Hortonworks DataFlow

Apache Ambari Installation

(Feb 24, 2017)

docs.hortonworks.com

Hortonworks DataFlow: Apache Ambari Installation

Hortonworks DataFlow (HDF) is powered by Apache NiFi. A version of this documentation originally appeared on the Apache NiFi website.

HDF is the first integrated platform that solves the real time challenges of collecting and transporting data from a multitude of sources and provides interactive command and control of live flows with full and automated data provenance. HDF is a single combined platform that provides the data acquisition, simple event processing, transport and delivery mechanism designed to accommodate the diverse dataflows generated by a world of connected people, systems and things.

Unlike other providers of platforms built using Apache Hadoop, Hortonworks contributes 100% of our code back to the Apache Software Foundation. Hortonworks DataFlow is Apache-licensed and completely open source. We sell only expert technical support, training and partner-enablement services. All of our technology is, and will remain free and open source.

Please visit the Hortonworks page for more information on Hortonworks technology. For more information on Hortonworks services, please visit either the Support or Training page. Feel free to Contact Us directly to discuss your specific needs.

Table of Contents

List of Tables

1. Getting Ready

You should be sure you have the following materials and information prepared before you install an HDF cluster using Ambari. Ambari provides an end-to-end management and monitoring solution for your HDF cluster. Using the Ambari Web UI and REST APIs, you can deploy, operate, manage configuration changes, and monitor services for all nodes in your cluster from a central point.

1.1. System Requirements

Before you get started, you should be sure your system meets the following requirements.

1.1.1. Interoperability Requirements

You cannot install HDF on a system where HDP is already installed.

1.1.2. Supported Operating Systems

- Red Hat Enterprise Linux / CentOS 6 (64-bit)
- Red Hat Enterprise Linux / CentOS 7 (64-bit)
- Ubuntu Precise (12.04) (64-bit)
- Ubuntu Trusty (14.04) (64-bit)
- Debian 7
- SUSE Linux Enterprise Server (SLES) v12 SP1
- SUSE Linux Enterprise Server (SLES) v11 SP4
- SUSE Linux Enterprise Server (SLES) v11 SP3

1.1.3. Supported Browsers

- Mozilla Firefox: current & current - 1
- Google Chrome: current & current - 1
- Microsoft Edge
- Safari 8

1.1.4. Software Requirements

On each of your hosts:

- yum and rpm (RHEL/CentOS/Oracle Linux)
- zypper and php_curl (SLES)
- apt (Debian/Ubuntu)

- scp, curl, unzip, tar, and wget
- OpenSSL (v1.01, build 16 or later)
- Python
  - **For CentOS 6:** Python 2.6.*
  - **For SLES 11, SLES 12:** Python 2.6.8 or later
  - **For CentOS 7, Ubuntu 12, Ubuntu 14, and Debian 7:** Python 2.7.*

1.1.5. Supported JDKs

You must have one of the following JDKs installed on the system running HDF.

- Open JDK8
- Oracle JDK 8

1.1.6. Database Requirements

Ambari requires a relational database to store information about the cluster configuration and topology. The following table outlines these database requirements:

Component	Databases	Description
Ambari	- PostgreSQL 8 - PostgreSQL 9.1.13+,9.3 - MariaDB 10* - MySQL 5.6 - Oracle 11gr2 - Oracle 12c**	By default, Ambari will install an instance of PostgreSQL on the Ambari Server host. Optionally, to use an existing instance of PostgreSQL, MySQL or Oracle.
Ranger	- PostgreSQL 9.1.13+, 9.3 - MariaDB 10* - MySQL 5.6 - Oracle 11gr2 - Oracle 12c**	You must have an existing instance of PostgreSQL , MySQL or Oracle available for Ranger.

Important

For the Ambari database, if you use an existing Oracle database, make sure the Oracle listener runs on a port other than 8080 to avoid conflict with the default Ambari port.

1.1.7. Memory Requirements

The Ambari host should have at least 1 GB RAM, with 500 MB free.

To check available memory on any host, run:

```
free -m
```

Number of hosts	Memory Available	Disk Space
1	1024 MB	10 GB
10	1024 MB	20 GB
50	2048 MB	50 GB
100	4096 MB	100 GB
300	4096 MB	100 GB
500	8096 MB	200 GB
1000	12288 MB	200 GB
2000	16384 MB	500 GB

1.1.8. Check the Maximum Open File Descriptors

The recommended maximum number of open file descriptors is 10000, or more. To check the current value set for the maximum number of open file descriptors, execute the following shell commands on each host:

```
ulimit -Sn
```

```
ulimit -Hn
```

If the output is not greater than 10000, run the following command to set it to a suitable default:

```
ulimit -n 10000
```

1.2. Prepare the Environment

To deploy HDF using Ambari, you need to prepare your deployment enivornment:

- Set up Password-less SSH
- Set Up Service User Accounts
- Enable NTP on the Cluster
- Check DNS and NSCD
- Configure iptables
- Disable SELinux, PackageKit and Check umask Value

1.2.1. Set Up Password-less SSH

To have Ambari Server automatically install Ambari Agents on all your cluster hosts, you must set up password-less SSH connections between the Ambari Server host and all other hosts in the cluster. The Ambari Server host uses SSH public key authentication to remotely access and install the Ambari Agent.

1. Generate public and private SSH keys on the Ambari Server host.

   ```
   ssh-keygen
   ```

2. Copy the SSH Public Key (id_rsa.pub) to the root account on your target hosts.

   `.ssh/id_rsa`

   `.ssh/id_rsa.pub`

3. Add the SSH Public Key to the authorized_keys file on your target hosts.

   ```
   cat id_rsa.pub >> authorized_keys
   ```

4. Depending on your version of SSH, you may need to set permissions on the .ssh directory (to 700) and the authorized_keys file in that directory (to 600) on the target hosts.

   ```
   chmod 700 ~/.ssh
   ```

   ```
   chmod 600 ~/.ssh/authorized_keys
   ```

5. From the Ambari Server, make sure you can connect to each host in the cluster using SSH, without having to enter a password.

   `ssh root@<remote.target.host>` where `<remote.target.host>` has the value of each host name in your cluster.

6. If the following warning message displays during your first connection: `Are you sure you want to continue connecting (yes/no)?` Enter `Yes`.

7. Retain a copy of the SSH Private Key on the machine from which you will run the web-based Ambari Install Wizard.

Note

It is possible to use a non-root SSH account, if that account can execute `sudo` without entering a password.

1.2.2. Set Up Service User Accounts

Each service requires a service user account. The Ambari Install wizard creates new and preservers any existing service user accounts. Service user account creates applices to service user accounts on the local operating system and to LDAP/AD accounts.

1.2.3. Enable NTP on the Cluster and on the Browser Host

The clocks of all the nodes in your cluster and the machine that runs the browser through which you access the Ambari Web interface must be able to synchronize with each other.

To check that the NTP service will be automatically started upon boot, run the following command on each host:

RHEL/CentOS/Oracle 6

```
chkconfig --list ntpd
```

RHEL/CentOS/Oracle 7

```
systemctl is-enabled ntpd
```

To set the NTP service to auto-start on boot, run the following command on each host:

RHEL/CentOS/Oracle 6

```
chkconfig ntpd on
```

RHEL/CentOS/Oracle 7

```
systemctl enable ntpd
```

To start the NTP service, run the following command on each host:

RHEL/CentOS/Oracle 6

```
service ntpd start
```

RHEL/CentOS/Oracle 7

```
systemctl start ntpd
```

1.2.4. Check DNS and NSCD

All hosts in your system must be configured for both forward and and reverse DNS.

If you are unable to configure DNS in this way, you should edit the /etc/hosts file on every host in your cluster to contain the IP address and Fully Qualified Domain Name of each of your hosts. The following instructions are provided as an overview and cover a basic network setup for generic Linux hosts. Different versions and flavors of Linux might require slightly different commands and procedures. Please refer to the documentation for the operating system(s) deployed in your environment.

To reduce the load on your DNS infrastructure, it is recommended to use the Name Service Caching Daemon (NSCD) on cluster nodes running Linux. This daemon caches host, user, and group lookups and provide better resolution performance, and reduced load on DNS infrastructure.

1.2.4.1. Edit the Host File

1. Using a text editor, open the hosts file on every host in your cluster. For example:

   ```
   vi /etc/hosts
   ```

2. Add a line for each host in your cluster. The line should consist of the IP address and the FQDN. For example:

   ```
   1.2.3.4 <fully.qualified.domain.name>
   ```

Important

Do **not** remove the following two lines from your hosts file. Removing or editing the following lines may cause various programs that require network functionality to fail.

```
127.0.0.1 localhost.localdomain localhost

::1 localhost6.localdomain6 localhost6
```

1.2.4.2. Set the Hostname

1. Confirm that the hostname is set by running the following command:

   ```
   hostname -f
   ```

   This should return the <fully.qualified.domain.name> you just set.

2. Use the "hostname" command to set the hostname on each host in your cluster. For example:

   ```
   hostname <fully.qualified.domain.name>
   ```

1.2.4.3. Edit the Network Configuration File

1. Using a text editor, open the network configuration file on every host and set the desired network configuration for each host. For example:

   ```
   vi /etc/sysconfig/network
   ```

2. Modify the HOSTNAME property to set the fully qualified domain name.

   ```
   NETWORKING=yes

   HOSTNAME=<fully.qualified.domain.name>
   ```

1.2.5. Configuring iptables

For Ambari to communicate during setup with the hosts it deploys to and manages, certain ports must be open and available. The easiest way to do this is to temporarily disable iptables, as follows:

RHEL/CentOS/Oracle Linux 6

```
chkconfig iptables off

/etc/init.d/iptables stop
```

RHEL/CentOS/Oracle Linux 7

```
systemctl disable firewalld

service firewalld stop
```

You can restart iptables after setup is complete. If the security protocols in your environment prevent disabling iptables, you can proceed with iptables enabled, if all required ports are open and available.

Ambari checks whether iptables is running during the Ambari Server setup process. If iptables is running, a warning displays, reminding you to check that required ports are open

and available. The Host Confirm step in the Cluster Install Wizard also issues a warning for each host that has iptables running.

1.2.6. Disable SELinux and PackageKit and check the umask Value

1. You must disable SELinux for the Ambari setup to function. On each host in your cluster,

   ```
   setenforce 0
   ```

   Note

   To permanently disable SELinux set `SELINUX=disabled` in `/etc/selinux/config` This ensures that SELinux does not turn itself on after you reboot the machine .

2. On an installation host running RHEL/CentOS with PackageKit installed, open `/etc/yum/pluginconf.d/refresh-packagekit.conf` using a text editor. Make the following change:

   ```
   enabled=0
   ```

   Note

   PackageKit is not enabled by default on Debian, SLES, or Ubuntu systems. Unless you have specifically enabled PackageKit, you may skip this step for a Debian, SLES, or Ubuntu installation host.

3. UMASK (User Mask or User file creation MASK) sets the default permissions or base permissions granted when a new file or folder is created on a Linux machine. Most Linux distros set 022 as the default umask value. A umask value of 022 grants read, write, execute permissions of 755 for new files or folders. A umask value of 027 grants read, write, execute permissions of 750 for new files or folders.

   Ambari & HDP support umask values of 022 (0022 is functionally equivalent), 027 (0027 is functionally equivalent). These values must be set on all hosts.

   UMASK Examples:

   Setting the umask for your current login session:

   ```
   umask 0022
   ```

   Checking your current umask:

   ```
   umask 0022
   ```

   Permanently changing the umask for all interactive users:

   ```
   echo umask 0022 >> /etc/profile
   ```

2. Installing Ambari

To install Ambari for HDF, use the following steps:

- Download the Ambari Repository
- Set Up the Ambari Server
- Install the HDF Management Pack
- Start the Ambari Server

Warning

You cannot install Ambari to manage an HDF cluster on a system where HDP is already installed.

2.1. Download the Ambari Repository

Follow the instructions in the section for the operating system that runs your installation host.

- RHEL/CentOS/Oracle Linux 6
- RHEL/CentOS/Oracle Linux 7
- SLES 12
- SLES 11
- Ubuntu 12
- Ubuntu 14
- Debian 7

Use a command line editor to perform each instruction.

2.1.1. RHEL/CentOS/Oracle Linux 6

On a server host that has Internet access, use a command line editor to perform the following steps:

1. Log in to your host as `root`.
2. Download the Ambari repository file to a directory on your installation host.

```
wget -nv http://public-repo-1.hortonworks.com/ambari/
centos6/2.x/updates/2.4.2.0/ambari.repo -O /etc/yum.repos.d/
ambari.repo
```

Important

Do not modify the `ambari.repo` file name. This file is expected to be available on the Ambari Server host during Agent registration.

3. Confirm that the repository is configured by checking the repo list.

   ```
   yum repolist
   ```

   You should see values similar to the following for Ambari repositories in the list.

   Version values vary, depending on the installation.

repo id	repo name	status
AMBARI.2.4.2.0-2.x	Ambari 2.x	5
base	CentOS-6 - Base	6,518
extras	CentOS-6 - Extras	15
updates	CentOS-6 - Updates	209

4. Install the Ambari bits. This also installs the default PostgreSQL Ambari database.

   ```
   yum install ambari-server
   ```

5. Enter y when prompted to to confirm transaction and dependency checks.

   A successful installation displays output similar to the following:

```
Installing : postgresql-libs-8.4.20-3.el6_6.x86_64          1/4
Installing : postgresql-8.4.20-3.el6_6.x86_64               2/4
Installing : postgresql-server-8.4.20-3.el6_6.x86_64        3/4
Installing : ambari-server-2.4.2.0-1470.x86_64                4/4
Verifying  : ambari-server-2.4.2.0-1470.x86_64                1/4
Verifying  : postgresql-8.4.20-3.el6_6.x86_64               2/4
Verifying  : postgresql-server-8.4.20-3.el6_6.x86_64        3/4
Verifying  : postgresql-libs-8.4.20-3.el6_6.x86_64          4/4

Installed:
  ambari-server.x86_64 0:2.4.2.0-1470

Dependency Installed:
 postgresql.x86_64 0:8.4.20-3.el6_6
 postgresql-libs.x86_64 0:8.4.20-3.el6_6
 postgresql-server.x86_64 0:8.4.20-3.el6_6
```

Note

Accept the warning about trusting the Hortonworks GPG Key. That key will be automatically downloaded and used to validate packages from Hortonworks. You will see the following message:

```
Importing GPG key 0x07513CAD: Userid: "Jenkins (HDP
Builds) <jenkin@hortonworks.com>" From : http://
s3.amazonaws.com/dev.hortonworks.com/ambari/centos6/RPM-
GPG-KEY/RPM-GPG-KEY-Jenkins
```

Note

Ambari Server by default uses an embedded PostgreSQL database. When you install the Ambari Server, the PostgreSQL packages and dependencies must be available for install. These packages are typically available as part of

your Operating System repositories. Please confirm you have the appropriate repositories available for the postgresql-server packages.

2.1.2. RHEL/CentOS/Oracle Linux 7

On a server host that has Internet access, use a command line editor to perform the following steps:

1. Log in to your host as `root`.
2. Download the Ambari repository file to a directory on your installation host.

   ```
   wget -nv http://public-repo-1.hortonworks.com/ambari/
   centos7/2.x/updates/2.4.2.0/ambari.repo -O /etc/yum.repos.d/
   ambari.repo
   ```

   Important

   Do not modify the `ambari.repo` file name. This file is expected to be available on the Ambari Server host during Agent registration.

3. Confirm that the repository is configured by checking the repo list.

   ```
   yum repolist
   ```

   You should see values similar to the following for Ambari repositories in the list.

   Version values vary, depending on the installation.

repo id	repo name	status
AMBARI.2.4.2.0-2.x	Ambari 2.x	5
base	CentOS-7 - Base	6,518
extras	CentOS-7 - Extras	15
updates	CentOS-7 - Updates	209

4. Install the Ambari bits. This also installs the default PostgreSQL Ambari database.

   ```
   yum install ambari-server
   ```

5. Enter y when prompted to to confirm transaction and dependency checks.

   A successful installation displays output similar to the following:

   ```
   Installing : postgresql-libs-8.4.20-3.el6_6.x86_64          1/4
   Installing : postgresql-8.4.20-3.el6_6.x86_64               2/4
   Installing : postgresql-server-8.4.20-3.el6_6.x86_64        3/4
   Installing : ambari-server-2.4.2.0-1470.x86_64                4/4
   Verifying  : ambari-server-2.4.2.0-1470.x86_64                1/4
   Verifying  : postgresql-8.4.20-3.el6_6.x86_64               2/4
   Verifying  : postgresql-server-8.4.20-3.el6_6.x86_64        3/4
   Verifying  : postgresql-libs-8.4.20-3.el6_6.x86_64          4/4

   Installed:
     ambari-server.x86_64 0:2.4.2.0-1470
   ```

```
Dependency Installed:
 postgresql.x86_64 0:8.4.20-3.el6_6
 postgresql-libs.x86_64 0:8.4.20-3.el6_6
 postgresql-server.x86_64 0:8.4.20-3.el6_6
```

Note

Accept the warning about trusting the Hortonworks GPG Key. That key will be automatically downloaded and used to validate packages from Hortonworks. You will see the following message:

```
Importing GPG key 0x07513CAD: Userid: "Jenkins (HDP
Builds) <jenkin@hortonworks.com>" From : http://
s3.amazonaws.com/dev.hortonworks.com/ambari/centos6/RPM-
GPG-KEY/RPM-GPG-KEY-Jenkins
```

Note

Ambari Server by default uses an embedded PostgreSQL database. When you install the Ambari Server, the PostgreSQL packages and dependencies must be available for install. These packages are typically available as part of your Operating System repositories. Please confirm you have the appropriate repositories available for the postgresql-server packages.

2.1.3. SLES 12

On a server host that has Internet access, use a command line editor to perform the following steps:

1. Log in to your host as `root`.
2. Download the Ambari repository file to a directory on your installation host.

   ```
   wget -nv http://public-repo-1.hortonworks.com/ambari/sles12/2.x/
   updates/2.4.2.0/ambari.repo -O /etc/zypp/repos.d/ambari.repo
   ```

   ### Important

   Do not modify the `ambari.repo` file name. This file is expected to be available on the Ambari Server host during Agent registration.

3. Confirm the downloaded repository is configured by checking the repo list.

   ```
   zypper repos
   ```

   You should see the Ambari repositories in the list.

   Version values vary, depending on the installation.

Alias	Name	Enabled	Refresh
AMBARI.2.4.2.0-2.x	Ambari 2.x	Yes	No
http-demeter.uni-regensburg.de-c997c8f9	SUSE-Linux-Enterprise-Software-Development-Kit-12-SP1 11.1.1-1.57	Yes	Yes

Alias	Name	Enabled	Refresh
opensuse	OpenSuse	Yes	Yes

4. Install the Ambari bits. This also installs the default PostgreSQL Ambari database.

   ```
   zypper install ambari-server
   ```

5. Enter y when prompted to to confirm transaction and dependency checks.

   A successful installation displays output similar to the following:

   Retrieving package postgresql-libs-8.3.5-1.12.x86_64 (1/4), 172.0 KiB (571.0 KiB unpacked)

   Retrieving: postgresql-libs-8.3.5-1.12.x86_64.rpm [done (47.3 KiB/s)]

   Installing: postgresql-libs-8.3.5-1.12 [done]

   Retrieving package postgresql-8.3.5-1.12.x86_64 (2/4), 1.0 MiB (4.2 MiB unpacked)

   Retrieving: postgresql-8.3.5-1.12.x86_64.rpm [done (148.8 KiB/s)]

   Installing: postgresql-8.3.5-1.12 [done]

   Retrieving package postgresql-server-8.3.5-1.12.x86_64 (3/4), 3.0 MiB (12.6 MiB unpacked)

   Retrieving: postgresql-server-8.3.5-1.12.x86_64.rpm [done (452.5 KiB/s)]

   Installing: postgresql-server-8.3.5-1.12 [done]

   Updating etc/sysconfig/postgresql...

   Retrieving package ambari-server-2.4.2.0-135.noarch (4/4), 99.0 MiB (126.3 MiB unpacked)

   Retrieving: ambari-server-2.4.2.0-135.noarch.rpm [done (3.0 MiB/s)]

   Installing: ambari-server-2.4.2.0-135 [done]

   ambari-server 0:off 1:off 2:off 3:on 4:off 5:on 6:off

Note

Ambari Server by default uses an embedded PostgreSQL database. When you install the Ambari Server, the PostgreSQL packages and dependencies must be available for install. These packages are typically available as part of your Operating System repositories. Please confirm you have the appropriate repositories available for the postgresql-server packages.

2.1.4. SLES 11

On a server host that has Internet access, use a command line editor to perform the following steps:

1. Log in to your host as `root`.

2. Download the Ambari repository file to a directory on your installation host.

   ```
   wget -nv http://public-repo-1.hortonworks.com/ambari/suse11/2.x/
   updates/2.4.2.0/ambari.repo -O /etc/zypp/repos.d/ambari.repo
   ```

   Important

   Do not modify the `ambari.repo` file name. This file is expected to be available on the Ambari Server host during Agent registration.

3. Confirm the downloaded repository is configured by checking the repo list.

   ```
   zypper repos
   ```

   You should see the Ambari repositories in the list.

   Version values vary, depending on the installation.

Alias	Name	Enabled	Refresh
AMBARI.2.4.2.0-2.x	Ambari 2.x	Yes	No
http-demeter.uni-regensburg.de-c997c8f9	SUSE-Linux-Enterprise-Software-Development-Kit-11-SP1 11.1.1-1.57	Yes	Yes
opensuse	OpenSuse	Yes	Yes

4. Install the Ambari bits. This also installs the default PostgreSQL Ambari database.

   ```
   zypper install ambari-server
   ```

5. Enter y when prompted to to confirm transaction and dependency checks.

   A successful installation displays output similar to the following:

   Retrieving package postgresql-libs-8.3.5-1.12.x86_64 (1/4), 172.0 KiB (571.0 KiB unpacked)

   Retrieving: postgresql-libs-8.3.5-1.12.x86_64.rpm [done (47.3 KiB/s)]

   Installing: postgresql-libs-8.3.5-1.12 [done]

   Retrieving package postgresql-8.3.5-1.12.x86_64 (2/4), 1.0 MiB (4.2 MiB unpacked)

   Retrieving: postgresql-8.3.5-1.12.x86_64.rpm [done (148.8 KiB/s)]

   Installing: postgresql-8.3.5-1.12 [done]

   Retrieving package postgresql-server-8.3.5-1.12.x86_64 (3/4), 3.0 MiB (12.6 MiB unpacked)

   Retrieving: postgresql-server-8.3.5-1.12.x86_64.rpm [done (452.5 KiB/s)]

   Installing: postgresql-server-8.3.5-1.12 [done]

Updating etc/sysconfig/postgresql...

Retrieving package ambari-server-2.4.2.0-135.noarch (4/4), 99.0 MiB (126.3 MiB unpacked)

Retrieving: ambari-server-2.4.2.0-135.noarch.rpm [done (3.0 MiB/s)]

Installing: ambari-server-2.4.2.0-135 [done]

ambari-server 0:off 1:off 2:off 3:on 4:off 5:on 6:off

Note

Ambari Server by default uses an embedded PostgreSQL database. When you install the Ambari Server, the PostgreSQL packages and dependencies must be available for install. These packages are typically available as part of your Operating System repositories. Please confirm you have the appropriate repositories available for the postgresql-server packages.

2.1.5. Ubuntu 12

On a server host that has Internet access, use a command line editor to perform the following steps:

1. Log in to your host as `root`.
2. Download the Ambari repository file to a directory on your installation host.

```
wget -nv http://public-repo-1.hortonworks.com/ambari/
ubuntu12/2.x/updates/2.4.2.0/ambari.list -O /etc/apt/
sources.list.d/ambari.list

apt-key adv --recv-keys --keyserver keyserver.ubuntu.com
B9733A7A07513CAD

apt-get update
```

Important

Do not modify the `ambari.list` file name. This file is expected to be available on the Ambari Server host during Agent registration.

3. Confirm that Ambari packages downloaded successfully by checking the package name list.

```
apt-cache showpkg ambari-server

apt-cache showpkg ambari-agent

apt-cache showpkg ambari-metrics-assembly
```

You should see the Ambari packages in the list.

4. Install the Ambari bits. This also installs the default PostgreSQL Ambari database.

```
apt-get install ambari-server
```

Note

Ambari Server by default uses an embedded PostgreSQL database. When you install the Ambari Server, the PostgreSQL packages and dependencies must be available for install. These packages are typically available as part of your Operating System repositories. Please confirm you have the appropriate repositories available for the postgresql-server packages.

2.1.6. Ubuntu 14

On a server host that has Internet access, use a command line editor to perform the following steps:

1. Log in to your host as `root`.
2. Download the Ambari repository file to a directory on your installation host.

```
wget -nv http://public-repo-1.hortonworks.com/ambari/
ubuntu14/2.x/updates/2.4.2.0/ambari.list -O /etc/apt/
sources.list.d/ambari.list
```

```
apt-key adv --recv-keys --keyserver keyserver.ubuntu.com
B9733A7A07513CAD
```

```
apt-get update
```

Important

Do not modify the `ambari.list` file name. This file is expected to be available on the Ambari Server host during Agent registration.

3. Confirm that Ambari packages downloaded successfully by checking the package name list.

```
apt-cache showpkg ambari-server
```

```
apt-cache showpkg ambari-agent
```

```
apt-cache showpkg ambari-metrics-assembly
```

You should see the Ambari packages in the list.

4. Install the Ambari bits. This also installs the default PostgreSQL Ambari database.

```
apt-get install ambari-server
```

Note

Ambari Server by default uses an embedded PostgreSQL database. When you install the Ambari Server, the PostgreSQL packages and dependencies

must be available for install. These packages are typically available as part of your Operating System repositories. Please confirm you have the appropriate repositories available for the postgresql-server packages.

2.1.7. Debian 7

On a server host that has Internet access, use a command line editor to perform the following steps:

1. Log in to your host as `root`.

2. Download the Ambari repository file to a directory on your installation host.

```
wget -nv http://public-repo-1.hortonworks.com/ambari/
debian7/2.x/updates/2.4.2.0/ambari.list -O /etc/apt/
sources.list.d/ambari.list

apt-key adv --recv-keys --keyserver keyserver.ubuntu.com
B9733A7A07513CAD

apt-get update
```

Important

Do not modify the `ambari.list` file name. This file is expected to be available on the Ambari Server host during Agent registration.

3. Confirm that Ambari packages downloaded successfully by checking the package name list.

```
apt-cache showpkg ambari-server

apt-cache showpkg ambari-agent

apt-cache showpkg ambari-metrics-assembly
```

You should see the Ambari packages in the list.

4. Install the Ambari bits. This also installs the default PostgreSQL Ambari database.

```
apt-get install ambari-server
```

Note

Ambari Server by default uses an embedded PostgreSQL database. When you install the Ambari Server, the PostgreSQL packages and dependencies must be available for install. These packages are typically available as part of your Operating System repositories. Please confirm you have the appropriate repositories available for the postgresql-server packages.

2.2. Set Up the Ambari Server

Before starting the Ambari Server, you **must** set up the Ambari Server. Setup configures Ambari to talk to the Ambari database, installs the JDK and allows you to customize the user account the Ambari Server daemon will run as. The `ambari-server setup` command manages the setup process. Run the following command on the Ambari server host to start the setup process.

You may also append Setup Options to the command.

```
ambari-server setup
```

Respond to the setup prompt:

1. If you have *not* temporarily disabled SELinux, you may get a warning. Accept the default (`y`), and continue.
2. By default, Ambari Server runs under `root`. Accept the default (`n`) at the `Customize user account for ambari-server daemon` prompt, to proceed as `root`. If you want to create a different user to run the Ambari Server, or to assign a previously created user, select `y` at the `Customize user account for ambari-server daemon` prompt, then provide a user name.
3. If you have not temporarily disabled `iptables` you may get a warning. Enter `y` to continue.
4. Select a JDK version to download. Enter `1` to download Oracle JDK 1.8. Alternatively, you can choose to enter a Custom JDK. If you choose Custom JDK, you must manually install the JDK on all hosts and specify the Java Home path.
5. Accept the Oracle JDK license when prompted. You must accept this license to download the necessary JDK from Oracle. The JDK is installed during the deploy phase.
6. Select `n` at `Enter advanced database configuration` to use the default, embedded PostgreSQL database for Ambari. The default PostgreSQL database name is `ambari`. The default user name and password are `ambari/bigdata`. Otherwise, to use an existing PostgreSQL, MySQL/MariaDB or Oracle database with Ambari, select `y`.
   - If you are using an existing PostgreSQL, MySQL/MariaDB, or Oracle database instance, use one of the following prompts:

     Important

     Using the **Microsoft SQL Server** or **SQL Anywhere** database options are not supported.

   - To use an existing Oracle instance, and select your own database name, user name, and password for that database, enter `2`.

     Select the database you want to use and provide any information requested at the prompts, including host name, port, Service Name or SID, user name, and password.

   - To use an existing MySQL/MariaDB database, and select your own database name, user name, and password for that database, enter `3`.

Select the database you want to use and provide any information requested at the prompts, including host name, port, database name, user name, and password.

- To use an existing PostgreSQL database, and select your own database name, user name, and password for that database, enter 4.

  Select the database you want to use and provide any information requested at the prompts, including host name, port, database name, user name, and password.

7. At Proceed with configuring remote database connection properties [y/n] choose y.

8. Setup completes.

2.2.1. Setup Options

The following table describes options frequently used for Ambari Server setup.

Option	Description
-j (or –java-home)	Specifies the JAVA_HOME path to use on the Ambari Server and all hosts in the cluster. By default when you do not specify this option, Ambari Server setup downloads the Oracle JDK 1.8 binary and accompanying Java Cryptography Extension (JCE) Policy Files to /var/lib/ambari-server/resources. Ambari Server then installs the JDK to /usr/jdk64. Use this option when you plan to use a JDK other than the default Oracle JDK 1.8. See Supported JDKs for more information on the supported JDKs. If you are using an alternate JDK, you must manually install the JDK on all hosts and specify the Java Home path during Ambari Server setup. If you plan to use Kerberos, you must also install the JCE on all hosts. This path must be valid on all hosts. For example: `ambari-server setup –j /usr/java/default`
–jdbc-driver	Should be the path to the JDBC driver JAR file. Use this option to specify the location of the JDBC driver JAR and to make that JAR available to Ambari Server for distribution to cluster hosts during configuration. Use this option with the –jdbc-db option to specify the database type.
–jdbc-db	Specifies the database type. Valid values are: [postgres | mysql | oracle] Use this option with the –jdbc-driver option to specify the location of the JDBC driver JAR file.
-s (or –silent)	Setup runs silently. Accepts all the default prompt values, such as: • User account "root" for the ambari-server • Oracle 1.8 JDK (which is installed at /usr/jdk64). This can be overridden by adding the -j option and specifying an existing JDK path. • Embedded PostgreSQL for Ambari DB (with database name "ambari")  **Important** By choosing the silent setup option and by not overriding the JDK selection, Oracle JDK will be installed and you will be agreeing to the Oracle Binary Code License agreement. Do not use this option if you do not agree to the license terms. The license terms can be found here: http://www.oracle.com/technetwork/java/javase/terms/license/index.html If you want to run the Ambari Server as non-root, you must run setup in interactive mode. When prompted to customize the ambari-server user account, provide the account information.
-v (or –verbose)	Prints verbose info and warning messages to the console during Setup.
-g (or –debug)	Prints debug info to the console during Setup.

2.3. Install the HDF Management Pack

A management pack (MPack) bundles service definitions, stack definitions, and stack add-on service definitions so they do not need to be included with the Ambari core functionality and can be updated in between major releases. The HDF management pack includes Ambari services allowing you to deploy:

- NiFi
- Storm
- Kafka
- ZooKeeper
- Ranger
- Ambari Infra
- Ambari Metrics
- Log Search
- Kerberos support

Warning

Do not install the HDF management pack on a system where HDP is already installed.

1. Back up your Ambari resources folder:

```
cp -r /var/lib/ambari-server/resources /var/lib/ambari-server/resources.
backup
```

2. Install the HDF managemet pack for your OS:

   - **CentOS 6**

```
ambari-server install-mpack \
--mpack=http://public-repo-1.hortonworks.com/HDF/centos6/2.x/updates/2.1.
2.0/tars/hdf_ambari_mp/hdf-ambari-mpack-2.1.2.0-10.tar.gz \
--purge \
--verbose
```

   - **CentOS 7**

```
ambari-server install-mpack \
--mpack=http://public-repo-1.hortonworks.com/HDF/centos7/2.x/updates/2.1.
2.0/tars/hdf_ambari_mp/hdf-ambari-mpack-2.1.2.0-10.tar.gz \
--purge \
--verbose
```

   - **SUSE Enterprise Linux 11, SP3 and SP4**

```
ambari-server install-mpack \
```

```
--mpack=http://public-repo-1.hortonworks.com/HDF/suse11sp3/2.x/updates/2.
1.2.0/tars/hdf_ambari_mp/hdf-ambari-mpack-2.1.2.0-10.tar.gz \
--purge \
--verbose
```

- **SUSE Linux Enterprise Server (SLES) v12 SP1**

```
ambari-server install-mpack \
--mpack=http://public-repo-1.hortonworks.com/HDF/sles12/2.x/updates/2.1.2.
0/tars/hdf_ambari_mp/hdf-ambari-mpack-2.1.2.0-10.tar.gz \
--purge \
--verbose
```

- **Debian 6**

```
ambari-server install-mpack \
--mpack=http://public-repo-1.hortonworks.com/HDF/debian6/2.x/updates/2.1.
2.0/tars/hdf_ambari_mp/hdf-ambari-mpack-2.1.2.0-10.tar.gz \
--purge \
--verbose
```

- **Debian 7**

```
ambari-server install-mpack \
--mpack=http://public-repo-1.hortonworks.com/HDF/debian7/2.x/updates/2.1.
2.0/tars/hdf_ambari_mp/hdf-ambari-mpack-2.1.2.0-10.tar.gz \
--purge \
--verbose
```

- **Ubuntu 12**

```
ambari-server install-mpack \
--mpack=http://public-repo-1.hortonworks.com/HDF/ubuntu12/2.x/updates/2.1.
2.0/tars/hdf_ambari_mp/hdf-ambari-mpack-2.1.2.0-10.tar.gz \
--purge \
--verbose
```

- **Ubuntu 14**

```
ambari-server install-mpack \
--mpack=http://public-repo-1.hortonworks.com/HDF/ubuntu14/2.x/updates/2.1.
2.0/tars/hdf_ambari_mp/hdf-ambari-mpack-2.1.2.0-10.tar.gz \
--purge \
--verbose
```

2.4. Start the Ambari Server

- Run the following command on the Ambari Server host:

  ```
  ambari-server start
  ```

- To check the Ambari Server processes:

  ```
  ambari-server status
  ```

- To stop the Ambari Server:

  ```
  ambari-server stop
  ```

On Ambari Server start, Ambari runs a database consistency check looking for issues. If any issues are found, Ambari Server **start will abort** and a message will be printed to console "DB configs consistency check failed." More details will be written to the following log file:

```
/var/log/ambari-server/ambari-server-check-database.log
```

You can force Ambari Server to start by skipping this check with the following option:

```
ambari-server start --skip-database-check
```

If you have database issues, by choosing to skip this check, **do not make any changes to your cluster topology or perform a cluster upgrade until you correct the database consistency issues**. Please contact Hortonworks Support and provide the `ambari-server-check-database.log` output for assistance.

3. Deploying HDF using Ambari

You can use the Ambari installation wizard to deploy HDF in a single or multi-node cluster.

3.1. Launching the Ambari Install Wizard

After you have started the Ambari Service, you can open Ambari Web in a browser and launch the Install Wizard.

1. Point your browser to `http://<your.ambari.server>:8080`,where <your.ambari.server> is the name of your ambari server host.
2. Log in to the Ambari Server using the default user name/password: admin/admin. You can change these credentials later.
3. From the Ambari Welcome page, choose Launch Install Wizard.

3.2. Installing HDF using Ambari

After launching the Install Wizard, proceed through the Wizard and provide the relevant information.

1. From **Name Your Cluster**, specify the name of the cluster you want to create. Cluster names do not support special characters or white spaces.
2. From **Select Version**:
   - Select the software version you want to install. HDF 2.0 is the only option available.
   - Select Use Public Repository.
   - Provide the HDF repo location without specifying the .repo file extension. For information on the repo location for your OS, see HDF Repoitory Locations.
   - Remove the options for operating systems that do not pertain to your installation.
3. From **Install Options**, enter:
   - One or more hosts on to which you want you install, specified with an FQDN.
   - Your SSH Private Key, or select the option to perform manual registrations on hosts.
4. From **Confirm Hosts**, review the hosts you have selected and click **Next**.
5. From **Choose Services**, select the services you want to install and click Next.
6. From **Assign Masters**, use the plus (+) and minus (-) buttons to assign components to the hosts you want to run them.

   If you are installing NiFi, you can choose the hosts on which you want NiFi to run from this dialog, or you can add them later using the **Add Services** dialog.

7. From **Slaves and Clients**, add slave and client components to the hosts you want to run them.

8. From **Customize Services**, provide the required information for each service you want to install.

   Browse the tabs to review and modify your HDF cluster setup information. The wizard attempts to set reasonable defaults for each of the options, but you are strongly encouraged to review these settings before proceeding.

   In each service tab, hovering your cursor over the properties displays a brief description of what the property does. The number of service tabs shown depends on the services you decided to install in your cluster. Any tab that requires input shows a red badge with the number of properties that need attention. Select each service tab that displays a red badge number and enter the appropriate information. Pay special attention to the values for the following:

Directories	The choice of directories where HDF stores information is critical. Ambari attempts to choose reasonable defaults based on the mount points available in your environment but you are should review the default directory settings.
Service Account Users and Groups	The service account users and groups are available under the Misc tab. These are the operating system accounts the service components will run as. If these users do not exist on your hosts, Ambari automatically creates the users and groups locally on the hosts. If these users already exist, Ambari uses those accounts. Depending on how your environment is configured, you might not allow groupmod or usermod operations. If this is the case, you must be sure all users and groups are already created and be sure to select the "Skip group modifications" option on the Misc tab. This tells Ambari to not modify group membership for the service users.
NiFi	You can enable SSL in the NiFi Customized Services tab. For more information, see the Enabling SSL section below.

9. When you have finished customizing each of your services, review your information and click **Next** to install HDF.

10. When `Successfully installed and started the services` appears, click Next.

4. Installing Using a Local Repository

Local repositories are frequently used in enterprise clusters that have limited outbound internet access. In these scenarios, having packages available locally provides more governance, and better installation performance. These repositories are used heavily during installation for package distribution, as well as post-install for routine cluster operations such as service start/restart operations. The following section describes the steps required to setup and use a local repository:

- Obtain the repositories
- Set up a local repository having:
  - No Internet Access
  - Temporary Internet Access
- Prepare the Ambari repository configuration file

4.1. Obtaining the Repositories

This section describes how to obtain:

- Ambari 2.4.2 Repositories
- HDF 2.1 Repositories

4.1.1. Ambari 2.4.2 Repositories

Obtain the Ambari repository information from the Ambari Installation documentation.

If you do not have Internet access, use the link appropriate for your OS family to **download a tarball** that contains the software for setting up Ambari.

If you have temporary Internet access, use the link appropriate for your OS family to **download a repository file** that contains the software for setting up Ambari.

4.1.2. HDF 2.1 Repositories

For HDF repo and additional download locations, see the HDF Release Notes.

If you do not have Internet access, use the link appropriate for your OS family to download a tarball that contains the software for setting up the Stack.

If you have temporary Internet access, use the link appropriate for your OS family to download a repository file that contains the software for setting up the Stack.

4.2. Setting Up a Local Repository

Based on your Internet access, choose one of the following options:

- No Internet Access

  This option involves downloading the repository tarball, moving the tarball to the selected mirror server in your cluster, and extracting to create the repository.

- Temporary Internet Access

  This option involves using your temporary Internet access to sync (using reposync) the software packages to your selected mirror server and creating the repository.

  Both options proceed in a similar, straightforward way. Setting up for each option presents some key differences, as described in the following sections:

  - Getting Started Setting Up a Local Repository
  - Setting Up a Local Repository with No Internet Access
  - Setting Up a Local Repository with Temporary Internet Access

4.3. Getting Started Setting Up a Local Repository

To get started setting up your local repository, complete the following prerequisites:

- Select an existing server in, or accessible to the cluster, that runs a supported operating system.
- Enable network access from all hosts in your cluster to the mirror server.
- Ensure the mirror server has a package manager installed such as yum (RHEL / CentOS / Oracle Linux), zypper (SLES), or apt-get (Debian/Ubuntu).
- **Optional:** If your repository has temporary Internet access, and you are using RHEL/ CentOS/Oracle Linux as your OS, install yum utilities:

  ```
  yum install yum-utils createrepo
  ```

1. Create an HTTP server.

   a. On the mirror server, install an HTTP server (such as Apache httpd) using the instructions provided here .

   b. Activate this web server.

   c. Ensure that any firewall settings allow inbound HTTP access from your cluster nodes to your mirror server.

   Note

   If you are using Amazon EC2, make sure that SELinux is disabled.

2. On your mirror server, create a directory for your web server.

   - For example, from a shell window, type:

     - **For RHEL/CentOS/Oracle Linux:**

```
mkdir -p /var/www/html/
```

- **For SLES:**

```
mkdir -p /srv/www/htdocs/rpms
```

- **For Debian/Ubuntu:**

```
mkdir -p /var/www/html/
```

- If you are using a symlink, enable the `followsymlinks` on your web server.

4.3.1. Setting Up a Local Repository with No Internet Access

After completing the Getting Started Setting up a Local Repository procedure, finish setting up your repository by completing the following steps:

1. Obtain the tarball for the repository you would like to create. For options, see Obtaining the Repositories.

2. Copy the repository tarballs to the web server directory and untar.

   a. Browse to the web server directory you created.

   - **For RHEL/CentOS/Oracle Linux:**

```
cd /var/www/html/
```

   - **For SLES:**

```
cd /srv/www/htdocs/rpms
```

   - **For Debian/Ubuntu:**

```
cd /var/www/html/
```

   b. Untar the repository tarballs to the following locations: where <web.server>, <web.server.directory>, <OS>, <version>, and <latest.version> represent the name, home directory, operating system type, version, and most recent release version, respectively.

   Untar Locations for a Local Repository - No Internet Access

Repository Content	Repository Location
Ambari Repository	Untar under <web.server.directory>
HDF Repositories	Create directory and untar under <web.server.directory>/hdf

3. Confirm you can browse to the newly created local repositories.

   URLs for a Local Repository - No Internet Access

Repository	Base URL
Ambari Base URL	http://<web.server>/Ambari-2.4.2.0/<OS>

Repository	Base URL
HDF Base URL	http://<web.server>/hdf/HDF/<OS>/2.x/updates/<latest.version>
HDP-UTILS Base URL	http://<web.server>/hdf/HDP-UTILS-<version>/repos/<OS>

where <web.server> = FQDN of the web server host, and <OS> is centos6, centos7, sles11, sles12, ubuntu12, ubuntu14, or debian7.

Important

Be sure to record these Base URLs. You will need them when installing Ambari and the cluster.

4. Optional: If you have multiple repositories configured in your environment, deploy the following plug-in on all the nodes in your cluster.

   - Install the plug-in.

     - **For RHEL and CentOS 7:**

       ```
       yum install yum-plugin-priorities
       ```

     - **For RHEL and CentOS 6:**

       ```
       yum install yum-plugin-priorities
       ```

   - Edit the `/etc/yum/pluginconf.d/priorities.conf` file to add the following:

     ```
     [main]
     enabled=1
     gpgcheck=0
     ```

4.3.2. Setting up a Local Repository With Temporary Internet Access

After completing the Getting Started Setting up a Local Repository procedure, finish setting up your repository by completing the following steps:

1. Put the repository configuration files for Ambari and the Stack in place on the host. For options, see Obtaining the Repositories.

2. Confirm availability of the repositories.

   - **For RHEL/CentOS/Oracle Linux:**

     ```
     yum repolist
     ```

   - **For SLES:**

     ```
     zypper repos
     ```

   - **For Debian/Ubuntu:**

```
dpkg-list
```

3. Synchronize the repository contents to your mirror server.

   - Browse to the web server directory:

     - **For RHEL/CentOS/Oracle Linux:**

       ```
       cd /var/www/html
       ```

     - **For SLES:**

       ```
       cd /srv/www/htdocs/rpms
       ```

     - **For Debain/Ubuntu:**

       ```
       cd /var/www/html
       ```

   - For Ambari, create `ambari` directory and reposync.

     ```
     mkdir -p ambari/<OS>

     cd ambari/<OS>

     reposync -r Updates-Ambari-2.4.2.0
     ```

     where <OS> is centos6, centos7, sles11, sles12, ubuntu12, ubuntu14, or debian7.

   - For HDF Repositories, create `hdf` directory and reposync.

     ```
     mkdir -p hdf/<OS>

     cd hdf/<OS>

     reposync -r HDF-<latest.version>

     reposync -r HDP-UTILS-<version>
     ```

4. Generate the repository metadata.

   - For Ambari:

     ```
     createrepo <web.server.directory>/ambari/<OS>/Updates-
     Ambari-2.4.2.0
     ```

   - For HDF Repositories:

     ```
     createrepo <web.server.directory>/hdf/<OS>/HDF-<latest.version>

     createrepo <web.server.directory>/hdf/<OS>/HDP-UTILS-<version>
     ```

5. Confirm that you can browse to the newly created repository.

Repository	Base URL
Ambari Base URL	http://<web.server>/ambari/<OS>/Updates-Ambari-2.4.2.0
HDF Base URL	http://<web.server>/hdf/<OS>/HDF-<latest.version>
HDP-UTILS Base URL	http://<web.server>/hdf/<OS>/HDP-UTILS-<version>

where <web.server> = FQDN of the web server host, and <OS> is centos6, centos7, sles11, sles12, ubuntu12, ubuntu14, or debian7.

Important

Be sure to record these Base URLs. You will need them when installing Ambari and the Cluster.

6. Optional. If you have multiple repositories configured in your environment, deploy the following plug-in on all the nodes in your cluster.

   - Install the plug-in.

     - **For RHEL and CentOS 7:**

       ```
       yum install yum-plugin-priorities
       ```

     - **For RHEL and CentOS 6:**

       ```
       yum install yum-plugin-priorities
       ```

   - Edit the `/etc/yum/pluginconf.d/priorities.conf` file to add the following:

     ```
     [main]

     enabled=1

     gpgcheck=0
     ```

4.4. Preparing The Ambari Repository Configuration File

1. Download the `ambari.repo` file from the public repository.

   ```
   http://public-repo-1.hortonworks.com/ambari/<OS>/2.x/
   updates/2.4.2.0/ambari.repo
   ```

   where <OS> is centos6, centos7, sles11, sles12, ubuntu12, ubuntu14, or debian7.

2. Edit the ambari.repo file and replace the Ambari Base URL `baseurl` obtained when setting up your local repository. Refer to step 3 in Setting Up a Local Repository with No Internet Access, or step 5 in Setting Up a Local Repository with Temporary Internet Access, if necessary.

Note

You can disable the GPG check by setting gpgcheck =0. Alternatively, you can keep the check enabled but replace the gpgkey with the URL to the GPG-KEY in your local repository.

```
[Updates-Ambari-2.4.2.0]

name=Ambari-2.4.2.0-Updates

baseurl=INSERT-BASE-URL

gpgcheck=1

gpgkey=http://public-repo-1.hortonworks.com/ambari/centos6/RPM-
GPG-KEY/RPM-GPG-KEY-Jenkins

enabled=1

priority=1
```

Base URL for a Local Repository

Local Repository	Base URL
Built with Repository Tarball (No Internet Access)	http://<web.server>/Ambari-2.4.2.0/<OS>
Built with Repository File (Temporary Internet Access)	http://<web.server>/ambari/<OS>/Updates-Ambari-2.4.2.0

where <web.server> = FQDN of the web server host, and <OS> is centos6, centos7, sles11, sles12, ubuntu12, ubuntu14, or debian7.

3. Place the ambari.repo file on the machine you plan to use for the Ambari Server.

   - **For RHEL/CentOS/Oracle Linux:**

     `/etc/yum.repos.d/ambari.repo`

   - **For SLES:**

     `/etc/zypp/repos.d/ambari.repo`

   - **For Debain/Ubuntu:**

     `/etc/apt/sources.list.d/ambari.list`

   - Edit the `/etc/yum/pluginconf.d/priorities.conf` file to add the following:

     ```
     [main]

     enabled=1
     ```

```
gpgcheck=0
```

4. Proceed to Installing Ambari to install and setup Ambari Server.

5. Enabling SSL for NiFi

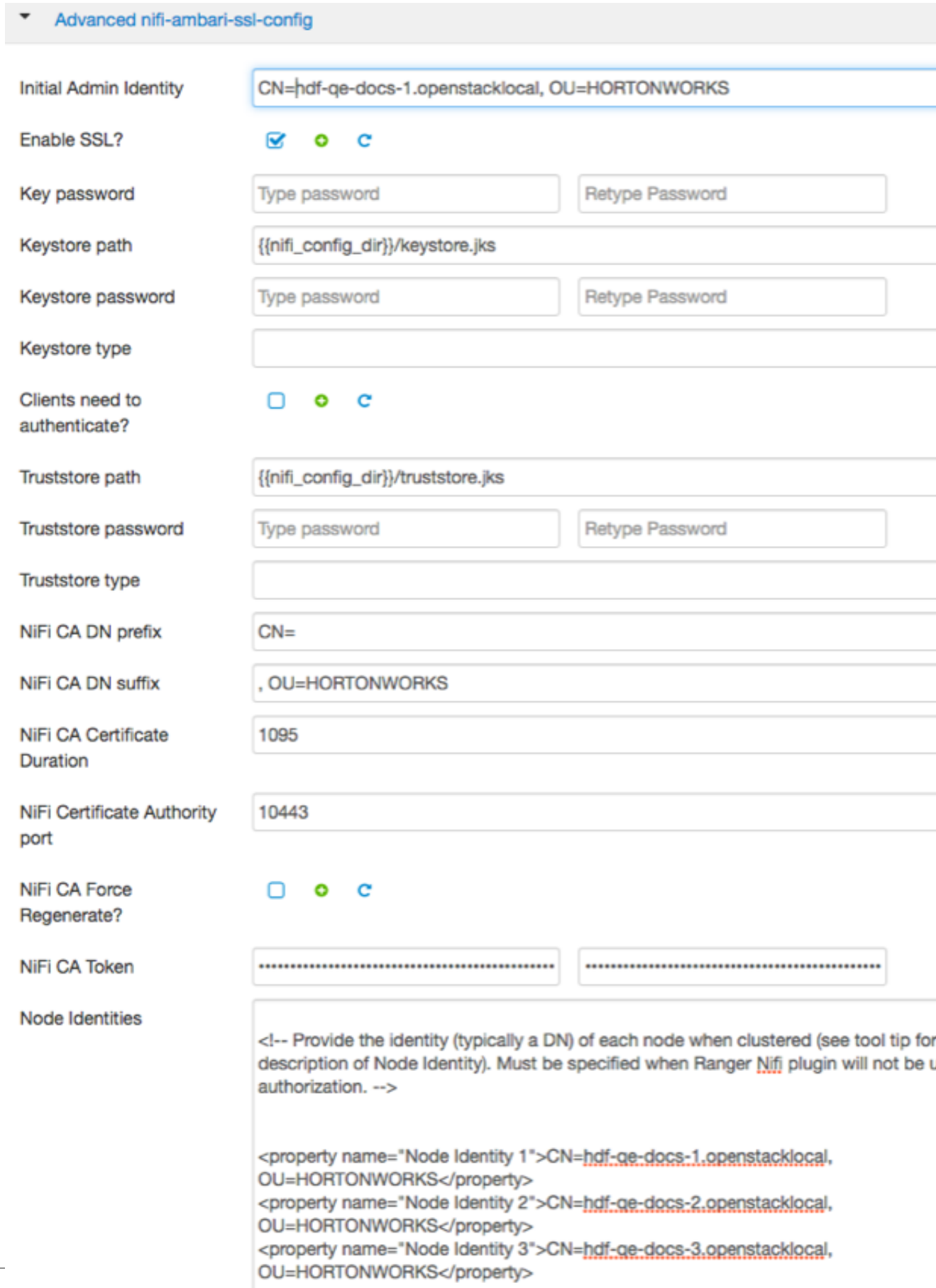
After you have installed Ambari and the HDF stack, you have a 2 options for enabling SSL for your NiFi services.

- Enabling SSL with a NiFi Certificate Authority
- Enabling SSL without a NIFI Certificate Authority

You can use the NiFi service Configs tab Advanced nifi-ambari-ssl-config dialog to configure security for these options.

To access the NiFi SSL configuration dialog:

1. From the Ambari services column, click NiFi.
2. Click the Configs tabs.
3. Click Advanced nifi-ambari-ssl-config.


Advanced nifi-ambari-ssl-config
Initial Admin Identity
CN=hdf-qe-docs-1.openstacklocal, OU=HORTONWORKS
Enable SSL?
Key password
Type password
Retype Password
Keystore path
{{nifi_config_dir}}/keystore.jks
Keystore password
Type password
Retype Password
Keystore type
Clients need to authenticate?
Truststore path
{{nifi_config_dir}}/truststore.jks
Truststore password
Type password
Retype Password
Truststore type
NiFi CA DN prefix
CN=
NiFi CA DN suffix
, OU=HORTONWORKS
NiFi CA Certificate Duration
1095
NiFi Certificate Authority port
10443
NiFi CA Force Regenerate?
NiFi CA Token
Node Identities
<!-- Provide the identity (typically a DN) of each node when clustered (see tool tip for
description of Node Identity). Must be specified when Ranger Nifi plugin will not be u
authorization. -->
<property name="Node Identity 1">CN=hdf-qe-docs-1.openstacklocal,
OU=HORTONWORKS</property>
<property name="Node Identity 2">CN=hdf-qe-docs-2.openstacklocal,
OU=HORTONWORKS</property>
<property name="Node Identity 3">CN=hdf-qe-docs-3.openstacklocal,
OU=HORTONWORKS</property>

5.1. Enabling SSL with a NiFi Certificate Authority

When you enable SSL with the NiFi Certificate Authority (CA) installed, the NiFi CA generates new client certificates for you through Ambari. If you want to enable SSL with a NiFi CA installed, and are planning to use Ranger to manage authorization:

1. Select the **Enable SSL?** box.
2. Specify the **NiFi CA** token.

If you want to enable SSL with a NiFi CA installed, and are not yet using Ranger to manage authorization:

1. Check the **Enable SSL?** box.
2. Specify the **NiFi CA Token**.
3. Verify that the `authorizations.xml` file on each node does not contain policies. The `authorizations.xml` is located in `{nifi_internal_dir}/conf`. By default, this location is `/var/lib/nifi/conf/`, and the value of `{nifi_internal_dir}` is specified in the **NiFi internal dir** field under **Advanced nifi-ambari-config**.

   Note

   If `authorizations.xml` does contain policies, you must delete it from each node. If you do not, your **Initial Admin Identity** and **Node Identities** changes do not take effect.

4. Specify the **Initial Admin Identity**. The **Initial Admin Identity** is the identity of an initial administrator and is granted access to the UI and has the ability to create additional users, groups, and policies. **This is a required value** when you are not using the Ranger plugin for NiFi for authorization.

   The **Initial Admin Identity** format is CN=admin, OU=NIFI.

   After you have added the **Initial Admin Identity**, you must immediately generate certificate for this user.

5. Specify the **Node Identities**. This indicates the identity of each node in a NiFi cluster and allows clustered nodes to communicate. **This is a required value** when you are not using the Ranger plugin for NiFi for authorization.

   ```
   <property name="Node Identity 1">CN=node1.fqdn, OU=NIFI</property>
   <property name="Node Identity 2">CN=node2.fqdn, OU=NIFI</property>
   <property name="Node Identity 3">CN=node3.fqdn, OU=NIFI</property>
   ```

   Replace node1.fqdn, node2.fqdn, and node3.fqdn with their respective fully qualified domain names.

5.2. Enabling SSL with Existing Certificates

If you want to enable SSL with existing certificates, and plan to use Ranger for authorization:

1. Check the **Enable SSL?** box.
2. Set **Keystore path**, **Keystore password**, and **Keystore type** values.

   The keystore path is similar to: /etc/security/nifi-certs/keystore.jks
3. Set the **Truststore path**, **Truststore password**, and **Truststore type** values.

   The truststore path is similar to: /etc/security/nifi-certs/truststore.jks
4. Check **Clients need to authenticate?** if you want to ensure that nodes in the cluster are authenticated and are required to have certificates that are trusted by the truststores.

If you want to enable SSL with existing certificates, and are not yet using Ranger for authorization:

1. Check the **Enable SSL?** box.
2. Set **Keystore path**, **Keystore password**, and **Keystore type** values.

   The keystore path is similar to: /etc/security/nifi-certs/keystore.jks

   This is a required value when you are not using the Ranger plugin for NiFi for authorization.
3. Set the **Truststore path**, **Truststore password**, and **Truststore type** values.

   The truststore path is similar to: /etc/security/nifi-certs/truststore.jks

   This is a required value when you are not using the Ranger plugin for NiFi for authorization.
4. Check **Clients need to authenticate?** to ensure that nodes in the cluster are authenticated and are required to have certificates that are trusted by the Truststores.
5. Specify the **Initial Admin Identity**. The **Initial Admin Identity** is the identity of an initial administrator and is granted access to the UI and has the ability to create additional users, groups, and policies. **This is a required value** when you are not using the Ranger plugin for NiFi for authorization.

   The **Initial Admin Identity** format is CN=admin, OU=NIFI.

   After you have added the **Initial Admin Identity**, you must immediately generate certificate for this user.
6. Specify the **Node Identities**. This indicates the identity of each node in a NiFi cluster and allows clustered nodes to communicate. **This is a required value** when you are not using the Ranger plugin for NiFi for authorization.

   ```
   <property name="Node Identity 1">CN=node1.fqdn, OU=NIFI</property>
   <property name="Node Identity 2">CN=node2.fqdn, OU=NIFI</property>
   <property name="Node Identity 3">CN=node3.fqdn, OU=NIFI</property>
   ```

   Replace node1.fqdn, node2.fqdn, and node3.fqdn with their respective fully qualified domain names.

5.3. (Optional) Setting Up Identity Mapping

To set up identity mapping:

1. From the NiFi service **Configs** tab, click **Advanced nifi-properties**.
2. Use the Filter box to search for nifi.security.identity.mapping.pattern.
3. Enter the following values:

Table 5.1. Identity mapping values

Field	Sample value
nifi.security.identity.mapping.pattern.dn	^CN=(.*?), OU=(.*?)$
nifi.security.identity.mapping.value.dn	$1@$2
nifi.security.identity.mapping.pattern.kerb	^(.*?)/instance@(.*?)$
nifi.security.identity.mapping.value.kerb	$1@$2

4. Click **Save**.
5. Restart NiFi using the **Restart all Required** option from the **Action** menu.

5.4. Generating Client Certificates

If you are using a CA, you can use the TLS Toolkit provided in the HDF management pack to generate the required client certificates so that you can log into NiFi after enabling SSL.

1. Navigate the TLS Toolkit directory:

```
cd /var/lib/ambari-agent/cache/common-services/NIFI/1.0.0/
package/files/nifi-toolkit-1.0.0.2.0.0.0-579
```

2. From the command line, run the following:

```
bin/tls-toolkit.sh client
-c <CA host name>
-D "<distinguished name>"
-p <CA host port>
-t <NiFi CA token>
-T <keystore type>
```

Your command should look similar to:

```
bin/tls-toolkit.sh client
-c nifi.cert.authority.example.com
-D "CN=admin, OU=NIFI"
-t nifi
-p 10443
-T pkcs12
```

3. To get your keystore password, enter:

```
cat config.json
```

4. Verify that the installation directory contains the following two files:

- `keystore.pkcs12`
- `nifi-cert.pem`

5. To double-click your keystore file to launch your OS certificate management application, change `keystore.pkcs12` to `keystore.p12`.
6. Import the `nifi-cert.pem` file as your trusted CA.
7. Import `keystore.pkcs12` as the client certificate.

Re-running the TLS Toolkit generates a new set of keystore and configuration files. To avoid having your files overwritten, save the keystore and configuration files to an alternate location before re-running the TLS Toolkit.

For more information about the TLS Toolkit, see TLS Generation Toolkit in the *Administration Guide*.

5.5. Logging into NiFi After Enabling SSL

Now that you have set up SSL, you need to enable logging into NiFi with a certificate:

1. Launch NiFi from the Ambari **Quick Links** menu.
2. Select the certificate you just imported from the browser prompt.
3. Log in with the user name and password you created during installation.

Note

When you are running NiFi on a host with Ambari and with SSL enabled, the default URL becomes secured **https**://<local-host>:**9091**/nifi.

6. Installing and Using Ranger

6.1. Installing Ranger Using Ambari

6.1.1. Overview

Apache Ranger can be installed either manually or using the Ambari UI. Unlike the manual installation process, which requires you to perform a number of installation steps, installing Ranger using the Ambari UI is simpler and easier. Once Ambari has been installed and configured, you can use the **Add Service** wizard to install the following components:

- Ranger Admin
- Ranger UserSync
- Ranger Key Management Service

6.1.2. Installation Prerequisites

Before you install Ranger, make sure your cluster meets the following requirements:

- You have installed Log Search or have an external Solr running.
- A MySQL, Oracle, or PostgreSQL database instance must be running and available to be used by Ranger. Configuration of the database instance for Ranger is described in the following sections for some of the databases supported by Ranger.
  - Configuring MySQL for Ranger
  - Configuring PostgreSQL for Ranger
  - Configuring Oracle for Ranger
- If you choose not to provide system Database Administrator (DBA) account details to the Ambari Ranger installer, you can use the `dba_script.py` Python script to create Ranger DB database users without exposing DBA account information to the Ambari Ranger installer. You can then run the normal Ambari Ranger installation without specifying a DBA user name and password. For more information see Setting up Database Users Without Sharing DBA Credentials.
- The Ranger installation creates two new users (default names: rangeradmin and rangerlogger) and two new databases (default names: ranger and ranger_audit).

6.1.2.1. Configuring MySQL for Ranger

1. The MySQL database administrator should be used to create the Ranger databases.

   The following series of commands could be used to create the `rangerdba` user with password `rangerdba`.

   a. Log in as the root user, then use the following commands to create the `rangerdba` user and grant it adequate privileges.

```
CREATE USER 'rangerdba'@'localhost' IDENTIFIED BY 'rangerdba';

GRANT ALL PRIVILEGES ON *.* TO 'rangerdba'@'localhost';

CREATE USER 'rangerdba'@'%' IDENTIFIED BY 'rangerdba';

GRANT ALL PRIVILEGES ON *.* TO 'rangerdba'@'%';

GRANT ALL PRIVILEGES ON *.* TO 'rangerdba'@'localhost' WITH GRANT OPTION;

GRANT ALL PRIVILEGES ON *.* TO 'rangerdba'@'%' WITH GRANT OPTION;

FLUSH PRIVILEGES;
```

b. Use the `exit` command to exit MySQL.

c. You should now be able to reconnect to the database as `rangerdba` using the following command:

```
mysql -u rangerdba -prangerdba
```

After testing the `rangerdba` login, use the `exit` command to exit MySQL.

2. Use the following command to confirm that the `mysql-connector-java.jar` file is in the Java share directory. This command must be run on the server where Ambari server is installed.

```
ls /usr/share/java/mysql-connector-java.jar
```

If the file is not in the Java share directory, use the following command to install the MySQL connector .jar file.

RHEL/CentOS/Oracle Linux

```
yum install mysql-connector-java*
```

SLES

```
zypper install mysql-connector-java*
```

3. Use the following command format to set the `jdbc/driver/path` based on the location of the MySQL JDBC driver .jar file.This command must be run on the server where Ambari server is installed.

```
ambari-server setup --jdbc-db={database-type} --jdbc-driver={/jdbc/driver/
path}
```

For example:

```
ambari-server setup --jdbc-db=mysql --jdbc-driver=/usr/share/java/mysql-
connector-java.jar
```

6.1.2.2. Configuring PostgreSQL for Ranger

1. On the PostgreSQL host, install the applicable PostgreSQL connector.

RHEL/CentOS/Oracle Linux

```
yum install postgresql-jdbc*
```

SLES

```
zypper install -y postgresql-jdbc
```

2. Confirm that the .jar file is in the Java share directory.

```
ls /usr/share/java/postgresql-jdbc.jar
```

3. Change the access mode of the .jar file to 644.

```
chmod 644 /usr/share/java/postgresql-jdbc.jar
```

4. The PostgreSQL database administrator should be used to create the Ranger databases.

   The following series of commands could be used to create the `rangerdba` user and grant it adequate privileges.

```
echo "CREATE DATABASE $dbname;" | sudo -u $postgres psql -U postgres
echo "CREATE USER $rangerdba WITH PASSWORD '$passwd';" | sudo -u $postgres
 psql -U postgres
echo "GRANT ALL PRIVILEGES ON DATABASE $dbname TO $rangerdba;" | sudo -u
 postgres psql -U $postgres
```

   Where:

   - `$postgres` is the Postgres user.
   - `$dbname` is the name of your PostgreSQL database

5. Use the following command format to set the `jdbc/driver/path` based on the location of the PostgreSQL JDBC driver .jar file. This command must be run on the server where Ambari server is installed.

```
ambari-server setup --jdbc-db={database-type} --jdbc-driver={/jdbc/driver/
path}
```

   For example:

```
ambari-server setup --jdbc-db=postgres --jdbc-driver=/usr/share/java/
postgresql.jar
```

6. Run the following command:

```
export HADOOP_CLASSPATH=${HADOOP_CLASSPATH}:${JAVA_JDBC_LIBS}:/connector jar
 path
```

7. Add Allow Access details for Ranger users:

   - change `listen_addresses='localhost'` to `listen_addresses='*' ('*' = any)` to listen from all IPs in `postgresql.conf`.
   - Make the following changes to the Ranger db user and Ranger audit db user in the `pg_hba.conf` file.

8. After editing the `pg_hba.conf` file, run the following command to refresh the PostgreSQL database configuration:

```
sudo -u postgres /usr/bin/pg_ctl -D $PGDATA reload
```

For example, if the `pg_hba.conf` file is located in the `/var/lib/pgsql/data` directory, the value of $PGDATA is `/var/lib/pgsql/data`.

6.1.2.3. Configuring Oracle for Ranger

1. On the Oracle host, install the appropriate JDBC .jar file.
   - Download the Oracle JDBC (OJDBC) driver from http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html.
   - For **Oracle Database 11g**: select Oracle Database 11g Release 2 drivers > ojdbc6.jar.
   - For **Oracle Database 12c**: select Oracle Database 12c Release 1 driver > ojdbc7.jar.
   - Copy the .jar file to the Java share directory. For example:

     ```
     cp ojdbc7.jar /usr/share/java
     ```

     Note

     Make sure the .jar file has the appropriate permissions. For example:

     ```
     chmod 644 /usr/share/java/ojdbc7.jar
     ```

2. The Oracle database administrator should be used to create the Ranger databases.

   The following series of commands could be used to create the RANGERDBA user and grant it permissions using SQL*Plus, the Oracle database administration utility:

```
# sqlplus sys/root as sysdba
CREATE USER $RANGERDBA IDENTIFIED BY $RANGERDBAPASSWORD;
GRANT SELECT_CATALOG_ROLE TO $RANGERDBA;
GRANT CONNECT, RESOURCE TO $RANGERDBA;
QUIT;
```

3. Use the following command format to set the `jdbc/driver/path` based on the location of the Oracle JDBC driver .jar file. This command must be run on the server where Ambari server is installed.

```
ambari-server setup --jdbc-db={database-type} --jdbc-driver={/jdbc/driver/
path}
```

For example:

```
ambari-server setup --jdbc-db=oracle --jdbc-driver=/usr/share/java/ojdbc6.
jar
```

6.1.3. Ranger Installation

To install Ranger using Ambari:

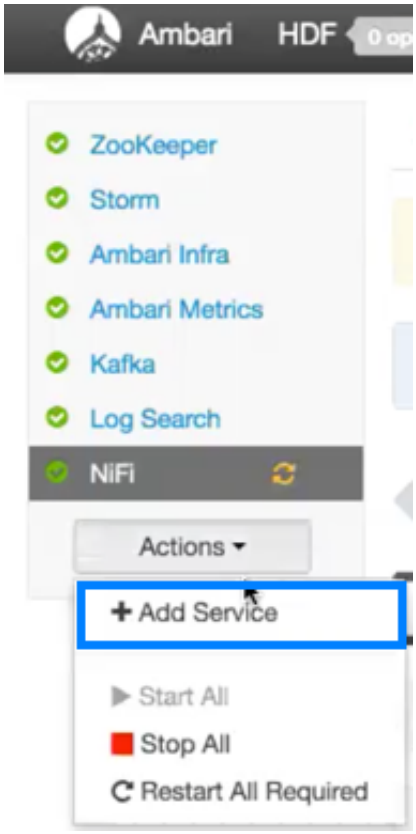
1. Start the Installation
2. Customize Services
3. Complete the Installation

Related Topics

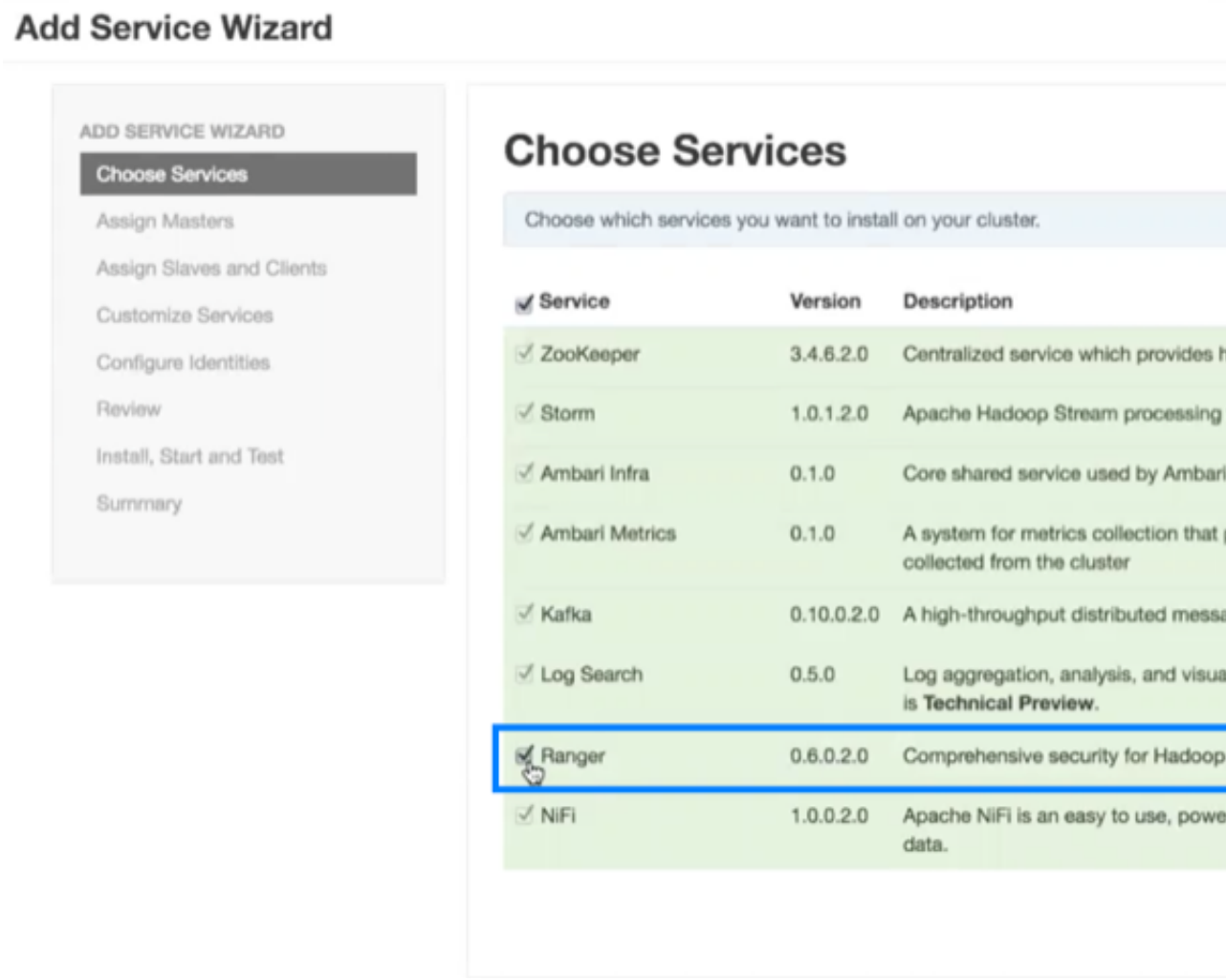
- Setting up Database Users Without Sharing DBA Credentials
- Updating Ranger Admin Passwords

6.1.3.1. Start the Installation

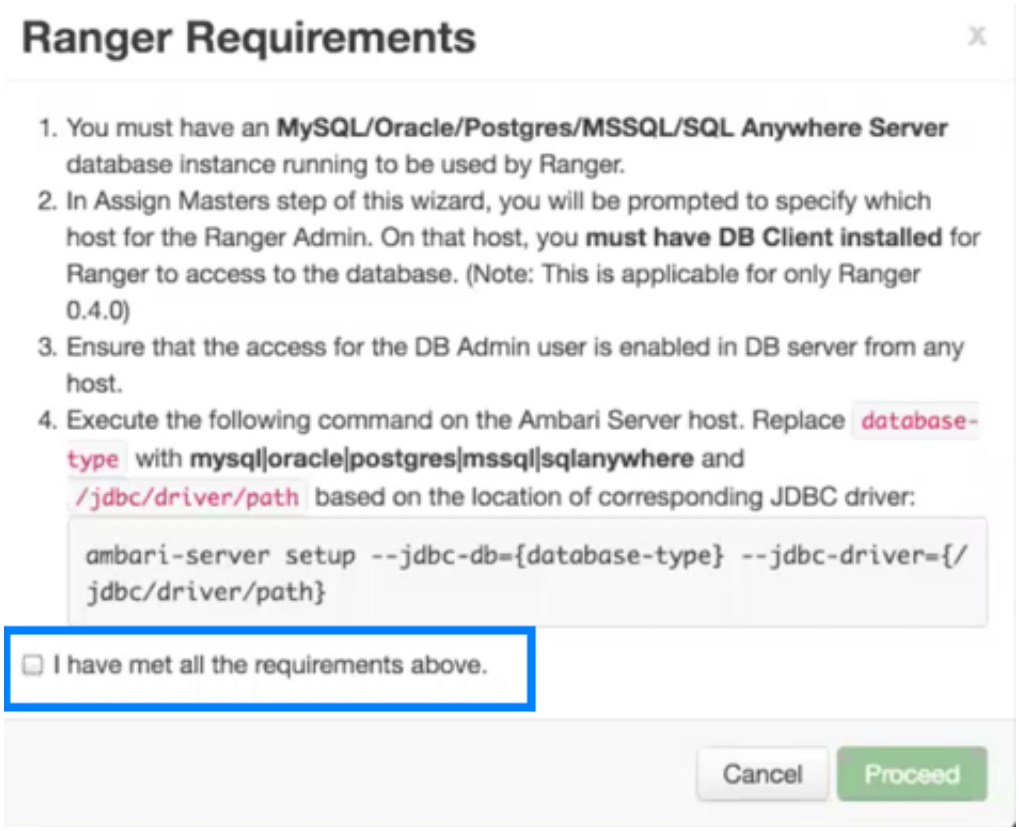
1. Log into your Ambari cluster with your designated user credentials. The main Ambari Dashboard page will be displayed.
2. From the main Ambari Dashboard page, click **Actions**, then select **Add Service**.



3. On the Choose Services page, select **Ranger**, then click **Next**.



4. The Ranger Requirements page appears. Ensure that you have met all of the installation requirements, then select the "I have met all the requirements above" check box and click **Proceed**.

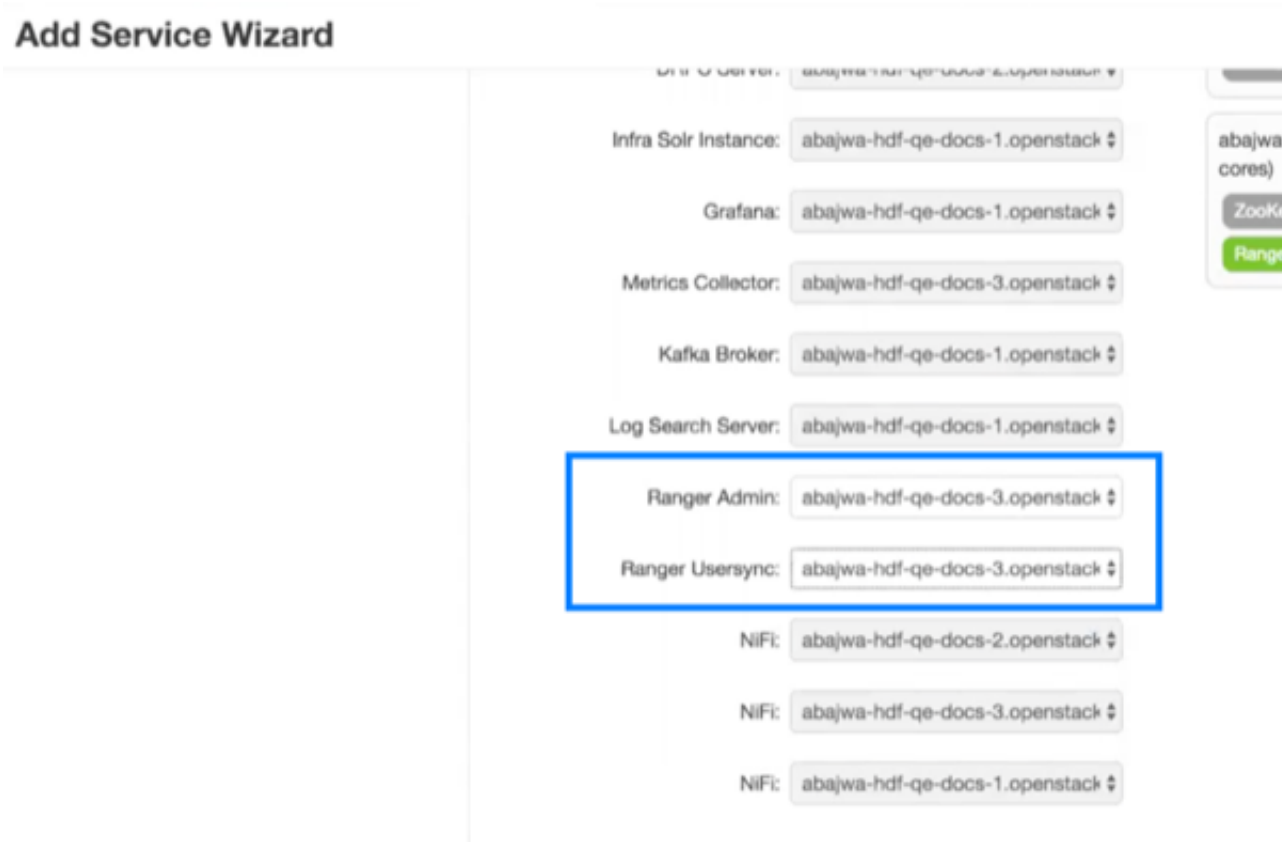


5. From the **Assign Masters** page, you are then prompted to select the host where you want to install Ranger Admin. This host must have DB admin access to the Ranger DB host and User Sync.

   Make a note of the Ranger Admin host for use in subsequent installation steps. Click **Next** when finished to continue with the installation.

Note

The Ranger Admin and Ranger User Sync services must be installed on the same cluster node.



6. From the **Assign Slaves and Clients** page, click **Next**.

7. The Customize Services page appears. These settings are described in the next section.

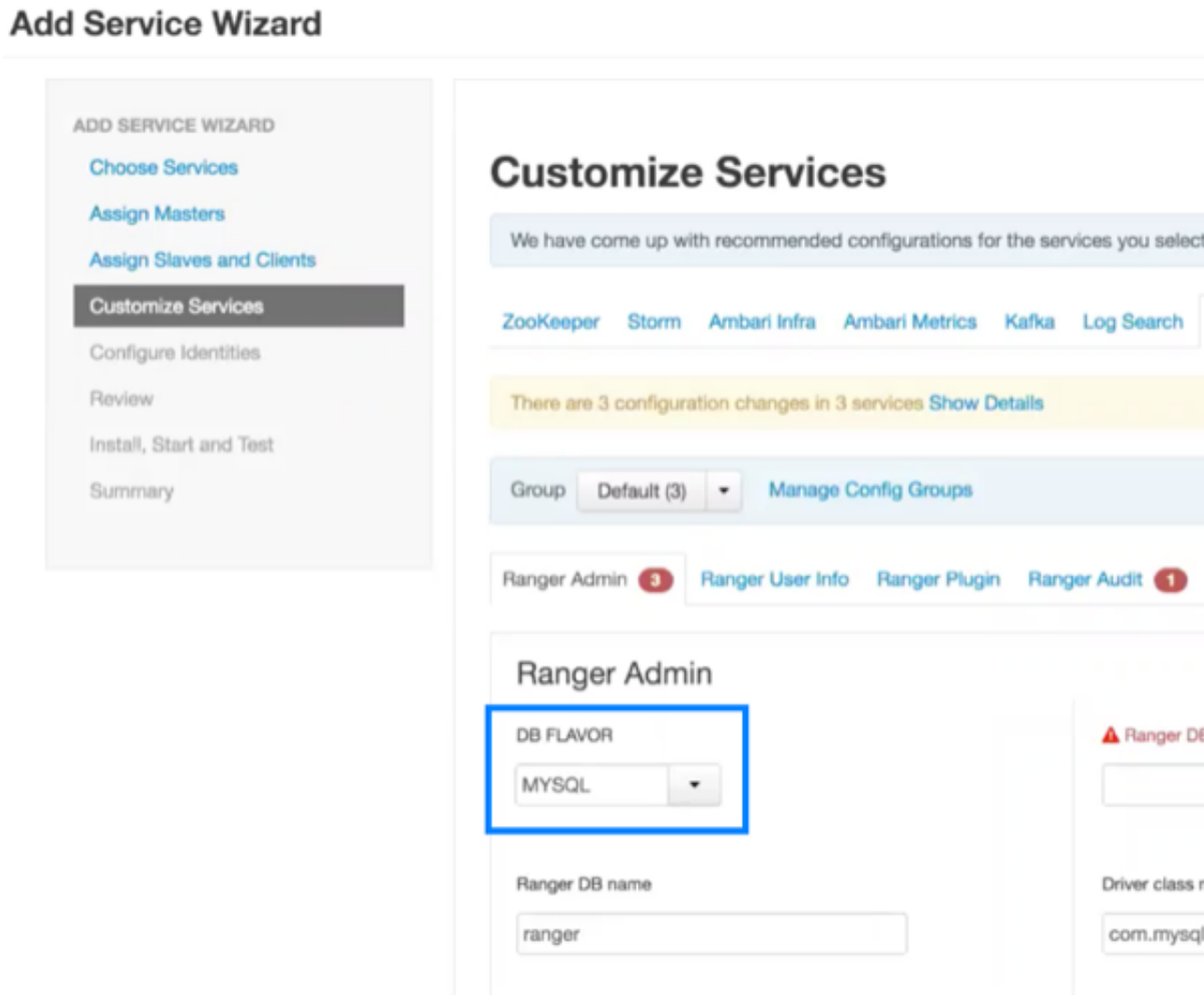
6.1.3.2. Customize Services

The next step in the installation process is to specify Ranger settings on the Customize Services page.

- Ranger Admin Settings
- Configure Ranger User Sync
- Specify Plugins to Enable
- Ranger Audit Settings
- Configure Ranger Authentication

6.1.3.2.1. Ranger Admin Settings

1. On the Customize Services page, select the Ranger Admin tab, then use the **DB Flavor** drop-down to select the database type that you are using with Ranger.



2. Enter the database server address in the **Ranger DB Host** box.

Table 6.1. Ranger DB Host

DB Flavor	Host	Example
MySQL	<HOST[:PORT]>	c6401.ambari.apache.org **or** c6401.ambari.apache.org:3306
Oracle	<HOST:PORT:SID>	c6401.ambari.apache.org:1521:ORCL
	<HOST:PORT/Service>	c6401.ambari.apache.org:1521/XE
PostgreSQL	<HOST[:PORT]>	c6401.ambari.apache.org **or** c6401.ambari.apache.org:5432
MS SQL	<HOST[:PORT]>	c6401.ambari.apache.org **or**

DB Flavor	Host	Example
		c6401.ambari.apache.org:1433
SQLA	<HOST[:PORT]>	c6401.ambari.apache.org **or** c6401.ambari.apache.org:2638

3. **Ranger DB name** – The name of the Ranger Policy database, i.e. ranger_db. Please not that if you are using Oracle, you must specify the Oracle tablespace name here.

4. Driver class name for a JDBC Ranger database – the driver class name is automatically generated based on the selected DB Flavor. The table below lists the default driver class settings. Currently Ranger does not support any third party JDBC driver.

Table 6.2. Driver Class Name

DB Flavor	Driver class name for a JDBC Ranger database
MySQL	com.mysql.jdbc.Driver
Oracle	oracle.jdbc.driver.OracleDriver
PostgreSQL	org.postgresql.Driver
MS SQL	com.microsoft.sqlserver.jdbc.SQLServerDriver
SQLA	sap.jdbc4.sqlanywhere.IDriver

5. **Ranger DB username** and **Ranger DB Password** – Enter the user name and passwords for your Ranger database server. The following table describes these settings in more detail. You can use the MySQL database that was installed with Ambari, or an external MySQL, Oracle, PostgreSQL, MS SQL or SQL Anywhere database.

Table 6.3. Ranger DB Username Settings

Property	Description	Default Value	Example Value	Required?
Ranger DB username	The username for the Policy database.	rangeradmin	rangeradmin	Yes
Ranger DB password	The password for the Ranger Policy database user.		PassWORd	Yes

6. **JDBC connect string**

Important

Currently the Ambari installer generates the JDBC connect string using the `jdbc:oracle:thin:@//host:port/db_name` format. You must replace the connection string as described in the following table:

Table 6.4. JDBC Connect String

DB Flavor	Syntax	Example Value
MySQL	jdbc:mysql://DB_HOST:PORT/db_name	jdbc:mysql://c6401.ambari.apache.org:3306/ranger_db
Oracle	**For Oracle SID:**	jdbc:oracle:thin:@c6401.ambari.apache.org:1521:**ORCL**

DB Flavor	Syntax	Example Value
	jdbc:oracle:thin:@DB_HOST:PORT:SID	
	For Oracle Service Name: jdbc:oracle:thin:@//DB_HOST[:PORT][/ServiceName]	jdbc:oracle:thin:@//c6401.ambari.apache.org:1521/**XE**
PostgreSQL	jdbc:postgresql://DB_HOST/db_name	jdbc:postgresql://c6401.ambari.apache.org:5432/ranger_db
MS SQL	jdbc:sqlserver://DB_HOST;databaseName=db_name	jdbc:sqlserver://c6401.ambari.apache.org:1433;databaseName=ranger_db
SQLA	jdbc:sqlanywhere:host=DB_HOST;database=db_name	jdbc:sqlanywhere:host=c6401.ambari.apache.org:2638;data

7. **Setup Database and Database User**

   - If set to **Yes** – The Database Administrator (DBA) user name and password will need to be provided as described in the next step.

     Note

     Ranger does not store the DBA user name and password after setup. Therefore, you can clear these values in the Ambari UI after the Ranger setup is complete.

   - If set to **No** – A **No** indicates that you do not wish to provide Database Administrator (DBA) account details to the Ambari Ranger installer. Setting this to No continues the Ranger installation process without providing DBA account details. In this case, you must perform the system database user setup as described in Setting up Database Users Without Sharing DBA Credentials, and then proceed with the installation.

     Note

     If **No** is selected and the UI still requires you to enter a user name and password in order to proceed, you can enter any value – the values do not need to be the actual DBA user name and password.

8. **Database Administrator (DBA) username** and **Database Administrator (DBA) password** – The DBA username and password are set when the database server is installed. If you do not have this information, contact the database administrator who installed the database server.

Table 6.5. DBA Credential Settings

Property	Description	Default Value	Example Value	Required?
Database Administrator (DBA) username	The Ranger database user that has administrative privileges to create database schemas and users.	root	root	Yes
Database Administrator (DBA) password	The root password for the Ranger database user.		root	Yes

If the Oracle DB root user Role is SYSDBA, you must also specify that in the **Database Administrator (DBA) username** parameter. For example, if the DBA user name is `orcl_root` you must specify `orcl_root AS SYSDBA`.

Note

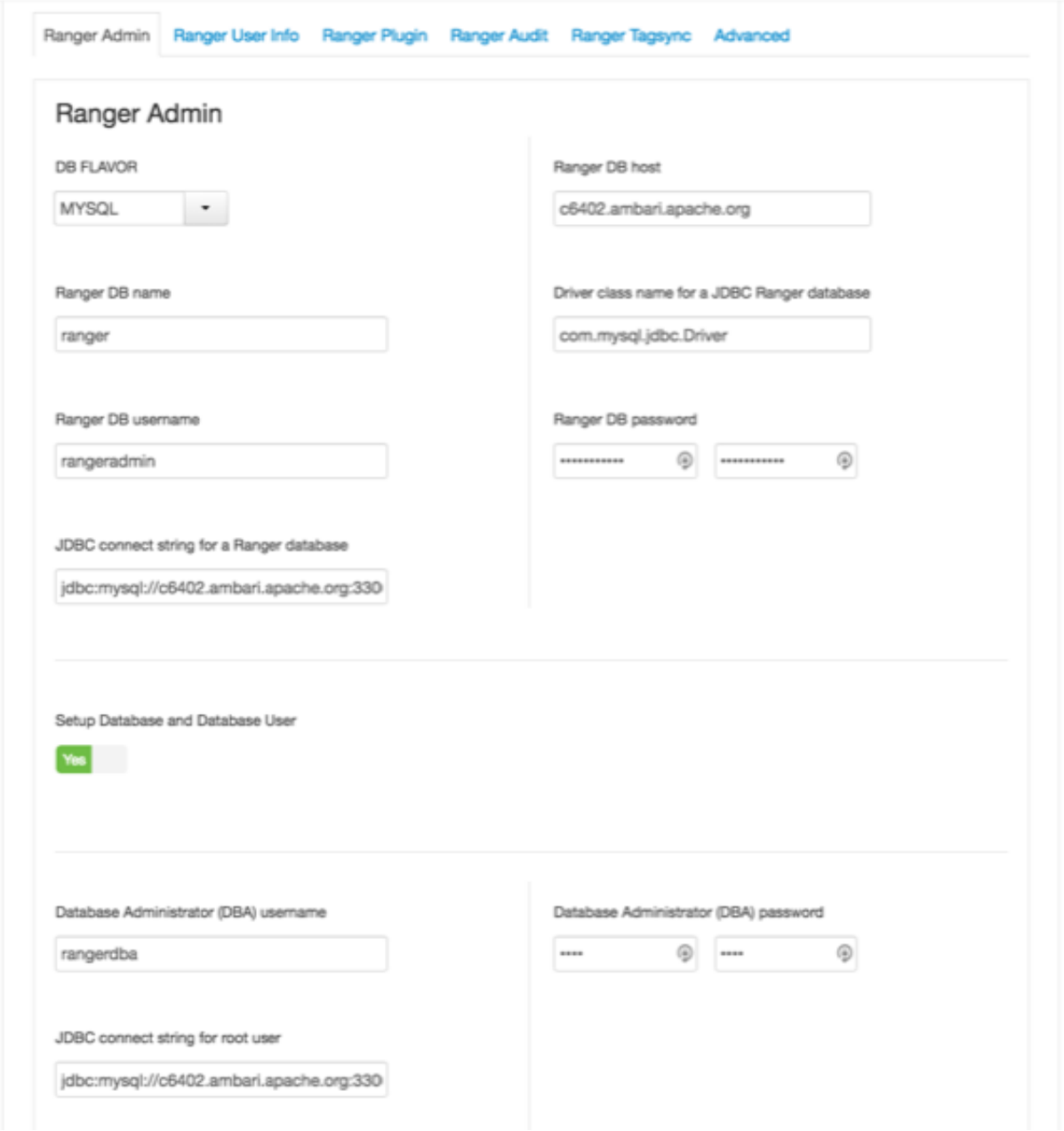
As mentioned in the note in the previous step, if **Setup Database and Database User** is set to **No**, a placeholder DBA username and password may still be required in order to continue with the Ranger installation.

The following images show examples of the DB settings for each Ranger database type.

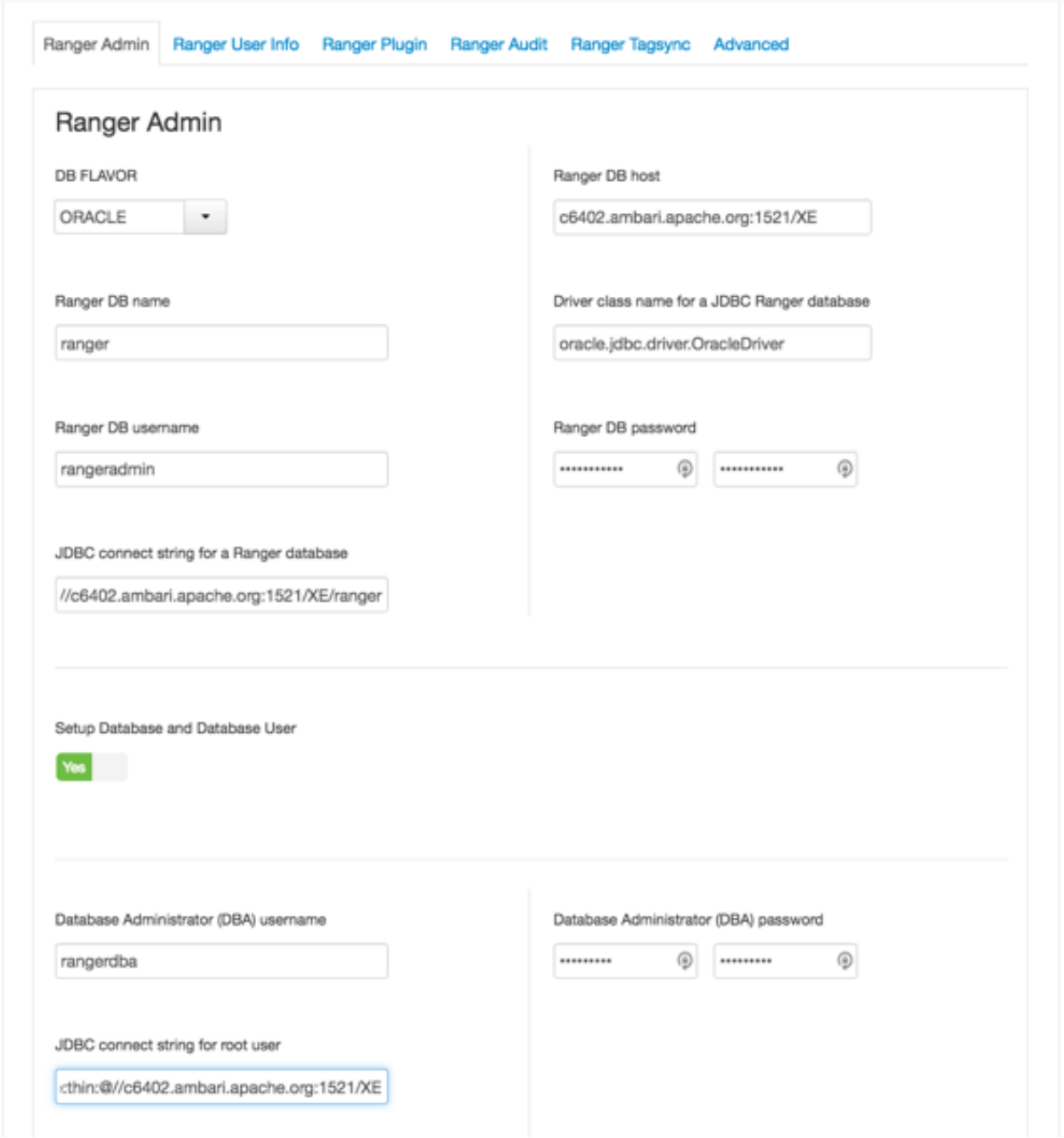
Note

To test the DB settings, click **Test Connection**. If a Ranger database has not been pre-installed, **Test Connection** will fail even for a valid configuration.

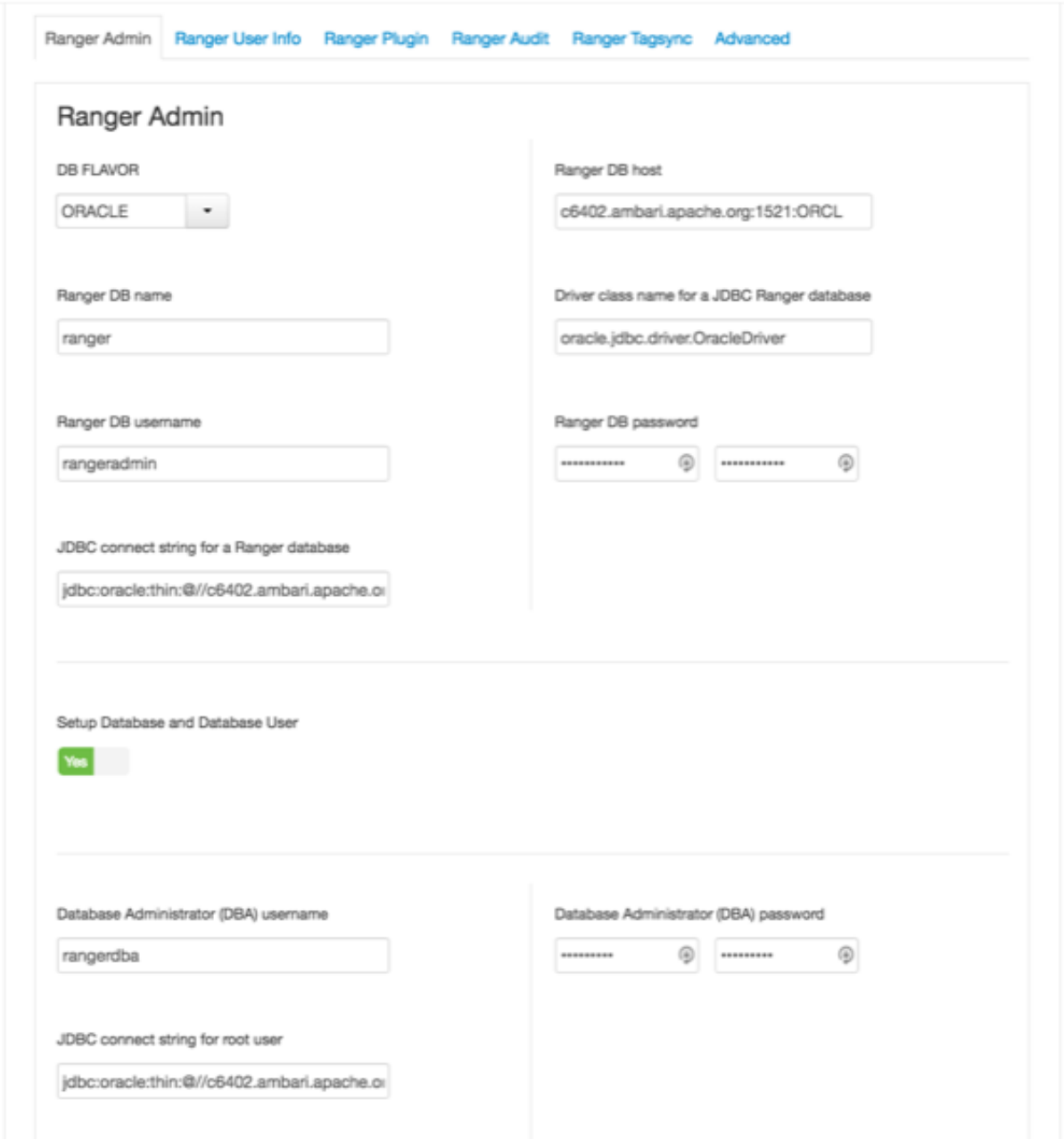
MySQL



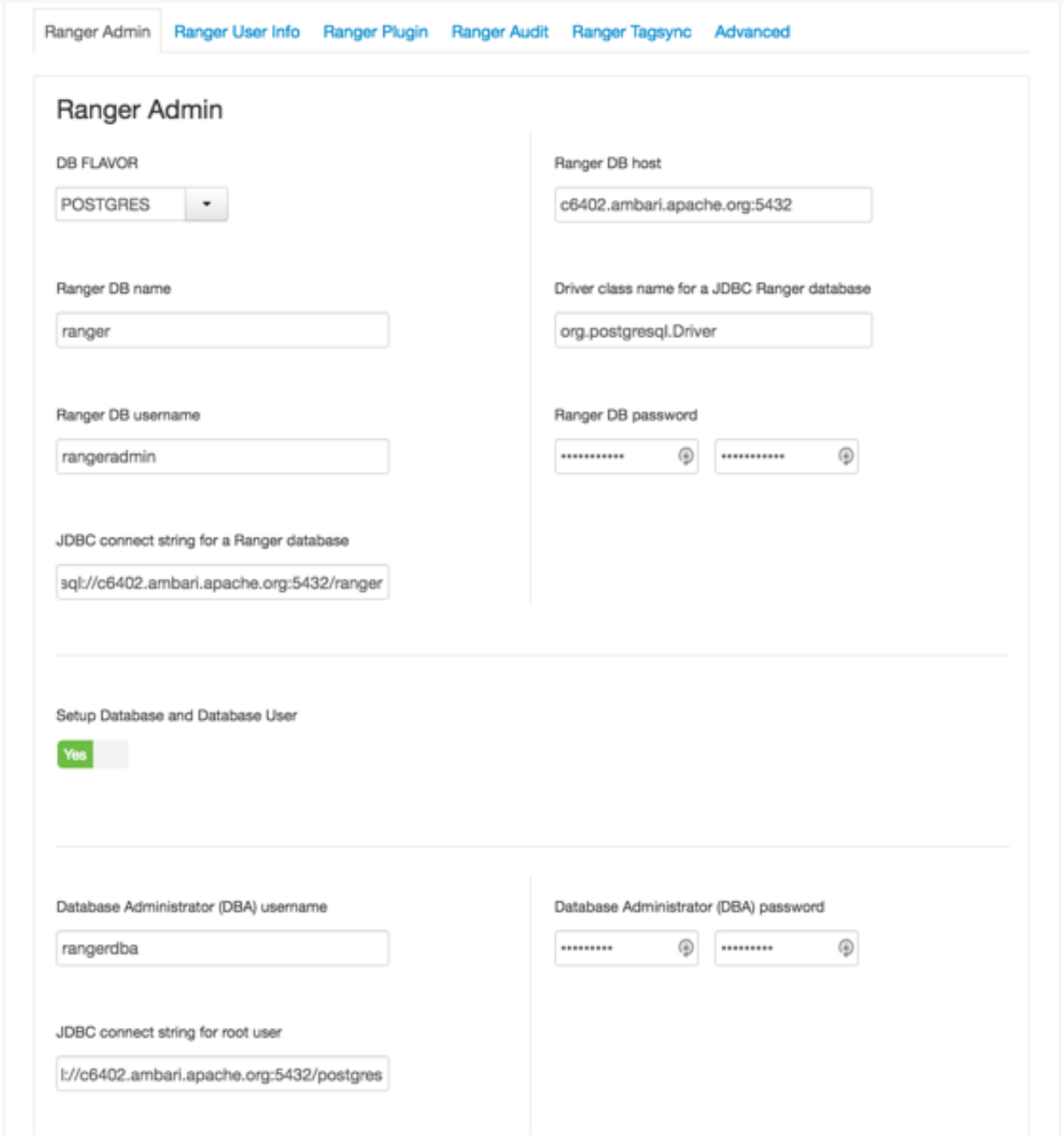
Oracle – if the Oracle instance is running with a Service name.



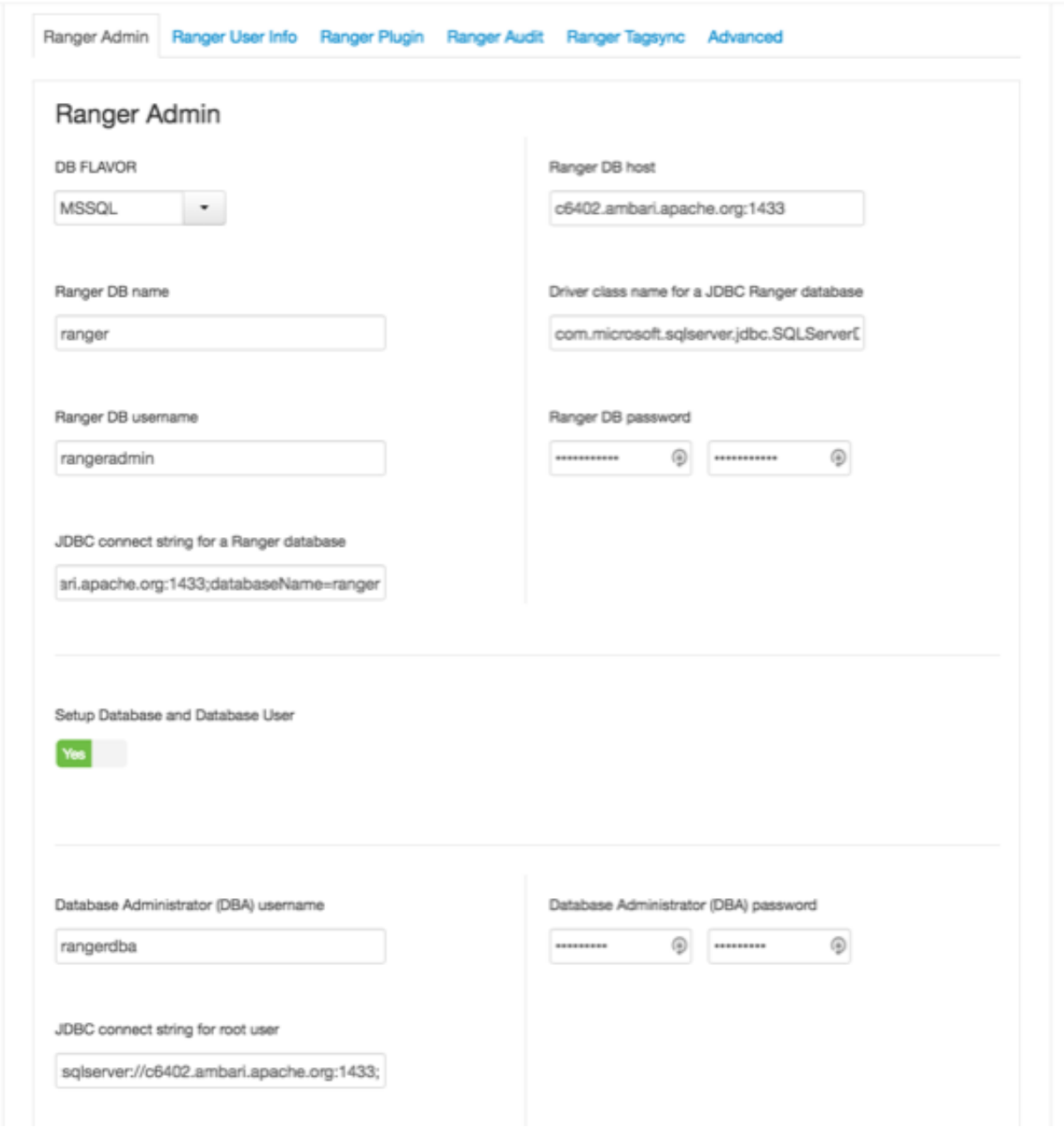
Oracle – if the Oracle instance is running with a SID.


Ranger Admin
Ranger User Info
Ranger Plugin
Ranger Audit
Ranger Tagsync
Advanced
Ranger Admin
DB FLAVOR
ORACLE
Ranger DB host
c6402.ambari.apache.org:1521:ORCL
Ranger DB name
ranger
Driver class name for a JDBC Ranger database
oracle.jdbc.driver.OracleDriver
Ranger DB username
rangeradmin
Ranger DB password
JDBC connect string for a Ranger database
jdbc:oracle:thin:@//c6402.ambari.apache.o
Setup Database and Database User
Yes
Database Administrator (DBA) username
rangerdba
Database Administrator (DBA) password
JDBC connect string for root user
jdbc:oracle:thin:@//c6402.ambari.apache.o

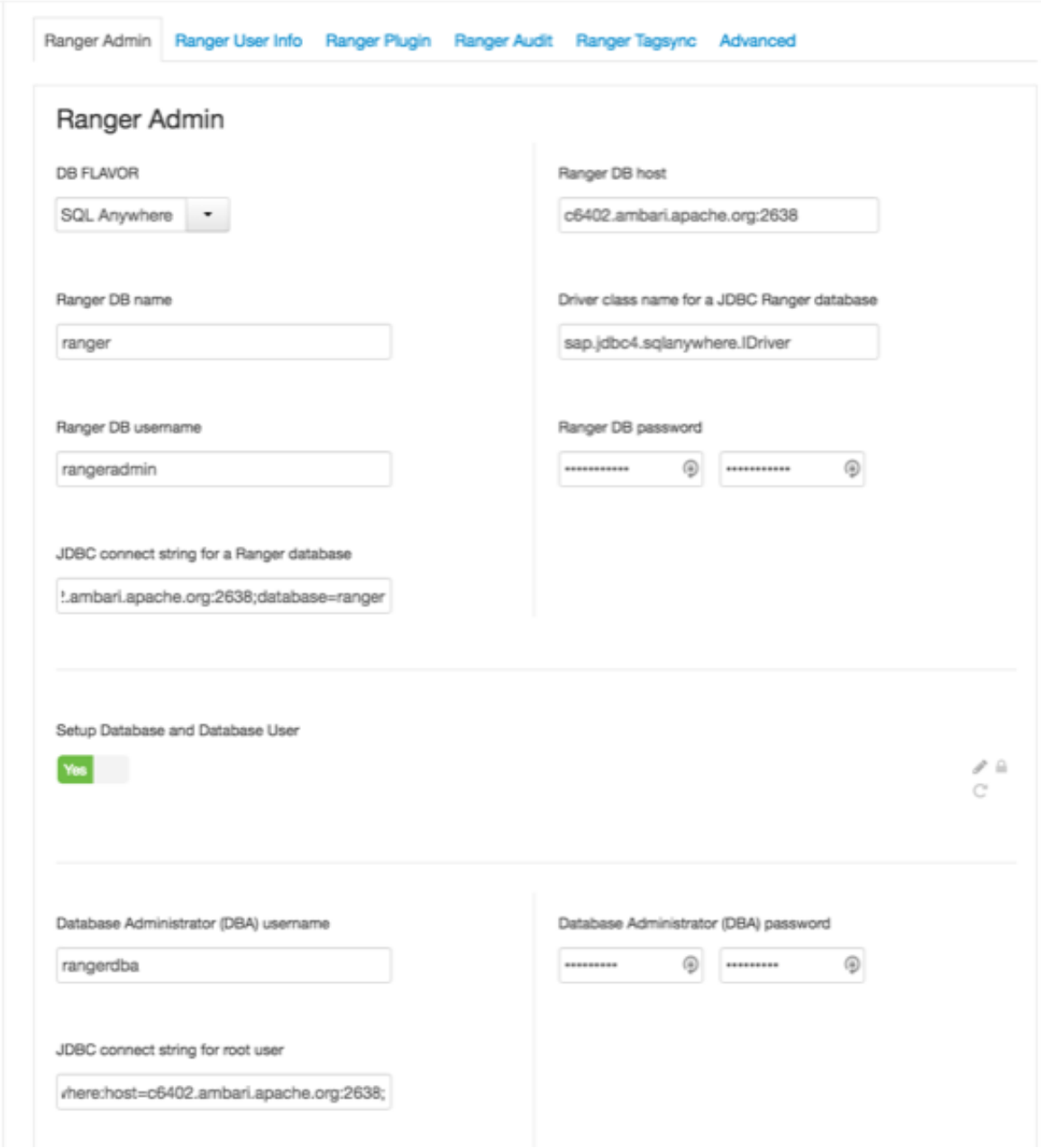
PostgreSQL


Ranger Admin
Ranger User Info
Ranger Plugin
Ranger Audit
Ranger Tagsync
Advanced
Ranger Admin
DB FLAVOR
POSTGRES
Ranger DB host
c6402.ambari.apache.org:5432
Ranger DB name
ranger
Driver class name for a JDBC Ranger database
org.postgresql.Driver
Ranger DB username
rangeradmin
Ranger DB password
JDBC connect string for a Ranger database
sql://c6402.ambari.apache.org:5432/ranger
Setup Database and Database User
Yes
Database Administrator (DBA) username
rangerdba
Database Administrator (DBA) password
JDBC connect string for root user
l://c6402.ambari.apache.org:5432/postgres

MS SQL



SQL Anywhere



6.1.3.2.2. Configure Ranger User Sync

This section describes how to configure Ranger User Sync for either UNIX or LDAP/AD.

- Configuring Ranger User Sync for UNIX
- Configuring Ranger User Sync for LDAP/AD

6.1.3.2.2.1. Configuring Ranger User Sync for UNIX

Use the following steps to configure Ranger User Sync for UNIX.

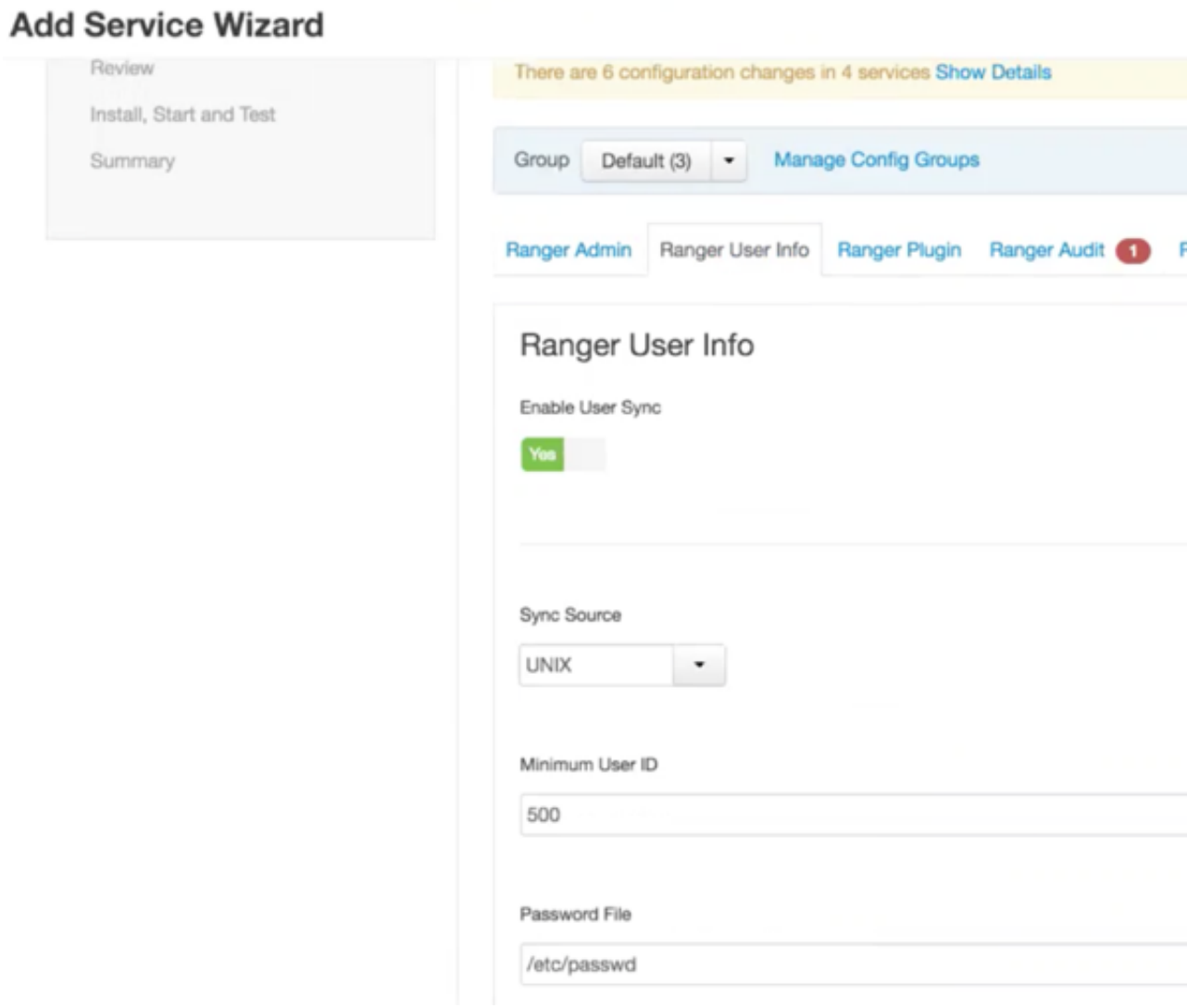
1. On the Customize Services page, select the **Ranger User Info** tab.

2. Click **Yes** under Enable User Sync.

3. Use the **Sync Source** drop-down to select UNIX, then set the following properties.

Table 6.6. UNIX User Sync Properties

Property	Description	Default Value
Sync Source	Only sync users above this user ID.	500
Password File	The location of the password file on the Linux server.	/etc/passwd
Group File	The location of the groups file on the Linux server.	/etc/group



6.1.3.2.2.2. Configuring Ranger User Sync for LDAP/AD

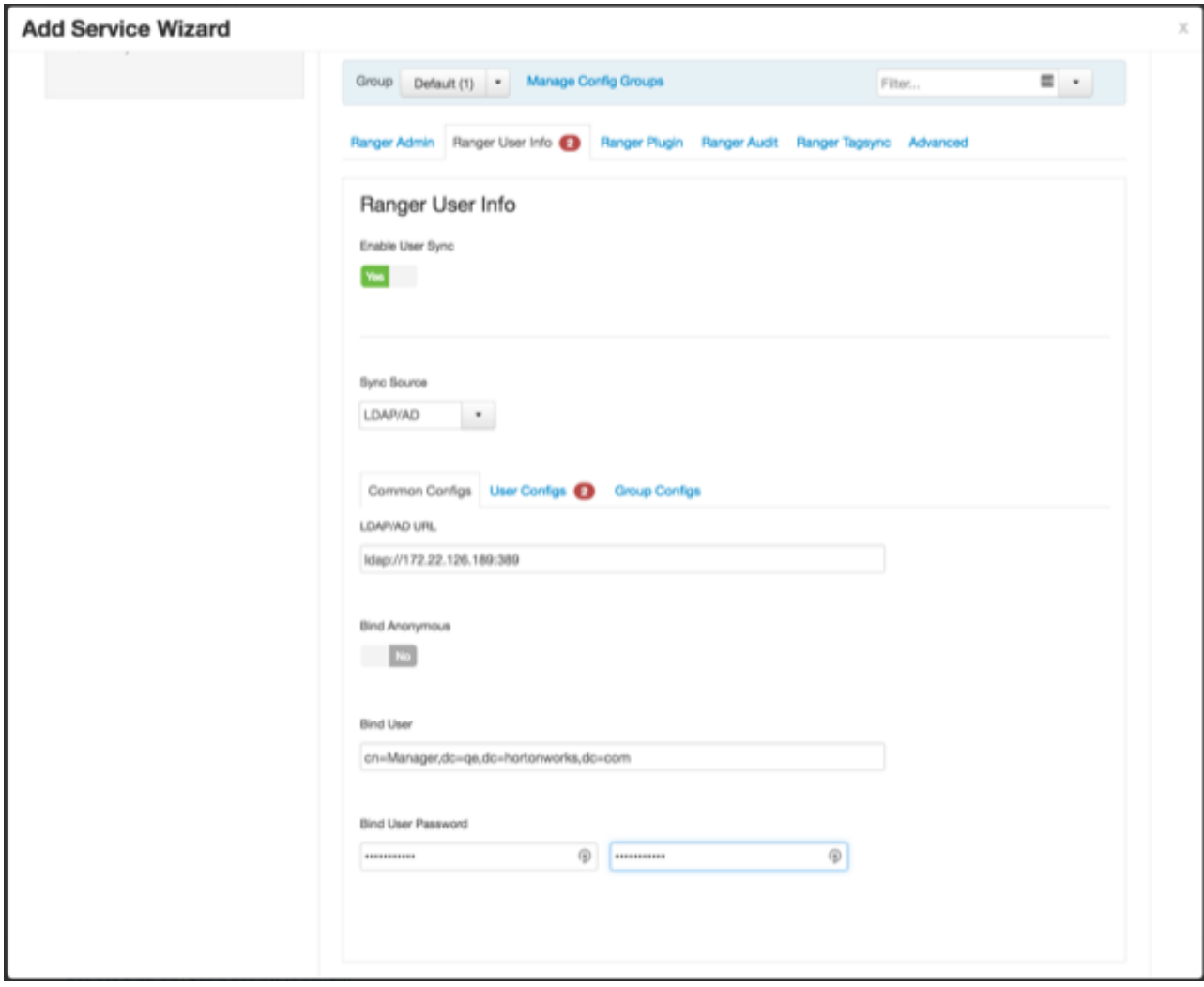
Use the following steps to configure Ranger User Sync for LDAP/AD.

1. On the Customize Services page, select the Ranger User Info tab.

2. Click **Yes** under Enable User Sync.
3. Use the Sync Source drop-down to select LDAP/AD.
4. Set the following properties on the Common Configs tab.

Table 6.7. LDAP/AD Common Configs

Property	Description	Default Value	Sample Values
LDAP/AD URL	Add URL depending upon LDAP/AD sync source	ldap://{host}:{port}	ldap://ldap.example.com:389 or ldaps://ldap.example.com:636
Bind Anonymous	If Yes is selected, the Bind User and Bind User Password are not required.	NO	
Bind User	The location of the groups file on the Linux server.	The full distinguished name (DN), including common name (CN), of an LDAP/AD user account that has privileges to search for users. The LDAP bind DN is used to connect to LDAP and query for users and groups.	cn=admin,dc=example,dc=com or admin@example.com
Bind User Password	The password of the Bind User.		

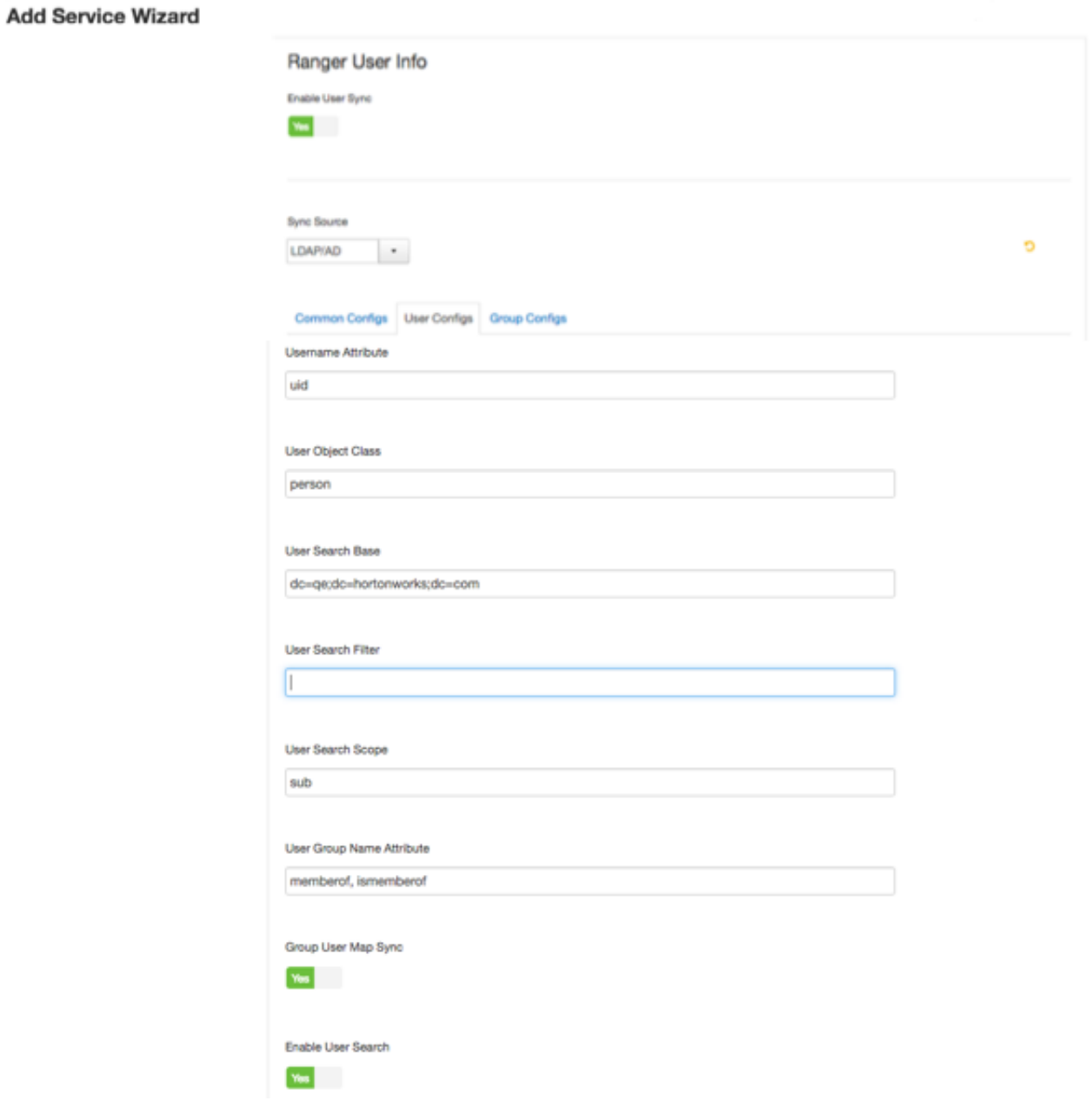


5. Set the following properties on the User Configs tab.

Table 6.8. LDAP/AD User Configs

Property	Description	Default Value	Sample Values
Group User Map Sync	Sync specific groups for users.	Yes	Yes
Username Attribute	The LDAP user name attribute.		sAMAccountName for AD, uid or cn for OpenLDAP
User Object Class	Object class to identify user entries.	person	top, person, organizationalPerson, user, or posixAccount
User Search Base	Search base for users. Ranger can search multiple OUs in AD. Ranger UserSync module performs a user search on each configured OU and adds all the users into single list. Once all the OUs are		cn=users,dc=example,dc=com;ou=example1,ou=e

Property	Description	Default Value	Sample Values
	processed, a user's group membership is computed based on the group search.		
User Search Filter	Optional additional filter constraining the users selected for syncing.		Sample filter to retrieve all the users: cn=* Sample filter to retrieve all the users who are members of groupA or groupB: (| (memberof=CN=GroupA,OU=groups,DC=example (memberof=CN=GroupB,OU=groups,DC=example
User Search Scope	This value is used to limit user search to the depth from search base.	sub	base, one, or sub
User Group Name Attribute	Attribute from user entry whose values would be treated as group values to be pushed into the Access Manager database. You can provide multiple attribute names separated by commas.	memberof,ismemberof	memberof, ismemberof, or gidNumber
Enable User Search	This option is available only when the "Enable Group Search First" option is selected.	No	Yes

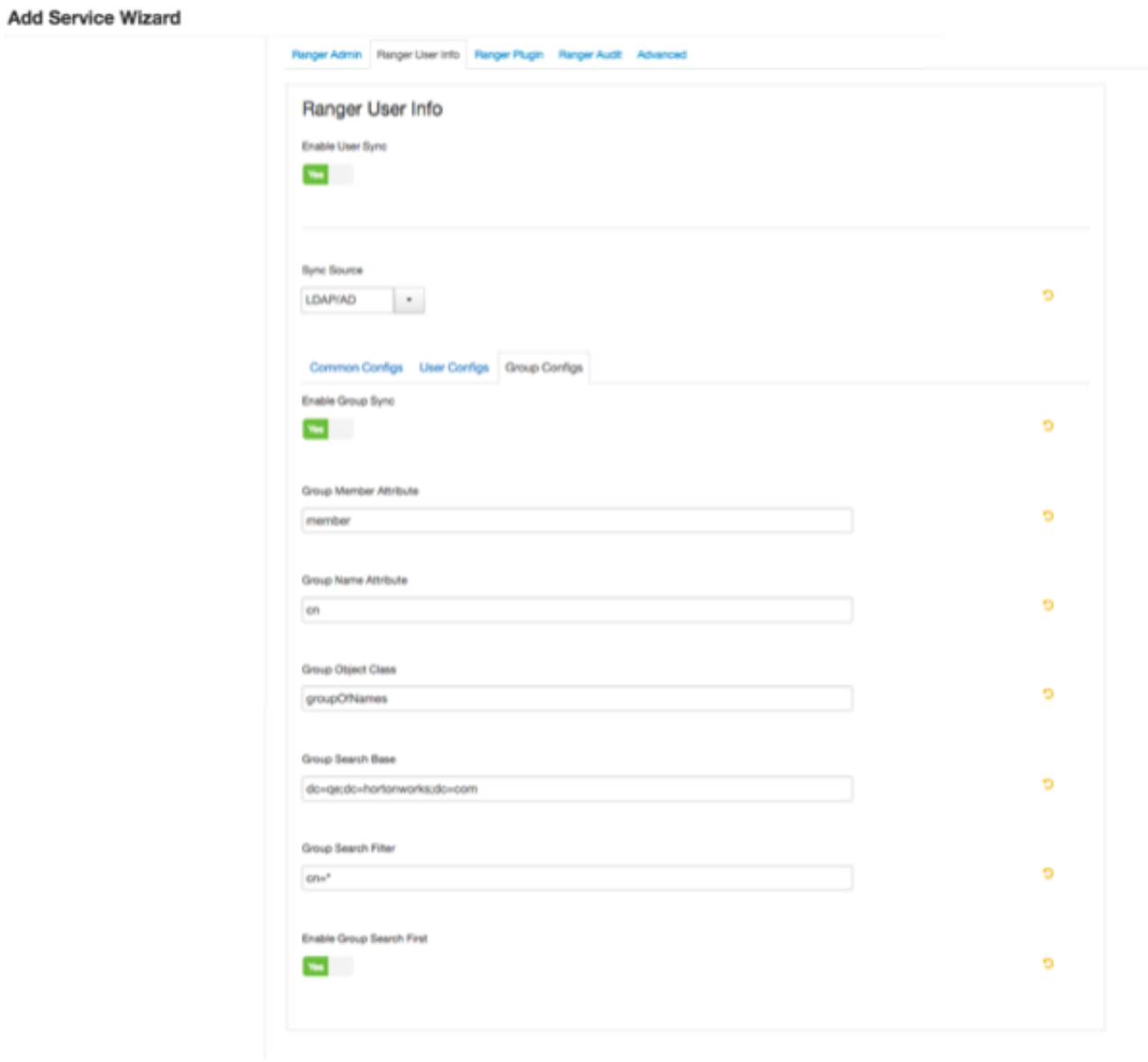


6. Set the following properties on the Group Configs tab.

Table 6.9. LDAP/AD Group Configs

Property	Description	Default Value	Sample Values
Enable Group Sync	If Enable Group Sync is set to No, the group names the users belong to are derived from "User Group Name Attribute". In this case no additional group filters are applied.	No	Yes

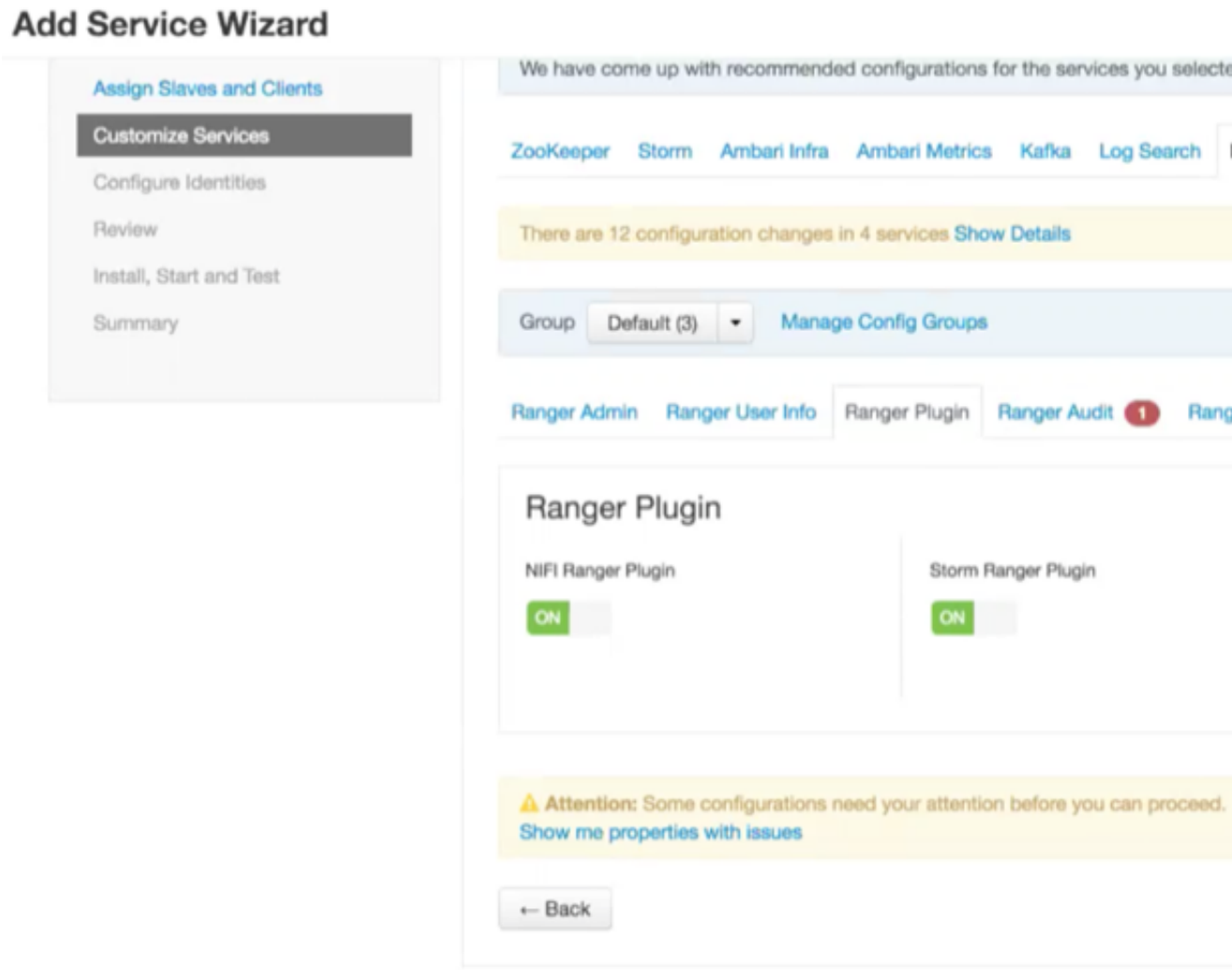
Property	Description	Default Value	Sample Values
	If Enable Group Sync is set to Yes, the groups the users belong to are retrieved from LDAP/AD using the following group-related attributes.		
Group Member Attribute	The LDAP group member attribute name.		member
Group Name Attribute	The LDAP group name attribute.		distinguishedName for AD, cn for OpenLdap
Group Object Class	LDAP Group object class.		group, groupofnames, or posixGroup
Group Search Base	Search base for groups. Ranger can search multiple OUs in AD. Ranger UserSync module performs a user search on each configured OU and adds all the users into single list. Once all the OUs are processed, a user's group membership is computed based on the group search configuration. Each OU segment needs to be separated by a ; (semi-colon).		ou=groups,DC=example,DC=com;ou=group1;ou=
Group Search Filter	Optional additional filter constraining the groups selected for syncing.		Sample filter to retrieve all groups: cn=* Sample filter to retrieve only the groups whose cn is Engineering or Sales: (|(cn=Engineering)(cn=Sales))
Enable Group Search First	When **Enable Group Search First** is selected, there are two possible ways of retrieving users: • If **Enable User Search** is not selected: users are retrieved from the "member" attribute of the group. • If **Enable User Search** is selected: user membership is computed by performing an LDAP search based on the user configuration.	No	Yes



6.1.3.2.3. Specify Plugins to Enable

From the **Ranger Plugin** tab, use the **ON**/**OFF** slider to indicate which plugins you want to enable. You can also enable plugins at a later time.

If you select the Storm or Kafka plugins here, they are not enabled until you also enable Kerberos.

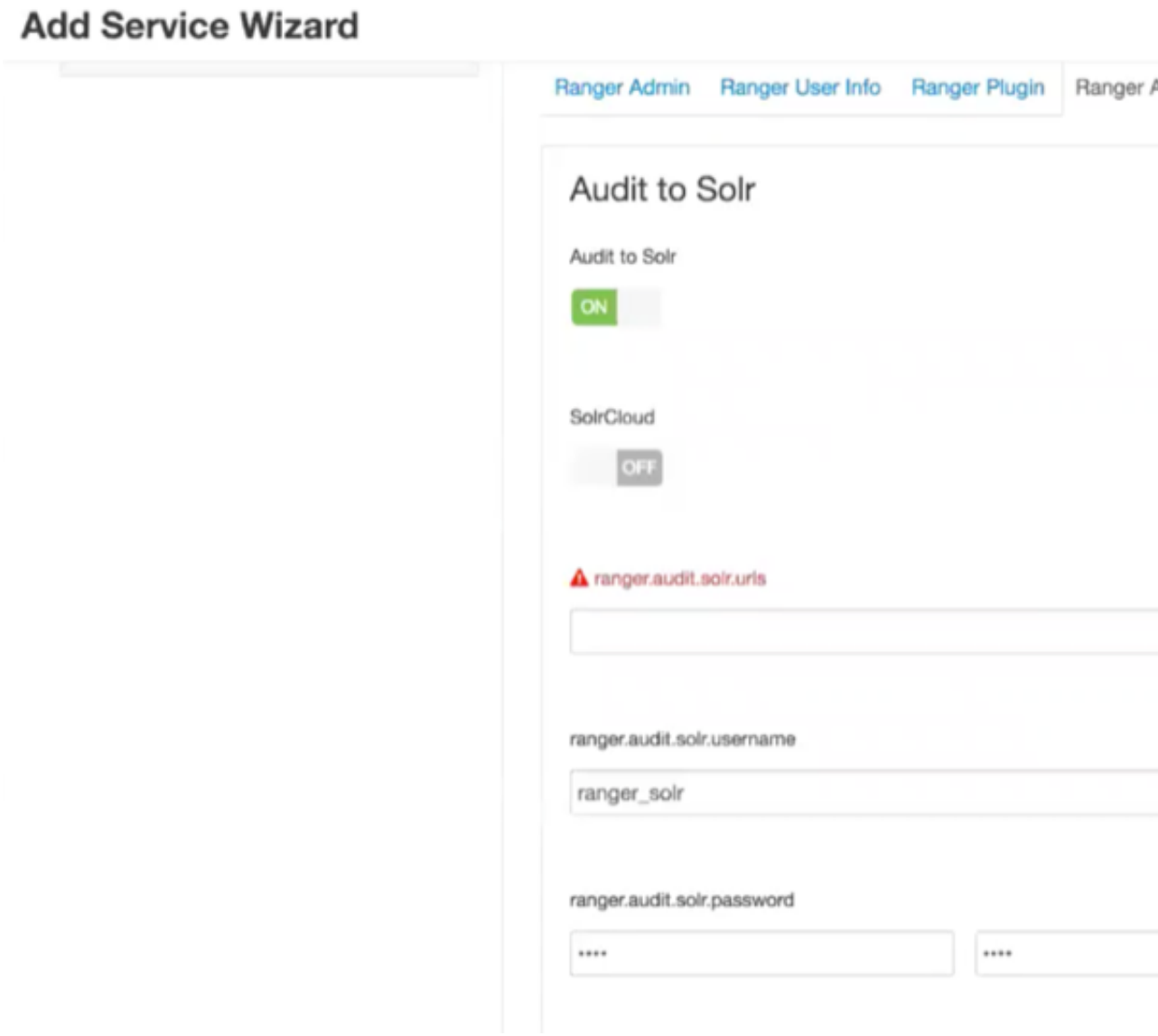


6.1.3.2.4. Ranger Audit Settings

Apache Ranger uses Apache Solr to store audit logs and provides UI searching through the audit logs. Solr must be installed and configured before installing Ranger Admin or any of the Ranger component plugins. The default configuration for Ranger Audits to Solr uses the shared Solr instance provided under the Ambari Infra service. Solr is both memory and CPU intensive. If your production system has high volume of access requests, make sure that the Solr host has adequate memory, CPU, and disk space.

SolrCloud is the preferred setup for production usage of Ranger. SolrCloud, which is deployed with the Ambari Infra service, is a scalable architecture that can run as a single node or multi-node cluster. It has additional features such as replication and sharding, which is useful for high availability (HA) and scalability. You should plan your deployment based on your cluster size. Because audit records can grow dramatically, plan to have at least 1 TB of free space in the volume on which Solr will store the index data. Solr works well with a minimum of 32 GB of RAM. You should provide as much memory as possible to the Solr process. It is highly recommended to use SolrCloud with at least two Solr nodes running on different servers with replication enabled. SolrCloud also requires Apache ZooKeeper.

1. On the Customize Services page, select the **Ranger Audit** tab.
2. Under Audit to Solr, click **OFF** under SolrCloud to enable SolrCloud. The button label will change to ON, and the SolrCloud configuration settings will be loaded automatically.



6.1.3.2.5. Configure Ranger Authentication

This section describes how to configure Ranger authentication for UNIX, LDAP, and AD.

- Configuring Ranger UNIX Authentication
- Configuring Ranger LDAP Authentication
- Configuring Ranger Active Directory Authentication

6.1.3.2.5.1. Configuring Ranger UNIX Authentication

Use the following steps to configure Ranger authentication for UNIX.

1. Select the Advanced tab on the Customize Services page.

2. Under Ranger Settings, specify the Ranger Access Manager/Service Manager host address in the **External URL** box in the format `http://<your_ranger_host>:6080`.

3. Under Ranger Settings, select **UNIX**.

   HTTP is enabled by default – if you disable HTTP, only HTTPS is allowed.

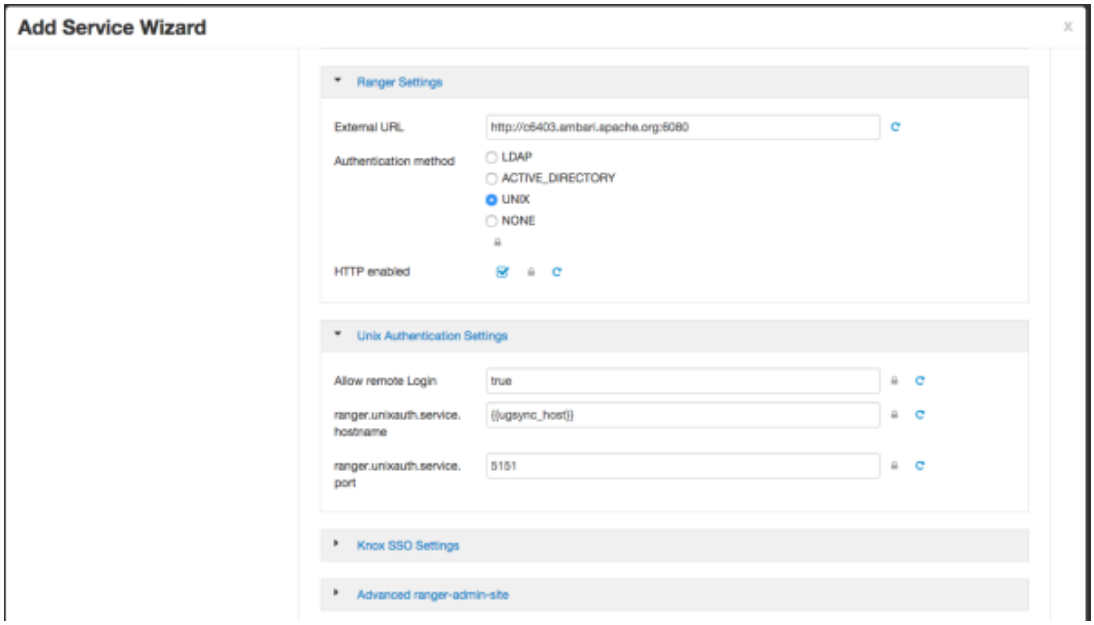
4. Under UNIX Authentication Settings, set the following properties.

Table 6.10. UNIX Authentication Settings

Property	Description	Default Value	Example Value
Allow remote Login	Flag to enable/ disable remote login. Only applies to UNIX authentication.	true	true
ranger.unixauth.service.hostname	The address of the host where the UNIX authentication service is running.	{{ugsync_host}}	{{ugsync_host}}
ranger.unixauth.service.port	The port number on which the UNIX authentication service is running.	5151	5151

Note

Properties with value `{{xyz}}` are macro variables that are derived from other specified values in order to streamline the configuration process. Macro variables can be edited if required – if you need to restore the original value, click the Set Recommended symbol at the right of the property box.



6.1.3.2.5.2. Configuring Ranger LDAP Authentication

Use the following steps to configure Ranger authentication for LDAP.

1. Select the Advanced tab on the Customize Services page.
2. Under Ranger Settings, specify the Ranger Access Manager/Service Manager host address in the **External URL** box in the format `http://<your_ranger_host>:6080`.
3. Under Ranger Settings, select **LDAP**.
4. Under LDAP Settings, set the following properties.

Table 6.11. LDAP Authentication Settings

Property	Description	Default Value	Example Value
ranger.ldap.base.dn	The Distinguished Name (DN) of the starting point for directory server searches.	dc=example,dc=com	dc=example,dc=com
Bind User	The full Distinguished Name (DN), including Common Name (CN) of an LDAP user account that has privileges to search for users. This is a macro variable value	{{ranger_ug_ldap_bind_dn}}	{{ranger_ug_ldap_bind_dn}}

Property	Description	Default Value	Example Value
	that is derived from the **Bind User** value from **Ranger User Info > Common Configs**.		
Bind User Password	Password for the Bind User. This is a macro variable value that is derived from the **Bind User Password** value from **Ranger User Info > Common Configs**.		
ranger.ldap.group.roleattribute	The LDAP group role attribute.	cn	cn
ranger.ldap.referral	See description below.	ignore	follow | ignore | throw
LDAP URL	The LDAP server URL. This is a macro variable value that is derived from the **LDAP/AD URL** value from **Ranger User Info > Common Configs**.	{{ranger_ug_ldap_url}}	{{ranger_ug_ldap_url}}
ranger.ldap.user.dnpattern	The user DN pattern is expanded when a user is being logged in. For example, if the user "ldapadmin" attempted to log in, the LDAP Server would attempt to bind against the DN "uid=ldapadmin,ou=users,dc=example,dc=com" using the password the user provided>	uid={0},ou=users,dc=xasecure,dc=net	cn=ldapadmin,ou=Users,dc=example,dc=com
User Search Filter	The search filter used for Bind Authentication.	{{ranger_ug_ldap_user_searchfilter}}	{{ranger_ug_ldap_user_searchfilter}}

Property	Description	Default Value	Example Value
	This is a macro variable value that is derived from the **User Search Filter** value from **Ranger User Info > User Configs**.		

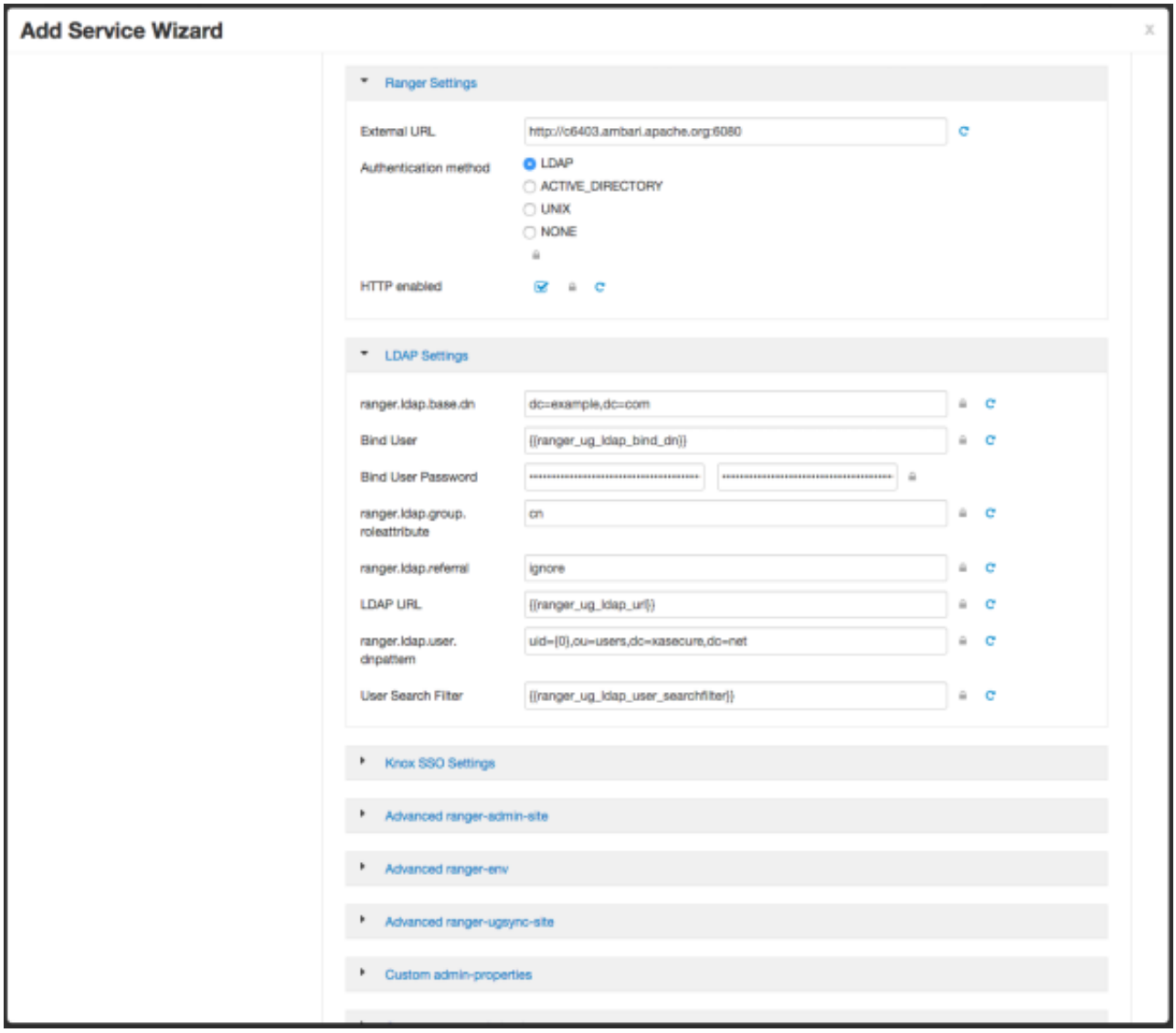
Note

Properties with value `{{xyz}}` are macro variables that are derived from other specified values in order to streamline the configuration process. Macro variables can be edited if required – if you need to restore the original value, click the Set Recommended symbol at the right of the property box.

There are three possible values for `ranger.ldap.referral`: `follow`, `throw`, and `ignore`. The recommended setting is `follow`.

When searching a directory, the server might return several search results, along with a few continuation references that show where to obtain further results. These results and references might be interleaved at the protocol level.

- When this property is set to `follow`, the LDAP service provider processes all of the normal entries first, and then follows the continuation references.
- When this property is set to `throw`, all of the normal entries are returned in the enumeration first, before the `ReferralException` is thrown. By contrast, a "referral" error response is processed immediately when this property is set to `follow` or `throw`.
- When this property is set to `ignore`, it indicates that the server should return referral entries as ordinary entries (or plain text). This might return partial results for the search.



6.1.3.2.5.3. Configuring Ranger Active Directory Authentication

Use the following steps to configure Ranger authentication for Active Directory.

1. Select the Advanced tab on the Customize Services page.
2. Under Ranger Settings, specify the Ranger Access Manager/Service Manager host address in the **External URL** box in the format `http://<your_ranger_host>:6080`.
3. Under Ranger Settings, select **ACTIVE_DIRECTORY**.
4. Under AD Settings, set the following properties.

Table 6.12. AD Settings

Property	Description	Default Value	Example Value
ranger.ldap.ad.base.dn	The Distinguished Name (DN) of the starting	dc=example,dc=com	dc=example,dc=com

Property	Description	Default Value	Example Value
	point for directory server searches.		
ranger.ldap.ad.bind.dn	The full Distinguished Name (DN), including Common Name (CN) of an LDAP user account that has privileges to search for users. This is a macro variable value that is derived from the **Bind User** value from **Ranger User Info > Common Configs**.	{{ranger_ug_ldap_bind_dn}}	{{ranger_ug_ldap_bind_dn}}
ranger.ldap.ad.bind.password	Password for the bind.dn. This is a macro variable value that is derived from the **Bind User Password** value from **Ranger User Info > Common Configs**.		
Domain Name (Only for AD)	The domain name of the AD Authentication service.		dc=example,dc=com
ranger.ldap.ad.referral	See description below.	ignore	follow | ignore | throw
ranger.ldap.ad.url	The AD server URL. This is a macro variable value that is derived from the **LDAP/ AD URL** value from **Ranger User Info > Common Configs**.	{{ranger_ug_ldap_url}}	{{ranger_ug_ldap_url}}
ranger.ldap.ad.user.searchfilter	The search filter used for Bind Authentication. This is a macro variable value that is derived from the **User Search Filter** value from **Ranger User Info > User Configs**.	{{ranger_ug_ldap_user_searchfilter}}	{{ranger_ug_ldap_user_searchfilter}}

Note

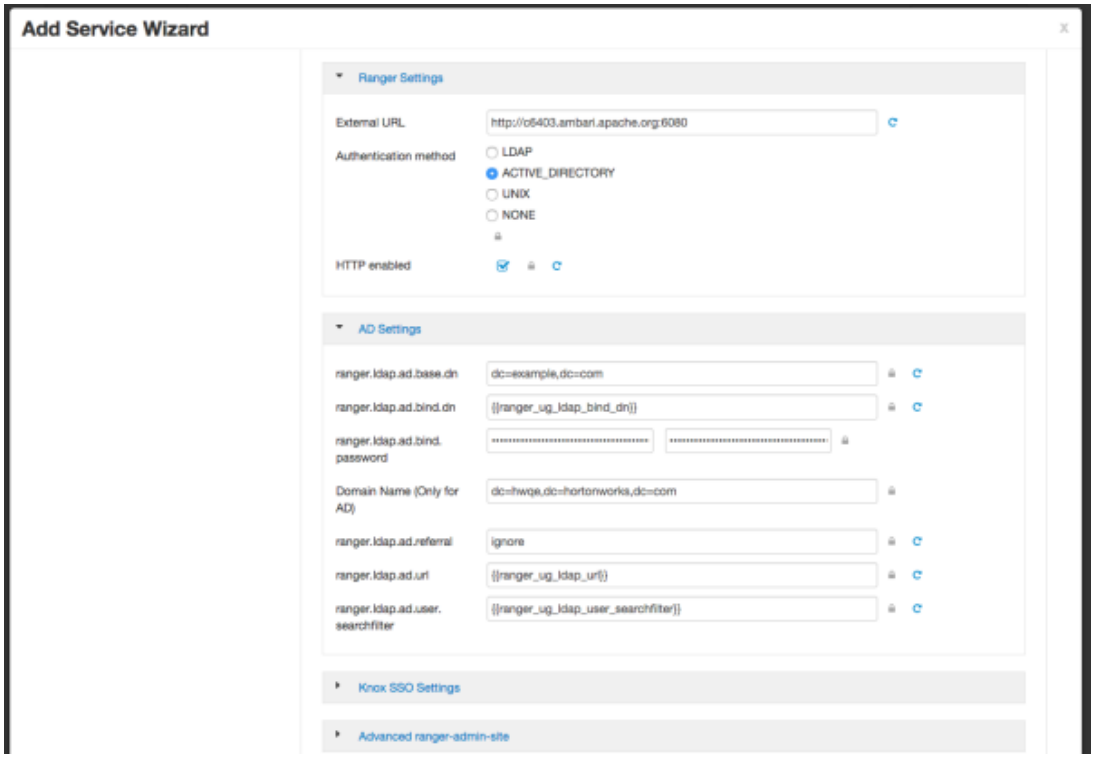
Properties with value `{{xyz}}` are macro variables that are derived from other specified values in order to streamline the configuration process. Macro variables can be edited if required – if you need to restore the original value, click the Set Recommended symbol at the right of the property box.

There are three possible values for `ranger.ldap.ad.referral`: `follow`, `throw`, and `ignore`. The recommended setting is `follow`.

When searching a directory, the server might return several search results, along with a few continuation references that show where to obtain further results. These results and references might be interleaved at the protocol level.

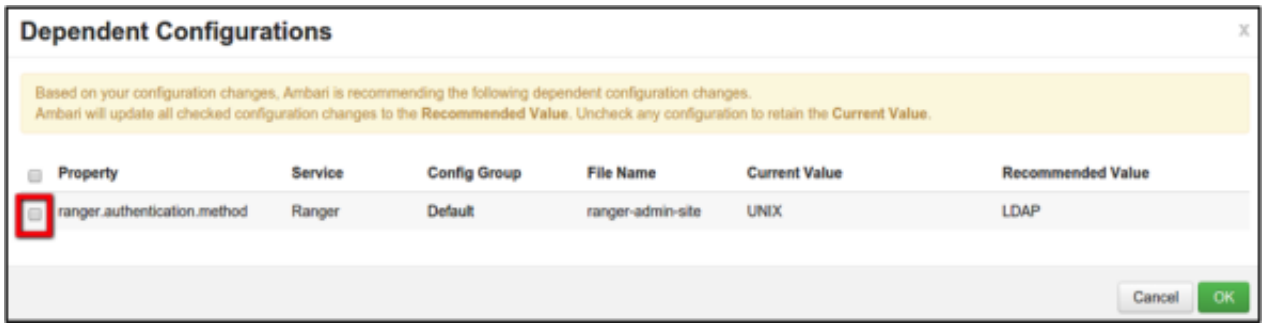
- When this property is set to `follow`, the AD service provider processes all of the normal entries first, and then follows the continuation references.
- When this property is set to `throw`, all of the normal entries are returned in the enumeration first, before the `ReferralException` is thrown. By contrast, a "referral" error response is processed immediately when this property is set to `follow` or `throw`.

- When this property is set to `ignore`, it indicates that the server should return referral entries as ordinary entries (or plain text). This might return partial results for the search. In the case of AD, a `PartialResultException` is returned when referrals are encountered while search results are processed.



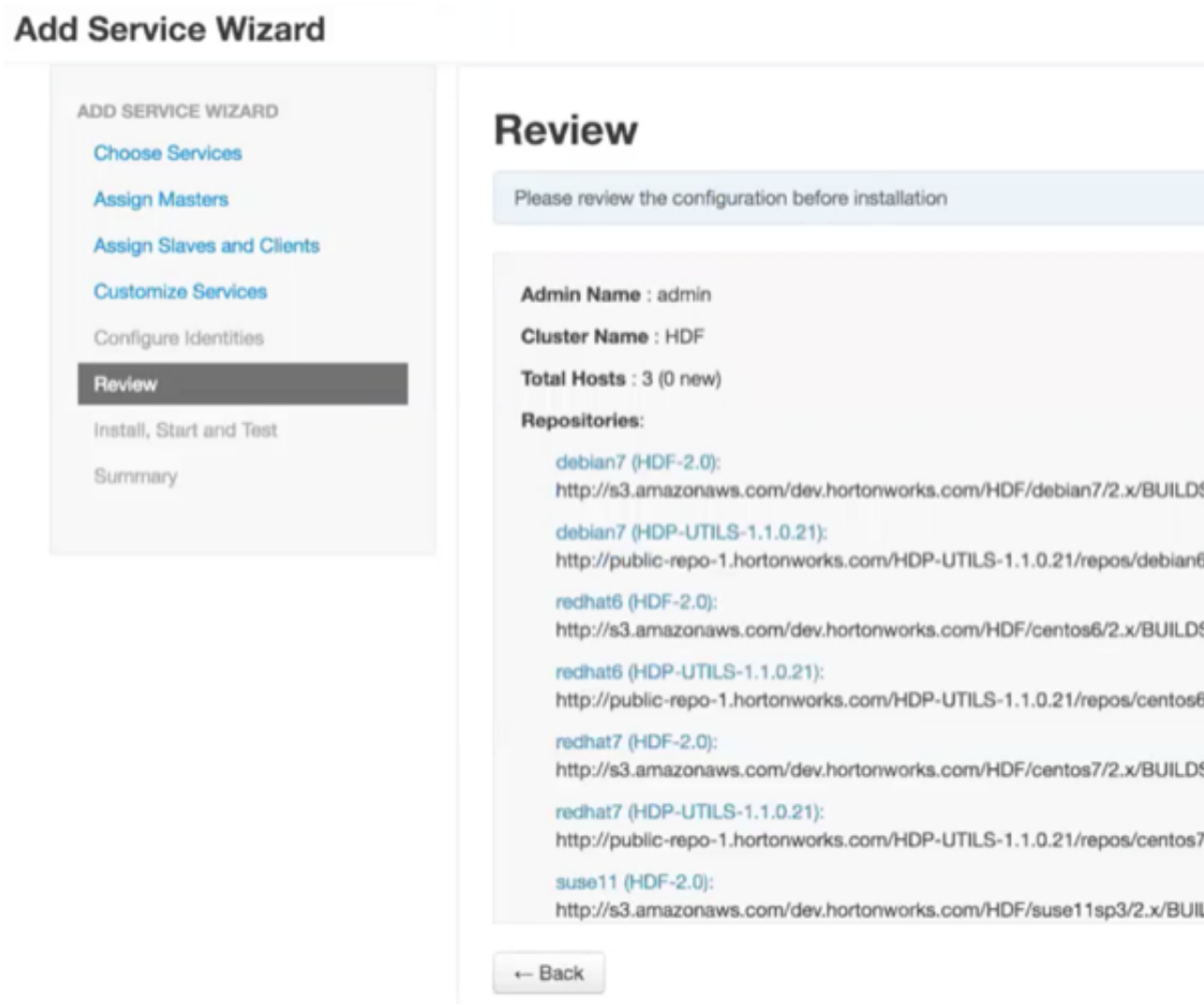
When you have finished configuring all of the Customize Services Settings, click **Next** at the bottom of the page to continue with the installation.

5. When you save the authentication method as Active Directory, a Dependent Configurations pop-up may appear recommending that you set the authentication method as LDAP. This recommended configuration should not be applied for AD, so you should clear (un-check) the **ranger.authentication.method** check box, then click **OK**.

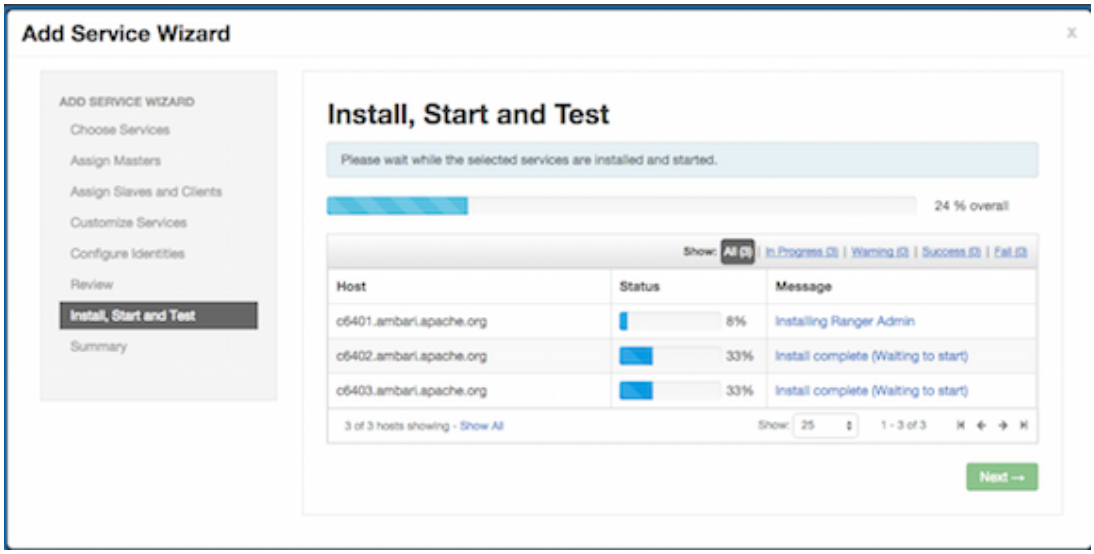


6.1.3.3. Complete the Ranger Installation

1. On the Review page, carefully review all of your settings and configurations. If everything looks good, click **Deploy** to install Ranger on the Ambari server.



2. When you click **Deploy**, Ranger is installed on the specified host on your Ambari server. A progress bar displays the installation progress.



3. When the installation is complete, a Summary page displays the installation details. You may need to restart services for cluster components after installing Ranger.

Note

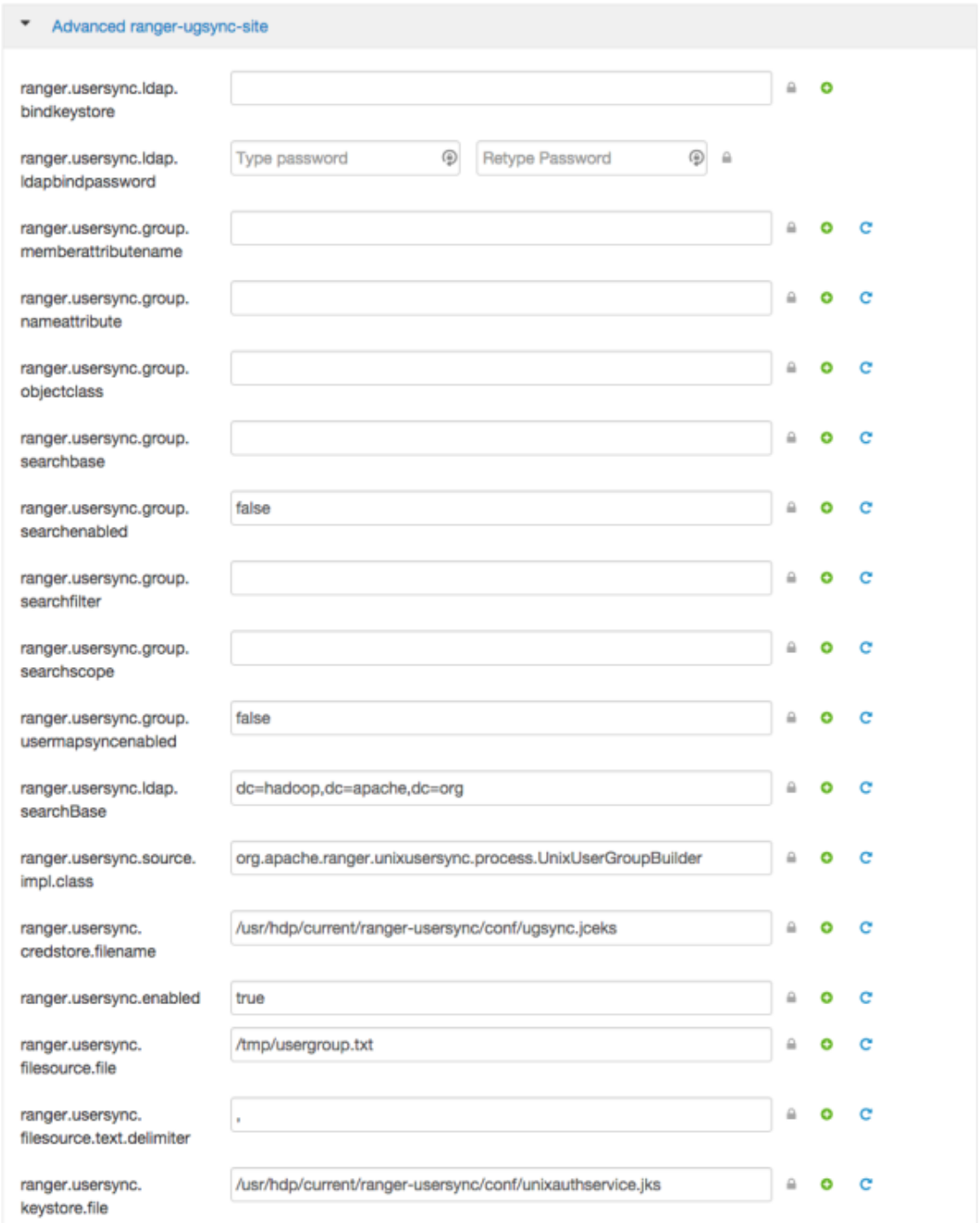
If the installation fails, you should complete the installation process, then reconfigure and reinstall Ranger.

6.1.3.4. Advanced Usersync Settings

To access Usersync settings, select the Advanced tab on the Customize Service page. Usersync pulls in users from UNIX, LDAP, or AD and populates Ranger's local user tables with these users.

6.1.3.4.1. UNIX Usersync Settings

If you are using UNIX authentication, the default values for the Advanced ranger-ugsync-site properties are the settings for UNIX authentication.



6.1.3.4.2. Required LDAP and AD Usersync Settings

If you are using LDAP authentication, you must update the following Advanced ranger-ugsync-site properties.

Table 6.13. LDAP Advanced ranger-ugsync-site Settings

Property Name	LDAP Value
ranger.usersync.ldap.bindkeystore	Set this to the same value as the `ranger.usersync.credstore.filename` property,

Property Name	LDAP Value
	i.e, the default value is `/usr/hdf/current/ranger-usersync/conf/ugsync.jceks`
ranger.usersync.ldap.bindalias	ranger.usersync.ldap.bindalias
ranger.usersync.source.impl.class	ldap

Table 6.14. AD Advanced ranger-ugsync-site Settings

Property Name	LDAP Value
ranger.usersync.source.impl.class	ldap

6.1.3.4.3. Additional LDAP and AD Usersync Settings

If you are using LDAP or Active Directory authentication, you may need to update the following properties, depending upon your specific deployment characteristics.

Table 6.15. Advanced ranger-ugsync-site Settings for LDAP and AD

Property Name	LDAP ranger-ugsync-site Value	AD ranger-ugsync-site Value
ranger.usersync.ldap.url	ldap://127.0.0.1:389	ldap://ad-conrowoller-hostname:389
ranger.usersync.ldap.binddn	cn=ldapadmin,ou=users, dc=example,dc=com	cn=adadmin,cn=Users, dc=example,dc=com
ranger.usersync.ldap.ldapbindpassword	secret	secret
ranger.usersync.ldap.searchBase	dc=example,dc=com	dc=example,dc=com
ranger.usersync.source.impl.class	org.apache.ranger. ladpusersync. process.LdapUserGroupBuilder	
ranger.usersync.ldap.user.searchbase	ou=users, dc=example, dc=com	dc=example,dc=com
ranger.usersync.ldap.user.searchscope	sub	sub
ranger.usersync.ldap.user.objectclass	person	person
ranger.usersync.ldap.user.searchfilter	Set to single empty space if no value. Do not leave it as “empty”	(objectcategory=person)
ranger.usersync.ldap.user.nameattribute	uid or cn	sAMAccountName
ranger.usersync.ldap.user.groupnameattribute	memberof,ismemberof	memberof,ismemberof
ranger.usersync.ldap.username.caseconversion	none	none
ranger.usersync.ldap.groupname.caseconversion	none	none
ranger.usersync.group.searchenabled *	false	false
ranger.usersync.group.usermapsyncenabled *	false	false
ranger.usersync.group.searchbase *	ou=groups, dc=example, dc=com	dc=example,dc=com
ranger.usersync.group.searchscope *	sub	sub

Property Name	LDAP ranger-ugsync-site Value	AD ranger-ugsync-site Value
ranger.usersync.group.objectclass *	groupofnames	groupofnames
ranger.usersync.group.searchfilter *	needed for AD authentication	(member=CN={0}, OU=MyUsers, DC=AD-HDP, DC=COM)
ranger.usersync.group.nameattribute *	cn	cn
ranger.usersync.group.memberattributename *	member	member
ranger.usersync.pagedresultsenabled *	true	true
ranger.usersync.pagedresultssize *	500	500
ranger.usersync.user.searchenabled *	false	false
ranger.usersync.group.search.first.enabled *	false	false

* Only applies when you want to filter out groups.

After you have finished specifying all of the settings on the Customize Services page, click **Next** at the bottom of the page to continue with the installation.

6.1.3.5. Configuring Ranger for LDAP SSL

You can use the following steps to configure LDAP SSL using self-signed certs in the default Ranger User Sync TrustStore.

1. The default location is `/usr/hdf/current/ranger-usersync/conf/mytruststore.jks` for the `ranger.usersync.truststore.file` property.
2. Alternatively, copy and edit the self-signed ca certs.
3. Set the `ranger.usersync.truststore.file` property to that new cacert file.

```
cd /usr/hdp/<version>/ranger-usersync
service ranger-usersync stop
service ranger-usersync start
```

Where `cert.pem` has the LDAPS cert.

6.1.3.6. Setting up Database Users Without Sharing DBA Credentials

If do not wish to provide system Database Administrator (DBA) account details to the Ambari Ranger installer, you can use the `dba_script.py` Python script to create Ranger DB database users without exposing DBA account information to the Ambari Ranger installer. You can then run the normal Ambari Ranger installation without specify a DBA user name and password.

To create Ranger DB users using the `dba_script.py` script:

1. Download the Ranger rpm using the yum install command.

```
yum install ranger-admin
```

2. You should see one file named `dba_script.py` in the `/usr/hdf/current/ranger-admin` directory.

3. Get the script reviewed internally and verify that your DBA is authorized to run the script.

4. Execute the script by running the following command:

```
python dba_script.py
```

5. Pass all values required in the argument. These should include `db flavor`, `JDBC jar`, `db host`, `db name`, `db user`, and other parameters.

   - If you would prefer not to pass runtime arguments via the command prompt, you can update the `/usr/hdf/current/ranger-admin/install.properties` file and then run:

   - ```
     python dba_script.py -q
     ```

     When you specify the `-q` option, the script will read all required information from the `install.properties` file

   - You can use the `-d` option to run the script in "dry" mode. Running the script in dry mode causes the script to generate a database script.

     ```
     python dba_script.py -d /tmp/generated-script.sql
     ```

     Anyone can run the script, but it is recommended that the system DBA run the script in dry mode. In either case, the system DBA should review the generated script, but should only make minor adjustments to the script, for example, change the location of a particular database file. No major changes should be made that substantially alter the script – otherwise the Ranger install may fail.

     The system DBA must then run the generated script.

6. Run the Ranger Ambari install procedure, but set **Setup Database and Database User** to **No** in the Ranger Admin section of the Customize Services page.

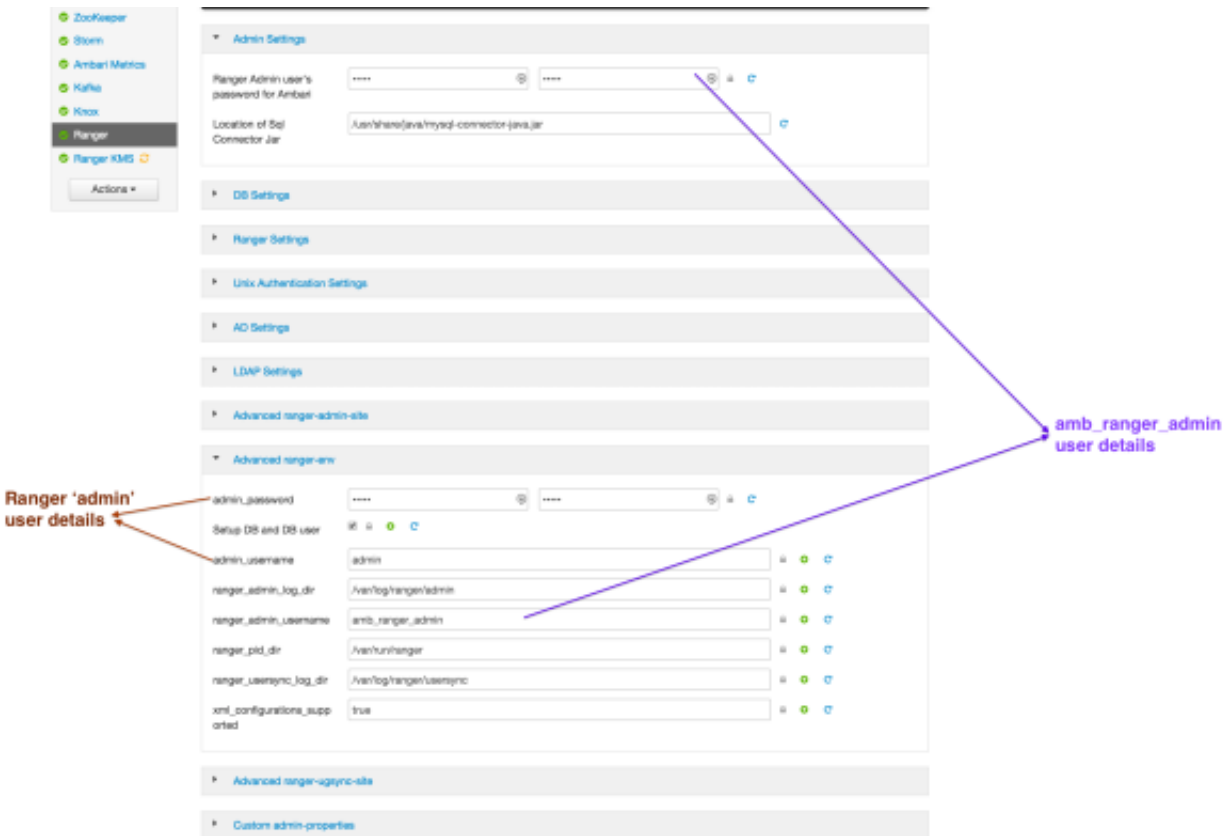
6.1.3.7. Updating Ranger Admin Passwords

For the following users, if you update the passwords on the Ranger Configs page, you must also update the passwords on the Configs page of each Ambari component that has the Ranger plugin enabled. Individual Ambari component configurations are not automatically updated – the service restart will fail if you do not update these passwords on each component.

- Ranger Admin user – The credentials for this user are set in **Configs > Advanced ranger-env** in the fields labeled **admin_username** (default value: `admin`) and **admin_password** (default value: `admin`).

- Admin user used by Ambari to create repo/policies – The user name for this user is set in **Configs > Admin Settings** in the field labeled **Ranger Admin username for Ambari** (default value: `amb_ranger_admin`). The password for this user is set in the field labeled **Ranger Admin user's password for Ambari**. This password is specified during the Ranger installation.

The following image shows the location of these settings on the Ranger Configs page:
     ```



6.1.4. Enabling Ranger Plugins

If you did not enable Ranger plugins during the initial Ranger installation, you can enable them later. This section describes how to enable each of these plugins. For performance reasons, it is recommended that you store audits in Solr, and not in a database.

If you are using a Kerberos-enabled cluster, there are a number of additional steps you must follow to ensure that you can use the Ranger plugins on a Kerberos cluster.

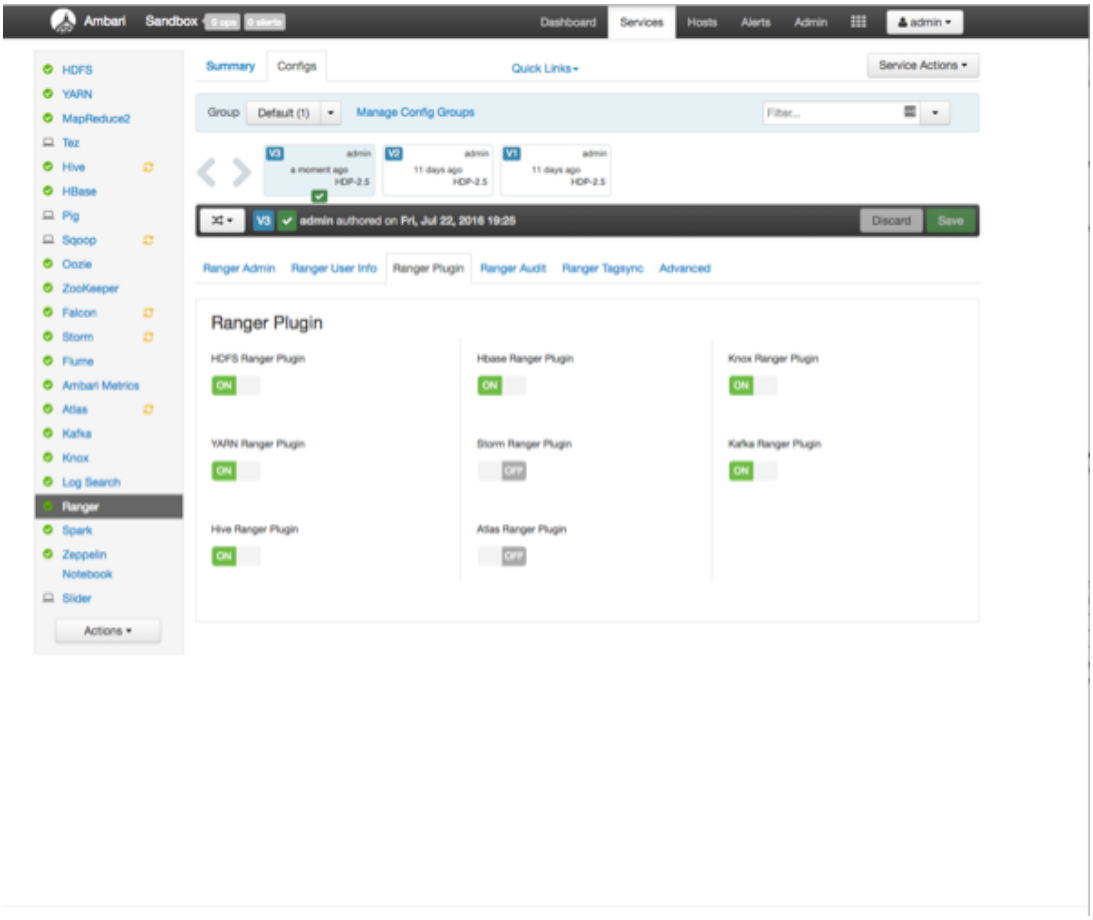
The following Ranger plugins are available:

- Kafka
- Storm
- NiFi

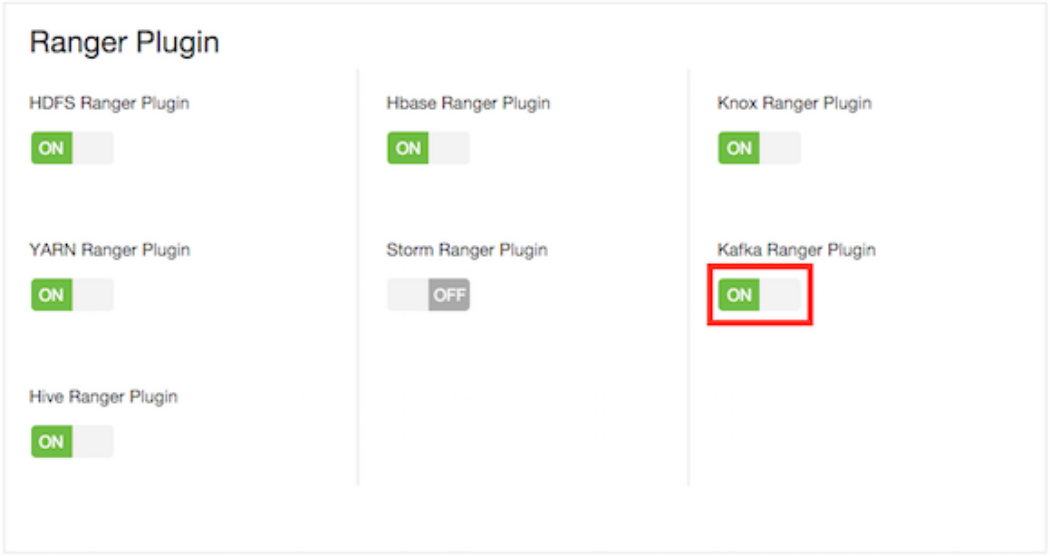
6.1.4.1. Kafka

Use the following steps to enable the Ranger Kafka plugin.

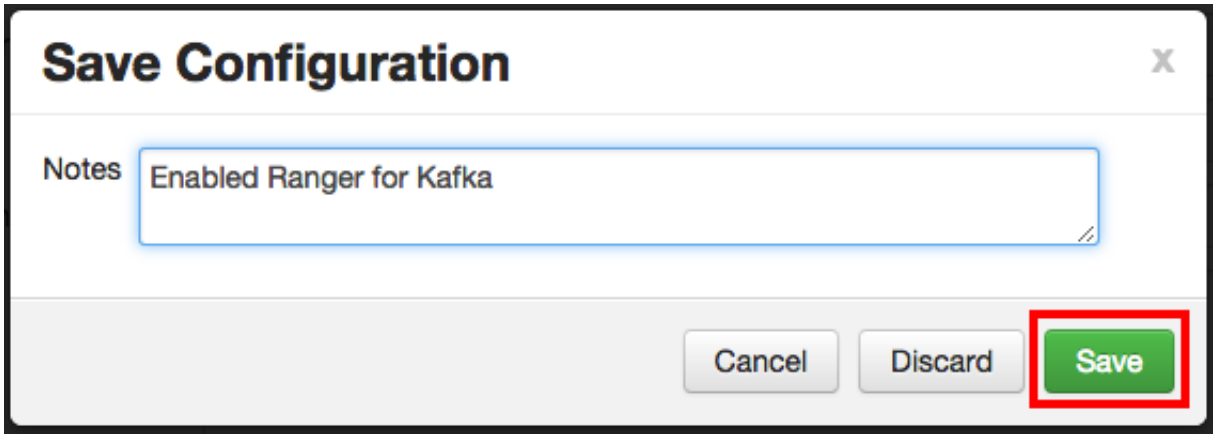
1. On the Ranger Configs page, select the **Ranger Plugin** tab.



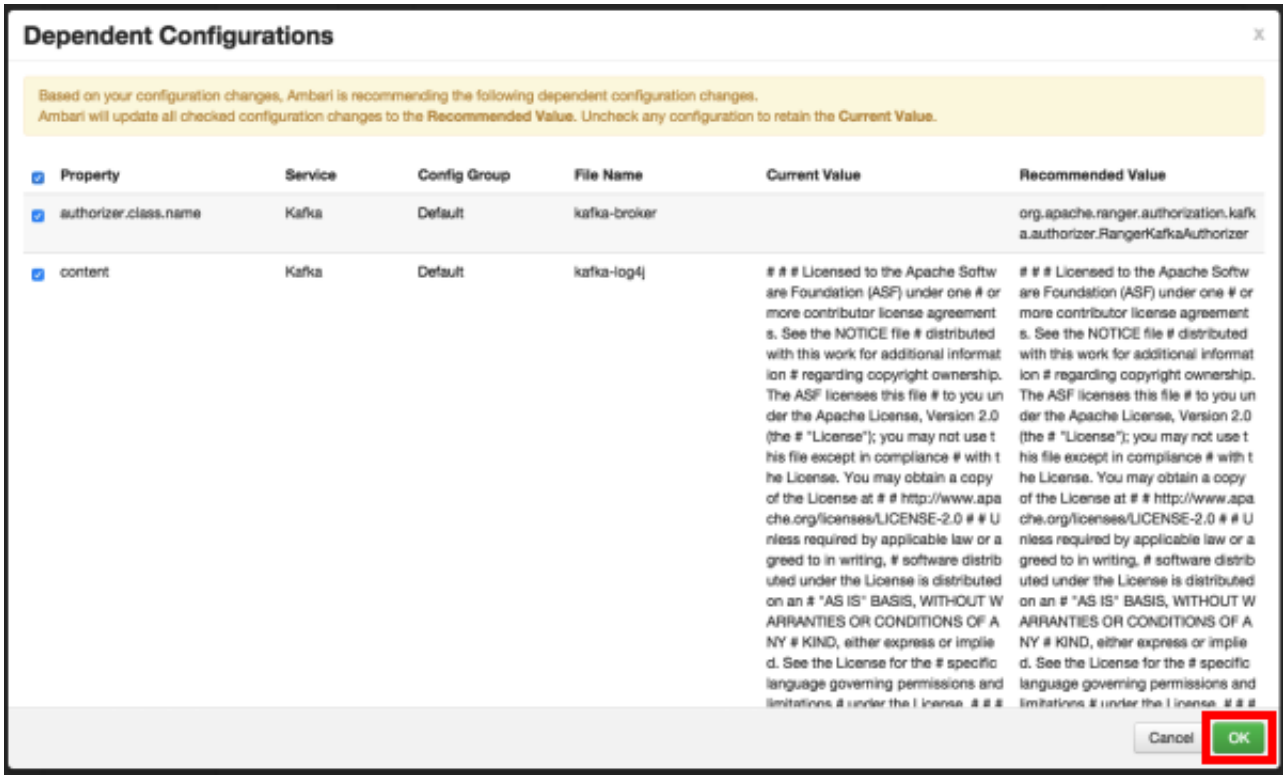
2. Under Kafka Ranger Plugin, select **On**, then click **Save** in the black menu bar.



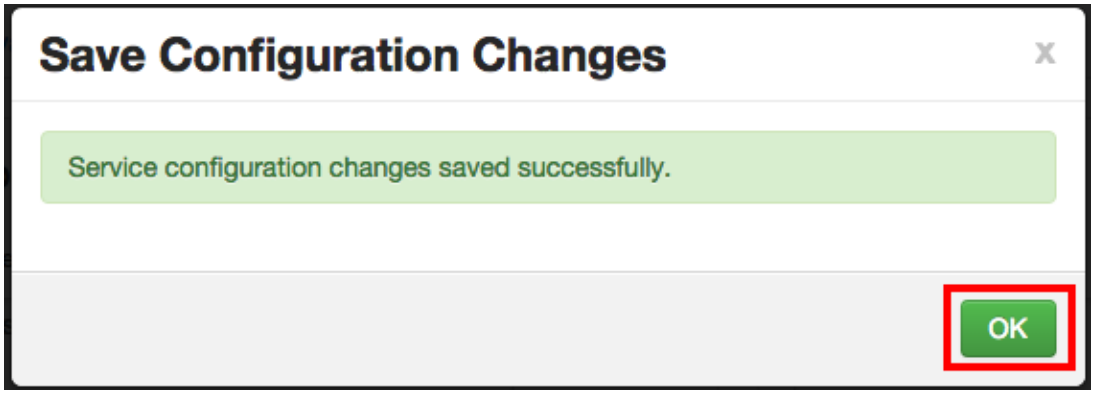
3. A Save Configuration pop-up appears. Type in a note describing the changes you just made, then click **Save**.



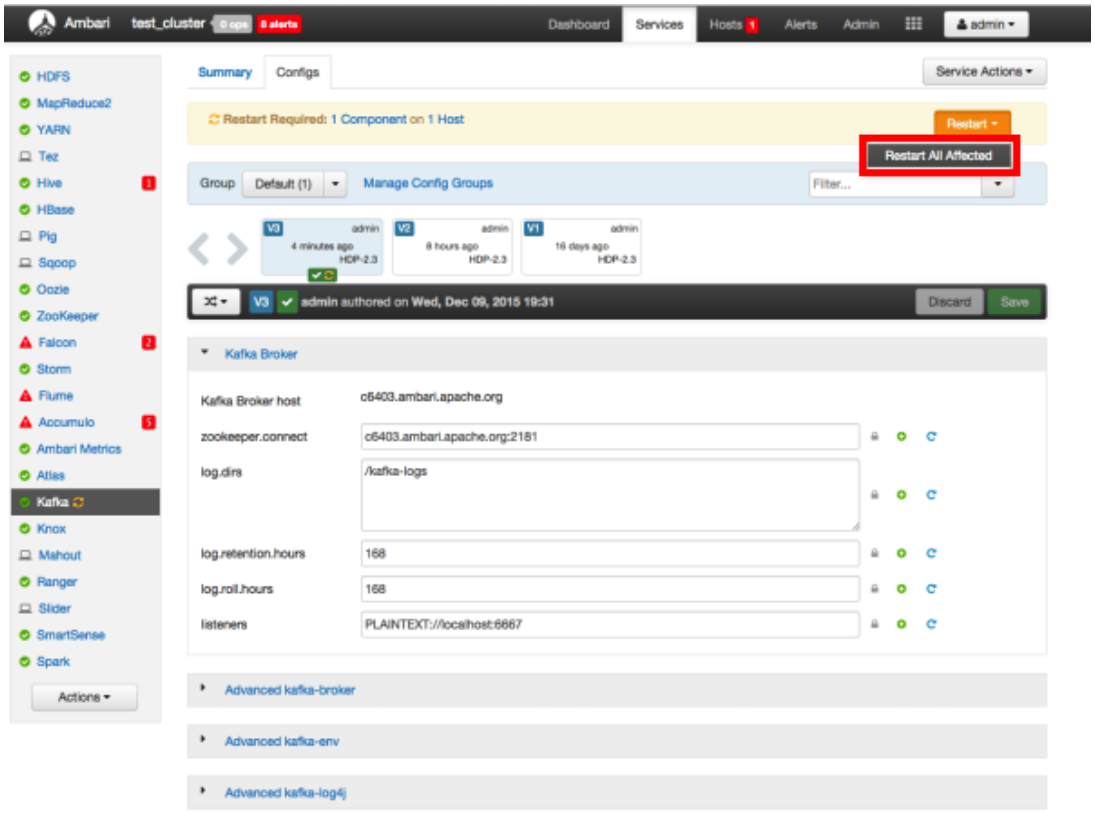
4. A Dependent Configuration pop-up appears. Click **OK** to confirm the configuration updates.



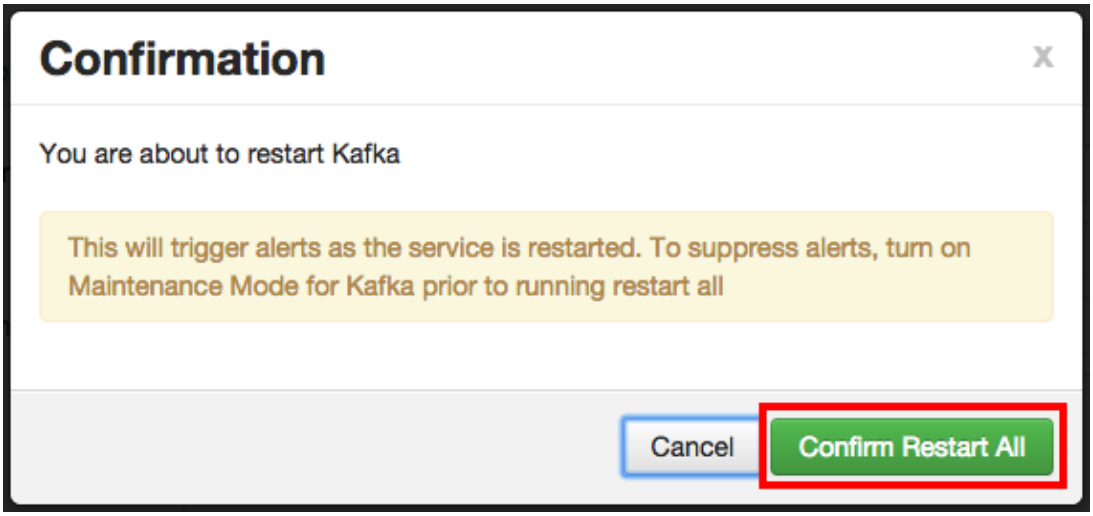
5. Click **OK** on the Save Configuration Changes pop-up.



6. Select **Kafka** in the navigation menu, then select **Restart > Restart All Affected** to restart the Kafka service and load the new configuration.



7. Click **Confirm Restart All** on the confirmation pop-up to confirm the Kafka restart.



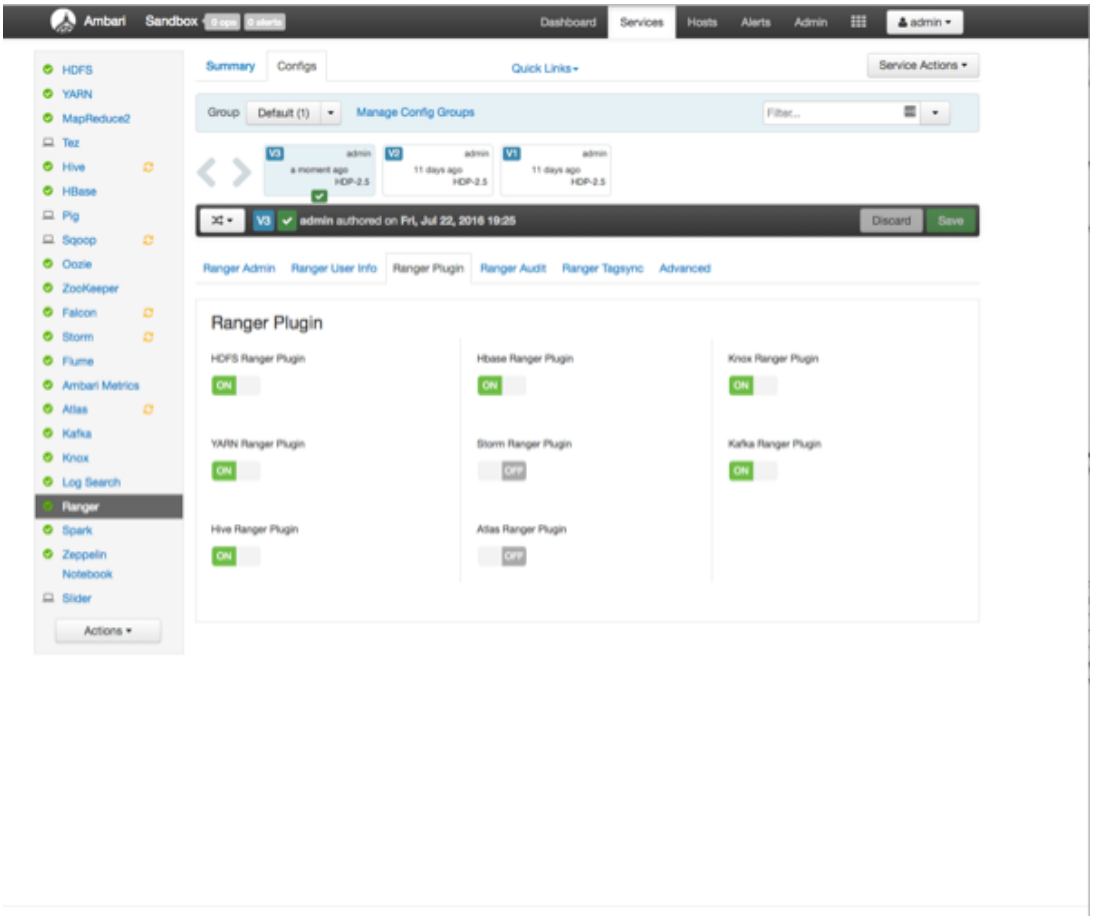
8. After Kafka restarts, the Ranger plugin for Kafka will be enabled.

6.1.4.2. Storm

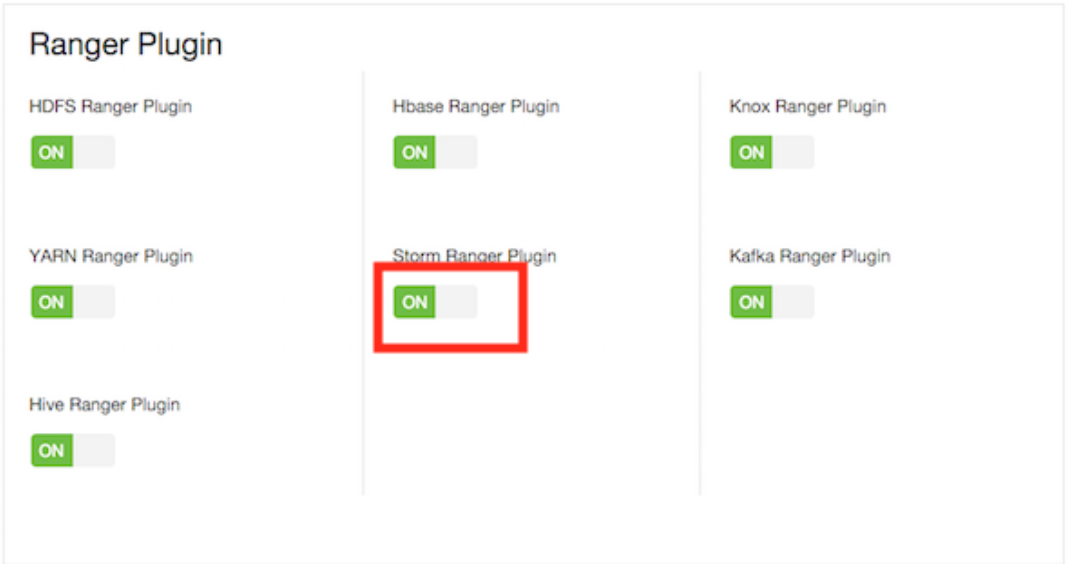
Before you can use the Storm plugin, you must first enable Kerberos on your cluster. To enable Kerberos on your cluster, see Enabling Kerberos Authentication Using Ambari.

Use the following steps to enable the Ranger Storm plugin.

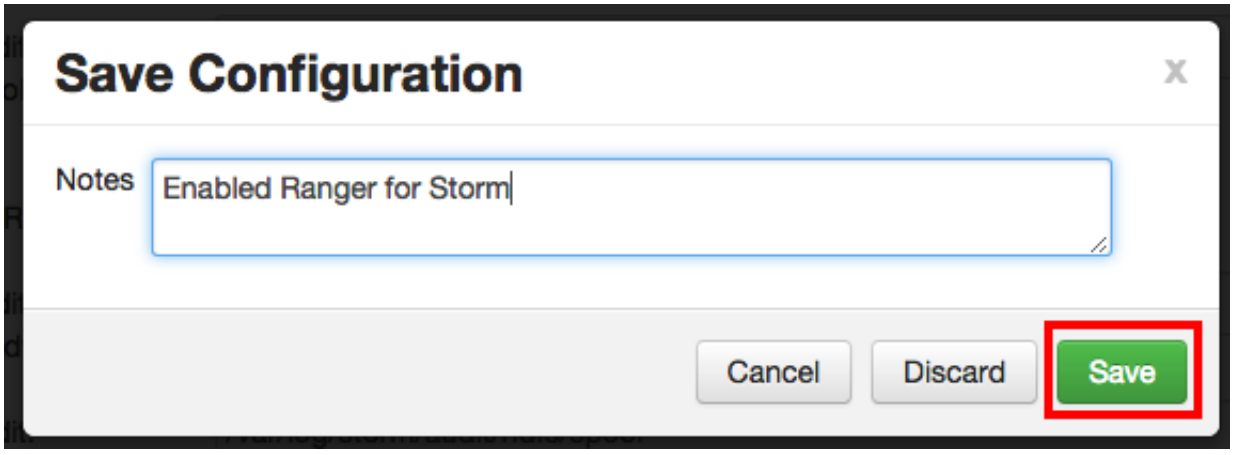
1. On the Ranger Configs page, select the **Ranger Plugin** tab.



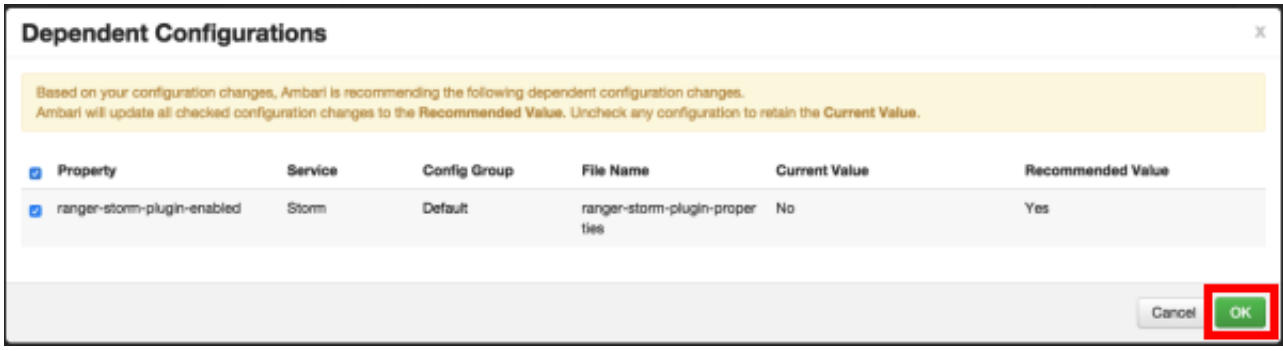
2. Under Storm Ranger Plugin, select **On**, then click **Save** in the black menu bar.



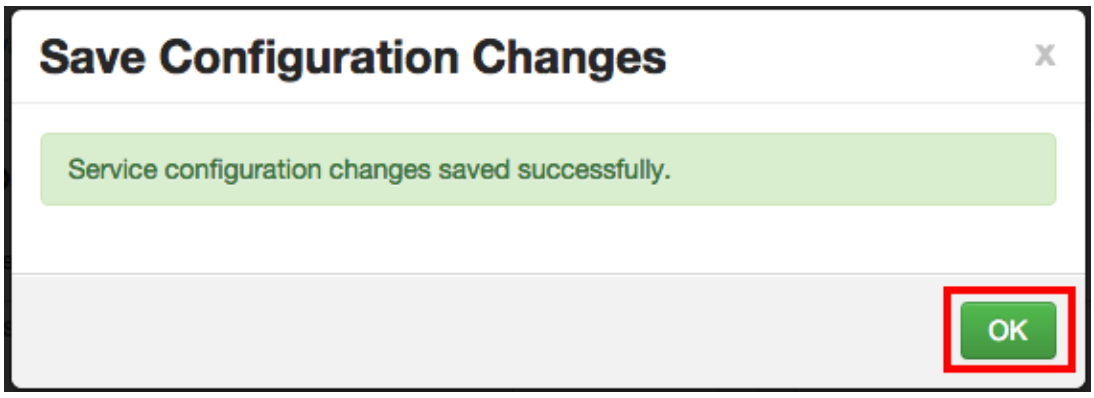
3. A Save Configuration pop-up appears. Type in a note describing the changes you just made, then click **Save**.



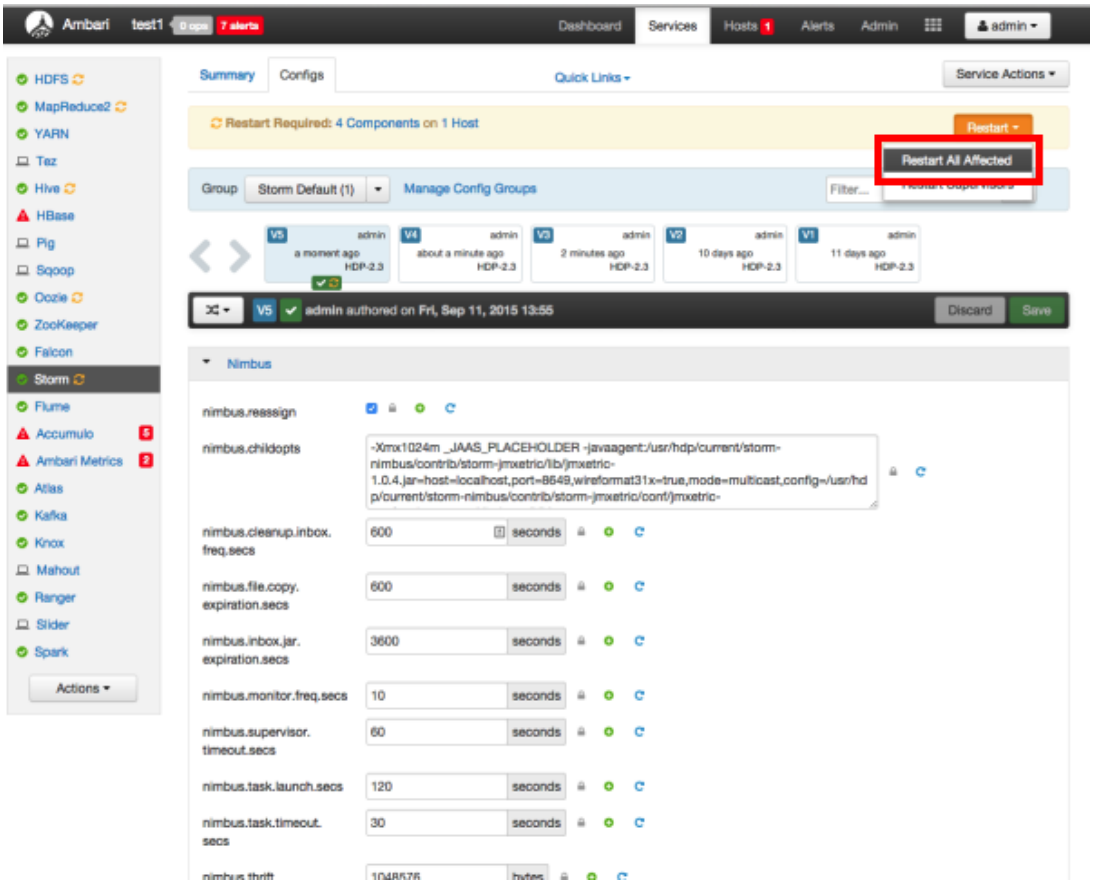
4. A Dependent Configuration pop-up appears. Click **OK** to confirm the configuration updates.



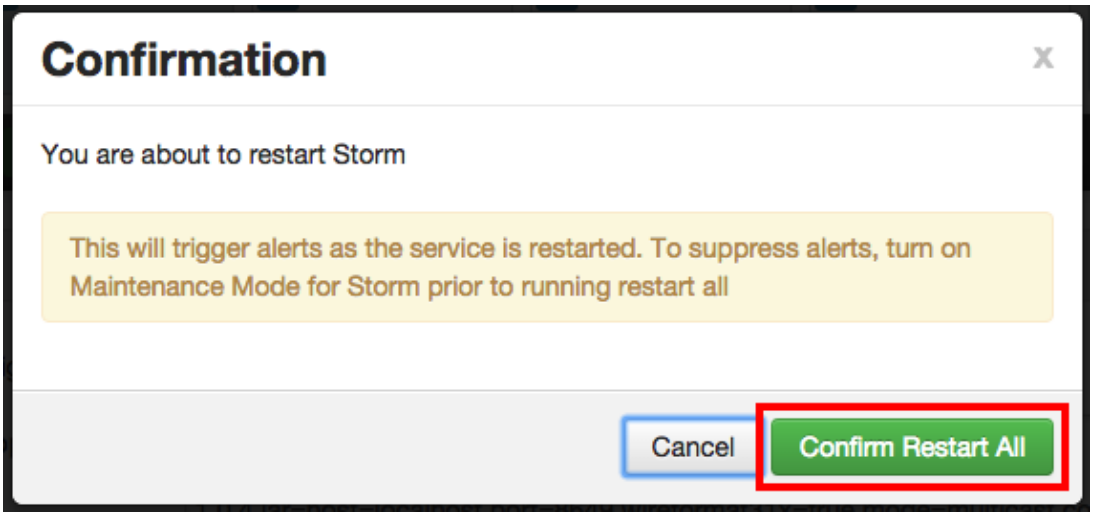
5. Click **OK** on the Save Configuration Changes pop-up.



6. Select **Storm** in the navigation menu, then select **Restart > Restart All Affected** to restart the Storm service and load the new configuration.



7. Click **Confirm Restart All** on the confirmation pop-up to confirm the Storm restart.

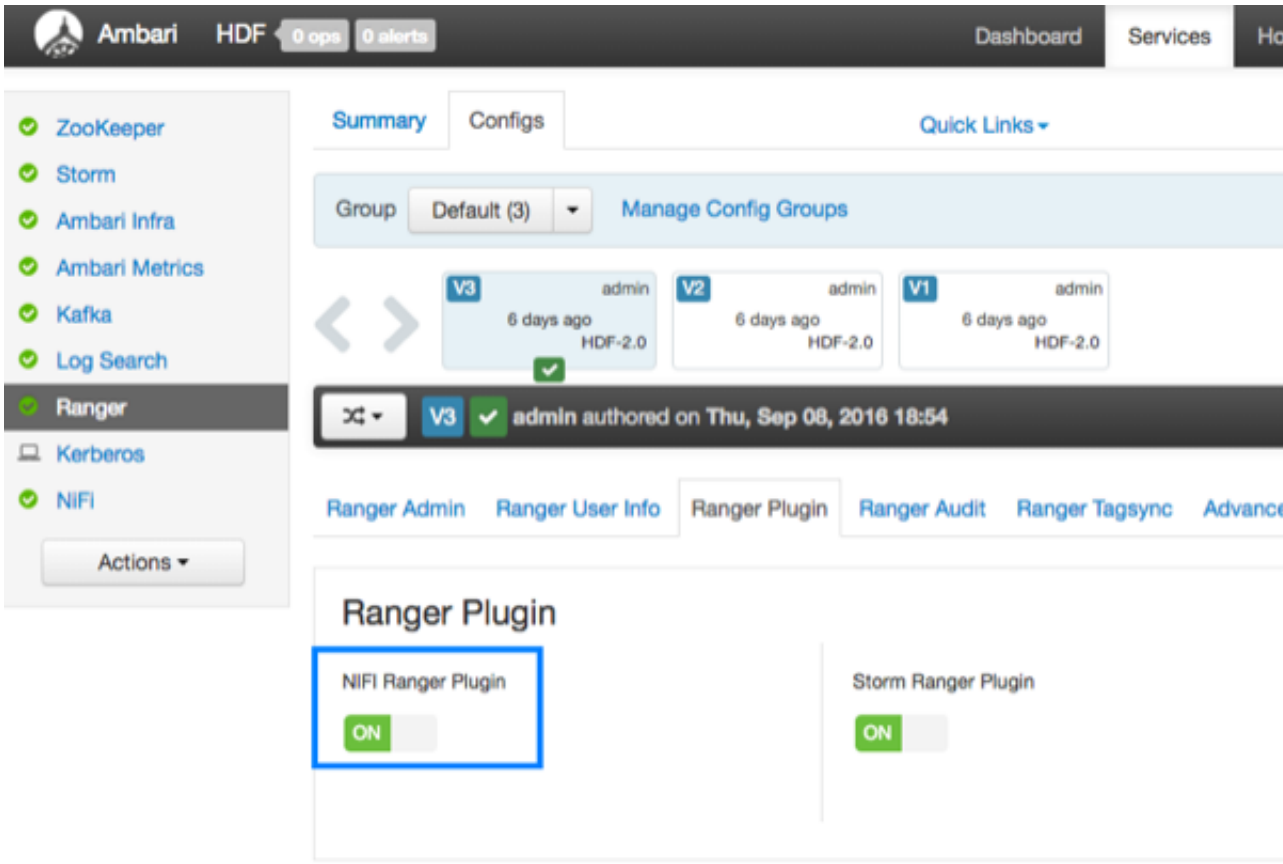


8. After Storm restarts, the Ranger plugin for Storm will be enabled.

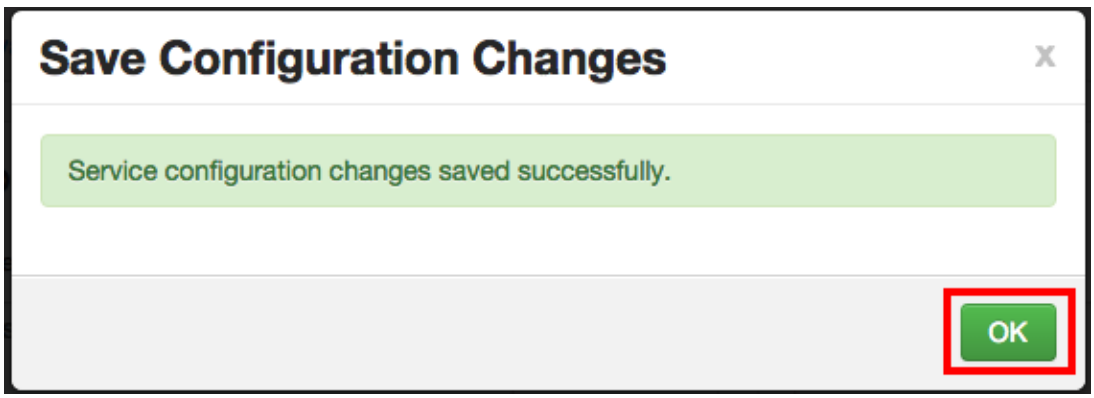
6.1.4.3. NiFi

Use the following steps to enable the Ranger NiFi plugin.

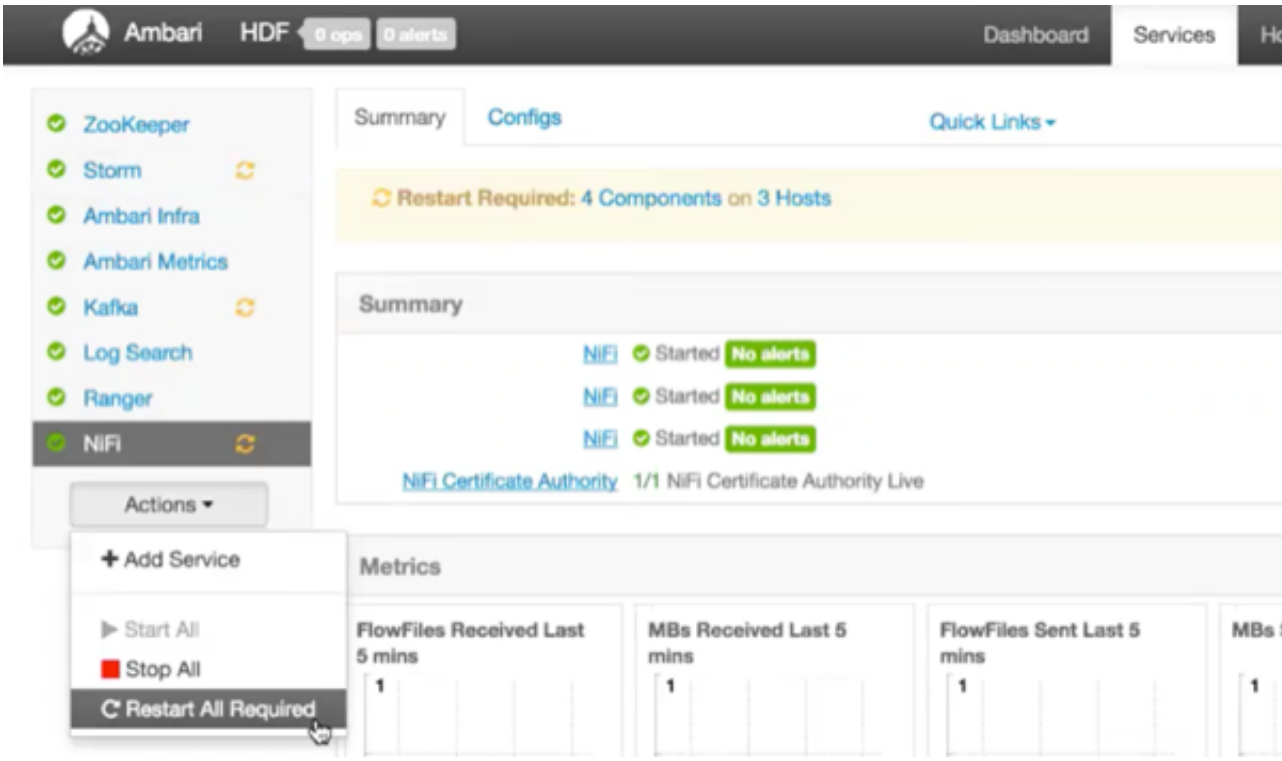
1. On the Ranger Configs page, select the **Ranger Plugin** tab.
2. Under NiFi Ranger Plugin, select **ON**, then click **Save** in the black menu bar.



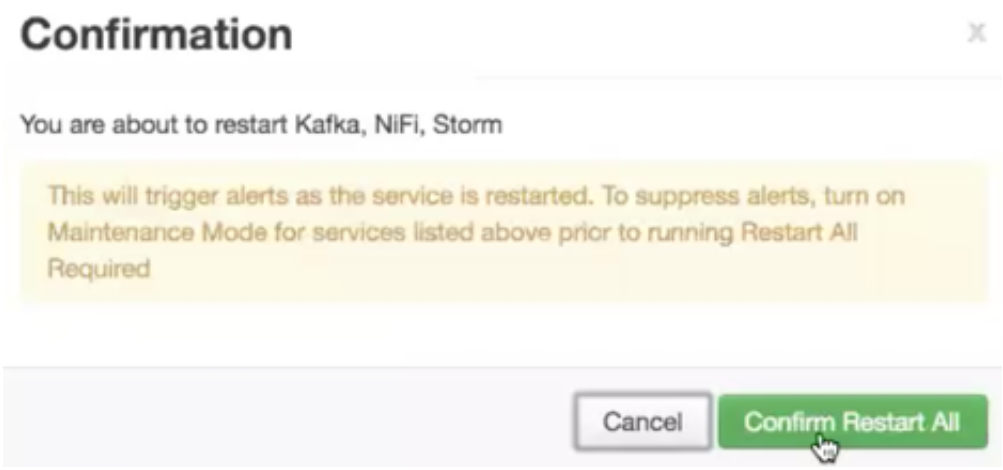
3. A Save Configuration pop-up appears. Type in a note describing the changes you just made, then click **Save**.
4. A Dependent Configuration pop-up appears. Click **OK** to confirm the configuration updates.
5. Click **OK** on the **Save Configuration Changes** pop-up.



6. From the left navigation menu click **Actions**, then **Restart All Required** to restart NiFi and any additional plugins you have enabled.



7. Click **Confirm Restart All** on the confirmation pop-up to confirm the NiFi restart along with any other servies requiring a restart.

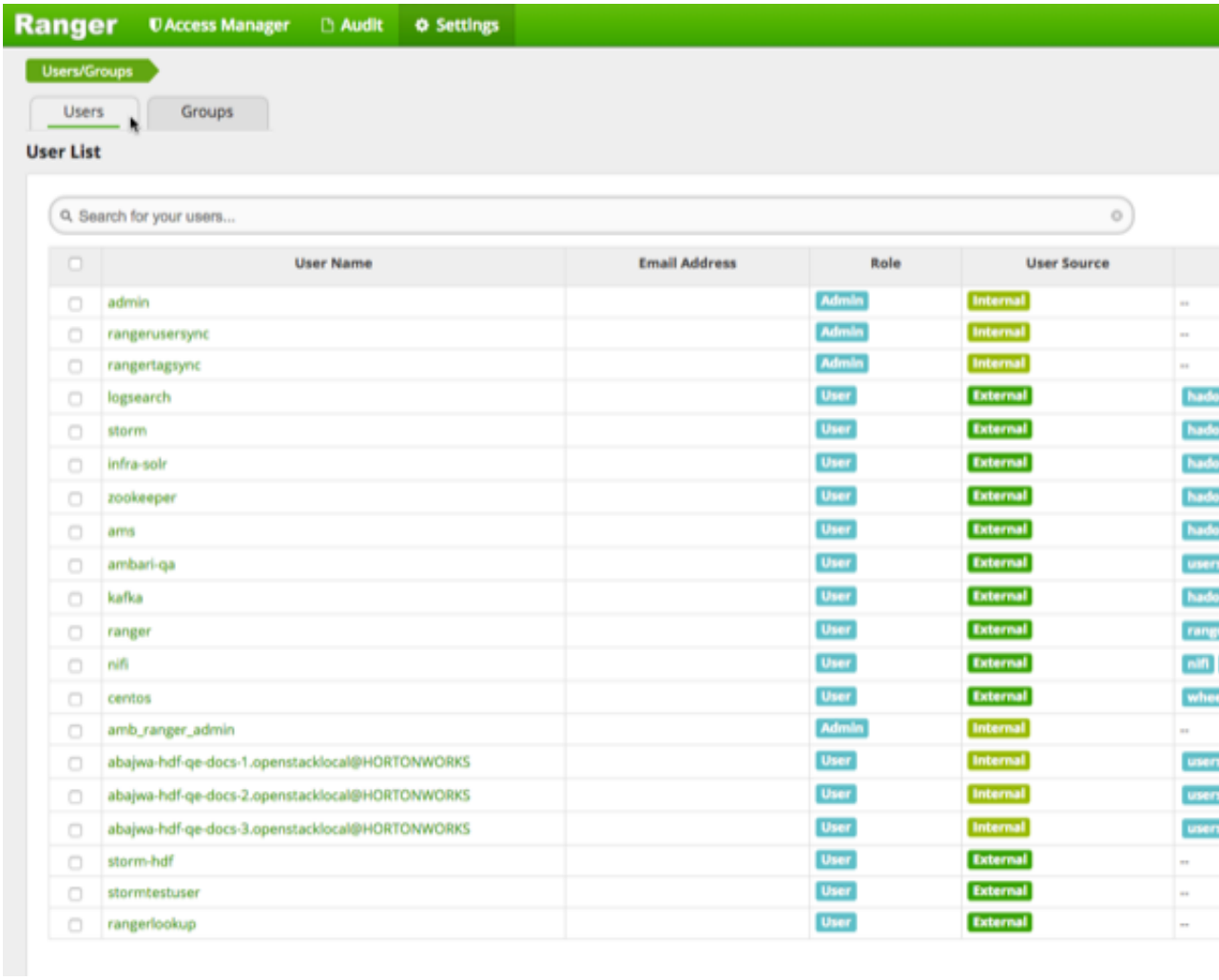


8. After your services restart, the Ranger plugin is enabled.

6.2. Adding Users to Ranger

After installing Ranger and enabling the Ranger plugins, add users to Ranger.

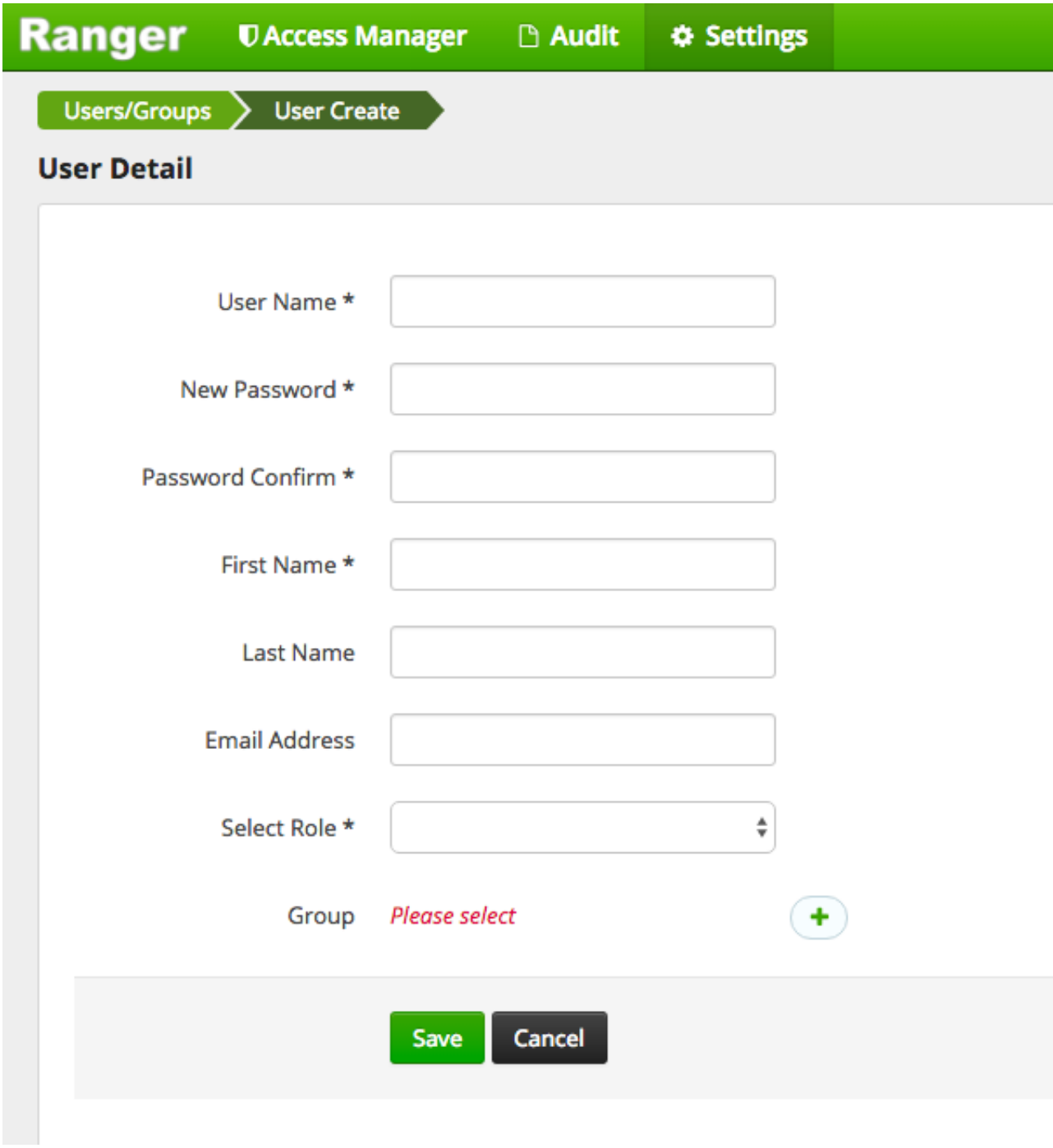
1. From the Ranger UI, click **Settings**, then **Users/Groups**.



2. Click **Add New User**.

3. In the **User Detail** screen, provide:

   - User Name in the `CN=<host> OU=<realm>` format. If you have set up identity mapping, use the `<host>@<realm>` format.
   - The password the user will use to access Ranger.
   - The Role you want the user to have.
   - The Group you want the user to be part of.



6.3. Creating Policies for NiFi Access

Once you have set up Ranger to manage NiFi authorization, you must create policies so that users can access and operate on the NiFi canvas.

- Creating Policies to View NiFi
- Allowing Users Read and Write Access to NiFi

6.3.1. Creating Policies to View NiFi

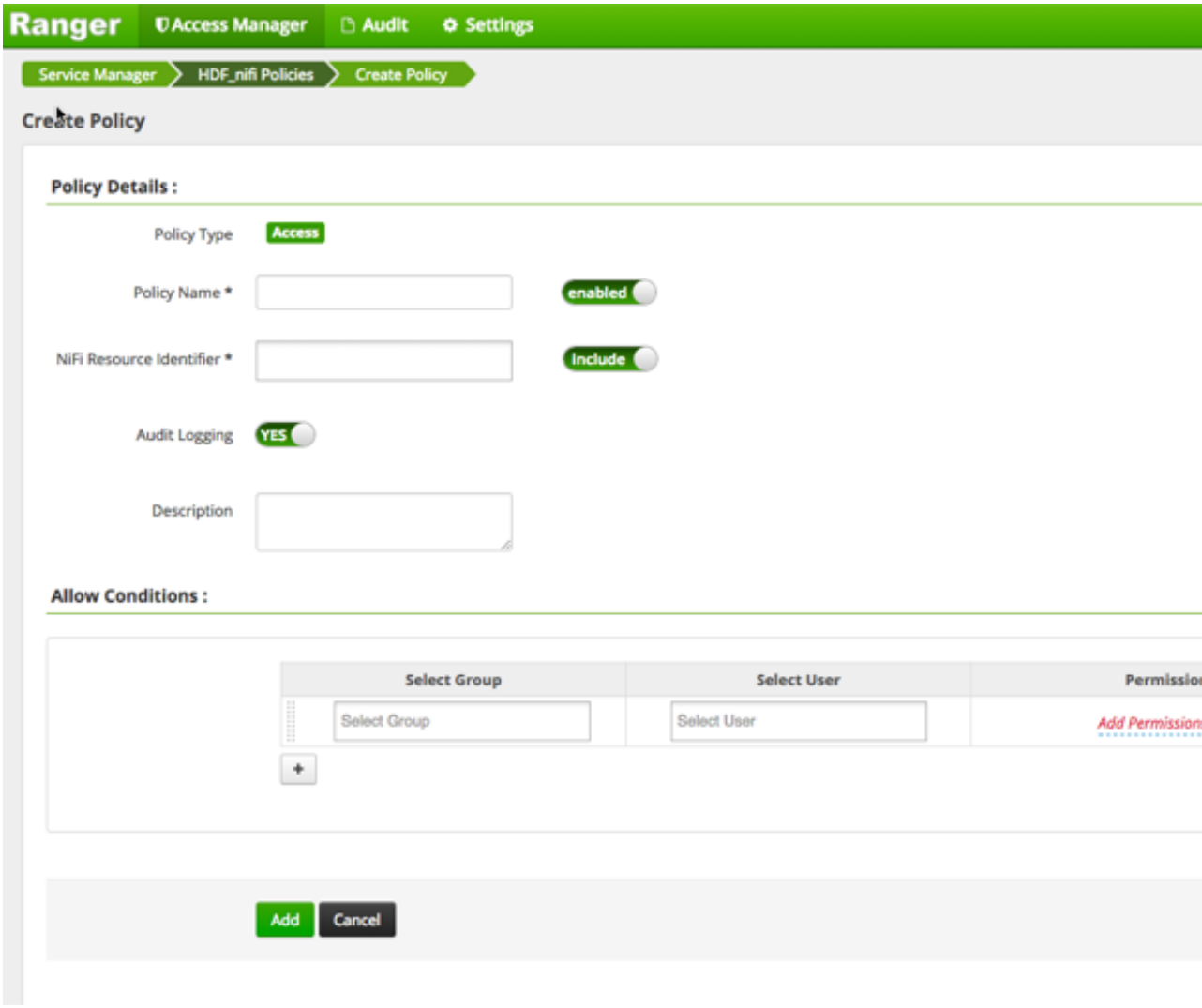
To allow users to view the NiFi UI, create the following policies for each host:

- /flow – read
- /proxy – read/write

To create policies:

1. From the Ranger console, click the NiFi Ranger plugin.

2. From the **List of Policies** page, click **Add New Policy**.
3. In the **Policy Details** dialog, create the /flow and /proxy policies.



4. To create the /flow policy:
   a. Provide the following information:
      - **Policy Name** – /flow
      - **NiFi Resource Identifier**- /flow
      - Select Users and Groups you want to immediately add.
      - Add **Read** permission
   b. Click **Add.**
5. To create the /proxy policy:
   a. Provide the following information:
      - **Policy Name** – /proxy

- **NiFi Resource Identifier**- /proxy
- Select Users and Groups you want to immediately add.
- Add **Read** and **Write** permissions.

b. Click **Add.**

6.3.2. Allowing Users Read and Write Access

To allow users complete read and write access to NiFi:

1. From the **Policy Details** page, select the global NiFi policy.
   - **Policy Name** – all - nifi-resource
   - **NiFi Resource Identifier** – x
2. Add users.
3. Add **Read** and **Write** permissions.

7. Enabling Kerberos

To enable Kerberos on Ambari, complete the following steps:

1. Installing and Configuring the KDC
2. Installing the JCE
3. Enabling Kerberos on Ambari

7.1. Installing and Configuring the KDC

Ambari is able to configure Kerberos in the cluster to work with an existing MIT KDC, or existing Active Directory installation. This section describes the steps necessary to prepare for this integration.

Note

If you do not have an existing KDC (MIT or Active Directory), Install a new MIT KDC. Installing a KDC on a cluster host ***after*** installing the Kerberos client may overwrite the krb5.conf file generated by Ambari.

You can choose to have Ambari connect to the KDC and automatically create the necessary Service and Ambari principals, generate and distribute the keytabs ("Automated Kerberos Setup"). Ambari also provides an advanced option to manually configure Kerberos. If you choose this option, you must create the principals, generate and distribute the keytabs. Ambari will not do this automatically ("Manual Kerberos Setup").

- Use an Existing MIT KDC
- Use an Existing Active Directory
- Use Manual Kerberos Setup

For convenience, use the instructions to (Optional) Install a new MIT KDC if you do not have an existing KDC available.

7.1.1. Use an Existing MIT KDC

To use an existing MIT KDC for the cluster, you must prepare the following:

- Ambari Server and cluster hosts have network access to both the KDC and KDC admin hosts.
- KDC administrative credentials are on-hand.

Note

You will be prompted to enter the KDC Admin Account credentials during the Kerberos setup so that Ambari can contact the KDC and perform the necessary

principal and keytab generation. By default, Ambari will not retain the KDC credentials unless you have configured Ambari for encrypted passwords.

7.1.2. Use an Existing Active Directory

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

- Ambari Server and cluster hosts have network access to, and be able to resolve the DNS names of, the Domain Controllers.
- Active Directory secure LDAP (LDAPS) connectivity has been configured.
- Active Directory User container for principals has been created and is on-hand. For example, "OU=Hadoop,OU=People,dc=apache,dc=org"
- Active Directory administrative credentials with delegated control of “Create, delete, and manage user accounts” on the previously mentioned User container are on-hand.

Note

You will be prompted to enter the KDC Admin Account credentials during the Kerberos setup so that Ambari can contact the KDC and perform the necessary principal and keytab generation. By default, Ambari will not retain the KDC credentials unless you have configured Ambari for encrypted passwords.

Note

If Centrify is installed and and being used on any of the servers in the cluster, it is critical that you refer to Centrify's integration guide before attempting to enable Kerberos Security on your cluster. The documentation can be found in the Centrify Server Suite documentation library, with a direct link to the Hortonworks specific PDF here.

7.1.3. Use Manual Kerberos Setup

To perform Manual Kerberos Setup, you must prepare the following:

- Cluster hosts have network access to the KDC.
- Kerberos client utilities (such as kinit) have been installed on every cluster host.
- The Java Cryptography Extensions (JCE) have been setup on the Ambari Server host and all hosts in the cluster.
- The Service and Ambari Principals will be manually created in the KDC before completing this wizard.
- The keytabs for the Service and Ambari Principals will be manually created and distributed to cluster hosts before completing this wizard.

7.1.4. (Optional) Install a new MIT KDC

The following gives a very high level description of the KDC installation process. To get more information see specific Operating Systems documentation, such as RHEL documentation, CentOS documentation, or SLES documentation.

Note

Because Kerberos is a time-sensitive protocol, all hosts in the realm must be time-synchronized, for example, by using the Network Time Protocol (NTP). If the local system time of a client differs from that of the KDC by as little as 5 minutes (the default), the client will not be able to authenticate.

Install the KDC Server

1. Install a new version of the KDC server:

   RHEL/CentOS/Oracle Linux

   ```
   yum install krb5-server krb5-libs krb5-workstation
   ```

   SLES

   ```
   zypper install krb5 krb5-server krb5-client
   ```

   Ubuntu/Debian

   ```
   apt-get install krb5-kdc krb5-admin-server
   ```

2. Using a text editor, open the KDC server configuration file, located by default here:

   ```
   vi /etc/krb5.conf
   ```

3. Change the [realms] section of this file by replacing the default “kerberos.example.com” setting for the kdc and admin_server properties with the Fully Qualified Domain Name of the KDC server host. In the following example, “kerberos.example.com” has been replaced with “my.kdc.server”.

   ```
   [realms]
    EXAMPLE.COM = {
      kdc = my.kdc.server
      admin_server = my.kdc.server
   }
   ```

4. Some components such as HUE require renewable tickets. To configure MIT KDC to support them, ensure the following settings are specified in the `libdefaults` section of the `/etc/krb5.conf` file.

   ```
   renew_lifetime = 7d
   ```

Note

For Ubuntu/Debian, the setup of the default realm for the KDC and KDC Admin hostnames is performed during the KDC server install. You can re-run

setup using dpkg-reconfigure krb5-kdc. Therefore, Steps 2 and 3 above are not needed for Ubuntu/Debian.

Create the Kerberos Database

- Use the utility kdb5_util to create the Kerberos database.

  RHEL/CentOS/Oracle Linux

  ```
  kdb5_util create -s
  ```

  SLES

  ```
  kdb5_util create -s
  ```

  Ubuntu/Debian

  ```
  krb5_newrealm
  ```

Start the KDC

- Start the KDC server and the KDC admin server.

  RHEL/CentOS/Oracle Linux 6

  ```
  /etc/rc.d/init.d/krb5kdc start
  ```

  ```
  /etc/rc.d/init.d/kadmin start
  ```

  RHEL/CentOS/Oracle Linux 7

  ```
  systemctl start krb5kdc
  ```

  ```
  systemctl start kadmin
  ```

  SLES

  ```
  rckrb5kdc start
  ```

  ```
  rckadmind start
  ```

  Ubuntu/Debian

  ```
  service krb5-kdc restart
  ```

  ```
  service krb5-admin-server restart
  ```

Important

When installing and managing your own MIT KDC, it is ***very important*** to ***set up the KDC server to auto-start on boot***. For example:

RHEL/CentOS/Oracle Linux 6

```
chkconfig krb5kdc on
```

```
chkconfig kadmin on
```

RHEL/CentOS/Oracle Linux 7

```
systemctl enable krb5kdc
```

```
systemctl enable kadmin
```

SLES

```
chkconfig rckrb5kdc on
```

```
chkconfig rckadmind on
```

Create a Kerberos Admin

Kerberos principals can be created either on the KDC machine itself or through the network, using an “admin” principal. The following instructions assume you are using the KDC machine and using the `kadmin.local` command line administration utility. Using `kadmin.local` on the KDC machine allows you to create principals without needing to create a separate "admin" principal before you start.

Note

You will need to provide these admin account credentials to Ambari when enabling Kerberos. This allows Ambari to connect to the KDC, create the cluster principals and generate the keytabs.

1. Create a KDC admin by creating an admin principal.

   ```
   kadmin.local -q "addprinc admin/admin"
   ```

2. Confirm that this admin principal has permissions in the KDC ACL. Using a text editor, open the KDC ACL file:

   RHEL/CentOS/Oracle Linux

   ```
   vi /var/kerberos/krb5kdc/kadm5.acl
   ```

   SLES

   ```
   vi /var/lib/kerberos/krb5kdc/kadm5.acl
   ```

   Ubuntu/Debian

   ```
   vi /etc/krb5kdc/kadm5.acl
   ```

3. Ensure that the KDC ACL file includes an entry so to allow the admin principal to administer the KDC for your specific realm. When using a realm that is different than EXAMPLE.COM, **be sure there is an entry for the realm you are using**. If not present, principal creation will fail. For example, for an admin/admin@HADOOP.COM principal, you should have an entry:

   ```
   */admin@HADOOP.COM *
   ```

4. After editing and saving the kadm5.acl file, you must restart the kadmin process.

   RHEL/CentOS/Oracle Linux 6

   ```
   /etc/rc.d/init.d/kadmin restart
   ```

   RHEL/CentOS/Oracle Linux 7

   ```
   systemctl restart kadmin
   ```

   SLES

   ```
   rckadmind restart
   ```

   Ubuntu/Debian

   ```
   service krb5-admin-server restart
   ```

7.2. Installing the JCE

Before enabling Kerberos in the cluster, you must deploy the Java Cryptography Extension (JCE) security policy files on the Ambari Server and on all hosts in the cluster.

Important

If you are using Oracle JDK, **you must** distribute and install the JCE **on all hosts** in the cluster, including the Ambari Server. **Be sure to restart Ambari Server after installng the JCE**. If you are using OpenJDK, some distributions of the OpenJDK come with unlimited strength JCE automatically and therefore, installation of JCE is not required.

7.2.1. Install the JCE

1. On the Ambari Server, obtain the JCE policy file appropriate for the JDK version in your cluster.

   - For Oracle JDK 1.8:

     http://www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html

   - For Oracle JDK 1.7:

     http://www.oracle.com/technetwork/java/javase/downloads/jce-7-download-432124.html

2. Save the policy file archive in a temporary location.

3. On Ambari Server and on each host in the cluster, add the unlimited security policy JCE jars to `$JAVA_HOME/jre/lib/security/`.

   For example, run the following to extract the policy jars into the JDK installed on your host:

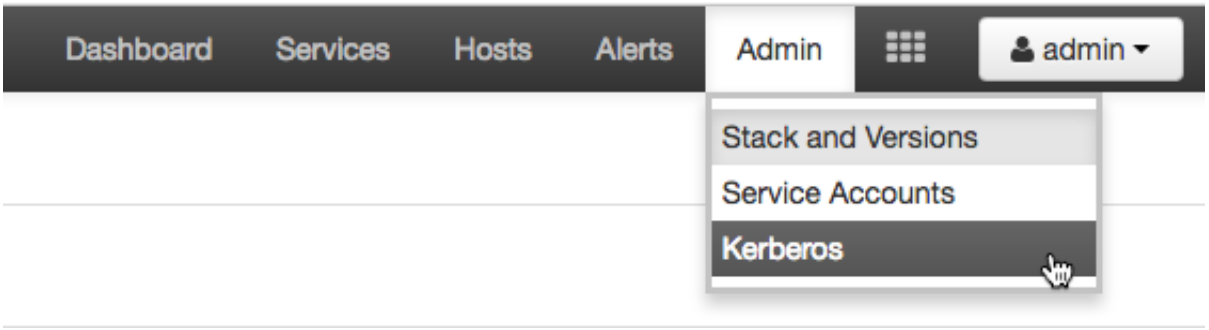
```
unzip -o -j -q jce_policy-8.zip -d /usr/jdk64/jdk1.8.0_60/jre/lib/security/
```

4. Restart Ambari Server.

7.3. Enabling Kerberos on Ambari

Once you have completed the prerequisites, you are ready to enable Kerberos for Ambari.

1. From the Ambari UI, click **Admin**, and select **Kerberos**.



2. Click **Enable Kerberos** to launch the **Enable Kerberos Wizard**.
3. From the **Get Started** screen, select the type of KDC you want to use.
4. Provide information about the KDC and admin account.
   a. In the **KDC** section, enter the following information:
      - In the **KDC Host** field, the IP address or FQDN for the KDC host. Optionally a port number may be included.
      - In the **Realm name** field, the default realm to use when creating service principals.
      - (Optional) In the **Domains** field, provide a list of patterns to use to map hosts in the cluster to the appropriate realm. For example, if your hosts have a common domain in their FQDN such as host1.hortonworks.local and host2.hortonworks.local, you would set this to:

        ```
        .hortonworks.local,hortonworks.local
        ```

   b. In the **Kadmin** section, enter the following information:
      - In the **Kadmin Host** field, the IP address or FQDN for the KDC administrative host. Optionally a port number may be included.
      - The **Admin principal** and **password** that will be used to create principals and keytabs.
      - (Optional) If you have configured Ambari for encrypted passwords, the **Save Admin Credentials** option will be enabled. With this option, you can have Ambari store the KDC Admin credentials to use when making cluster changes.

5. From the **Install and Test Kerberos Client** page, proceed with the install. Click **Next** when complete.

6. From the **Configure Identities** page, you can customize the Kerberos identities as needed, and proceed to kerberize the cluster.

   Be sure to review the principal names, particularly the **Ambari Principals** on the **General** tab. These principal names, by default, append the name of the cluster to each of the Ambari principals. You can leave this as default or adjust these by removing the "-${**cluster-name**}" from principal name string.

   Click the **Advanced** tab to review the principals and keytabs for each service.

7. Confirm your configurations, and click next to proceed kerberizing your cluster.