

# Build and install Ambari 2.2.1 from Source

## Build and install Ambari 2.2.1

### Step 1: Download and build Ambari 2.2.1 source

```
wget http://archive.apache.org/dist/ambari/ambari-2.2.1/apache-ambari-2.2.1-src.tar.gz
tar xfvz apache-ambari-2.2.1-src.tar.gz
cd apache-ambari-2.2.1-src
mvn versions:set -DnewVersion=2.2.1

pushd ambari-metrics
mvn versions:set -DnewVersion=2.2.1
popd
```

**Note:** If running into errors while compiling the ambari-metrics package due to missing the artifacts of jms, jmxri, jmxtools:

```
[ERROR] Failed to execute goal on project ambari-metrics-kafka-sink: Could not resolve dependencies for project
org.apache.ambari:ambari-metrics-kafka-sink:jar:2.2.1-0: The following artifacts could not be resolved: javax.
jms:jms:jar:1.1, com.sun.jdmk:jmxtools:jar:1.2.1, com.sun.jmx:jmxri:jar:1.2.1: Could not transfer artifact
javax.jms:jms:jar:1.1 from/to java.net (https://maven-repository.dev.java.net/nonav/repository): No connector
available to access repository java.net (https://maven-repository.dev.java.net/nonav/repository) of type legacy
using the available factories WagonRepositoryConnectorFactory
```

The work around is to manually install the three missing artifacts:

```
mvn install:install-file -Dfile=jms-1.1.pom -DgroupId=javax.jms -DartifactId=jms -Dversion=1.1 -Dpackaging=jar

mvn install:install-file -Dfile=jmxtools-1.2.1.pom -DgroupId=com.sun.jdmk -DartifactId=jmxtools -Dversion=1.2.1
-Dpackaging=jar

mvn install:install-file -Dfile=jmxri-1.2.1.pom -DgroupId=com.sun.jmx -DartifactId=jmxri -Dversion=1.2.1 -
Dpackaging=jar
```

The three poms are:

```
$ cat jms-1.1.pom
<project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>javax.jms</groupId>
  <artifactId>jms</artifactId>
  <version>1.1</version>
  <name>Java Message Service</name>
  <description>
    The Java Message Service (JMS) API is a messaging standard that allows application components based on the
    Java 2 Platform, Enterprise Edition (J2EE) to create, send, receive, and read messages. It enables distributed
    communication that is loosely coupled, reliable, and asynchronous.
  </description>
  <url>http://java.sun.com/products/jms</url>
  <distributionManagement>
    <downloadUrl>http://java.sun.com/products/jms/docs.html</downloadUrl>
  </distributionManagement>
```

```
$ cat jmxri-1.2.1.pom
<?xml version="1.0" encoding="UTF-8"?><project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.sun.jmx</groupId>
  <artifactId>jmxri</artifactId>
  <version>1.2.1</version>
  <distributionManagement>
    <status>deployed</status>
  </distributionManagement>
```

```
$ cat jmxtools-1.2.1.pom
<?xml version="1.0" encoding="UTF-8"?><project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.sun.jdmk</groupId>
  <artifactId>jmxtools</artifactId>
  <version>1.2.1</version>
  <distributionManagement>
    <status>deployed</status>
  </distributionManagement>
```

#### **RHEL (CentOS 5 or 6) & SUSE (SLES 11):**

```
mvn -B clean install package rpm:rpm -DnewVersion=2.2.1 -DskipTests -Dpython.ver="python >= 2.6"
```

#### **Ubuntu/Debian:**

```
mvn -B clean install package jdeb:jdeb -DnewVersion=2.2.1 -DskipTests -Dpython.ver="python >= 2.6"
```

**Note:** You need to have tools such as rpm-build tool, brunch, etc. For details on prerequisites, please see [Ambari Development](#).

## **Step 2: Install Ambari Server**

Install the rpm package from `ambari-server/target/rpm/ambari-server/RPMS/noarch/`

*[For CentOS 5 or 6]*

```
yum install ambari-server*.rpm #This should also pull in postgres packages as well.
```

*[For SLES 11]*

```
zypper install ambari-server*.rpm #This should also pull in postgres packages as well.
```

*[For Ubuntu/Debian]*

```
apt-get install ambari-server*.deb #This should also pull in postgres packages as well.
```

## **Step 3: Setup and Start Ambari Server**

Run the setup command to configure your Ambari Server, Database, JDK, LDAP, and other options:

```
ambari-server setup
```

Follow the on-screen instructions to proceed.

Once set up is done, start Ambari Server:

```
ambari-server start
```

## Step 4: Install and Start Ambari Agent on All Hosts

**Note:** This step needs to be run on all hosts that will be managed by Ambari.

Copy the rpm package from `ambari-agent/target/rpm/ambari-agent/RPMS/x86_64/` and run:

*[For CentOS 5 or 6]*

```
yum install ambari-agent*.rpm
```

*[For SLES 11]*

```
zypper install ambari-agent*.rpm
```

*[Ubuntu/Debian]*

```
apt-get install ambari-agent*.rpm
```

Edit `/etc/ambari-agent/ambari.ini`

```
...  
[server]  
hostname=localhost  
...
```

Make sure `hostname` under the `[server]` section points to the actual Ambari Server host, rather than "localhost".

```
ambari-agent start
```

## Step 5: Deploy Cluster using Ambari Web UI

Open up a web browser and go to `http://<ambari-server-host>:8080`.

Log in with username **admin** and password **admin** and follow on-screen instructions. Secure your environment by ensuring your administrator details are changed from the default values as soon as possible.

Under Install Options page, enter the hosts to add to the cluster. Do not supply any SSH key, and check "Perform manual registration on hosts and do not use SSH" and hit "Next".