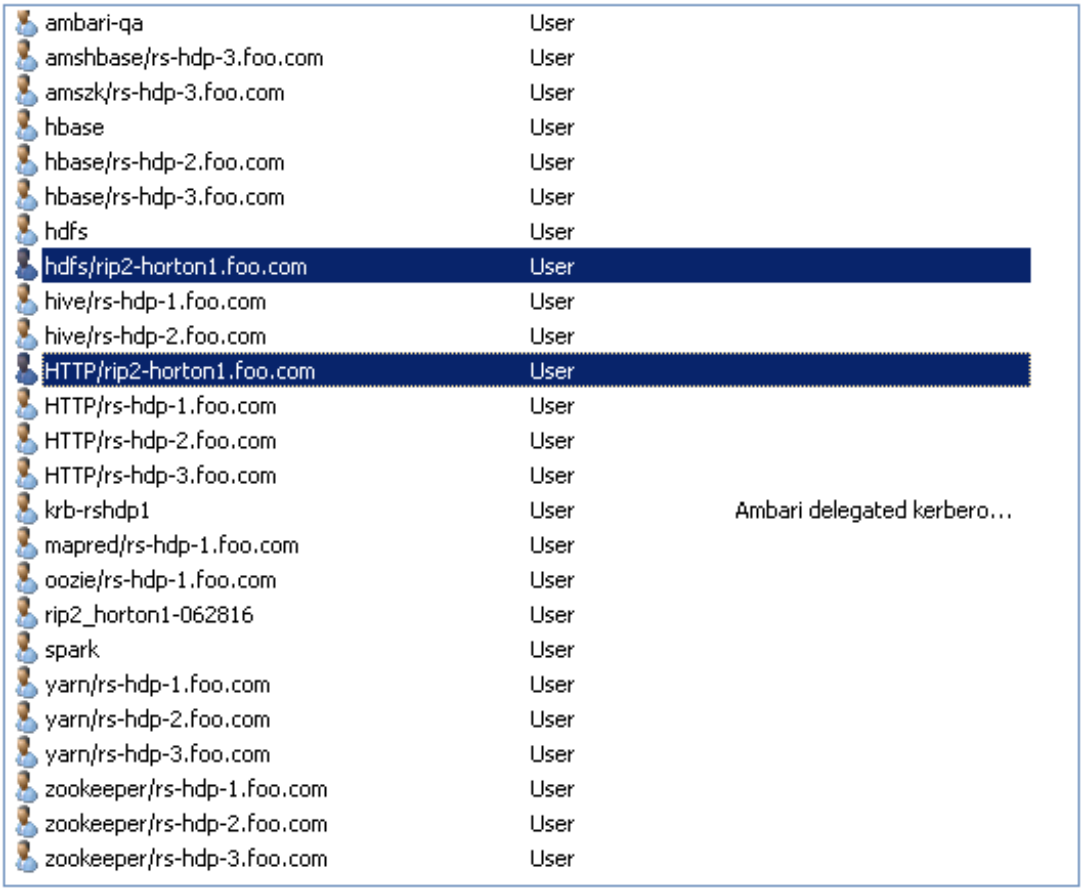


With Kerberization enabled, you can now prepare the OneFS cluster and Active Directory for the services to start.

Warning

Do not click **Next** on the **Kerberize Cluster** screen until you have completed steps 13 through 17 successfully.

15. Go to Active Directory and remove the Ambari-generated SPNs that are specific to OneFS. Refer to this ECN blog post for information about troubleshooting issues related to duplicate SPNs for Active Directory-based Kerberos with Hadoop. Remove `hdfs/<isilon-clustername>` and `HTTP/<isilon-clustername>` from the Hadoop OU in Active Directory. A sample Hadoop OU screen with the principals is shown:



16. Connect to a OneFS cluster and add or modify the required OneFS SPNs for the SmartConnect zone name in Active Directory. Perform this step because OneFS is a clustered file system running on multiple nodes. It is however joined to Active Directory as a single Computer Object. The SPN requirements for Kerberized AD-based Hadoop access are unique with OneFS. The SPNs for Hadoop access require additional SPNs for the Clustername and the SmartConnectZone that is configured for the HDFS NameNode access as shown in the next table:

SPN	Name	Role
hdfs/clustername.fdqn	Cluster name that is joined to Active Directory	HDFS service authentication to Active Directory
hdfs/smartconnectname.fdqn	The SmartConnect zone name FQDN that Ambari is connected to	HDFS service authentication to access zone
HTTP/smartconnectname.fdqn	The SmartConnect zone name FQDN that Ambari is connected to	WebHDFS service authentication to Active

SPN	Name	Role
		Directory per SmartConnect zone

Perform the following steps:

a. Review the registered SPNs on the OneFS cluster as shown in the following example:

```
isi auth ads spn list --provider-name=vlab.local
```

The output is similar to the following:

```
Cluster1-1# isi auth ads spn list --provider-name=vlab.local
SPN
------------------------------
hdfs/isilonsczone-hdp.vlab.local
nfs/isilonsczone-hdp.vlab.local
HOST/isilonsczone-hdp.vlab.local
HOST/isilonsczone-hdp
hdfs/cluster1.vlab.local
nfs/cluster1
HOST/cluster1
HOST/cluster1.vlab.local
------------------------------
Total: 8
```

b. Add or modify the required OneFS SPNs for the SmartConnect zone name in Active Directory. Sample commands for creating OneFS SPNs for kerberized HDFS are shown:

```
isi auth ads spn create --spn=hdfs/isilonsczone-hdp.vlab.local --
provider-name=vlab.local --user=administrator
```

```
isi auth ads spn create --spn=HTTP/isilonsczone-hdp.vlab.local --
provider-name=vlab.local --user=administrator
```

See OneFS SPN requirements for Kerberized Hadoop for specific OneFS SPN requirements. Also see the following documents for additional guidance and for troubleshooting:

- Duplicate SPN's with Isilon AD Kerberos and Hortonworks prevent services from starting for information about OneFS considerations for Active Directory-based Kerberos with Hadoop
- OneFS CLI Administration Guide for information regarding adding or modifying OneFS SPNs in Active Directory

17. **Review that the three required SPNs on the OneFS cluster are now present.**

18. Disable simple authentication either using the OneFS web administration interface or by running the following command. This action also ensures that WebHDFS uses only Kerberos for authentication:

```
isi hdfs settings modify --zone=$isilon_zone --authentication-
mode=kerberos_only
```

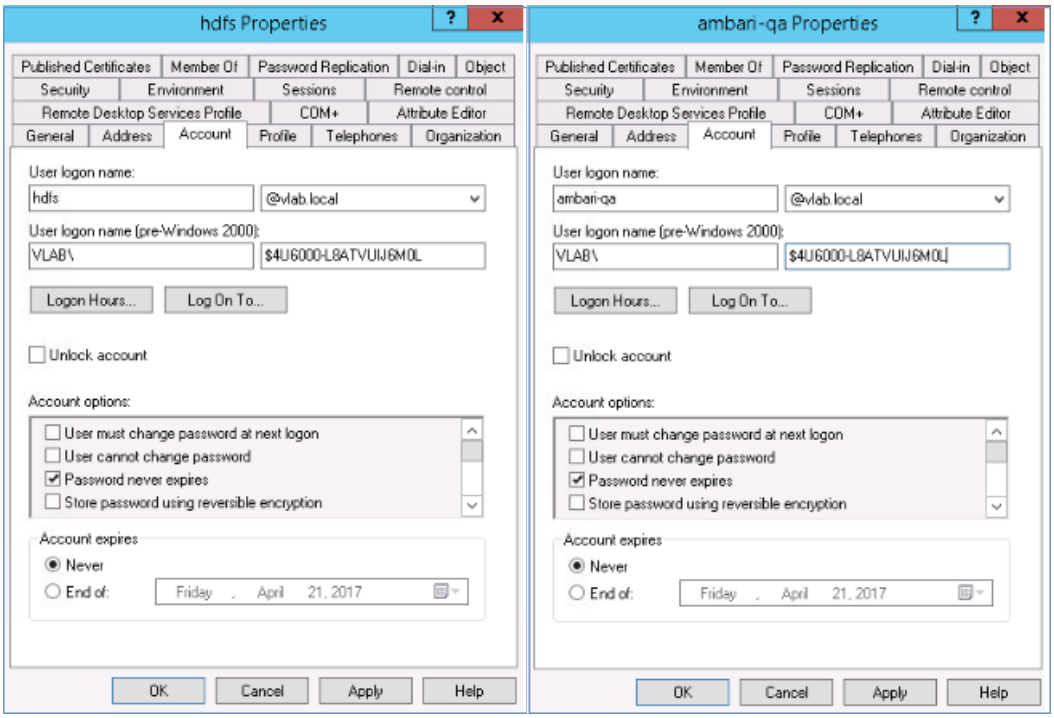
A sample command is shown:

```
isi hdfs settings modify --zone=zone1-hdp --authentication-mode=kerberos_only
```

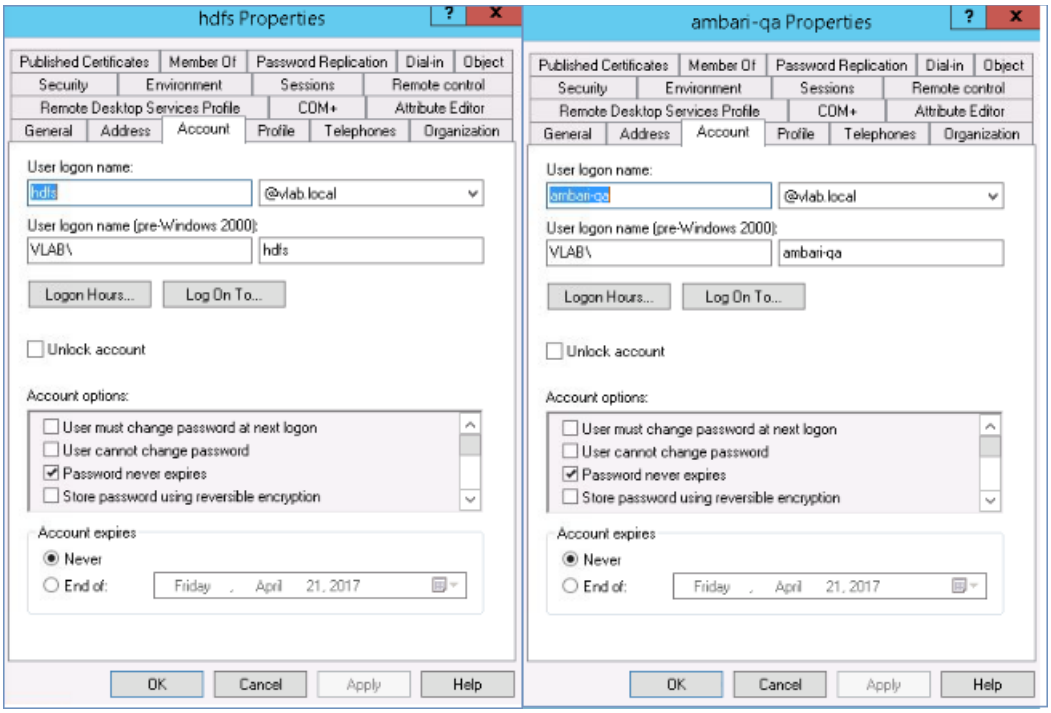
19. Perform the following OneFS Active Directory ID-mapping fixes. See the Hortonworks - Isilon Kerberization First Time Services Start article for more information.

    a. Fix the SAMAccount names for hdfs and ambari-qa in Active Directory if required, this is not needed if Step 7: the **Account Attribute Template** field, add the following lines below the `"servicePrincipalName": "$principal_name",` line was followed it is also described in the Automating the Creation of Valid SAMAccount Names with Ambari for Isilon ECN blog post.

    The SAMAccount name initially appears for hdfs and ambari-qa as shown in the following screens:



    Change the SAMAccount names for hdfs and ambari-qa as shown in the following screens:



b. Add mapping rules to the OneFS access zone to map the local OneFS service accounts to the UPNs as described in the Isilon and Hadoop Multitenant Installation and Integration Guide. If Multitenancy installation was followed.

```
isi zone zones modify --add-user-mapping-rules="domain\ambari-qa-
<clsname> => ambari-qa[]" --zone=<zone-name>

isi zone zones modify --add-user-mapping-rules="domain\ambari-server-
<clsname> => ambari-server[]" --zone=<zone-name>
```

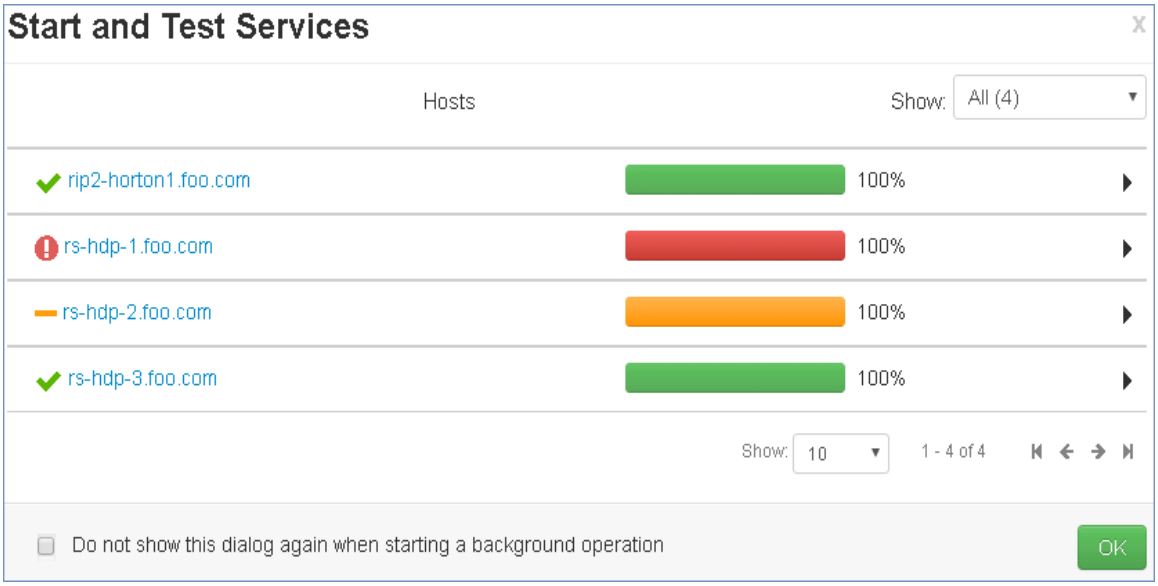
c. If multitenancy was followed, add the AD domain hdfs-clustername user to the RBAC role created earlier in this document.

20. Validate the permissions on the krb5.conf file as shown in the following example:

```
isi_for_array ls -le /etc/krb5.conf
```

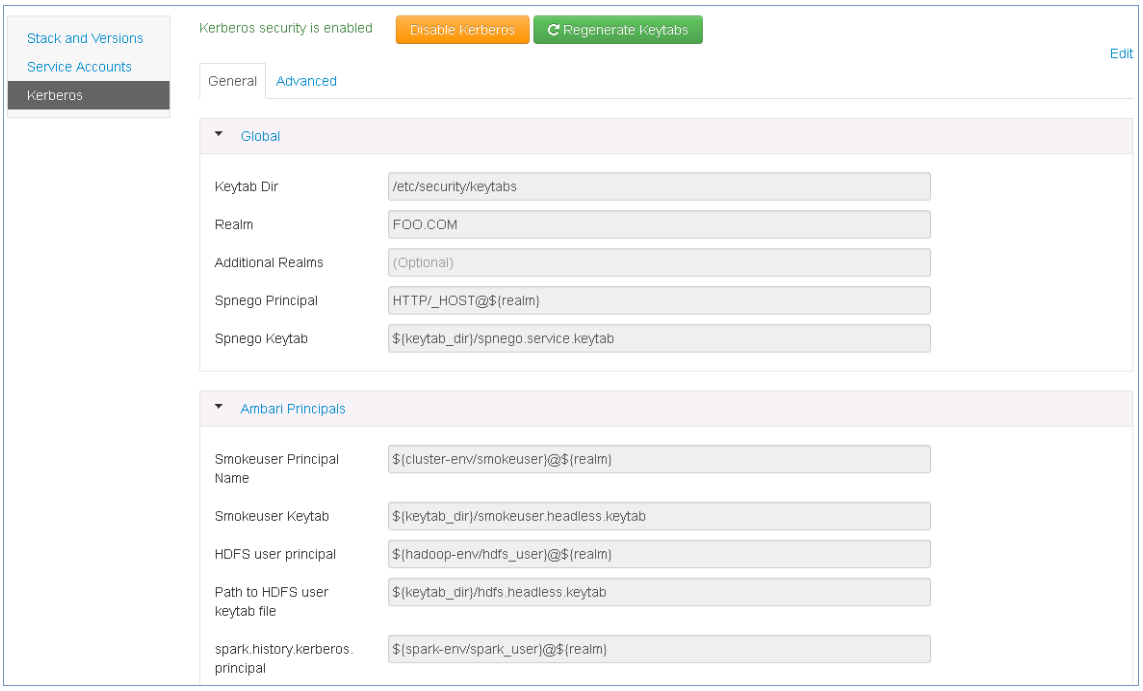
21. Return to the **Enable Kerberos Wizard** and click **Next** on the **Kerberize Cluster** screen.

    Ambari attempts to restart services. If some of the services fail to start, you can always restart them. Review the startup logs of the service and monitor the `isi_for_array tail -f /var/log/hdfs.log` file when the services are starting to review the progress. A sample screen reporting a failure is shown:

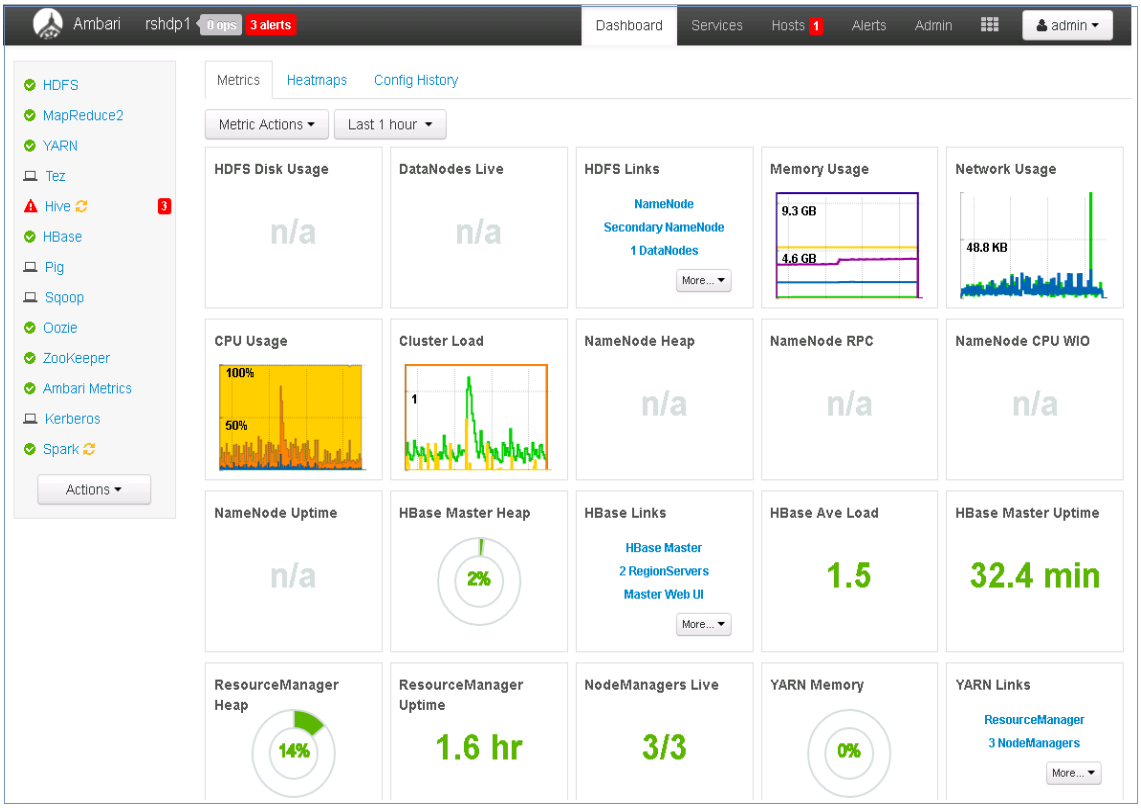


Troubleshoot each of the failed services individually.

When all the steps on the **Kerberos Cluster** wizard are completed, you can view the configuration in the Ambari Web interface.

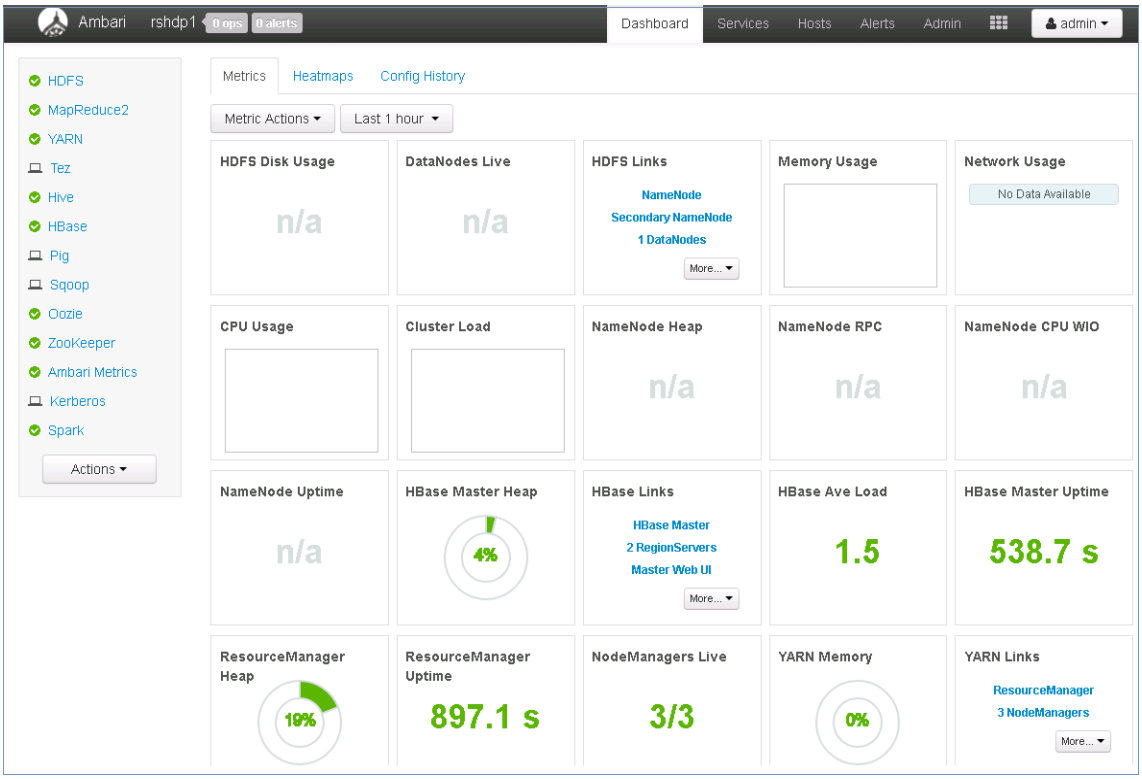


If some of the services must be restarted, the Ambari dashboard reports the failed services as shown:



The sequence of services that must appear green on the Ambari dashboard are HDFS, YARN (Apptimeline Service), MapReduce2, Zookeeper, Spark, HBase, Hive, followed by all of the remaining services.

Upon restarting the failed services, the cluster and all the HDFS services start running and the Ambari dashboard appears as shown:

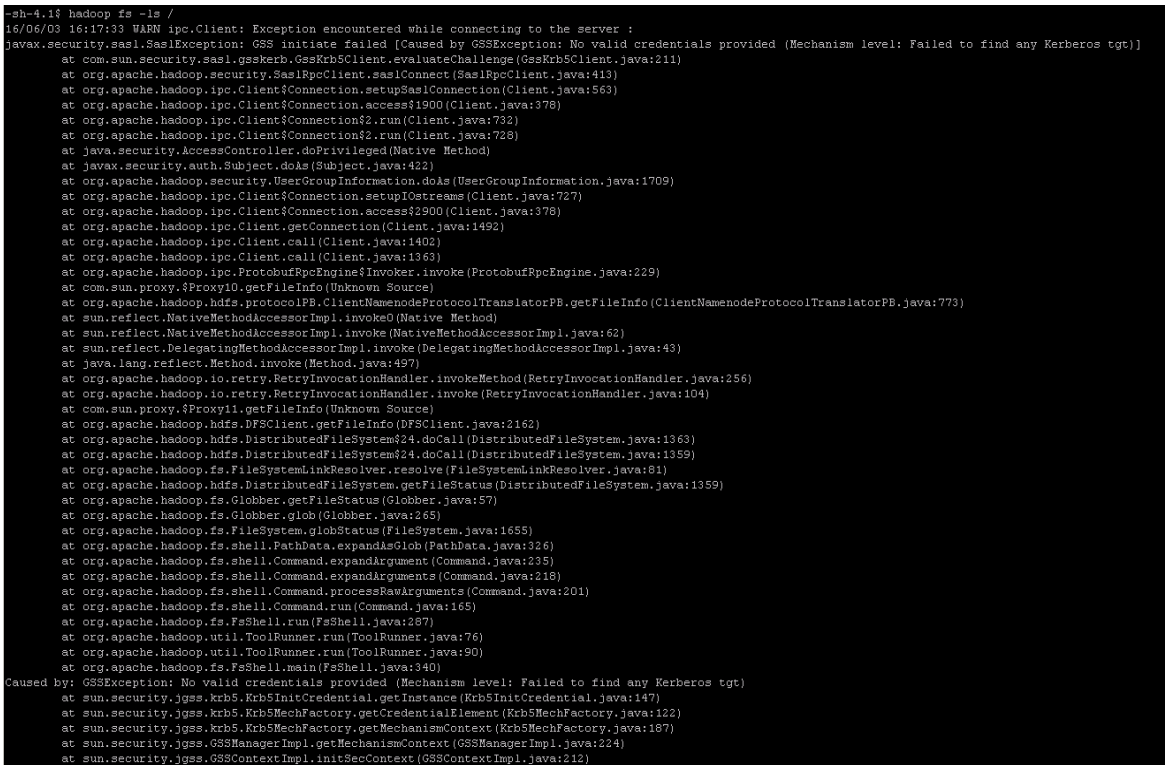


Note: You can overwrite the Spark2 properties for Kerberos. For directions, see "Scenario #5, Environment 5.2: AD environment" section in *Isilon: Spark2 fails to start after Kerberization with HDP 3 and OneFS due to missing configurations*, article 534442.

Test and validate Hadoop services

In order to validate the newly Kerberized cluster that is configured using Active Directory, run simple Hadoop commands and ensure that they fail since you do not have a valid Kerberos ticket. This is because the cluster is Kerberized and OneFS enforces Kerberos-only access to the HDFS root. This test also validates that simple authentication is not supported.

Sample output is as follows:



Addendum

This section discusses some of the security, management, and troubleshooting features that are available in an installation of OneFS with Hadoop.

Apache Ranger authorization policy integration

Apache Ranger is a centralized management console that enables you to monitor and manage data security across the Hortonworks Hadoop distribution system. A Ranger administrator can define and apply authorization policies across Hadoop components including HDFS.

HDFS policies that are defined in Ranger are checked before the native file access control is applied. This two-layered authorization model differs in the way the standard Ranger HDFS policies are checked with Directed Attached Storage (DAS), but the model is suitable for using OneFS as a multiprotocol data lake with Hadoop. OneFS native file system ACL allows a storage administrator to correctly set up access control for multiple workloads and with multiprotocol access to the HDFS dataset. A Ranger administrator can apply a further restrictive Hadoop user access control to the same HDFS dataset, thus providing the administrators the appropriate control span within their management domains.

In a OneFS cluster with Hadoop deployment, Ranger authorization policies serve as a filter before applying the native file access control.

Note:

- The Ranger Audit and Transparent Data Encryption components are not supported.
- You can enable Apache Ranger on OneFS clusters and then check for new authorization policies, receive HDFS requests from clients, and apply authorization policies to the HDFS requests which can be one of DENY, ALLOW, or UNDETERMINED.
- The Ranger DENY policy takes precedence over the ALLOW policy.
- The Ranger DENY policy prevents user or group access to files and directories in OneFS that the file system would have otherwise allowed the users or groups to access.

To understand how the Ranger policies are applied, consider an example where a user in the Sales group requires access to certain files and directories that have specific HDFS file system ownership and permissions as shown:

Scenario	Files and directories	HDFS file system ownership and permissions
A	`/ifs/data/<zone>/<hdfs-root>` (the HDFS root directory for the zone)	Read, write, and execute (rwx) permissions for everyone.
B	`/ifs/data/<zone>/<hdfs-root>/corp-confidential`	Neither the user or the Sales group have read, write, and execute (rwx) permissions.
C	`/ifs/data/<zone>/<hdfs-root>/hdfs-confidential`	The Sales group owns the folder, and the folder has read, write, and execute (rwx) permissions for the group including the user.
D	`/ifs/data/<zone>/<hdfs-root>/hdfs-not-for-user`	No read, write, and execute permissions for hidden files.

If Ranger policies are further applied, the user's access to the files and directories changes as indicated in the following table:

Ranger HDFS service instance	User's view and edit permissions per scenario			
	A	B	C	D
None	Yes	No	Yes	No
One policy that provides everyone in the Hadoop group, access to the root directory.	No	No	No	No
One policy that provides everyone including the Sales group, access to the root directory.	Yes	No	Yes	No
Two policies that are defined as follows: • Provide everyone access to the root directory. • Deny the Sales group, access to the `hdfs-confidential` directory.	Yes	No	No	No
Three policies that are defined as follows: • Provide everyone access to the root directory. • Deny the Sales group, access to the `hdfs-confidential` directory. • Deny the user, access to the `hdfs-not-for-user` directory.	Yes	No	No	No

Deploy Apache Ranger

Deploy the Apache Ranger service and enable or create a deny policy. See these documents:

- Dell EMC Isilon: Apache Ranger Setup and Operations
- Hortonworks Security Guide: Installing Apache Ranger

A summary of the workflow follows:

1. On a per-access zone basis, perform the following steps using the OneFS Web administration interface or the OneFS command-line administration interface to configure Ranger:

   a. Enable the Ranger plug-in on OneFS.

   b. Specify the URL of the Apache Ranger Management console to use port 6182 for https and 6080 for http to get the policies.

   c. Specify the name of the service instance.

   See the OneFS with HDFS Reference Guide for details.

2. Ensure that a Ranger service account user is configured on OneFS within your access zone.

3. Install Ranger using the steps outlined in the Hortonworks Security Guide.

Note: Do not assign Ranger components to the OneFS host as mentioned in step 5 of section 1.3.1 in the above-mentioned guide.

4. Enable the Apache Ranger HDFS plug-in using the steps outlined in the Hortonworks Security Guide.
5. If you have a Kerberos-enabled cluster, follow the instructions in the Hortonworks Security Guide to enable the Ranger HDFS plug-in on the cluster.
6. Enable the Ranger Deny policy using the instructions in the Apache Ranger deny policies with OneFS 8.0.1.0 article.

   Create a service instance for OneFS using the **Create Service** page in Ranger. See the Hortonworks Security Guide for details. Specify values in the following fields to create a service instance for OneFS and make note of the values:

   - Specify a value in the **Service Name** field and make note of it because you must use the same value in OneFS.
   - Specify a username and in the **Config Properties** section specific to the service instance. The **Test Connection** option continues to fail until you have saved and reopened the service instance.
   - Specify the Namenode URL as `FQDN: hdfs://onefs.smartconnect.name:8020`.

     A service instance is created with a default policy `all - path`, granting access to all the files for the user that you included in the **Service Details** page.
7. Add all your groups and individual users who are associated with an access zone within OneFS to the default policy in order to grant access to the groups and users. If you create local users in OneFS, or use Active Directory, you must change the **UserSync** settings in Ambari or add the users in the Ranger interface.

Note: OneFS file system permissions take precedence even if the policy indicates that the user or group can access everything.

8. Using **the Edit Policy** page in the Ranger interface, specify the group or users who will have limited access to the repository within OneFS and indicate the permissions that must be denied to that path.
9. Create a DENY policy in Ranger using the steps outlined in the Hortonworks Security Guide, if required. After you have saved the policy, OneFS enforces the policy at the next download. If you attempt to take action on a path that is denied access by the Ranger policy, this will be reported in the OneFS HDFS log at `/var/log/hdfs.log`. For more information, see the Apache Ranger deny policies with OneFS 8.0.1.0 article.

Ambari metrics and alerts overview

A OneFS node can monitor, collect, and push metrics data at one-minute intervals to the Ambari Metrics Collector which is one of the components of the Ambari Metrics System. Ambari Management Pack for Isilon OneFS presents OneFS access zones as a service. OneFS is identified by Ambari as a single host running the HDFS service, even though it is a clustered file system. As a result, all the metrics and alert data that is provided by OneFS to Ambari are cluster-wide. For example, for a three-node OneFS cluster, the network HDFS traffic aggregated across all the three nodes is reported to Ambari.

To use the Ambari Metrics Collector, ensure that Ambari Metrics is deployed and is running (green) on the Ambari dashboard.

Note: OneFS metrics for specific Access Zones that contain HDFS dataset is not supported.

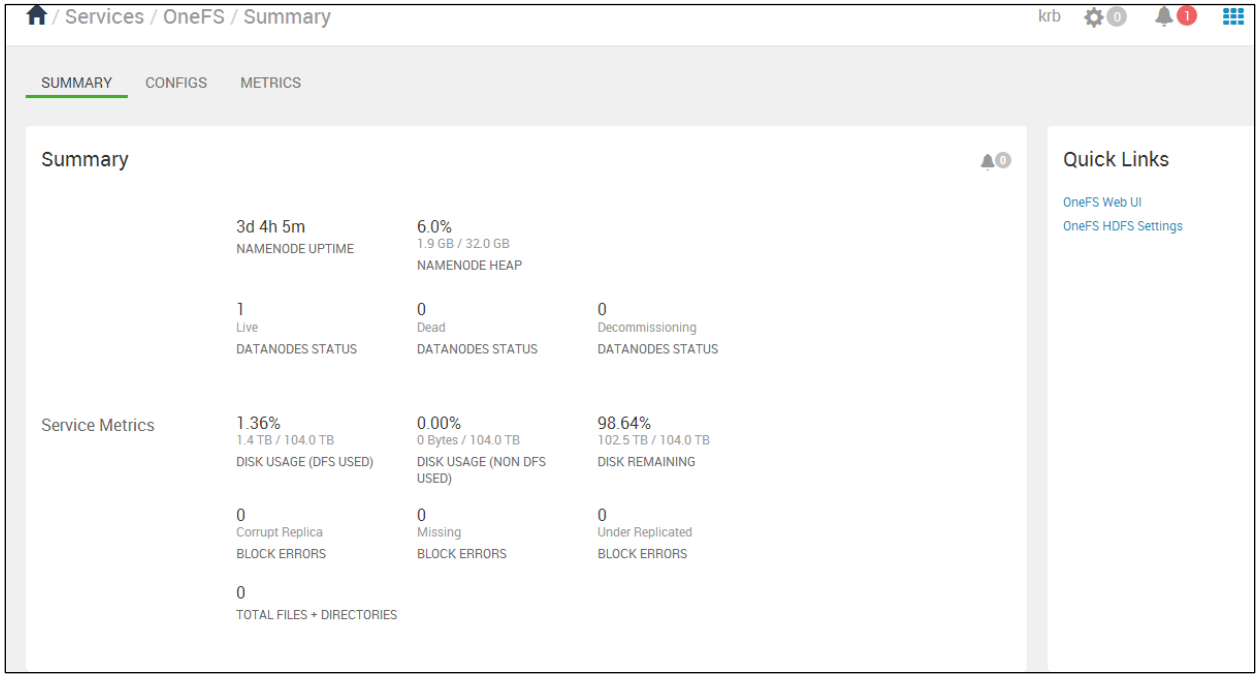
View Ambari metrics and alerts

In order to view the Ambari metrics and alerts, follow the steps that are given below:

1. Access Ambari Web by opening a supported browser and entering the Ambari Web URL.
2. Click **Ambari Metrics Service** › **Metrics Collector** to determine the hostname where Ambari Metrics Collector has been installed.
3. From a node in your OneFS cluster, run the following command to set the access zone and to specify the name of the external Ambari host where the Ambari Metrics Collector component is installed:

```
isi hdfs settings modify --zone=ZONE --ambari-metrics-collector=<FQDN of metrics collector>
```

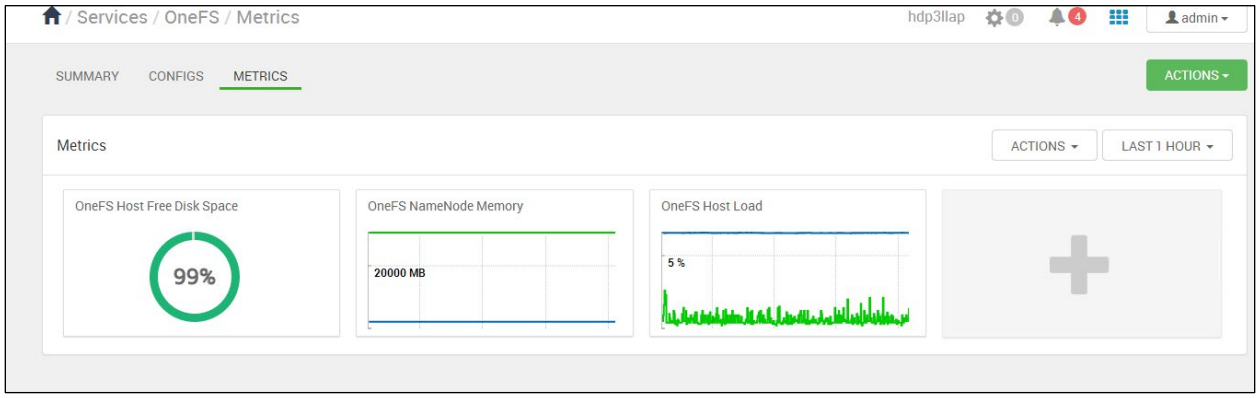
4. From the Ambari Web home page, select the **OneFS** service and verify that Ambari can collect metrics details from the OneFS SmartConnect zone as shown in the following sample screen:



The following OneFS metrics data is reported:

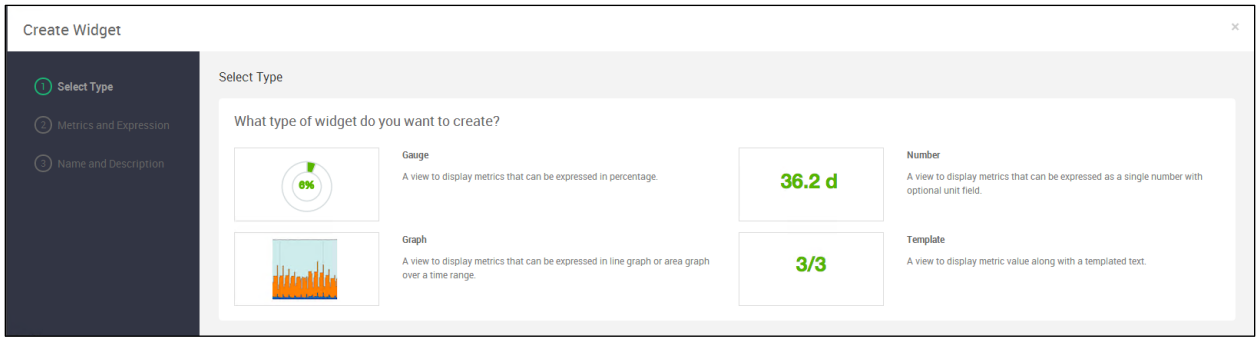
Metric	OneFS data reported
NameNode Uptime	The OneFS node that is running for the longest time.
NameNode Heap	Used: The sum of the current memory allocated by the HDFS process (cluster-wide).
Datanodes Status	OneFS node status (OneFS represents a single datanode).
Disk Usage (DFS Used)	OneFS Storage space used for distributed filesystem.
Disk Usage (Non-DFS Used)	OneFS Storage space used for non-distributed filesystem.
Disk Remaining	OneFS storage remaining.
Block Errors	Always one because there are no blocks in OneFS.
Total Files + Directories	Always zero. The file system does not maintain the file count.

5. From the Ambari Web home page, select the **OneFS** service and then click the **Metrics** tab to create widgets to monitor and view OneFS metrics data.



a. Click (+) to open the **Create Widget** wizard.

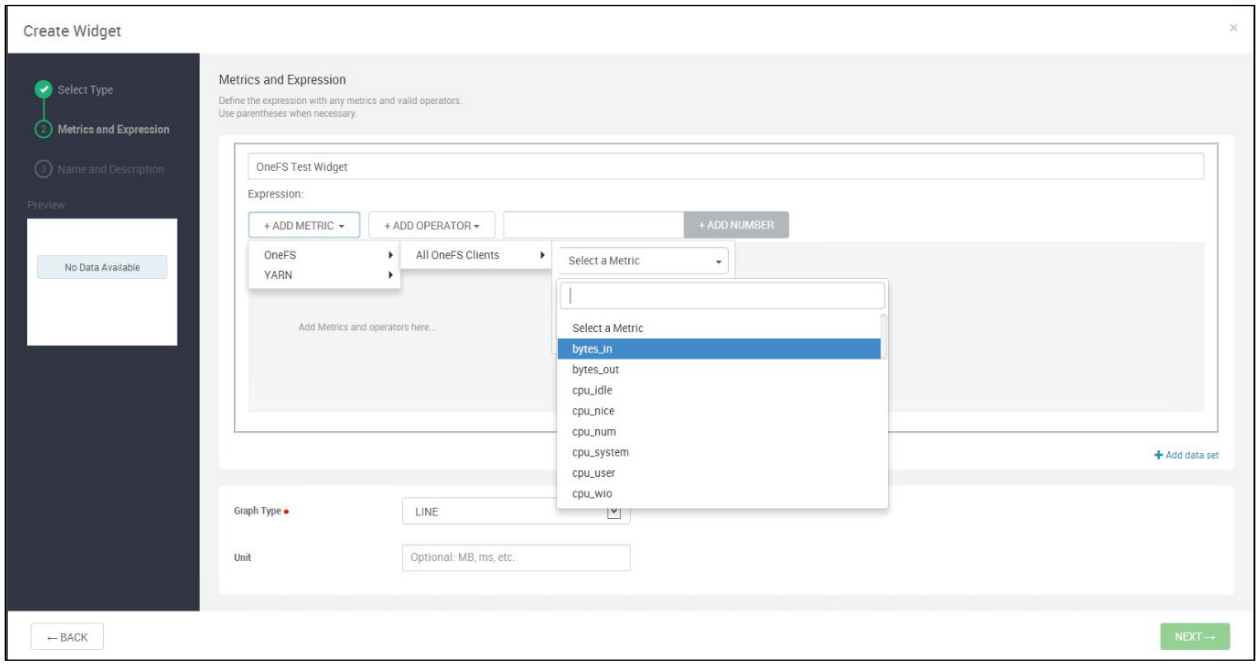
b. Select one of **Gauge**, **Number**, or **Graph** widget types. Alternatively, you can create a widget using a new template.



c. On the **Metrics and Expression** screen, perform the following steps:

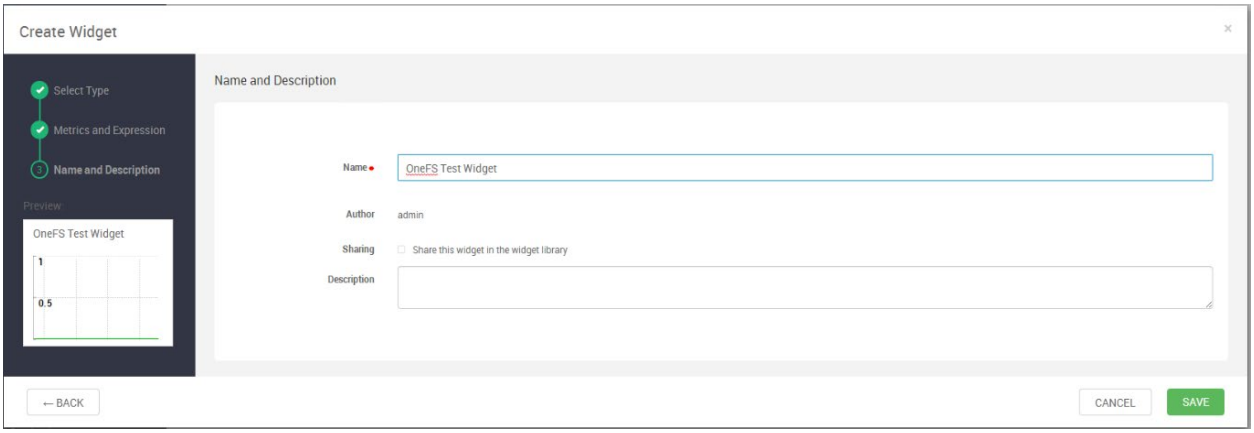
i. Provide a widget title.

ii. Under **Expression**, click **ADD METRIC** > **OneFS** > **All OneFS Clients** and then select a metric.

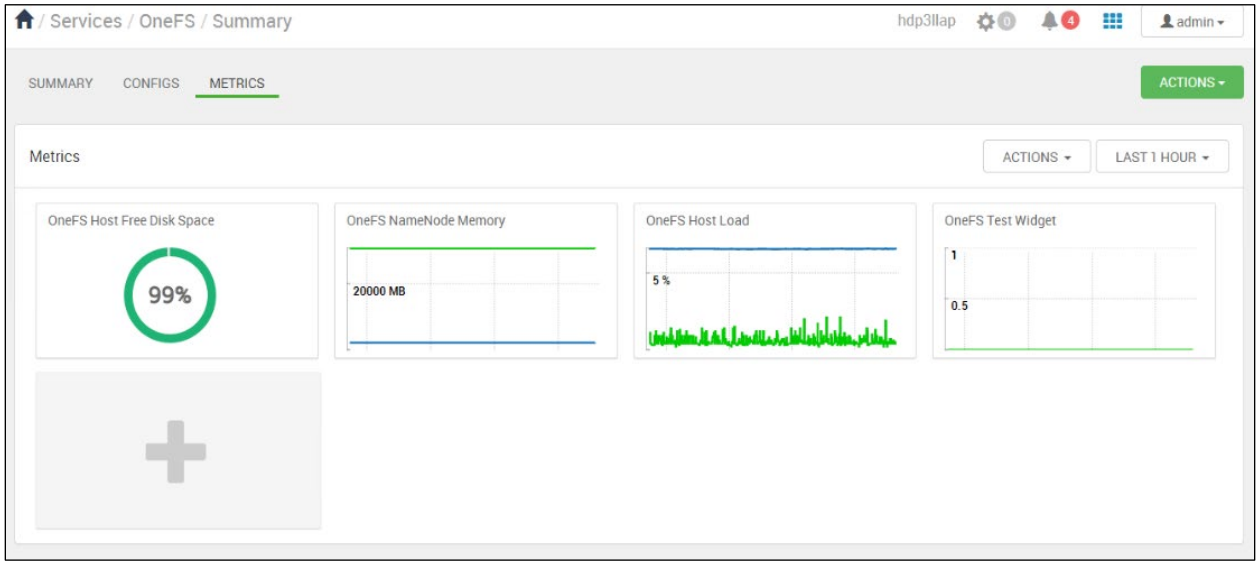


iii. Click **Next**.

d. On the **Name and Description** screen, provide the necessary details and click **SAVE**.



The new widget is created and appears on the **Metrics** page as shown:



HDFS wire encryption overview

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

Note:

- You can enable wire encryption per access zone in OneFS.
- Enabling HDFS wire encryption with an Access Zone could result in HDFS traffic performance degradation while accessing data in that zone. You can characterize the performance impact as wire encryption enabled to determine whether this is acceptable to your workload.

Configure HDFS Wire Encryption with OneFS

To use the wire encryption feature, you must deploy Kerberos on your Hadoop cluster. The following instructions assume that you have already deployed and enabled Kerberos on your cluster. You can then enable HDFS wire encryption in OneFS either using the OneFS web administration interface or the OneFS command-line administration interface.

Note: HDFS wire encryption that is supported by Dell EMC PowerScale is different from the Apache HDFS Transparent Data Encryption technology.

You can configure HDFS wire encryption using the OneFS web administration interface or command-line administration interface. See the Isilon OneFS HDFS Reference Guide for details.

Configure Apache Hadoop for HDFS wire encryption

To enable HDFS wire encryption with OneFS, edit the following attributes that are associated with Apache Hadoop:

Properties in core-site.xml	Value	Properties in hdfs-site.xml	Value
`hadoop.rpc.protection`	privacy	`dfs.encrypt.data.transfer`	true
		`dfs.encrypt.data.transfer.algorithm`	3des (Default value)
		`dfs.encrypt.data.transfer.cipher.suites`	`AES/CTR/NoPadding` (Default value)
		`dfs.encrypt.data.transfer.cipher.key.bitlength`	Select one of 128,192,356. The default value is 128.

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