IBM OPEN PLATFORM FOR APACHE HADOOP & DELL EMC ISILON INSTALLATION GUIDE

Abstract

This guide walks you through the process of configuring Dell EMC Isilon OneFS for the IBM Open Platform for Apache Hadoop

IBM Corporation
IBM Analytics
Route 100
Somers, NY 10589

Dell Technologies Corporation
Dell EMC
176 South Street
Hopkington, MA 01748

Produced in the United States of America

November 2016

CONTENTS

1 Introduction

Hadoop is an open-source framework that enables the distributed processing of large data sets across clusters of computers. 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 Dell EMC Isilon OneFS with Hadoop for use with the IBM Open Platform (IOP) and the Apache Ambari manager.

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

1.1 Audience

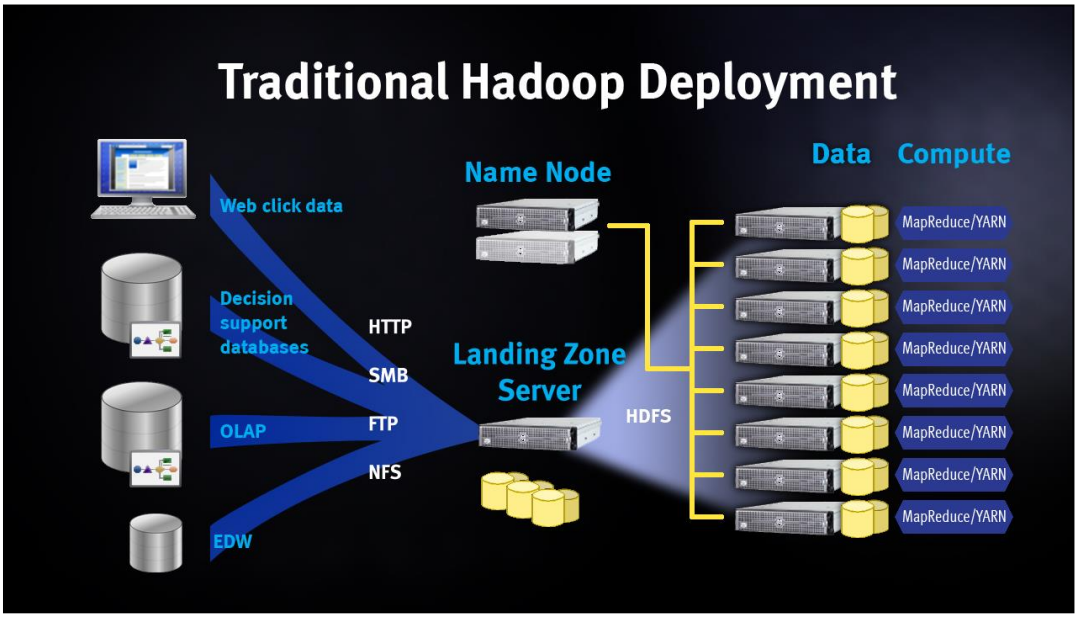
This guide is intended for systems administrators, IT program managers, IT architects, and IT managers who will be installing the Dell EMC Isilon OneFS 8.0.x with BigInsights 4.2.

1.2 IBM & Dell EMC Technology Highlights

The IBM® Open Platform with Apache Hadoop is comprised of entirely Apache Hadoop open source components, such as Apache Ambari, YARN, Spark, Knox, Slider, Sqoop, Flume, Hive, Oozie, HBase, ZooKeeper, and more. After installing IBM Open Platform, you can install additional IBM value-add service modules.

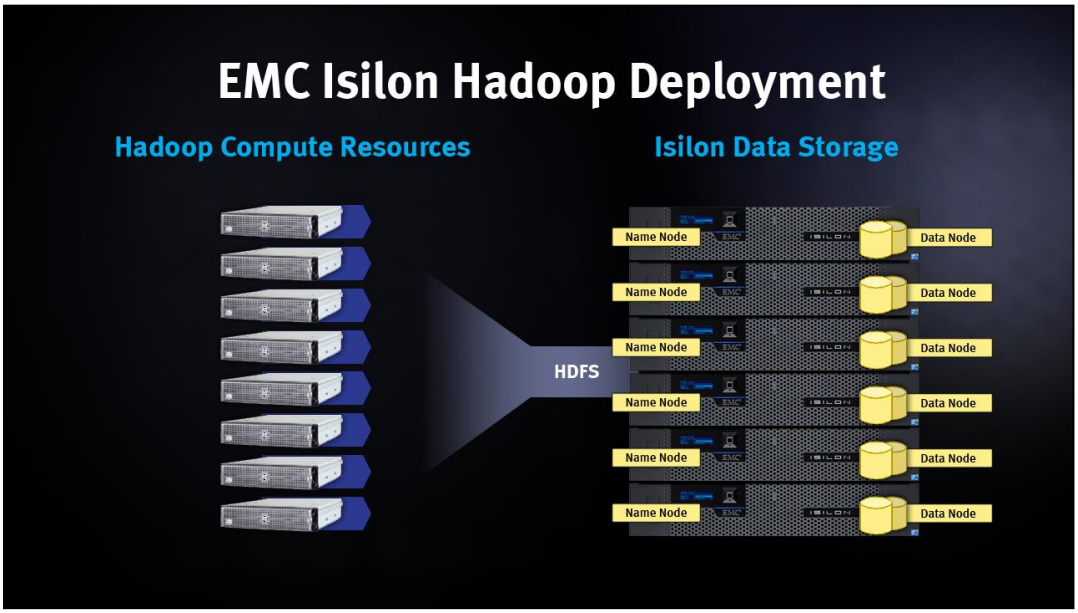
The Dell EMC Isilon OneFS scale-out network-attached storage (NAS) platform provides Hadoop clients with direct access to Big Data through a Hadoop Distributed File System (HDFS) interface. A Dell EMC Isilon cluster powered by the OneFS operating system delivers a scalable pool of storage with a global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

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 to be loaded on to HDFS. This process is network intensive and exposes the NameNode as a potential single point of failure.

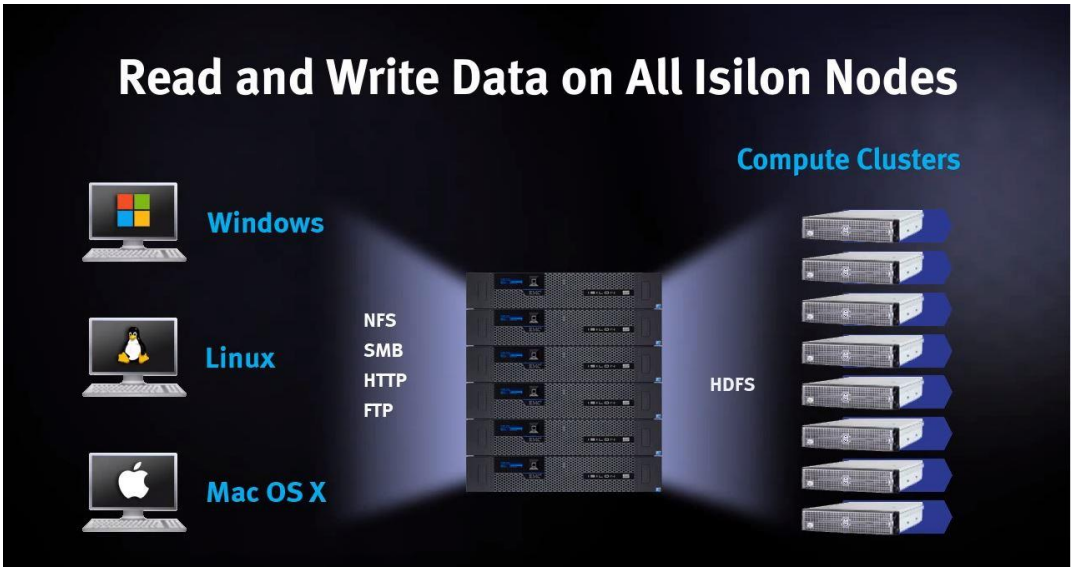


In a Dell EMC Isilon with Hadoop deployment, Isilon OneFS serves as the file system for Hadoop compute clients. On a OneFS cluster, every node in the cluster acts as a NameNode and DataNode, providing automated failover protection.

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



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


Read and Write Data on All Isilon Nodes
Compute Clusters
Windows
Linux
Mac OS X
NFS
SMB
HTTP
FTP
HDFS

1.3 Apache Hadoop Projects

Apache Hadoop is an open source, batch data processing system for enormous amounts of data. Hadoop runs as a platform that provides cost-effective, scalable infrastructure for building Big Data analytic applications. All Hadoop clusters contain a distributed file system called the Hadoop Distributed File System (HDFS) and a computation layer called MapReduce.

The Apache Hadoop project contains the following subprojects:

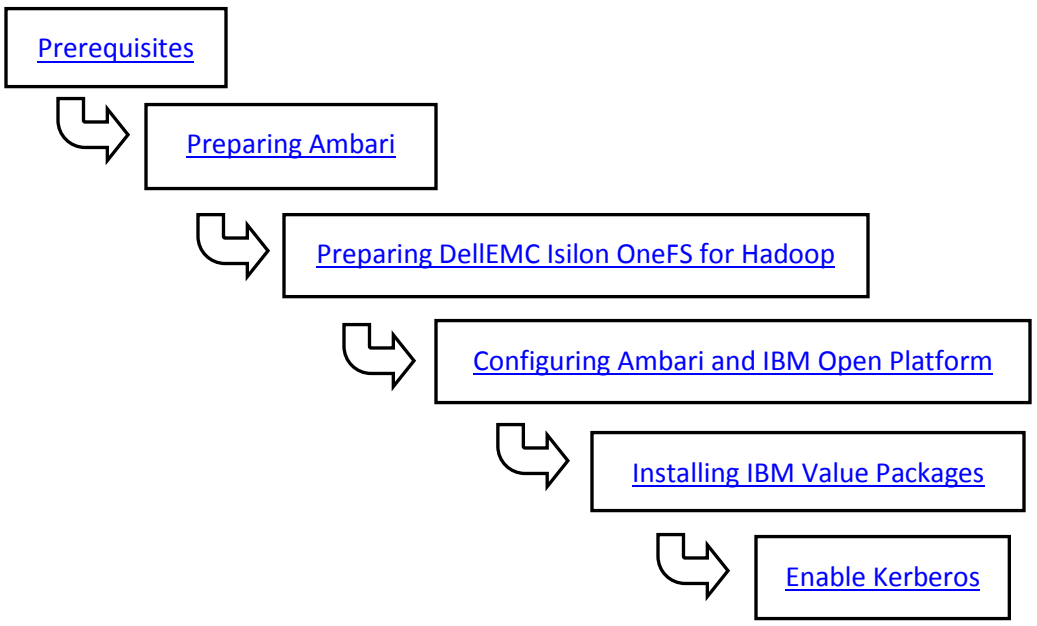
- Hadoop Distributed File System (HDFS) – A distributed file system that provides high-throughput access to application data.
- Hadoop MapReduce – A software framework for writing applications to reliably process large amounts of data in parallel across a cluster.

Hadoop is supplemented by an ecosystem of Apache projects, such as Pig, Hive, Sqoop, Flume, Oozie, Slider, HBase, Zookeeper and more that extend the value of Hadoop and improves its usability.

Version 2 of Apache Hadoop introduces YARN, a sub-project of Hadoop that separates the resource management and processing components. YARN was born of a need to enable a broader array of interaction patterns for data stored in HDFS beyond MapReduce. The YARN- based architecture of Hadoop 2.0 provides a more general processing platform that is not constrained to MapReduce.

For full details of the Apache Hadoop project see http://hadoop.apache.org/.

1.4 The following flow chart provides the process for installing and configuring:



2 Prerequisites

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

IBM Open Platform and the Ambari Manager

Ensure that the following requirements are met:

- Ambari 2.1.x or later.
- IBM Open Platform (IOP) 4.1.0.2 or later.
- Password-less SSH configured. See the IBM Open Platform documentation for configuring Password-less SSH.
- Familiarity with the Ambari and IBM Open Platform documentation and the installation instructions.
    - To view the Ambari and the IBM Open Platform (IOP) documents, go to https://www.ibm.com/support/knowledgecenter/SSPT3X_4.1.0/com.ibm.swg.im.infosphere.biginsights.welcome.doc/doc/welcome.html

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

Component	Version
Ambari version	
IOP stack version	
OneFS cluster name	
Ambari server (FQDN)	

BigInsights supported OS versions are as follow. Additional information is available at this link.

BigInsights Version	OS Version
4.1	RHEL6.5 or higher (64bit) RHEL7.1 or higher (64bit) Power system RHEL7.1 or higher (64bit) SLES11 SP3 or higher (64bit)
4.2	RHEL6.7 or higher (64bit) RHEL7.2 or higher (64bit)

Biginsights and Isilon software compatibility:

BigInsights Version	Isilon Version	Kerberos certified
4.1	7.x	No
4.2	8.0.0.x	Yes
	8.0.1.0	Yes

2.1 Additional Linux Packages to Install

Install the following packages on all Linux compute nodes.

- deltarpm
- python-deltarpm
- createrepo
- pam-1.1.1-17.el6.i686.rpm
- mysql-connector-java-5.1.17-6.el6.noarch.rpm
- ksh
- nc
- libdbi
- libstdc
- libaio
- java-1.7.0-openjdk-devel
- python-paramiko
- python-rrdtool-1.4.5-1.el6.rfx.x86_64
- snappy-1.0.5-1.el6.x86_64
- web-ui-framework

Install the above packages using the `yum install` command.

2.2 Dell EMC Isilon Requirements

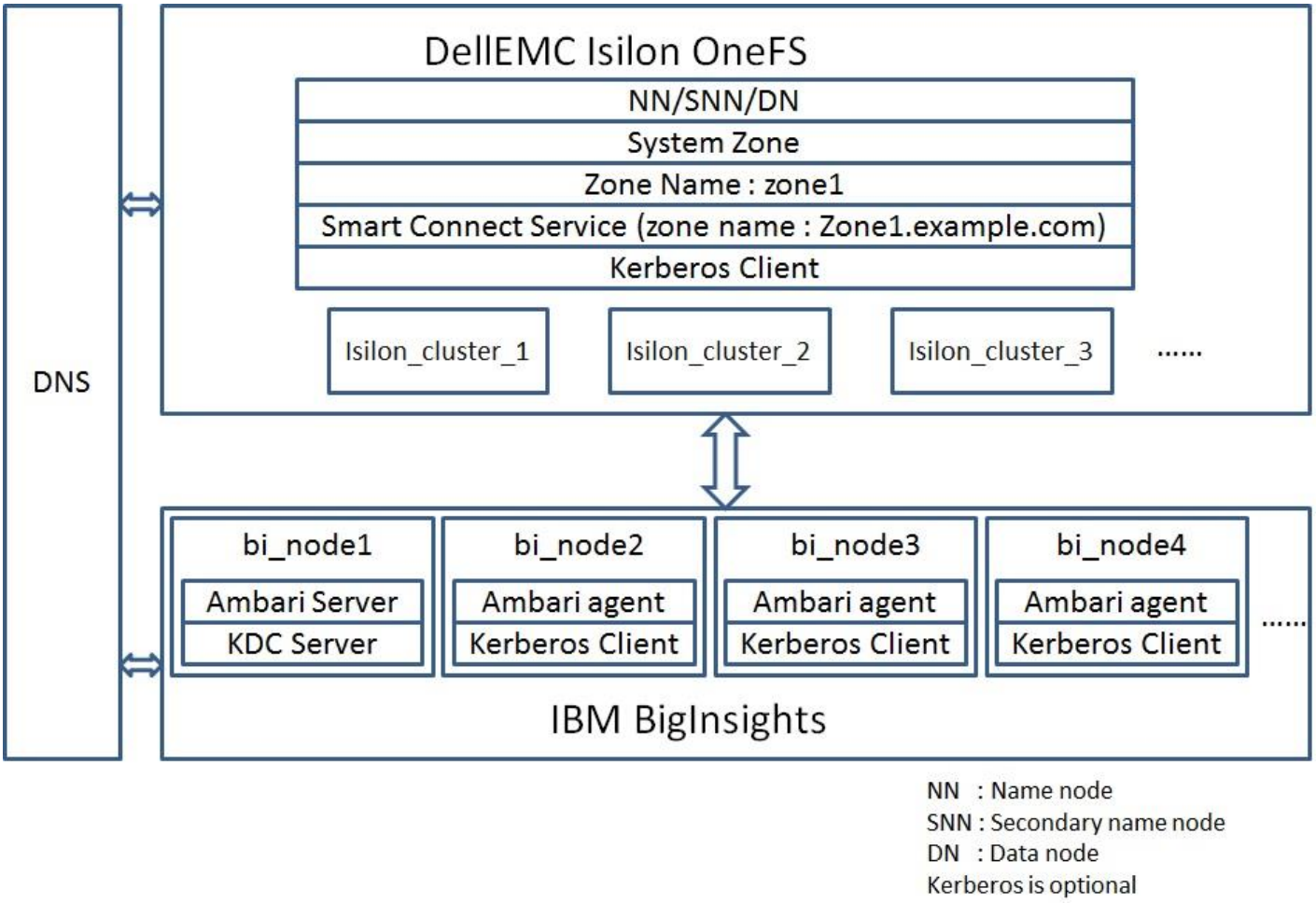
Ensure that the following requirements are met:

- A Dell EMC Isilon OneFS cluster is running per the supported compatibility matrix.
- The Dell EMC Isilon 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 Isilon module, is activated and SmartConnect is configured on your Isilon OneFS cluster.
- HDFS, a separately licensed Isilon module, is activated on your Isilon OneFS cluster. Contact your Dell EMC Isilon sales representative for more information about receiving your license keys.
- A valid Isilon SmartConnect SSIP and Domain Name System (DNS) delegation is in place to provide name resolution services for a SmartConnect zone. For more information, see Isilon External Network Connectivity Guide—Routing, Network Topologies, and Best Practices for SmartConnect.

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

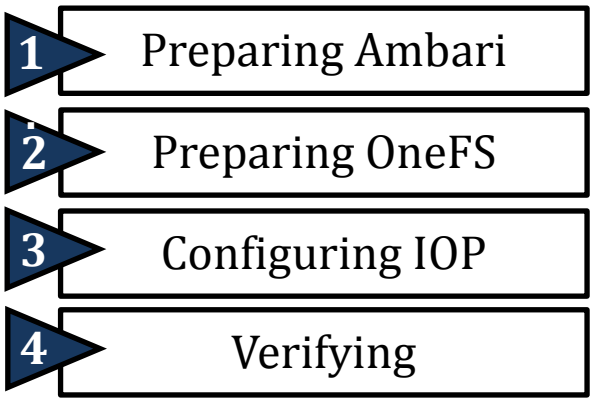
Component	Version/License
Isilon OneFS	
SmartConnect module	
HDFS module	
OneFS cluster name	

Following is the architecture of Isilon OneFS working with IBM BigInsights.



2.2.1 Installing Isilon OneFS with Hadoop

The installation of Isilon OneFS with Hadoop can be separated into four stages as represented in the following illustration. To complete the installation, you must perform tasks on both the IBM Open Platform /Ambari cluster and the Isilon OneFS cluster as outlined in this document.



2.3 Preparing Ambari

The steps in this phase will occur on the Ambari hosts, which will 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 needed by Ambari and IBM Open Platform to complete a successful Hadoop cluster installation. For more information on the installation process, go to the IBM Open Platform website: https://www.ibm.com/support/knowledgecenter/SSPT3X_4.2.0/com.ibm.swg.im.infosphere.biginsights.welcome.doc/doc/welcome.html

Steps to perform on the Hadoop client

To prepare Ambari for implementation, follow the instructions for your version of Ambari in the IBM Open Platform guide. In the topic, **check for your version from the drop-down menu** and proceed with the installation.

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

Important

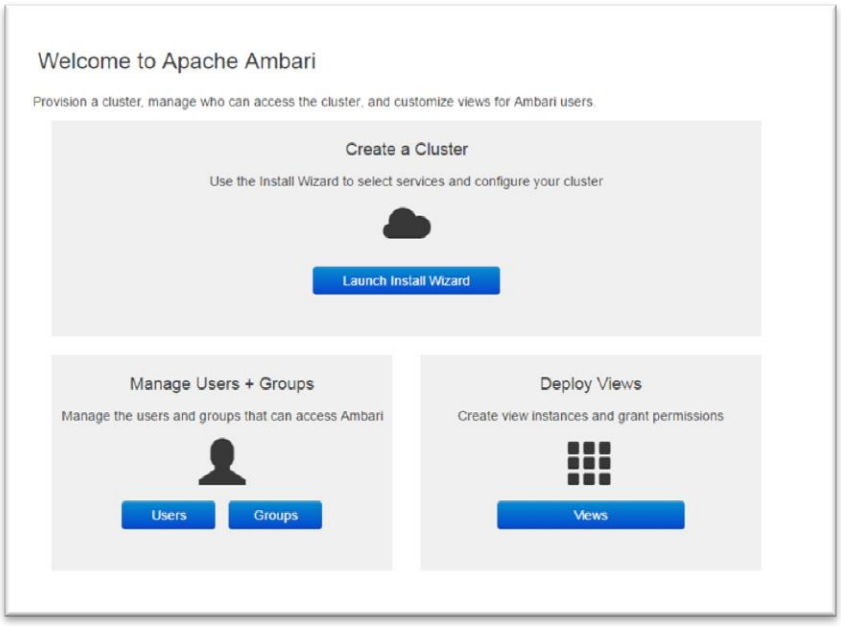
Complete the steps in the IBM Open Platform guide in section 1, “Important installation information ", until section 12, “Adding nodes to your cluster ". After you start the Ambari Server, do not continue with integration of the IBM Open Platform until after you complete the Preparing Dell EMC Isilon OneFS for Hadoop section of this guide.

You must complete the following steps in the IBM Open Platform guide:

1. Download the Ambari repository for the operating system that runs your installation host.
2. Set up the Ambari server.
3. Start the Ambari server.

After completing section 12 of the IBM Open Platform guide, Ambari should be operational, with the Ambari server running on one of the Hadoop nodes.

Logging in to the Ambari server will bring you to the create cluster page.



Note: Do not continue to integration until the OneFS cluster is prepared in the following steps and is ready to integrate into Ambari during the installation.

2.4 Preparing Dell EMC Isilon OneFS for Hadoop

Complete the following steps to configure your Dell EMC OneFS cluster for use with Ambari and IBM Open 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.

2.4.1 Validate OneFS version and license activation

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

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

   ```
   isi version
   ```

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

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

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

   ```
   isi license licenses list
   ```

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

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

```
isi license activate --key <key>
```

2.4.2 Configure DNS for Isilon

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

Set up DNS records for a SmartConnect zone. You must create the required DNS records that are used to access your OneFS cluster from the Hadoop cluster. All hosts in your Hadoop cluster must be configured for both forward and reverse DNS lookups Hadoop relies heavily on DNS and performs many DNS lookups during normal operation.

You can set up a SmartConnect zone for the connections from Hadoop compute clients. SmartConnect is a module that specifies how the Dell EMC OneFS cluster handles connection requests from clients. For additional information and best practices for SmartConnect, see the Isilon External Network Connectivity Guide.

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

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

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

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

2.4.3 Configure Isilon OneFS components

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

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

Use the following table to record the configuration information for the OneFS cluster IOP 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 server	
Ambari NameNode	
Open Data Platform (ODP) version	


2.4.4 Create an access zone

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

Steps to perform on the OneFS cluster

1. On a node in the OneFS cluster, create your Hadoop access zone by typing the following command:
```
isi zone zones create --name zone1 --path /ifs/zone1 --create-path
```

2. View the list of access zones. You should see the access zone that you just created (for example, zone1).
```
isi zone zones list
```
Output similar to the following appears:
```
Name   Path
--------------------------------
System /ifs
zone1  /ifs/zone1
--------------------------------
```
3. Create the HDFS root directory within the access zone that you created.
```
mkdir -p /ifs/zone1/hadoop
```

2.4.5 Configure SmartConnect

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

Steps to perform on the OneFS cluster

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

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

```
   isi network pools create --id=groupnet0:subnet0:<name> --ranges=x.x.x.x-x.x.x.x --access-
zone=<my-access-zone> --alloc-method=static --ifaces=1-3:ext-1 --sc-subnet=<subnet0> --sc-
dns-zone=<my-smartconnectzone-name> --description=<description>
```
If the command succeeds, the OneFS cluster responds with output similar to the following:
```
   Creating pool 'subnet0:pool1':                                          OK
   Saving:                                                                 OK
```

2. View the list of IP pools by typing the following command:
```
   isi network pools list -v
```

2.4.6 Verify the SmartConnect configuration

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

Step to perform on the Hadoop client

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

```
ping zone1.example.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.

2.5 Create and configure Isilon HDFS root

On a node in the OneFS cluster, assign the HDFS root directory.

Steps to perform on the OneFS cluster

1. Set the HDFS root directory for the access zone by running the following command:
   ```
   isi hdfs settings modify --zone=zone1 --root-directory=/ifs/zone1/hadoop
   ```
2. Assign the Ambari NameNode in the access zone and associate the SmartConnect name with it by running the following command:
   ```
   isi hdfs settings modify --zone=zone1 --ambari-namenode=zone1.example.com
   ```
3. Assign the Ambari server in the access zone.
   Note that the host FQDN specified here is the one running Ambari Server.
   ```
   isi hdfs settings modify --zone=zone1 --ambari-server=bi_cluster1.example.com
   ```
4. Map the HDFS user to the Isilon super user. Create a user mapping rule to map the HDFS user to the OneFS root account. This mapping enables the services from the Hadoop cluster to communicate with the OneFS cluster using the correct credentials.
   ```
   isi zone zones modify --user-mapping-rules="hdfs=>root" --zone=zone1
   ```
5. Set the HDFS block size that is used for reading from Isilon by running the following command:
   ```
   isi hdfs settings modify --zone=zone1 --default-block-size=128M
   ```
6. Verify that the access zones are set up correctly by running the following command:
   ```
   isi zone zones view zone1
   ```
   Output similar to the following appears.

   ```
                  Name: zone1
                  Path: /ifs/zone1
              Groupnet: groupnet0
        Map Untrusted: -
       Auth Providers: lsa-activedirectory-provider:FOO.COM, lsa-ldap-provider:AD-FOO, lsa-
local-provider:IBM.COM
          NetBIOS Name: -
    User Mapping Rules: hdfs => root [], FOO\* &= * []
Home Directory Umask: 0077
    Skeleton Directory: /usr/share/skel
    Cache Entry Expiry: 4H
               Zone ID: 2
   ```
7. Review the HDFS settings.
   ```
   isi hdfs settings view --zone=zone1
   ```
8. Create an indicator file in the Hadoop directory to view your OneFS cluster and access zone through HDFS.
   ```
   touch /ifs/isiloncluster1/zone1/hadoop/THIS_IS_ISILON_isiloncluster1_zone1.txt
   ```

2.5.1 Modify the access control list (ACL) setting for OneFS

Step to perform on the OneFS cluster

Run the following command to modify ACL settings BEFORE you create directories or files in the next section. This creates the correct permission behavior on the cluster for HDFS. **Note**: ACL policies are cluster-wide, so you should understand this change before performing it on production clusters.

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

2.5.2 Creating users, group IDs, and directories

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

Important

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

When installing Hadoop, the 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 Isilon OneFS, they need to be created manually. You must create the Isilon *yarn* local account user in the Isilon access zone in which *yarn* will access data. You need to create a local user *yarn* with the UID of 502 and the 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:

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

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

2.5.3 Creating users on the OneFS cluster using Isilon scripts

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

1. Create local users and groups on Isilon OneFS.

2. Collect the UIDs and GIDs of the users.

3. Pre-create local users and groups on all IOP hosts to be deployed.

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

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

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

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

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

hdfs, hadoop, mapred, hbase, knox, uiuser, dsmadmin, ambari-qa, rrdcached, hive, yarn, hcat, bigsql, tauser, bigr, flume, ams, kafka, solr, spark, sqoop, zookeeper, oozie, bighome, titan, ranger

For more information on the list of accounts you need based on services, go to https://www.ibm.com/support/knowledgecenter/SSPT3X_4.2.0/com.ibm.swg.im.infosphere.biginsights.install.doc/doc/c0057609.html

Steps to perform on the OneFS cluster

1. On a node in the OneFS cluster, create a scripts directory. You will extract the scripts to this directory.
```
mkdir -p /ifs/scripts
```

2. Change directories to the scripts directory that you created.
```
cd /ifs/scripts
```
Download the latest version of the Isilon Hadoop tools into the */ifs/isiloncluster1/scripts* directory from https://github.com/Isilon/Isilon_hadoop_tools

3. To create all required local users and groups on your OneFS cluster for the Hadoop services and applications, run the following script in the onefs subdirectory

Script usage:

```
bash isilon_create_users.sh --dist <DIST> --startgid <GID> --startuid <UID> --zone <ZONE> --append-cluster-name
```
Where:

`Dist`	**This will correspond to your Hadoop distribution - bi.**
`Startgid`	The beginning UID range for the creation of users (the default is 1000).
`Startuid`	The beginning GID range for the creation off users (the default is 1000).
`Zone`	The name of the access zone where the users should be created (useful for multi-tenant environments that will use a single KDC).
`append-cluster-name`	The Hadoop cluster name the script should append to the usernames.

Example:

```
   bash /ifs/scripts/isilon-hadoop-tools/onefs/isilon_create_users.sh --dist bi --startgid 501 --startuid 501 --zone zone1
```

Output like the following appears:

```
   Info: Hadoop distribution:  bi
   Info: groups will start at GID 501
   Info: users will start at UID 501
   Info: will put users in zone:  zone1
   Info: HDFS root:  /ifs/zone1/hadoop
   SUCCESS -- Hadoop users created successfully!
   Done!
```

To create directories, download the *isilon_create_directories.sh* from https://github.com/Isilon/Isilon_hadoop_tools

The following table lists the users and directories that the script creates:

Users	Directories they own
accumulo	/apps/accumulo
ambari-qa	/app-logs/ambari-qa
ambari-qa	/app-logs/ambari-qa/logs
ambari-qa	/user/ambari-qa
hbase	/apps/hbase/data
hbase	/apps/hbase/staging
Hcat	/apps/webhcat
Hcat	/user/hcat
Hdfs	/
Hdfs	/tmp
Hdfs	/apps
Hdfs	/apps/hbase
Hdfs	/apps/hive
Hdfs	/user
Hdfs	/user/hdfs

Users	Directories they own
Hive	/apps/hive/warehouse
Hive	/user/hive
mapred	/mapred
mapred	/mapred/system
Oozie	/user/oozie
Yarn	/app-logs
Yarn	/user/yarn
Yarn	/system/yarn/node-labels

2.5.4 Creating users on the OneFS cluster manually

To add more users, in addition to the users that the Isilon script configures, on the OneFS cluster you can add a user for each additional Hadoop user that submits MapReduce jobs. The following commands show how to manually add a single user called *biuser1*.

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 under this situation.

Steps to perform on the OneFS cluster

1. Add a group to the OneFS cluster.
```
isi auth groups create biuser1 --zone zone1 --provider local --gid <GID>
```
2. Create the user and the user's Hadoop home directories on the OneFS cluster.
```
isi auth users create biuser1 --primary-group biuser1 --zone zone1 --provider local --
home-directory /ifs/isiloncluster1/zone1/hadoop/user/biuser1 --uid <UID>
```
3. Assign permissions to the user's home directory on the Hadoop cluster. The ID 2 in the example below is from when you previously ran the `isi zone zones view zone1` command.
```
isi_run -z2 chown biuser1:biuser1 /ifs/isiloncluster1/hadoop/user/biuser1
chmod 755 /ifs/isiloncluster1/hadoop/user/biuser1
```

Steps to perform on the Hadoop client

Since you created a new user on Isilon OneFS to run jobs, you need to create the same user with UID parity on any Linux hosts that the user will access to run jobs if you want to use Isilon NFS with these edge nodes as well.

1. Add the user to the Hadoop cluster.
```
adduser biuser1 -u <UID>
or run the script : bi_create_users.sh to batch create users.
```

2.6 Configuring Ambari and IBM Open Platform

Perform the steps of the IBM Open Platform Ambari installation guide.

You must configure OneFS as follows:

1. Specify the Hadoop client and Isilon SmartConnect zone as a Target Hosts.
2. Assign NameNode and Secondary NameNode components to Isilon SmartConnect.
3. Assign DataNodes only to Isilon SmartConnect. Do not add DataNodes to any of the IOP hosts.
4. Remove any client components for Isilon.

2.6.1 Steps to perform in your browser from your client

From your browser, type the URL for your Hadoop client. Launch the Ambari Install wizard, and complete the installation steps in the IBM Open Platform guide, and the following steps that are specific to Isilon.

1. Access the Ambari web user interface from a web browser by using the server name (the fully qualified domain name, or the short name) on which you installed the software, and port 8080. For example, enter *abc.com*:8080.

   You can use any available port other than 8080 that will allow you to connect to the Ambari server. In some networks, port 8080 is already in use. To use another port, do the following:

   a. Edit the ambari.properties file:

   ```
   vi /etc/ambari-server/conf/ambari.properties
   ```

   b. Add a line in the file to select another port:

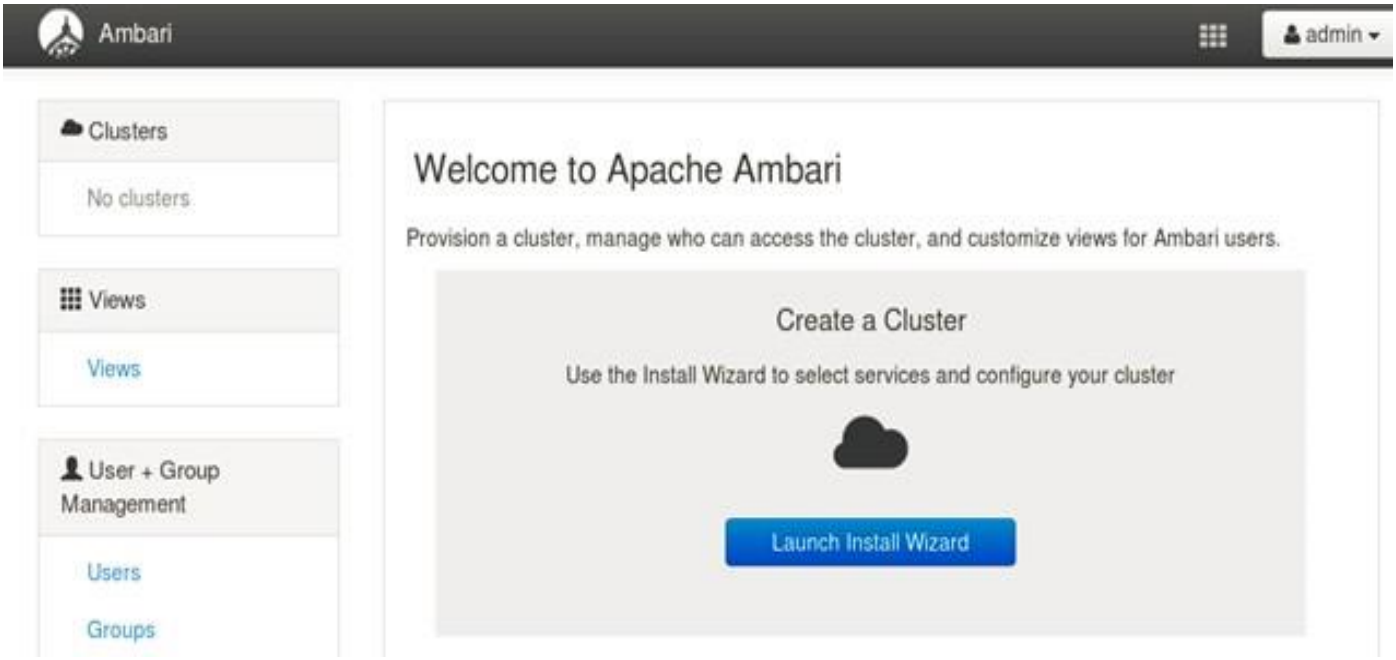
   ```
   client.api.port=8081
   ```

   c. Save the file and restart the Ambari server:

   ```
   ambari-server restart
   ```

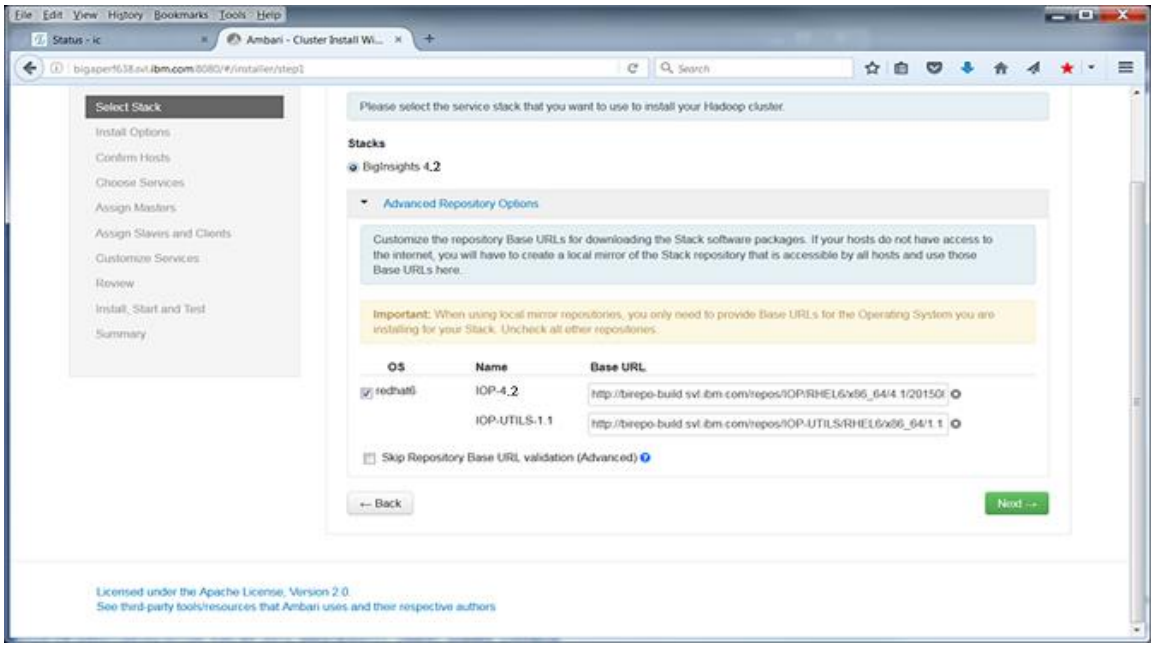
2. Log in to the Ambari server with the default username and password: **admin**/**admin**.

   The default **username** and **password** is required only for the first login. You can configure users and groups after the first login to the Ambari web interface.



On the Welcome page, click **Launch Install Wizard**.

3. On the Get Started page, enter a name for the cluster you want to create. The name cannot contain blank spaces or special characters. Click **Next**.



4. You will deploy IBM Open Platform for Apache Hadoop with Dell EMC Isilon. The Ambari Server allows for the immediate usage of an Isilon cluster for all HDFS services (NameNode and DataNode), no reconfiguration will be necessary once the IBM Open Platform installation is completed.
   SSH into Isilon as root and configure the Ambari Agent.

   isi hdfs settings modify --zone=zone1 --ambari-namenode= zone1.example.com

   isi hdfs settings modify --zone=zone1 --ambari-server=bi_node1.example.com

   The settings afterwards:

```
ic-1# isi hdfs settings view --zone=zone1
                 Service: Yes
    Default Block Size: 128M
Default Checksum Type: none
  Authentication Mode: all
       Root Directory: /ifs/zone1/hadoop
      WebHDFS Enabled: Yes
        Ambari Server: bi_node1.example.com
      Ambari Namenode: zone1.example.com
          Odp Version: -
```

   ic-1# isi services hdfs disable

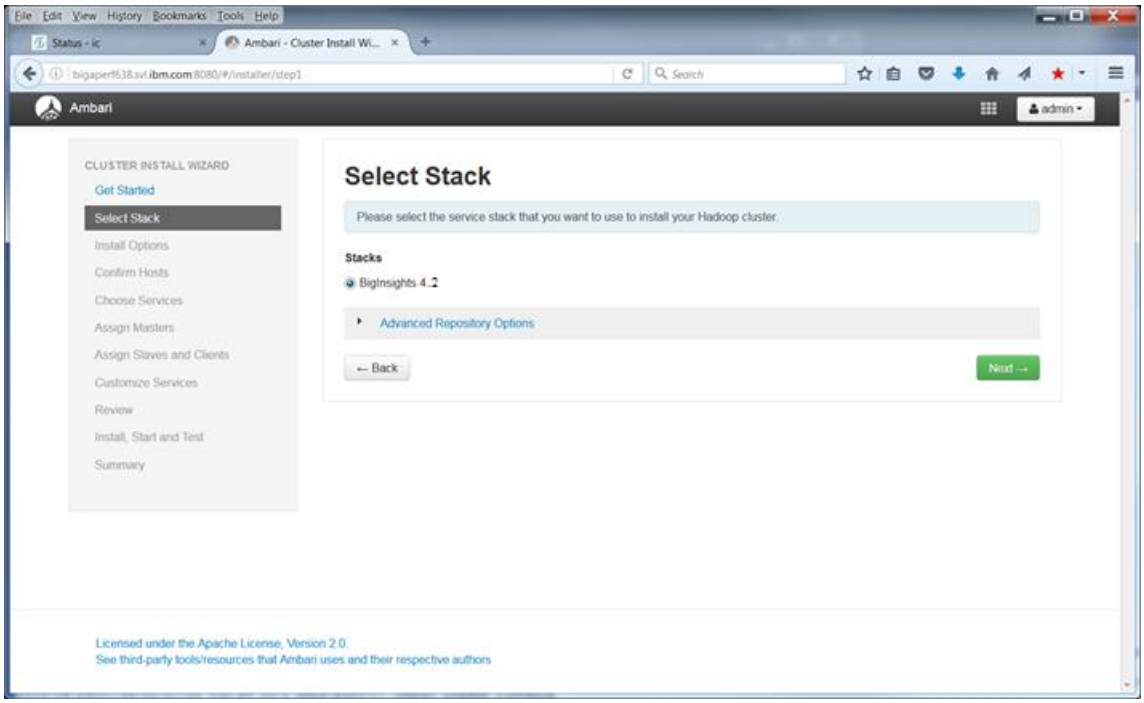
   The service 'hdfs' has been disabled.

   ic-1# isi services hdfs enable

   The service 'hdfs' has been enabled.

   ic-1#

5. On the **Select Stack** page, click the Stack version you want to install (for example, IBM BigInsights 4.2).



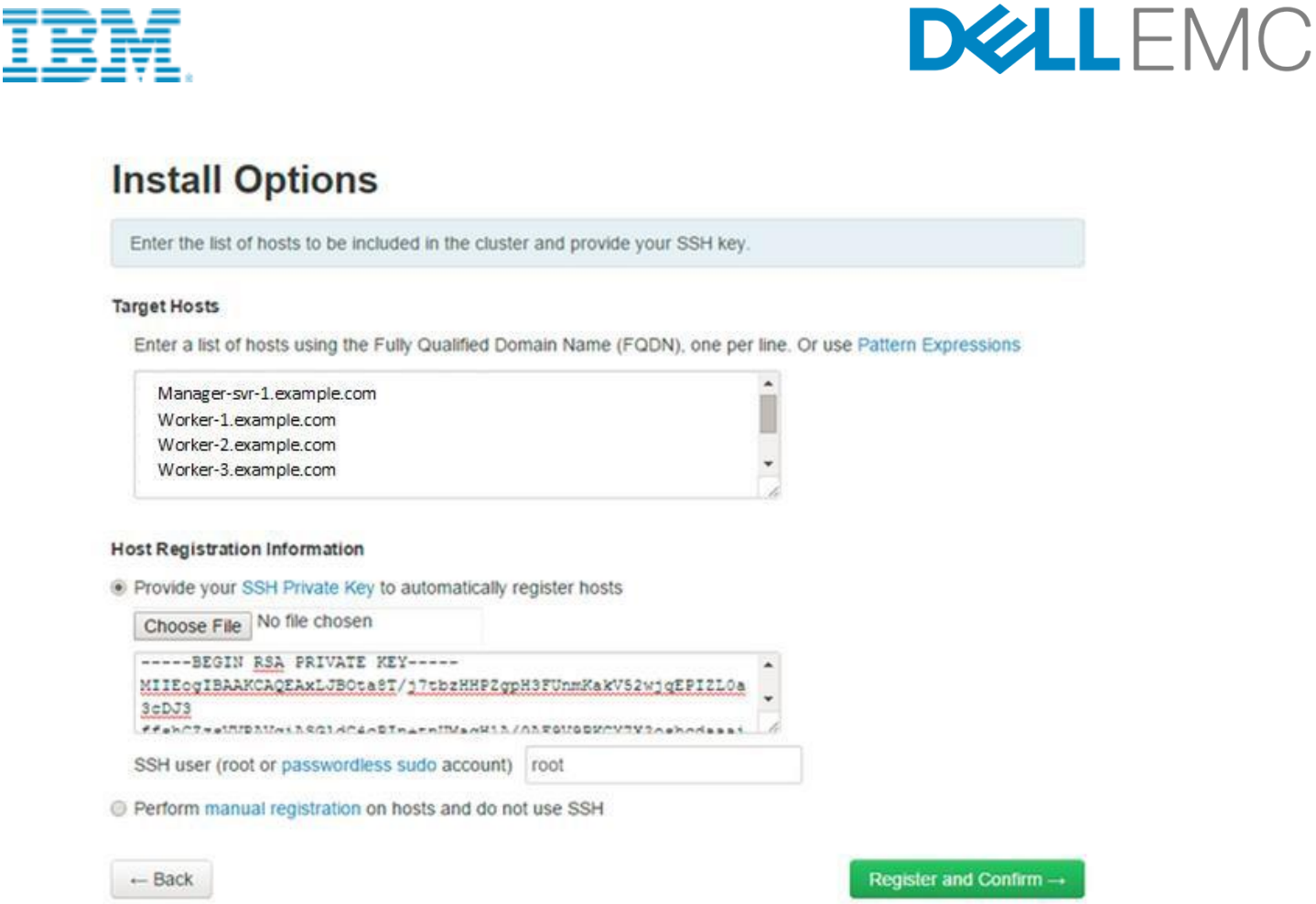
Click **Next**.

6. On the Install Options page in **Target Hosts**, add the list of Linux hosts that the Ambari server will manage and the IBM Open Platform with Apache Hadoop software will deploy one node per line. For example, enter

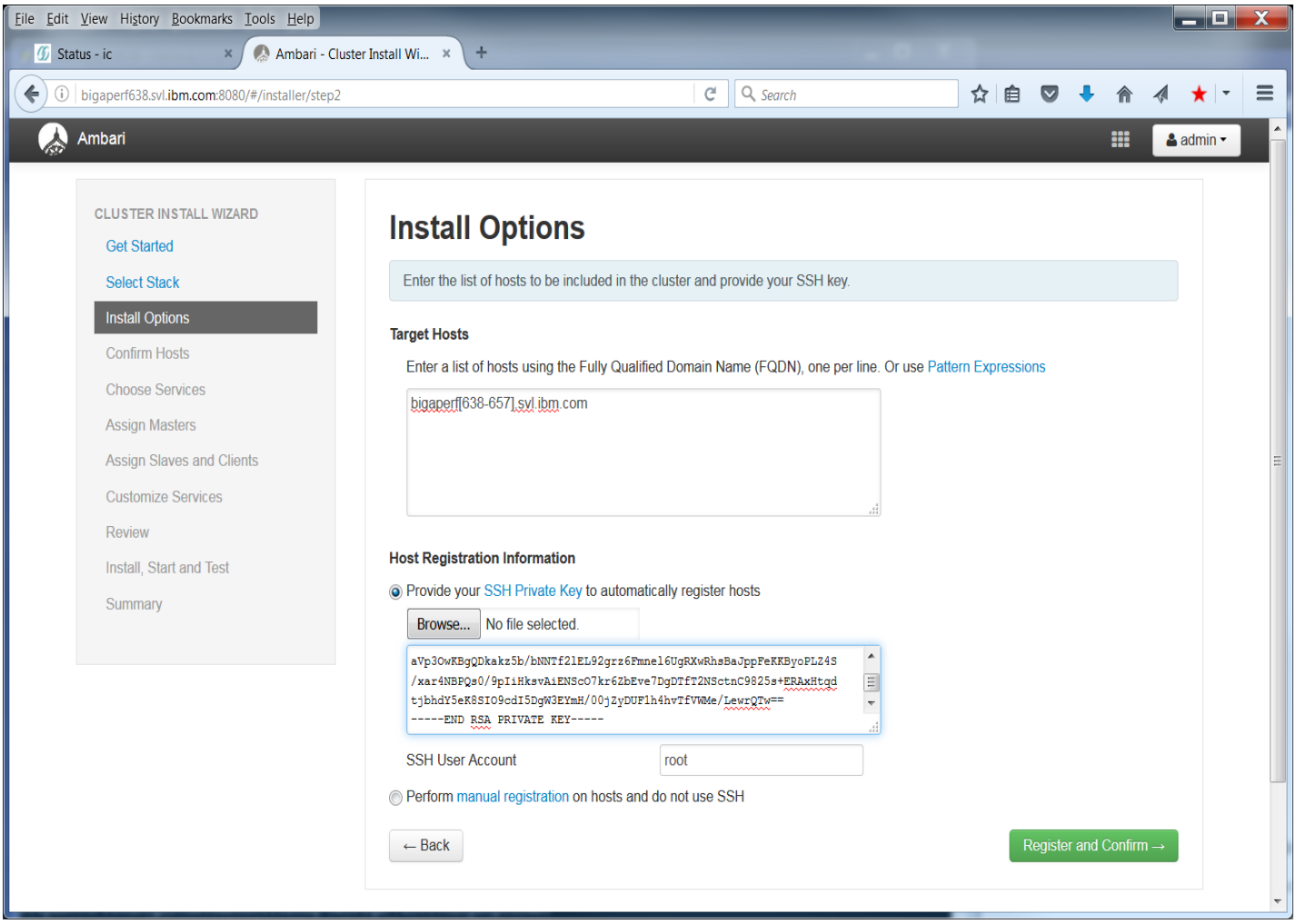
```
host1.example.com
host2.example.com
host3.example.com
host4.example.com
```

In the **Host Registration Information**, select one of the two options:

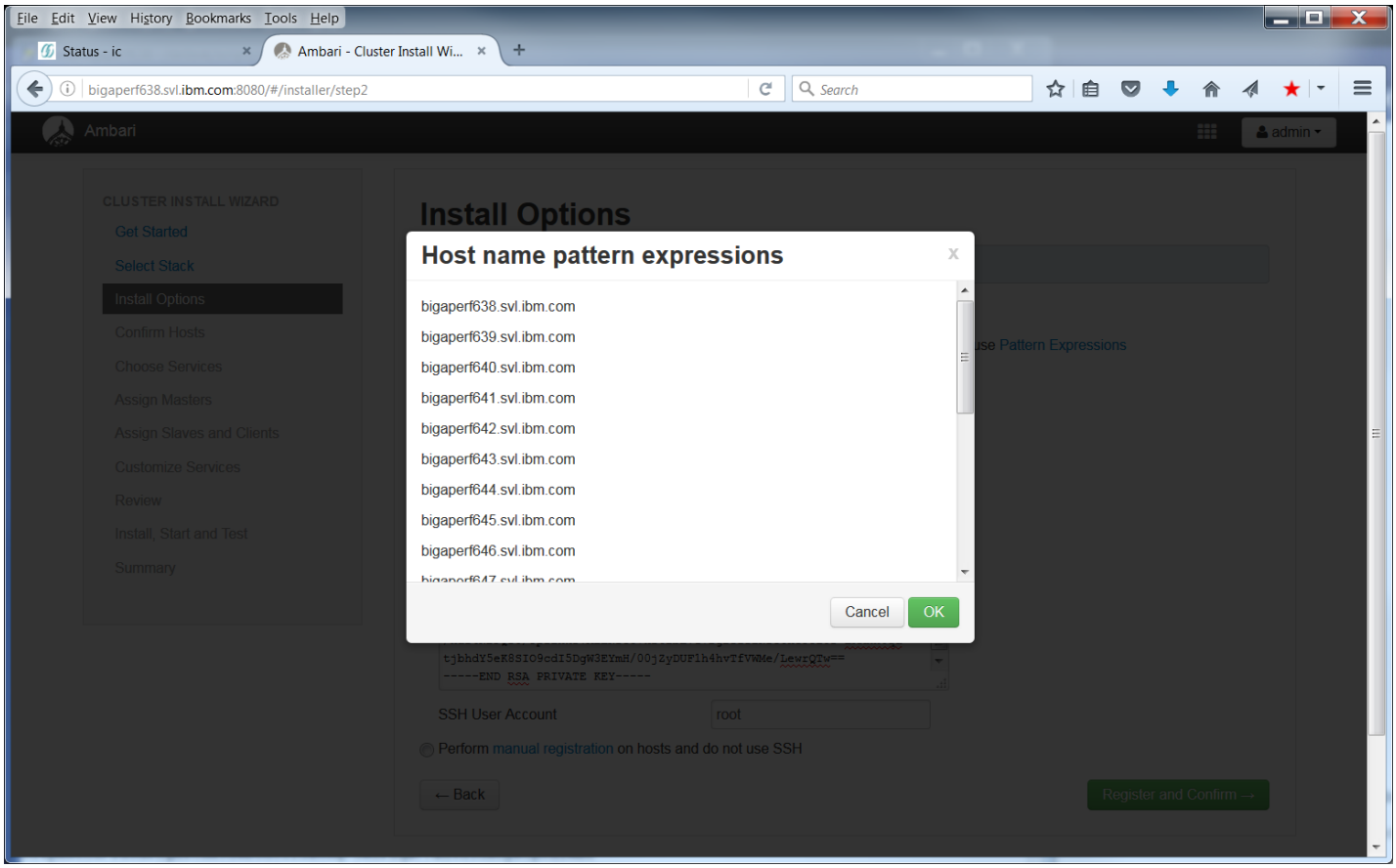
Provide the SSH Private Key to automatically register hosts or the manual registration.


Install Options
Enter the list of hosts to be included in the cluster and provide your SSH key.
Target Hosts
Enter a list of hosts using the Fully Qualified Domain Name (FQDN), one per line. Or use Pattern Expressions
Manager-svr-1.example.com
Worker-1.example.com
Worker-2.example.com
Worker-3.example.com
Host Registration Information
Provide your SSH Private Key to automatically register hosts
Choose File
No file chosen
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAxLJBOta8T/17tbzHHPZgpH3FUnmKaKV52WjqEPIZL0a
3cDJ3
SSH user (root or passwordless sudo account)
root
Perform manual registration on hosts and do not use SSH
← Back
Register and Confirm →

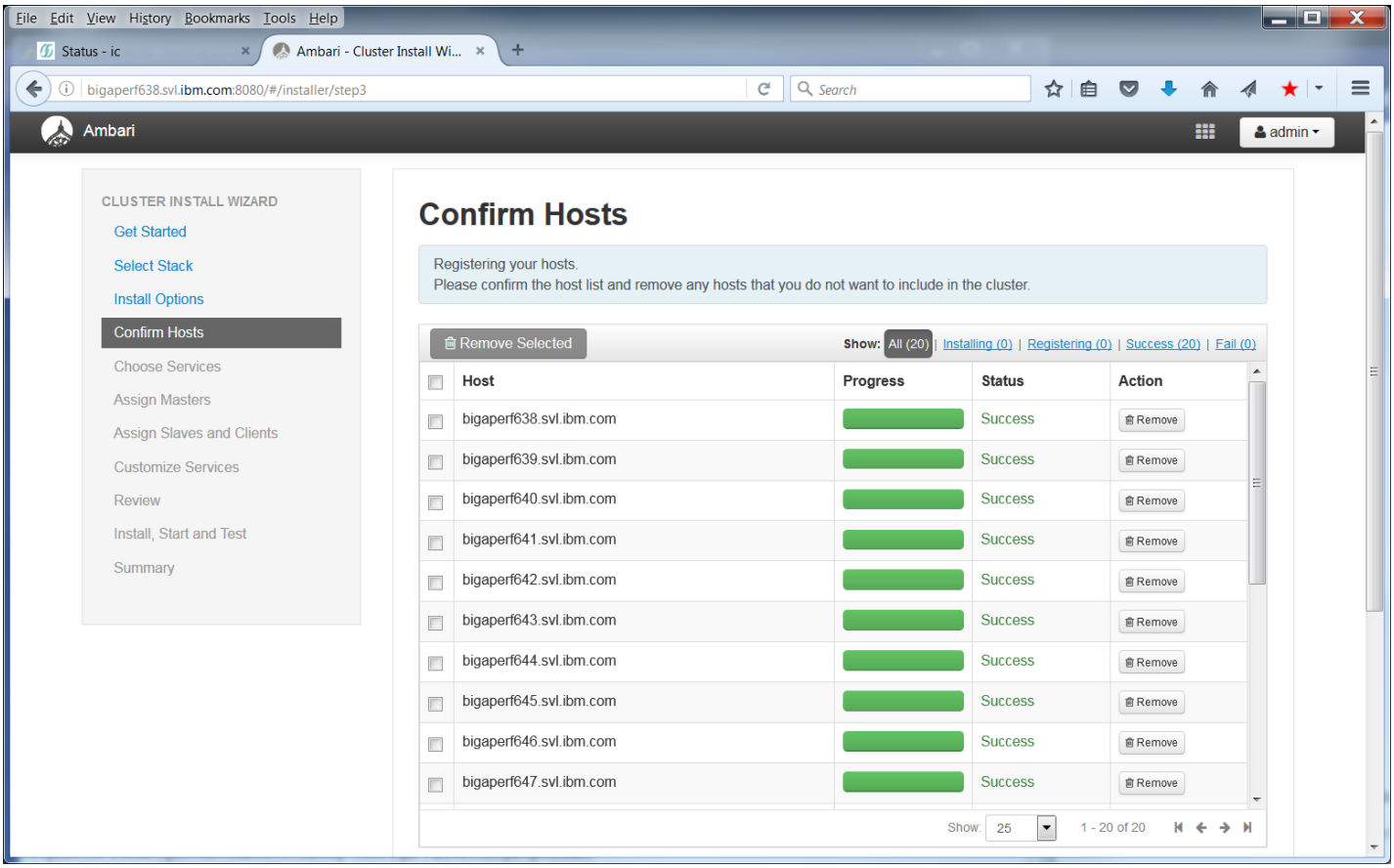
7. Click **SSH Private Key**. The private key file is /root/.ssh/id_rsa, where the root user installed the Ambari server.



Click **Choose File** to find the private key file you installed previously. You should have retained a copy of the SSH private key (.ssh/id_rsa) in your local directory when you set up password-less SSH. Copy and paste the key into the text box manually. Click the **Register and Confirm** button.

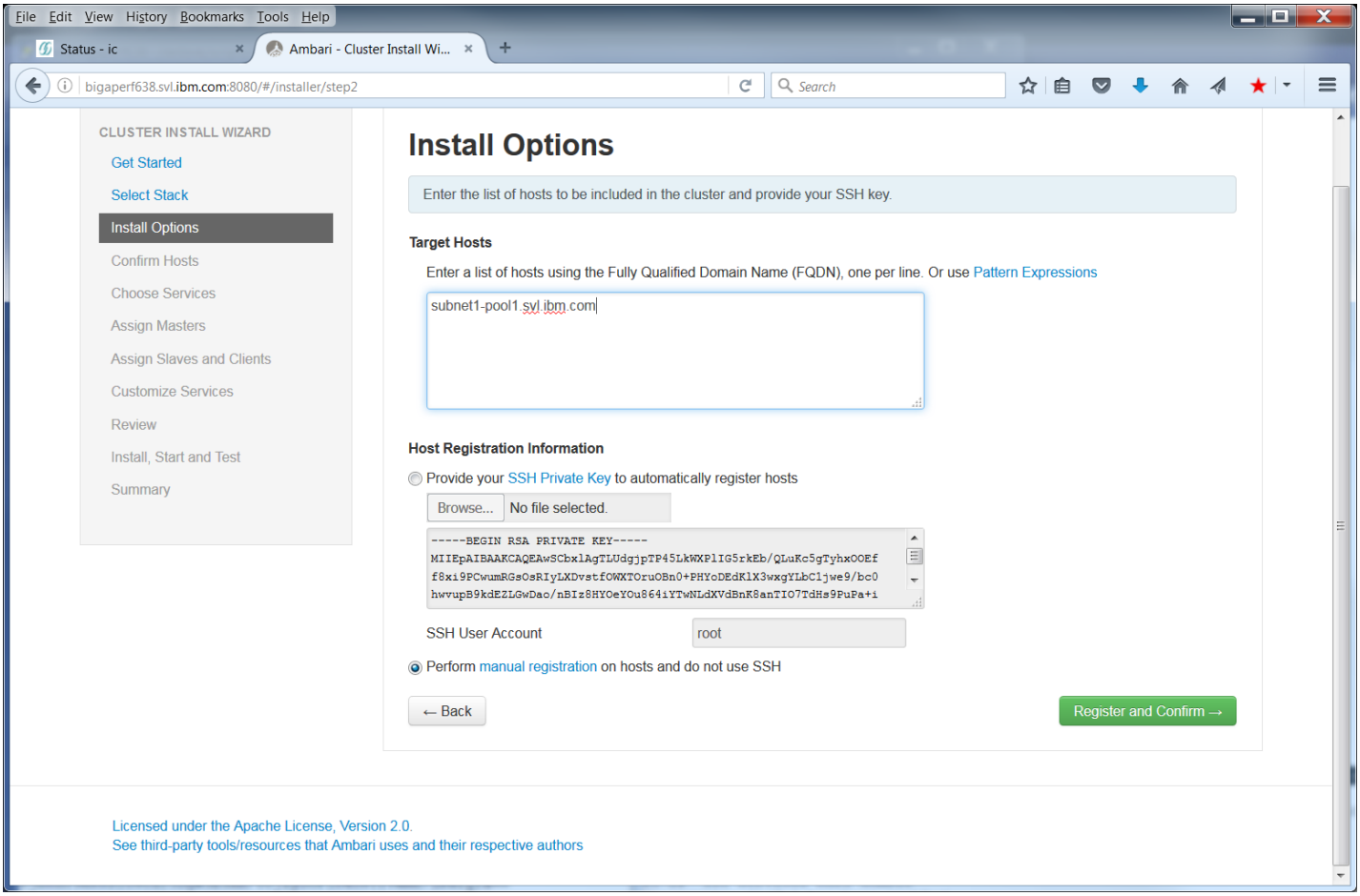


Click **OK**. Registering starts.



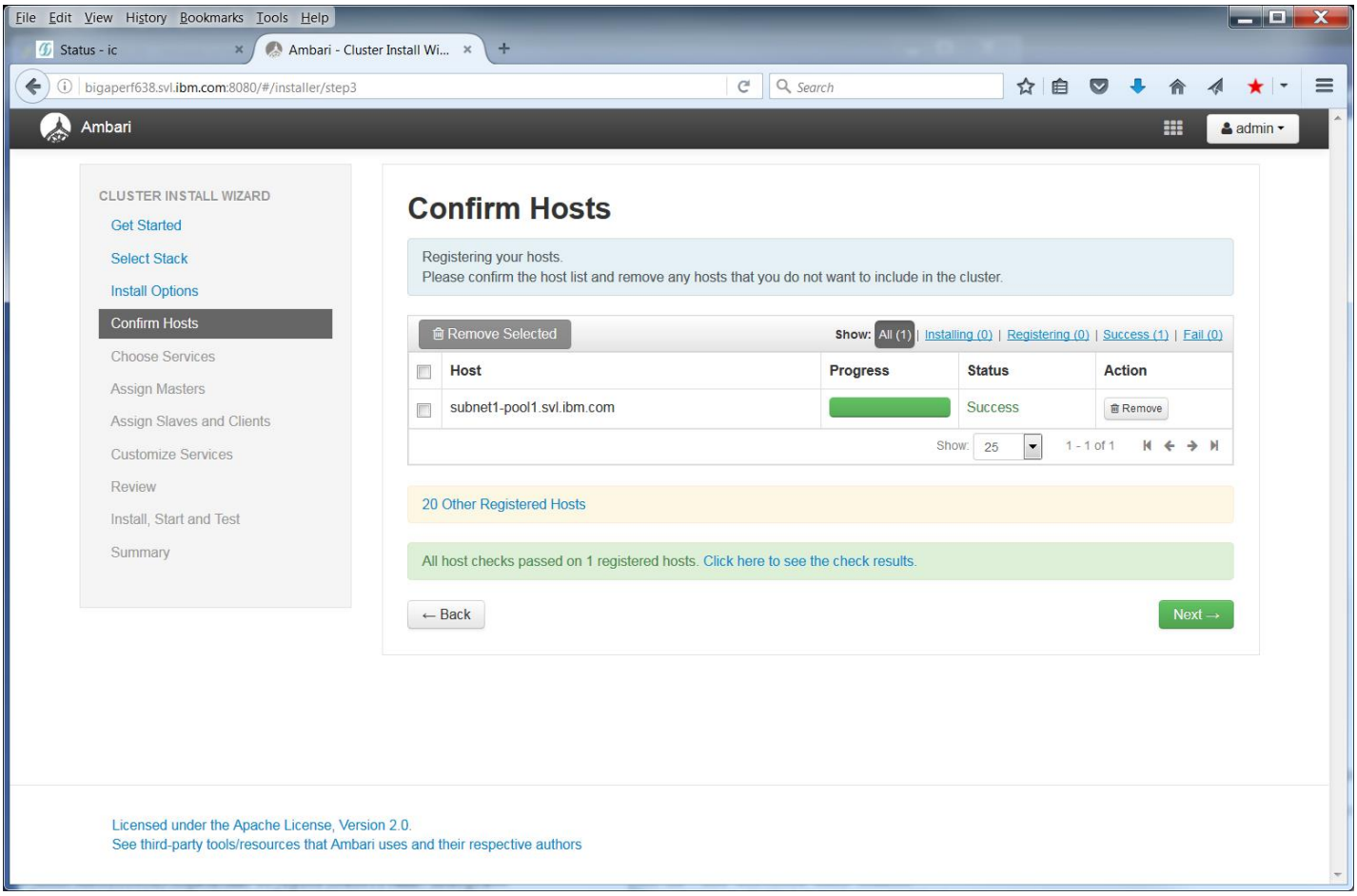
Note: After the Linux hosts are registered, click the back button, then click **Perform manual registration for Isilon and do not use SSH**.

8. Enter the Isilon subnet address and click **Perform manual registration**:



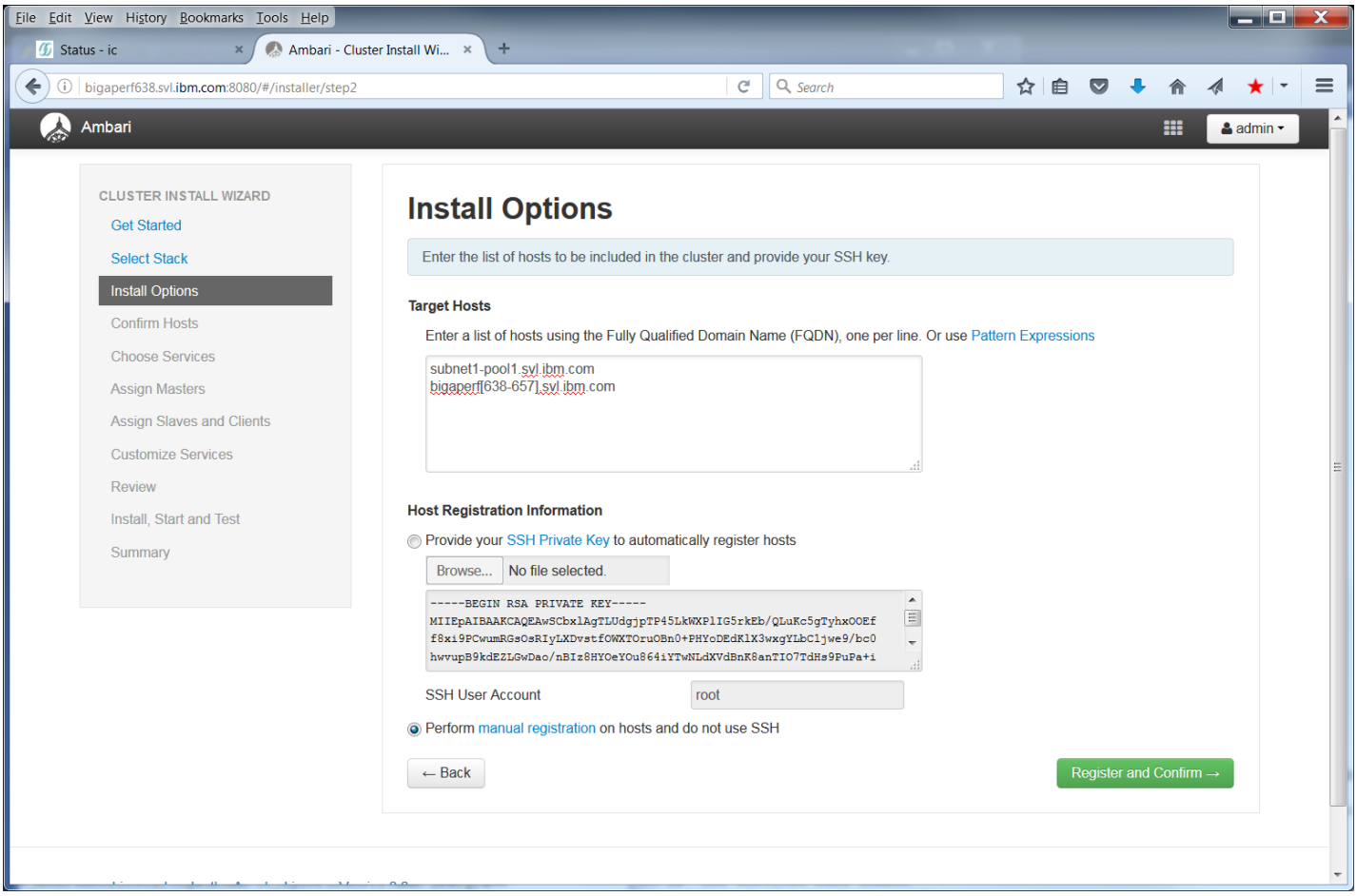
Click **Register and Confirm.**

The reminder to have an Ambari agent on Isilon installed can be ignored by clicking **OK.**



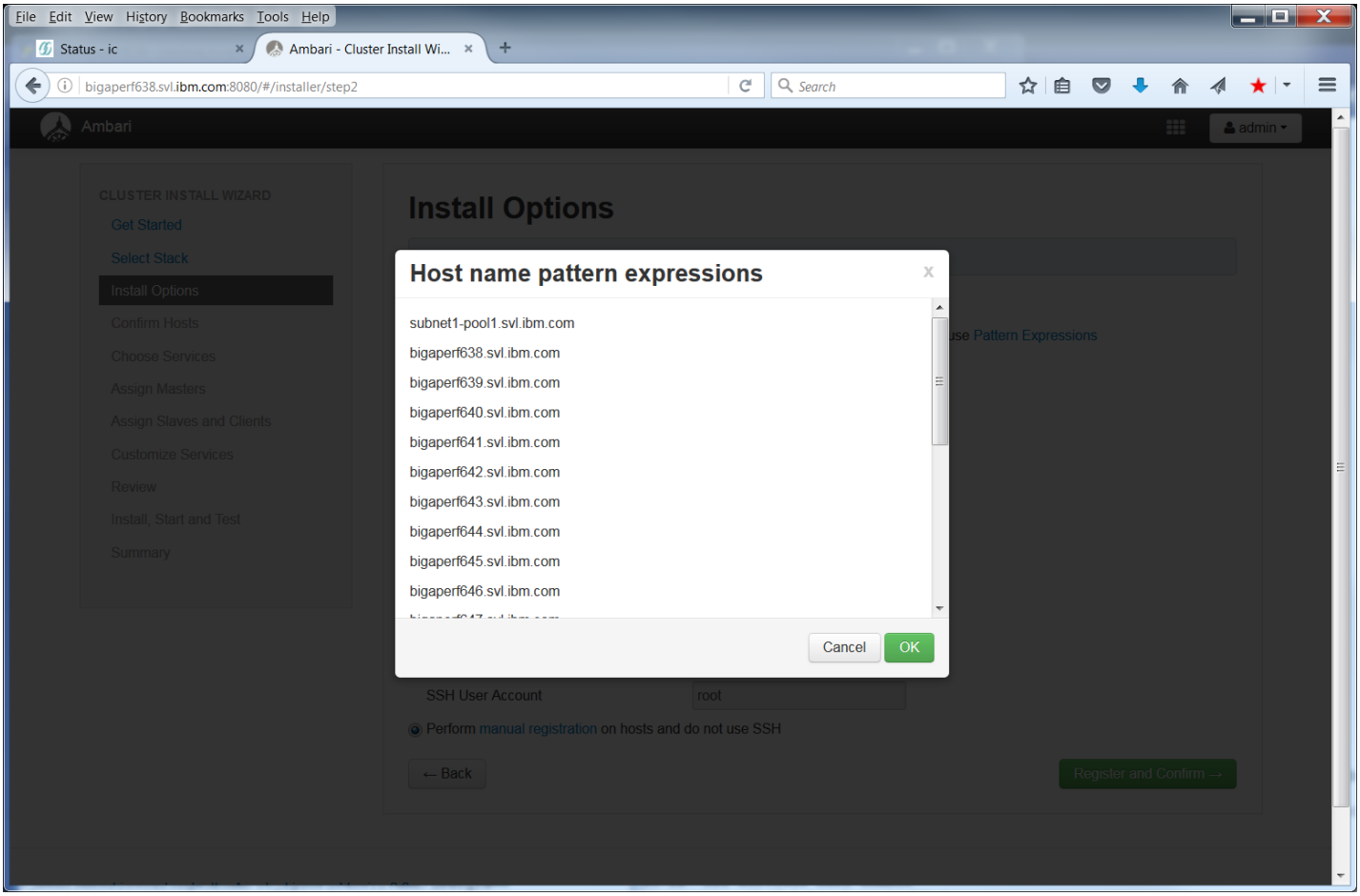
Isilon has an Ambari-agent within OneFS and needs to be manually registered in Ambari. After registering Isilon manually, click the Next button. You should see the Ambari agents on both your Linux hosts and Isilon become registered.

If you do not see all nodes listed in screenshot above, go back and add the missing nodes.



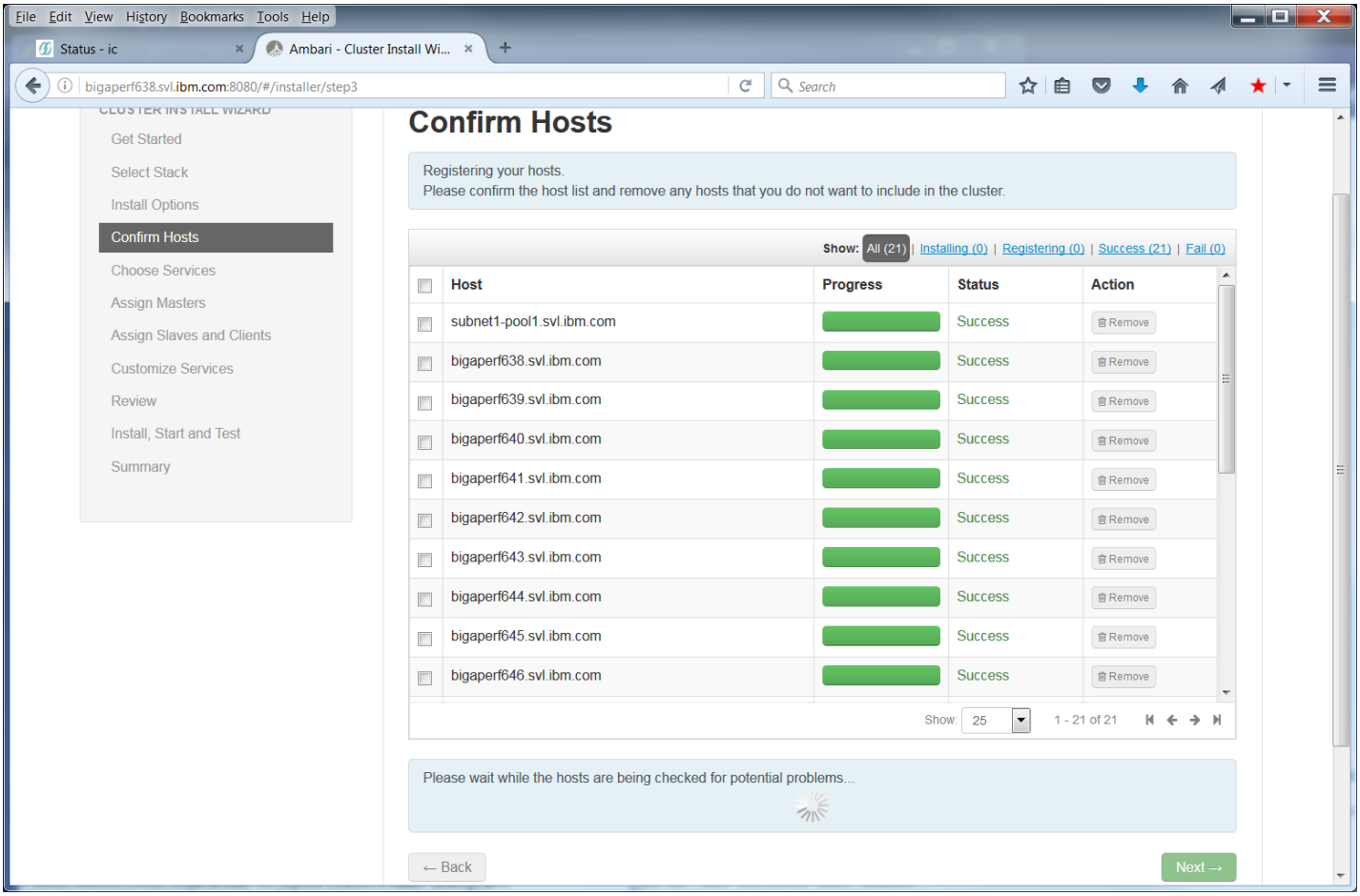
Click **Register and Confirm** using the manual registration again.

Now all nodes should be listed:



Click **OK** and confirm that Ambari agents have been installed on all nodes.

Finally, you should see all nodes in the **Confirm Hosts** screen as below:



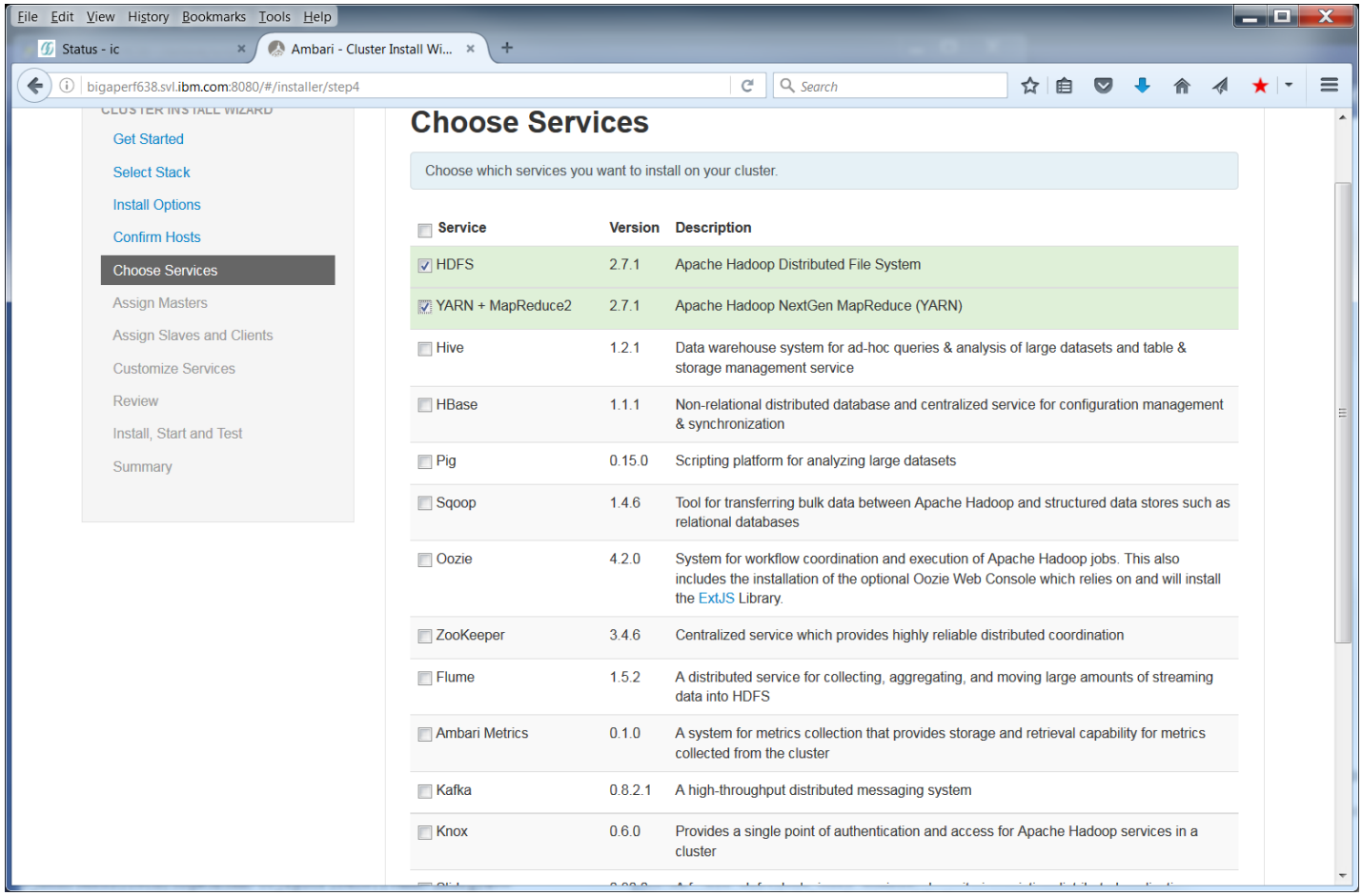
9. On the Confirm Hosts page, verify the host names, directories, and packages before proceeding with the installation.

   If hosts were selected in error, click the check boxes next to the hosts you want to remove. Click **Remove Selected**. To remove a single host, click **Remove** in the **Action** column.

   If warnings are found during the check process, you can **Click here to see the warnings** to see what caused the warnings. The Host Checks page identifies any issues with the hosts. For example, a host may have Transparent Huge Pages or Firewall issues. **You can ignore errors related to user names and groups as we pre-created the users in the pre-installation steps of this document**.

   After you resolve the issues, click **Rerun Checks** on the Host Checks page. When you have confirmed the hosts, click **Next**.

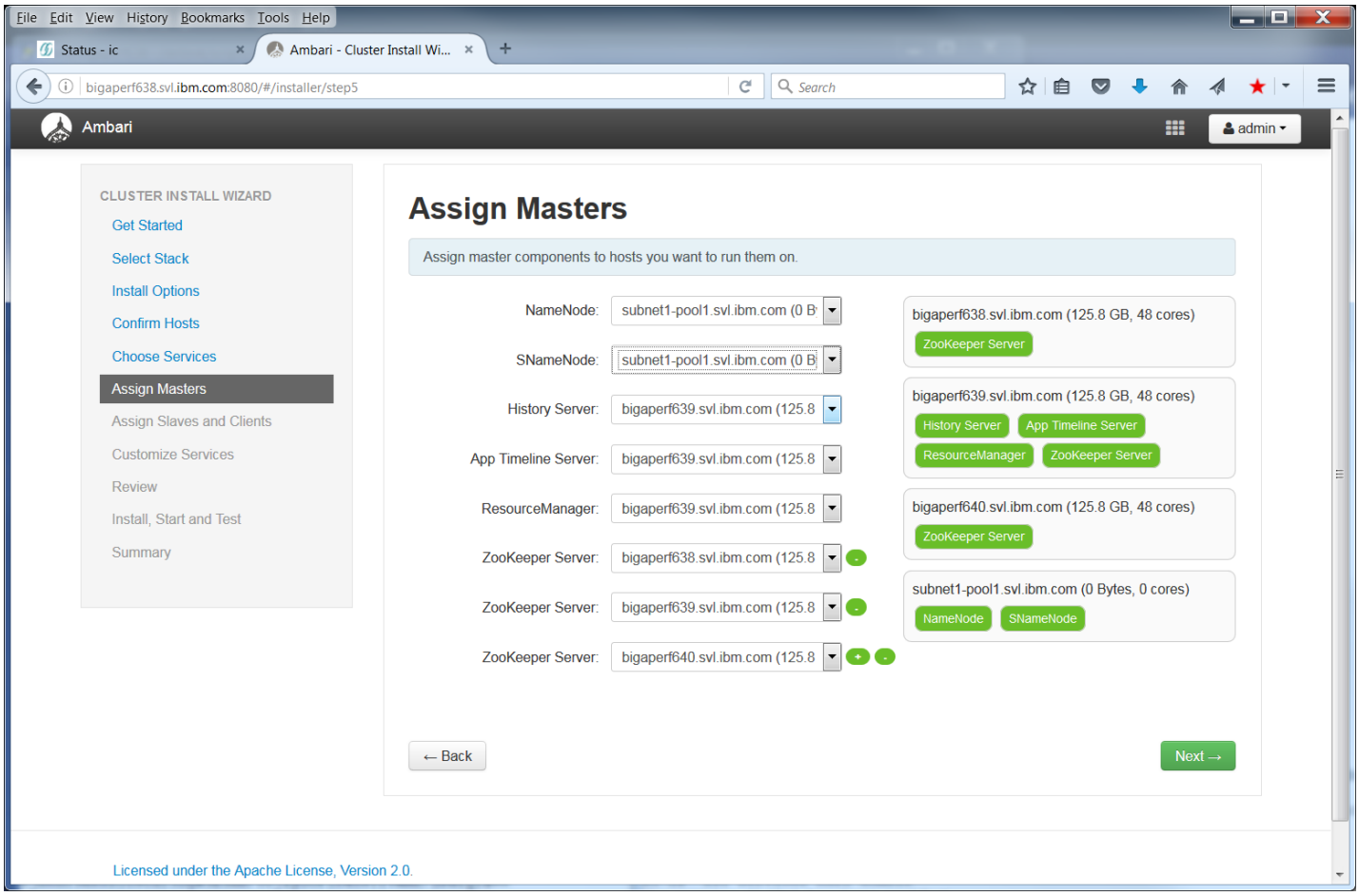
10. On the **Choose Services** page, select the services you want to install.



Zookeeper is required. Click **OK** to accept. In next screen click **Proceed** anyway to skip Ambari Metrics.

Ambari shows a confirmation message to install the required service dependencies. For example, when selecting Oozie only, the Ambari web interface shows messages for accepting YARN/MR2, HDFS and Zookeeper installations. It also shows Nagios and Ganglia for monitoring and alerting, but they are not required services.

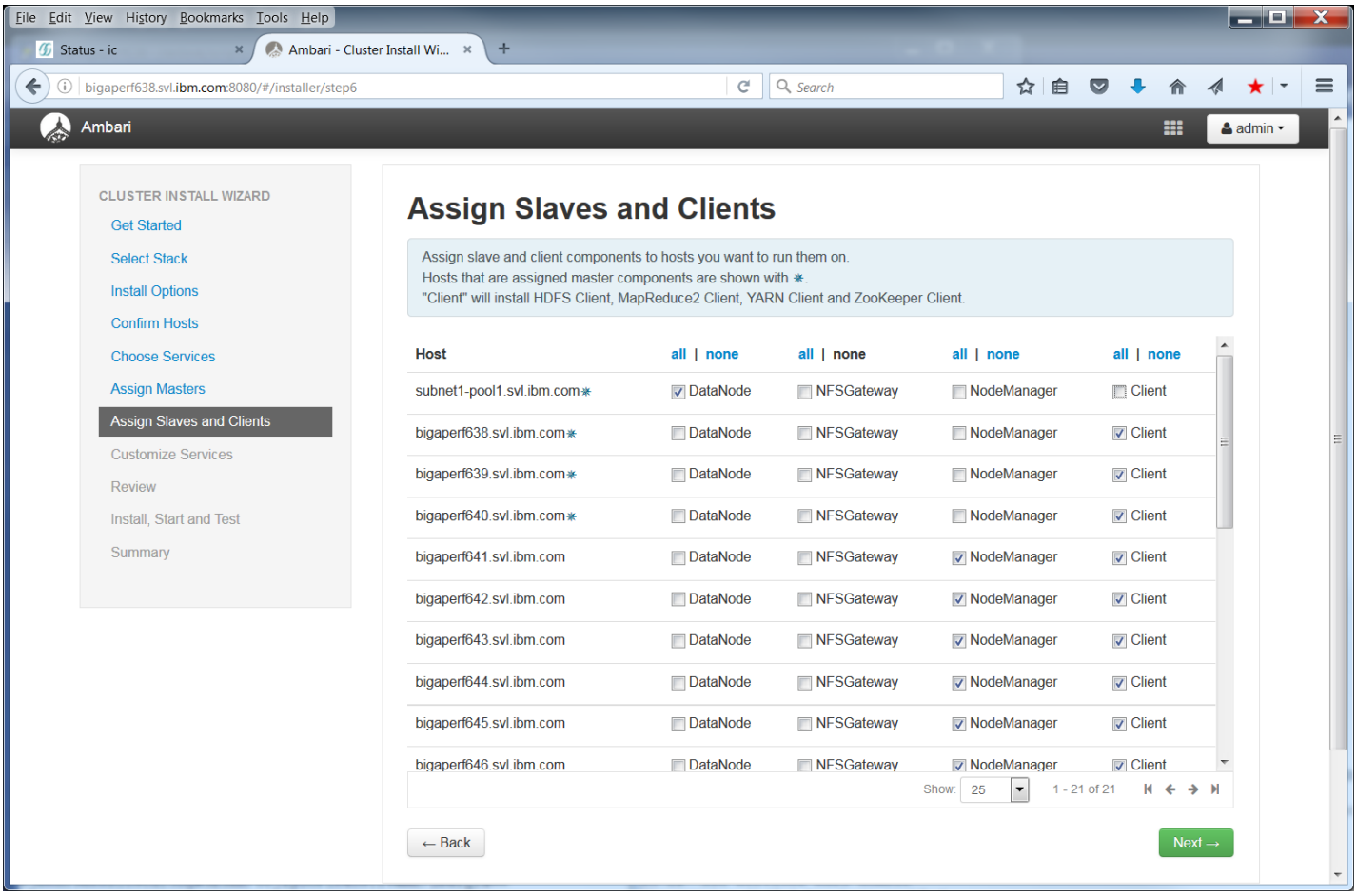
11. On the Assign Masters page, assign NameNode and SNameNode components to the Isilon SmartConnect address e.g. mycluster1-hdfs.example.com. The rest of the services can be deployed per the recommended services layout – refer to Table 1. **Make sure you assign Namenode and SNameNode only to the Isilon SmartConnect address and none of the Linux nodes**, e.g. only mycluster1-hdfs.example.com. **Notice**: The Zookeeper Service that selects the Isilon server should be deleted.



Click **Next**.

On the Assign Slaves and Clients page, assign the components to Linux hosts in your cluster and make sure **datanode is only assigned to Isilon**.

- Assign Client to the client nodes.
- Confirm that Isilon is checked for DataNode only:



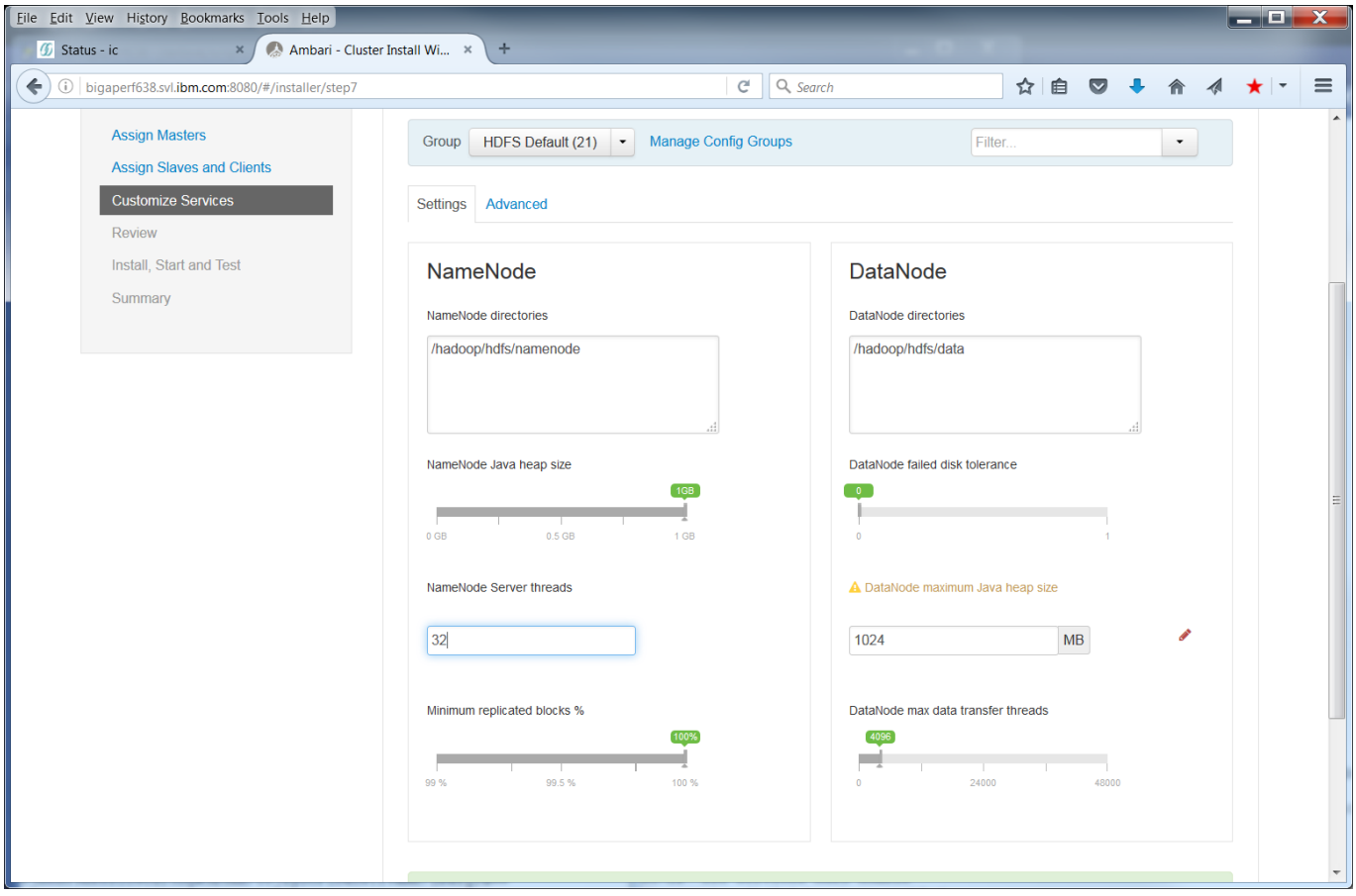
Click **Next**.

Tip: If you anticipate adding the Big SQL service at some later time, you must include all clients on all of the anticipated Big SQL worker nodes. Big SQL specifically needs the HDFS, Hive, HBase, Sqoop, HCat, and Oozie clients.

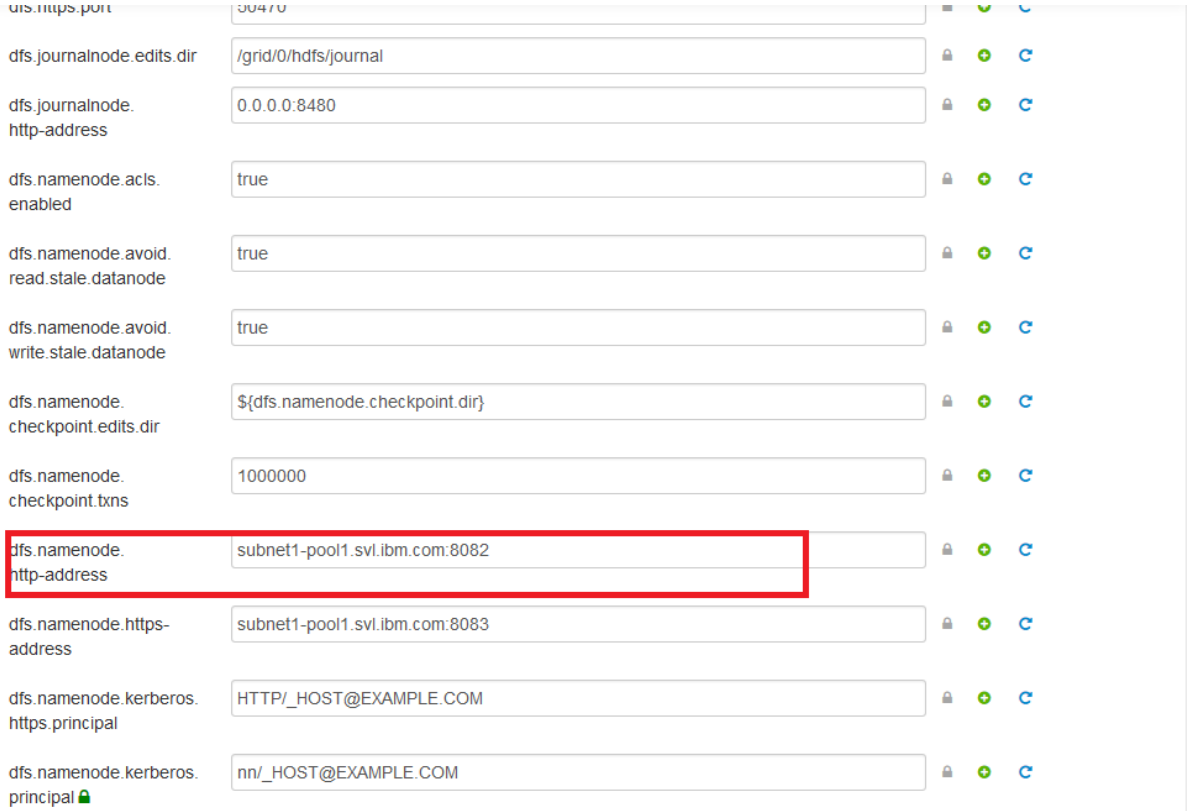
12. On the **Customize Services** page, select configuration settings for the services selected. Default values are filled in automatically when available and they are the recommended values. The installation wizard prompts you for required fields (such as password entries) by displaying a number in a circle next to an installed service.

    Assign passwords to Hive, Oozie, and any other selected services that require them. The following settings should be checked:

    - YARN Node Manager log-dirs
    - YARN Node Manager local-dirs
    - HBase local directory
    - ZooKeeper directory
    - HDFS: set the NameNode Server threads to 32:



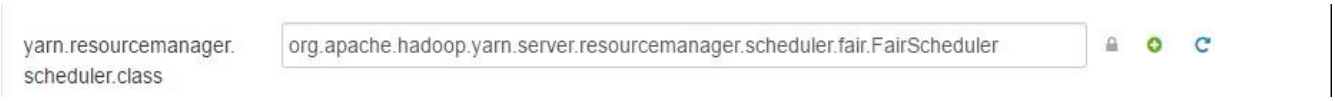
On the Advanced tab, change **dfs.namenode.http-address** port to 8082:



- Oozie Data Dir
- Storm storm.local.dir

Click the number and enter the requested information in the field outlined in red. Make sure that the service port that is set is not already used by another component. For example, the Knox gateway port is, by default, set as 8443. But, when the Ambari server is set up with HTTPs, and the SSL port is set up using 8443, then you must change the Knox gateway port to some other value.

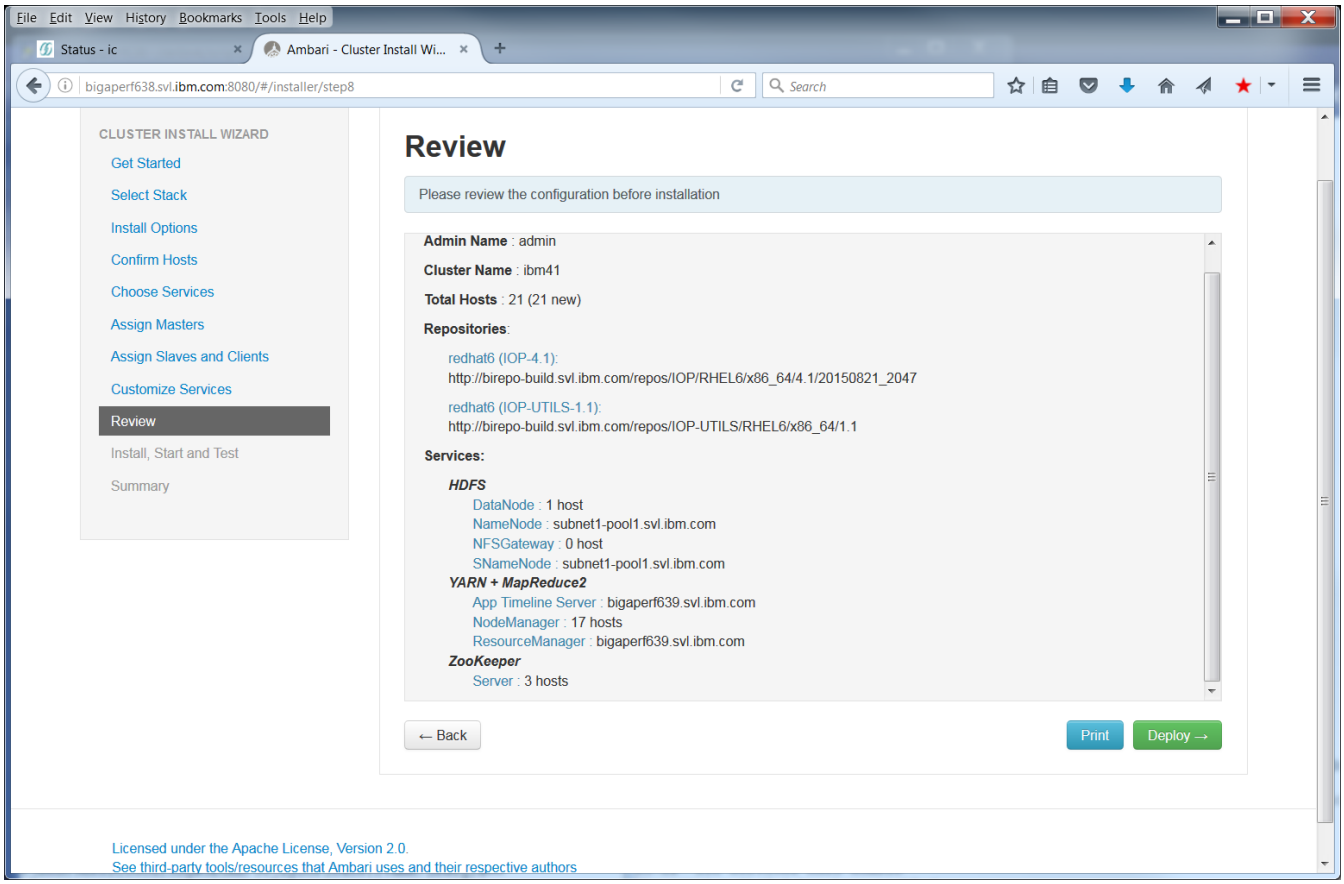
Important: Configure the correct Yarn resource manager scheduler. The parameter is shown below. This will improve MR jobs considerably when storage is shared.



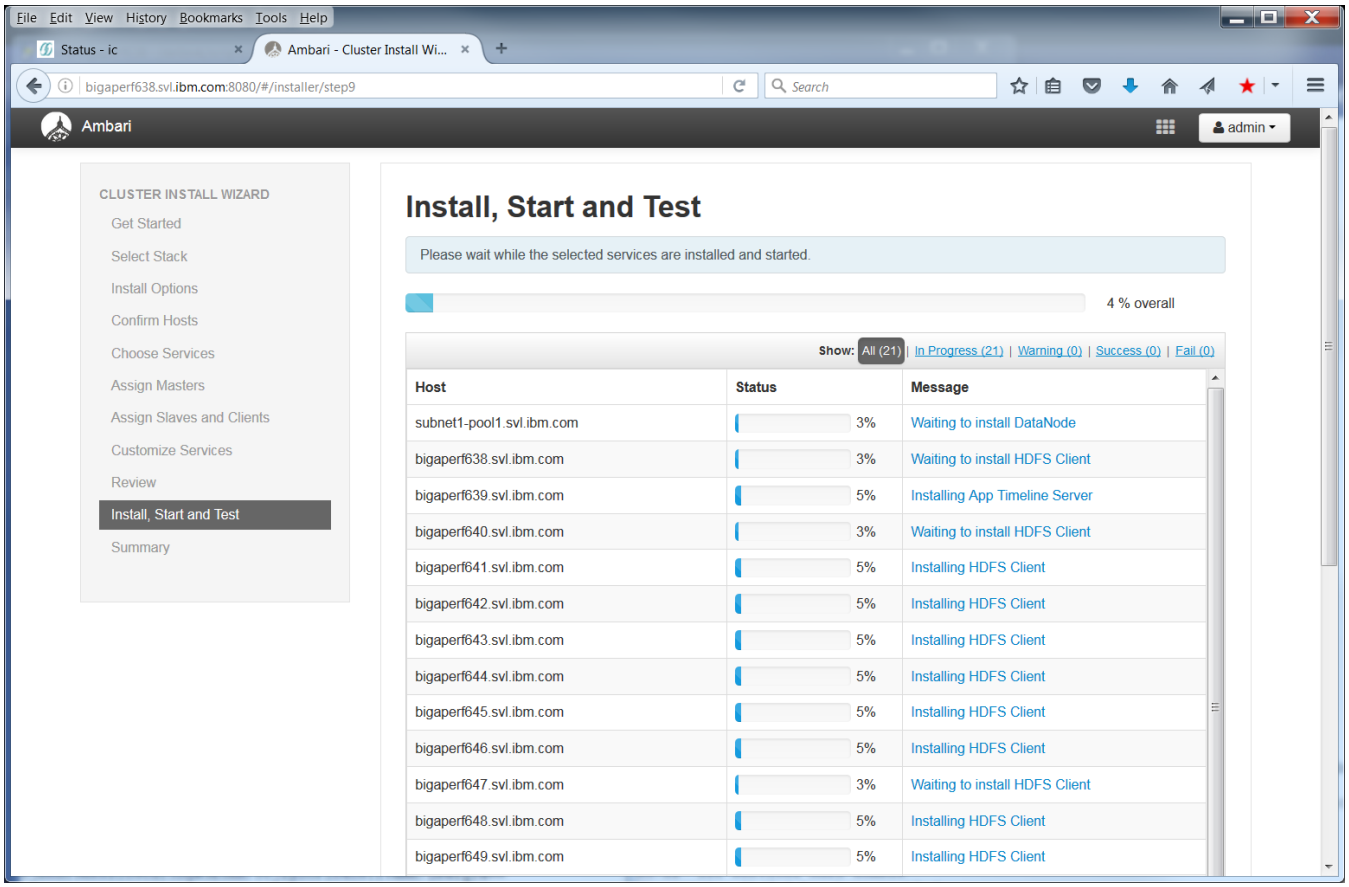
Note: If you are working in an LDAP environment where users are set up centrally by the LDAP administrator and therefore, already exist, selecting the defaults can cause the installation to fail. Open the **Misc** tab and check the box to ignore user modification errors.

13. When you have completed the configuration of the services, click **Next**.

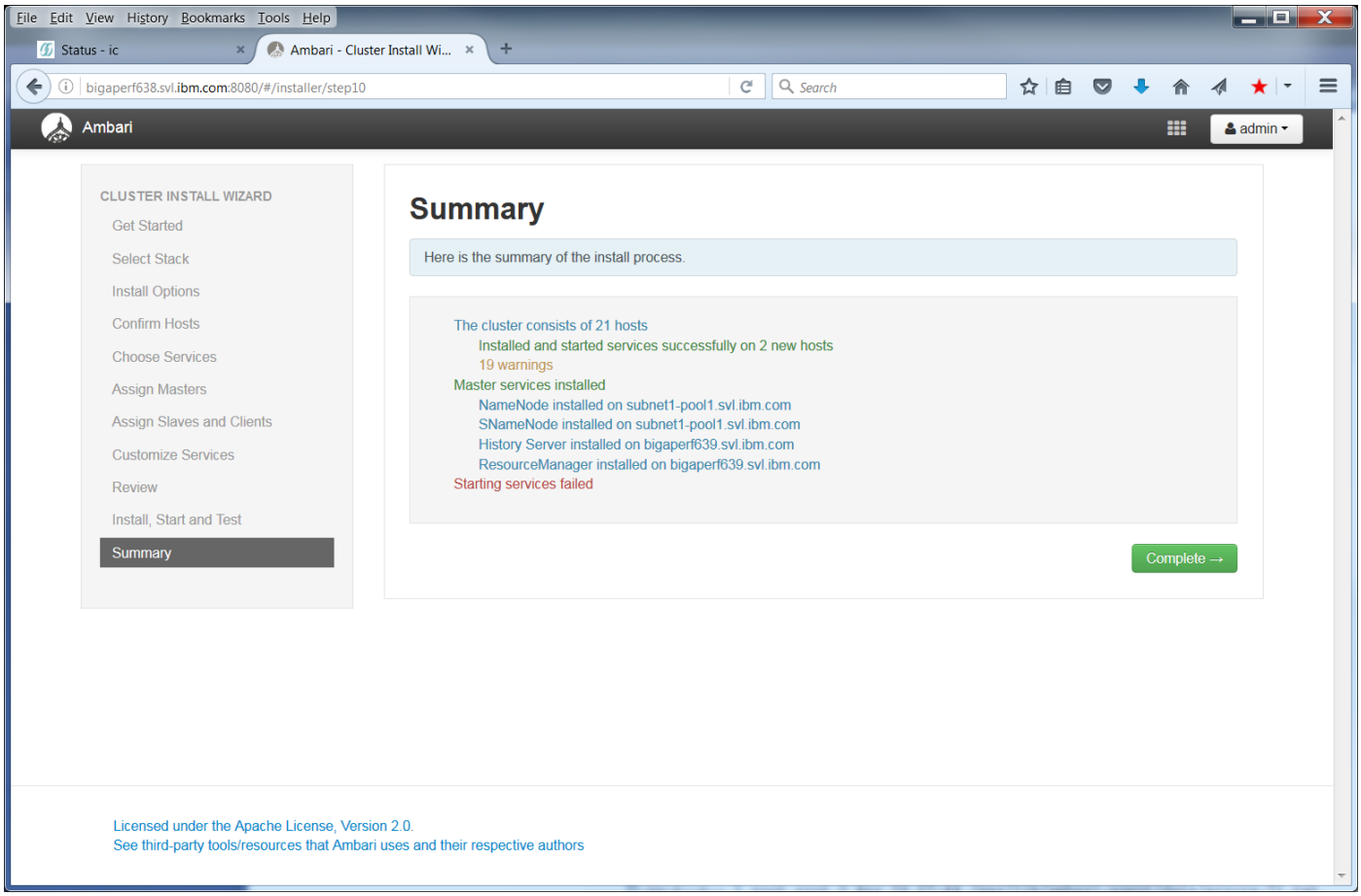
14. On the Review page, verify that your settings are correct.



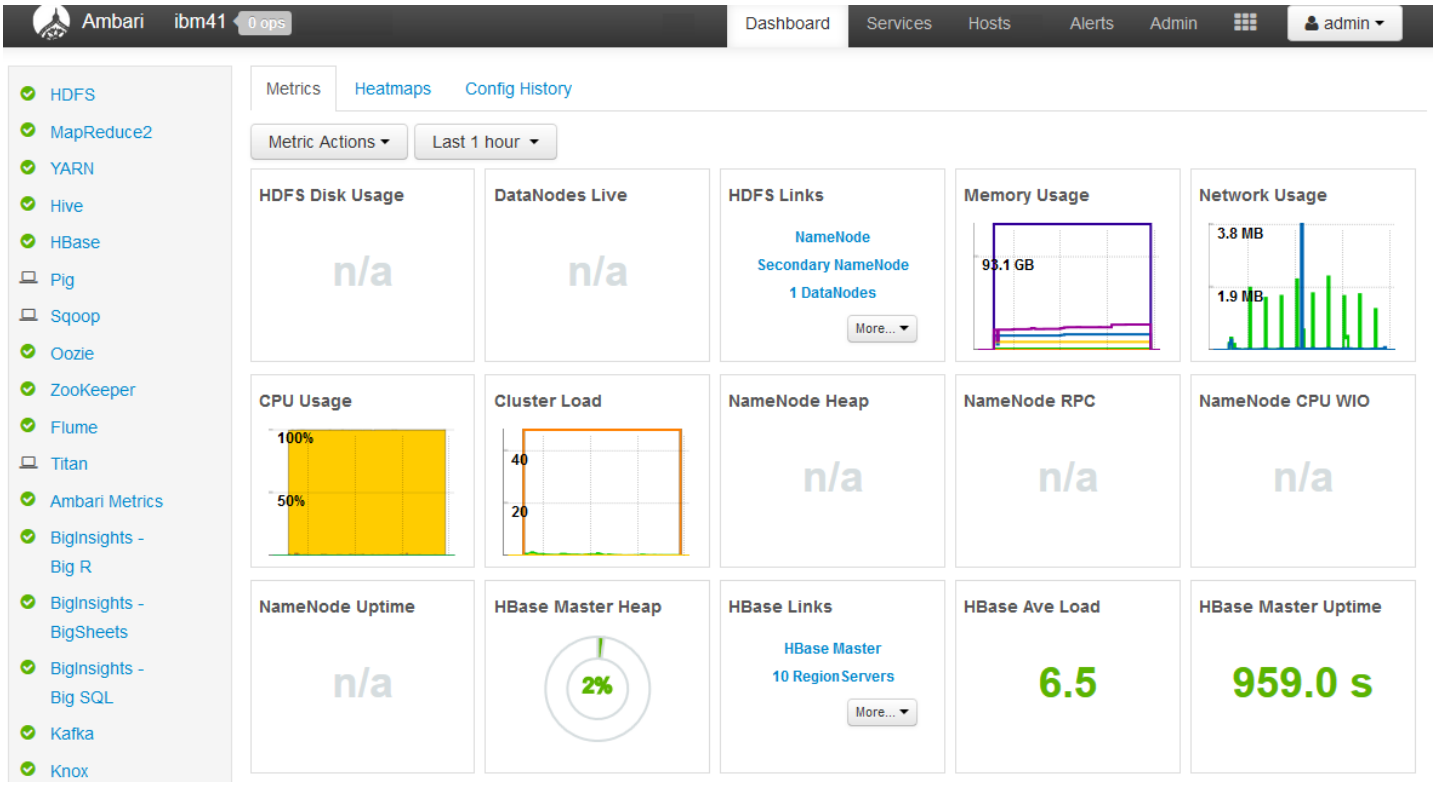
Click **Deploy**.



15. The **Install**, **Start**, and **Test** page shows the progress of the installation. The progress bar at the top of the page gives the overall status while the main section of the page gives the status for each host. Logs for a specific task can be displayed by clicking on the task. Click the link in the **Message** column to find out what tasks have been completed for a specific host or to see the warnings that have been encountered. When the message "Successfully installed and started the services" appears, click **Next**.



16. On the Summary page, review the accomplished tasks. Click **Complete** to go to the IBM Open Platform with Apache Hadoop dashboard.



2.7 Validating IBM Open Platform Installation

Ambari provides service checks for all the supported services. These checks run automatically after each service installation, or they can be run manually at any time. You can access the Ambari web interface and use the Services View to make sure all the components pass their checks successfully.

The following steps provide another way to validate your installation.

1. As the root user on a node on which Apache Hadoop is installed, enter the following command to become the ambari-qa user:
   `su - ambari-qa`
   Note: Do not use a management node for this validation. Terasort might fail.

2. As the ambari-qa user, run the following command:

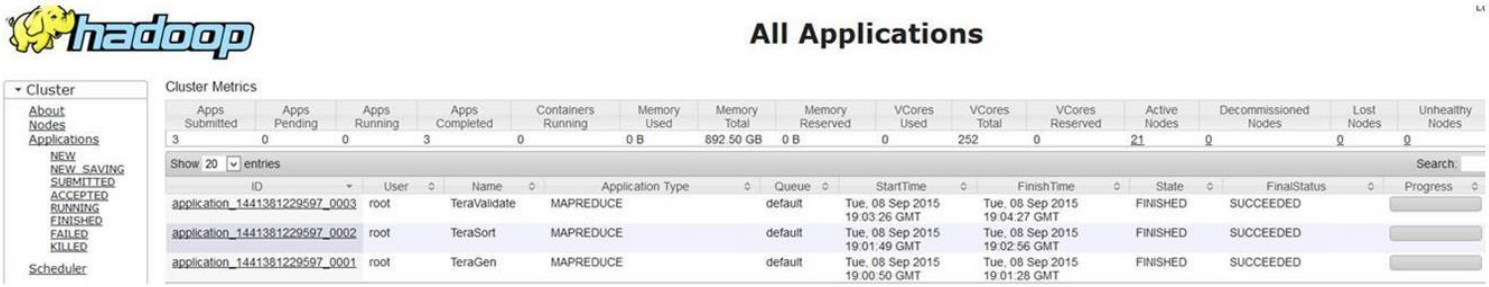
```
export HADOOP_MR_DIR=/usr/iop/current/hadoop-mapreduce-client
# Generate data with 1000 rows. Each row is about 100 bytes.
yarn jar $HADOOP_MR_DIR/hadoop-mapreduce-examples.jar teragen 1000 /tmp/tgout
# Sort data
yarn jar $HADOOP_MR_DIR/hadoop-mapreduce-examples.jar terasort /tmp/tgout
/tmp/tsout
# Validate data
yarn jar $HADOOP_MR_DIR/hadoop-mapreduce-examples.jar teravalidate /tmp/tsout
/tmp/tvout
```

If the job is successful, you will see a log record similar to the following:

```
INFO mapreduce Job: Job job_id completed successfully.
```

Browse to your cluster on port 8088 to see the results of your validation tests, for example http://x.x.x.x:8088/cluster.

Example YARN test results are shown below:



You are now set up for Isilon OneFS Hadoop and IBM Open Platform.

2.7.1 Ambari Service Check

Ambari has built-in functional tests for each component. These are executed automatically when you install your cluster with Ambari. To execute them after installation, select the service in Ambari, click the Service Actions button, and select Run Service Check.

2.8 Installing IBM Value Packages

2.8.1 Before You Begin

Note that “BigInsights Analyst” and “BigInsights Data Scientist” value packages have been sanity tested on Dell EMC Isilon but have not been performance profiled and tested under load with Isilon 8.0.0.x. Dell EMC and IBM BigInsights plan to validate these components under load as part of future integration efforts. Refer to Dell EMC – IBM BigInsights Joint Support Statement for further details.

The following sub-sections are dedicated to BigInsights 4.2. Please refer to the Appendix for the BigInsights 4.1 installation procedures before processing with section 2.8.3

2.8.2 BigInsights 4.2

Get the IBM.repo and put it to /etc/yum.repos.d directory and run the following command:

```
yum install BigInsights-IOP*
```

After the Biginsights module for 4.1 or 4.1 installed, then restart the ambary server.

1. When the module is installed, restart the Ambari server.

   ```
   ambari-server restart
   ```

2. Open the Ambari web interface and log in. The default address is the following URL:

   ```
   http://<server-name>:8080
   ```

The default login name is *admin* and the default password is *admin*.

3. Click **Actions** > **Add service**. In the list of services you will see the services that you previously added as well as the BigInsights services you can now add.

Example

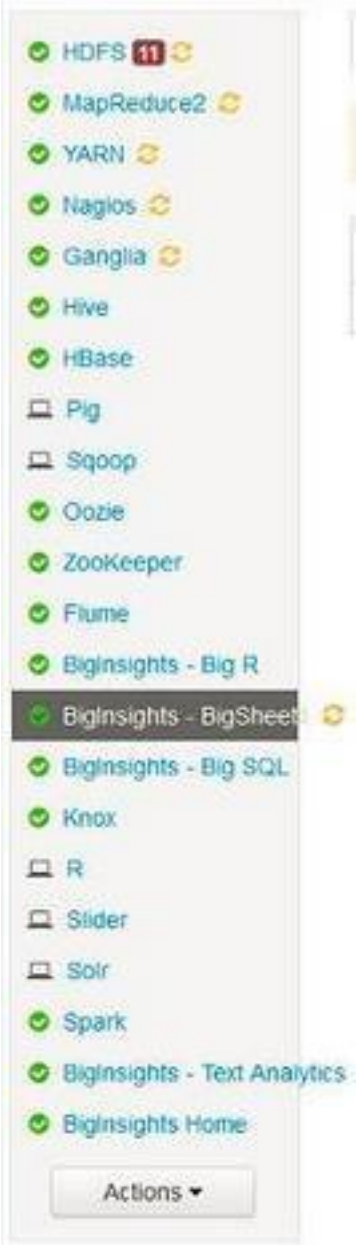
This is an example of running the IBM BigInsights Analyst module for OFFLINE installation:

```
[root@mgt1 ~]# chmod +x BI-Analyst-1.0.0.1-IOP-4.2.x86_64.bin
[root@mgt1 ~]# ./BI-Analyst-1.0.0.1-IOP-4.2.x86_64.bin
Creating directory ./BigInsights
Verifying archive integrity... All good.
Uncompressing IBM BigInsights Analyst Package Installer   100%
*******************************************************************
License Files
*******************************************************************
License files are available in the /root/BigInsights/licenses
Do you Accept the Terms and Conditions in the Licenses Directory ? (y/n) :
y
Will this be an ONLINE(1) or OFFLINE(2) Installation ?
2
*******************************************************************
Installing Package...
*******************************************************************
Downloading Package...
Installing Package...
BigInsights Analyst RPMs have been extracted to the BigInsights/packages
        Directory
Installation Complete
```

2.8.3 Select IBM BigInsights Service to Install

Select the service that you want to install and deploy. Even though your module might contain multiple services, install the specific service that you want and the BigInsights™ Home service. Installing one value-add service at a time is recommended. Follow the service specific installation instructions for more information.

After installing all the IBM BigInsights Services, the Ambari GUI Software List will have a □ next to each service as shown below:




2.8.4 Installing BigInsights Home

The BigInsights Home service is the main interface to launch BigInsights - BigSheets, BigInsights - Text Analytics, and BigInsights - Big SQL.

The BigInsights Home service requires Knox to be installed, configured and started.

Open a browser and access the Ambari server dashboard. The following is the default URL:

```
http://<server-name>:8080
```

The default user name is admin, and the default password is admin. In the Ambari dashboard, click Actions > Add Service.

In the **Add Service Wizard** > **Choose Services**, select the BigInsights – BigInsights Home service. Click **Next**. If you do not see the option for BigInsights – BigInsights Home, follow the instructions described in Installing the BigInsights value-add packages.

In the Assign Masters page, select a Management node (edge node) that your users can communicate with. BigInsights Home is a web application that your users must be able to open with a web browser.

In the Assign Slaves and Clients page, make selections to assign slaves and clients.

The nodes that you select will have JSQSH (an open source, command line interface to SQL for Big SQL and other database engines) and SFTP client. Select nodes that might be used to ingest data as an SFTP client, where you might want to work with Big SQL scripts, or other databases interactively.

Click **Next** to review any options that you might want to customize. Click Deploy.

If the BigInsights – BigInsights Home service fails to install, run the remove_value_add_services.sh cleanup script. The following code is an example command:

```
cd /usr/ibmpacks/bin/<version>

remove_value_add_services.sh

-uadmin -p admin

-x 8080 -s WEBUIFRAMEWORK -r
```

For more information about cleaning the value-add service environment, see Removing BigInsights value-add services.

After installation is complete, click **Next** > **Complete**.


2.8.5 Configure Knox

The Apache Knox gateway is a system that provides a single point of authentication and access for Apache Hadoop services on the compute nodes in a cluster; however authentication to HDFS services is completely controlled by Isilon OneFS only.

The Knox gateway simplifies Hadoop security for users that access the cluster and execute jobs and operators that control access and manage the cluster. The gateway runs as a server, or a cluster of servers, providing centralized access to one or more Hadoop clusters.

In IBM® Open Platform with Apache Hadoop, Knox is a service that you start, stop, and configure in the Ambari web interface.

Users access the following BigInsights™ value added components through Knox by going to the IBM BigInsights home service.

```
https://<knox_host>:<knox_port>/<knox_gateway_path>/default/BigInsightsWeb/index.html
```

- BigSheets
- Text Analytics
- Big SQL

Knox supports only REST API calls for the following Hadoop services:

- WebHCat
- Oozie
- HBase
- Hive
- Yarn

Click the **Knox** service from the Ambari web interface to see the summary page. Select **Service Actions** > **Restart All** to restart it and all of its components.

If you are using LDAP, you must also start LDAP if it is not already started. Click the **BigInsights Home** service in the Ambari User Interface.

Select **Service Actions** > **Restart All** to restart it and all of its components. Open the BigInsights Home page from a web.

The URL for BigInsights Home is:

```
https://<knox_host>:<knox_port>/<knox_gateway_path>/default/BigInsightsWeb/index.html
```

where:

knox_host

The host where Knox is installed and running

knox_port

The port where Knox is listening (by default this is 8443)

knox_gateway_path

The value entered in the gateway.path field in the Knox configuration (by default this is 'gateway')

For example, the URL might look like the following address:

```
https://bi_node1.example.com:8443/gateway/default/BigInsightsWeb/index.html
```

If you are using the Knox Demo LDAP, a default user ID and password is created for you. When you access the web page, use the following preset credentials:

```
User Name = guest
Password = guest-password
```

2.8.6 Installing Big SQL

To extend the power of the Open Platform for Apache Hadoop, install and deploy the BigInsights Big SQL service, which is the IBM SQL interface to the Hadoop-based platform, IBM Open Platform with Apache Hadoop.

1. Open a browser and access the Ambari server dashboard. The following is the default URL.

   ```
   https://<server-name>:8080
   ```

   The default user name is *admin*, and the default password is `admin`.

2. In the Ambari web interface, click **Actions** > **Add Service**.

3. In the **Add Service Wizard**, **Choose Services**, select the **BigInsights - Big SQL** service, and the **BigInsights Home** service. Click **Next**.

   If you do not see the option to select the **BigInsights - Big SQL** service, complete the steps.

4. In the **Assign Masters** page, decide which nodes of your cluster you want to run the specified components, or accept the default nodes. Follow these guidelines:

   a. For the Big SQL monitoring and editing tool, make sure that the Data Server Manager (DSM) is assigned to the same node that is assigned to the Big SQL Head node.

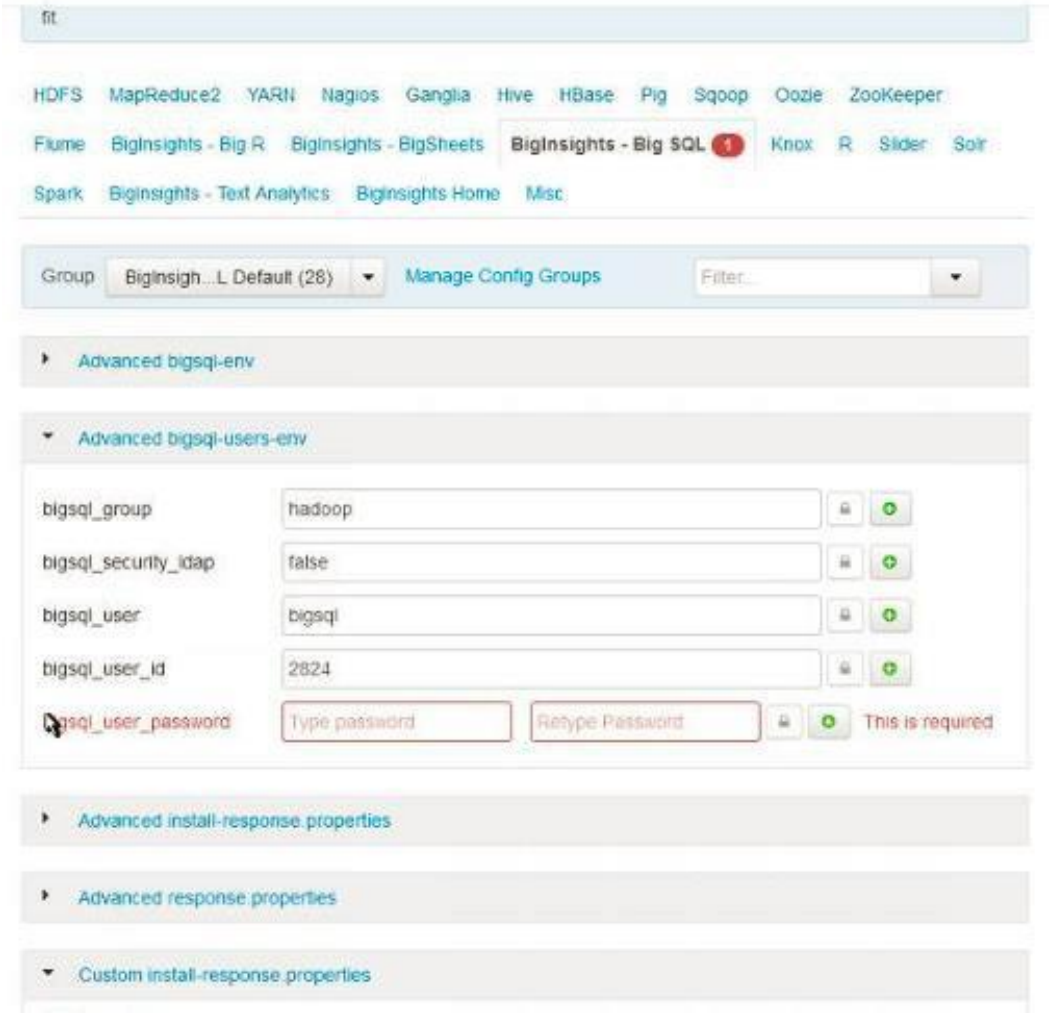
5. Click **Next**.

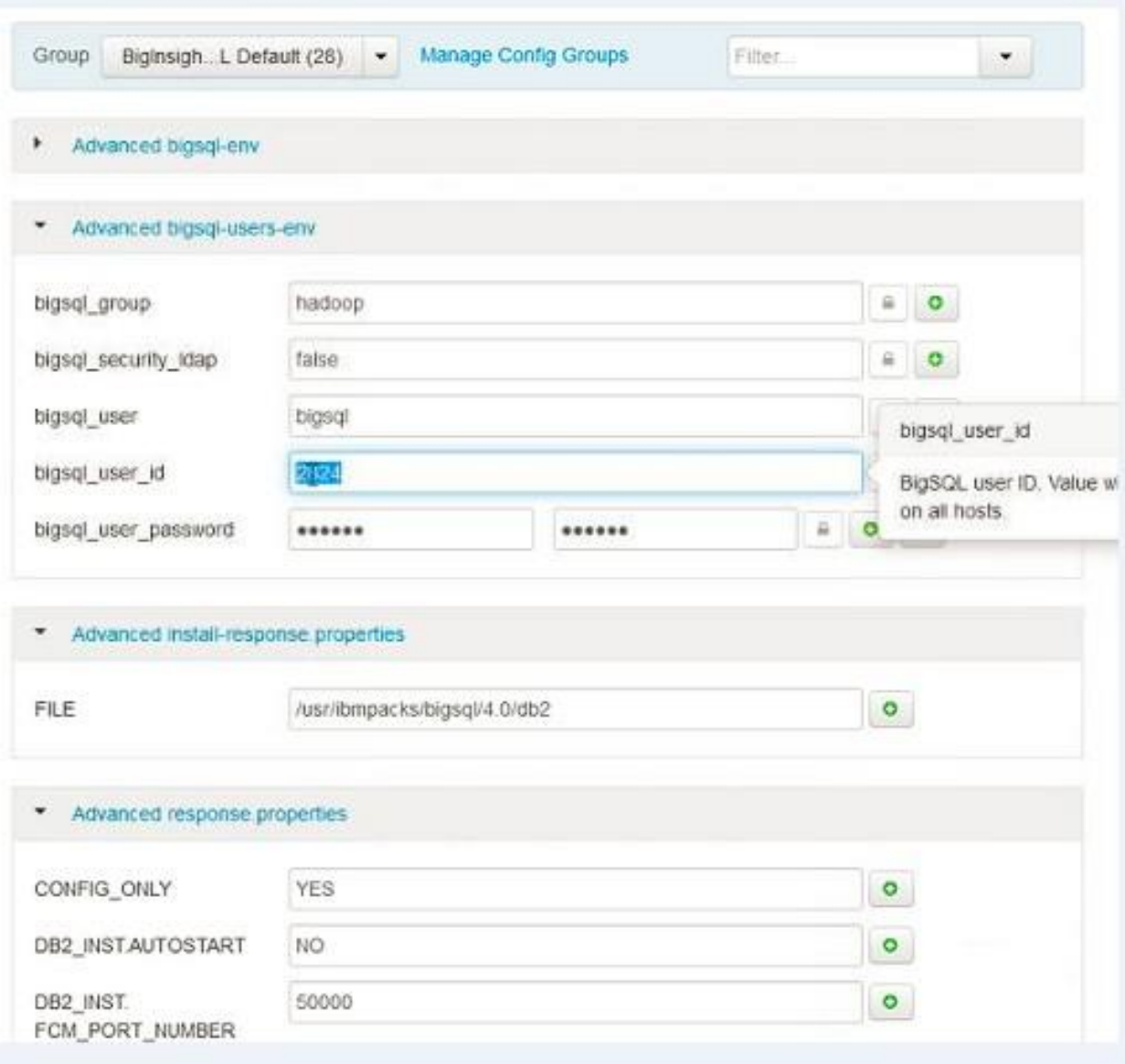
6. In the **Assign Slaves and Clients** page, accept the defaults, or make specific assignments for your nodes. Follow these guidelines:

    a. Select the non-head nodes for the Big SQL Worker components. You must select at least one node as the worker node.
    b. Select all nodes for the CLIENT. This puts JSqsh and SFTP clients on the nodes.

Notice: Ensure the ssh passwordless configuration among worker nodes, head node, and Isilon machines.

7. In the **Customize Services** page, accept the recommended configurations for the Big SQL service, or customize the configuration by expanding the configuration files and modifying the values. Make sure that you have a valid **bigsql_user** and **bigsql_user_password** (see reference screen below) and **user_id (**created by the **bi_create_users.sh script**) in the appropriate fields in the**Advanced bigsql-users-env** section.





8. You can review your selections in the **Review** page before accepting them. If you want to modify any values, click the **Back** button. If you are satisfied with your setup, click **Deploy**.

9. In the **Install, Start and Test** page, the Big SQL service is installed and verified. If you have multiple nodes, you can see the progress on each node. When the installation is complete, either view the errors or warnings by clicking the link, or click **Next** to see a summary and then the new service added to the list of services.

If the **BigInsights – Big SQL** service fails to install, run the **remove_value_add_services.sh** cleanup script. The following code is an example of the command:

```
cd /usr/ibmpacks/bin/<version>

./remove_value_add_services.sh -u admin -p admin -x 8080 -s BIGSQL -r
```

For more information about cleaning the value-add service environment, see Removing BigInsights value-add services.

10. A web application interface for Big SQL monitoring and editing is available to your end- users to work with Big SQL. You access this monitoring utility from the IBM BigInsights Home service. If you have not added the BigInsights Home service yet, do that now.

11. Restart the **Knox** Service as well as the Knox Demo LDAP service, if you have not configured your own LDAP.

12. Restart the BigInsights Home services.

13. To run SQL statements from the Big SQL monitoring and editing tool, type the following address in your browser to open the BigInsights Home service:

```
https://<knox_host>:<knox_port>/<knox_gateway_path>/default/BigInsightsWeb/index.html
```

Where:

knox_host

The host where Knox is installed and running

knox_port

The port where Knox is listening (by default this is 8443)

knox_gateway_path

The value entered in the **gateway.path** field in the Knox configuration (by default this is 'gateway')

For example, the URL might look like the following address:

```
https://bi_node1.example.com:8443/gateway/default/BigInsightsWeb/index.html
```

If you use the Knox Demo LDAP service, the default credential is:

```
userid = guest
password = guest-password
```

Your end users can also use the JSqsh client, which is a component of the **BigInsights - Big SQL** service.

14. If the BigInsights - Big SQL service shows as unavailable, there might have been a problem with post-installation configuration. Run the following commands as **root** (or **sudo**) where the Big SQL monitoring utility (DSM) server is installed:

    a. Run the dsmKnoxSetup script:

    ```
    cd /usr/ibmpacks/bigsql/<version-number>/dsm/1.1/ibm-datasrvrmgr/bin/
    ./dsmKnoxSetup.sh -knoxHost <knox-host>
    ```

    where <*knox-host*> is the node where the Knox gateway service is running.

    b. Make sure that you do not stop and restart the Knox gateway service within Ambari. If you do, then run the **dsmKnoxSetup** script again.

    c. Restart the **BigInsights Home** service so that the Big SQL monitoring utility (DSM) can be accessed from the **BigInsights Home** interface.

15. For HBase, do the following post-installation steps:

    a. For all nodes where HBase is installed, check that the symlinks to hive-serde.jar and hive-common.jar in the hbase/lib directory are valid.

    b. To verify the symlinks are created and valid:

    ```
    namei /usr/iop/<version-number>/hbase/lib/hive-serde.jar
    namei /usr/iop/<version-number>/hbase/lib/hive-common.jar
    ```

    c. If they are not valid, use the following commands:

    ```
    cd /usr/iop/<version-number>/hbase/lib
    rm -rf hive-serde.jar
    rm -rf hive-common.jar
    ln -s /usr/iop/<version-number>/hive/lib/hive-serde.jar hive-serde.jar
    ln -s /usr/iop/<version-number>/hive/lib/hive-common.jar hive-common.jar
    usr/ibmpacks/bigsql/<version-number>/dsm/1.1/ibm-datasrvrmgr/bin/
    ./dsmKnoxSetup.sh -knoxHost <knox-host>
    ```

After installing the Big SQL service, and fixing the symlinks, restart the HBase service from the Ambari web interface.

After you add Big SQL worker nodes, make sure that you stop and then restart the Hive service.

When you run service check for bigsql failed on drop table, please check the workaround "HDFS Caching"

If there is an error message about "java.lang.IllegalArgumentException: /user/bigsql/sync is not a directory." during create hadoop table please refer to the URL:

http://www.ibm.com/support/knowledgecenter/SSPT3X_4.2.0/com.ibm.swg.im.infosphere.biginsights.trb.doc/doc/trb_bsql_hiveautocatsync1.html

2.8.6.1 Connecting to Big SQL

You can run Big SQL queries from Java SQL Shell (JSqsh), or from the IBM Data Server Manager. You can also run queries from a client application, such as IBM Data Studio, that uses JDBC or ODBC drivers. You must identify a running Big SQL server and configure either a JDBC or ODBC driver.

For more information about JSqsh, or IBM Data Studio, see the related topics in the IBM® BigInsights™ Knowledge Center.

2.8.6.2 Running JSqsh

JSqsh is installed in /usr/ibmpacks/common-utils/current/jsqsh/bin. Change to that directory and type./jsqsh to open the JSqsh shell:

```
cd /usr/ibmpacks/common-utils/current/jsqsh/bin
./jsqshln -s /usr/iop/<version-number
```

You can then run any JSqsh commands from the prompt.

Connection setup

To use the JSqsh command shell, you can use the default connections or define and test a connection to the Big SQL server.

1. The first time that you open the JSqsh command shell, a configuration wizard is started. When you are at the Jsqsh command prompt, type **\drivers** to determine the available drivers.
    a. On the driver selection screen, select the Big SQL instance that you want to run

    Note: Big SQL is designated as *DB2* in this example:

```
Name          Target        Class
- ------- ------------------- ---------------------------------------------
...
2 *db2 IBM Data Server(DB2 com.ibm.db2.jcc.DB2Driver
```

b. Verify the port, server, and user name. Run **\setup** and click **C** to define a password for the connection. The *username* must have database administration privileges, or must be granted those privileges by the Big SQL administrator.

c. Test the connection to the Big SQL server.

d. Save and name this connection.

2. Generally, you can access JSqsh from `/usr/ibmpacks/common- utils/current/jsqsh/bin` with the following command:

```
./jsqsh --driver=db2 --user=<username>
--password=<user_password>
```

3. Open the saved configuration wizard any time by typing \setup while in the command interface, or./jsqsh --setup when you open the command interface.

4. Specify the following connection name in the JSqsh command shell to establish a connection:

```
./jsqsh name
```

5. Use the \connect command when you are already inside the JSQSH shell to establish a connection at the JSqsh prompt:

```
\connect name
```

2.8.6.3 Commands and queries

At the JSqsh command prompt, you can run JSqsh commands or database server commands. JSqsh commands usually begin with a backslash (****) character.

JSqsh commands accept command-line arguments and allow for common shell activities, such as I/O redirection and pipes.

For example, consider this set of commands:

```
1> select * from t1 2>

   where c1 > 10

3> \go --style csv > /tmp/t1.csv
```

Because the commands do not begin with a backslash character, the first two commands are assumed to be SQL statements, and are sent to the Big SQL server.

The **\go** command sends the statements to run on the server. The **\go** command has a built-in alias so that you can omit the backslash. Additionally, you can specify a trailing semicolon to indicate that you want to run a statement, for example:

```
1> select * from t1 2>
where c1 > 10;
```

The --style option in the **\go** command indicates that the display shows comma-separated values (CSV). The **\go** form is most useful if you provide additional arguments to affect how the query is run. Changing the display style is an example of this feature.

The redirection operator (**>**) specifies that the results of the command are sent to a file called /tmp/t1.csv.

A set of frequently run commands does not require the leading backslash. Any JSqsh command can bealiased to another name (without a leading backslash, if you choose), by using the **\alias** command. For example, if you want to be able to type *bye* to leave the JSqsh shell, you establish that word as the alias for the **\quit** command:

```
\alias
bye='\quit'
```

You can run a script that contains one or more SQL statements. For example, assume that you have a file called mySQL.sql. That file contains these statements:

```
select tabschema, tabname from syscat.tables fetch first 5 rows only;
select tabschema, colname, colno, typename, length from syscat.columns fetch first 10 rows only;
```

You can start JSqsh and run the script at the same time with this command:

```
/usr/ibmpacks/common-utils/current/jsqsh/bin/jsqsh bigsql < /home/bigsql/mySQL.sql
```

The redirection operator specifies to JSqsh to get the commands from the file located in the `/home/bigsqldirectory`, and then run the statements within the file.

2.8.6.4 Command and query edit

The JSqsh command shell uses the JLine2 library, which allows you to edit previously entered commands and queries. You use the command-line edit features to move the arrow keys and to edit the command or query on the current line.

The JLine2 library provides the same key bindings (**vi** and **emacs**) as the GNU Readline library. In addition, it attempts to apply any custom key maps that you created in a GNU Readline configuration file, (.inputrc) in the local file system $HOME/ directory.

In addition to individual line editing, the JSqsh command shell remembers the 50 most recently run statements, which you can view by using the **\history** command:

```
1> /history
   (1) use tph;
   (2) select count(*) from lineitem
```

Previously run statements are prefixed with a number in parentheses. You use this number to recall that query by using the JSqsh recall operator (!), for example:

```
1> !2
```

```
1> select count(*) from lineitem 2>
```

The "!" recall operator has the following behavior:

!! Recalls the previously run statement.

!5 Recalls the fifth query from history.

!-2 Recalls the query from two prior runs.

You can also edit queries that span multiple lines by using the **\buf-edit** command, which pulls the current query into an external editor, for example:

```
1> select id, count(*) 2> from t1, t2
3> where t1.c1 = t2.c2 4> \buf-edit
```

The query is opened in an external editor (/usr/bin/vi by default. However, you can specify a different editor on the environment variable **$EDITOR**). When you close the editor, the edited query is entered at the JSqsh command shell prompt.

The JSqsh command shell provides built-in aliases, **vi** and **emacs**, for the **\buf- edit** command. The following commands, for example, open the query in the vi editor:

```
1> select id, count(*) 2>
from t1, t2
3> where t1.c1 = t2.c2 4> vi
```

2.8.6.5 Configuration variables

You can use the **\set** command to list or define values for several configuration variables, for example:

```
1> \set
```

If you want to redefine the prompt in the command shell, you run the following command with the prompt option:

```
1> \set prompt='foo $lineno> ' foo 1>
```

Every JSqsh configuration variable has built-in help available:

```
1> \help prompt
```

If you want to permanently set a specific variable, you can do so by editing your $HOME/.jsqsh/sqshrc file and including the appropriate **\set** command in it.

2.8.7 Installing Text Analytics

The Text Analytics service provides powerful text extraction capabilities. You can extract structured information from unstructured and semi-structured text.

It is recommended that you make sure that the python-paramiko package is installed prior to installing the Text Analytics service.

```
yum install python-paramiko
```

You must select a Master node for Text Analytics, and this node should contain the **python-paramiko** package. The master node is the node where Text Analytics Web Tooling and Text Analytics Runtime are both installed.

1. Open a browser and access the Ambari server dashboard. The following is the default URL.

   ```
   http://<server-name>:8080
   ```

   The default user name is *admin*, and the default password is *admin*.

2. In the Ambari dashboard, click **Actions** > **Add Service**.

3. In the Add Service Wizard, **Choose Services**, select the **BigInsights - Text Analytics** service.
   If you do not see the option to select the **BigInsights - Text Analytics** service, complete the steps in Installing the BigInsights value-add packages.

4. To assign master nodes, select the **Text Analytics Master Server Node**.

5. Click **Next**. The Assign Slaves and Clients page displays.

6. Assign slave and client components to the hosts on which you want them to run. An asterisk (*) after a host name indicates the host is assigned a master component.

   - To assign slaves nodes and clients, click **All** on the Clients column.

     The client package that is installed contains runtime binaries that are needed to run Text Analytics. This client needs to be installed on all datanodes that belong to your cluster.

     Client nodes will install only the Text Analytics Runtime artifacts. (/usr/ibmpacks/current/text-analytics-runtime). Choose one or more clients. You do not have to choose the Master node as a client since it already installs Text Analytics Runtime.

7. Click **Next** and select **BigInsights - Text Analytics**.

8. Expand **Advanced ta-database-config** and enter the password in the database.password field.Recommended configurations for the service are completed automatically but you can edit these default settings as desired. By default, the database server is MySQL. There are two options:
   - **database.create.new** = Yes (default)
     i. You must enter the password for the database.

ii. You must ensure that the default port, 32050 is free. You can change the port to any free port.
iii. You can change the **database.username**, but any changes to the **database.hostname** are ignored.

- **database.create.new** = N
  i. You must enter the database.hostname, database.port (where the existing database server instance is running) , database.user and database.password. Ensure that the user and password have full access to create a database in the existing database server instance you specify. Especially if it is a remote MySQL server instance, ensure that all permissions are given to the user and password to access this remote instance. Ensure that the server instance is up and running so that the Text Analytics service can be started successfully.

9. Click **Next** and in the Review screen that opens, click **Deploy**.

10. After installation is complete, click **Next** > **Complete**.

11. After the installation is successful, click **Next** and **Complete**.

    If the BigInsights - Text Analytics service fails to install, run the remove_value_add_services.shcleanup script. The following code is an example command:

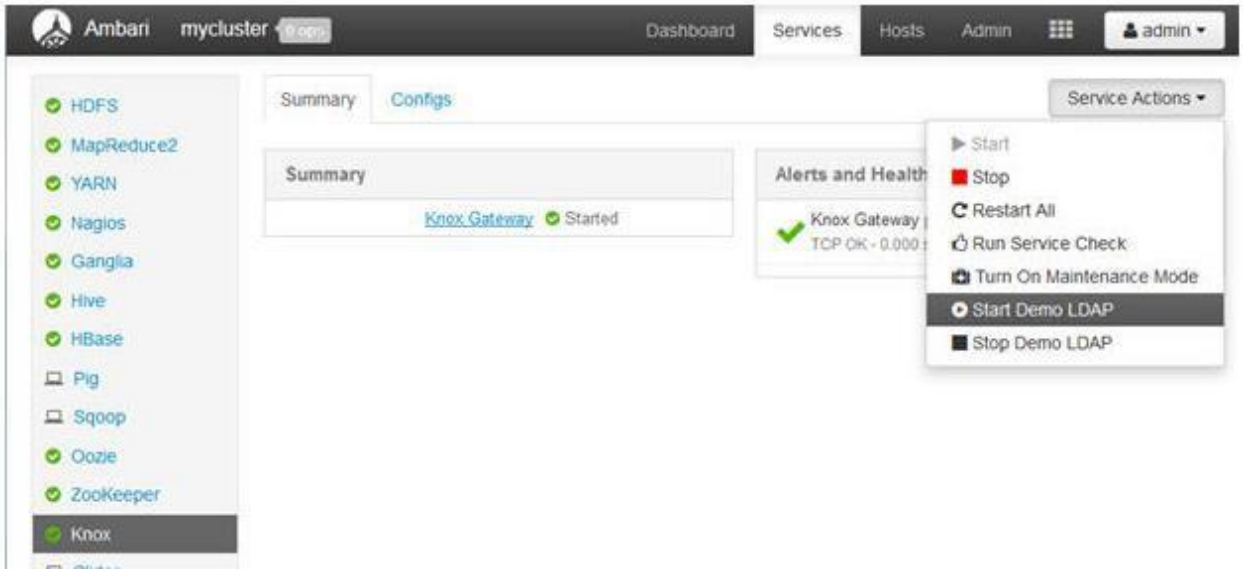
```
cd /usr/ibmpacks/bin/<version> remove_value_add_services.sh
-u admin -p admin
-x 8080 -s TEXTANALYTICS -r
```

    For more information about cleaning the value-add service environment, see Removing BigInsights value-add services.

12. The Text Analytics directory on all nodes where Text Analytics components are installed is created with world-writable permissions, which are not required. Change the permissions to **rwxr-x-r-x** on all nodes to improve security:

```
chmod go-w /usr/ibmpacks/text-analytics-runtime
```

13. Restart the **Knox** service. If you have not configured LDAP service, start the Knox Demo LDAP service.



Open the BigInsights Home and launch Text Analytics at the following address:

```
https://<knox_host>:<knox_port>/<knox_gateway_path>/default/BigInsightsWeb/index.html
```

Where:

knox_host

The host where Knox is installed and running

knox_port

The port where Knox is listening (by default this is 8443)

knox_gateway_path

The value entered in the **gateway.path** field in the Knox configuration (by default this is 'gateway')

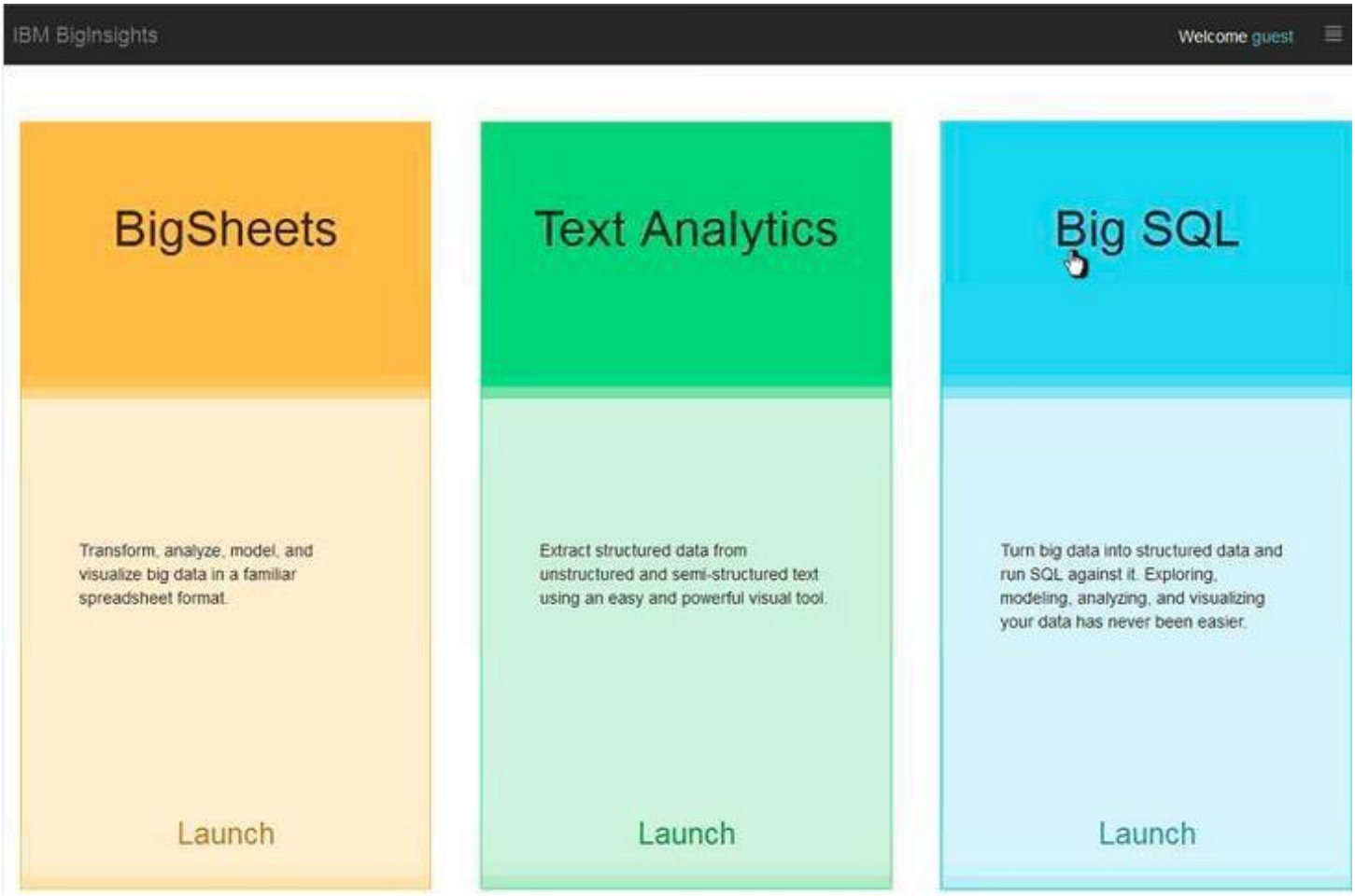
```
https://bi_node1.example.com:8443/gateway/default/BigInsightsWeb/index.html
```

If you use the Knox Demo LDAP service and have not modified the default configuration, the default credential to log into the BigInsights - Home service is:

```
userid = guest
password = guest-password
```

Note: If you do not see the Text Analytics service from BigInsights Home, restart the BigInsights Home service in the Ambari interface.

At this point, IBM BigInsights Home should show all three BigInsights Services as shown:



2.8.8 Installing Big R

To extend the power of the Open Platform for Apache Hadoop, install and deploy the Big R service, which is the IBM R extension, to the Hadoop-based platform, IBM Open Platform with Apache Hadoop.

1. Open a browser and access the Ambari server dashboard. The following is the default URL.

   ```
   http://<server-name>:8080
   ```

   The default user name is *admin*, and the default password is *admin*.

2. In the Ambari web interface, click **Actions** > **Add Service**.

3. **Optional**: If you do not already have the R Service installed, you can add it now. Big R service depends on the R statistics environment and the following three R packages: base64enc, rJava and data.table. If these have been installed on all nodes in the cluster, this step can be skipped. Otherwise, you can choose to install the above dependencies with your own approach, or, if your cluster has external network access, you can use the following R service to install these dependencies.

a. In the **Add Service Wizard**, **Choose Services**, select the **R** service and click **Next**.

b. In the **Assign Slaves and Clients** page, for client nodes, mark all the nodes as the **R Client** node and click **Next**.

c. In the **Customize Services** page, accept the recommended configurations for the R service, or customize the configuration by expanding the configuration files and modifying the values.

   - Make sure that you read the R license, and indicate acceptance by typing Y in the field **accept.R.Licenses**. The value is case sensitive, so make sure you type an uppercase letter. The **R Licenses** field contains a URL where you can find the licensing information.
   - In the **user.R.packages** you must ensure that the following required packages are listed:base64enc, rJava, and data.table.
   - In the **user.R.repository** field, enter the preferred repository. The default is epel-release, which uses the EPEL repository, but you can also type a different repository by entering a URL, such as http://repos.*domain*.com/repos.

   Note: When installing R from the EPEL repository, you might have the following GPG key error: GPG key retrieval failed: [Errno 14] Could not open/read

   If you receive this error, you can import the key with the following rpm command, then retry: **rpm --import**

d. Click **Next** and in the Review Page that opens, click **Deploy**.

e. If R deployment fails, review and correct the errors before reattempting the installation. Remove the R service from Ambari and delete the RSERV server by using the following command:

```
curl -u [uid]:[pwd] -H "X-Requested-By:ambari" -X DELETE
http://[hostname]:8080/api/v1/clusters/[clustername]/services/RSERV
```

Where

[uid:[pwd]]

The Ambari administrator user ID and password.

[hostname]

The correct host name for your environment.

8080

The port number 8080 is the default. Modify this according to your environment.

[cluster name]

The correct name of your cluster.

The following command is an example:

```
curl -u admin:admin -H "X-Requested-By:ambari" -X DELETE
http://my_host.localdomain:8080/api/v1/clusters/my_cluster/services/RSERV
```

f. In the Summary page, click **Complete**. When you return to the Ambari Dashboard Services tab, you notice that the R service is now listed.

4. In the **Add Service Wizard**, **Choose Services** page, select the **Big R** service and click **Next**.

5. In the **Assign Masters** page, decide which nodes of your cluster you want to run the specified components, or accept the default nodes. You must assign the Big R Connector to the same node that is running the MapReduce2 Client service, which is a required service that runs MapReduce2 Hadoop jobs. Click **Next**.

6. In the **Assign Slaves and Clients** page, accept the defaults, or make specific assignments for your nodes. For client nodes, mark all of the nodes as the **Big R Client** node and click **Next**.

7. In the **Customize Services** page, default Big R environment variables are set in the **bigr-env template** field. Review these entries for accuracy and completeness. Make any necessary changes and click **Next**

8. You can review your selections in the **Review** page before accepting them. If you want to modify any values, click the **Back** button. If you are satisfied with your setup, click **Deploy**.

9. In the **Install, Start and Test** page, the Big R service is installed and verified. If you have multiple nodes, you can see the progress on each node. When the installation is complete, either view the errors or warnings by clicking the link, or click **Next** to see a summary and then the new service added to the list of services.

   If the **BigInsights – Big R** service fails to install, run the **remove_value_add_services.sh** cleanup script. The following code is an example of the command:

   ```
   cd /usr/ibmpacks/bin/<version>
   ./remove_value_add_services.sh -u admin -p admin -x 8080 -s BIGR -r
   ```

   For more information about cleaning the value-add service environment, see Removing BigInsights value-add services.

10. Advise your end users that the service is deployed and ready for their use by having them launch the Value Added packages welcome page.

11. In the Summary page, click **Complete**.

Running BigInsights - Big R as the YARN application master

You must update the Linux Container Executor as the default executor in the yarn- site.xml file to change the owner to the **bigr** server user (the application process owner).

1. In the Ambari web interface, from the YARN service Configs page, scroll down to find the **Advanced yarn-site** and expand it.

2. Change the **yarn.nodemanager.container-executor.class** property to have the following value:

```
org.apache.hadoop.yarn.server.nodemanager.LinuxContainerExecutor
```

3. In the **Custom yarn-site** section,click **Add Property** to add the following properties:

Property name	Value
yarn.nodemanager.linux-container-executor.nonsecure-mode.local-user	Yarn
yarn.nodemanager.linux-container-executor.nonsecure-mode.limit-users	False

4. Make sure that the property **yarn.nodemanager.linux-container- executor.group** has the valuehadoop.

5. Click **Save** in the Configs page to save your configuration changes.

6. Make sure that the directories on ALL the nodes set in the **Node Manager** section for the properties **yarn.nodemanager.local-dirs** and **yarn.nodemanager.log- dirs** have permissionsyarn:hadoop:

   On ALL nodes do the following commands:

```
$ echo yarn.nodemanager.linux-container-executor.group=hadoop" >> /etc/hadoop/conf/container-executor.cfg
$ echo "banned.users=hdfs,yarn,mapred,bin" >>/etc/hadoop/conf/container-executor.cfg
$ echo "min.user.id=1000" >> /etc/hadoop/conf/container-executor.cfg
$ chown root:hadoop /etc/hadoop/conf/container-executor.cfg
$ chown root:hadoop /usr/iop/4.2.0.0/hadoop-yarn/bin/container-executor
$ chmod 6050 /usr/iop/4.2.0.0/hadoop-yarn/bin/container-executor
```

7. Make sure that the user ID with which the BigR connection is made (by using bigr.connect) is present on ALL nodes, and that the user belongs to groups **users**, hadoop. If the user does not exist, run the following command as the root user on ALL nodes:

```
$ useradd -G users,hadoop someuser
```

8. Change the SystemML configuration file, /usr/ibmpacks/current/bigr/machine- learning/SystemML-config.xml:

```
dml.yarn.appmaster value: true
```

9. You can optionally update the MapReduce configuration to get better performance:

a. In the Ambari web interface, from the **MapReduce2** service **Configs** page, scroll down to find the **Advanced map-red** site section and expand it.

b. Update the property **mapreduce.task.io.sort.mb** to **384** . This should be approximately three times the HDFS block size.

   Note: If the property is not available, add it to the **Custom map-red site**.

10. Click **Save** in the Configs page to save your configuration changes.

For information about using BigInsights - Big R, see Analyzing data with IBM BigInsights Big R .

2.8.9 IBM BigInsights Online Tutorials

Learn how to use BigInsights™ by completing online tutorials, which use real data and teach you to run applications. Complete the tutorials in any order.

http://www.ibm.com/support/knowledgecenter/SSPT3X_4.2.0/com.ibm.swg.im.infosphere.biginsights.tut.doc/doc/tut_Introduction.html

You can find additional information, tutorials, and articles about BigInsights, Hadoop, and related components at Hadoop Dev.

http://developer.ibm.com/hadoop/docs/tutorials/

2.8.10 Ranger Installation

Please refer to the link for installation of Ranger for IOP 4.2

http://www.ibm.com/support/knowledgecenter/SSPT3X_4.2.0/com.ibm.swg.im.infosphere.biginsights.install.doc/doc/bi_install_ranger_opensource.html

Configuring Ranger plugins

To configure the Ranger plugins, please follow the below link

http://www.ibm.com/support/knowledgecenter/SSPT3X_4.2.0/com.ibm.swg.im.infosphere.biginsights.install.doc/doc/bi_install_ranger_plugins.html#bi_install_ranger_plugins

2.9 Kerberos Setup

2.9.1 Prerequisites

- OneFS 8.0.0.1 or higher.
- Ambari 2.0 or higher.
- MIT KDC running (Heimdal is not supported). Follow the steps here to setup up your Kerberos infrastructure.
- http://www.ibm.com/support/knowledgecenter/SSPT3X_4.2.0/com.ibm.swg.im.infosphere.biginsights.admin.doc/doc/admin_kerb_mankdc2.html
- Forward and reverse DNS between all hosts.
- All services are running (green) on the Ambari Dashboard.

2.9.2 Pre-Configuration

Before launching the wizard, you must set two configurations and restart all services.

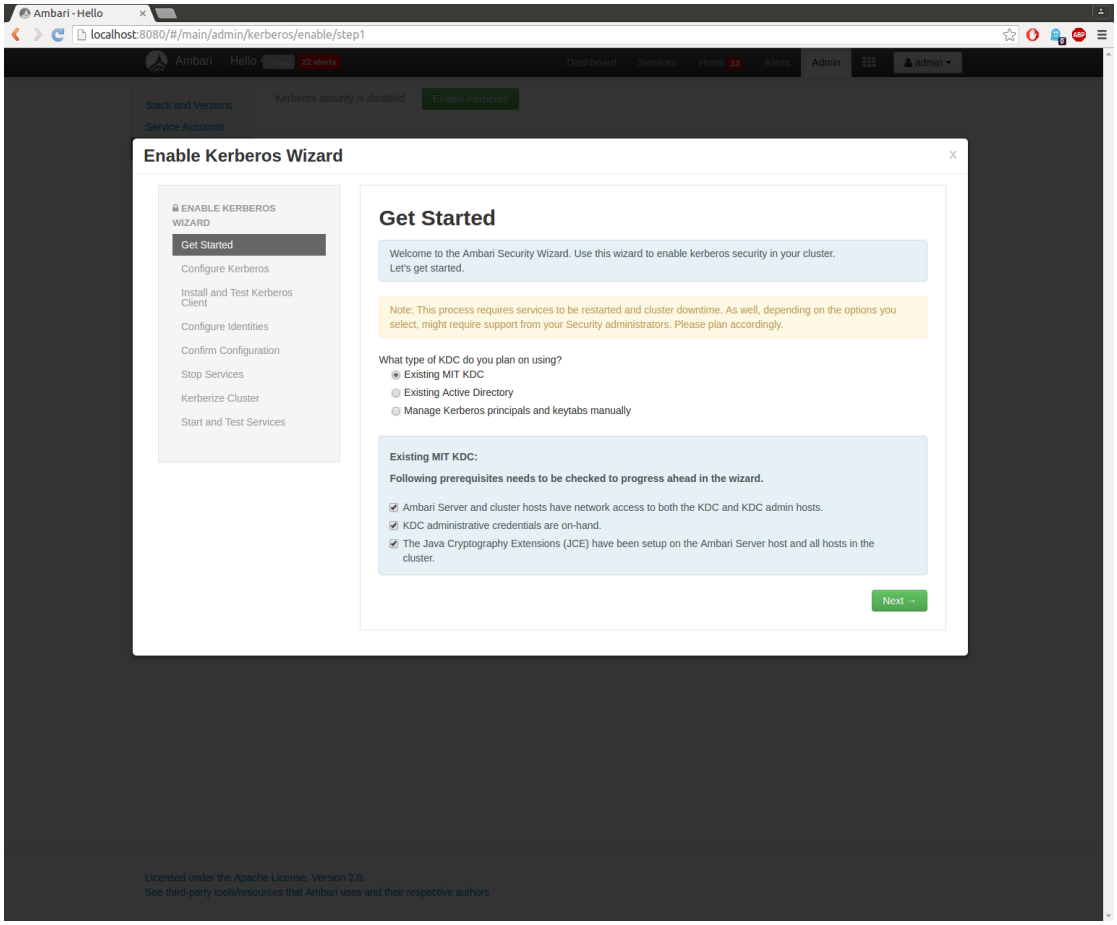
1. In **HDFS -> Custom core-site** set "hadoop.security.token.service.use_ip" to "false"

2. In **MapReduce2 -> Advanced mapred-site** add "`hadoop classpath`:" to the beginning of "mapreduce.application.classpath". Note the colon and backticks (but do **not** copy the quotation marks).

Make sure that the Spark Thrift server and the Hive server have been installed on the same node in the cluster.

2.9.3 Get Started

Navigate to **Admin** -> **Kerberos** and press the **Enable Kerberos** button. The titles within this section refer to the titles of the Kerberos wizard pages.

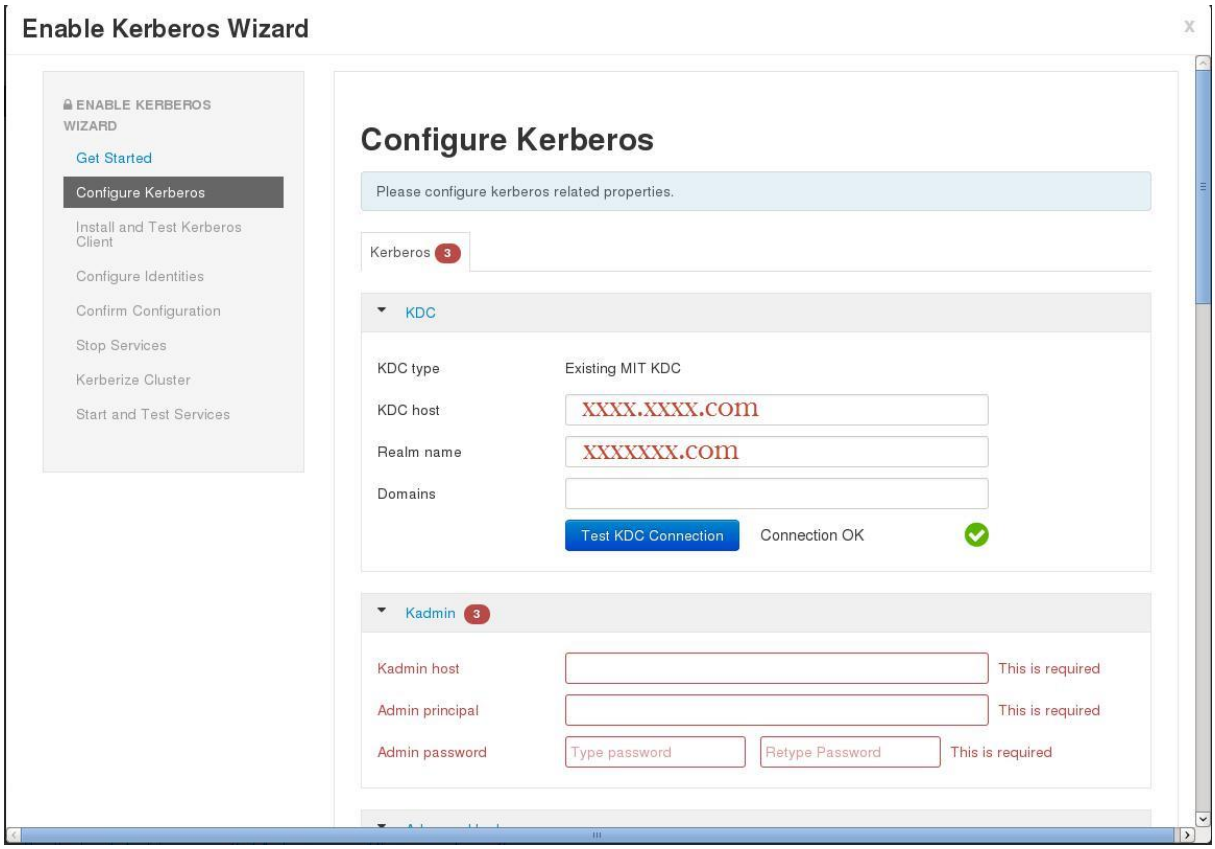


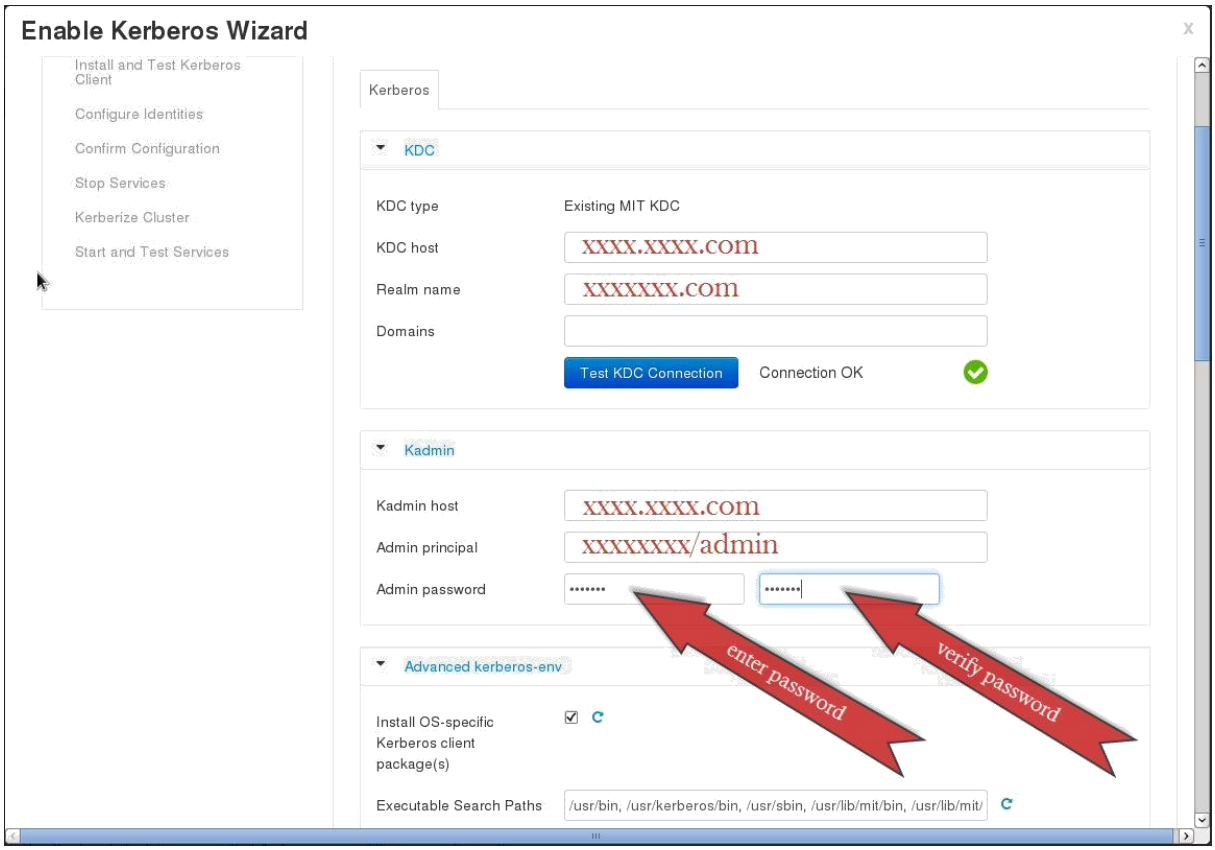
Select **Existing MIT KDC**, and ensure that the pre-requisites are met, then click **Next**. Note that Isilon does not use Java, and does not need the JCE.

2.9.4 Configure Kerberos

Enter the appropriate “KDC” values for your environment:

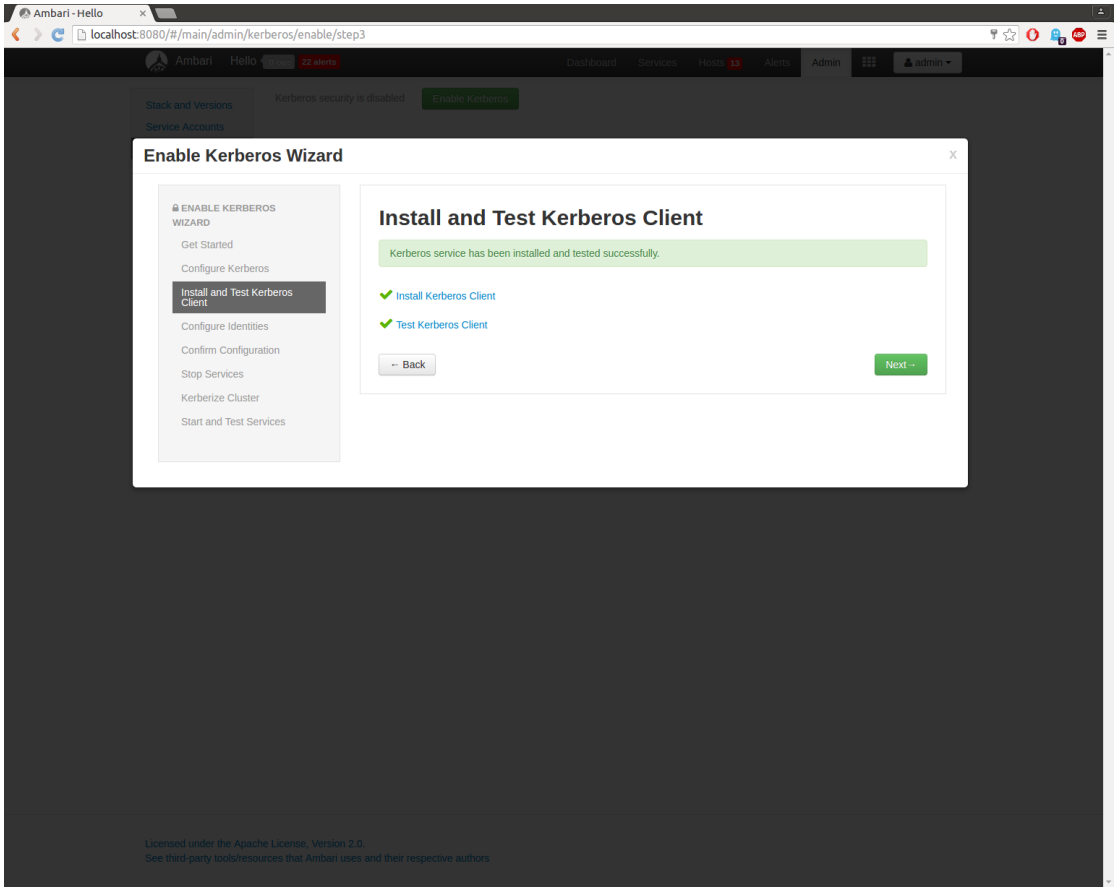
KDC host	= KDCHost.Example.com
Realm name	= Example.COM
Kadmin host	= KDCHost.Example.com
Admin principal	= admin/admin@Example.com
Admin password	= YourAdminPassword


Enable Kerberos Wizard
ENABLE KERBEROS WIZARD
Get Started
Configure Kerberos
Install and Test Kerberos Client
Configure Identities
Confirm Configuration
Stop Services
Kerberize Cluster
Start and Test Services
Configure Kerberos
Please configure kerberos related properties.
Kerberos 3
KDC
KDC type
Existing MIT KDC
KDC host
xxxx.xxxx.com
Realm name
xxxxxxx.com
Domains
Test KDC Connection
Connection OK
Kadmin 3
Kadmin host
This is required
Admin principal
This is required
Admin password
Type password
Retype Password
This is required


Enable Kerberos Wizard
Install and Test Kerberos Client
Configure Identities
Confirm Configuration
Stop Services
Kerberize Cluster
Start and Test Services
Kerberos
KDC
KDC type
Existing MIT KDC
KDC host
xxxx.xxxx.com
Realm name
xxxxxxx.com
Domains
Test KDC Connection
Connection OK
Kadmin
Kadmin host
xxxx.xxxx.com
Admin principal
xxxxxxx/admin
Admin password
enter password
verify password
Advanced kerberos-env
Install OS-specific Kerberos client package(s)
Executable Search Paths
/usr/bin, /usr/kerberos/bin, /usr/sbin, /usr/lib/mit/bin, /usr/lib/mit/

2.9.5 Install and test the Kerberos Client

On step 3 (Install and Test Kerberos Client), the Ambari server will do a smoke test to ensure you have configured Kerberos correctly.



2.9.6 Configure Identities & Confirm Configuration

a) Ambari User Principals (UPNs)

Ambari creates user principals in the form ${username}-${clustername}@${realm}, then uses hadoop.security.auth_to_localin core-site.xml to map the principals into just ${username} on the filesystem.

Isilon does not honor the mapping rules, so you must remove the -${clustername} from all principals in the "Ambari Principals" section. Isilon will strip off the @${realm}, so no aliasing is necessary. Make the following modifications in the "General" tab:

Smokeuser Principal Name:

```
${cluster-env/smokeuser}-${cluster_name}@${realm} => ${cluster-env/smokeuser}@${realm}
```

HDFS user principal:

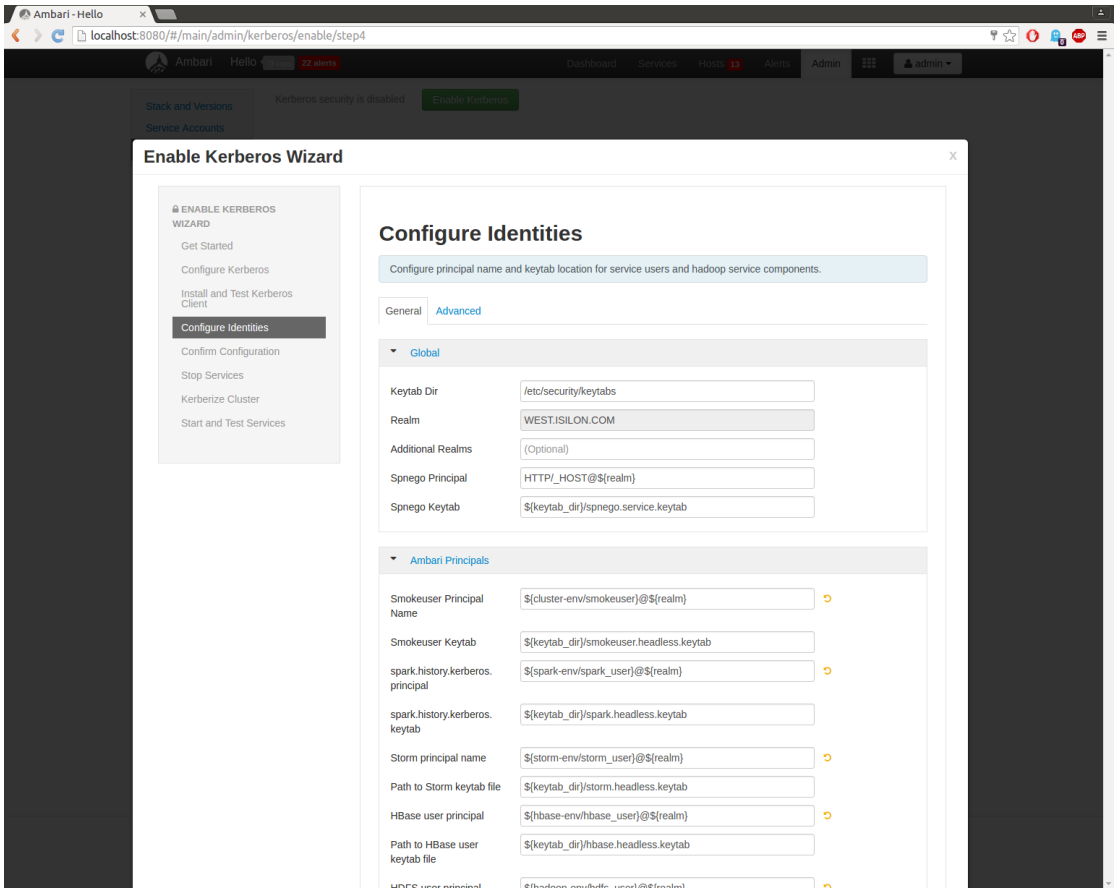
```
${hadoop-env/hdfs_user}-${cluster_name}@${realm} => ${hadoop-env/hdfs_user}@${realm}
```

spark.history.kerberos.principal:

```
${spark-env/spark_user}-${cluster_name}@${realm} => ${spark-env/spark_user}-@${realm}
```

HBase user principal:

```
$hbase-env/hbase_user}-${cluster_name}@${realm} => ${hbase-env/hbase_user}@${realm}
```



b) Service Principals (SPNs)

Ambari creates service principals, some of which are different than their UNIX usernames. Again, since Isilon does not honor the mapping rules, you must modify the principal names to match their UNIX usernames. In my Ambari 2.2.1 cluster, I made the following modifications in the "Advanced" tab:

Expand HDFS and replace “nn” with “hdfs”

Before	After
`dfs.namenode.kerberos.principal = nn/_HOST@${realm}`	`dfs.namenode.kerberos.principal = hdfs/_HOST@${realm}`
`dfs.namenode.keytab.file = ${keytab_dir}/nn.service.keytab`	`dfs.namenode.keytab.file = ${keytab_dir}/hdfs.service.keytab`
`dfs.secondary.namenode.kerberos.principal = nn/_HOST@${realm}`	`dfs.secondary.namenode.kerberos.principal = hdfs/_HOST@${realm}`
`dfs.secondary.namenode.keytab.file = ${keytab_dir}/nn.service.keytab`	`dfs.secondary.namenode.keytab.file = ${keytab_dir}/hdfs.service.keytab`

Within HDFS replace “dn” with “hdfs”

Before	After
dfs.datanode.kerberos.principal = dn/_HOST@${realm}	dfs.datanode.kerberos.principal = hdfs/_HOST@${realm}
dfs.datanode.keytab.file = ${keytab_dir}/dn.service.keytab	dfs.datanode.keytab.file = ${keytab_dir}/hdfs.service.keytab

Within HDFS replace “jn” with “hdfs”

Before	After
dfs.journalnode.kerberos.principal = jn/_HOST@${realm}	dfs.journalnode.kerberos.principal = hdfs/_HOST@${realm}
dfs.journalnode.keytab.file = ${keytab_dir}/jn.service.keytab	dfs.journalnode.keytab.file = ${keytab_dir}/hdfs.service.keytab

Expand MapReduce2 and replace “jhs” with “mapred”

Before	After
mapreduce.jobhistory.principal = jhs/_HOST@${realm}	mapreduce.jobhistory.principal = mapred/_HOST@${realm}
mapreduce.jobhistory.keytab = ${keytab_dir}/jhs.service.keytab	mapreduce.jobhistory.keytab = ${keytab_dir}/mapred.service.keytab

Expand Yarn and replace “nm” with “yarn”

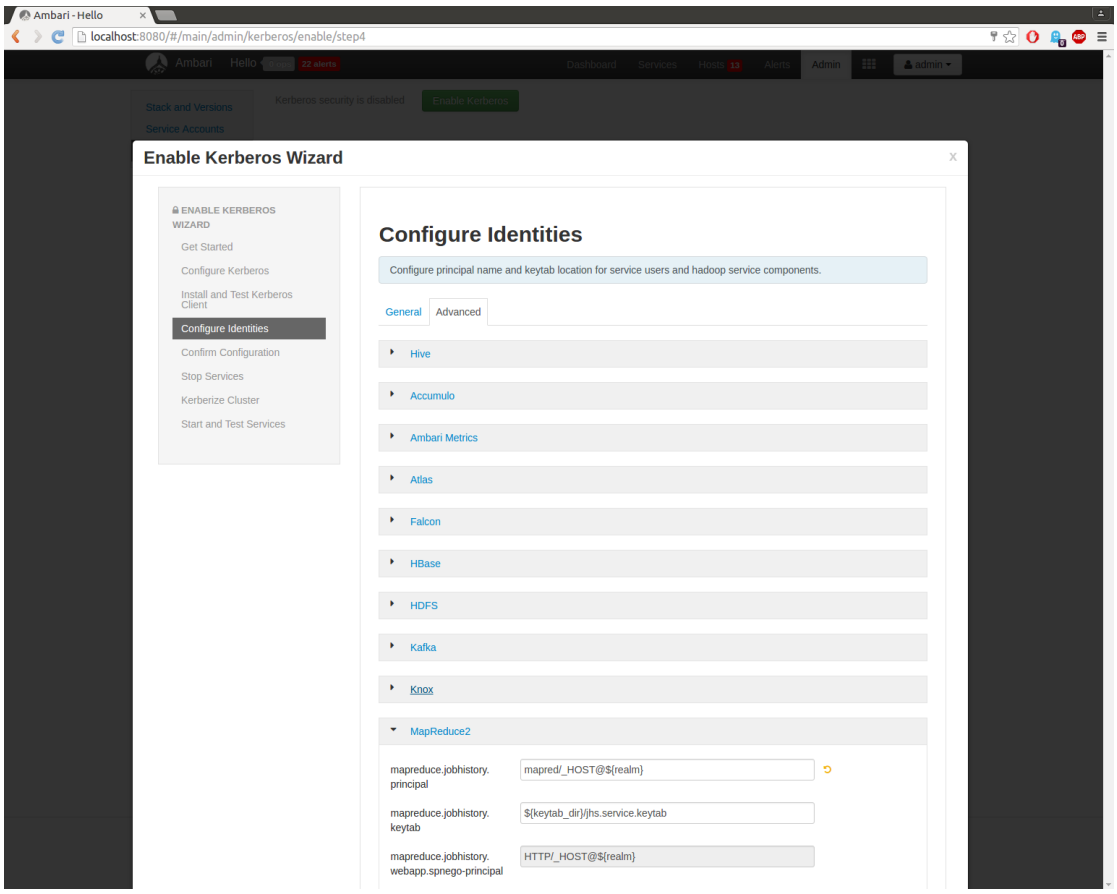
Before	After
yarn.nodemanager.principal = nm/_HOST@${realm}	yarn.nodemanager.principal = yarn/_HOST@${realm}
yarn.nodemanager.keytab = ${keytab_dir}/nm.service.keytab	yarn.nodemanager.keytab = ${keytab_dir}/yarn.service.keytab

Within Yarn replace “rm” with “yarn”

Before	After
`yarn.resourcemanager.principal = rm/_HOST@${realm}`	`yarn.resourcemanager.principal = yarn/_HOST@${realm}`
`yarn.resourcemanager.keytab = ${keytab_dir}/rm.service.keytab`	`yarn.resourcemanager.keytab = ${keytab_dir}/yarn.service.keytab`

Within Spark replace “spark-{$CLUSTER_NAME}” with “spark”

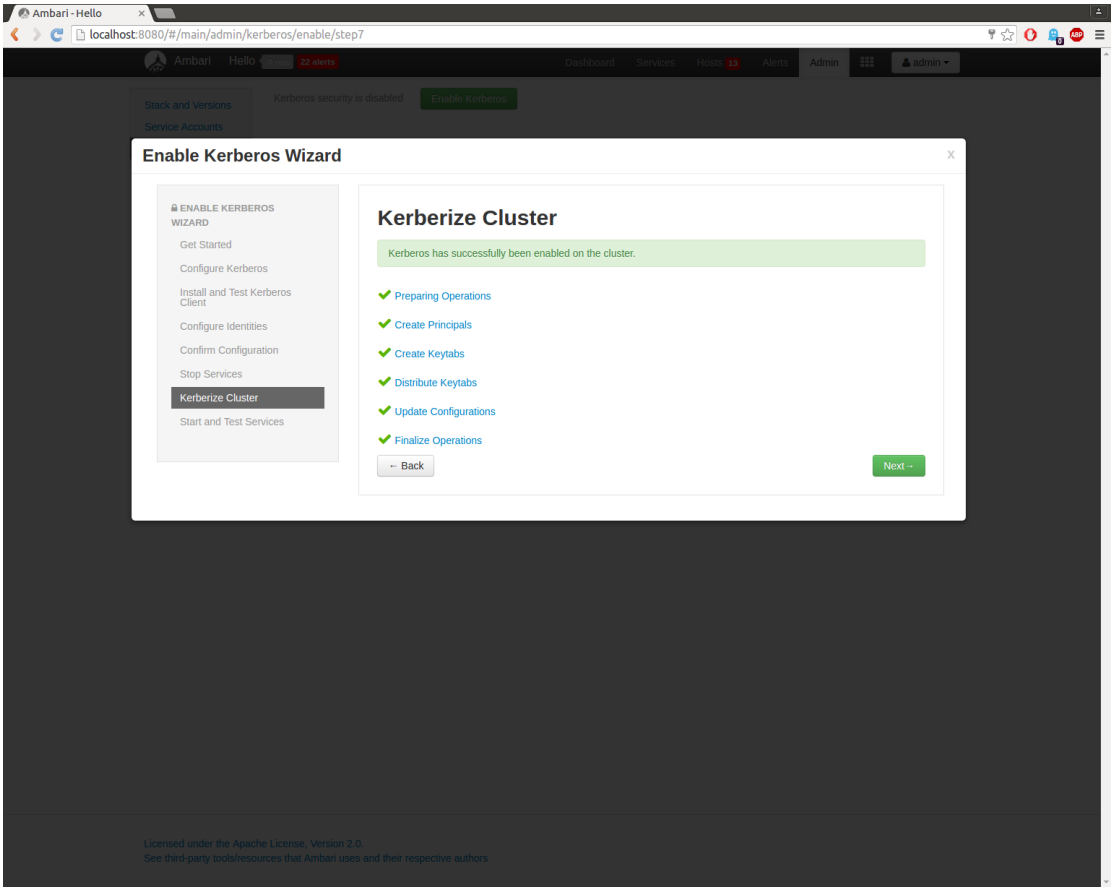
Before	After
`Dsprk.history.kerberos.principal = spark-{$CLUSTER_NAME}/_HOST@${realm}`	`Dsprk.history.kerberos.principal = spark/_HOST@${realm}`
`Dsprk.history.kerberos.keytab = ${keytab_dir}/ spark-{$CLUSTER_NAME}.headless.keytab`	`Dsprk.history.kerberos.keytab = ${keytab_dir}/ spark.headless.keytab`



After configuring the appropriate principals, press **Next**. At the **Confirm Configuration** screen, press **Next**.

2.9.7 Stop Services / Kerberize Cluster

Stopping and Kerberizing services should succeed.



Do not proceed: Isilon does not allow Ambari to create keytabs for Isilon principals. Instead, you must manually configure Kerberos on Isilon using the steps below.

a) Create KDC as an Isilon auth provider

Note: If this Isilon zone is already configured to use your MIT KDC, you can skip these steps.

```
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
```

b) Create service principals for HDFS and HTTP (for WebHDFS)

```
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
```

c) Create any necessary proxy users

In unsecured clusters, any user can impersonate any other user. In secured clusters, proxy users need to be explicitly specified.

If you have Hive or Oozie, add the appropriate proxy users.

```
isi hdfs proxyusers create oozie --zone=$isilon_zone --add-user=ambari-qa
isi hdfs proxyusers create hive --zone=$isilon_zone --add-user=ambari-qa
isi hdfs proxyusers create hbase --zone=$isilon_zone --add-user=ambari-qa
isi hdfs proxyusers create bigsql --zone=$isilon_zone --add-user=ambari-qa
isi hdfs proxyusers modify --proxyuser=hive --zone=$isilon_zone --add-user=bigsql
isi hdfs proxyusers modify --proxyuser=hbase --zone=$isilon_zone --add-user=bigsql
```

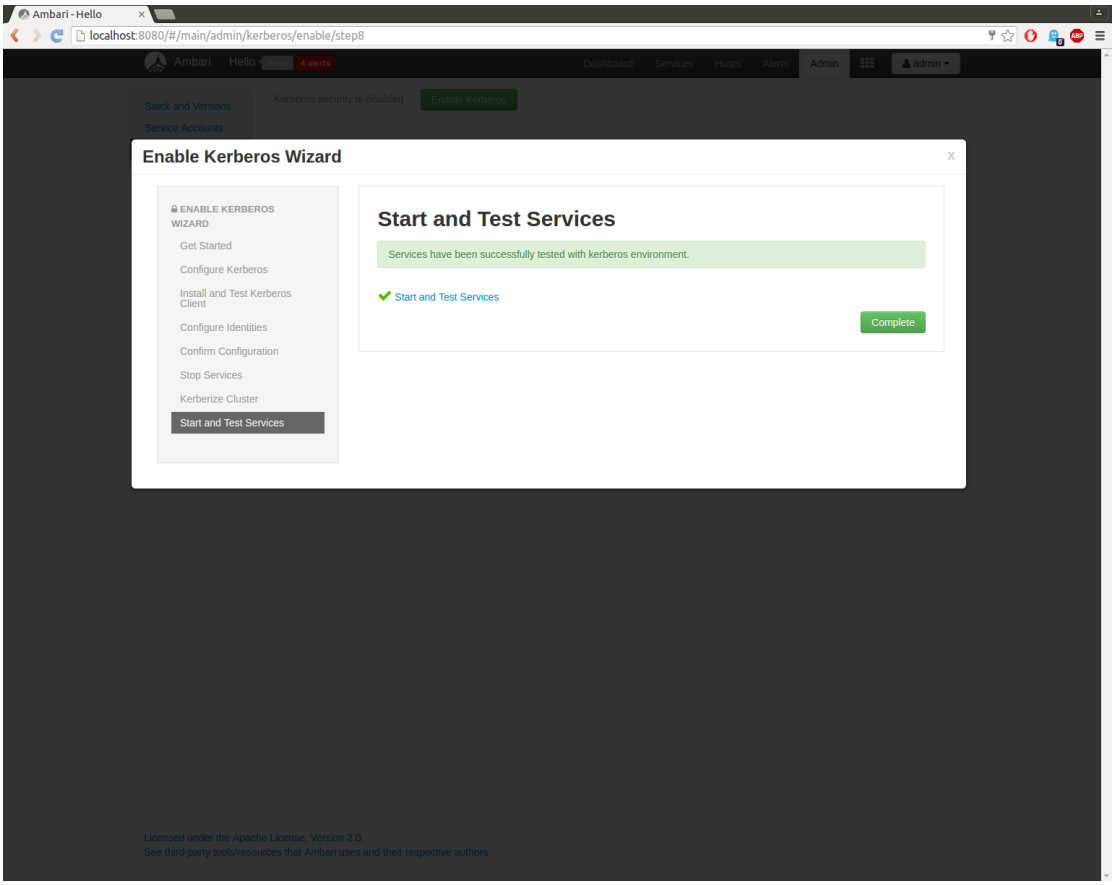
d) Disable simple authentication

Only Kerberos or delegation token authentication will be allowed.

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

Now that Isilon is configured as well, press "Next" in Ambari to move on to the last step of the wizard.

2.9.8 Start and test the Services



If services do not start up, here are some tricks for debugging Kerberos issues:

1. Due to a bug in YARN, you need to set the "yarn.resourcemanager.principal" to yarn/$rm_hostname@$REALM inYARN -> Custom yarn-site. The "_HOST" syntax does not work with Kerberos enabled.

2. To debug Java GSSAPI/Kerberos errors, add "-Dsun.security.krb5.debug=true" to HADOOP_OPTS.

3. For HTTP 401 errors, use curl with -iv for extra debug information.

4. Ensure forward and reverse DNS is set up between all hosts.

5. Due to a bug in oozie(service check failed), you need to apply the patch in the link.

(Optional) Strong RPC Security

In **HDFS -> Custom core-site** set "hadoop.rpc.protection" to "integrity" or "privacy". In addition to authentication, integrity guarantees messages have not been tampered with, and privacy encrypts all messages.

Run a job!

From any client host, try a MapReduce job!

```
kinit <some-user>
yarn jar /usr/iop/current/hadoop-mapreduce-client/hadoop-mapreduce-examples.jar pi 1 1000
```

Job Finished in 37.635 seconds

Estimated value of Pi is 3.14800000000000000000

Congratulations--you have secured your cluster with Kerberos!

2.9.9 (Optional) Disable Kerberos

Clean up Isilon

Let's clean up Isilon first. This is essentially the inverse of enabling Kerberos.

a) Disable Kerberos authentication

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

b) Delete any proxy users

```
isi isi hdfs proxyusers delete oozie --zone=$isilon_zone
isi hdfs proxyusers delete hive --zone=$isilon_zone
```

c) Delete principals

```
isi auth krb5 spn delete --provider-name=$REALM --spn=hdfs/$isilon_smartconnect@$REALM -all
isi auth krb5 spn delete --provider-name=$REALM --spn=HTTP/$isilon_smartconnect@$REALM --all
```

Note: The above commands only remove those principals from Isilon, but do not remove them from the KDC. Use these commands to remove the Isilon principals from the KDC:

```
kadmin -p $admin_principal
kadmin: delete_principal hdfs/$isilon_smartconnect@$REALM
kadmin: delete_principal HTTP/$isilon_smartconnect@$REALM
```

d) Remove KDC as an Isilon authentication provider

```
isi zone zones modify --zone=$isilon_zone --remove-auth-provider=krb5:$REALM
isi auth krb5 delete --provider-name=$REALM
```

Clean up clients using Ambari

Press **Disable Kerberos** in Admin -> Kerberos. All the services should come up green.

3 Known Issues

3.1 Disable HDFS Caching

HDFS Caching must be disabled for Big SQL to work with a Dell EMC Isilon cluster. The $BIGSQL_HOME/conf/bigsql-conf.xml file must modified as follow:

```
<property>
      <name>bigsql.enable.caching.code</name>
      <value>false</value>
</property>
```

Note: The Big SQL services must be restarted after the modifications have been applied.

3.2 Patch for oozie

Due to a bug in oozie (service check failed), you need to apply this patch for Biginsights 4.2.

a. Open iop-patch-management blog by https://developer.ibm.com/hadoop/2015/12/17/iop-patch-management/
b. Follow the guide to apply patch

1. Extract Patch Management tool

2. Modify the IOP repository URL http://ibm-open-platform.ibm.com/repos/IOP/rhel/6/x86_64/4.2.x/Updates/4.2.0.0_20161024/

3. Stop the service and turn on the maintenance mode on confirmation window and Confirm Stop.
   - HDFS
   - MAPREDUCE2
   - YARN
   - OOZIE
   - ZOOKEEPER
4. Run Patch Management script

Note: User should input "Service name to patch" as HDFS,MAPREDUCE2,YARN,OOZIE separately, if there is any warning message during applying patch on MAPREDUCE2 and YARN, just ignore it.

5. Start the stopped services

For BigInsights 4.1, please follow the steps outlined in the guide located at https://ibm.box.com/s/ecvfa7z9uq9ebfl2oqxyp3mb8jmm44sv to apply the patch manually.

3.3 Multiple Repos

The Dell EMC Isilon OneFS Ambari agent has host type set to RHEL 6.x, if installing BigInsights 4.x on a RHEL 7.x environment, the following changes must be applied to successfully install BigInsights 4.x in a RHEL 7.x environment with Isilon OneFS. After the ambari-server setup and before creating the cluster in the Ambari GUI. Perform the following steps:

1. On the Ambari Server host, open /var/lib/ambari-server/resources/stacks/BigInsights/4.2/repos/repoinfo.xml
2. Duplicate the '<os family="redhat7"' block. Ensure it is placed before the '</reposinfo>' tag.
3. In the duplicated block, rename 'redhat7' to 'redhat6'.
4. Restart the Ambari server process.

3.4 Solr service issue

When you start Solr service or Solr service check, if there is some error message as following, please double check the JAVA_HOME environment parameter

…

resource_management.core.exceptions.Fail: Failed to create collection

…

Solution:

Add the following command in /etc/profile to export JAVA_HOME parameter and restart ambari-server, solr.

```
export JAVA_HOME=/usr/jdk64/java-1.8.0-openjdk-1.8.0.77-0.b03.el6_7.x86_64
```

3.5 How to bypass ssh issue during setup of BigInsights - BigSQL against custom zone

When you set up BigInsights - BigSQL against custom zone, you might run into an ssh specific error in the preCheck phase:

"Passwordless ssh needs to be configured between the headnode and all hosts in the cluster."

Double check the passwordless ssh configuration is completed among the head node, worker nodes and Isilon cluster. If the ssh configuration is completed but the above error message is still showing in the preCheck phase, create a file "bigsqlPreChecker.Success" in /tmp as a workaround to complete the preCheck phase.

4 IBM Open Platform stack Version Comparison

Apache Components	Version	
	IOP 4.1.0.x	IOP 4.2.x.x
Ambari	2.1	2.2.0
Apache Kafka	0.8.2	0.9.0.1
Flume	1.5.2	1.6.0
Ganglia	3.1.7	
Hadoop	2.7.1	2.7.2
HBase	1.1.1	1.2.0
Hive	1.2.1	1.2.1
Knox	0.6.0	0.7.0
Lucene	4.7.0	
Nagios	3.5.1	
Oozie	4.2.0	4.2.0
Parquet	4	
Parquet (MR / format)	1.6.0/2.2	1.6.00/2.2
Pig	0.15.0	0.15.0
Slider	0.80.0	0.90.2
Solr	5.1.0	5.5.0
Spark	1.4.1	1.6.1
Sqoop	1.4.6	1.4.6
Teradata Connector for Hadoop	1.4	
Zookeeper	3.4.6	3.4.6
Phoenix		4.6.1
Ranger		0.5.2
SystemML		0.10.0
Titan		1.0.0

The highlighted Apache Components in yellow are NEW ecosystem components for IOP 4.2