

Hortonworks DataFlow

Installing HDF Services on a New HDP Cluster

(August 9, 2017)

Hortonworks DataFlow: Installing HDF Services on a New HDP Cluster

Copyright © 2012-2017 Hortonworks, Inc. Some rights reserved.



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. Installing Ambari	1
1.1. Getting Ready for an Ambari Installation	1
1.1.1. Review System Requirements	1
1.1.2. Set Up Password-less SSH	1
1.1.3. Set Up Service User Accounts	2
1.1.4. Enable NTP on the Cluster and on the Browser Host	2
1.1.5. Check DNS and NSCD	3
1.1.6. Configuring iptables	4
1.1.7. Disable SELinux and PackageKit and check the umask Value	4
1.2. Download the Ambari Repository	5
1.2.1. RHEL/CentOS/Oracle Linux 6	6
1.2.2. RHEL/CentOS/Oracle Linux 7	7
1.2.3. SLES 12	8
1.2.4. SLES 11	9
1.2.5. Ubuntu 14	10
1.2.6. Debian 7	11
1.3. Install the Ambari Server	11
1.3.1. RHEL/CentOS/Oracle Linux 6	12
1.3.2. RHEL/CentOS/Oracle Linux 7	13
1.3.3. SLES 12	14
1.3.4. SLES 11	15
1.3.5. Ubuntu 14	16
1.3.6. Debian 7	16
1.4. Set Up the Ambari Server	17
1.4.1. Setup Options	18
1.5. Start the Ambari Server	19
2. Installing Databases	21
2.1. Install MySQL	21
2.2. Configure SAM and Schema Registry Metadata Stores in MySQL	22
2.3. Configure Druid and Superset Metadata Stores in MySQL	22
2.4. Install Postgres	23
2.5. Configure Postgres to Allow Remote Connections	24
2.6. Configure SAM and Schema Registry Metadata Stores in Postgres	24
2.7. Configure Druid and Superset Metadata Stores in Postgres	25
3. Deploy an HDP Cluster Using Ambari	26
3.1. Install an HDP Cluster	26
3.2. Customize Druid Services	27
3.3. Configure Superset	27
3.4. Deploy the Cluster Services	28
3.5. Access the Stream Insight Superset UI	29
4. Installing the HDF Management Pack	31
5. Update the HDF Base URL	32
6. Add HDF Services to an HDP Cluster	33
7. Configure HDF Components	34
7.1. Configure Schema Registry	34
7.2. Configure SAM	35
7.3. Configure NiFi	36
7.4. Configure Kafka	36

7.5. Configure Storm	37
7.6. Deploy the Cluster Services	37
7.7. Access the UI for Deployed Services	37
8. Configuring Schema Registry and SAM for High Availability	38
9. Install the Storm Ambari View	39
10. Using a Local Repository	41
10.1. Setting Up a Local Repository	41
10.2. Getting Started Setting Up a Local Repository	41
10.2.1. Setting Up a Local Repository with No Internet Access	42
10.2.2. Setting up a Local Repository With Temporary Internet Access	44
10.3. Preparing The Ambari Repository Configuration File	46
11. Where to Go Next?	48

1. Installing Ambari

Perform the following tasks to install Ambari.

1. [Getting Ready for an Ambari Installation \[1\]](#)
2. [Download the Ambari Repository \[5\]](#)
3. [Install the Ambari Server \[11\]](#)
4. [Set Up the Ambari Server \[17\]](#)
5. [Start the Ambari Server \[19\]](#)

1.1. Getting Ready for an Ambari Installation

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.

1.1.1. Review System Requirements

Review the Support Matrices for system requirements, supported operating systems, component interoperability, and similar information.

- [HDF Support Matrices](#)

1.1.2. 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.

1.1.3. 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.

1.1.4. 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:

RHEL/CentOS/Oracle 6 `yum install -y ntp`

RHEL/CentOS/Oracle 7 `yum install -y ntp`

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

RHEL/CentOS/Oracle 6 `chkconfig --list ntpd`

RHEL/CentOS/Oracle 7 `systemctl is-enabled ntpd`

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

RHEL/CentOS/Oracle 6 `chkconfig ntpd on`

RHEL/CentOS/Oracle 7 `systemctl enable ntpd`

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

RHEL/CentOS/Oracle 6 `service ntpd start`

RHEL/CentOS/Oracle 7 `systemctl start ntpd`

1.1.5. 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.

1.1.5.1. 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
```

1.1.5.2. 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>
```

1.1.5.3. 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>
```

1.1.6. Configuring iptables

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

RHEL/CentOS/Oracle Linux 6 `chkconfig iptables off`

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

RHEL/CentOS/Oracle Linux 7 `systemctl disable firewalld`

```
service firewalld stop
```

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

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

1.1.7. 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 0022
```

Permanently changing the umask for all interactive users:

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

1.2. Download the Ambari Repository

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

- [RHEL/CentOS/Oracle Linux 6 \[6\]](#)
- [RHEL/CentOS/Oracle Linux 7 \[7\]](#)
- [SLES 12 \[8\]](#)

- [SLES 11 \[9\]](#)
- [Ubuntu 14 \[10\]](#)
- [Debian 7 \[11\]](#)

Use a command line editor to perform each instruction.

1.2.1. RHEL/CentOS/Oracle Linux 6

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

Steps

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

```
wget -nv http://public-repo-1.hortonworks.com/ambari/centos6/2.x/updates/2.5.1.0/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-2.5.1.0-159	ambari Version - ambari-2.5.1.0-159	12
base	CentOS-6 - Base	6,696
extras	CentOS-6 - Extras	64
updates	CentOS-6 - Updates	974
repolist: 7,746		

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 \[11\]](#)

- [Set Up the Ambari Server \[17\]](#)

1.2.2. RHEL/CentOS/Oracle Linux 7

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

Steps

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

```
wget -nv http://public-repo-1.hortonworks.com/ambari/centos7/2.x/updates/2.5.1.0/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-2.5.1.0-159    ambari Version - ambari-2.5.1.0-159
12
epel/x86_64           Extra Packages for Enterprise Linux 7 - x86_64
11,387
ol7_UCKR4/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 \[11\]](#)
- [Set Up the Ambari Server \[17\]](#)

1.2.3. 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/2.5.1.0/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-2.5.1.0-159	ambari Version - ambari-2.5.1.0-159	No	Yes
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 \[11\]](#)
- [Set Up the Ambari Server \[17\]](#)

1.2.4. SLES 11

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

Steps

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

```
wget -nv http://public-repo-1.hortonworks.com/ambari/suse11/2.x/updates/2.5.1.0/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-2.5.1.0-159	ambari Version - ambari-2.5.1.0-159	No	Yes
2	http-demeter.uni-regensburg.de-c997c8f9	SUSE-Linux-Enterprise-Software -Development-Kit-11-SP3 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 \[11\]](#)
- [Set Up the Ambari Server \[17\]](#)

1.2.5. 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/2.5.1.0/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 \[11\]](#)
- [Set Up the Ambari Server \[17\]](#)

1.2.6. Debian 7

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

Steps

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

```
wget -O /etc/apt/sources.list.d/ambari.list http://public-repo-1.hortonworks.com/ambari/debian7/2.x/updates/2.5.1.0/ambari.list
```

```
apt-key adv --recv-keys --keyserver keyserver.debian.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 \[11\]](#)
- [Set Up the Ambari Server \[17\]](#)

1.3. Install the Ambari Server

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

- [RHEL/CentOS/Oracle Linux 6 \[12\]](#)
- [RHEL/CentOS/Oracle Linux 7 \[13\]](#)
- [SLES 12 \[14\]](#)
- [SLES 11 \[15\]](#)
- [Ubuntu 14 \[16\]](#)
- [Debian 7 \[16\]](#)

Use a command line editor to perform each instruction.

1.3.1. RHEL/CentOS/Oracle Linux 6

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

Steps

1. 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-8.4.20-6.el6.x86_64      1/4
Installing : postgresql-8.4.20-6.el6.x86_64         2/4
Installing : postgresql-server-8.4.20-6.el6.x86_64  3/4
Installing : ambari-server-2.5.1.0-159.x86_64       4/4
Verifying  : ambari-server-2.5.1.0-159.x86_64      1/4
Verifying  : postgresql-8.4.20-6.el6.x86_64       2/4
Verifying  : postgresql-server-8.4.20-6.el6.x86_64 3/4
Verifying  : postgresql-libs-8.4.20-6.el6.x86_64  4/4
```

```
Installed:
  ambari-server.x86_64 0:2.5.1.0-159
```

```
Dependency Installed:
  postgresql.x86_64 0:8.4.20-6.el6
  postgresql-libs.x86_64 0:8.4.20-6.el6
  postgresql-server.x86_64 0:8.4.20-6.el6
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/centos6/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 \[17\]](#)

1.3.2. RHEL/CentOS/Oracle Linux 7

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

Steps

1. 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-2.5.1.0-159.x86_64            4/4
Verifying  : ambari-server-2.5.1.0-159.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 0:2.5.1.0-1050

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/centos6/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 \[17\]](#)

1.3.3. 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-2.5.1.0-159.noarch (4/4), 99.0 MiB (126.3  
MiB unpacked)  
Retrieving: ambari-server-2.5.1.0-159.noarch.rpm [done (3.0 MiB/s)]  
Installing: ambari-server-2.5.1.0-159 [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 \[17\]](#)

1.3.4. SLES 11

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 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-2.5.1.0-159.noarch (4/4), 99.0 MiB (126.3 MiB unpacked)
Retrieving: ambari-server-2.5.1.0-159.noarch.rpm [done (3.0 MiB/s)]
Installing: ambari-server-2.5.1.0-159 [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 \[17\]](#)

[Using a Local Repository](#)

1.3.5. 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 \[17\]](#)

1.3.6. Debian 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.

```
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 \[17\]](#)

1.4. Set Up the Ambari Server

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

```
ambari-server setup
```

command manages the setup process. Run the following command on the Ambari server host to start the setup process. You may also append Setup Options to the command.

```
ambari-server setup
```

Respond to the setup prompt:

1. If you have *not* temporarily disabled SELinux, you may get a warning. Accept the default (**y**), and continue.
2. By default, Ambari Server runs under `root`. Accept the default (**n**) at the `Customize user account for ambari-server daemon` prompt, to proceed as `root`. If you want to create a different user to run the Ambari Server, or to assign a previously created user, select **y** at the `Customize user account for ambari-server daemon` prompt, then provide a user name.
3. If you have not temporarily disabled `iptables` you may get a warning. Enter **y** to continue.
4. Select a JDK version to download. Enter 1 to download Oracle JDK 1.8. Alternatively, you can choose to enter a Custom JDK. If you choose Custom JDK, you must manually install the JDK on all hosts and specify the Java Home path.



Note

JDK support depends entirely on your choice of Stack versions. By default, Ambari Server setup downloads and installs Oracle JDK 1.8 and the accompanying Java Cryptography Extension (JCE) Policy Files.

5. Accept the Oracle JDK license when prompted. You must accept this license to download the necessary JDK from Oracle. The JDK is installed during the deploy phase.
6. Select **n** at `Enter advanced database configuration` to use the default, embedded PostgreSQL database for Ambari. The default PostgreSQL database name is `ambari`. The default user name and password are `ambari/bigdata`. Otherwise, to use an existing PostgreSQL, MySQL/MariaDB or Oracle database with Ambari, select **y**.
 - If you are using an existing PostgreSQL, MySQL/MariaDB, or Oracle database instance, use one of the following prompts:



Important

You must prepare a non-default 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.

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

8. Setup completes.



Note

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

1.4.1. 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 you want to run the Ambari Server as non-root, you must run setup in interactive mode. When prompted to customize the ambari-server user account, provide the account information.

- v (or -verbose)** Prints verbose info and warning messages to the console during Setup.
- g (or -debug)** Prints debug info to the console during Setup.

1.5. 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 a non-default 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.

2. Installing Databases

Schema Registry, SAM, Druid, and Superset require a relational data store to store metadata. You can use either MySQL or Postgres for this. This chapter describes how to install either MySQL or Postgres, and how create a databases for SAM and Registry. If you are installing on an existing HDP cluster with 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 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

- [Install MySQL \[21\]](#)
- [Configure SAM and Schema Registry Metadata Stores in MySQL \[22\]](#)
- [Configure Druid and Superset Metadata Stores in MySQL \[22\]](#)

Installing and Configuring Postgres

- [Install Postgres \[23\]](#)
- [Configure Postgres to Allow Remote Connections \[24\]](#)
- [Configure SAM and Schema Registry Metadata Stores in Postgres \[24\]](#)
- [Configure Druid and Superset Metadata Stores in Postgres \[25\]](#)

2.1. Install 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 into the node where 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 a 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, followed by the password you obtained in the previous step. MySQL will ask you to change the password.

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

2.2. Configure SAM and Schema Registry Metadata Stores in MySQL

Steps

1. Launch the MySQL monitor:

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

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

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

3. Create Schema Registry and SAM user accounts, replacing the last 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;
```

2.3. Configure 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 last 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;
```

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 RPM (change as necessary for your OS)

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

2. Install 9.5+ Postgres database

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

3. Initialize the database:

- For CentOS 7:

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

- For CentOS 6:

```
sudo service postgresql initdb
```

4. Start Postgres (below is for centos7):

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

5. Ensure that you can Log in:

```
sudo su postgres  
psql
```

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

```
listen_addresses = '*'
```

3. Restart Postgres

4.

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

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 into 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;
```

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 done this using MySQL, you do not need to configure additional metadata stores in Postgres.

Steps

1. Log into Postgres:

```
sudo su postgres
psql
```

2. Create a database, user, and password 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 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;
```

3. Deploy an HDP Cluster Using Ambari

To install and configure an HDP cluster for use with HDF's stream analytics capabilities, perform the following steps.

1. [Install an HDP Cluster \[26\]](#)
2. [Customize Druid Services \[27\]](#)
3. [Configure Superset \[27\]](#)
4. [Deploy the Cluster Services \[28\]](#)
5. [Access the Stream Insight Superset UI \[29\]](#)

3.1. Install an HDP Cluster

About This Task

If you are installing an HDF cluster that includes SAM, you require an existing HDP cluster with Druid installed. This section provides instructions for installing HDP for use with SAM. For complete HDP installation instructions, see the *Ambari Installation for HDP*.

Steps

1. Log on to the Ambari UI, and start the **Cluster Installation** wizard.

The default Ambari user name and password are `admin` and `admin`.

2. On the **Select Version** step under public repositories, remove all Base URLs that do not apply to your operating system.

Change the HDP Base URL to the URL appropriate for your operating system, provided in the [HDF Release Notes](#).

3. On the **Choose Services** step, you must select the following services to run an HDF cluster with full SAM capabilities. You may further customize as required by your use case and operational objectives.
 - HDFS
 - YARN + MapReduce2
 - ZooKeeper
 - Ambari Infra
 - Ambari Metrics
 - SmartSense
 - Druid

4. On the **Assign Masters** screen, distribute master services using the deployment diagram available in the *Deployment Index*.
5. On the **Assign Slaves and Clients** screen, distribute slave services using the deployment image as a guide.

More Information

[Ambari Installation for HDP](#)

3.2. Customize Druid Services

About This Task

After creating a databases for the Druid and Superset store, configure the Druid and Superset services to use the database store.

Steps

1. On the **Customize Services** page of the Wizard, navigate to the **META DATA STORAGE CONFIG** section of the Druid tab:
 - a. Select a database for the Metadata Storage type, for both Druid and Superset.
 - b. Add user `druid` and the password you created for Druid. Add user `superset` and the password you created for superset.
 - c. Set the "Metadata Storage host name" to the FQDN of the node running your database server.
 - d. Set the Metadata Storage port. The default ports are:
 - MySQL – 3306
 - Postgres – 5432
 - e. Set the "Database hostname" to the FQDN of the node running your database server.
2. Set the **SECRET_KEY** value, used to encrypt sensitive properties.



Note

You can find additional configuration information for HDP services in the [Ambari Installation for HDP](#).

3.3. Configure Superset

About This Task

Configure Superset, the data exploration and visualization UI platform.

Steps

1. Provide values for required fields.



Note

The password is used to log into the SuperSet UI. You should NOT use special characters in the password due to a bug.

2. One of the more powerful visualizations that Superset offers is integration with [MapBox](#). If you want to use this visualization, do the following:
 - a. Create an account at [MapBox](#).
 - b. After account creation, in MapBox, you can need create a token for API Access. Leave all the defaults for token creation.
 - c. You are given a token API key.
 - d. Enter that token API key in the Druid Service under Ambari for the config called `MAPBOX_API_KEY` under section **Advanced druid-superset**.

3.4. Deploy the Cluster Services

Complete the **Cluster Install** wizard and deploy all the services. If the Druid Services do not start, verify that the Druid Meta data storage settings are correct. After correcting, choose **Start All** (not Restart).

After successfully starting Druid services, you should see something like the following:

The screenshot shows the Ambari Stream Insight interface. The top navigation bar includes 'Ambari streaminsight', '0 ops', '1 alert', and tabs for 'Dashboard', 'Services', 'Hosts', 'Alerts', and 'Admin'. The left sidebar lists services like HDFS, YARN, MapReduce2, ZooKeeper, Ambari Infra, Ambari Metrics, SmartSense, and Druid. The main content area is titled 'Summary' and shows the following status for Druid components:

- Druid Broker: Started, No alerts
- Druid Coordinator: Started, No alerts
- Druid Overlord: Started, No alerts
- Druid Router: Started, No alerts
- Superset: Started, No alerts
- Druid Historicals: 4/5 Druid Historicals Live
- Druid MiddleManagers: 5/5 Druid MiddleManagers Live

3.5. Access the Stream Insight Superset UI

About This Task

As part of the install, Superset is configured with the Druid cluster that was installed as part of Ambari. To verify installation and access the UI.

Steps

1. Select the Druid Service in Ambari and under the Quick Links menu, choose **Superset** to navigate to the Stream Insight UI.

The default user account is `admin`; the password is what you configured for the Superset service.

2. Click **Sources > Druid Clusters**.
3. You should see a Druid cluster already configured. It should look like the following:

The screenshot shows the Superset UI. The top navigation bar includes 'Superset', 'Security', 'Manage', 'Sources', 'Slices', 'Dashboards', and 'SQL Lab'. The main content area is titled 'List Druid Cluster' and features a search bar and a table with the following data:

Cluster	Metadata Last Refreshed
druid-ambari	2017-02-16 14:21:46

4. Once a Druid cluster is set up, you can pull in all of the data sources or cubes associated with that cluster clicking on **Sources > Refresh Druid Metadata**.

If you have a streaming app that pushes data into data sources, the refresh should pull in all data sources within the Druid cluster. See the following image for an example.



Important





Whenever you add new data sources to Druid, perform the **Refresh Druid Metadata** action.

Superset Security Manage Sources Slices Dashboards SQL Lab

List Druid Datasource

Search

Actions Record Count: 4

	Data Source	Cluster	Changed By	Changed On	Time Offset
<input type="checkbox"/>	 Alerts-High-Speed-Cube	Streaming Analytics Manager - Stream Insight	George Veticaden	2017-02-08 05:38:49.791384	0
<input type="checkbox"/>	 Alerts-High-Speed-Cube-V2	Streaming Analytics Manager - Stream Insight	George Veticaden	2017-02-07 15:50:59.807995	0
<input type="checkbox"/>	 driver-violations-cube	Streaming Analytics Manager - Stream Insight	George Veticaden	2017-02-08 05:38:50.170646	0
<input type="checkbox"/>	 driver-violations-cube-2	Streaming Analytics Manager - Stream Insight	George Veticaden	2017-02-07 17:29:22.544088	0

4. Installing the HDF Management Pack

About This Task

A management pack (mpack) bundles service definitions, stack definitions, and stack add-on service definitions so they do not need to be included with the Ambari core functionality and can be updated in between major releases.

Steps

1. Download the Hortonworks HDF management pack. You can find the download location for your operating system in the *HDF Release Notes*.
2. Copy the bundle to `/tmp` on the node where you installed Ambari.
3. Install the management pack:

```
ambari-server install-mpack \  
--mpack=/tmp/hdf-ambari-mpack-<version>.tar.gz \  
--verbose
```

4. Restart the Ambari server:

```
ambari-server restart
```

More Information

[HDF Release Notes](#)

5. Update the HDF Base URL

About This Task

Adding the base URL tells Ambari where to look for the HDF repository. This step is necessary when you are using an existing Ambari instance, already managing an HDP cluster, to install and manage an HDF cluster.

Steps

1. From the Ambari menu, click the **admin** drop-down in the top right of your Ambari Dashboard view. Then select **Manage Ambari**.
2. From the **Clusters** view on the left, click **Versions**, and then click the **HDP version** link.
3. Configure the HDF Base URL to the base URL appropriate for your operating system. Find the HDF Base URLs in the [HDF Release Notes](#).
4. Click **Save**.

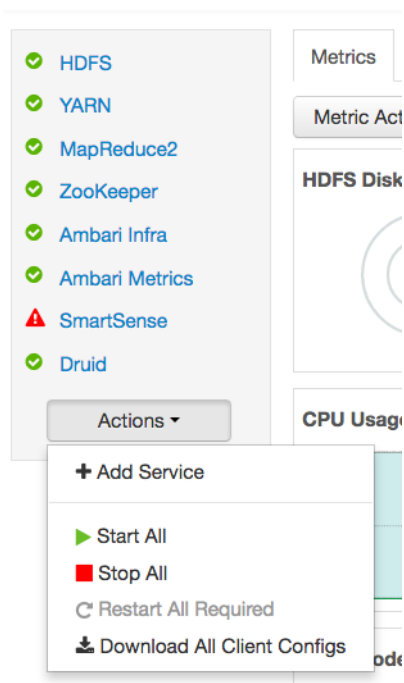
6. Add HDF Services to an HDP Cluster

About This Task

You can use the HDF management pack and Ambari to add HDF services to an HDP cluster.

Steps

1. If you are installing HDF services on an existing HDP Cluster, on the Ambari home page, click the button **Actions** and select **+ Add Service**.



2. Chose HDF services select the HDF Services needed (NiFi, Storm, Kafka, Streaming Analytics Manager, and Schema Registry)
3. On the Assign Masters screen, distribute master services using the preceding deployment diagram of the Stream Processing cluster.
4. On the Assign Slaves and Clients screen, distribute slave services using the deployment diagram of the Stream Processing cluster.

7. Configure HDF Components

You can customize your HDF component configurations during installation, or any later time. During installation, customize HDF component configurations in the **Customize Services** step of the Installation Wizard. At any other time, click the service you want to configure in the left-hand Services pane, in the Ambari Dashboard, then click the **Configs** tab.

- [Configure Schema Registry \[34\]](#)
- [Configure SAM \[35\]](#)
- [Configure NiFi \[36\]](#)
- [Configure Kafka \[36\]](#)
- [Configure Storm \[37\]](#)
- [Deploy the Cluster Services \[37\]](#)
- [Access the UI for Deployed Services \[37\]](#)

7.1. Configure Schema Registry

About This Task

The main Schema Registry configuration task you have is to establish a connection between Schema Registry and the database you are going to use as the metadata store.

Steps

1. In the **Customize Services** step, navigate to the **REGISTRY CONFIG** section of the **Registry** tab.
2. Select **Jar Storage Type**. If you plan to enable HA for Schema Registry on this cluster, you must select **HDFS**.
3. If you selected **HDFS** as the **Jar Storage Type**, configure **Jar Storage HDFS URL**. This specifies the HDFS location where you want the jars to be stored. For example, `hdfs://<NN_HOST:8020>/hdfs/registry`.
4. Configure **jar.storage** to the directory where you want to store `.jar` files for serializers and deserializers.
5. Configure the **REGISTRY STORAGE** configurations based on the database you created to use as the Schema Registry metadata store.
6. Ensure the registry storage connector URL has the fully qualified host name of where the database was installed and the connector url and default port for the database selected.

Example

```
MYSQL example:
```

```
jdbc:mysql://FQDN_MYSQL:3306/registry
```

```
Postgres Example:
```

```
jdbc:postgresql://FQDN_POSTGRES:5432/registry
```

More Information

[Installing Databases \[21\]](#)

7.2. Configure SAM

About This Task

When you configure SAM, you need to provide information about the metadata store database, configure a connection with Schema Registry, and establish the URL for Druid's Supersets

Steps

1. In the **Customize Services** step, navigate to the **STREAMLINE CONFIG** section of the **Streaming Analytics Manager** tab.
2. Select **Jar Storage Type**. If you plan to enable HA for SAM on this cluster, you must select **HDFS**.
3. If you selected **HDFS** as the **Jar Storage Type**, configure **Jar Storage HDFS URL**. This specifies the HDFS location where you want the jars to be stored. For example, `hdfs://<NN_HOST:8020:/hdfs/registry`.
4. Configure **jar.storage** to the directory where you want to store `.jar` files for custom processors.
5. Set the **streamline.dashboard.url** to the Superset URL which you can access using **Quick Links** for Druid.
6. Configure **registry.url** to the REST API Endpoint URL for the Registry. The format should be `http://$FQDN_REGISTRY_HOST:$REGISTRY_PORT/api/v1`, where:
 - `$FQDN_REGISTRY_HOST` – Specifies the host on which you are running Schema Registry
 - `$REGISTRY_PORT` – Specifies the Schema Registry port number. You can find the Schema Registry port in the **REGISTRY_CONFIG** section of the **Registry** tab.

For example: `http://FQDN_REGISTRY_HOST:7788/api/v1`

7. Configure the **STREAMLINE STORAGE** configurations based on the database you created to use as a SAM metadata store.
8. Ensure the registry storage connector URL has the fully qualified host name of where the database was installed and the connector url and default port for the database selected.

Example

```
MYSQL example:
```

```
jdbc:mysql://FQDN_MYSQL:3306/streamline
```

```
Postgres Example:
```

```
jdbc:postgresql://FQDN_POSTGRES:5432/streamline
```

More Information

[Installing Databases \[21\]](#)

7.3. Configure NiFi

About This Task

The **NiFi** tab in the **Customize Services** step to configure NiFi. Generally, you can accept the defaults during initial installation. However, there are some configurations that you must set before proceeding.

Steps

1. From **Advanced-nifi-ambari-config**, specify the **Encrypt Configuration Master Key Passwords**. This password is used to generate the master key for sensitive properties encryption in the NiFi properties file when it is written to disk. It must be at least 12 characters.
2. From **Advanced-nifi-ambari-config** provide the **Sensitive property values encryption password**. This is the password used to encrypt any sensitive property values that are configured in processors. It is recommend that it be at least 10 characters.

7.4. Configure Kafka

About This Task

You can configure Kafka from the **Kafka** tab in the **Customize Services** step.

Steps

1. For your initial installation, accept the default values set by Ambari.
2. If Ambari prompts you with:
 - Some configurations need your attention before you can proceed.
 - Review the list of properties and provide the required information.
3. Review the *Apache Kafka Component Guide* for information about configuring storm to meet your operational objectives.

More Information

[Configuring Kafka for Production Environments](#)

7.5. Configure Storm

About This Task

You can configure Storm from the **Storm** tab in the **Customize Services** step.

Steps

1. For your initial installation, accept the default values set by Ambari.
2. If Ambari prompts you with:

Some configurations need your attention before you can proceed.

Review the list of properties and provide the required information.
3. Review the *Apache Storm Component Guide* for information about configuring storm to meet your operational objectives.

More Information

[Configuring Storm for Production Environments](#)

7.6. Deploy the Cluster Services

Finish the wizard and deploy the cluster. After the cluster has been deployed, some services might fail in starting. If this is the case, start those services individually.

7.7. Access the UI for Deployed Services

About This Task

Once you have deployed your Ambari-managed cluster, you can launch the UI for any of the services from Ambari.

Steps

1. From Ambari's left-hand **Services** pane, click the service you want.
2. From the **Quick Links** drop-down, select the UI option.
3. Find links for the SAM UI under **Streaming Analytics Manager** and for the Schema Registry UI under **Registry**.

Result

The UI for your HDF service opens in a new window.

8. Configuring Schema Registry and SAM for High Availability

About This Task

You can configure Schema Registry and SAM for high availability.

Steps for Configuring SAM for HA

1. Install two or more instances of SAM on unique nodes.
2. From the **Services** pane, select **Streaming Analytics Manager** and click the **Configs** tab.
3. In the **Jar Storage Type** drop down, select **HDFS**.

Steps for Configuring Schema Registry for HA

1. Install two or more instances of Schema Registry on unique nodes.
2. From the **Services** pane, select **Schema Registry** and click the **Configs** tab.
3. In the **Jar Storage Type** drop down, select **HDFS**.

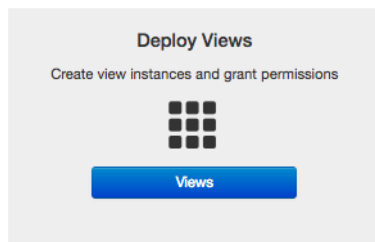
9. Install the Storm Ambari View

About This Task

The Storm Ambari view provides you a number of different troubleshooting and debugging tools.

Steps

1. From the **admin** drop-down, select **Manage Ambari**.
2. Click the **Views** button.



3. From the list of available Views, expand **Storm_Monitoring** and click **+ Create Instance**.



4. Configure the Storm Ambari View.

Views / Create Instance

View **Storm_Monitoring**

Version

Details

Instance Name*

Display Name*

Description*

Visible

Settings

Storm Hostname*

Storm Port*

SSL Enabled*

- Instance Name and Display Name may not have an spaces.
- The Storm Hostname refers to the host where the Storm UI Server is deployed.
- The Storm port is the Storm UI port server (keep it as default 8744 if you have not changed it).
- Click **Save**.

Result

After saving it, you should see a menu item for the Storm Ambari View.

Host	Slots	CPU	Memory	Uptime
172.26.246.18	0%	0%	0%	1h 41m 33s
vett-hdf-sam8.field.hortonworks.com	0%	0%	0%	1h 41m 33s
vett-hdf-sam9.field.hortonworks.com	0%	0%	0%	1h 41m 24s

10. Using a Local Repository

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

- Obtain Public Repositories from the [HDF Release Notes](#)
- Set up a local repository having:
 - [Setting Up a Local Repository with No Internet Access \[42\]](#)
 - [Setting up a Local Repository With Temporary Internet Access \[44\]](#)
- [Preparing The Ambari Repository Configuration File \[46\]](#)

10.1. 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 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 \[41\]](#)
- [Setting Up a Local Repository with No Internet Access \[42\]](#)
- [Setting up a Local Repository With Temporary Internet Access \[44\]](#)

10.2. Getting Started Setting Up a Local Repository

Before setting up your local repository, complete the following:

Prerequisites

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

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

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

To begin setting 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 this web server.
 - c. Ensure that any firewall settings allow inbound HTTP access from your cluster nodes to your mirror server.



Note

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

2. On your mirror server, create a directory for your web server.
 - For example, from a shell window, type:
 - For RHEL/CentOS/Oracle Linux:** `mkdir -p /var/www/html/`
 - For SLES:** `mkdir -p /srv/www/htdocs/rpms`
 - For Debian/Ubuntu:** `mkdir -p /var/www/html/`
 - If you are using a symlink, enable the `followsymlinks` on your web server.

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

10.2.1. Setting Up a Local Repository with No Internet Access

Prerequisites

Complete the [Getting Started Setting up a Local Repository](#) procedure.

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 the archive.
 - a. Browse to the web server directory you created.

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

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

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

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

Untar Locations for a Local Repository - No Internet Access

Ambari Repository	Untar under <code><web.server.directory></code>
HDF Stack Repositories	Create a directory and untar it under <code><web.server.directory>/hdf</code>
HDP Stack Repositories	Create directory and untar under <code><web.server.directory>/hdp</code>

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

URLs for a Local Repository - No Internet Access

Ambari Base URL	<code>http://<web.server>/Ambari-2.5.1.0/<OS></code>
HDF Base URL	<code>http://<web.server>/hdf/HDF/<OS>/3.x/updates/<latest.version></code>
HDP Base URL	<code>http://<web.server>/hdp/HDP/<OS>/2.x/updates/<latest.version></code>
HDP-UTILS Base URL	<code>http://<web.server>/hdp/HDP-UTILS-<version>/repos/<OS></code>

Where:

- `<web.server>` – The FQDN of the web server host
- `<version>` – The Hortonworks stack version number
- `<OS>` – centos6, centos7, sles11, sles12, ubuntu14, ubuntu16, or debian7



Important

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

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

a. Install the plug-in.

For RHEL and CentOS 7: `yum install yum-plugin-priorities`

For RHEL and CentOS 6: `yum install yum-plugin-priorities`

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

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

10.2.2. Setting up a Local Repository With Temporary Internet Access

Prerequisites

Complete the [Getting Started Setting up a Local Repository](#) procedure.

Steps

1. Put the repository configuration files for Ambari and the Stack in place on the host.
2. Confirm availability of the repositories.

For RHEL/CentOS/Oracle Linux: `yum repolist`

For SLES: `zypper repos`

For Debian/Ubuntu: `dpkg-list`

3. Synchronize the repository contents to your mirror server.

- Browse to the web server directory:

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

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

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

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

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

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

```
reposync -r Updates-Ambari-2.5.1.0
```

where `<OS>` is `centos6`, `centos7`, `sles11`, `sles12`, `ubuntu14`, `ubuntu16`, or `debian7`.

- For HDP Stack Repositories, create `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-2.5.1.0
```

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.

URLs for the New Repository

Ambari Base URL <http://<web.server>/ambari/<OS>/Updates-Ambari-2.5.1.0>

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> – centos6, centos7, sles11, sles12, ubuntu14, ubuntu16, or debian7



Important

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

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

- a. Install the plug-in.

For RHEL and CentOS 7: `yum install yum-plugin-priorities`

For RHEL and CentOS 6: `yum install yum-plugin-priorities`

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

```
[main]
```

```
enabled=1
```

```
gpgcheck=0
```

10.3. Preparing The Ambari Repository Configuration File

Steps

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

```
http://public-repo-1.hortonworks.com/ambari/<OS>/2.x/updates/2.5.1.0/ambari.repo
```

where `<OS>` is `centos6`, `centos7`, `sles11`, `sles12`, `ubuntu14`, `ubuntu16`, or `debian7`.

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



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-2.5.1.0]
```

```
name=Ambari-2.5.1.0-Updates
```

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

```
gpgcheck=1
```

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

```
enabled=1
```

```
priority=1
```

Base URL for a Local Repository

Built with Repository Tarball (No Internet Access) `http://<web.server>/Ambari-2.5.1.0/<OS>`

Built with Repository File <http://<web.server>/ambari/<OS>/Updates-Ambari-2.5.1.0>
(Temporary Internet Access)

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

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

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

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

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

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

```
[main]
```

```
enabled=1
```

```
gpgcheck=0
```

11. Where to Go Next?

Use this table to help you navigate the HDF documentation library.

If you want to ...	See this document ...
Install or Upgrade an HDF cluster using Ambari	<ul style="list-style-type: none"> • Release Notes • Support Matrices • Planning Your Deployment • Ambari Upgrade
Manually install or upgrade HDF components. This option is not available for Streaming Analytics Manager and Schema Registry.	<ul style="list-style-type: none"> • Command Line Installation • MiNiFi Java Agent Quick Start • Manual Upgrade
Get Started with HDF	<ul style="list-style-type: none"> • Getting Started with Apache NFi • Getting Started with Stream Analytics
Use and administer HDF Flow Management capabilities	<ul style="list-style-type: none"> • Apache NiFi User Guide • Apache NiFi Administration Guide • Apache NiFi Developer Guide • Apache NiFi Expression Language Guide • MiNiFi Java Agent Administration Guide
Use and administer HDF Stream Analytics capabilities	<ul style="list-style-type: none"> • Streaming Analytics Manager User Guide • Schema Registry User Guide • Apache Storm Component Guide • Apache Kafka Component Guide