Hortonworks Data Platform

Apache Ambari Installation for IBM Power Systems

(August 23, 2018)

docs.cloudera.com

Hortonworks Data Platform: Apache Ambari Installation for IBM Power Systems

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.

Table of Contents

List of Tables

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.

- Meet Minimum System Requirements [1]

- Collect Information [4]

- Prepare the Environment [5]

- Using a Local Repository [9]

1.1. Meet Minimum System Requirements

To run Hadoop, your system must meet the following minimum requirements:

- Operating Systems Requirements [1]

- Browser Requirements [2]

- Software Requirements [2]

- JDK Requirements [2]

- Database Requirements [2]

- Memory Requirements [3]

- Package Size and Inode Count Requirements [3]

- Check the Maximum Open File Descriptors [4]

1.1.1. Operating Systems Requirements

The following, power operating system is supported:

- Red Hat Enterprise Linux (RHEL) v7.4, v7.5

> ⚠️ **Important**
>
> The installer pulls many packages from the base OS repositories. If you do not have a complete set of base OS repositories available to all your machines at the time of installation you may run into issues.

If you encounter problems with base OS repositories being unavailable, please contact your system administrator to arrange for these additional repositories to be proxied or mirrored.

More Information

Using a Local Repository [9]

1.1.2. Browser Requirements

The Ambari Install Wizard runs as a browser-based Web application. You must have a machine capable of running a graphical browser to use this tool. The minimum required browser versions are:

- Linux (RHEL 7)

  - Firefox 18

  - Google Chrome 26

We recommend updating your browser to the latest, stable version.

1.1.3. Software Requirements

On each of your hosts:

- yum and rpm (RHEL 7)

- curl, jsch, scp, tar, unzip, wget, and gcc*

- OpenSSL (v1.01, build 16 or later)

- Python 2.7 (with python-devel)

  *Ambari Metrics Monitor uses a python library (psutil) which requires gcc and python-devel packages.

1.1.4. JDK Requirements

OpenJDK 1.8 for PPC is the only JDK supported.

You must install the following OpenJDK 1.8 for PPC packages as a prerequisite on all machines in the cluster:

- java-1.8.0-openjdk

- java-1.8.0-openjdk-devel

- java-1.8.0-openjdk-headless

Set your JAVA_HOME:

```
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk
```

1.1.5. Database Requirements

Ambari requires a relational database to store information about the cluster configuration and topology. If you install Druid, Hive, Oozie, or Ranger; they also require a relational database. The following table outlines these database requirements:

Component	Databases	Description
Ambari	MariaDB - 10.1	By default, Ambari installs an instance of PostgreSQL on the Ambari Server host.*
Druid	MariaDB - 10.1	
Hive	MariaDB - 10.1	By default (on RHEL 7), Ambari installs an instance of MySQL on the Hive Metastore host.
Oozie	MariaDB - 10.1	

Component	Databases	Description
Ranger	MariaDB - 10.1	

More Information

Using Ambari with MySQL/MariaDB

Configuring a Database Instance for Ranger

1.1.6. 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 the Using Ambari Metrics section 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

The above is offered as guidelines. Be sure to test for your particular environment.

More Information

Understanding Ambari Metrics Service

Package Size and Inode Count Requirements [3]

1.1.7. Package Size and Inode Count Requirements

*Size and Inode values are approximate

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

	Size	Inodes
Ambari Metrics Monitor	1MB	100
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

1.1.8. Check the Maximum Open File Descriptors

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

```
ulimit -Sn
```

```
ulimit -Hn
```

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

```
ulimit -n 10000
```

1.2. 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 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 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.

1.3. Prepare the Environment

To deploy your Hadoop instance, you must prepare your deployment environment:

1.3.1. Set Up Password-less SSH

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

> **Note**
>
> You can choose to manually install the Agents on each cluster host. In this case, you do not need to generate and distribute SSH keys.

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.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.

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

1.3.2. Set Up Service User Accounts

Each service requires a service user account. The Ambari 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

1.3.3. 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, run the following command on each host:

```
yum install -y ntp
```

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

```
systemctl enable ntpd
```

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

```
systemctl start ntpd
```

1.3.4. Check DNS and NSCD

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

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

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.

1.3.4.1. 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
> ```

1.3.4.2. 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>
```

1.3.4.3. 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>
```

1.3.5. 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:

```
systemctl disable firewalld
```

```
service firewalld stop
```

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

Ambari checks whether iptables is running during the Ambari Server setup process. If iptables is running, a warning displays, reminding you to check that required ports are open and available. The Host Confirm step in the Cluster Install Wizard also issues a warning for each host that has iptables running.

More Information

Configuring network port numbers

1.3.6. Disable SELinux and PackageKit and check the umask Value

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

   ```
   setenforce 0
   ```

   Note

   To permanently disable SELinux set

   ```
   SELINUX=disabled
   ```

   in /etc/selinux/config This ensures that SELinux does not turn itself on after you reboot the machine .

2. 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 supports 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. Using a Local Repository

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

• Obtaining the Repositories [9]

• Set up a local repository having:

  • Setting Up a Local Repository with No Internet Access [11]

  • Setting up a Local Repository With Temporary Internet Access [13]

• Preparing The Ambari Repository Configuration File [14]

Important

Setting up and using a local repository is **optional**. After obtaining

1.4.1. Obtaining the Repositories

This section describes how to obtain:

• Ambari Repositories [9]

• HDP Stack Repositories [10]

1.4.1.1. Ambari Repositories

Use the link appropriate for your OS family to **download a repository file** that contains the software for setting up Ambari.

Ambari 2.7.1 Repositories

OS	Format	URL
RedHat 7	Base URL	https://archive.cloudera.com/p/ambari/2.x/2.7.1.0/centos7-ppc
	Repo File	https://archive.cloudera.com/p/ambari/2.x/2.7.1.0/centos7-ppc/ambari.repo

| | Tarball md5 \| asc | https://archive.cloudera.com/p/ambari/2.x/2.7.1.0/centos7-ppc/ambari-2.7.1.0-centos7-ppc.tar.gz |

1.4.1.2. HDP Stack Repositories

Use the link appropriate for your OS family to download a repository file that contains the software for setting up the Stack.

- HDP 3.0.1 Repositories [10]

1.4.1.2.1. HDP 3.0.1 Repositories

OS	Version Number	Repository Name	Format	URL
RedHat 7	HDP-3.0.1.0	HDP	Version Definition File (VDF)	https://archive.cloudera.com/p/HDP/3.x/3.0.1.0/centos7-ppc/HDP-3.0.1.0-187.xml
			Base URL	https://archive.cloudera.com/p/HDP/3.x/3.0.1.0/centos7-ppc
			Repo File	https://archive.cloudera.com/p/HDP/3.x/3.0.1.0/centos7-ppc/hdp.repo
			Tarball md5 | asc	https://archive.cloudera.com/p/HDP/3.x/3.0.1.0/centos7-ppc/HDP-3.0.1.0-centos7-ppc-rpm.tar.gz
		HDP-UTILS	Base URL	https://archive.cloudera.com/p/HDP-UTILS/1.1.0.22/repos/centos7-ppc
			Repo File	https://archive.cloudera.com/p/HDP-UTILS/1.1.0.22/repos/centos7-ppc/hdp-utils.repo
			Tarball md5 | asc	https://archive.cloudera.com/p/HDP-UTILS/1.1.0.22/repos/centos7-ppc/HDP-UTILS-1.1.0.22-centos7-ppc.tar.gz
		HDP-GPL	URL	https://archive.cloudera.com/p/HDP-GPL/3.x/3.0.1.0/centos7-ppc/hdp.gpl.repo
			Tarball md5 | asc	https://archive.cloudera.com/p/HDP-GPL/3.x/3.0.1.0/centos7-ppc/HDP-GPL-3.0.1.0-centos7-ppc-gpl.tar.gz

1.4.1.3. Setting Up a Local Repository

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

- No Internet Access

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

- Temporary Internet Access

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

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

  - Getting Started Setting Up a Local Repository [11]

  - Setting Up a Local Repository with No Internet Access [11]

  - Setting up a Local Repository With Temporary Internet Access [13]

1.4.1.4. Getting Started Setting Up a Local Repository

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

• Select an existing server in, or accessible to the cluster, that runs a supported operating system.

• Enable network access from all hosts in your cluster to the mirror server.

• Ensure the mirror server has a package manager installed such as yum (RHEL7).

• **Optional:** If your repository has temporary Internet access, and you are using RHEL/CentOS/Oracle Linux as your OS, install yum utilities:

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

1. Create an HTTP server.

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

   b. Activate this web server.

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

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

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

   • For example, from a shell window, type:

   ```
   mkdir –p /var/www/html/
   ```

   • If you are using a symlink, enable the

   ```
   followsymlinks
   ```

   on your web server.

Next Steps

After you have completed the steps in this section, move on to specific set up for your repository internet access type.

More Information

httpd://apache.org/download.cgi

1.4.1.4.1. Setting Up a Local Repository with No Internet Access

Prerequisites

Complete the Getting Started Setting up a Local Repository procedure.

--

To finish setting up your repository, complete the following:

Steps

1. Obtain the tarball for the repository you would like to create.

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

   a. Browse to the web server directory you created.

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

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

   Untar Locations for a Local Repository - No Internet Access

Repository Content	Repository Location
Ambari Repository	Untar under <web.server.directory>
HDP Stack Repositories	Create directory and untar under <web.server.directory>/hdp

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

   URLs for a Local Repository - No Internet Access

Repository	Base URL
Ambari Base URL	http://<web.server>/Ambari-2.7.1.0/centos7
HDP Base URL	http://<web.server>/hdp/HDP/centos7/3.x/updates/<latest.version>
HDP-UTILS Base URL	http://<web.server>/hdp/HDP-UTILS-<version>/repos/centos7

   where <web.server> = FQDN of the web server host.

   ⚠️ **Important**

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

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

   • Install the plug-in.

   - ```
     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 the Repositories [9]

1.4.1.4.2. Setting up a Local Repository With Temporary Internet Access

Prerequisites

Complete the Getting Started Setting up a Local Repository procedure.

--

To finish setting up your repository, complete the following:

Steps

1. Put the repository configuration files for Ambari and the Stack in place on the host.

2. Confirm availability of the repositories.

   ```
   yum repolist
   ```

3. Synchronize the repository contents to your mirror server.

   • Browse to the web server directory:

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

   ```
   mkdir -p ambari/centos7
   ```

   ```
   cd ambari/centos7
   ```

   ```
   reposync -a ppc64le -r ambari-2.7.1.0
   ```

   • For HDP Stack Repositories, create `hdp` directory and reposync.

   ```
   mkdir -p hdp/centos7
   ```

   ```
   cd hdp/centos7
   ```

   ```
   reposync -a ppc64le -r HDP-3.0.1.0
   ```

   ```
   reposync -a ppc64le -r HDP-UTILS-1.1.0.22
   ```

4. Generate the repository metadata.

   • For Ambari:

   ```
   createrepo <web.server.directory>/ambari/centos7/Updates-Ambari-2.7.1.0
   ```

   • For HDP Stack Repositories:

   ```
   createrepo <web.server.directory>/hdp/centos7/HDP-<latest.version>
   ```

   ```
   createrepo <web.server.directory>/hdp/centos7/HDP-UTILS-<version>
   ```

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

   URLs for the New Repository

Repository	Base URL
Ambari Base URL	http://<web.server>/ambari/centos7/Updates-Ambari-2.7.1.0
HDP Base URL	http://<web.server>/hdp/centos7/HDP-<latest.version>
HDP-UTILS Base URL	http://<web.server>/hdp/centos7/HDP-UTILS-<version>

where <web.server> = FQDN of the web server host.

> ⚠️ **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.

      ∘ `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 the Repositories [9]](#)

1.4.1.5. Preparing The Ambari Repository Configuration File

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

```
https://archive.cloudera.com/p/ambari/2.x/2.7.1.0/centos7-ppc/ambari.repo
```

2. In the ambari.repo file, replace the base URL `baseurl` with the local repository URL.

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

```
[Updates-Ambari-2.7.1.0]
```

```
name=Ambari-2.7.1.0-Updates
```

```
baseurl=INSERT-BASE-URL
```

```
gpgcheck=1
```

```
gpgkey=https://archive.cloudera.com/p/ambari/2.x/2.7.1.0/centos7-ppc/RPM-
GPG-KEY/RPM-GPG-KEY-Jenkins
```

```
enabled=1
```

```
priority=1
```

Base URL for a Local Repository

Local Repository	Base URL
Built with Repository Tarball (No Internet Access)	http://<web.server>/Ambari-2.7.1.0/centos7
Built with Repository File (Temporary Internet Access)	http://<web.server>/ambari/centos7/Updates-Ambari-2.7.1.0

where <web.server> = FQDN of the web server host.

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

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

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

  ```
  [main]
  ```

  ```
  enabled=1
  ```

  ```
  gpgcheck=0
  ```

Next Steps

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

More Information

Setting Up a Local Repository with No Internet Access [11]

Setting up a Local Repository With Temporary Internet Access [13]

2. Installing Ambari

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

1.

2.

3.

2.1. Download the Ambari Repository

Ambari 2.7.1 Repositories

OS	Format	URL
RedHat 7	Base URL	https://archive.cloudera.com/p/ambari/2.x/2.7.1.0/centos7-ppc
	Repo File	https://archive.cloudera.com/p/ambari/2.x/2.7.1.0/centos7-ppc/ambari.repo
	Tarball md5 | asc	https://archive.cloudera.com/p/ambari/2.x/2.7.1.0/centos7-ppc/ambari-2.7.1.0-centos7-ppc.tar.gz

Use a command line editor to perform each instruction.

2.1.1. RHEL 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 https://archive.cloudera.com/p/ambari/2.x/2.7.1.0/centos7-ppc/
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 those in the following table for the Ambari repository listing.

The values in this table are examples. Your version values depend on your installation.

repo id	repo name	status
AMBARI.2.7.1.0-2.x	Ambari 2.x	12
base	CentOS-7 - Base	6,518
extras	CentOS-7 - Extras	15
updates	CentOS-7 - Updates	209

4. Install the Ambari bits.

```
yum install ambari-server
```

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

A successful installation displays output similar to the following:

```
Installing : postgresql-libs-9.2.15-1.e17_2.ppc64le          1/4
Installing : postgresql-9.2.15-1.e17_2.ppc64le               2/4
Installing : postgresql-server-9.2.15-1.e17_2.ppc64le        3/4
Installing : ambari-server-2.7.1.0-143.ppc64le               4/4
Verifying  : postgresql-server-9.2.15-1.e17_2.ppc64le        1/4
Verifying  : ambari-server-2.7.1.0-143.ppc64le               2/4
Verifying  : postgresql-9.2.15-1.e17_2.ppc64le               3/4
Verifying  : postgresql-libs-9.2.15-1.e17_2.ppc64le          4/4

Installed:
  ambari-server.ppc64le  0:2.7.1.0-143

Dependency Installed:
 postgresql.ppc64le 0:9.2.15-1.e17_2

 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 :
https://archive.cloudera.com/p/ambari/2.x/2.7.4.0/centos7-ppc/
RPM-GPG-KEY
```

Note

When deploying HDP on 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.

More Information

2.2. Set Up the Ambari Server

Before starting the Ambari Server, you **must** set up the Ambari Server. Setup configures Ambari to talk to the Ambari database, installs the JDK and allows you to customize the user account the Ambari Server daemon will run as. The

```
ambari-server setup
```

command manages the setup process.

Prerequisites

To use MySQL as the Ambari database, you must set up the mysql connector, create a user and grant user permissions before running ambari-setup.

[Using Ambari with MySQL or MariaDB](#)

Steps

1. To start the setup process, run the following command on the Ambari server host. You may also append setup options to the command.

   ```
   ambari-server setup -j $JAVA_HOME
   ```

2. Respond to the setup prompt:

   Setup Options

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

Option	Description
-j (or –java-home)	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] 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*. If you select the silent setup option, you must also include the -j (or –java-home) option.
-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.

   > ⚠️ **Important**
   >
   > *If you choose the silent setup option and do not override the JDK selection, Oracle JDK installs and you agree to the Oracle Binary Code License agreement.
   >
   > **Oracle JDK is NOT supported for IBM-PPC.**
   >
   > If the Ambari Server is behind a firewall, you must instruct the ambari-server setup commad 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.

3. If you have *not* temporarily disabled SELinux, you may get a warning. Accept the default **y**, and continue.

4. 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.

5. If you have not temporarily disabled `iptables` you may get a warning. Enter **y** to continue.

6. Select **Custom JDK**, you must manually install the JDK on all hosts and specify the Java Home path.

> **Note**
>
> Open JDK v1.8 is the only supported JDK.

7. 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.

8. Select **y** at `Enter advanced database configuration`.

9. In `Advanced database configuration`, enter **Option [3] MySQL/MariaDB**, then enter the credentials you defined for user name, password and database name.

10 At `Proceed with configuring remote database connection properties [y/n]` choose **y**.

11 Setup completes.

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

Next Steps

Start the Ambari Server

More Information

Using Ambari with MySQL or MariaDB

Configuring Ambari for Non-Root

Setting up Ambari to use an Internet proxy server

Configuring LZO Compression

3. Installing, Configuring, and Deploying a Cluster

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

3.1. 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 those databases **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 a message will be printed to console "DB configs consistency check failed." More details will be written to the following log file:

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

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

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

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

Next Steps

Install, Configure and Deploy a Cluster

More Information

Using a new or existing database with Hive

Using an existing database with Oozie

3.2. 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 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 Ambari install wizard displays a Welcome page.

Next Step

Launching the Ambari Install Wizard [23]

More Information

Start the Ambari Server [21]

3.3. Launching the Ambari Install Wizard

From the Ambari Welcome page, choose Launch Install Wizard.
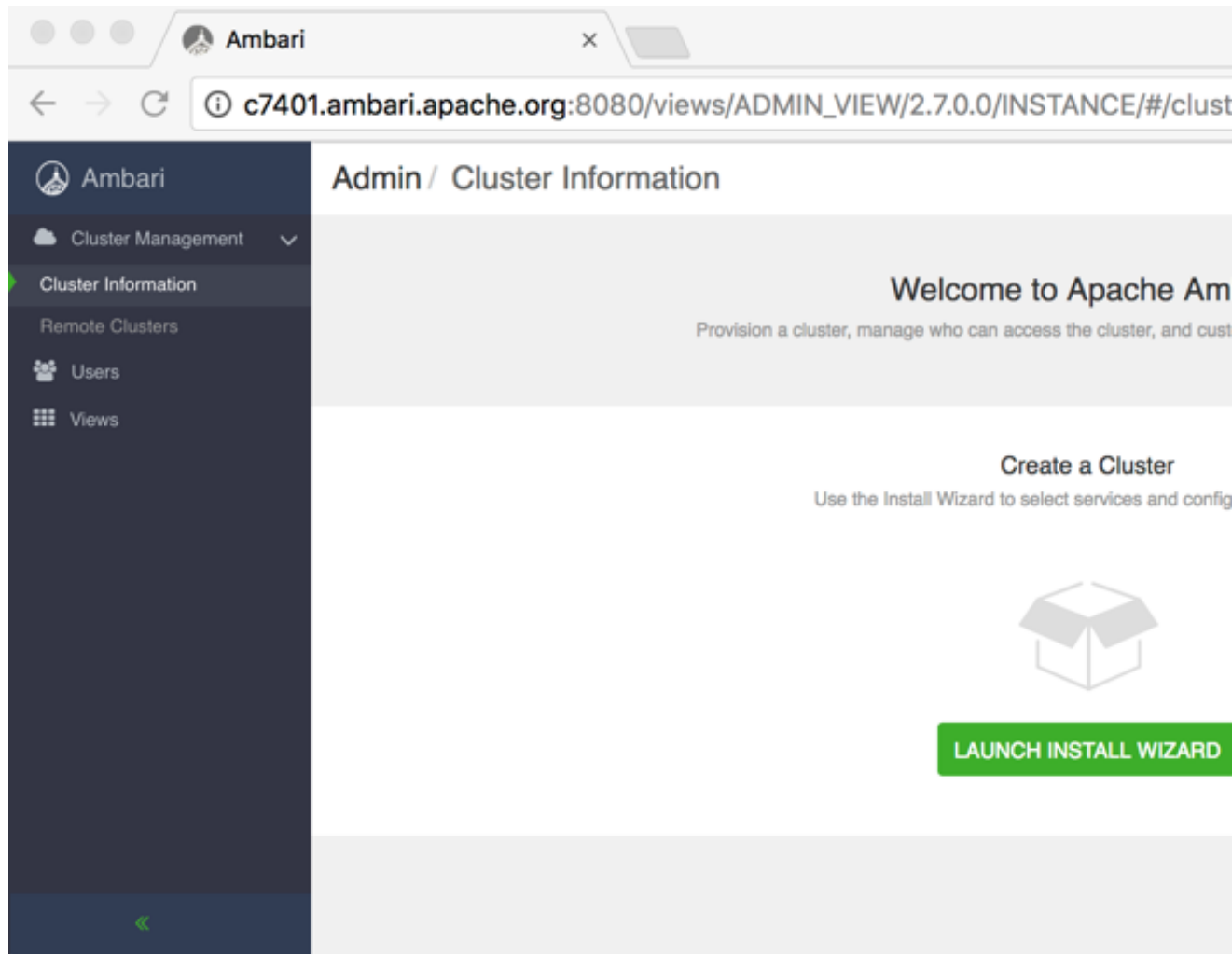
Next Step

Name Your Cluster [23]

3.4. 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.

2. Choose **Next**.

Next Step

Select Version [24]

3.5. Select Version

In this Step, you will select the software version and method of delivery for your cluster. Using a Private 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-3.0

HDP-3.0.0.0 ▾

Accumulo	1.7.0
Infra Solr	0.1.0
Ambari Metrics	0.1.0
Atlas	1.0.0
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-3.0.0.0 ▾

HDP-3.0.0.0
HDP-3.0.0.0 (Default Version Definition)
Add Version ...

Choosing Repositories

Ambari gives you a choice to install the software from the Private Repositories (if you have Internet access) or Local Repositories. Regardless of your choice, you can edit the Base URL of the repositories.

- **Ambari:** https://archive.cloudera.com/p/ambari/2.x/2.7.1.0/centos7-ppc/ambari.repo

- **HDP:** https://archive.cloudera.com/p/HDP/3.x/3.0.1.0/centos7-ppc/hdp.repo

- **HDP Utils:** https://archive.cloudera.com/p/HDP-UTILS/1.1.0.22/repos/centos7-ppc/hdp-utils.repo

- **GPL URL:** https://archive.cloudera.com/p/HDP-GPL/3.x/3.0.1.0/centos7-ppc/hdp.gpl.repo

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.

Next Step

Install Options [28]

More Information

Setting up Ambari to use an Internet proxy server

Using a Local Repository

3.5.1. 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 Hortonworks 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 [26]

3.5.1.1. 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 3.0 Repositories table to your local desktop. Once downloaded, open the VDF file in your preferred editor and change the <repoid/> 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 3.1. Example Channel Names for Hortonworks Repositories

Hortonworks Repository	RedHat Satellite or Spacewalk Channel Name
HDP-3.0.1.0	hdp_3.0.1.0
HDP-3.0.1-GPL*	hdp_3.0.1_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_1_0_*</package-version>
      <repo>
 <baseurl>https://archive.cloudera.com/p/HDP/3.x/3.0.1.0/centos7-ppc</baseurl>
        <repoid>hdp_3.0.1.0</repoid>
        <reponame>HDP</reponame>
        <unique>true</unique>
      </repo>
      <repo>
 <baseurl>https://archive.cloudera.com/p/HDP-GPL/3.x/3.0.1.0/centos7-ppc</
baseurl>
        <repoid>hdp_3.0_gpl</repoid>
        <reponame>HDP-GPL</reponame>
        <unique>true</unique>
        <tags>
          <tag>GPL</tag>
        </tags>
      </repo>
      <repo>
 <baseurl>https://archive.cloudera.com/p/HDP-UTILS/1.1.0.22/repos/centos7-
ppc</baseurl>
        <repoid>hdp_utils_1.1.0.22</repoid>
```

```
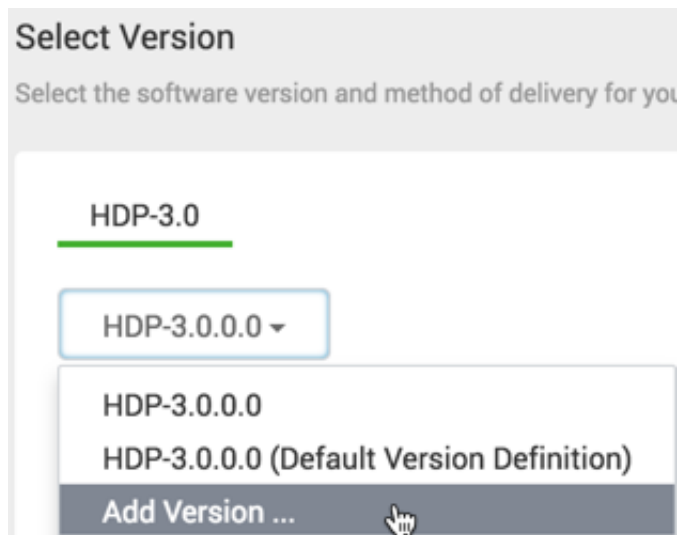        <reponame>HDP-UTILS</reponame>
        <unique>false</unique>
      </repo>
    </os>
  </repository-info>
```

Next Step

3.5.1.2. 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**.

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

4. Click the **Use RedHat Satellite/Spacewalk** checkbox.

5. Verify that the **Name** of the repository for your Operating System version is correct, and matches the VDF and channel names in your RedHat Satellite or Spacewalk installation.

6. Click **Next**.

Next Step

Install Options [28]

More Information

Setting up Ambari to use an Internet proxy server

Using a Local Repository

3.6. Install Options

In order to build up the cluster, the 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. Use the **Target Hosts** text box to 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. Fill in 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 as well.

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

More Information

Installing Ambari agents manually

3.7. 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 check boxes 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
2018-04-02 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

3.8. Choose Services

Based on the Stack chosen during Select Stack, you are presented with the choice of Services to install into the cluster. Your Stack comprises many services. You may choose to install any other available services now, or to add services later. The 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.

| ☑ | SmartSense | 1.5.0.2.7.0.0-685 | 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. |

Steps

1. Choose **none** to clear all selections, or choose **all** to select all listed services.

2. Choose or clear individual checkboxes 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 [31]

More Information

Add a service

Introduction to SmartSense

3.9. Assign Masters

The Ambari 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.

Steps

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 minus 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 [31]

3.10. Assign Slaves and Clients

The Ambari installation wizard assigns the slave components (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 checkboxes next to specific hosts.

3. When you are satisfied with your assignments, choose **Next**.

Next Steps

3.11. 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.

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**
>
> **Oracle**, **PostgreSQL**, **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:

Use Ambari to Manage Service Accounts and Groups	Ambari will create the service accounts and groups that are required for each service if they do not exist in /etc/password, and in /etc/group of the Ambari Managed hosts.
Use Ambari to Manage Group Memberships	Ambari will add or remove the service accounts from groups.
Use Ambari to Manage Service Accounts UID's	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 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 [34]

More Information

Using an existing or installing a default database

3.12. Review

The assignments you have made are displayed. 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

3.13. 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 the **Open** icon or to copy the contents to the clipboard, use the **Copy** icon.

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

Next Step

3.14. Complete

The Summary page provides you a summary list of the accomplished tasks. Choose **Complete**. Ambari Web GUI displays.