

# Hortonworks Data Platform

## Apache Ambari Installation

(November 15, 2018)

## Hortonworks Data Platform: Apache Ambari Installation

Copyright © 2012-2018 Hortonworks, Inc. Some rights reserved.

The Hortonworks Data Platform, powered by Apache Hadoop, is a massively scalable and 100% open source platform for storing, processing and analyzing large volumes of data. It is designed to deal with data from many sources and formats in a very quick, easy and cost-effective manner. The Hortonworks Data Platform consists of the essential set of Apache Hadoop projects including MapReduce, Hadoop Distributed File System (HDFS), HCatalog, Pig, Hive, HBase, ZooKeeper and Ambari. Hortonworks is the major contributor of code and patches to many of these projects. These projects have been integrated and tested as part of the Hortonworks Data Platform release process and installation and configuration tools have also been included.

Unlike other providers of platforms built using Apache Hadoop, Hortonworks contributes 100% of our code back to the Apache Software Foundation. The Hortonworks Data Platform 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 Data Platform](#) 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.



Except where otherwise noted, this document is licensed under  
**Creative Commons Attribution ShareAlike 4.0 License.**  
<http://creativecommons.org/licenses/by-sa/4.0/legalcode>

# Table of Contents

1. Getting Ready .....	1
1.1. Product Interoperability .....	1
1.2. Meet Minimum System Requirements .....	1
1.2.1. Software Requirements .....	1
1.2.2. Memory Requirements .....	2
1.2.3. Package Size and Inode Count Requirements .....	2
1.2.4. Maximum Open Files Requirements .....	3
1.3. Collect Information .....	3
1.4. Prepare the Environment .....	4
1.4.1. Set Up Password-less SSH .....	4
1.4.2. Set Up Service User Accounts .....	5
1.4.3. Enable NTP on the Cluster and on the Browser Host .....	5
1.4.4. Check DNS and NSCD .....	6
1.4.5. Configuring iptables .....	7
1.4.6. Disable SELinux and PackageKit and check the umask Value .....	8
1.4.7. Download and set up database connectors .....	9
2. Using a Local Repository .....	22
2.1. Setting Up a Local Repository .....	22
2.1.1. Preparing to Set Up a Local Repository .....	22
2.1.2. Setting up a Local Repository with Temporary Internet Access .....	23
2.1.3. Setting Up a Local Repository with No Internet Access .....	26
2.2. Preparing the Ambari Repository Configuration File to Use the Local Repository .....	27
3. Obtaining Public Repositories .....	29
3.1. Ambari Repositories .....	29
3.2. HDP Stack Repositories .....	30
3.2.1. HDP 3.1.0 Repositories .....	30
4. Installing Ambari .....	33
4.1. Download the Ambari Repository .....	33
4.1.1. <b>RHEL/CentOS/Oracle Linux 7</b> .....	33
4.1.2. <b>Amazon Linux 2</b> .....	34
4.1.3. <b>SLES 12</b> .....	35
4.1.4. <b>Ubuntu 14</b> .....	36
4.1.5. <b>Ubuntu 16</b> .....	37
4.1.6. <b>Ubuntu 18</b> .....	38
4.1.7. <b>Debian 9</b> .....	39
4.2. Install the Ambari Server .....	40
4.2.1. <b>RHEL/CentOS/Oracle Linux 7</b> .....	41
4.2.2. <b>SLES 12</b> .....	42
4.2.3. <b>Ubuntu 14</b> .....	43
4.2.4. <b>Ubuntu 16</b> .....	43
4.2.5. <b>Ubuntu 18</b> .....	44
4.2.6. <b>Debian 9</b> .....	44
4.3. Set Up the Ambari Server .....	45
4.3.1. Setup Options .....	47
5. Working with Management Packs .....	50
6. Installing, Configuring, and Deploying a Cluster .....	51
6.1. Start the Ambari Server .....	51

---

6.2. Log In to Apache Ambari .....	52
6.3. Launch the Ambari Cluster Install Wizard .....	53
6.4. Name Your Cluster .....	53
6.5. Select Version .....	53
6.5.1. Using a local RedHat Satellite or Spacewalk repository .....	55
6.6. Install Options .....	58
6.7. Confirm Hosts .....	59
6.8. Choose Services .....	60
6.9. Assign Masters .....	61
6.10. Assign Slaves and Clients .....	61
6.11. Customize Services .....	61
6.12. Review .....	64
6.13. Install, Start and Test .....	65
6.14. Complete .....	65

# List of Tables

6.1. Example Channel Names for Hortonworks Repositories ..... 56

# 1. Getting Ready

This section describes the information and materials you should get ready to install a cluster using Ambari. Ambari provides an end-to-end management and monitoring solution for your 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.

- [Product Interoperability \[1\]](#)
- [Meet Minimum System Requirements \[1\]](#)
- [Collect Information \[3\]](#)
- [Prepare the Environment \[4\]](#)

## 1.1. Product Interoperability

Ambari 2.7.3 supports only HDP-3.1.0 and HDF-3.2.0

The Support Matrix tool provides information about:

- Operating Systems
- Databases
- Browsers
- JDK

Use the following URL to determine support for each product version.

<https://supportmatrix.hortonworks.com>

## 1.2. Meet Minimum System Requirements

Your system must meet the following minimum requirements:

- [Software Requirements \[1\]](#)
- [Memory Requirements \[2\]](#)
- [Package Size and Inode Count Requirements \[2\]](#)
- [Maximum Open Files Requirements \[3\]](#)

### 1.2.1. Software Requirements

On each of your hosts:

- `yum` and `rpm` (RHEL/CentOS/Oracle/Amazon Linux)
- `zypper` and `php_curl` (SLES)

- apt (Debian/Ubuntu)
- scp, curl, unzip, tar, and wget
- OpenSSL (v1.01, build 16 or later)
- Python

**For SLES 12:** Python 2.7.x

**For Amazon Linux 2, CentOS 7, Ubuntu 14, Ubuntu 16, Ubuntu 18, and Debian 9:** Python 2.7.x

## 1.2.2. 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
```

If you plan to install the Ambari Metrics Service (AMS) into your cluster, you should review Using Ambari Metrics in Hortonworks Data Platform Apache Ambari Operations, for guidelines on resources requirements. In general, the host you plan to run the Ambari Metrics Collector host should have the following memory and disk space available based on cluster size:

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



### Note

Use these values as guidelines. Be sure to test them for your specific environment.

## 1.2.3. Package Size and Inode Count Requirements

	Size	Inodes
Ambari Server	100MB	5,000
Ambari Agent	8MB	1,000
Ambari Metrics Collector	225MB	4,000
Ambari Metrics Monitor	1MB	100

	Size	Inodes
Ambari Metrics Hadoop Sink	8MB	100
After Ambari Server Setup	N/A	4,000
After Ambari Server Start	N/A	500
After Ambari Agent Start	N/A	200

\*Size and Inode values are approximate

## 1.2.4. Maximum Open Files Requirements

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
```

## 1.3. Collect Information

Before deploying a cluster, you should collect the following information:

- The fully qualified domain name (FQDN) of each host in your system. The Ambari Cluster Install wizard supports using IP addresses. You can use

```
hostname -f
```

to check or verify the FQDN of a host.



### Note

Deploying all components on a single host is possible, but is appropriate only for initial evaluation purposes. Typically, you set up at least three hosts; one master host and two slaves, as a minimum cluster.

- A list of components you want to set up on each host.
- The base directories you want to use as mount points for storing:
  - NameNode data
  - DataNodes data
  - Secondary NameNode data
  - Oozie data
  - YARN data



- ZooKeeper data, if you install ZooKeeper
- Various log, pid, and db files, depending on your install type



### Important

You must use base directories that provide persistent storage locations for your components and your Hadoop data. Installing components in locations that may be removed from a host may result in cluster failure or data loss. For example: Do Not use `/tmp` in a base directory path.

## 1.4. Prepare the Environment

To deploy your Hortonworks stack using Ambari, you need to prepare your deployment environment:

- [Set Up Password-less SSH \[4\]](#)
- [Set Up Service User Accounts \[5\]](#)
- [Enable NTP on the Cluster and on the Browser Host \[5\]](#)
- [Check DNS and NSCD \[6\]](#)
- [Configuring iptables \[7\]](#)
- [Disable SELinux and PackageKit and check the umask Value \[8\]](#)
- [Download and set up database connectors \[9\]](#)
- [Configuring a Database Instance for Ranger \[9\]](#)
- [Install Databases for HDF services \[14\]](#)

### 1.4.1. Set Up Password-less SSH

#### About This Task

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.



#### Note

You can choose to manually install an Ambari Agent on each cluster host. In this case, you do not need to generate and distribute SSH keys.

#### Steps

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.

#### More Information

[Installing Ambari agents manually](#)

## 1.4.2. Set Up Service User Accounts

Each service requires a service user account. The Ambari Cluster Install wizard creates new and preserves any existing service user accounts, and uses these accounts when configuring Hadoop services. Service user account creation applies to service user accounts on the local operating system and to LDAP/AD accounts.

#### More Information

[Understanding service users and groups](#)

## 1.4.3. 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 install the NTP service and ensure it's started on boot, run the following commands on each host:

**RHEL/CentOS/Oracle 7**

```
yum install -y ntp
systemctl enable ntpd
```

**SLES**

```
zypper install ntp
chkconfig ntp on
```

**Ubuntu**

```
apt-get install ntp
update-rc.d ntp defaults
```

**Debian**

```
apt-get install ntp
update-rc.d ntp defaults
```

## 1.4.4. Check DNS and NSCD

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

Hadoop relies heavily on DNS, and as such performs many DNS lookups during normal operation. To reduce the load on your DNS infrastructure, it's highly recommended to use the Name Service Caching Daemon (NSCD) on cluster nodes running Linux. This daemon will cache host, user, and group lookups and provide better resolution performance, and reduced load on DNS infrastructure.

### 1.4.4.1. 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
```

### 1.4.4.2. 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>
```

### 1.4.4.3. 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>
```

### 1.4.5. 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/Amazon Linux

```
systemctl disable firewalld
service firewalld stop
```

#### SLES

```
rcSuSEfirewall2 stop
chkconfig SuSEfirewall2_setup off
```

#### Ubuntu

```
sudo ufw disable
sudo iptables -X
sudo iptables -t nat -F
sudo iptables -t nat -X
sudo iptables -t mangle -F
sudo iptables -t mangle -X
sudo iptables -P INPUT ACCEPT
sudo iptables -P FORWARD ACCEPT
sudo iptables -P OUTPUT ACCEPT
```

#### Debian

```
sudo iptables -X
sudo iptables -t nat -F
sudo iptables -t nat -X
sudo iptables -t mangle -F
sudo iptables -t mangle -X
sudo iptables -P INPUT ACCEPT
sudo iptables -P FORWARD ACCEPT
sudo iptables -P OUTPUT ACCEPT
```

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.

### More Information

[Configuring network port numbers](#)

## 1.4.6. 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, enter:

```
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, and HDF 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
```

Permanently changing the umask for all interactive users:

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

## 1.4.7. Download and set up database connectors

Components like Druid, Hive, Ranger, Oozie, and Superset require an operational database. During installation, you have the option to use an existing database or have Ambari install a new instance, in the case of Hive. For Ambari to connect to the database of your choice, you must download the necessary database drivers and connectors directly from the database vendor *before* installing the component. To better prepare for your install or upgrade, set up the database connectors as you set up your environment.

### More Information

[Using an existing or installing a default database](#)

### 1.4.7.1. Configuring a Database Instance for Ranger

A MySQL, Oracle, PostgreSQL, or Amazon RDS database instance must be running and available to be used by Ranger. The Ranger installation will create two new users (default names: `rangeradmin` and `rangerlogger`) and two new databases (default names: `ranger` and `ranger_audit`).

Choose from the following:

- [Configuring MySQL for Ranger \[9\]](#)
- [Configuring PostgreSQL for Ranger \[10\]](#)
- [Configuring Oracle for Ranger \[12\]](#)

If you are using Amazon RDS, there are additional requirements:

- [Amazon RDS Requirements \[13\]](#)

#### 1.4.7.1.1. Configuring MySQL for Ranger

##### Prerequisites

When using MySQL, the storage engine used for the Ranger admin policy store tables **MUST** support transactions. InnoDB is an example of engine that supports transactions. A storage engine that does not support transactions is not suitable as a policy store.

##### Steps

If you are using Amazon RDS, see the [Amazon RDS Requirements](#).

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/Aamazon 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
```

### **1.4.7.1.2. Configuring PostgreSQL for Ranger**

If you are using Amazon RDS, see the [Amazon RDS Requirements](#).

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

**RHEL/CentOS/Oracle/Aamazon 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-jdbc.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.



```
# TYPE DATABASE USER CIDR-ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all postgres,rangeradmin,rangerlogger trust
# IPv4 local connections:
host all postgres,rangeradmin,rangerlogger 0.0.0.0/0 trust
# IPv6 local connections:
host all postgres,rangeradmin,rangerlogger ::/0 trust
"/var/lib/pgsql/data/pg_hba.conf" 74L, 3445C
```

- 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`.

### 1.4.7.1.3. Configuring Oracle for Ranger

If you are using Amazon RDS, see the [Amazon RDS Requirements](#).

- 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
```

- 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;
```

- 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
```

#### 1.4.7.1.4. Amazon RDS Requirements

Ranger requires a relational database as its policy store. There are additional prerequisites for Amazon RDS-based databases due to how Amazon RDS is set up and managed:

- [MySQL/MariaDB Prerequisite \[13\]](#)
- [PostgreSQL Prerequisite \[13\]](#)
- [Oracle Prerequisite \[14\]](#)

##### 1.4.7.1.4.1. MySQL/MariaDB Prerequisite

You must change the variable `log_bin_trust_function_creators` to 1 during Ranger installation.

From RDS Dashboard>Parameter group (on the left side of the page):

1. Set the MySQL Server variable `log_bin_trust_function_creators` to 1.
2. (Optional) After Ranger installation is complete, reset `log_bin_trust_function_creators` to its original setting. The variable is only required to be set to 1 during Ranger installation.

For more information, see:

- [Stratalux: Why You Should Always Use a Custom DB Parameter Group When Creating an RDS Instance](#)
- [AWS Documentation>Amazon RDS DB Instance Lifecycle » Working with DB Parameter Groups](#)
- [MySQL 5.7 Reference Manual >Binary Logging of Stored Programs](#)

##### 1.4.7.1.4.2. PostgreSQL Prerequisite

The Ranger database user in Amazon RDS PostgreSQL Server should be created before installing Ranger and should be granted an existing role which must have the role `CREATEDB`.

1. Using the master user account, log in to the Amazon RDS PostgreSQL Server from master user account (created during RDS PostgreSQL instance creation) and execute following commands:

```
a. CREATE USER $rangerdbuser WITH LOGIN PASSWORD 'password'
```

b. `GRANT $rangerdbuser to $postgresroot`

Where `$postgresroot` is the RDS PostgreSQL master user account (for example: `postgresroot`) and `$rangerdbuser` is the Ranger database user name (for example: `rangeradmin`).

2. If you are using Ranger KMS, execute the following commands:

a. `CREATE USER $rangerkmsuser WITH LOGIN PASSWORD 'password'`

b. `GRANT $rangerkmsuser to $postgresroot`

Where `$postgresroot` is the RDS PostgreSQL master user account (for example: `postgresroot`) and `$rangerkmsuser` is the Ranger KMS user name (for example: `rangerkms`).

#### 1.4.7.1.4.3. Oracle Prerequisite

Due to [limitations in Amazon RDS](#), the Ranger database user and tablespace must be created manually and the required privileges must be manually granted to the Ranger database user.

1. Log in to the RDS Oracle Server from the master user account (created during RDS Oracle instance creation) and execute following commands:

```
create user $rangerdbuser identified by "password";
GRANT CREATE SESSION,CREATE PROCEDURE,CREATE TABLE,CREATE VIEW,CREATE
SEQUENCE,CREATE PUBLIC SYNONYM,CREATE ANY SYNONYM,CREATE TRIGGER,UNLIMITED
Tablespace TO $rangerdbuser;
create tablespace $rangerdb datafile size 10M autoextend on;
alter user $rangerdbuser DEFAULT Tablespace $rangerdb;
```

Where `$rangerdb` is a actual Ranger database name (for example: `ranger`) and `$rangerdbuser` is Ranger database username (for example: `rangeradmin`).

2. If you are using Ranger KMS, execute the following commands:

```
create user $rangerdbuser identified by "password";
GRANT CREATE SESSION,CREATE PROCEDURE,CREATE TABLE,CREATE VIEW,CREATE
SEQUENCE,CREATE PUBLIC SYNONYM,CREATE ANY SYNONYM,CREATE TRIGGER,UNLIMITED
Tablespace TO $rangerkmsuser;
create tablespace $rangerkmsdb datafile size 10M autoextend on;
alter user $rangerkmsuser DEFAULT Tablespace $rangerkmsdb;
```

Where `$rangerkmsdb` is a actual Ranger database name (for example: `rangerkms`) and `$rangerkmsuser` is Ranger database username (for example: `rangerkms`).

### 1.4.7.2. Install Databases for HDF services

When installing Schema Registry, SAM, Druid, and Superset, you require a relational data store to store metadata. You can use either MySQL, Postgres, Oracle, or MariaDB. These topics describe how to install MySQL, Postgres, and Oracle and how create a databases for SAM and Schema Registry. If you are installing on an existing HDP cluster by using Superset, you can skip the installation instructions, because MySQL was installed with Druid. In this case, configure the databases.



### Note

You should install either Postgres, Oracle or MySQL; both are not necessary. It is recommended that you use MySQL.



### Warning

If you are installing Postgres, you must install Postgres 9.5 or later for SAM and Schema Registry. Ambari does not install Postgres 9.5, so you must perform a manual Postgres installation.

## Installing and Configuring MySQL

- [Installing MySQL \[15\]](#)
- [Configuring SAM and Schema Registry Metadata Stores in MySQL \[16\]](#)
- [Configuring Druid and Superset Metadata Stores in MySQL \[16\]](#)

## Installing and Configuring Postgres

- [Install Postgres \[17\]](#)
- [Configure Postgres to Allow Remote Connections \[18\]](#)
- [Configure SAM and Schema Registry Metadata Stores in Postgres \[18\]](#)
- [Configure Druid and Superset Metadata Stores in Postgres \[19\]](#)

## Using an Oracle Database

- [Section 1.4.7.2.8, “Specifying an Oracle Database to Use with SAM and Schema Registry” \[19\]](#)
- [Section 1.4.7.2.9, “Switching to an Oracle Database After Installation” \[20\]](#)

### 1.4.7.2.1. Installing MySQL

#### About This Task

You can install MySQL 5.5 or later.

#### Before You Begin

On the Ambari host, install the JDBC driver for MySQL, and then add it to Ambari:

```
yum install mysql-connector-java* \  
sudo ambari-server setup --jdbc-db=mysql \  
--jdbc-driver=/usr/share/java/mysql-connector-java.jar
```

#### Steps

1. Log in to the node on which you want to install the MySQL metastore to use for SAM, Schema Registry, and Druid.
2. Install MySQL and the MySQL community server, and start the MySQL service:

```
yum localinstall \
https://dev.mysql.com/get/mysql57-community-release-el7-8.noarch.rpm

yum install mysql-community-server

systemctl start mysqld.service
```

3. Obtain the randomly generated MySQL root password.

```
grep 'A temporary password is generated for root@localhost' \
/var/log/mysqld.log |tail -1
```

4. Reset the MySQL root password. Enter the following command. You are prompted for the password you obtained in the previous step. MySQL then asks you to change the password.

```
/usr/bin/mysql_secure_installation
```

### 1.4.7.2.2. Configuring SAM and Schema Registry Metadata Stores in MySQL

#### Steps

1. Launch the MySQL monitor:

```
mysql -u root -p
```

2. Create the database for Schema Registry and SAM metastore:

```
create database registry;
create database streamline;
```

3. Create Schema Registry and SAM user accounts, replacing the final IDENTIFIED BY string with your password:

```
CREATE USER 'registry'@'%' IDENTIFIED BY 'R12$%34qw';
CREATE USER 'streamline'@'%' IDENTIFIED BY 'R12$%34qw';
```

4. Assign privileges to the user account:

```
GRANT ALL PRIVILEGES ON registry.* TO 'registry'@'%' WITH GRANT OPTION ;
GRANT ALL PRIVILEGES ON streamline.* TO 'streamline'@'%' WITH GRANT OPTION ;
```

5. Commit the operation:

```
commit;
```

### 1.4.7.2.3. Configuring Druid and Superset Metadata Stores in MySQL

#### About This Task

Druid and Superset require a relational data store to store metadata. To use MySQL for this, install MySQL and create a database for the Druid metastore.

#### Steps

1. Launch the MySQL monitor:

```
mysql -u root -p
```

2. Create the database for the Druid and Superset metastore:

```
CREATE DATABASE druid DEFAULT CHARACTER SET utf8;  
CREATE DATABASE superset DEFAULT CHARACTER SET utf8;
```

3. Create druid and superset user accounts, replacing the final IDENTIFIED BY string with your password:

```
CREATE USER 'druid'@'%' IDENTIFIED BY '9oNio)ex1ndL';  
CREATE USER 'superset'@'%' IDENTIFIED BY '9oNio)ex1ndL';
```

4. Assign privileges to the druid account:

```
GRANT ALL PRIVILEGES ON *.* TO 'druid'@'%' WITH GRANT OPTION;  
GRANT ALL PRIVILEGES ON *.* TO 'superset'@'%' WITH GRANT OPTION;
```

5. Commit the operation:

```
commit;
```

#### 1.4.7.2.4. Install Postgres

##### Before You Begin

If you have already installed a MySQL database, you may skip these steps.



##### Warning

You must install Postgres 9.5 or later for SAM and Schema Registry. Ambari does not install Postgres 9.5, so you must perform a manual Postgres installation.

##### Steps

1. Install Red Hat Package Manager (RPM) according to the requirements of your operating system:

```
yum install https://yum.postgresql.org/9.6/redhat/rhel-7-x86_64/pgdg-redhat96-9.6-3.noarch.rpm
```

2. Install Postgres version 9.5 or later:

```
yum install postgresql96-server postgresql96-contrib postgresql96
```

3. Initialize the database:

- For CentOS 7, use the following syntax:

```
/usr/pgsql-9.6/bin/postgresql96-setup initdb
```

- For CentOS 6, use the following syntax:

```
sudo service postgresql initdb
```

4. Start Postgres.

For example, if you are using CentOS 7, use the following syntax:

```
systemctl enable postgresql-9.6.service
systemctl start postgresql-9.6.service
```

5. Verify that you can log in:

```
sudo su postgres
psql
```

### 1.4.7.2.5. Configure Postgres to Allow Remote Connections

#### About This Task

It is critical that you configure Postgres to allow remote connections before you deploy a cluster. If you do not perform these steps in advance of installing your cluster, the installation fails.

#### Steps

1. Open `/var/lib/pgsql/9.6/data/pg_hba.conf` and update to the following

```
# "local" is for Unix domain socket connections only
local all all trust

# IPv4 local connections:
host all all 0.0.0.0/0 trust

# IPv6 local connections:
host all all ::/0 trust
```

2. Open `/var/lib/pgsql/9.6/data/postgresql.conf` and update to the following:

```
listen_addresses = '*'
```

3. Restart Postgres:

```
systemctl stop postgresql-9.6.service
systemctl start postgresql-9.6.service
```

### 1.4.7.2.6. Configure SAM and Schema Registry Metadata Stores in Postgres

#### About This Task

If you have already installed MySQL and configured SAM and Schema Registry metadata stores using MySQL, you do not need to configure additional metadata stores in Postgres.

#### Steps

1. Log in to Postgres:

```
sudo su postgres
psql
```

2. Create a database called `registry` with the password `registry`:

```
create database registry;  
CREATE USER registry WITH PASSWORD 'registry';  
GRANT ALL PRIVILEGES ON DATABASE "registry" to registry;
```

3. Create a database called `streamline` with the password `streamline`:

```
create database streamline;  
CREATE USER streamline WITH PASSWORD 'streamline';  
GRANT ALL PRIVILEGES ON DATABASE "streamline" to streamline;
```

### 1.4.7.2.7. Configure Druid and Superset Metadata Stores in Postgres

#### About This Task

Druid and Superset require a relational data store to store metadata. To use Postgres for this, install Postgres and create a database for the Druid metastore. If you have already created a data store using MySQL, you do not need to configure additional metadata stores in Postgres.

#### Steps

1. Log in to Postgres:

```
sudo su postgres  
psql
```

2. Create a database, user, and password, each called `druid`, and assign database privileges to the user `druid`:

```
create database druid;  
CREATE USER druid WITH PASSWORD 'druid';  
GRANT ALL PRIVILEGES ON DATABASE "druid" to druid;
```

3. Create a database, user, and password, each called `superset`, and assign database privileges to the user `superset`:

```
create database superset;  
CREATE USER superset WITH PASSWORD 'superset';  
GRANT ALL PRIVILEGES ON DATABASE "superset" to superset;
```

### 1.4.7.2.8. Specifying an Oracle Database to Use with SAM and Schema Registry

#### About This Task

You may use an Oracle database with SAM and Schema Registry. Oracle databases 12c and 11g Release 2 are supported

#### Prerequisites

You have an Oracle database installed and configured.

#### Steps

1. Register the Oracle JDBC driver jar.



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

2. From the SAM or Schema Registry configuration screen, select Oracle as the database type and provide the necessary Oracle Server JDBC credentials and connection string.

### 1.4.7.2.9. Switching to an Oracle Database After Installation

#### About This Task

If you want to use an Oracle database with SAM or Schema Registry after you have performed your initial HDF installation or upgrade, you can switch to an Oracle database. Oracle databases 12c and 11g Release 2 are supported.

#### Prerequisites

You have an Oracle database installed and configured.

#### Steps

1. Log into Ambari Server and shut down SAM or Schema Registry.
2. From the configuration screen, select Oracle as the database type and provide Oracle credentials, the JDBC connection string and click **Save**.
3. From the command line where Ambari Server is running, register the Oracle JDBC driver jar:

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

4. From the host where SAM or Schema Registry are installed, copy the JDBC jar to the following location, depending on which component you are updating.

```
cp ojdbc6.jar /usr/hdf/current/registry/bootstrap/lib/.
cp ojdbc6.jar /usr/hdf/current/streamline/bootstrap/lib/.
```

5. From the host where SAM or Schema Registry are installed, run the following command to create the required schemas for SAM or Schema Registry.

```
export JAVA_HOME=/usr/jdk64/jdk1.8.0_112 ; source /usr/hdf/current/streamline/conf/streamline-env.sh ; /usr/hdf/current/streamline/bootstrap/bootstrap-storage.sh create

export JAVA_HOME=/usr/jdk64/jdk1.8.0_112 ; source /usr/hdf/current/registry/conf/registry-env.sh ; /usr/hdf/current/registry/bootstrap/bootstrap-storage.sh create
```



#### Note

You only run this command once, from a single host, to prepare the database.

6. Confirm that new tables are created in the Oracle database.
7. From Ambari, restart SAM or Schema Registry.

8. If you are specifying an Oracle database for SAM, run the following command after you have restarted SAM.

```
export JAVA_HOME=/usr/jdk64/jdk1.8.0_112 ; source /usr/hdf/current/  
streamline/conf/streamline-env.sh ; /usr/hdf/current/streamline/bootstrap/  
bootstrap.sh
```

9. Confirm that Sam or Schema Registry are available and turn off maintenance mode.

## 2. Using a Local Repository

If your enterprise clusters have limited outbound Internet access, you should consider using a local repository, which enables you to benefit from more governance and better installation performance. You can also use a local repository for routine post-installation cluster operations such as service start and restart operations. Using a local repository includes obtaining public repositories, setting up the repository using either no internet access or limited internet access, and preparing the Apache Ambari repository configuration file to use your new local repository.

- [Obtain Public Repositories](#)
- Set up a local repository having:
  - [Setting Up a Local Repository with No Internet Access \[26\]](#)
  - [Setting up a Local Repository with Temporary Internet Access \[23\]](#)
- [Preparing the Ambari Repository Configuration File to Use the Local Repository \[27\]](#)

### 2.1. 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 the tarball to create the repository.

- Temporary Internet Access

This option involves using your temporary Internet access to synchronize (using `reposync`) the software packages to your selected mirror server to create 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:

- [Preparing to Set Up a Local Repository \[22\]](#)
- [Setting Up a Local Repository with No Internet Access \[26\]](#)
- [Setting up a Local Repository with Temporary Internet Access \[23\]](#)

#### 2.1.1. Preparing to Set Up a Local Repository

Before setting up your local repository, you must have met certain requirements.

- Selected an existing server, in or accessible to the cluster, that runs a supported operating system.
- Enabled network access from all hosts in your cluster to the mirror server.

- Ensured that the mirror server has a package manager installed such as yum (for RHEL, CentOS, Oracle, or Amazon Linux), zypper (for SLES), or apt-get (for Debian and Ubuntu).
- **Optional:** If your repository has temporary Internet access, and you are using RHEL, CentOS, Oracle, or Amazon Linux as your OS, installed yum utilities:

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

After meeting these requirements, you can take steps to prepare to set up your local repository.

### Steps

1. Create an HTTP server:
  - a. On the mirror server, install an HTTP server (such as Apache httpd) using the instructions provided on the Apache community website.
  - b. Activate the 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/  
Amazon 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.

### Next Steps

You next must set up your local repository, either with no Internet access or with temporary Internet access.

### More Information

[httpd.apache.org/download.cgi](http://httpd.apache.org/download.cgi)

## 2.1.2. Setting up a Local Repository with Temporary Internet Access

### Prerequisites

You must have completed the [Getting Started Setting up a Local Repository](#) procedure.

--

To finish setting up your local repository, complete the following:

### Steps

1. Install the repository configuration files for Ambari and the Stack on the host.
2. Confirm repository availability;

**For RHEL, CentOS, Oracle or Amazon Linux:** `yum repolist`

**For SLES:** `zypper repos`

**For Debian and Ubuntu:** `dpkg-list`

3. Synchronize the repository contents to your mirror server:

- Browse to the web server directory:

**For RHEL, CentOS, Oracle or Amazon Linux:** `cd /var/www/html`

**For SLES:** `cd /srv/www/htdocs/rpms`

**For Debian and Ubuntu:** `cd /var/www/html`

- For Ambari, create the `ambari` directory and `reposync`:

```
mkdir -p ambari/<OS>
```

```
cd ambari/<OS>
```

```
reposync -r Updates-Ambari-2.7.3.0
```

In this syntax, the value of `<OS>` is `amazonlinux2`, `centos7`, `sles12`, `ubuntu14`, `ubuntu16`, `ubuntu18`, or `debian9`.



### Important

Due to a known issue in version 1.1.31-2 of the Debian 9 `reposync`, we advise using `reposync` version 11.3.1-3 or above when working on a Debian 9 host.

- For Hortonworks Data Platform (HDP) stack repositories, create the `hdp` directory and `reposync`:

```
mkdir -p hdp/<OS>
```

```
cd hdp/<OS>
```

```
reposync -r HDP-<latest.version>
```

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

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

```
mkdir -p hdf/<OS>
```

```
cd hdf/<OS>
```

```
reposync -r HDF-<latest.version>
```

#### 4. Generate the repository metadata:

##### For Ambari:

```
createrepo <web.server.directory>/ambari/  
<OS>/Updates-Ambari-2.7.3.0
```

##### For HDP Stack Repositories:

```
createrepo <web.server.directory>/hdp/<OS>/  
HDP-<latest.version>
```

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

##### For HDF Stack Repositories:

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

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

**Ambari Base URL**      `http://<web.server>/ambari/<OS>/Updates-Ambari-2.7.3.0`

**HDF Base URL**      `http://<web.server>/hdf/<OS>/HDF-<latest.version>`

**HDP Base URL**      `http://<web.server>/hdp/<OS>/HDP-<latest.version>`

**HDP-UTILS Base URL**      `http://<web.server>/hdp/<OS>/HDP-UTILS-<version>`

Where:

- `<web.server>` – The FQDN of the web server host
- `<version>` – The Hortonworks stack version number
- `<OS>` – `centos7`, `sles12`, `ubuntu14`, `ubuntu16`, `ubuntu 18`, or `debian9`



### 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/CentOS/Oracle 7:

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

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

```
[main]
```

```
enabled=1
```

```
gpgcheck=0
```

### More Information

[Obtaining Public Repositories](#)

## 2.1.3. Setting Up a Local Repository with No Internet Access

### Prerequisites

You must have completed the [Getting Started Setting up a Local Repository](#) procedure.

--

To finish setting up your local repository, complete the following:

### Steps

1. Obtain the compressed tape archive file (tarball) for the repository you want to create.
2. Copy the repository tarball to the web server directory and uncompress (untar) the archive:
  - a. Browse to the web server directory you created.

**For RHEL/CentOS/Oracle/  
Amazon Linux:** `cd /var/www/html/`

**For SLES:** `cd /srv/www/htdocs/rpms`

**For Debian/Ubuntu:** `cd /var/www/html/`

- b. Untar the repository tarballs and move the files 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:

**Ambari Repository**                      Untar under <web.server.directory>.

**HDF Stack Repositories**              Create a directory and untar it under <web.server.directory>/hdf.

**HDP Stack Repositories**              Create a directory and untar it under <web.server.directory>/hdp.

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

**Ambari Base URL**                      `http://<web.server>/Ambari-2.7.3.0/<OS>`

HDF Base URL	<code>http://&lt;web.server&gt;/hdf/HDF/&lt;OS&gt;/3.x/updates/&lt;latest.version&gt;</code>
HDP Base URL	<code>http://&lt;web.server&gt;/hdp/HDP/&lt;OS&gt;/3.x/updates/&lt;latest.version&gt;</code>
HDP-UTILS Base URL	<code>http://&lt;web.server&gt;/hdp/HDP-UTILS-&lt;version&gt;/repos/&lt;OS&gt;</code>



### 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.

a. **For RHEL/CentOS/Oracle 7:** `yum install yum-plugin-priorities`

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

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

### More Information

[Obtaining Public Repositories](#)

## 2.2. Preparing the Ambari Repository Configuration File to Use the Local Repository

### Steps

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

```
http://public-repo-1.hortonworks.com/ambari/<OS>/2.x/updates/2.7.3.0/ambari.repo
```

<OS> – centos7, sles12, ubuntu14, ubuntu16, ubuntu 18, or debian9

2. Edit the `ambari.repo` file and replace the Ambari Base URL `baseurl` obtained when setting up your local repository.

```
[Updates-Ambari-2.7.3.0]
name=Ambari-2.7.3.0-Updates
baseurl=INSERT-BASE-URL
gpgcheck=1
gpgkey=http://public-repo-1.hortonworks.com/ambari/centos7/RPM-GPG-KEY/RPM-GPG-KEY-Jenkins
```



```
enabled=1
```

```
priority=1
```



### Note

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

### Base URL for a Local Repository

**Built with Repository Tarball** (No Internet Access) `http://<web.server>/Ambari-2.7.3.0/<OS>`

**Built with Repository File** (Temporary Internet Access) `http://<web.server>/ambari/<OS>/Updates-Ambari-2.7.3.0`

where `<web.server>` = FQDN of the web server host, and `<OS>` is `amazonlinux2`, `centos7`, `sles12`, `ubuntu14`, `ubuntu16`, `ubuntu18`, or `debian9`.

- Place the `ambari.repo` file on the host you plan to use for the Ambari server:

**For RHEL/CentOS/Oracle/Amazon Linux:** `/etc/yum.repos.d/ambari.repo`

**For SLES:** `/etc/zypp/repos.d/ambari.repo`

**For Debian/Ubuntu:** `/etc/apt/sources.list.d/ambari.list`

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

```
[main]
```

```
enabled=1
```

```
gpgcheck=0
```

### Next Steps

Proceed to [Installing Ambari](#) to install and setup Ambari Server.

### More Information

[Setting Up a Local Repository with No Internet Access](#)

[Setting Up a Local Repository with Temporary Internet Access](#)

## 3. Obtaining Public Repositories

These sections describe how to obtain:

- [Ambari Repositories \[29\]](#)
- [HDP Stack Repositories \[30\]](#)

### 3.1. Ambari Repositories

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.

#### Ambari 2.7.3 Repositories

OS	Format	URL
RedHat 7	Base URL	<a href="http://public-repo-1.hortonworks.com/ambari/centos7/2.x/updates/2.7.3.0">http://public-repo-1.hortonworks.com/ambari/centos7/2.x/updates/2.7.3.0</a>
CentOS 7	Repo File	<a href="http://public-repo-1.hortonworks.com/ambari/centos7/2.x/updates/2.7.3.0/ambari.repo">http://public-repo-1.hortonworks.com/ambari/centos7/2.x/updates/2.7.3.0/ambari.repo</a>
	Tarball <a href="#">md5</a>   <a href="#">asc</a>	<a href="http://public-repo-1.hortonworks.com/ambari/centos7/2.x/updates/2.7.3.0/ambari-2.7.3.0-centos7.tar.gz">http://public-repo-1.hortonworks.com/ambari/centos7/2.x/updates/2.7.3.0/ambari-2.7.3.0-centos7.tar.gz</a>
amazonlinux 2	Base URL	<a href="http://public-repo-1.hortonworks.com/ambari/amazonlinux2/2.x/updates/2.7.3.0">http://public-repo-1.hortonworks.com/ambari/amazonlinux2/2.x/updates/2.7.3.0</a>
	Repo File	<a href="http://public-repo-1.hortonworks.com/ambari/amazonlinux2/2.x/updates/2.7.3.0/ambari.repo">http://public-repo-1.hortonworks.com/ambari/amazonlinux2/2.x/updates/2.7.3.0/ambari.repo</a>
	Tarball <a href="#">md5</a>   <a href="#">asc</a>	<a href="http://public-repo-1.hortonworks.com/ambari/amazonlinux2/2.x/updates/2.7.3.0/ambari-2.7.3.0-amazonlinux2.tar.gz">http://public-repo-1.hortonworks.com/ambari/amazonlinux2/2.x/updates/2.7.3.0/ambari-2.7.3.0-amazonlinux2.tar.gz</a>
SLES 12	Base URL	<a href="http://public-repo-1.hortonworks.com/ambari/sles12/2.x/updates/2.7.3.0">http://public-repo-1.hortonworks.com/ambari/sles12/2.x/updates/2.7.3.0</a>
	Repo File	<a href="http://public-repo-1.hortonworks.com/ambari/sles12/2.x/updates/2.7.3.0/ambari.repo">http://public-repo-1.hortonworks.com/ambari/sles12/2.x/updates/2.7.3.0/ambari.repo</a>
	Tarball <a href="#">md5</a>   <a href="#">asc</a>	<a href="http://public-repo-1.hortonworks.com/ambari/sles12/2.x/updates/2.7.3.0/ambari-2.7.3.0-sles12.tar.gz">http://public-repo-1.hortonworks.com/ambari/sles12/2.x/updates/2.7.3.0/ambari-2.7.3.0-sles12.tar.gz</a>
Ubuntu 14	Base URL	<a href="http://public-repo-1.hortonworks.com/ambari/ubuntu14/2.x/updates/2.7.3.0">http://public-repo-1.hortonworks.com/ambari/ubuntu14/2.x/updates/2.7.3.0</a>
	Repo File	<a href="http://public-repo-1.hortonworks.com/ambari/ubuntu14/2.x/updates/2.7.3.0/ambari.list">http://public-repo-1.hortonworks.com/ambari/ubuntu14/2.x/updates/2.7.3.0/ambari.list</a>
	Tarball <a href="#">md5</a>   <a href="#">asc</a>	<a href="http://public-repo-1.hortonworks.com/ambari/ubuntu14/2.x/updates/2.7.3.0/ambari-2.7.3.0-ubuntu14.tar.gz">http://public-repo-1.hortonworks.com/ambari/ubuntu14/2.x/updates/2.7.3.0/ambari-2.7.3.0-ubuntu14.tar.gz</a>
Ubuntu 16	Base URL	<a href="http://public-repo-1.hortonworks.com/ambari/ubuntu16/2.x/updates/2.7.3.0">http://public-repo-1.hortonworks.com/ambari/ubuntu16/2.x/updates/2.7.3.0</a>
	Repo File	<a href="http://public-repo-1.hortonworks.com/ambari/ubuntu16/2.x/updates/2.7.3.0/ambari.list">http://public-repo-1.hortonworks.com/ambari/ubuntu16/2.x/updates/2.7.3.0/ambari.list</a>
	Tarball <a href="#">md5</a>   <a href="#">asc</a>	<a href="http://public-repo-1.hortonworks.com/ambari/ubuntu16/2.x/updates/2.7.3.0/ambari-2.7.3.0-ubuntu16.tar.gz">http://public-repo-1.hortonworks.com/ambari/ubuntu16/2.x/updates/2.7.3.0/ambari-2.7.3.0-ubuntu16.tar.gz</a>
Ubuntu 18	Base URL	<a href="http://public-repo-1.hortonworks.com/ambari/ubuntu18/2.x/updates/2.7.3.0">http://public-repo-1.hortonworks.com/ambari/ubuntu18/2.x/updates/2.7.3.0</a>
	Repo File	<a href="http://public-repo-1.hortonworks.com/ambari/ubuntu18/2.x/updates/2.7.3.0/ambari.list">http://public-repo-1.hortonworks.com/ambari/ubuntu18/2.x/updates/2.7.3.0/ambari.list</a>
	Tarball <a href="#">md5</a>   <a href="#">asc</a>	<a href="http://public-repo-1.hortonworks.com/ambari/ubuntu18/2.x/updates/2.7.3.0/ambari-2.7.3.0-ubuntu18.tar.gz">http://public-repo-1.hortonworks.com/ambari/ubuntu18/2.x/updates/2.7.3.0/ambari-2.7.3.0-ubuntu18.tar.gz</a>
Debian 9	Base URL	<a href="http://public-repo-1.hortonworks.com/ambari/debian9/2.x/updates/2.7.3.0">http://public-repo-1.hortonworks.com/ambari/debian9/2.x/updates/2.7.3.0</a>
	Repo File	<a href="http://public-repo-1.hortonworks.com/ambari/debian9/2.x/updates/2.7.3.0/ambari.list">http://public-repo-1.hortonworks.com/ambari/debian9/2.x/updates/2.7.3.0/ambari.list</a>
	Tarball <a href="#">md5</a>   <a href="#">asc</a>	<a href="http://public-repo-1.hortonworks.com/ambari/debian9/2.x/updates/2.7.3.0/ambari-2.7.3.0-debian9.tar.gz">http://public-repo-1.hortonworks.com/ambari/debian9/2.x/updates/2.7.3.0/ambari-2.7.3.0-debian9.tar.gz</a>

## 3.2. HDP Stack Repositories

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.

- [HDP 3.1.0 Repositories \[30\]](#)

### 3.2.1. HDP 3.1.0 Repositories

OS	Version Number	Repository Name	Format	URL
RedHat 7 CentOS 7 Oracle Linux 7	HDP-3.1.0.0	HDP	Version Definition File (VDF)	<a href="http://public-repo-1.hortonworks.com/HDP/centos7/3.x/updates/3.1.0.0/HDP-3.1.0.0-78.xml">http://public-repo-1.hortonworks.com/HDP/centos7/3.x/updates/3.1.0.0/HDP-3.1.0.0-78.xml</a>
			Base URL	<a href="http://public-repo-1.hortonworks.com/HDP/centos7/3.x/updates/3.1.0.0">http://public-repo-1.hortonworks.com/HDP/centos7/3.x/updates/3.1.0.0</a>
			Repo File	<a href="http://public-repo-1.hortonworks.com/HDP/centos7/3.x/updates/3.1.0.0/hdp.repo">http://public-repo-1.hortonworks.com/HDP/centos7/3.x/updates/3.1.0.0/hdp.repo</a>
			Tarball <a href="#">md5</a>   <a href="#">asc</a>	<a href="http://public-repo-1.hortonworks.com/HDP/centos7/3.x/updates/3.1.0.0/HDP-3.1.0.0-centos7-rpm.tar.gz">http://public-repo-1.hortonworks.com/HDP/centos7/3.x/updates/3.1.0.0/HDP-3.1.0.0-centos7-rpm.tar.gz</a>
		HDP-UTILS	Base URL	<a href="http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/centos7">http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/centos7</a>
			Tarball <a href="#">md5</a>   <a href="#">asc</a>	<a href="http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/centos7/HDP-UTILS-1.1.0.22-centos7.tar.gz">http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/centos7/HDP-UTILS-1.1.0.22-centos7.tar.gz</a>
		HDP-GPL	URL	<a href="http://public-repo-1.hortonworks.com/HDP-GPL/centos7/3.x/updates/3.1.0.0/hdp.gpl.repo">http://public-repo-1.hortonworks.com/HDP-GPL/centos7/3.x/updates/3.1.0.0/hdp.gpl.repo</a>
			Tarball <a href="#">md5</a>   <a href="#">asc</a>	<a href="http://public-repo-1.hortonworks.com/HDP-GPL/centos7/3.x/updates/3.1.0.0/HDP-GPL-3.1.0.0-centos7-gpl.tar.gz">http://public-repo-1.hortonworks.com/HDP-GPL/centos7/3.x/updates/3.1.0.0/HDP-GPL-3.1.0.0-centos7-gpl.tar.gz</a>
amazonlinux2	HDP-3.1.0.0	HDP	Version Definition File (VDF)	<a href="http://public-repo-1.hortonworks.com/HDP/amazonlinux2/3.x/updates/3.1.0.0/HDP-3.1.0.0-1634.xml">http://public-repo-1.hortonworks.com/HDP/amazonlinux2/3.x/updates/3.1.0.0/HDP-3.1.0.0-1634.xml</a>
			Base URL	<a href="http://public-repo-1.hortonworks.com/HDP/amazonlinux2/3.x/updates/3.1.0.0">http://public-repo-1.hortonworks.com/HDP/amazonlinux2/3.x/updates/3.1.0.0</a>
			Repo File	<a href="http://public-repo-1.hortonworks.com/HDP/amazonlinux2/3.x/updates/3.1.0.0/hdp.repo">http://public-repo-1.hortonworks.com/HDP/amazonlinux2/3.x/updates/3.1.0.0/hdp.repo</a>
			Tarball <a href="#">md5</a>   <a href="#">asc</a>	<a href="http://public-repo-1.hortonworks.com/HDP/amazonlinux2/3.x/updates/3.1.0.0/HDP-3.1.0.0-amazonlinux2-rpm.tar.gz">http://public-repo-1.hortonworks.com/HDP/amazonlinux2/3.x/updates/3.1.0.0/HDP-3.1.0.0-amazonlinux2-rpm.tar.gz</a>
		HDP-UTILS	Base URL	<a href="http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/amazonlinux2">http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/amazonlinux2</a>
			Tarball <a href="#">md5</a>   <a href="#">asc</a>	<a href="http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/amazonlinux2/HDP-UTILS-1.1.0.22-amazonlinux2.tar.gz">http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/amazonlinux2/HDP-UTILS-1.1.0.22-amazonlinux2.tar.gz</a>
		HDP-GPL	URL	<a href="http://public-repo-1.hortonworks.com/HDP-GPL/amazonlinux2/3.x/updates/3.1.0.0/hdp.gpl.repo">http://public-repo-1.hortonworks.com/HDP-GPL/amazonlinux2/3.x/updates/3.1.0.0/hdp.gpl.repo</a>
			Tarball <a href="#">md5</a>   <a href="#">asc</a>	<a href="http://public-repo-1.hortonworks.com/HDP-GPL/amazonlinux2/3.x/updates/3.1.0.0/HDP-GPL-3.1.0.0-amazonlinux2-gpl.tar.gz">http://public-repo-1.hortonworks.com/HDP-GPL/amazonlinux2/3.x/updates/3.1.0.0/HDP-GPL-3.1.0.0-amazonlinux2-gpl.tar.gz</a>
SLES 12	HDP-3.1.0.0	HDP	Version Definition File (VDF)	<a href="http://public-repo-1.hortonworks.com/HDP/sles12/3.x/updates/3.1.0.0/HDP-3.1.0.0-78.xml">http://public-repo-1.hortonworks.com/HDP/sles12/3.x/updates/3.1.0.0/HDP-3.1.0.0-78.xml</a>
			Base URL	<a href="http://public-repo-1.hortonworks.com/HDP/sles12/3.x/updates/3.1.0.0/">http://public-repo-1.hortonworks.com/HDP/sles12/3.x/updates/3.1.0.0/</a>

			Repo File	<a href="http://public-repo-1.hortonworks.com/HDP/sles12/3.x/updates/3.1.0.0/hdp.repo">http://public-repo-1.hortonworks.com/HDP/sles12/3.x/updates/3.1.0.0/hdp.repo</a>
			Tarball <a href="#">md5</a>   <a href="#">asc</a>	<a href="http://public-repo-1.hortonworks.com/HDP/sles12/3.x/updates/3.1.0.0/HDP-3.1.0.0-sles12-rpm.tar.gz">http://public-repo-1.hortonworks.com/HDP/sles12/3.x/updates/3.1.0.0/HDP-3.1.0.0-sles12-rpm.tar.gz</a>
		HDP-UTILS	Base URL	<a href="http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/sles12">http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/sles12</a>
			Tarball <a href="#">md5</a>   <a href="#">asc</a>	<a href="http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/sles12/HDP-UTILS-1.1.0.22-sles12.tar.gz">http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/sles12/HDP-UTILS-1.1.0.22-sles12.tar.gz</a>
		HDP-GPL	URL	<a href="http://public-repo-1.hortonworks.com/HDP-GPL/sles12/3.x/updates/3.1.0.0/hdp.gpl.repo">http://public-repo-1.hortonworks.com/HDP-GPL/sles12/3.x/updates/3.1.0.0/hdp.gpl.repo</a>
			Tarball <a href="#">md5</a>   <a href="#">asc</a>	<a href="http://public-repo-1.hortonworks.com/HDP-GPL/sles12/3.x/updates/3.1.0.0/HDP-GPL-3.1.0.0-sles12-gpl.tar.gz">http://public-repo-1.hortonworks.com/HDP-GPL/sles12/3.x/updates/3.1.0.0/HDP-GPL-3.1.0.0-sles12-gpl.tar.gz</a>
Ubuntu 14	HDP-3.1.0.0	HDP	Version Definition File (VDF)	<a href="http://public-repo-1.hortonworks.com/HDP/ubuntu14/3.x/updates/3.1.0.0/HDP-3.1.0.0-78.xml">http://public-repo-1.hortonworks.com/HDP/ubuntu14/3.x/updates/3.1.0.0/HDP-3.1.0.0-78.xml</a>
			Base URL	<a href="http://public-repo-1.hortonworks.com/HDP/ubuntu14/3.x/updates/3.1.0.0/">http://public-repo-1.hortonworks.com/HDP/ubuntu14/3.x/updates/3.1.0.0/</a>
			Repo File	<a href="http://public-repo-1.hortonworks.com/HDP/ubuntu14/3.x/updates/3.1.0.0/hdp.list">http://public-repo-1.hortonworks.com/HDP/ubuntu14/3.x/updates/3.1.0.0/hdp.list</a>
			Tarball <a href="#">md5</a>   <a href="#">asc</a>	<a href="http://public-repo-1.hortonworks.com/HDP/ubuntu14/3.x/updates/3.1.0.0/HDP-3.1.0.0-ubuntu14-deb.tar.gz">http://public-repo-1.hortonworks.com/HDP/ubuntu14/3.x/updates/3.1.0.0/HDP-3.1.0.0-ubuntu14-deb.tar.gz</a>
		HDP-UTILS	Base URL	<a href="http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/ubuntu14">http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/ubuntu14</a>
			Tarball <a href="#">md5</a>   <a href="#">asc</a>	<a href="http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/ubuntu14/HDP-UTILS-1.1.0.22-ubuntu14.tar.gz">http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/ubuntu14/HDP-UTILS-1.1.0.22-ubuntu14.tar.gz</a>
		HDP-GPL	URL	<a href="http://public-repo-1.hortonworks.com/HDP-GPL/ubuntu14/3.x/updates/3.1.0.0/hdp.gpl.list">http://public-repo-1.hortonworks.com/HDP-GPL/ubuntu14/3.x/updates/3.1.0.0/hdp.gpl.list</a>
			Tarball <a href="#">md5</a>   <a href="#">asc</a>	<a href="http://public-repo-1.hortonworks.com/HDP-GPL/ubuntu14/3.x/updates/3.1.0.0/HDP-GPL-3.1.0.0-ubuntu14-gpl.tar.gz">http://public-repo-1.hortonworks.com/HDP-GPL/ubuntu14/3.x/updates/3.1.0.0/HDP-GPL-3.1.0.0-ubuntu14-gpl.tar.gz</a>
Ubuntu 16	HDP-3.1.0.0	HDP	Version Definition File (VDF)	<a href="http://public-repo-1.hortonworks.com/HDP/ubuntu16/3.x/updates/3.1.0.0/HDP-3.1.0.0-78.xml">http://public-repo-1.hortonworks.com/HDP/ubuntu16/3.x/updates/3.1.0.0/HDP-3.1.0.0-78.xml</a>
			Base URL	<a href="http://public-repo-1.hortonworks.com/HDP/ubuntu16/3.x/updates/3.1.0.0/">http://public-repo-1.hortonworks.com/HDP/ubuntu16/3.x/updates/3.1.0.0/</a>
			Repo File	<a href="http://public-repo-1.hortonworks.com/HDP/ubuntu16/3.x/updates/3.1.0.0/hdp.list">http://public-repo-1.hortonworks.com/HDP/ubuntu16/3.x/updates/3.1.0.0/hdp.list</a>
			Tarball <a href="#">md5</a>   <a href="#">asc</a>	<a href="http://public-repo-1.hortonworks.com/HDP/ubuntu16/3.x/updates/3.1.0.0/HDP-3.1.0.0-ubuntu16-deb.tar.gz">http://public-repo-1.hortonworks.com/HDP/ubuntu16/3.x/updates/3.1.0.0/HDP-3.1.0.0-ubuntu16-deb.tar.gz</a>
		HDP-UTILS	Base URL	<a href="http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/ubuntu16">http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/ubuntu16</a>
			Tarball <a href="#">md5</a>   <a href="#">asc</a>	<a href="http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/ubuntu16/HDP-UTILS-1.1.0.22-ubuntu16.tar.gz">http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/ubuntu16/HDP-UTILS-1.1.0.22-ubuntu16.tar.gz</a>
		HDP-GPL	URL	<a href="http://public-repo-1.hortonworks.com/HDP-GPL/ubuntu16/3.x/updates/3.1.0.0/hdp.gpl.list">http://public-repo-1.hortonworks.com/HDP-GPL/ubuntu16/3.x/updates/3.1.0.0/hdp.gpl.list</a>
			Tarball <a href="#">md5</a>   <a href="#">asc</a>	<a href="http://public-repo-1.hortonworks.com/HDP-GPL/ubuntu16/3.x/updates/3.1.0.0/HDP-GPL-3.1.0.0-ubuntu16-gpl.tar.gz">http://public-repo-1.hortonworks.com/HDP-GPL/ubuntu16/3.x/updates/3.1.0.0/HDP-GPL-3.1.0.0-ubuntu16-gpl.tar.gz</a>
Ubuntu 18	HDP-3.1.0.0	HDP	Version Definition File (VDF)	<a href="http://public-repo-1.hortonworks.com/HDP/ubuntu18/3.x/updates/3.1.0.0/HDP-3.1.0.0-78.xml">http://public-repo-1.hortonworks.com/HDP/ubuntu18/3.x/updates/3.1.0.0/HDP-3.1.0.0-78.xml</a>
			Base URL	<a href="http://public-repo-1.hortonworks.com/HDP/ubuntu18/3.x/updates/3.1.0.0/">http://public-repo-1.hortonworks.com/HDP/ubuntu18/3.x/updates/3.1.0.0/</a>
			Repo File	<a href="http://public-repo-1.hortonworks.com/HDP/ubuntu18/3.x/updates/3.1.0.0/hdp.list">http://public-repo-1.hortonworks.com/HDP/ubuntu18/3.x/updates/3.1.0.0/hdp.list</a>
			Tarball <a href="#">md5</a>   <a href="#">asc</a>	<a href="http://public-repo-1.hortonworks.com/HDP/ubuntu18/3.x/updates/3.1.0.0/HDP-3.1.0.0-ubuntu18-deb.tar.gz">http://public-repo-1.hortonworks.com/HDP/ubuntu18/3.x/updates/3.1.0.0/HDP-3.1.0.0-ubuntu18-deb.tar.gz</a>

		HDP-UTILS	Base URL	<a href="http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/ubuntu18">http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/ubuntu18</a>
			Tarball <a href="#">md5</a>   <a href="#">asc</a>	<a href="http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/ubuntu18/HDP-UTILS-1.1.0.22-ubuntu18.tar.gz">http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/ubuntu18/HDP-UTILS-1.1.0.22-ubuntu18.tar.gz</a>
		HDP-GPL	URL	<a href="http://public-repo-1.hortonworks.com/HDP-GPL/ubuntu18/3.x/updates/3.1.0.0/hdp.gpl.list">http://public-repo-1.hortonworks.com/HDP-GPL/ubuntu18/3.x/updates/3.1.0.0/hdp.gpl.list</a>
			Tarball <a href="#">md5</a>   <a href="#">asc</a>	<a href="http://public-repo-1.hortonworks.com/HDP-GPL/ubuntu18/3.x/updates/3.1.0.0/HDP-GPL-3.1.0.0-ubuntu18-gpl.tar.gz">http://public-repo-1.hortonworks.com/HDP-GPL/ubuntu18/3.x/updates/3.1.0.0/HDP-GPL-3.1.0.0-ubuntu18-gpl.tar.gz</a>
Debian9	HDP-3.1.0.0	HDP	Version Definition File (VDF)	<a href="http://public-repo-1.hortonworks.com/HDP/debian9/3.x/updates/3.1.0.0/HDP-3.1.0.0-78.xml">http://public-repo-1.hortonworks.com/HDP/debian9/3.x/updates/3.1.0.0/HDP-3.1.0.0-78.xml</a>
			Base URL	<a href="http://public-repo-1.hortonworks.com/HDP/debian9/3.x/updates/3.1.0.0/">http://public-repo-1.hortonworks.com/HDP/debian9/3.x/updates/3.1.0.0/</a>
			Repo File	<a href="http://public-repo-1.hortonworks.com/HDP/debian9/3.x/updates/3.1.0.0/hdp.list">http://public-repo-1.hortonworks.com/HDP/debian9/3.x/updates/3.1.0.0/hdp.list</a>
			Tarball <a href="#">md5</a>   <a href="#">asc</a>	<a href="http://public-repo-1.hortonworks.com/HDP/debian9/3.x/updates/3.1.0.0/HDP-3.1.0.0-debian9-deb.tar.gz">http://public-repo-1.hortonworks.com/HDP/debian9/3.x/updates/3.1.0.0/HDP-3.1.0.0-debian9-deb.tar.gz</a>
		HDP-UTILS	Base URL	<a href="http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/debian9">http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/debian9</a>
			Tarball <a href="#">md5</a>   <a href="#">asc</a>	<a href="http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/debian9/HDP-UTILS-1.1.0.22-debian9.tar.gz">http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/debian9/HDP-UTILS-1.1.0.22-debian9.tar.gz</a>
		HDP-GPL	URL	<a href="http://public-repo-1.hortonworks.com/HDP-GPL/debian9/3.x/updates/3.1.0.0/hdp.gpl.list">http://public-repo-1.hortonworks.com/HDP-GPL/debian9/3.x/updates/3.1.0.0/hdp.gpl.list</a>
			Tarball <a href="#">md5</a>   <a href="#">asc</a>	<a href="http://public-repo-1.hortonworks.com/HDP-GPL/debian9/3.x/updates/3.1.0.0/HDP-GPL-3.1.0.0-debian9-gpl.tar.gz">http://public-repo-1.hortonworks.com/HDP-GPL/debian9/3.x/updates/3.1.0.0/HDP-GPL-3.1.0.0-debian9-gpl.tar.gz</a>

## 4. Installing Ambari

To install Ambari server on a single host in your cluster, complete the following steps:

1. [Download the Ambari Repository \[33\]](#)
2. [Install the Ambari Server \[40\]](#)
3. [Set Up the Ambari Server \[45\]](#)

### 4.1. Download the Ambari Repository

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

- [RHEL/CentOS/Oracle Linux 7 \[33\]](#)
- [Amazon Linux 2 \[34\]](#)
- [SLES 12 \[35\]](#)
- [Ubuntu 14 \[36\]](#)
- [Ubuntu 16 \[37\]](#)
- [Debian 9 \[39\]](#)

Use a command line editor to perform each instruction.

#### 4.1.1. 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/2.7.3.0/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.

```

repo id                repo name
status
ambari-2.7.3.0-139    ambari Version - ambari-2.7.3.0-139    12
epel/x86_64           Extra Packages for Enterprise Linux 7 - x86_64
11,387
ol7_UEKR4/x86_64      Latest Unbreakable Enterprise Kernel Release 4
for Oracle Linux 7Server (x86_64)    295
ol7_latest/x86_64     Oracle Linux 7Server Latest (x86_64)
18,642
puppetlabs-deps/x86_64 Puppet Labs Dependencies El 7 - x86_64
17
puppetlabs-products/x86_64 Puppet Labs Products El 7 - x86_64
225
repolist: 30,578

```

Version values vary, depending on the installation.



### Note

When deploying a cluster having limited or no Internet access, you should provide access to the bits using an alternative method.

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.

### Next Step

- [Install the Ambari Server \[40\]](#)
- [Set Up the Ambari Server \[45\]](#)

### More Information

[Using a Local Repository](#)

## 4.1.2. Amazon Linux 2

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/amazonlinux2/2.x/updates/2.7.3.0/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.

```
repo id                repo name
status
ambari-2.7.3.0-139    ambari Version - ambari-2.7.3.0-139      12
epel/x86_64           Extra Packages for Enterprise Linux 7 - x86_64
11,387
ol7_UEKR4/x86_64      Latest Unbreakable Enterprise Kernel Release 4
for Amazon Linux 2Server (x86_64)      295
ol7_latest/x86_64     Amazon Linux 2Server Latest (x86_64)
18,642
puppetlabs-deps/x86_64 Puppet Labs Dependencies El 7 - x86_64
17
puppetlabs-products/x86_64 Puppet Labs Products El 7 - x86_64
225
repolist: 30,578
```

Version values vary, depending on the installation.



### Note

When deploying a cluster having limited or no Internet access, you should provide access to the bits using an alternative method.

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.

### Next Step

- [Install the Ambari Server \[40\]](#)
- [Set Up the Ambari Server \[45\]](#)

### More Information

[Using a Local Repository](#)

## 4.1.3. 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/2.7.3.0/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.

#	Alias	Name	Enabled	Refresh
1	ambari-2.7.3.0-139	ambari Version - ambari-2.7.3.0-139	Yes	No
2	http-demeter.uni-regensburg.de-c997c8f9	SUSE-Linux-Enterprise-Software -Development-Kit-12-SP1 12.1.1-1.57	Yes	Yes
3	opensuse	OpenSuse	Yes	Yes

Version values vary, depending on the installation.



## Note

When deploying a cluster having limited or no Internet access, you should provide access to the bits using an alternative method.

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.

### Next Step

- [Install the Ambari Server \[40\]](#)
- [Set Up the Ambari Server \[45\]](#)

### More Information

[Using a Local Repository](#)

## 4.1.4. 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 -O /etc/apt/sources.list.d/ambari.list http://public-repo-1.hortonworks.com/ambari/ubuntu14/2.x/updates/2.7.3.0/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.



### Note

When deploying a cluster having limited or no Internet access, you should provide access to the bits using an alternative method.

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.

### Next Step

- [Install the Ambari Server \[40\]](#)
- [Set Up the Ambari Server \[45\]](#)

### More Information

[Using a Local Repository](#)

## 4.1.5. Ubuntu 16

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 -O /etc/apt/sources.list.d/ambari.list http://public-repo-1.
hortonworks.com/ambari/ubuntu16/2.x/updates/2.7.3.0/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.



### Note

When deploying a cluster having limited or no Internet access, you should provide access to the bits using an alternative method.

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.

### Next Step

- [Install the Ambari Server \[40\]](#)
- [Set Up the Ambari Server \[45\]](#)

### More Information

[Using a Local Repository](#)

## 4.1.6. Ubuntu 18

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 -O /etc/apt/sources.list.d/ambari.list http://public-repo-1.
hortonworks.com/ambari/ubuntu18/2.x/updates/2.7.3.0/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.



### Note

When deploying a cluster having limited or no Internet access, you should provide access to the bits using an alternative method.

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.

### Next Step

- [Install the Ambari Server \[40\]](#)
- [Set Up the Ambari Server \[45\]](#)

### More Information

[Using a Local Repository](#)

## 4.1.7. Debian 9

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 -O /etc/apt/sources.list.d/ambari.list http://public-repo-1.
hortonworks.com/ambari/debian9/2.x/updates/2.7.3.0/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.



### Note

When deploying a cluster having limited or no Internet access, you should provide access to the bits using an alternative method.

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.

#### Next Step

- [Install the Ambari Server \[40\]](#)
- [Set Up the Ambari Server \[45\]](#)

#### More Information

[Using a Local Repository](#)

## 4.2. Install the Ambari Server

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

- [RHEL/CentOS/Oracle Linux 7 \[41\]](#)
- [SLES 12 \[42\]](#)
- [Ubuntu 14 \[43\]](#)
- [Ubuntu 16 \[43\]](#)

- [Ubuntu 18 \[44\]](#)
- [Debian 9 \[44\]](#)

Use a command line editor to perform each instruction.

## 4.2.1. RHEL/CentOS/Oracle Linux 7

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

### Steps

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

```
yum install ambari-server
```

2. Enter **y** when prompted to confirm transaction and dependency checks.

A successful installation displays output similar to the following:

```
Installing : postgresql-libs-9.2.18-1.el7.x86_64      1/4
Installing : postgresql-9.2.18-1.el7.x86_64        2/4
Installing : postgresql-server-9.2.18-1.el7.x86_64  3/4
Installing : ambari-server-2.7.3.0-139.x86_64      4/4
Verifying  : ambari-server-2.7.3.0-139.x86_64     1/4
Verifying  : postgresql-9.2.18-1.el7.x86_64       2/4
Verifying  : postgresql-server-9.2.18-1.el7.x86_64 3/4
Verifying  : postgresql-libs-9.2.18-1.el7.x86_64  4/4

Installed:
  ambari-server.x86_64 0:2.7.3.0-139
Dependency Installed:
  postgresql.x86_64 0:9.2.18-1.el7
  postgresql-libs.x86_64 0:9.2.18-1.el7
  postgresql-server.x86_64 0:9.2.18-1.el7
Complete!
```



### 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/centos7/RPM-
GPG-KEY/RPM-GPG-KEY-Jenkins
```



### Note

When deploying a cluster having limited or no Internet access, you should provide access to the bits using an alternative method.

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.

### Next Step

[Set Up the Ambari Server \[45\]](#)

### More Information

[Using a Local Repository](#)

## 4.2.2. SLES 12

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

### Steps

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

```
zypper install ambari-server
```

2. Enter `y` when prompted 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-2.7.3.0-139.noarch (4/4), 99.0 MiB (126.3 MiB unpacked)
Retrieving: ambari-server-2.7.3.0-139.noarch.rpm [done (3.0 MiB/s)]
Installing: ambari-server-2.7.3.0-139 [done]
ambari-server 0:off 1:off 2:off 3:on 4:off 5:on 6:off
```



### Note

When deploying a cluster having limited or no Internet access, you should provide access to the bits using an alternative method.

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.

**Next Step**[Set Up the Ambari Server \[45\]](#)**More Information**[Using a Local Repository](#)

## 4.2.3. Ubuntu 14

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

**Steps**

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

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

**Note**

When deploying a cluster having limited or no Internet access, you should provide access to the bits using an alternative method.

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.

**Next Step**[Set Up the Ambari Server \[45\]](#)**More Information**[Using a Local Repository](#)

## 4.2.4. Ubuntu 16

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

**Steps**

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

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

**Note**

When deploying a cluster having limited or no Internet access, you should provide access to the bits using an alternative method.



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.

**Next Step**

[Set Up the Ambari Server \[45\]](#)

**More Information**

[Using a Local Repository](#)

## 4.2.5. Ubuntu 18

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

**Steps**

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

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

**Note**

When deploying a cluster having limited or no Internet access, you should provide access to the bits using an alternative method.

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.

**Next Step**

[Set Up the Ambari Server \[45\]](#)

**More Information**

[Using a Local Repository](#)

## 4.2.6. Debian 9

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

**Steps**

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

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



### Note

When deploying a cluster having limited or no Internet access, you should provide access to the bits using an alternative method.

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.

### Next Step

[Set Up the Ambari Server \[45\]](#)

### More Information

[Using a Local Repository](#)

## 4.3. 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.

By default, Ambari Server setup downloads and installs Oracle JDK 1.8 and the accompanying Java Cryptography Extension (JCE) Policy Files.

5. To proceed with the default installation, 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.

Alternatively, you can enter 2 to download a Custom JDK. If you choose Custom JDK, you must manually install the JDK on all hosts and specify the Java Home path.



### Note

To install OpenJDK, use the Custom option. Be prepared to provide the valid `JAVA_HOME` value to Ambari. We strongly recommend that you install the JDK packages consistently on all hosts.

6. Review the GPL license agreement when prompted. To explicitly enable Ambari to download and install LZO data compression libraries, you must answer `y`. If you enter `n`, Ambari will not automatically install LZO on any new host in the cluster. In this case, you must ensure LZO is installed and configured appropriately. Without LZO being installed and configured, data compressed with LZO will not be readable. If you do not want Ambari to automatically download and install LZO, you must confirm your choice to proceed.
7. 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

You must prepare an existing database instance, before running setup and entering advanced database configuration.



### 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.

8. At `Proceed with configuring remote database connection properties [y/n]` choose `y`.
9. Setup completes.



### Note

If your host accesses the Internet through a proxy server, you must configure Ambari Server to use this proxy server.

#### More Information

[Setup Options](#)

[Configuring Ambari for Non-Root](#)

[Changing your JDK](#)

[Configuring LZO compression](#)

[Using an existing database with Ambari](#)

[Setting up Ambari to use an Internet proxy server](#)

## 4.3.1. Setup Options

The following options are frequently used for Ambari Server setup.

### **-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. 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.

If the Ambari Server is behind a firewall, you must instruct the `ambari-server setup` command to use a proxy when downloading a JDK. To do so, define the `http_proxy` environment variable in the shell before running the setup command. For example:

```
export http_proxy=http://{username}:  
{password}@{proxyHost}:{proxyPort}  
ambari-server setup
```

where `{username}` and `{password}` are optional.

If you do not define the `http_proxy` environment variable in a firewalled environment, the Oracle JDK download will not succeed.

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.

- enable-lzo-under-gpl-license** Use this option to download and install LZO compression, subject to the General Public License.
- 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.

**More Information**[JDK Requirements](#)[Configuring Ambari for Non-Root](#)[Configuring LZO Compression](#)[Oracle Java License Terms](#)

## 5. Working with Management Packs

Management packs allow you to deploy a range of services to your Ambari-managed cluster. You can use a management pack to deploy a specific component or service, or to deploy an entire platform, like HDF.

In general, when working with management packs, you perform the following tasks in this order:

1. Install the management pack.
2. Update the repository URL in Ambari.
3. Start the Ambari Server.
4. Launch the Ambari Installation Wizard.

## 6. Installing, Configuring, and Deploying a Cluster

Use the Ambari Cluster Install Wizard running in your browser to install, configure, and deploy your cluster, as follows:

- [Start the Ambari Server \[51\]](#)
- [Log In to Apache Ambari \[52\]](#)
- [Launch the Ambari Cluster Install Wizard \[53\]](#)
- [Name Your Cluster \[53\]](#)
- [Select Version \[53\]](#)
- [Install Options \[58\]](#)
- [Confirm Hosts \[59\]](#)
- [Choose Services \[60\]](#)
- [Assign Masters \[61\]](#)
- [Assign Slaves and Clients \[61\]](#)
- [Customize Services \[61\]](#)
- [Review \[64\]](#)
- [Install, Start and Test \[65\]](#)
- [Complete \[65\]](#)

### 6.1. 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
```



#### Note

If you plan to use an existing database instance for Hive or for Oozie, you must prepare to use an existing database **before** installing your Hadoop cluster.



On Ambari Server start, Ambari runs a database consistency check looking for issues. If any issues are found, Ambari Server **start will abort** and display the following message: `DB configs consistency check failed. Ambari writes more details about database consistency check results to the /var/log/ambari-server/ambari-server-check-database.log` file.

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.

### Next Steps

[Install, Configure and Deploy a Hadoop cluster](#)

### More Information

- [Using a new or existing database with Hive](#)
- [Using an existing database with Oozie](#)

## 6.2. Log In to Apache Ambari

### Prerequisites

Ambari Server must be running.

To log in to Ambari Web using a web browser:

### Steps

1. Point your web browser to

```
http://<your.ambari.server>:8080
```

,where `<your.ambari.server>` is the name of your ambari server host.

For example, a default Ambari server host is located at `http://c7401.ambari.apache.org:8080`.

2. Log in to the Ambari Server using the default user name/password: `admin/admin`. You can change these credentials later.

For a new cluster, the Cluster Install wizard displays a Welcome page.

### Next Step

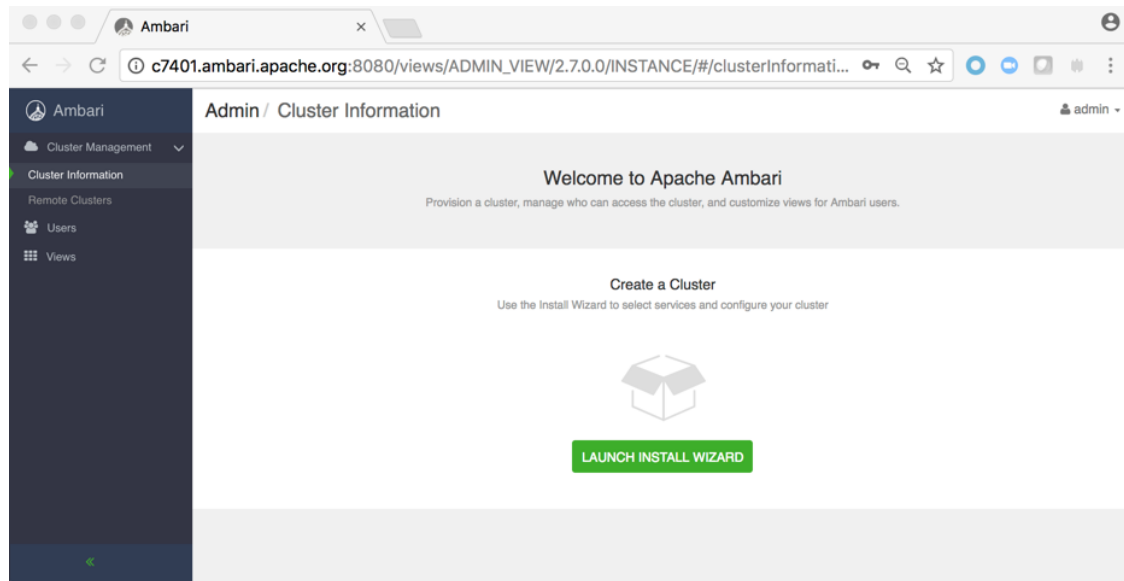
[Launch the Ambari Cluster Install Wizard \[53\]](#)

### More Information

[Start the Ambari Server \[51\]](#)

## 6.3. Launch the Ambari Cluster Install Wizard

From the Ambari Welcome page, choose **Launch Install Wizard**.



### Next Step

[Name Your Cluster \[53\]](#)

## 6.4. Name Your Cluster

### Steps

1. In **Name your cluster**, type a name for the cluster you want to create.

Use no white spaces or special characters in the name.



### Note

If you plan to Kerberize the cluster, consider limiting the cluster name (to 12 characters or less), to accommodate the fact that Kerberos principals will be appended to the cluster name string and that some identity providers impose a limit on the total principal name length.

2. Choose **Next**.

### Next Step

[Select Version \[53\]](#)

## 6.5. Select Version

In this Step, you will select the software version and method of delivery for your cluster. Using a Public Repository requires Internet connectivity. Using a Local Repository requires you have configured the software in a repository available in your network.

## Choosing Stack

The available versions are shown in TABs. When you select a TAB, Ambari attempts to discover what specific version of that Stack is available. That list is shown in a DROPDOWN. For that specific version, the available Services are displayed, with their Versions shown in the TABLE.

HDP-3.1 HDP-3.0

HDP-3.1 ▾

Accumulo	1.7.0
Infra Solr	0.1.0
Ambari Metrics	0.1.0
Atlas	0.7.0.3.1
Druid	0.12.1

## Choosing Version

If Ambari has access to the Internet, the specific Versions will be listed as options in the DROPDOWN. If you have a Version Definition File for a version that is not listed, you can click **Add Version...** and upload the VDF file. In addition, a **Default Version Definition** is also included in the list if you do not have Internet access or are not sure which specific version to install.



### Note

In case your Ambari Server has access to the Internet but has to go through an Internet Proxy Server, be sure to setup the Ambari Server for an Internet Proxy.

HDP-3.1 HDP-3.0

HDP-3.1 ▾

- HDP-3.1 (Default Version Definition)
- Add Version ...

## Choosing Repositories

Ambari gives you a choice to install the software from the Public Repositories (if you have Internet access) or Local Repositories. Regardless of your choice, you can edit the Base URL of the repositories. The available operating systems are displayed and you can add/remove operating systems from the list to fit your environment.

The UI displays repository Base URLs based on Operating System Family (OS Family). Be sure to set the correct OS Family based on the Operating System you are running.

<b>redhat7</b>	Red Hat 7, CentOS 7, Oracle Linux 7, Amazon Linux 2
<b>sles12</b>	SUSE Linux Enterprise Server 12
<b>ubuntu14</b>	Ubuntu 14
<b>ubuntu16</b>	Ubuntu 16
<b>ubuntu18</b>	Ubuntu 18
<b>debian9</b>	Debian 9

### Advanced Options

There are advanced repository options available.

- **Skip Repository Base URL validation (Advanced):** When you click **Next**, Ambari will attempt to connect to the repository Base URLs and validate that you have entered a valid repository. If not, an error will be shown that you must correct before proceeding.
- **Use RedHat Satellite/Spacewalk:** This option will only be enabled when you plan to use a Local Repository. When you choose this option for the software repositories, you are responsible for configuring the repository channel in Satellite/Spacewalk and confirming the repositories for the selected **stack version** are available on the hosts in the cluster.

### More Information

[Using a local RedHat Satellite or Spacewalk repository \[55\]](#)

## 6.5.1. Using a local RedHat Satellite or Spacewalk repository

Many Ambari users use RedHat Satellite or Spacewalk to manage Operating System repositories in their cluster. The general process to configure Ambari to work with your Satellite or Spacewalk infrastructure is to:

1. Ensure you have created channels for the [public repositories](#) that correspond to the products you intend to use.
2. Ensure the created channels are available on all machines in the cluster.
3. Install the Ambari Server and start it.
4. Before starting a cluster install, update Ambari so it knows not to delegate repository management to Satellite or Spacewalk, and use the appropriate channel names when installing or upgrading packages.



### Note

Please have the names of your channels on hand before proceeding.

Next Step

[Configuring Ambari to use RedHat Satellite or Spacewalk \[56\]](#)

### 6.5.1.1. Configuring Ambari to use RedHat Satellite or Spacewalk

The Ambari Server uses Version Definition Files (VDF) to understand which product and component versions are included in a release. In order for Ambari to work well with Satellite or Spacewalk, you must create a custom VDF file for the specific Operating System versions in your cluster that tells Ambari which RedHat Satellite or Spacewalk channel names to use when installing or upgrading the cluster.

To create a custom VDF file, we recommend downloading an existing VDF from the our [HDP 3.1 Repositories](#) table to your local desktop. Once downloaded, open the VDF file in your preferred editor and change the <repopid/> tags for each repository to match the Satellite or Spacewalk channel names previously configured. For this example, I've created the following channels in Satellite or Spacewalk:

**Table 6.1. Example Channel Names for Hortonworks Repositories**

Hortonworks Repository	RedHat Satellite or Spacewalk Channel Name
HDP-3.1.0.0	hdp_3.1.0.0
HDP-3.1-GPL*	hdp_3.1_gpl
HDP-UTILS-1.1.0.22	hdp_utils_1.1.0.22

\* If LZO compression is going to be used in your cluster, see [Configuring LZO Compression](#) for more information.

```
<repository-info>
  <os family="redhat7">
    <package-version>3_0_0_0_*</package-version>
    <repo>
      <baseurl>http://public-repo-1.hortonworks.com/HDP/centos7/3.x/updates/3.1.0.0</baseurl>
      <repopid>hdp_3.1.0.0</repopid>
      <reponame>HDP</reponame>
      <unique>true</unique>
    </repo>
    <repo>
      <baseurl>http://public-repo-1.hortonworks.com/HDP-GPL/centos7/3.x/updates/3.1.0.0</baseurl>
      <repopid>hdp_3.1_gpl</repopid>
      <reponame>HDP-GPL</reponame>
      <unique>true</unique>
      <tags>
        <tag>GPL</tag>
      </tags>
    </repo>
    <repo>
      <baseurl>http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.22/repos/centos7</baseurl>
      <repopid>hdp_utils_1.1.0.22</repopid>
      <reponame>HDP-UTILS</reponame>
      <unique>false</unique>
    </repo>
  </os>
</repository-info>
```

#### Next Step

[Import the custom VDF into Ambari \[57\]](#)

### 6.5.1.2. Import the custom VDF into Ambari

To import the custom VDF into Ambari, follow these steps:

1. In the cluster install wizard, **Select Version** step, click the drop down with the HDP version listed and select **Add Version**.



2. In **Add Version**, choose **Upload Version Definition File** and click **Choose File**. Browse to the directory on your local desktop where the VDF file has been stored, click **Choose File**, then click **Read Version Info**.

#### Add Version

Upload Version Definition File

No file chosen

Version Definition File URL

3. In **Select Version**, under **Repositories**, click **Use Local Repository**.signal to Ambari that repositories should not be downloaded from the internet.

**Repositories**  
Using a Public Repository requires Internet connectivity. Using a Local Repository requires you have configured the software in a repository available in your network.

Use Public Repository  Use Local Repository

Provide Base URLs for the Operating Systems you are configuring.

OS	Name	Base URL
redhat7	HDP-3.1	<input type="text" value="Enter Base URL or remove this OS"/>
	HDP-UTILS-1.1.0.22	<input type="text"/>

Skip Repository Base URL validation (Advanced)

Use RedHat Satellite/Spacewalk

This signals to Ambari that repositories should not be downloaded from the internet.

4. In **Base URL**, type the protocol that prefixes your Base URL. For example: `http://`

5. Verify that the OS matches the operating system specific in the Base URL value.
6. Edit the **Name** of the repository to match the channel names in your RedHat Satellite or Spacewalk installation.

Provide Base URLs for the Operating Systems you are configuring.

OS	Name	Base URL
redhat7	custom_HDP-3.1	http://
	custom_HDP-UTILS-1.1.0.22	http://

7. In **Repositories**, click the **Use RedHat Satellite/Spacewalk** checkbox.
8. Click **Next**.

### Next Step

[Install Options \[58\]](#)

### More Information

[Setting up an Internet Proxy Server for Ambari](#)

[Using a Local Repository](#)

## 6.6. Install Options

In order to build up the cluster, the Cluster Install wizard prompts you for general information about how you want to set it up. You need to supply the FQDN of each of your hosts. The wizard also needs to access the private key file you created when you set up password-less SSH. Using the host names and key file information, the wizard can locate, access, and interact securely with all hosts in the cluster.

### Steps

1. In **Target Hosts**, enter your list of host names, one per line.

You can use ranges inside brackets to indicate larger sets of hosts. For example, for host01.domain through host10.domain use `host[01-10].domain`



### Note

If you are deploying on EC2, use the `internal Private DNS` host names.

2. If you want to let Ambari automatically install the Ambari Agent on all your hosts using SSH, select **Provide your SSH Private Key** and either use the **Choose File** button in the **Host Registration Information** section to find the private key file that matches the public key you installed earlier on all your hosts or cut and paste the key into the text box manually.
3. Enter the user name for the SSH key you have selected. If you do not want to use `root`, you must provide the user name for an account that can execute `sudo` without entering

a password. If SSH on the hosts in your environment is configured for a port other than 22, you can change that also.

4. If you do not want Ambari to automatically install the Ambari Agents, select **Perform manual registration**.
5. Choose **Register and Confirm** to continue.

#### Next Step

[Confirm Hosts \[59\]](#)

#### More Information

[Set Up Password-less SSH](#)

[Installing Ambari Agents Manually](#)

## 6.7. Confirm Hosts

**Confirm Hosts** prompts you to confirm that Ambari has located the correct hosts for your cluster and to check those hosts to make sure they have the correct directories, packages, and processes required to continue the install.

If any hosts were selected in error, you can remove them by selecting the appropriate checkboxes and clicking the grey **Remove Selected** button. To remove a single host, click the small white **Remove** button in the Action column.

At the bottom of the screen, you may notice a yellow box that indicates some warnings were encountered during the check process. For example, your host may have already had a copy of `wget` or `curl`. Choose **Click here to see the warnings** to see a list of what was checked and what caused the warning. The warnings page also provides access to a python script that can help you clear any issues you may encounter and let you run

Rerun Checks



### Note

If Ambari Agents fail to register with Ambari Server during the **Confirm Hosts** step in the Cluster Install wizard. Click the **Failed** link on the Wizard page to display the Agent logs. The following log entry indicates the SSL connection between the Agent and Server failed during registration:

```
INFO 2014-04-02 04:25:22,669 NetUtil.py:55 - Failed
to connect to https://<ambari-server>:8440/cert/ca due
to [Errno 1] _ssl.c:492: error:100AE081:elliptic curve
routines:EC_GROUP_new_by_curve_name:unknown group
```

When you are satisfied with the list of hosts, choose **Next**.

#### Next Step



[Choose Services \[60\]](#)

## 6.8. Choose Services

Based on the Stack chosen during the **Select Stack** step, you are presented with the choice of Services to install into the cluster. A Stack comprises many services. You may choose to install any other available services now, or to add services later. The Cluster Install wizard selects all available services for installation by default.

SmartSense deployment is mandatory. You cannot clear the option to install SmartSense using the Cluster Install wizard.

<input type="checkbox"/>	SmartSense	1.5.1.2.7.3.0-138	SmartSense - Hortonworks SmartSense Tool (HST) helps quickly gather configuration, metrics, logs from common HDP services that aids to quickly troubleshoot support cases and receive cluster-specific recommendations.
--------------------------	------------	-------------------	---

To choose the services that you want to deploy:

### Steps

1. Choose **none** to clear all selections, or choose **all** to select all listed services.
2. Choose or clear individual check boxes to define a set of services to install now.
3. After selecting the services to install now, choose **Next**.



### Note

After adding some services, you may need to perform additional tasks. For more information about installing and configuring specific services, see the following topics:

- [Installing Apache Spark](#)
- [Installing and Configuring Apache Storm](#)
- [Configuring Storm for Kerberos](#)
- [Installing and Configuring Apache Kafka](#)
- [Configuring Kafka for Kerberos](#)
- [Installing Apache Atlas](#)
- [Installing Apache Ranger Using Ambari](#)
- [Apache Solr Search Installation](#)

### Next Step

[Assign Masters \[61\]](#)

### More Information

[Add a service](#)

[Introduction to SmartSense](#)

## 6.9. Assign Masters

The Cluster Install wizard assigns the master components for selected services to appropriate hosts in your cluster and displays the assignments in Assign Masters. The left column shows services and current hosts. The right column shows current master component assignments by host, indicating the number of CPU cores and amount of RAM installed on each host.

1. To change the host assignment for a service, select a host name from the drop-down menu for that service.
2. To remove a ZooKeeper instance, click the green - icon next to the host address you want to remove.
3. When you are satisfied with the assignments, choose **Next**.

**Next Step**

[Assign Slaves and Clients \[61\]](#)

## 6.10. Assign Slaves and Clients

The Cluster Install wizard assigns the slave components, such as DataNodes, NodeManagers, and RegionServers, to appropriate hosts in your cluster. It also attempts to select hosts for installing the appropriate set of clients.

**Steps**

1. Use **all** or **none** to select all of the hosts in the column or none of the hosts, respectively.  
  
If a host has an asterisk next to it, that host is also running one or more master components. Hover your mouse over the asterisk to see which master components are on that host.
2. Fine-tune your selections by using the check boxes next to specific hosts.
3. When you are satisfied with your assignments, choose **Next**.

**Next Step**

[Customize Services \[61\]](#)

## 6.11. Customize Services

The Customize Services step presents you with a set of tabs that let you review and modify your cluster setup. The Cluster Install wizard attempts to set reasonable defaults for each of the options. You are **strongly encouraged** to review these settings as your requirements might be more advanced.

**Install Ambari**

Get Started  
 Select Version  
 Confirm Hosts  
 Choose Services  
 Assign Masters  
 Assign Slaves and Clients  
 Customize Services  
 Review  
 Install, Start and Test  
 Summary

CREDENTIALS — DATABASES — DIRECTORIES — **ACCOUNTS** — ALL CONFIGURATIONS

Please review these settings for the Service Accounts

- Use Ambari to Manage Service Accounts and Groups @
- Use Ambari to Manage Group Memberships @
- Use Ambari to Manage Service Accounts UID's @

Users/Groups	Username
Ambari Metrics	<input type="text" value="ams"/>
Smoke	<input type="text" value="ambari-gs"/>
HDFS	<input type="text" value="hdfs"/>
HBase	<input type="text" value="hbase"/>
Mapreduce	<input type="text" value="mapred"/>
Yarn	<input type="text" value="yarn"/>
ZooKeeper	<input type="text" value="zookeeper"/>
Hadoop Group	<input type="text" value="hadoop"/>

Ambari will group the commonly customized configuration elements together into four categories: Credentials, Databases, Directories, and Accounts. All other configuration will be available in the All Configurations section of the Installation Wizard

### Credentials

Passwords for administrator and database accounts are grouped together for easy input. Depending on the services chosen, you will be prompted to input the required passwords for each item, and have the option to change the username used for administrator accounts



#### Note

Ranger and Atlas require strong passwords for your security. Hover over each property to see its password requirements. Passwords that do not meet these requirements will be highlighted on the **All Configurations** tab at the end of the **Customize Services** step.

### Databases

Some services require a backing database to function. For each service that has been chosen for install that requires a database, you will be asked to select which database should be used and configure the connectivity details for the selected database.



#### Note

By default, Ambari installs a new MySQL instance for the Hive Metastore and installs a Derby instance for Oozie. If you plan to use existing databases for

MySQL/MariaDB, Oracle or PostgreSQL, modify the database type and host before proceeding. For a quick example on creating external databases on MariaDB, see *Example: Install MariaDB for use with multiple components*, in *Administering Ambari*.



### Important

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

## Directories

Choosing the right directories for data and log storage is critical. Ambari chooses reasonable defaults based on the mount points available in your environment but you are strongly encouraged to review the default directory settings recommended by Ambari. In particular, confirm directories such as /tmp and /var are not being used for HDFS NameNode directories and DataNode directories under the HDFS tab.

## Accounts

The service account users and groups are also configurable from the Accounts tab. These are the operating system accounts the service components will run as. If these users do not exist on your hosts, Ambari will automatically create the users and groups locally on the hosts. If these users already exist, Ambari will use those accounts.

Depending on how your environment is configured, you might not allow groupmod or usermod operations. If this is the case, there are multiple options to choose how Ambari should handle user creation and modification:

<b>Use Ambari to Manage Service Accounts and Groups</b>	Ambari will create the service accounts and groups that are required for each service if they do not exist in /etc/passwd, and in /etc/group of the Ambari Managed hosts.
<b>Use Ambari to Manage Group Memberships</b>	Ambari will add or remove the service accounts from groups.
<b>Use Ambari to Manage Service Accounts UID's</b>	Ambari will be able to change the UID's of all service accounts.

## All Configurations

Here you have an opportunity to review and revise the remaining configurations for your services. Browse through each configuration tab. Hovering your cursor over each of the properties, displays a brief description of what the property does. The number of service tabs shown here depends on the services you decided to install in your cluster. **Any service with configuration issues that require attention will show up in the bell icon with the number properties that need attention.**

The screenshot shows the Ambari Installer interface. On the left is a navigation sidebar with steps: Get Started, Select Version, Install Options, Confirm Hosts, Choose Services, Assign Masters, Assign Slaves and Clients, Customize Services, Review, Install, Start and Test, and Summary. The main area is titled 'NameNode' and shows configuration sliders for 'NameNode Java heap size' and 'NameNode Server threads'. A 'Required Configurations' dialog box is open, displaying a table of properties that must be set to proceed with the install. Below this, 'Review Recommendations' are shown, including a table of configuration changes.

Property	Service	Config Group	File Name	Original Value	Recommended Value
namenode_opt_maxheapsize	HDFS	Default	hadoop-env	128	160
namenode_opt_newsize	HDFS	Default	hadoop-env	128	160

The bell popover contains configurations that require your attention, configurations that are highly recommended to be reviewed and changed, and configurations that will be automatically changed based on Ambari's recommendations unless you choose to opt out of those changes. Required Configuration must be addressed in order to proceed on to the next step in the Wizard. Carefully review the required and recommended settings and address issues before proceeding

After you complete Customizing Services, choose **Next**.

### Next Step

[Review \[64\]](#)

### More Information

[Using an existing or installing a default database](#)

[Understanding service users and groups](#)

## 6.12. Review

**Review** displays the assignments you have made. Check to make sure everything is correct. If you need to make changes, use the left navigation bar to return to the appropriate screen.

To print your information for later reference, choose **Print**.

To export the blueprint for this cluster, choose **Generate Blueprint**.

When you are satisfied with your choices, choose **Deploy**.

### Next Step

[Install, Start and Test \[65\]](#)

## 6.13. Install, Start and Test

The progress of the install displays on the screen. Ambari installs, starts, and runs a simple test on each component. Overall status of the process displays in progress bar at the top of the screen and host-by-host status displays in the main section. Do not refresh your browser during this process. Refreshing the browser may interrupt the progress indicators.

To see specific information on what tasks have been completed per host, click the link in the **Message** column for the appropriate host. In the **Tasks** pop-up, click the individual task to see the related log files. You can select filter conditions by using the **Show** drop-down list. To see a larger version of the log contents, click **Open** or to copy the contents to the clipboard, use **Copy**.

When Successfully installed and started the services appears, choose **Next**.

**Next Step**

[Complete \[65\]](#)

## 6.14. Complete

The **Summary** page provides you a summary list of the accomplished tasks. Choose **Complete**. Ambari Web opens in your web browser.