Hortonworks DataFlow

Installing an HDF Cluster

Hortonworks DataFlow: Installing an HDF Cluster

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

Table of Contents

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. Set Up the Ambari Server [7]

4. Start the Ambari Server [8]

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. Reviewing System Requirements

Your first task in installing Ambari is to review the Hortonworks DataFlow (HDF) support matrices for system requirements, supported operating systems, component interoperability, and similar information.

• HDF Support Matrices

1.1.2. 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

1.1.3. 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

Defining Service Users and Groups for a HDP 2.x Stack

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:

```
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
```

1.1.5. 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:

```
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

1.1.7. 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,

```
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
```

1.2. Download the Ambari Repository

Ambari 2.6.0.0 Repositories

OS	Format	URL
RedHat 7	Base URL	http://public-repo-1.hortonworks.com/ambari/centos7-ppc/2.x/updates/2.6.0.5-1
	Repo File	http://public-repo-1.hortonworks.com/ambari/centos7-ppc/2.x/updates/2.6.0.5-1/ambari.repo
	Tarball	http://public-repo-1.hortonworks.com/ambari/centos7-ppc/2.x/updates/2.6.0.5-1/ambari-2.6.0.5-1-centos7-ppc.tar.gz

Use a command line editor to perform each instruction.

1.2.1. 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 http://public-repo-1.hortonworks.com/ambari/centos7-ppc/2.x/
updates/2.6.0.5-1/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.2.6.0.0-2.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-2.6.0.0-7.ppc64le               4/4
Verifying  : postgresql-server-9.2.15-1.e17_2.ppc64le      1/4
Verifying  : ambari-server-2.6.0.0-7.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  0:2.6.0.0-7

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 :
> http://s3.amazonaws.com/dev.hortonworks.com/ambari/centos7-ppc/
> RPM-GPG-KEY/RPM-GPG-KEY-Jenkins
> ```

> **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.

1.3. Set Up the Ambari Server

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

```
ambari-server setup
```

command manages the setup process.

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/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.

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

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

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

10 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

1.4. Start the Ambari Server

- Run the following command on the Ambari Server host:

```
ambari-server start
```

- To check the Ambari Server processes:

```
ambari-server status
```

- To stop the Ambari Server:

```
ambari-server stop
```

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

2. Installing Databases

Schema Registry, SAM, Druid, and Superset require a relational data store to store metadata. You can use MySQL for this. This chapter describes how to install MySQL and how create a databases for SAM and Registry. If you are installing on an existing HDP cluster by using Superset, you can skip the installation instructions, because MySQL was installed with Druid. In this case, configure the databases.

Installing and Configuring MySQL

- Installing MySQL [10]

- Configuring SAM and Schema Registry Metadata Stores in MySQL [11]

- Configuring Druid and Superset Metadata Stores in MySQL [11]

2.1. Installing MySQL

About This Task

You can install MySQL 5.5 or later.

Before You Begin

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

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

Steps

1. Log in to the node on which you want to install the MySQL metastore to use for SAM, Schema Registry, and Druid.

2. Install MySQL and the MySQL community server, and start the MySQL service:

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

   yum install mysql-community-server

   systemctl start mysqld.service
   ```

3. Obtain 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. Configuring 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. Configuring Druid and Superset Metadata Stores in MySQL

About This Task

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

Steps

1. Launch the MySQL monitor:

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

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

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

3. Create `druid` and `superset` user accounts, replacing the 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;
```

3. 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.

Warning

If you are installing an HDF cluster only, begin the installation with a fresh Ambari instance. Do not install the HDF management pack on a system where HDP is already installed.

Steps

1. Back up your Ambari resources folder:

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

2. Download the Hortonworks HDF management pack. You can find the download location for your operating system in the *HDF Release Notes*.

3. Copy the bundle to `/tmp` on the node where you installed Ambari.

4. Install the management pack:

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

5. Restart the Ambari server:

   ```
   ambari-server restart
   ```

More Information

[HDF Release Notes](#)

4. Install an HDF Cluster Using Ambari

About This Task

After you start the Ambari service, you can open Ambari web in a browser and launch the Install wizard to prepare for installing an HDF cluster.

Steps

1. Navigate to http://<your.ambari.server>:8080, where <your.ambari.server> is the name of your Ambari server host.

2. Log in to the Ambari server by using the default user name and password: admin and admin. You can change these credentials later.

3. In the **Ambari Welcome** page, choose **Launch Install Wizard**.

4. In the **Get Started** step, specify a name for your cluster.

5. In the **Select Version** page, remove all repositories except the one appropriate for your operating system. Change the Base URL for HDF to the base URL appropriate for your operating system. Find the HDF Base URLs in the HDF Release Notes.

   The wizard page should look similar to the following example:

5. Configure HDF Components

You can customize your Hortonworks DataFlow (HDF) component configurations either during or after installation. During installation, you customize HDF component configurations in the **Customize Services** page of the installation wizard. After installation, you can navigate to Services > Configs in the Ambari dashboard.

5.1. 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 want 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** and then the storage type that you want to use.

   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 in which 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 that the registry storage connector URL has the fully qualified name of the host on which 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 [10]

5.2. Configure SAM

About This Task

When you configure Hortonworks Streaming Analytics Manager (SAM), you must provide information about the metadata store database, configure a connection with Schema Registry, and establish the URL for the Druid 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 on which 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 and

   • $REGISTRY_PORT specifies the Schema Registry port number, as in the following example:

   http://FQDN_REGISTRY_HOST:7788/api/v1

   You can find the Schema Registry port in the **REGISTRY_CONFIG** section of the **Registry** tab.

7. Configure the **STREAMLINE STORAGE** configurations based on the database you created to use as a SAM metadata store.

8. Ensure that the registry storage connector URL has the fully qualified name of the host on which 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 \[10\]](#)

5.3. Configure NiFi

About This Task

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

Steps

1. From **Advanced-nifi-ambari-config**, specify the **Encrypt Configuration Master Key Passwords**.

   This password is used when you generate the master key for sensitive properties encryption in the NiFi properties file when it is written to disk. It must contain at least 12 characters.

2. From **Advanced-nifi-ambari-config**, provide the **Sensitive property values encryption password**.

   This is the password used when you encrypt any sensitive property values that are configured in processors. For enhanced security, it should contain at least 10 characters.

5.4. Configure Kafka

About This Task

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

Steps

1. For your initial installation, accept the default values set by Apache 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 Apache Storm to meet your operational objectives.

More Information

Configuring Kafka for Production Environments

5.5. 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

5.6. Deploy the Cluster Services

After you finish the wizard and deploy the cluster, some services might fail to start. If this is the case, you can start those services individually by launching them from the Ambari dashboard Services pane.

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.

5.7. 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.

6. 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**.

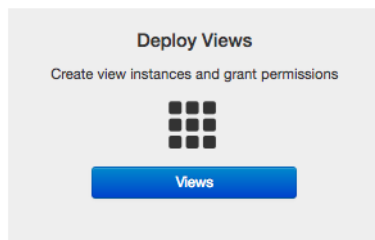# 7. Install the Storm Ambari View

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**.
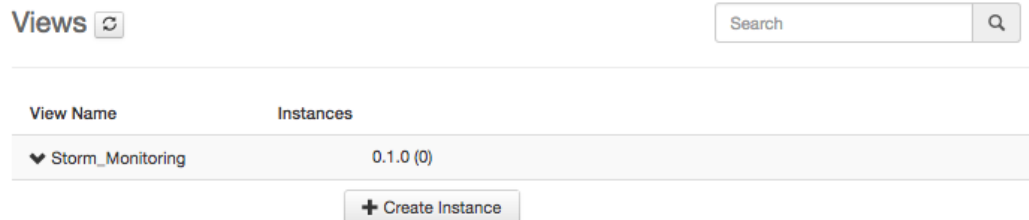
4. Configure the Storm Ambari View.

a. **Instance Name** and **Display Name** may not have an spaces.

b. The **Storm Hostname** refers to the host where the Storm UI Server is deployed.

c. The Storm port is the Storm UI port server (keep it as default 8744 if you have not changed it).

d. Click **Save**.

Result

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

8. 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 [23]

- Set up a local repository having:

    - Setting Up a Local Repository with No Internet Access [25]

    - Setting up a Local Repository With Temporary Internet Access [26]

- Preparing The Ambari Repository Configuration File [28]

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

8.1. Obtaining the Repositories

This section describes how to obtain:

- Ambari Repositories [23]

- HDP Stack Repositories [23]

8.1.1. Ambari Repositories

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

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

Ambari 2.6 Repositories

OS	Format	URL
RedHat 7	Base URL	http://public-repo-1.hortonworks.com/ambari/centos7-ppc/2.x/updates/2.6.0.0
	Repo File	http://public-repo-1.hortonworks.com/ambari/centos7-ppc/2.x/updates/2.6.0.0/ambari.repo
	Tarball md5 | asc	http://public-repo-1.hortonworks.com/ambari/centos7-ppc/2.x/updates/2.6.0.0/ambari-2.6.0.0-centos7-ppc.tar.gz

8.1.2. HDP Stack Repositories

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

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

• HDP 2.6 Repositories [24]

8.1.2.1. HDP 2.6 Repositories

OS	Version Number	Repository Name	Format	URL
RedHat 7	HDP-2.6.3.0	HDP	Version Definition File (VDF)	http://public-repo-1.hortonworks.com/HDP/centos7-ppc/2.x/updates/2.6.3.0-xxxx.xml
			Base URL	http://public-repo-1.hortonworks.com/HDP/centos7-ppc/2.x/updates/2.6.3.0
			Repo File	http://public-repo-1.hortonworks.com/HDP/centos7-ppc/2.x/updates/2.6.3.0/hdp.repo
			Tarball md5 | asc	http://public-repo-1.hortonworks.com/HDP/centos7-ppc/2.x/updates/2.6.3.0/HDP-2.6.3.0-centos7-ppc-rpm.tar.gz
		HDP-UTILS	Base URL	http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.21/repos/centos7-ppc
			Repo File	http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.21/repos/centos7-ppc/hdp-util.repo
			Tarball md5 | asc	http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.21/repos/centos7-ppc/HDP-UTILS-1.1.0.21-centos7-ppc.tar.gz

8.1.3. 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 [24]

• Setting Up a Local Repository with No Internet Access [25]

• Setting up a Local Repository With Temporary Internet Access [26]

8.1.4. 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

8.1.4.1. 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-2.6.0.0/centos7
HDP Base URL	http://<web.server>/hdp/HDP/centos7/2.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

8.1.4.2. 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-2.6.0.0
   ```

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

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

   ```
   cd hdp/centos7
   ```

   ```
   reposync -a ppc64le -r HDP-2.6.3.0
   ```

   ```
   reposync -a ppc64le -r HDP-UTILS-1.1.0.21
   ```

4. Generate the repository metadata.

   • For Ambari:

   ```
   createrepo <web.server.directory>/ambari/centos7/Updates-Ambari-2.6.0.0
   ```

   • 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-2.6.0.0
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

8.1.5. Preparing The Ambari Repository Configuration File

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

   ```
   http://public-repo-1.hortonworks.com/ambari/centos7-ppc/2.x/updates/2.6.0.0/
   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-2.6.0.0]
   ```

   ```
   name=Ambari-2.6.0.0-Updates
   ```

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

   ```
   gpgcheck=1
   ```

   ```
   gpgkey=http://public-repo-1.hortonworks.com/ambari/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-2.6.0.0/centos7
Built with Repository File (Temporary Internet Access)	http://<web.server>/ambari/centos7/Updates-Ambari-2.6.0.0

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
```

More Information

Setting Up a Local Repository with No Internet Access [25]

Setting up a Local Repository With Temporary Internet Access [26]

9. Navigating the HDF Library

To navigate the Hortonworks DataFlow (HDF) documentation library, begin by deciding your current goal.

If you want to...	See this document...
Install or upgrade an HDF cluster using Apache Ambari	• Release Notes • Support Matrices • Planning Your Deployment
Get started with HDF	• Getting Started with Apache NFi • Getting Started with Stream Analytics
Use and administer HDF Flow Management capabilities	• Apache NiFi User Guide • Apache NiFi Administration Guide • Apache NiFi Developer Guide • Apache NiFi Expression Language Guide
Use and administer HDF Stream Analytics capabilities	• Streaming Analytics Manager User Guide • Schema Registry User Guide • Apache Storm Component Guide • Apache Kafka Component Guide