

ambari administration 2

Administering Ambari

Date of Publish: 2018-04-30



<http://docs.hortonworks.com>

Contents

Introducing Ambari administration.....	5
Understanding Ambari terminology.....	5
Accessing Cloudera's private repositories with Ambari.....	6
Setting up a local repository.....	6
Case study for setting up a local repository.....	7
Case study for setting up an HDP-GPL local repository.....	8
Using the Administrator role in Ambari Web.....	8
Log In to Ambari as administrator.....	8
Using the Ambari Admin page.....	9
Create a cluster.....	10
Using Ambari Blueprints.....	10
Export an Ambari Blueprint.....	11
Change the admin password.....	11
Changing your JDK.....	11
Move the ZooKeeper server.....	13
Rename a cluster.....	14
Register a remote cluster.....	14
Setting up Ambari to use an Internet proxy server.....	15
Configure Ambari server to use a proxy server.....	16
Configure yum to use Internet proxy settings.....	16
Managing cluster roles.....	16
Understanding cluster roles and access.....	17
Role based access control.....	17
Modify access levels for users and groups.....	20
Managing versions.....	21
Register a new version.....	21
Update version repository base urls.....	22
De-register a version.....	23
Managing local users.....	23
Manage privileges for local and ldap users.....	23
Create a local user.....	24
Set user status.....	24
Grant Ambari admin privileges.....	24

Change the password for a local user.....	25
Delete a local user.....	25
Enable user home directory creation.....	25
Managing local group membership.....	26
Understanding group types.....	26
Modify group membership.....	26
Create a local group.....	26
Delete a local group.....	27
Installing Ambari agents manually.....	27
Download the Ambari repository on RHEL-CentOS-Oracle Linux 7.....	27
Install the Ambari agents manually on RHEL-CentOS-Oracle 7.....	28
Download the Ambari repository on Amazon Linux 2.....	28
Install the Ambari agents manually on Amazon Linux 2.....	29
Download the Ambari repository on SLES 11.....	30
Install the Ambari agents manually on SLES 11.....	30
Download the Ambari repository on SLES 12.....	31
Install the Ambari agents manually on SLES 12.....	31
Download the Ambari repository on Ubuntu 14.....	32
Install the Ambari agents manually on Ubuntu 14.....	32
Download the Ambari repository on Ubuntu 16.....	33
Install the Ambari agents manually on Ubuntu 16.....	33
Download the Ambari repository on Debian 9.....	33
Install the Ambari agents manually on Debian 9.....	34
Understanding service users and groups.....	34
Default user accounts.....	35
Default group accounts.....	36
Setting properties that depend on service or group user names.....	36
Understanding custom and private host names.....	37
Using custom and private host names.....	37
Configure a public host name.....	38
Configure a custom host name.....	38
Public host name limitations.....	39
Verifying public host name configuration.....	39
Changing host names.....	39
Moving the Ambari server.....	40
Back up current Ambari database.....	41
Update all Ambari agents.....	41
Install the new Ambari server.....	41
Populate the new Ambari database.....	42
Start the new Ambari server and agents.....	42
Configuring LZO compression.....	42
Enable LZO compression.....	43
Configure core-site.xml for LZO.....	43
Optional - Enable LZO using Ambari Blueprints.....	44

Disable automatic LZO library download and installation.....	44
Manually installing LZO libraries.....	45
Manually installing LZO on RHEL-CentOS-Oracle.....	45
Manually Installing LZO on SUSE Linux.....	45
Manually installing LZO on Ubuntu or Debian.....	46
Using LZO compression with Hive queries.....	46
Create LZO files.....	46
Write custom Java to create LZO files.....	46
Using an existing or installing a default database.....	47
Using an existing database with Ambari.....	47
Using Ambari with Oracle.....	47
Using Ambari with MySQL or MariaDB.....	48
Using Ambari with PostgreSQL.....	49
Using a new or existing database with Hive.....	50
Using Hive with Oracle.....	50
Using Hive with MySQL.....	51
Using Hive with PostgreSQL.....	51
Using an existing database with Oozie.....	52
Using Oozie with Oracle.....	52
Using Oozie with MySQL.....	53
Using Oozie with PostgreSQL.....	53
Example: Install MariaDB for use with multiple components.....	54
Configuring network port numbers.....	55
Default network port numbers for Ambari.....	55
Change the default Ambari server port.....	56
Change the default Ambari server-agent port.....	56
Tuning Ambari performance.....	57
Adjust Ambari server heap size.....	57
Increase Ambari server cache size.....	58
Adjust jdbc connection pool settings.....	58
Increase MySQL wait_timeout and interactive_timeout settings.....	58
Purge Ambari server database history.....	59
Optimize Ambari agent performance.....	60
Customizing Ambari log and pid directories.....	61
Finding Ambari log files.....	61
Configure Ambari logging level.....	61
Customizing Ambari agent log and pid directories.....	61
Managing host participation for HDFS and YARN.....	62
Ambari-managed host participation.....	62
Enable manage.include.files for HDFS.....	63
Enable manage.include.files for Yarn.....	63
Disable manage.include.files for HDFS.....	64
Disable manage.include.files for Yarn.....	64

Introducing Ambari administration

Apache Ambari enables you to provision, manage, and monitor your Hadoop cluster.

Installing Ambari creates the default user admin/admin. This Ambari-level administrator user, or Ambari Admin, has full control over all aspects of Ambari, including all clusters managed by the Ambari instance, as well as the abilities to create a cluster, and manage users, groups, and clusters. When you log in to Ambari as Ambari Admin, you can perform all tasks that require permissions of the Ambari Administrator role.

Related Information

[Installing, Configuring, and Deploying a Cluster](#)

Understanding Ambari terminology

Familiarity with the following basic terms can help you to understand the key concepts associated with Ambari administration:

Ambari Admin	Specific privileges granted to a user that enables that user to administer Ambari. Users with the Ambari Admin privilege can grant this privilege to other users, or revoke it from them.
Ambari Admin Page	Ambari Web page accessible only to users with the Ambari Admin privilege.
Ambari Web	Graphical User Interface (GUI) that provides user access to Ambari-managed cluster resources.
account	User name, password, and privileges.
cluster	An installation of a Hadoop cluster, based on a particular stack, that is managed by Ambari.
group	Unique group of users in Ambari.
group type	Local and LDAP. Local groups are maintained in the Ambari database. LDAP groups are imported to (and synchronized with) an external LDAP, if one is configured.
permissions	The permissions granted to a principal user or group for a particular view.
principal	User or group that can be authenticated by Ambari.
privilege	The mapping of a principal to a permission or role and a resource. For example, the user joe.operator is granted the role of Cluster Operator on the cluster DevCluster.
resource	The resource available and managed in Ambari. Ambari supports two types of resources: cluster and view. An Ambari Admin assigns permissions for a resource for users and groups.
role	The role that is assigned to a principal (user or group) on a particular cluster.
user	Unique user in Ambari.

user type	Local and LDAP. Local users are maintained in the Ambari database and authentication is performed against the Ambari database. LDAP users are imported to (and synchronized with) an external LDAP, if one is configured.
version	Stack version, which includes a set of repositories to install that version on a cluster.
view	A user interface component that is available to Ambari.

Accessing Cloudera's private repositories with Ambari

Ambari 2.7.0.0 will not be fully functional after 31st of January 2021. To continue using all the features of this version of Ambari, a paywalled version is made available to you. To upgrade to that version follow this 2-step process.



Note: As of January 31, 2021, all downloads of HDP and Ambari require a username and password and use a modified URL. You must use the modified URL, including the username and password when downloading the repository content.

Upgrade Ambari to 2.7.0.7

Ambari needs to be upgraded from 2.7.0.0 to 2.7.0.7 so that you can download repository files and binaries behind the paywall and continue using the existing functionality. The instructions are the same as the [Apache Ambari Upgrade section](#), but the repository URLs must be replaced as listed in the Ambari Repositories section. For more information, see [Ambari Repositories](#).

To access the binaries, you must have the required authentication credentials (**username** and **password**).

Authentication credentials for new customers and partners are provided in an email sent from Cloudera to registered support contacts. Existing users can file a non-technical case in the support portal (<https://my.cloudera.com>) to obtain credentials.

After you receive your authentication credentials, use them to form the URL where you can access the Ambari repository in the Ambari archive, as shown below. Insert your username and password in the beginning -of the URL as shown in the following example:

```
https://[username]:[password]@archive.cloudera.com/p/ambari/2.x/2.7.0.7/centos7/ambari.repo
```

Change the HDP repo baseUrl

Along with the Ambari paywalled repositories HDP is also moved behind the paywall. The version definition baseUrl must be updated with the new address that includes the username and password. You can follow this [guide](#) to update the baseUrl. The HDP repository URLs' must replaced as listed in the HDP 3.0.0.0 Repositories section. For more information, see [HDP 3.0.0 Repositories](#).

Setting up a local repository

You need to set up a local repository, update the version repository base URLs, and edit the repository configuration file.

As a first step, you must set up a local repository for Ambari and HDP. For more information, see [set up a local repository](#). In case the public repository (<http://public-repo-1.hortonworks.com>) is not available anymore you can get the files from the official archive: <https://archive.cloudera.com>.

A case-study of setting up local repositories: [Case study for setting up a local repo](#)

When the local repository is created, update the version repository base urls in Ambari. For more information, see [update the version repository base urls](#).

Finally, edit the repository configuration file to use this new local repository. For more information, see [edit the repository configuration file](#)

Updating Ambari repo files

Some services depend on components that are installed from the Ambari repository. It is not updated automatically. Also, Package Manager displays an error about the unavailable repository URL when you update the package lists. Cloudera recommends you to manually update the URLs located at `/etc/yum.repos.d/ambari.repo` on all hosts (including server host).

Updating HDP repo files

When the cluster settings for the HDP repository URL is updated, repository files on hosts are not immediately regenerated. The files are re-generated when you add a new component or service. But an inaccessible repository URL causes the Package Manager to display an error about the unavailable repository URL when you update the package lists. Cloudera recommends you to manually update the URLs in the HDP repository files (for example, `/etc/yum.repos.d/ambari-hdp-1.repo`) on all agent hosts.

Case study for setting up a local repository

Review this case study to understand - how to prepare a local repository for Ambari 2.7.0 and HDP 3.0.0 on Centos7:

Procedure

1. `yum install yum-utils createrepo -y`
2. `yum install httpd -y`
3. //firewall configuration
4. `sudo systemctl start httpd`
5. `sudo systemctl status httpd`
6. `mkdir -p /var/www/html/`
7. `wget -nv https://archive.cloudera.com/p/ambari/2.x/2.7.0.0/centos7/ambari.repo -O /etc/yum.repos.d/ambari.repo`
8. `wget -nv https://archive.cloudera.com/p/HDP/centos7/3.x/updates/3.0.0.0/hdp.repo -O /etc/yum.repos.d/hdp.repo`
9. `yum repolist`
10. `cd /var/www/html`
11. `mkdir -p ambari/centos7`
12. `cd ambari/centos7/`
13. `reposync -r ambari-2.7.0.0`
14. `cd ../../`
15. `mkdir -p hdp/centos7`
16. `cd hdp/centos7/`
17. `reposync -r HDP-3.0.0.0`
18. `reposync -r HDP-UTILS-1.1.0.22`
19. `createrepo /var/www/html/ambari/centos7/ambari-2.7.0.0`
20. `createrepo /var/www/html/hdp/centos7/HDP-3.0.0.0`
21. `createrepo /var/www/html/hdp/centos7/HDP-UTILS-1.1.0.22/`

The repositories will be available at the local web server:

- `http://<web.server>/ambari/centos7/ambari-2.7.0.0/`
- `http://<web.server>/hdp/centos7/HDP-3.0.0.0/`

- <http://<web.server>/hdp/centos7/HDP-UTILS-1.1.0.22/>

Case study for setting up an HDP-GPL local repository

If the cluster is using the GPL components, you must replace the HDP-GPL repository URL with a local repository.

Procedure

1. Set up a local HDP-GPL repository
 - a) `wget -nv https://archive.cloudera.com/p/HDP-GPL/3.x/3.0.0.0/centos7/hdp.gpl.repo -O /etc/yum.repos.d/hdp.gpl.repo`
 - b) `mkdir -p /var/www/html/hdp-gpl/centos7`
 - c) `cd /var/www/html/hdp-gpl/centos7/`
 - d) `reposync -r HDP-GPL-3.0.0.0`
 - e) `createrepo /var/www/html/hdp-gpl/centos7/HDP-GPL-3.0.0.0/`
2. Edit `/etc/ambari-server/conf/ambari.properties`
 - a) Replace `gpl.license.accepted=false` with `gpl.license.accepted=true`
3. Restart Ambari server
4. You must edit the HDP-GPL repository similar to the HDP repository. However, the URL and Local repository contents, and the UI fields are different for both the repositories. For more information on updating the version repository, see [Update version repository base urls](#).
5. If you do not plan use the GPL components, disable the `gpl.license.accepted` property
6. Restart Ambari server

Using the Administrator role in Ambari Web

An Ambari-level Administrator can access the Ambari Admin page using the default, `admin/admin` credentials. Use the Ambari Admin page to perform the following tasks.

Related Information

[Managing Cluster Roles](#)

Log In to Ambari as administrator

After installing Ambari, you can log in to Ambari as administrator using the default credentials: `username/password = admin/admin`

About this task

An Ambari administrator accesses Ambari Server, using a web browser to display the Ambari Admin web page.

Procedure

1. To access the Ambari Admin page, enter the following URL in a web browser:
`http://[YOUR_AMBARI_SERVER_FQDN]:8080`
[YOUR_AMBARI_SERVER_FQDN] is the fully qualified domain name of your Ambari Server host and 8080 is the default HTTP port.
2. Enter your credentials.
`username/password = admin/admin`

Results

The Ambari Admin page displays.

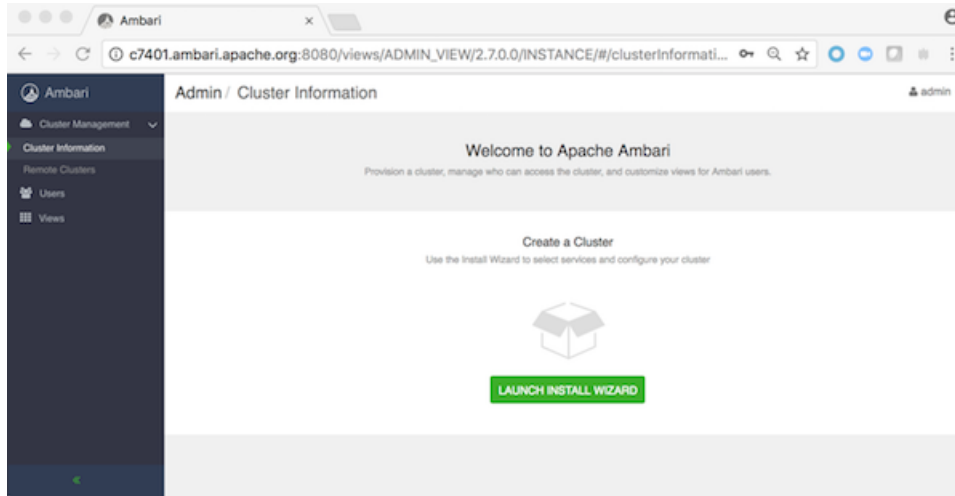
What to do next

If necessary, change the Admin password.

Using the Ambari Admin page

Use the Ambari Admin page to create new clusters, register remote clusters, deploy views, and manage users and groups.

When you first log in to Ambari as an administrator, **Ambari Admin > Cluster Management > Cluster Information** page displays the **Launch Install Wizard** option. Click **Launch Install Wizard** to deploy an Ambari-managed, cluster.

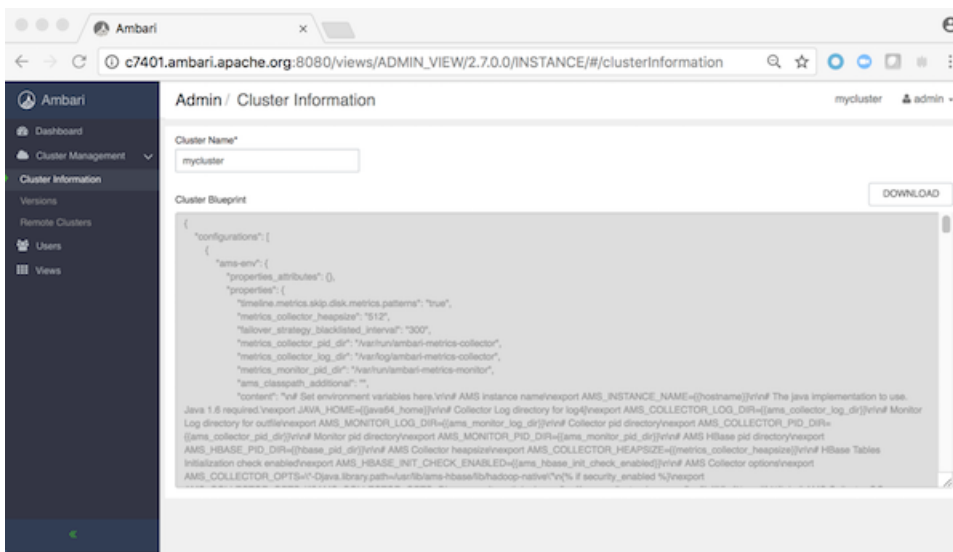


Other **Ambari Admin** options include:

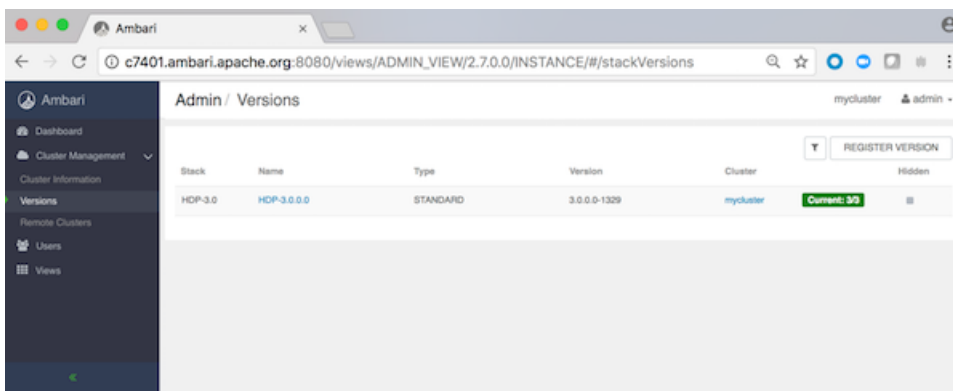
- **Dashboard** will provide access to **Ambari Admin** operations options.
- **Cluster Information** initially provides the option to install a cluster.
- **Remote Clusters** provides the option to register an existing (remote) cluster with this instance of Ambari server.
- **Users** provides options for you to create and edit users and group those users.
- **Views** provides options to create and configure views and to manage access permissions for view instances.

After you create a cluster, the **Dashboard** option displays. Use the **Dashboard** option to operate an Ambari-managed cluster. From **Dashboard**, you can manage and monitor cluster services, including managing the service life cycle, changing configurations, reviewing alerts, and so on.

After you create a cluster, the **Ambari Admin > Cluster Management > Cluster Information** page displays the cluster name and blueprint information for the cluster you just created. You can export (download) the Ambari blueprint.



Versions displays current component version information, and the option to register versions stored in specific repositories.



Create a cluster

After you have successfully installed Ambari, you can create a cluster by using the Ambari cluster install wizard.

Procedure

- In the **Ambari Admin** page, click **LAUNCH INSTALL WIZARD**.

What to do next

Complete all steps in the cluster install wizard.

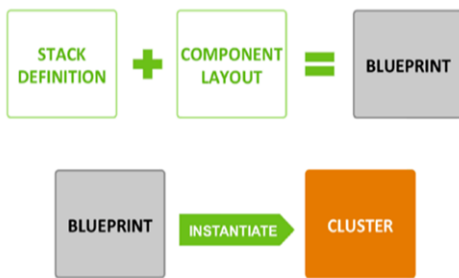
Related Information

[Installing, Configuring, and Deploying a Cluster](#)

Using Ambari Blueprints

Ambari Blueprints provide an API to perform cluster installations.

You can build a reusable "blueprint" that defines which Stack to use, how Service Components should be laid out across a cluster and what configurations to set.



After setting up a blueprint, you can call the API to instantiate the cluster by providing the list of hosts to use. The Ambari Blueprint framework promotes reusability and facilitates automating cluster installations without UI interaction.

Related Information

[Ambari Wiki - Blueprints](#)

Export an Ambari Blueprint

You can export a cluster definition as a JSON file for reuse.

About this task

An Ambari Blueprint defines the Stack version and service components for a cluster, saved as JSON format file. An Ambari administrator can export a blueprint during cluster installation, or after the cluster has been deployed.

Procedure

- During cluster installation, in the Ambari Installation wizard **Review** dialog, click **Generate Blueprint**.
- After a cluster has been deployed:
 - a) From the Ambari Dashboard user menu, click **Manage Ambari**.
 - b) In **Admin > Cluster Information**, click **Download**.

A file named `blueprint.json` exports to the default directory for file downloads on your local machine.

Change the admin password

Using the Ambari Admin page, you can change the password for the default admin user to create a unique administrator credential for your system.

Procedure

1. In **Admin/Users**, for a listed Ambari Administrator user, click **Actions > Edit**.
2. In **Users/[USERNAME]**, click **CHANGE PASSWORD**.
3. Click **Change Password**.
4. In **Change Password for [USERNAME]**, type the current password and then your new password, twice.
5. Click **OK**.

Changing your JDK

During your initial Ambari Server Setup, you select the JDK to use or provide a path to a custom JDK already installed on your hosts. After setting up your cluster, you may change the JDK.

About this task

Hortonworks HDP 2.6.x and 3.x platforms support both Oracle JDK 8 and Open JDK 8 versions. HDP does not bundle either JDK libraries but certify HDP components using both Oracle and Open JDK libraries. Oracle announced Oracle JDK 11 release in September 2018 that requires support subscription. Please note that Oracle will not post further updates of Java SE 8 to its public download sites for commercial use after January 2019. Customers who need continued access to critical bug fixes and security fixes as well as general maintenance for Java SE 8 or previous versions can get long term support through Oracle Java SE Subscription or Oracle Java SE Desktop Subscription. For further details on Oracle JDK, please see the Oracle Java SE Support Roadmap.

To change the JDK implementation for an existing cluster:

Procedure

1. On the Ambari Server host, re-run Ambari Server Setup.
ambari-server setup
2. At the prompt: change the JDK ?, Enter y.
3. At the prompt: Do you want to change Oracle JDK [y/n] (n)?, Enter y.
4. At the prompt:choose a JDK:, Enter 2 to change the JDK implementation.

Option	Description
[1] - Oracle JDK 1.8 + Java Cryptography Extension (JCE) Policy Files 8	If you choose Oracle JDK 1.8, the JDK you choose downloads and installs automatically on the Ambari Server host. This option requires that you have an internet connection as the Ambari Server will download and install this JDK on all hosts in the cluster.
[2] - Custom JDK	If you choose Custom JDK, verify or add the custom JDK path on all hosts in the cluster. Use this option if you want to use OpenJDK or do not have an internet connection (and have pre-installed the JDK on all hosts).



Important: If you use a custom JDK, AND if kerberos is enabled with AES-256 encryption. you must also update your JCE security policy files on the Ambari Server and all hosts in the cluster to match the new JDK version. If you are running Kerberos and do not update the JCE to match the JDK, you will have issues starting services.

What to do next

After setup completes, you must restart each component for the new JDK to be used by the Hadoop services. Using the Ambari Web UI, do the following tasks:

1. Restart Ambari Server.
ambari-server restart
2. Log in to the Ambari Server and restart all cluster services.



Important:

If, after changing the JDK, you experience issues with communication between Ambari Server and Ambari Agents, refer to Java/Python updates and Ambari Agent TLS settings in Hortonworks Community Connection for more information.

Additionally if customer certificates were imported into the previous JDK's cacerts trust store, they need to be re-imported into the new JDK's cacerts trust store. See Preparing for LDAPS integration for more details.

Related Information

[Hortonworks Support Matrix](#)

[Restart Services](#)

- [Install the JCE for Kerberos](#)
- [Java/Python Updates and Ambari Agent TLS Settings](#)
- [Preparing for LDAPS integration](#)
- [Oracle SE Support Roadmap](#)

Move the ZooKeeper server

To move the ZooKeeper server to a new host:

Procedure

1. In **Ambari Web** > **Services**, click **ZooKeeper**.
2. Click **Actions** > **Stop**.
All ZooKeeper servers stop.
3. In **Ambari Web** > **Hosts**, click the host on which you want to install the new ZooKeeper server.
4. On the **Summary** page of the new ZooKeeper host, click **+Add** > **ZooKeeper Server**.
5. Update the following properties on the new ZooKeeper server (use the existing ZooKeeper server settings as a reference).
 - ha.zookeeper.quorum
 - hbase.zookeeper.quorum
 - templeton.zookeeper.hosts
 - yarn.resourcemanager.zk-address
 - hive.zookeeper.quorum
 - hive.cluster.delegation.token.store.zookeeper.connectString
6. In **Ambari Web** > **Hosts**, click the original ZooKeeper server host.
7. In **Components**, for the ZooKeeper Master component, click **Actions** > **Delete Service** to delete the original ZooKeeper server.
8. Save the HDFS namespace.
9. Restart the new ZooKeeper server and the Hive service.



Note: In Ambari 2.4.0.0, adding or removing ZooKeeper servers requires manually editing the following Atlas properties. Select **Atlas** > **Configs** > **Advanced**, then select **Advanced application-properties** and edit the following properties to reflect the new ZooKeeper server settings:

- atlas.graph.index.search.solr.zookeeper-url
Example format: host1:2181/infra-solr,host2:2181/infra-solr,host3:2181/infra-solr
- atlas.kafka.zookeeper.connect
Example format: host1:2181,host2:2181,host3:2181
- atlas.audit.hbase.zookeeper.quorum
Example format: host1,host2,host3

After updating these properties (to refresh the configuration files), restart Atlas and the following services that contain Atlas hooks :

- Hive
- Storm
- Falcon
- Sqoop

What to do next

Review and confirm all recommended configuration changes.

Related Information

[Review and Confirm Configuration Changes](#)

Rename a cluster

After you create a cluster, you can give it a new name.

Procedure

1. On the Ambari Admin page, in **Cluster Information** > **Cluster Name***, enter up to 80 alphanumeric characters to rename your cluster.

The screenshot shows the 'Admin / Cluster Information' page. The 'Cluster Name*' field contains the text 'mycluster2'. To the right of the field is a 'SAVE' button. A tooltip is displayed over the field, stating: 'Only alpha-numeric characters, up to 80 characters'.



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

2. Click **Save**.
3. Confirm.
4. Optional step(s)
 - a) IF you are running Ranger: Edit each service name to match the new cluster name.

If you do not rename each service to include the new cluster name, Ranger automatically creates a new repository using the new cluster name that contains only the default policies. In this case, to recover your original data, export the customized service policy repository, and then import that repository to overwrite the default policy list generated by Ranger. See [Importing and Exporting Resource-based Policies](#) for more detailed import and export steps.

- b) IF your cluster is Kerberos-enabled: Regenerate keytabs.



Note: Regenerating keytabs will incur cluster downtime.

5. Restart Ambari server and the Ambari agents.

After renaming the cluster, alert checks must be re-queued on the agents. Therefore, you must restart Ambari Server and the Ambari Agents for the change to take effect.
6. Adjust any API calls you make to use the new name.

Changing the name of the cluster changes the name of the Ambari REST API resource for the cluster.

Related Information

[Importing and Exporting Resource-Based Policies](#)

Register a remote cluster

Ambari-managed clusters not local to your Ambari server host must be registered as remote clusters with your Ambari Server.

About this task

You might work with clusters that are managed by Ambari but are not local to your Ambari server. These clusters are considered remote with respect to your local Ambari server. They are managed by a remote Ambari server. If you plan to run a standalone server to host views, including accessing clusters managed by a different Ambari server, you can register the cluster managed by the standalone Ambari server as a remote cluster. After you register a remote cluster, use it to configure view instances.

Procedure

1. In **Ambari Admin > Remote Clusters**, click **Register Remote Cluster**.
2. In **Remote Clusters/Register**, enter a name for the remote cluster, the Ambari cluster URL, and a cluster user name and associated password.
3. Click **Save**.

Results

The remote cluster is now available for configuring View instances.

Related Information

[Run a remote, standalone Ambari Views server](#)

Setting up Ambari to use an Internet proxy server

To access private software repositories using the Internet, Ambari should use a proxy server.

If you plan to use private repositories (repositories available on the Internet) for installing Apache Ambari and deploying Hadoop components in your cluster, you must provide Ambari and the hosts in the cluster Internet access to obtain the software from those repositories. Specifically:

- Ambari Server: uses Internet access to validate the repositories.
- yum (or an equivalent, OS-specific package manager): performs the software installation from the repositories.

Therefore, if your environment requires use of an Internet proxy server for access, you must:

1. Configure Ambari Server to use a proxy server.
2. Configure yum on all cluster hosts to use that proxy server.

Ambari can install software if you have no Internet access. If you have no Internet access (via a proxy server or otherwise), you can use local repositories for installing the cluster software. In that case, configuring Ambari to use a proxy server is not required. However, Ambari and the hosts in the cluster must have access to your local repositories.

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

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

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

Defining a proxy server, username and password in `/etc/yum.conf` means all users of yum connect to the proxy server with those details. Please consult your system administrators and refer to your operating system documentation for more details on this configuration and possible alternatives.

Related Information

<https://docs.centos.org/>

https://docs.oracle.com/cd/E37670_01/E37355/html/ol_proxy_config.html

<https://help.ubuntu.com/community/AptGet/Howto>

[Using a local repository](#)

Configure Ambari server to use a proxy server

You can configure Ambari Server to access the Internet using a proxy server.

Procedure

1. On the Ambari Server host, stop Ambari Server.
`ambari-server stop`
2. Add proxy settings to the following script: `/var/lib/ambari-server/ambari-env.sh`.
`-Dhttp.proxyHost=[YOUR_PROXY_HOST] -Dhttp.proxyPort=[YOUR_PROXY_PORT]`
3. Optionally, to prevent some host names from accessing the proxy server, define the list of excluded hosts.
`-Dhttp.nonProxyHosts=[pipe|separated|list|of|hosts]`
4. If your proxy server requires authentication, add the username and password.
`-Dhttp.proxyUser=[USER_NAME] -Dhttp.proxyPassword=[PASSWORD]`
5. Restart the Ambari Server to pick up this change.
`ambari-server restart`

Configure yum to use Internet proxy settings

You can set up a yum configuration file to use your proxy server.

About this task

Setting up yum to use a proxy server depends a lot on your environment and operating system. These instructions provide some guidance but we strongly recommend you consult with your system administrators and operating system documentation for assistance and specific instructions.

It is important to highlight that defining a proxy server, username and password in `/etc/yum.conf` means all users of yum connect to the proxy server with those details.

Procedure

1. On each host in the cluster, specify the proxy settings in `/etc/yum.conf` by adding an entry such as:
`proxy=http://[YOUR_PROXY_HOST]:[YOUR_PROXY_PORT]`
2. If your proxy server requires authentication, add the username and password, as follows:

```
enableProxyAuth=1
proxy_username=[ USER_NAME ]
proxy_password=[ PASSWORD ]
```

3. Save the yum configuration file.

Managing cluster roles

Ambari-level administrators can assign user and group access to Ambari-, Cluster-, Host-, Service-, and User- (view-only) level permissions.

A granted level of user access is a role. Role-based access control effectively distributes the responsibilities of managing a cluster while not relinquishing total control of the Ambari management facility.

Understanding cluster roles and access

Access levels allow administrators to categorize cluster users and groups based on the permissions that each level includes.

The following roles are based on access-levels. Access levels enhance the granularity of permissions that can be granted to Ambari users and groups:

Cluster User

Users assigned to the Cluster User role can view information about the cluster and its services, including configurations, service status, and health alerts. In Ambari 2.2 and earlier, this user was referred to as the Read-only user. Effectively, the cluster user is a view-only user.

Service Operator

Users assigned to the Service Operator role have control over service life cycles, such as starting and stopping services, performing service checks, and performing service-specific tasks such as rebalancing HDFS and refreshing the YARN Capacity Scheduler.

Service Administrator

Users assigned to the Service Administrator role have the same permissions as users assigned to the Service Operator role but have the added ability to configure services. This includes the ability to manage configuration groups, move service masters, and enable HA.

Cluster Operator

Users assigned to the Cluster Operator role have the same permissions as users assigned to the Service Administrator role but have the added ability to perform host-level tasks such as adding and removing hosts and components.

Cluster Administrator

Users assigned to the Cluster Administrators role have control over the relevant cluster, its hosts, and services. In Ambari 2.2 and earlier, this user was referred to as the Operator user.

Ambari Administrator

Ambari Administrator users have full control over all aspects of Ambari. This includes the ability to create clusters, change cluster names, register new versions of cluster software, and fully control all clusters managed by the Ambari instance.

Role based access control

Permissions that an Ambari-level administrator assigns each user or group define each role.

Use these tables to determine what permissions each role includes. For example: A user with any role can view metrics, but only an Ambari Administrator can create a new Ambari-managed cluster.

Table 1: Service-Level Permissions

Permissions	Cluster User	Service Operator	Service Administrator	Cluster Operator	Cluster Administrator	Ambari Administrator
View metrics						
View status information						
View configurations						
Compare configurations						
View service alerts						
Start, stop, or restart service						
Decommission or recommission						
Run service checks						
Turn maintenance mode on or off						
Perform service-specific tasks						
Modify configurations						
Manage configuration groups						
Move to another host						
Enable HA						
Enable or disable service alerts						
Add service to cluster						










Table 2: Host-Level Permissions

Permissions	Cluster User	Service Operator	Service Administrator	Cluster Operator	Cluster Administrator	Ambari Administrator
View metrics						
View status information						
View configuration						
Turn maintenance mode on or off						
Install components						
Add or delete hosts						

Table 3: Cluster-Level Permissions

Permissions	Cluster User	Service Operator	Service Administrator	Cluster Operator	Cluster Administrator	Ambari Administrator
View metrics						
View status information						
View configuration						
View stack version details						
View alerts						
Enable or disable alerts						
Enable or disable Kerberos						
Upgrade or downgrade stack						

Table 4: Ambari-Level Permissions

Permissions	Cluster User	Service Operator	Service Administrator	Cluster Operator	Cluster Administrator	Ambari Administrator
Create new clusters						
Set service users and groups						
Rename clusters						
Manage users						
Manage groups						
Manage Ambari Views						
Assign permission and roles						
Manage stack versions						
Edit stack repository URLs						

Modify access levels for users and groups

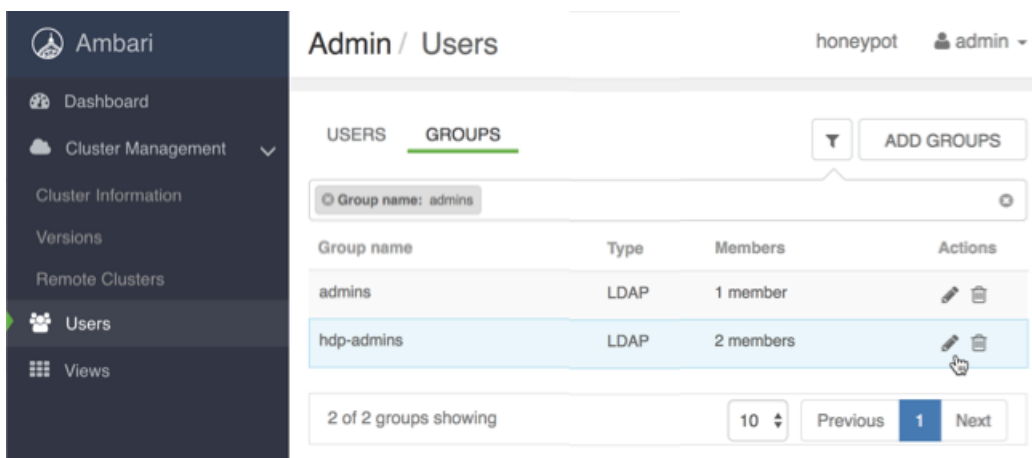
Use **Ambari Admin > Users** to manage access levels for users and groups.

About this task

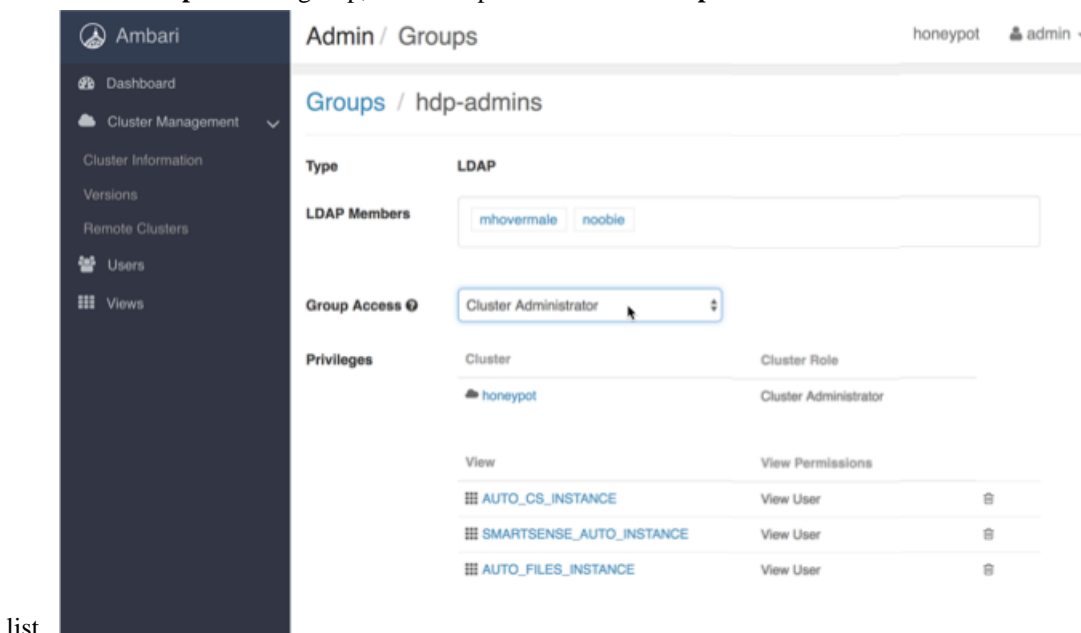
An Ambari Admin can manage the role assignment of local and remote users imported from LDAP. An Ambari administrator can control the access level for any user or group.

Procedure

- In **Ambari Admin > Users**, click **Users** to display the current users known to Ambari.
- To modify access for a user, click **Edit** next to a user name.
- In **Admin/Users**, click an option from the **User Access** list.
- In **Ambari Admin > Users**, click **Groups** to display the current groups known to Ambari.
- To modify access for a group, click **Edit** next to a group name.



- In **Admin/Groups** for that group, click an option from the **Group Access**



list.

Managing versions

Ambari stores multiple versions of components and provides the ability to set one of those versions as current.

Ambari enables you to manage versions of the stack installed in the cluster and registered with Ambari Server. You can register a new version, upgrade version-repository base URLs, and de-register a version. After you install a cluster, Ambari automatically registers the version of the stack software. **Admin/Versions** lists the current stack, name, version, and cluster that is running.

Related Information

[Register and install a target version](#)

Register a new version

Use **Ambari Admin > Versions** to register a new stack version.

Procedure

1. Browse to **Ambari Admin > Versions**.

If necessary, click **Manage Versions**, then **OK**.

2. In **Ambari Admin > Versions** click **Register Version**.

3. Select the software version and method of delivery for your cluster.

Choose	Description
HDP Stack	The available HDP versions are shown in TABs. When you select a TAB, Ambari attempts to discover what specific version of that HDP Stack is available. That list is shown in a DROPDOWN. For that specific version, the available Services are displayed, with their Versions shown in the TABLE.
HDP Version	If Ambari has access to the Internet, the specific Versions will be listed as options in the DROPDOWN. If you have a Version Definition File for a version that is not listed, you can click Add Version... and upload the VDF file. In addition, a Default Version Definition is also included in the list if you do not have Internet access or are not sure which specific version to install. If you choose the Default Version Definition, you must enter the two-digit Version Number in the Name input field.
Repository Delivery Method	Using a private Repository requires Internet connectivity. Using a Local Repository requires you have configured the software in a repository available in your network. To use the private software repositories, see the list of available HDP Repositories for each OS. Or, if you are using a local repository, enter the Base URLs for the local repository you have created.

4. Review **Advanced Options**.

Choice	Description
Skip Repository Base URL validation (Advanced)	Ambari will attempt to connect to the repository Base URLs and validate that you have entered a validate repository. If not, an error will be shown that you must correct before proceeding. This option will skip the Base URL validation.
Use RedHat Satellite/Spacewalk:	This option will only be enabled when you plan to use a Local Repository. When you choose this option for the software repositories, you are responsible for configuring the repository channel in Satellite/Spacewalk. Please refer to the Red Hat Satellite/Spacewalk documentation for more information. Once configured, it is very important that ensure the repositories you confirm for the selected stack version are available on the hosts in the cluster. Ambari will not distribute or use .repo files and will rely on Satellite/Spacewalk as having the repositories configured with the correct stack version.

5. Click **Save**.

What to do next

Update Version Repository Base URLs

Update version repository base urls

Use **Admin/Versions/Repositories** to provide URLs for your source repositories.

Procedure

1. Browse to **Ambari Admin > Versions**.
If necessary, click **Manage Versions**, then **OK**.
2. In **Admin/Versions**, click the version name you want to modify.
3. In **Repositories**, modify the base URLs for the repositories. To use the private software repositories, see the list of available HDP repositories for each OS. Or, if you are using a local repository, enter the Base URLs for the local repository you have created.
4. Click **Save**.
5. Click **Confirm Change**. You must confirm the change since you are about to change repository Base URLs that are already in use. Please confirm that you intend to make this change and that the new Base URLs point to the same exact stack version and build.

De-register a version

Use **Admin/Versions** to de-register a selected stack version.

Procedure

1. Browse to **Ambari Admin > Versions**.
If necessary, click **Manage Versions**, then **OK**.
2. Click the version you want to de-register.
Only versions that are not installed in a cluster can be de-registered.
3. Click **De-register Version** and then confirm.

Managing local users

As an Ambari administrator, you can create and manage users and groups available to Ambari. You can also import user and group information into Ambari from external LDAP systems.

You use the Ambari Admin page to manage both local and LDAP users. Local users are stored in and authenticate against the Ambari database. LDAP users have basic account information stored in the Ambari database. Unlike local users, LDAP users authenticate against an external LDAP system. To use LDAP users with Ambari, you must configure Ambari to authenticate against an external LDAP system. Ambari grants no permissions by default to a new user, created either locally or by synchronizing against LDAP, You, as an Ambari administrator, must explicitly grant each user permissions to access clusters or views.

You, as an Ambari administrator, can create new users, delete users, change user passwords, and edit user settings.

Manage privileges for local and ldap users

You can control certain privileges for local and LDAP users.

The following table lists the privileges available and those not available to the Ambari administrator for local and LDAP Ambari users.

Table 5: Ambari Administrator Privileges for Local and LDAP Users

Ambari Administrator Privilege	Local User	LDAP User
Change password	Available	Not Available
Set Ambari Admin flag	Available	Available

Ambari Administrator Privilege	Local User	LDAP User
Change group membership	Available	Not Available
Delete user	Available	Not Available
Set active or inactive status	Available	Available
Set user access	Available	Available

Related Information

[Configuring Ambari Authentication for LDAP/AD](#)

Create a local user

You, as an Ambari administrator, can create new, local users.

Procedure

1. On the Ambari Admin page, browse to **Users**.
2. Click **Create Local User**.
3. Enter a unique user name.
All user name characters are converted to lowercase.
4. Enter a password, and then confirm that password.
5. Click **Save**.

Set user status

User status determines whether a user can or cannot log in to Ambari.

About this task

User status indicates whether the user is active and allowed to log in to Ambari or is inactive and denied the ability to log in. By setting the status flag as active or inactive, you can effectively disable user account access to Ambari while preserving the user account information related to permissions.

Procedure

1. On the Ambari Admin page, browse to **Users**.
2. Click the name of the user to modify.
3. Click the **Status** control to toggle between Active or Inactive.
4. Click **OK**.

The change is saved immediately.

Grant Ambari admin privileges

Only an Ambari Administrator can grant another user Ambari Administrator privileges.

About this task

You, as an Ambari administrator can grant one or more users Ambari administrator privileges by setting the Ambari Admin flag. Only an Ambari administrator can set or remove the Ambari Admin flag. Ambari prevents you from accidentally removing the flag from your own account.

Procedure

1. On the Ambari Admin page, browse to **Users**.
2. Click the name of the user to modify.
3. Click the Ambari Admin control.
4. Click **Yes** to set or **No** to remove the Ambari Admin flag.

Change the password for a local user

An Ambari administrator can change local user passwords, but not LDAP user passwords.

Procedure

1. On the Ambari Admin page, browse to **Users**.
2. Click the name of the user to modify.
3. Click **Change password**.
4. Enter your administrator password, to confirm that you have required privileges.
5. Enter a password for the local user, and then confirm that password.
6. Click **Save**.

Delete a local user

Deleting a local user removes the user account from the system, including all privileges associated with the user. If you want only to disable user log in, set the user status to Inactive.

Procedure

1. On the Ambari Admin page, browse to **Users**.
2. Click **Delete User**.
3. Confirm the deletion.



Note: You can reuse the name of a local user that has been deleted.

Enable user home directory creation

You can enable automated creation of a `/user/[USER_NAME]` HDFS home directory for each user that you create.

About this task

A common requirement to initialize user accounts to run Hadoop components is the existence of a unique, `/user/[USER_NAME]` HDFS home directory. You can enable automated creation of a `/user/[USER_NAME]` HDFS home directory for each user that you create. Home directory creation occurs for users created either manually using the Ambari Admin page, or through LDAP synchronization.

Procedure

1. On your Ambari Server host, edit the `ambari-properties` file, using a command line editor (`vi`, in this example):
`vi /etc/ambari-server/conf/ambari.properties`
2. Add the following property: `ambari.post.user.creation.hook.enabled=true`.

3. Add the script path to the ambari properties file: `ambari.post.user.creation.hook=/var/lib/ambari-server/resources/scripts/post-user-creation-hook.sh`



Important: In a Kerberized environment, you must modify the kinit file path in the default user creation hook script.

`/var/lib/ambari-server/resources/scripts/post-user-creation-hook.sh`

4. Restart Ambari server.
`ambari-server restart`

Results

After enabling the post-user creation script, Ambari executes the script whenever a user is created and logs a message each time the script is invoked. If the script has a non-zero exit code, an ERROR is logged, otherwise an INFO-level message that includes the script path and parameters is logged.

Managing local group membership

You can manage membership of local groups by adding or removing users.

Understanding group types

Ambari supports two types of groups: local and LDAP.

You use the Ambari Admin page to manage both local and LDAP groups. Local groups are stored in the Ambari database. LDAP groups have basic information stored in the Ambari database, including group membership information. Unlike local groups, LDAP groups are imported and synchronized from an external LDAP system. To use LDAP groups with Ambari, you must configure Ambari to authenticate against an external LDAP system. Ambari grants no permissions by default to a new group, created either locally or by synchronizing against LDAP.

Modify group membership

Use **Admin/Groups** to add local users to an existing group.

Procedure

1. On the Ambari Admin page, browse to **Users > Groups**.
2. For a group name, under **Actions**, click edit.
3. In Groups/[group name], use the **Local Members** control to modify group membership.
 - a) Click in the **Local Members** text area.
 - b) In the New control, type a new local user name,
 - c) Click the **x** next to the name of a user to remove that user.
 - d) To save your changes, click the check mark.
 - e) To discard your changes, click **x**.

Create a local group

Use **Admin/Groups** to create or manage an existing local group.

Procedure

1. On the Ambari Admin page, browse to **Users > Groups**.
2. Click **Add Groups**.

3. Enter a unique group name.
4. Click **Save**.

Delete a local group

Use **Admin/Groups** to create or manage an existing local group.

Procedure

1. On the Ambari Admin page, browse to **Users > Groups**.
2. On **Admin/Groups**, review the list of group names.
3. For a named group, under Actions, click the can to delete the group.
4. Confirm.

Results

Deleting a local group also removes associated group membership information, including privileges.

Installing Ambari agents manually

Install Ambari agents manually if you do not have SSH for Ambari to install them.

About this task

In cases where you do not have SSH for Ambari to automatically install the agents or you want to pre-install the agents, you can install an Ambari agent manually. On each host in the cluster, use the instructions specific to the os family running on your installation host.

Procedure

1. Download the Ambari repository.
2. Install the Ambari agent manually.

Download the Ambari repository on RHEL-CentOS-Oracle Linux 7

To install Ambari agents, first download the Ambari repository to a host in your cluster.

About this task

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

Procedure

1. Log in to your host as root.
2. Download the Ambari repository file to a directory on your installation host.
`wget -nv https://archive.cloudera.com/p/ambari/2.x/2.7.0.0/centos7/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 for Ambari repositories listed similar to the following. Version values vary, depending on the installation.

```

repo id          repo name          status
ambari-2.7.0.0-143  ambari Version - ambari-2.7.0.0-143
12
epel/x86_64      Extra Packages for Enterprise Linux 7 - x86_64
11,387
ol7_UEKR4/x86_64 Latest Unbreakable Enterprise Kernel Release 4
for Oracle Linux 7Server (x86_64) 295
ol7_latest/x86_64 Oracle Linux 7Server Latest (x86_64)
18,642
puppetlabs-deps/x86_64 Puppet Labs Dependencies El 7 - x86_64
17
puppetlabs-products/x86_64 Puppet Labs Products El 7 - x86_64
225
repolist: 30,578

```

4. Accept the warning about trusting the Cloudera GPG Key.

That key will be automatically downloaded and used to validate packages from Cloudera. You will see the following message:

```

Importing GPG key 0x07513CAD: Userid: "Jenkins (HDP Builds)
<jenkin@hortonworks.com>"
      From : https://archive.cloudera.com/p/ambari/2.x/2.7.0.0/
centos7/RPM-GPG-KEY/RPM-GPG-KEY-Jenkins

```

Install the Ambari agents manually on RHEL-CentOS-Oracle 7

Install an Ambari agent on each host in your cluster.

About this task

On every host in your cluster, use a command line editor to do the following steps.

Procedure

1. Install the Ambari Agent.
yum install ambari-agent
2. Using a text editor, configure the Ambari Agent by editing the ambari-agent.ini file, as shown in the following:

```

vi /etc/ambari-agent/conf/ambari-agent.ini
[server]
hostname=<your.ambari.server.hostname>
url_port=8440
secured_url_port=8441

```

3. Start the agent on every host in your cluster.
ambari-agent start
The agent registers with the Server on start.

Download the Ambari repository on Amazon Linux 2

To install Ambari agents, first download the Ambari repository to a host in your cluster.

About this task

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

Procedure

1. Log in to your host as root.
2. Download the Ambari repository file to a directory on your installation host.
`wget -nv https://archive.cloudera.com/p/ambari/2.x/2.7.0.0/amazonlinux2/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 for Ambari repositories listed similar to the following. Version values vary, depending on the installation.

```
repo id          repo name          status
ambari-2.7.0.0-143  ambari Version - ambari-2.7.0.0-143
12
epel/x86_64      Extra Packages for Enterprise Linux 7 - x86_64
11,387
ol7_UEKR4/x86_64 Latest Unbreakable Enterprise Kernel Release 4
for Oracle Linux 7Server (x86_64) 295
ol7_latest/x86_64 Oracle Linux 7Server Latest (x86_64)
18,642
puppetlabs-deps/x86_64 Puppet Labs Dependencies El 7 - x86_64
17
puppetlabs-products/x86_64 Puppet Labs Products El 7 - x86_64
225
repolist: 30,578
```

4. Accept the warning about trusting the Cloudera GPG Key.
 That key will be automatically downloaded and used to validate packages from Cloudera. You will see the following message:

```
Importing GPG key 0x07513CAD: Userid: "Jenkins (HDP Builds)
<jenkin@hortonworks.com>"
      From : https://archive.cloudera.com/p/ambari/2.x/2.7.0.0/
centos7/RPM-GPG-KEY/RPM-GPG-KEY-Jenkins
```

Install the Ambari agents manually on Amazon Linux 2

Install an Ambari agent on each host in your cluster.

About this task

On every host in your cluster, use a command line editor to do the following steps.

Procedure

1. Install the Ambari Agent.
`yum install ambari-agent`
2. Using a text editor, configure the Ambari Agent by editing the `ambari-agent.ini` file, as shown in the following:

```
vi /etc/ambari-agent/conf/ambari-agent.ini
[server]
hostname=<your.ambari.server.hostname>
url_port=8440
secured_url_port=8441
```

3. Start the agent on every host in your cluster.
`ambari-agent start`
 The agent registers with the Server on start.

Download the Ambari repository on SLES 11

To install Ambari agents, first download the Ambari repository to a host in your cluster.

About this task

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

Procedure

1. Log in to your host as root.
2. Download the Ambari repository file to a directory on your installation host.
`wget -nv https://archive.cloudera.com/p/ambari/2.x/2.7.0.0/suse11/ambari.repo -O /etc/zypp/repos.d/ambari.repo`**Note:** 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, similar to the following. Version values vary, depending on the installation.

```
# | Alias | Name |
  Enabled | Refresh |
-----+-----+-----+
+-----+-----+-----+
1 | ambari-2.7.0.0-143 | ambari Version - ambari-2.7.0.0-143 | Yes
  | No |
2 | http-demeter.uni | SUSE-Linux-Enterprise-Software
  -regensburg.de-c997c8f9 | -Development-Kit-11-SP3 12.1.1-1.57 | Yes
  | Yes |
3 | opensuse | OpenSuse | Yes
  | Yes |
```

Install the Ambari agents manually on SLES 11

Install an Ambari agent on each host in your cluster.

About this task

On every host in your cluster, use a command line editor to do the following steps.

Procedure

1. Install the Ambari Agent.
`zypper install ambari-agent`
2. Configure the Ambari Agent by editing the `ambari-agent.ini` file as shown in the following:

```
vi /etc/ambari-agent/conf/ambari-agent.ini
[server]
hostname=<your.ambari.server.hostname>
url_port=8440
secured_url_port=8441
```

3. Start the agent on every host in your cluster.
`ambari-agent start`
 The agent registers with the Server on start.

Download the Ambari repository on SLES 12

To install Ambari agents, first download the Ambari repository to a host in your cluster.

About this task

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

Procedure

1. Log in to your host as root.
2. Download the Ambari repository file to a directory on your installation host.
`wget -nv https://archive.cloudera.com/p/ambari/2.x/2.7.0.0/sles12/ambari.repo -O /etc/zypp/repos.d/ambari.repo`



Note: 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, similar to the following. Version values vary, depending on the installation.

```
# | Alias | Name |
  Enabled | Refresh |
-----+-----+-----+
1 | ambari-2.7.0.0-143 | ambari Version -
  | Yes | No |
2 | http-demeter.uni | SUSE-Linux-Enterprise-
Software
  | Yes | Yes |
12.1.1-1.57 | Yes | Yes |
  | Yes | Yes |
3 | opensuse | OpenSuse
```

Install the Ambari agents manually on SLES 12

Install an Ambari agent on each host in your cluster.

About this task

On every host in your cluster, use a command line editor to do the following steps.

Procedure

1. Install the Ambari Agent.
`zypper install ambari-agent`
2. Configure the Ambari Agent by editing the `ambari-agent.ini` file as shown in the following:

```
vi /etc/ambari-agent/conf/ambari-agent.ini
[server]
hostname=<your.ambari.server.hostname>
url_port=8440
secured_url_port=8441
```

3. Start the agent on every host in your cluster.
ambari-agent start
The agent registers with the Server on start.

Download the Ambari repository on Ubuntu 14

To install Ambari agents, first download the Ambari repository to a host in your cluster.

About this task

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

Procedure

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 https://archive.cloudera.com/p/ambari/2.x/2.7.0.0/ubuntu14/ambari.list  
apt-key adv --recv-keys --keyserver keyserver.ubuntu.com B9733A7A07513CAD  
apt-get update
```



Note: 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.

Install the Ambari agents manually on Ubuntu 14

Install an Ambari agent on each host in your cluster.

About this task

On every host in your cluster, use a command line editor to do the following steps.

Procedure

1. Install the Ambari Agent.
apt-get install ambari-agent
2. Configure the Ambari Agent by editing the ambari-agent.ini file as shown in the following:

```
vi /etc/ambari-agent/conf/ambari-agent.ini  
[server]  
hostname=<your.ambari.server.hostname>  
url_port=8440  
secured_url_port=8441
```

3. Start the agent on every host in your cluster.
ambari-agent start
The agent registers with the Server on start.

Download the Ambari repository on Ubuntu 16

To install Ambari agents, first download the Ambari repository to a host in your cluster.

About this task

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

Procedure

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 https://archive.cloudera.com/p/ambari/2.x/2.7.0.0/ubuntu16/ambari.list
apt-key adv --recv-keys --keyserver keyserver.ubuntu.com B9733A7A07513CAD
apt-get update
```



Note: 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.

Install the Ambari agents manually on Ubuntu 16

Install an Ambari agent on each host in your cluster.

About this task

On every host in your cluster, use a command line editor to do the following steps.

Procedure

1. Install the Ambari Agent.
`apt-get install ambari-agent`
2. Configure the Ambari Agent by editing the ambari-agent.ini file as shown in the following:

```
vi /etc/ambari-agent/conf/ambari-agent.ini
[server]
hostname=<your.ambari.server.hostname>
url_port=8440
secured_url_port=8441
```

3. Start the agent on every host in your cluster.
`ambari-agent start`
The agent registers with the Server on start.

Download the Ambari repository on Debian 9

To install Ambari agents, first download the Ambari repository to a host in your cluster.

About this task

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

Procedure

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 https://archive.cloudera.com/p/ambari/2.x/2.7.0.0/debian9/ambari.list
apt-key adv --recv-keys --keyserver keyserver.debian.com B9733A7A07513CAD
apt-get update
```



Note: 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.

Install the Ambari agents manually on Debian 9

Install an Ambari agent on each host in your cluster.

About this task

On every host in your cluster, use a command line editor to do the following steps.

Procedure

1. Install the Ambari Agent.
apt-get install ambari-agent
2. Configure the Ambari Agent by editing the ambari-agent.ini file as shown in the following:

```
vi /etc/ambari-agent/conf/ambari-agent.ini
[server]
hostname=<your.ambari.server.hostname>
url_port=8440
secured_url_port=8441
```

3. Start the agent on every host in your cluster.
ambari-agent start
The agent registers with the Server on start.

Understanding service users and groups

Ambari creates Unix accounts for each service user and group during cluster installation.

Each Hadoop service runs under the ownership of their respective Unix accounts. These accounts are known as service users. These service users belong to a special Unix group. "Smoke Test" is a service user dedicated specifically for running smoke tests on components during installation using the **Services** tab of the Ambari Web UI. You can also run service checks as the Smoke Test user on-demand after installation. You can customize any of these users and groups using the **Misc** tab during the **Customize Services** installation step.

Use the **Skip Group Modifications** option to not modify the Linux groups in the cluster. Choosing this option is typically required if your environment manages groups using LDAP and not on the local Linux machines.

If you choose to customize names, Ambari checks to see if these custom accounts already exist. If they do not exist, Ambari creates them. The default accounts are always created during installation whether or not custom accounts are specified. These default accounts are not used and can be removed post-install.

All new service user accounts, and any existing user accounts used as service users, must have a UID greater than or equal to 1000.

Default user accounts

The Ambari Installation wizard creates default service user account names for each installed service.

Table 6: Default User Account Names for Services and Components

Service	Component	Default User Account
Accumulo	Accumulo Tracer, Accumulo Monitor, Accumulo GC, Accumulo Master	accumulo
Ambari Metrics	Metrics Collector, Metrics Monitor	ams
Ambari Infra	Infra Solr Instance	infra-solr
Atlas	Atlas Metadata Server	atlas
Falcon	Falcon Server	falcon
Flume	Flume Agents	flume
HBase	MasterServer RegionServer	hbase
HDFS	NameNode SecondaryNameNode DataNode	hdfs
Hive	Hive Metastore, HiveServer2	hive
HUE	HUE	hue
Kafka	Kafka Broker	kafka
Knox	Knox Gateway	knox
Mahout	Mahout clients	mahout
MapReduce2	HistoryServer	mapred
Oozie	Oozie Server	oozie
PostgreSQL	PostgreSQL (with Ambari Server)	postgres (Created as part of installing the default PostgreSQL database with Ambari Server. If you are not using the Ambari PostgreSQL database, this user is not needed.)
Ranger	Ranger Admin, Ranger Usersync	ranger
Ranger KMS	Ranger KMS Server	kms
Slider	Slider clients	slider
SmartSense	HST Server, HST Agent, Activity Analyzer, Activity Explorer	same as ambari agent

Service	Component	Default User Account
Spark	Lively Servers	livy
Spark	Spark History Server	spark
Sqoop	Sqoop	sqoop
Storm	Masters (Nimbus, DRPC Server, Storm REST API, Storm UI Server) Slaves (Supervisors, Logviewers)	storm
Tez	Tez clients	tez
WebHCat	WebHCat Server	hcat
YARN	NodeManager ResourceManager	yarn
Zeppelin Notebook	Zeppelin Notebook	zeppelin
ZooKeeper	ZooKeeper	zookeeper

For all components, the Smoke Test user performs smoke tests against cluster services as part of the install process. It also can perform these on-demand, from the Ambari Web UI. The default user account for the smoke test user is ambari-qa.

Default group accounts

The Ambari Installation wizard creates default service group account names for each installed service.

Table 7: Default Group Account Names for Services and Components

Service	Components	Default Group Account
All	All	hadoop
Atlas	Atlas Metadata Server	atlas
Knox	Knox Gateway	knox
Ranger	Ranger Admin, Ranger Usersync	ranger
Ranger KMS	Ranger KMS Server	kms
Spark	Spark History Server	spark

Setting properties that depend on service or group user names

You must configure Service properties to match customized user and group names for each service.

About this task

Some property values must be set to match specific service or service group user names. You set such property values using Ambari Web, on the Advanced Configurations tab for the service.

Procedure

1. Browse the Ambari Web UI > **Services** > **Service.Name** > **Configs** > **Advanced** > tabs.
2. Set the property value.

If you have set up non-default, customized service user names for the **HDFS** service, you must edit the following properties in **HDFS Settings: Advanced**:

dfs.permissions.superusergroup The same as the HDFS username. The default is hdfs .
dfs.cluster.administrators A single space followed by the HDFS username.

If you have set up non-default, customized service user names for the **HBase** service, you must edit the following properties in **HDFS Settings: Advanced**:

dfs.block.local-path-access.user The HBase username. The default is hbase .

If you have set up non-default, customized service user names for the Hadoop group name, you must edit the following properties, in **MapReduce Settings: Advanced**:

mapreduce.cluster.administrators A single space followed by the Hadoop group name.

Understanding custom and private host names

The Ambari Server relies on the host names of Ambari Agent to communicate to operators which hosts belong members of the cluster. You can configure custom or private host names.

It is common in large cluster deployments to use DNS aliases for specific hosts, so that configuration files mentioning those hosts do not need to be changed when the services on that host are moved to another physical machine. For example, if you have multiple deployed applications that write to HDFS, using a DNS alias instead of a physical host name to refer to the HDFS NameNode allows you to move the NameNode to other physical machines without having to change those deployed application’s HDFS client configuration.

In Ambari, individual hosts can be configured to use a public host name when referencing individual hosts in configuration files or in **Quick Links**. For example, if you have a physical host with a FQDN of rev01.hortonworks.local, and you have a DNS CNAME that also points to that host using nn1.hortonworks.local, it is possible to configure Ambari to use nn1.hortonworks.local for **Quick Links** associated with that host, and whenever nn1.hortonworks.local is used in configuration, Ambari will understand that it is associated with the rev01.hortonworks.local host. That way if you need to move the NameNode to rev04.hortonworks.local, you can configure that new host to use nn1.hortonworks.local for its public host name without having to make client configuration changes.



Note: You still have to modify the specific configuration properties that reference rev01.hortonworks.local and update them with the alias which you have chosen to use, nn1.hortonworks.local in this case, in order for this feature to work as expected.

Using custom and private host names

Example scenarios for using custom and public host names.

Ambari Agents can be configured to register with the Ambari Server using custom host names and public host names. Ambari uses the host name of the agent to name and refer to that host in Ambari Web, for example in the **Hosts** list. The public host name, if configured, is used as an alias for the host when referenced in configuration and is used in **Quick Links** URLs.

To determine whether to use a custom host name or public host name, consider the following scenarios:

Table 8: When To Use Custom and Public Host Names

Scenario	Configuration
If you have a host with the host name rev01.hortonworks.local, but you want it to show up in Ambari web UI as c1r1.hortonworks.local, you should:	Configure a custom host name

Scenario	Configuration
If you have a host with the host name <code>rev01.hortonworks.local</code> and want to use a DNS CNAME of <code>nn1.hortonworks.local</code> to be used for Quick Links and as a configuration alias, you should:	Configure a public host name

Configure a public host name

Edit the host name script on each host to use a public host name.

Procedure

1. Edit the contents of the `/var/lib/ambari-agent/public_host name.sh` script to return the public host name that you want the Ambari Agent to be configured with.

Make sure that you `chmod` the script so it is executable by the user running the Ambari Agent. For most installations, `0755` is the sufficient permission to use.

The following script could be used to configure the Ambari Agent to use `'nn1.hortonworks.local'` as the public host name.

```
#!/bin/sh
echo 'nn1.hortonworks.local'
```

2. Configure the Ambari Agent to use this script by editing the `/etc/ambari-agent/conf/ambari-agent.ini` using a text editor.

Add the following property to the `[agent]` section: `public_host name_script=/var/lib/ambari-agent/public_host name.sh`

In this case, Ambari Agent will use the `/var/lib/ambari-agent/public_host name.sh` script to determine the host name that it will use as the public host name in the Ambari Server.

3. Restart the agent for these changes to take effect.
`ambari-agent restart`

Configure a custom host name

Before deploying components to a host, customize the host name, if desired.

About this task

Do not customize the host name of an Ambari Agent after components have already been deployed to that host. Customizing the host name for the Ambari Agent will result in a new agent being registered with the custom host name, not the host name of an existing agent being updated.

Procedure

1. Edit the contents of the `/var/lib/ambari-agent/host name.sh` script to return the host name that you want the Ambari Agent to register with.

Make sure that you `chmod` the script so it is executable by the user running the Ambari Agent. For most installations, `0755` is the sufficient permission to use.

The following script could be used to have the Ambari Agent register with the `'c1r1.hortonworks.local'` host name.

```
#!/bin/sh
echo 'c1r1.hortonworks.local'
```

2. Configure the Ambari Agent to use this script, by editing the `/etc/ambari-agent/conf/ambari-agent.ini` using a text editor.

Add the following property to the `[agent]` section: `host name_script=/var/lib/ambari-agent/host name.sh`

In this case, Ambari Agent will use the `/var/lib/ambari-agent/host name.sh` script to determine the host name that it will use to register with the Ambari Server.

3. Restart the agent to ensure that it registers with the custom host name.
`ambari-agent restart`

Public host name limitations

Public host name capability is limited to the host. You cannot define public host names for components.

The public host name capability is currently available on the host level, and not the host-component level. For example, if you have a host which has a NameNode and ResourceManager on it, that host's public host name will be used in the configuration for both the NameNode and the ResourceManager. It is not possible to specify a public host name for only the NameNode or only the ResourceManager. It is only possible to specify the public host name for the physical host that is running both components.

Verifying public host name configuration

You can use the Ambari REST API to double-check that Ambari has associated the public host name configured for the Agent.

Before you begin

Log in to Ambari.

Procedure

1. In your web browser, open a new tab.
2. In the new tab Enter the following URL to look at what has been configured for a specific host: `http://ambari.server:8080/api/v1/hosts/your.hosts.fqdn`
3. Look for 'public_host_name' in the JSON returned from that request to ensure that the correct public host name has been configured for the host in question.

Changing host names

A non-trivial change in which you must update the Ambari and all hosts of the new names.

About this task

Circumstances may require that you change the names of the hosts in your existing cluster. Beyond any infrastructure and environment changes you need to make, you must also change the host names that Ambari uses to manage the cluster.

Before you begin

- Make a backup of your Ambari database.
- Disable Kerberos. Using **Ambari Web**, browse to **AdminKerberos** and click **Disable Kerberos**.

Procedure

1. In **Ambari Web** > **Background Operations**, stop all pending commands and jobs.
2. In **Ambari Web** > **Dashboard**, stop all services.
3. Stop `ambari-server` and `ambari-agents` on all hosts.

```
ambari-server stop
ambari-agent stop
```

4. Create *.json file with host names changes.

host_names_changes.json

```
{
  "cluster1" : {
    "c6400.ambari.apache.org" : "c6410.ambari.apache.org",
    "c6401.ambari.apache.org" : "c6411.ambari.apache.org",
    ....
  }
}
```

where cluster1 is cluster name and "c6400.ambari.apache.org" : "c6410.ambari.apache.org" is the host names pair in the format:

```
[ CURRENT_HOST_NAME ] : [ NEW_HOST_NAME ]
```

5. Execute the following command on the ambari-server host:
ambari-server update-host-names host_names_changes.json
6. After successful end of this action, update host names for all nodes, according to changes that you added to *.json file.
7. If you changed the host name for the node on which the ambari server resides, then you must update that name for every ambari-agent.
In /etc/ambari-agent/conf/ambari-agent.ini, update the hostname field to the new host name for node on which the ambari-server resides.
8. Start ambari-server and ambari-agents on all hosts.

```
ambari-server start
ambari-agent start
```

9. If you have NameNode HA enabled, after starting the ZooKeeper service, you must:
 - a) Start all ZooKeeper components.
 - b) Execute the following command on one of the NameNode hosts:
hdfs zkfc -formatZK -force
10. Start all services, using Ambari Web. For each, browse to **Services > Service.name > Service Actions > Start**.

What to do next

If you disabled Kerberos before starting this procedure, you must enable Kerberos security by working through either the automated or manual setup procedure. If you enable Kerberos with the manual option, you must generate and deploy new keytabs that contain the new host names.

Related Information

[Enabling Kerberos Authentication Using Ambari](#)

Moving the Ambari server

Moving an existing Ambari Server to a new host requires installing a new Ambari Server and transferring data from the existing Ambari database to the new Ambari database.

About this task

This topic describes the high-level process for moving an Ambari Server with a default, PostgreSQL database. If your Ambari Server uses a non-default databases, such as MySQL, Oracle, or an existing PostgreSQL instance, you must use backup, restore, and stop/start procedures specific to that database type.

Back up current Ambari database

Creates a copy of the current Ambari database and meta info.

Procedure

1. On the Ambari Server host, stop the original Ambari Server.
ambari-server stop
2. Create a directory to hold the database backups.

```
cd /tmp
mkdir dbdumps/
cd dbdumps/
```

3. Create the database backups.

```
pg_dump -U [AMBARI_DB_USERNAME] -f ambari.sql
Password: [AMBARI_DB_PASSWORD]
```

where:

Table 9: Database Dump Script Variables, Values, and Descriptions

Variable	Description	Default
[AMBARI_DB_USERNAME]	The database username.	ambari
[AMBARI_DB_PASSWORD]	The database password.	bigdata

4. Create a backup of the Ambari Server meta info.
ambari-server backup

Update all Ambari agents

Remove existing certificates from each host and point each agent to the new Ambari server host.

Procedure

1. On each agent host, stop the agent.
ambari-agent stop
2. Remove any existing agent certificates.
rm /var/lib/ambari-agent/keys/*
3. Using a text editor, edit /etc/ambari-agent/conf/ambari-agent.ini to point to the new host.

```
[server]
hostname={NEW_AMBARI_SERVER_FQDN}
url_port=8440
secured_url_port=8441
```

Install the new Ambari server

Install a new Ambari server on the new host, drop the old Ambari database, and add a new Ambari database.

Procedure

1. Install the new Ambari server on the new host.

- yum install ambari-server
- 2. Run setup the Ambari Server and setup similar to how the original Ambari Server is configured.
ambari-server setup
- 3. Restart the PostgreSQL instance.
service postgresql restart
- 4. Open the PostgreSQL interactive terminal.

```
su - postgres
psql
```

- 5. Using the interactive terminal, drop the ambari database created by the new ambari setup and install.
drop database ambari;
- 6. Check to make sure the databases have been dropped.
\l
No ambari databases should appear in the list.
- 7. Create a new ambari database to hold the transferred data.
create database ambari;
- 8. Exit the PostgreSQL interactive terminal.
\q

Populate the new Ambari database

Add the saved, database backup to the new database.

Procedure

- 1. Copy the saved data in /tmp/dbdumps/ambari.sql from Back up Current Ambari Database to the new Ambari Server host.
- 2. Load the saved data into the new database.
psql -d ambari -f /tmp/dbdumps/ambari.sql

Start the new Ambari server and agents

Procedure

- 1. Start the new Server.
ambari-server start
- 2. On each Agent host, start the Ambari Agent.
ambari-agent start
- 3. Open Ambari Web.
Point your browser to: [NEW_AMBARI_SERVER]:8080

Configuring LZO compression

LZO is a lossless data compression library that favors speed over compression ratio. Ambari does not install nor enable LZO compression libraries by default, and must be explicitly configured to do so. To enable LZO compression in your HDP cluster, you must install LZO compression libraries throughout the cluster, and configure core-site.xml for LZO.

Related Information

[Using LZO Compression with Hive queries](#)

Enable LZO compression

Enable automatic download and installation of LZO compression libraries and verify a valid HDP-GPL repository location.

About this task

The LZO compression libraries are GPL software, and Ambari must be explicitly configured to download these libraries and install them throughout the cluster. The LZO compression libraries are hosted in a separate repository. To configure Ambari to automatically download and install LZO compression libraries:

Procedure

1. Re-run Ambari Server Setup.

```
ambari-server setup
...
GPL License for LZO: https://www.gnu.org/licenses/old-licenses/
gpl-2.0.en.html
Enable Ambari Server to download and install GPL Licensed LZO packages [y/
n] (n)?
```

2. When prompted, review the GPL license and choose y.

3. Restart the Ambari Server.

```
ambari-server restart
```

The LZO compression library packages are stored in a separate HDP-GPL repository.

Now that the Ambari Server has been configured to download and install the LZO packages, it must be configured with the location of the HDP-GPL repository. Ensure the location of the HDP-GPL repositories are correct for your installation:

4. Log in to Ambari.

5. Browse to **Admin > Stack and Versions**.

6. Click the **Versions** tab.

You see the version currently running, marked as **Current**.

7. Click **Manage Versions**.

8. Click on the version in the list that matches your current version.

9. Verify that the HDP-GPL repository is pointed to a valid location for your installation.

If you are using a local repository installation, please use HDP 2.6 Repositories to obtain the repository.

What to do next

Configure core-site.xml for LZO.

Configure core-site.xml for LZO

To enable LZO compression in your HDP cluster, you must install LZO compression libraries throughout the cluster and configure core-site.xml for LZO.

Before you begin

Enable automatic download and installation of LZO compression libraries.

Procedure

1. In **Ambari WebBrowse** to **Services > HDFS > Configs**, then expand **Advanced core-site**.
2. Set the `io.compression.codecs` property value to: `com.hadoop.compression.lzo.LzoCodec`.
3. Add a description of the config modification, then click **Save**.
4. Expand the **Custom core-site.xml** section.
5. Click **Add Property**.
6. Add to Custom core-site.xml the following property key and value:

Property Key	<code>io.compression.codec.lzo.class</code>
Property Value	<code>com.hadoop.compression.lzo.LzoCodec</code>

7. Click **Save**.
8. Add a description of the config modification, then click **Save**.
9. Restart the HDFS, MapReduce2 and YARN services.



Note: If performing a Restart or a Restart All does not start the required package install, you may need to stop, then start the HDFS service to install the necessary LZO packages. Restart is only available for a service in the Running or Started state.

Optional - Enable LZO using Ambari Blueprints

When using Ambari to provision a cluster using Ambari Blueprints, you must do additional steps to configure Ambari to download and install LZO packages if the Blueprint configuration calls for LZO to be used.

About this task

The Ambari server has a new silent setup parameter that can be used to enable Ambari to download and install LZO compression libraries.

Procedure

1. Re-run Ambari Server Setup.
2. Append the `--enable-lzo-under-gpl-license` parameter.
`ambari-server setup --enable-lzo-under-gpl-license`

When this flag is passed, Ambari will download and install GPL licensed LZO compression libraries from the HDP-GPL repository. By default, HDP 2.6.4 and later include repository locations for the HDP-GPL repository.

What to do next

If you are installing the cluster and require a local repository, please refer to HDP 3.0 Repositories to obtain the repository.

The GPL license for LZO can be obtained at: <https://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html>.

Disable automatic LZO library download and installation

If you no longer want Ambari to automatically download and install LZO compression libraries, you can disable this behavior by editing the Ambari Server property file and restarting the Ambari Server.

Procedure

1. Edit the Ambari Server properties file.
`vi /etc/ambari-server/conf/ambari.properties`
2. Change the `gpl.license.accepted` property value to `false`.
3. Restart Ambari Server.

ambari-server restart
Automatic LZO library download and installation is disabled.

What to do next

Manually install LZO.

Manually installing LZO libraries

If you have disabled automatic download and installation of LZO compression libraries, you must manually install them on each host in the cluster.

If you do not want the Ambari Server to automatically download and install GPL-licensed, LZO compression libraries, and you intend to configure HDP components to use LZO, you must manually install LZO compression libraries on each node in the cluster. You can find the LZO compression libraries in the HDP-GPL repository. Be sure to obtain the repository appropriate for your operating system.

Manually installing LZO on RHEL-CentOS-Oracle

If you have disabled automatic download and installation of LZO compression libraries, you must manually install them on each host in the cluster.

About this task

If you do not wish for the Ambari Server to automatically download and install GPL-licensed, LZO compression libraries, and you intend to configure HDP components to use LZO, you must manually install LZO compression libraries on each node in the cluster. You can find the LZO compression libraries in the HDP-GPL repository. Be sure to obtain the repository appropriate for your Operating System.

Procedure

1. On each node in your cluster, configure the HDP-GPL repository.
2. Install the Hadoop LZO compression libraries.
`yum install hadoop_lzo_2_6_4_0_91.x86_64 hadoop_lzo_2_6_4_0_91-native.x86_64`

Manually Installing LZO on SUSE Linux

If you have disabled automatic download and installation of LZO compression libraries, you must manually install them on each host in the cluster.

About this task

If you do not wish for the Ambari Server to automatically download and install GPL-licensed, LZO compression libraries, and you intend to configure HDP components to use LZO, you must manually install LZO compression libraries on each node in the cluster. You can find the LZO compression libraries in the HDP-GPL repository. Be sure to obtain the repository appropriate for your Operating System.

Procedure

1. On each node in your cluster, configure the HDP-GPL repository.
2. Install the Hadoop LZO compression libraries.
`zypper install hadoop_lzo_2_6_4_0_91.x86_64 hadoop_lzo_2_6_4_0_91-native.x86_64`

Manually installing LZO on Ubuntu or Debian

If you have disabled automatic download and installation of LZO compression libraries, you must manually install them on each host in the cluster.

About this task

If you do not wish for the Ambari Server to automatically download and install GPL-licensed, LZO compression libraries, and you intend to configure HDP components to use LZO, you must manually install LZO compression libraries on each node in the cluster. You can find the LZO compression libraries in the HDP-GPL repository. Be sure to obtain the repository appropriate for your Operating System.

Procedure

1. On each node in your cluster, configure the HDP-GPL repository.
2. Install the Hadoop LZO compression libraries.
`apt-get install hadoop-lzo_2.6.4_0.91.x86_64 hadoop-lzo_2.6.4_0.91-native.x86_64`

Using LZO compression with Hive queries

Using LZO Compression with Hive queries requires creating LZO files.

Create LZO files

You can output LZO files directly from a Hive query.

Procedure

1. Create LZO files as the output of the Hive query.
2. Use lzo command utility or your custom Java to generate lzo.index for the .lzo files.
 Prefix the query string with these parameters:

Table 10: Hive LZO Query String Parameters and Values

Parameter	Value
SET mapreduce.output.fileoutputformat.compress.codec	com.hadoop.compression.lzo.LzoCodec
SET hive.exec.compress.output	true
SET mapreduce.output.fileoutputformat.compress	true

```
hive -e "SET mapreduce.output.fileoutputformat.compress.codec=com.hadoop.compression.lzo.LzoCodec;SET
hive.exec.compress.output=true;SET mapreduce.output.fileoutputformat.compress=true;"
```

Write custom Java to create LZO files

You may also write java that creates LZO files from text-formatted Hive query output.

Procedure

1. Create text files as the output of the Hive query.
2. Write custom Java code to convert Hive query generated text files to .lzo files and generate lzo.index files for the .lzo files.

3. Prefix the query string with these parameters:

Table 11: Hive Query Parameters

Parameter	Value
SET hive.exec.compress.output	false
SET mapreduce.output.fileoutputformat.compress	false

```
hive -e "SET hive.exec.compress.output=false;SET mapreduce.output.fileoutputformat.compress=false;<query-string>"
```

Using an existing or installing a default database

Ambari installs a default database for each component. You may choose to set up components with a new or existing, non-default database.

Ambari installs the PostgreSQL, MySQL, and Derby databases for use with Ambari, Hive, and Oozie respectively, as default options. You may instead use a new, or an existing, non-default database instance with these components. To prepare Ambari to connect to a non-default database, you must download and set up database connectors before you set up the Ambari Server by running `ambari-server setup`.



Important: Using Microsoft SQL Server or SQL Anywhere database options are not supported.

Using MySQL requires the default, InnoDB engine for MySQL 5.6.

Related Information

[Hortonworks Support Matrix](#)

Using an existing database with Ambari

other than the embedded PostgreSQL database instance that Ambari Server uses by default.

Before using Ambari with an existing database, consider:

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

Using Ambari with Oracle

Before setting up Ambari Sever with an existing Oracle database; obtain the appropriate drivers and .jar files, create an Ambari user with sufficient permissions, and load the Ambari database schema.

Before you begin

Determine the appropriate Oracle database version and obtain the release drivers and .jar file.

Table 12: Oracle Database Version Information

Oracle Database Version	Drivers	File
Oracle Database 11g	Oracle Database 11g Release 2 drivers	ojdbc6.jar
Oracle Database 12c	Oracle Database 12c Release 1 drivers	ojdbc7.jar

Procedure

1. On the Ambari Server host, stage the appropriate JDBC driver file for later deployment.

- a) Download the Oracle JDBC (OJDBC) driver from <https://www.oracle.com/technetwork/database/application-development/jdbc/downloads/index.html>.
 - b) Copy the .jar file to the Java share directory.
cp ojdbc7.jar /usr/share/java/
 - c) Make sure the .jar file has the appropriate permissions.
chmod 644 /usr/share/java/ojdbc7.jar
2. Create a user for Ambari and grant that user appropriate permissions.
using the Oracle database admin utility, run the following commands:

```
# sqlplus sys/root as sysdba
CREATE USER [AMBARI_USER] IDENTIFIED BY [AMBARI_PASSWORD] default
  tablespace "USERS" temporary tablespace "TEMP";
GRANT unlimited tablespace to [AMBARI_USER];
GRANT create session to [AMBARI_USER];
GRANT create TABLE to [AMBARI_USER];
GRANT create SEQUENCE to [AMBARI_USER];
QUIT;
```

Where [AMBARI_USER] is the Ambari user name and [AMBARI_PASSWORD] is the Ambari user password.

3. Load the Ambari Server database schema.
 - a) You must pre-load the Ambari database schema into your Oracle database using the schema script.
sqlplus [AMBARI_USER]/[AMBARI_PASSWORD] Ambari-DDL-Oracle-CREATE.sql
 - b) Find the Ambari-DDL-Oracle-CREATE.sql file in the /var/lib/ambari-server/resources/ directory of the Ambari Server host after you have installed Ambari Server.

What to do next

When setting up the Ambari Server, select Advanced Database Configuration] Option [2] Oracle and respond to the prompts using the username/password credentials you created in step 2.

Using Ambari with MySQL or MariaDB

Before setting up Ambari Sever with an existing, MySQL or Maria DB database; obtain the appropriate connectors and .jar files, create an Ambari user with sufficient permissions, and load the Ambari database schema.

Before you begin

Determine the appropriate database version and obtain the release drivers and .jar file.

Procedure

1. On the Ambari Server host, stage the appropriate connector/JDBC driver file for later deployment.
 - a) On the Ambari Server host, [Download the MySQL Connector/JDBC driver from MySQL](#)
 - b) Run ambari-server setup --jdbc-db=mysql --jdbc-driver=/path/to/mysql/mysql-connector-java.jar
 - c) Confirm that .jar is in the Java share directory.
ls /usr/share/java/mysql-connector-java.jar
 - d) Make sure the .jar file has the appropriate permissions - 644.
2. Create a user for Ambari and grant it permissions.
using the MySQL database admin utility:

```
# mysql -u root -p
CREATE USER '[AMBARI_USER]'@'%' IDENTIFIED BY '[AMBARI_PASSWORD]';
GRANT ALL PRIVILEGES ON *.* TO '[AMBARI_USER]'@'%';
CREATE USER '[AMBARI_USER]'@'localhost' IDENTIFIED BY '[AMBARI_PASSWORD]';
GRANT ALL PRIVILEGES ON *.* TO '[AMBARI_USER]'@'localhost';
CREATE USER '[AMBARI_USER]'@[AMBARI_SERVER_FQDN] IDENTIFIED BY
  '[AMBARI_PASSWORD]';
GRANT ALL PRIVILEGES ON *.* TO '[AMBARI_USER]'@[AMBARI_SERVER_FQDN]';
```



```
FLUSH PRIVILEGES;
```

Where [AMBARI_USER] is the Ambari user name, [AMBARI_PASSWORD] is the Ambari user password and [AMBARI_SERVER_FQDN] is the Fully Qualified Domain Name of the Ambari Server host.

3. Load the Ambari Server database schema.

You must pre-load the Ambari database schema into your MySQL/MariaDB database using the schema script. Run the script in the same location where you find the Ambari-DDL-MySQL-CREATE.sql file. You should find the Ambari-DDL-MySQL-CREATE.sql file in the /var/lib/ambari-server/resources/ directory of the Ambari Server host, after you have installed Ambari Server.

```
mysql -u [AMBARI_USER] -p
CREATE DATABASE [AMBARI_DATABASE];
USE [AMBARI_DATABASE];
SOURCE Ambari-DDL-MySQL-CREATE.sql;
```

Where [AMBARI_USER] is the Ambari user name and [AMBARI_DATABASE] is the Ambari database name.

What to do next

When setting up the Ambari Server, select Advanced Database Configuration > Option [3] MySQL/MariaDB and enter the credentials you defined in Step 2. for user name, password and database name.

Using Ambari with PostgreSQL

Before setting up Ambari Server with an existing PostgreSQL database; obtain the appropriate connectors and .jar files, create an Ambari user with sufficient permissions, and load the Ambari database schema.

Before you begin

Determine the appropriate database version and obtain the release drivers and .jar file.

Procedure

1. On the Ambari Server host, stage the appropriate JDBC driver file for later deployment.
 - a) On the Ambari server host, [Download the PostgreSQL JDBC Driver from PostgreSQL](#).
 - b) Run `ambari-server setup --jdbc-db=postgres --jdbc-driver=/path/to/postgres/postgresql.jar`
2. Create a user for Ambari and grant it permissions.
Using the PostgreSQL database admin utility:

```
# sudo -u postgres psql
CREATE DATABASE [AMBARI_DATABASE];
CREATE USER [AMBARI_USER] WITH PASSWORD '[AMBARI_PASSWORD]';
GRANT ALL PRIVILEGES ON DATABASE [AMBARI_DATABASE] TO [AMBARI_USER];
\connect [AMBARI_DATABASE];
CREATE SCHEMA [AMBARI_SCHEMA] AUTHORIZATION [AMBARI_USER];
ALTER SCHEMA [AMBARI_SCHEMA] OWNER TO [AMBARI_USER];
ALTER ROLE [AMBARI_USER] SET search_path to '[AMBARI_SCHEMA]', 'public';
```

Where [AMBARI_USER] is the Ambari user name [AMBARI_PASSWORD] is the Ambari user password, [AMBARI_DATABASE] is the Ambari database name and [AMBARI_SCHEMA] is the Ambari schema name.

3. Load the Ambari Server database schema.

You must pre-load the Ambari database schema into your PostgreSQL database using the schema script.

```
# psql -U [AMBARI_USER] -d [AMBARI_DATABASE]
\connect [AMBARI_DATABASE];
\i Ambari-DDL-Postgres-CREATE.sql;
```

Find the Ambari-DDL-Postgres-CREATE.sql file in the /var/lib/ambari-server/resources/ directory of the Ambari Server host after you have installed Ambari Server.

What to do next

When setting up the Ambari Server, select Advanced Database Configuration > Option[4] PostgreSQL and enter the credentials you defined in Step 2. for user name, password, and database name.

Using a new or existing database with Hive

including the embedded MySQL database instance that Ambari installs and Hive uses by default.

Before using Hive with a new or existing database, consider:



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

Using Hive with Oracle

Before using Hive with a new or existing Oracle database; obtain the appropriate driver and .jar files, and create a Hive user with sufficient permissions.

Before you begin

Determine the appropriate Oracle database version and obtain the release drivers and .jar file.

Table 13: Oracle Database Version Information

Oracle Database Version	Drivers	File
Oracle Database 11g	Oracle Database 11g Release 2 drivers	ojdbc6.jar
Oracle Database 12c	Oracle Database 12c Release 1 drivers	ojdbc6.jar



Note: Although Oracle recommends ojdbc version 7 for use with Oracle12, Hive 1.x, 2.x, and 3 currently work best with ojdbc6.

Procedure

- On the Ambari Server host, stage the appropriate JDBC driver file for later deployment.
 - Download the Oracle JDBC (OJDBC) driver from <https://www.oracle.com/technetwork/database/application-development/jdbc/downloads/index.html>.
 - Make sure the .jar file has the appropriate permissions.
`chmod 644 ojdbc6.jar`
 - Add the path to the downloaded .jar file.
`ambari-server setup --jdbc-db=oracle --jdbc-driver=/path/to/downloaded/ojdbc6.jar`
- Create a user for Hive and grant it permissions.
using the Oracle database admin utility:

```
# sqlplus sys/root as sysdba
CREATE USER [HIVE_USER] IDENTIFIED BY [HIVE_PASSWORD];
GRANT SELECT_CATALOG_ROLE TO [HIVE_USER];
GRANT CONNECT, RESOURCE TO [HIVE_USER];
QUIT;
```

Where [HIVE_USER] is the Hive user name and [HIVE_PASSWORD] is the Hive user password.

Using Hive with MySQL

Before using Hive with a new or existing MySQL database; obtain the appropriate driver and .jar files, create a Hive user with sufficient permissions, and load the Hive database.

Before you begin

Determine the appropriate database version and obtain the release drivers and .jar file.

Procedure

1. On the Ambari Server host, stage the appropriate MySQL connector for later deployment.
 - a) On the Ambari Server host, [Download the MySQL Connector/JDBC driver from MySQL](#).
 - b) Run `ambari-server setup --jdbc-db=mysql --jdbc-driver=/path/to/mysql/mysql-connector-java.jar`
 - c) Confirm that `mysql-connector-java.jar` is in the Java share directory.
`ls /usr/share/java/mysql-connector-java.jar`
 - d) Make sure the .jar file has the appropriate permissions - 644.
 - e) Execute the following command:
`ambari-server setup --jdbc-db=mysql --jdbc-driver=/usr/share/java/mysql-connector-java.jar`
2. Create a user for Hive and grant it permissions.
using the MySQL database admin utility:

```
# mysql -u root -p
CREATE USER '[HIVE_USER]@'localhost' IDENTIFIED BY '[HIVE_PASSWORD]';
GRANT ALL PRIVILEGES ON *.* TO '[HIVE_USER]@'localhost';
CREATE USER '[HIVE_USER]@%' IDENTIFIED BY '[HIVE_PASSWORD]';
GRANT ALL PRIVILEGES ON *.* TO '[HIVE_USER]@%';
CREATE USER '[HIVE_USER]@[HIVE_METASTORE_FQDN]' IDENTIFIED BY
'[HIVE_PASSWORD]';
GRANT ALL PRIVILEGES ON *.* TO '[HIVE_USER]@[HIVE_METASTORE_FQDN]';
FLUSH PRIVILEGES;
```

Where `[HIVE_USER]` is the Hive user name, `[HIVE_PASSWORD]` is the Hive user password and `[HIVE_METASTORE_FQDN]` is the Fully Qualified Domain Name of the Hive Metastore host.

3. Create the Hive database.

The Hive database must be created before loading the Hive database schema.

```
# mysql -u root -p
CREATE DATABASE [HIVE_DATABASE]
```

Where `[HIVE_DATABASE]` is the Hive database name.

Using Hive with PostgreSQL

Before using Hive with a new or existing PostgreSQL database; obtain the appropriate driver and .jar files, and create a Hive user with sufficient permissions.

Before you begin

Determine the appropriate database version and obtain the release drivers and .jar file.

Procedure

1. On the Ambari Server host, stage the appropriate PostgreSQL connector for later deployment.
 - a) On the Ambari Server host, [Download the PostgreSQL JDBC Driver from PostgreSQL](#).
 - b) Confirm that .jar is in the Java share directory.
`ls /usr/share/java/postgresql-jdbc.jar`
 - c) Change the access mode of the .jar file to 644.
`chmod 644 /usr/share/java/postgresql-jdbc.jar`

- d) Execute the following command:
ambari-server setup --jdbc-db=postgres --jdbc-driver=/usr/share/java/postgresql-jdbc.jar

2. Create a user for Hive and grant it permissions.
using the PostgreSQL database admin utility:

```
echo "CREATE DATABASE [HIVE_DATABASE];" | psql -U postgres
echo "CREATE USER [HIVE_USER] WITH PASSWORD '[HIVE_PASSWORD]';" | psql -U
postgres
echo "GRANT ALL PRIVILEGES ON DATABASE [HIVE_DATABASE] TO [HIVE_USER];" |
psql -U postgres
```

Where [HIVE_USER] is the Hive user name, [HIVE_PASSWORD] is the Hive user password and [HIVE_DATABASE] is the Hive database name.

Using an existing database with Oozie

other than the Derby database instance that Ambari installs by default.

Before using Oozie with an existing database, consider:



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

Using Oozie with Oracle

Before using OOZIE with a new or existing Oracle database; obtain the appropriate driver and .jar files, and create a Hive user with sufficient permissions.

Before you begin

Determine the appropriate Oracle database version and obtain the release drivers and .jar file.

Table 14: Oracle Database Version Information

Oracle Database Version	Drivers	File
Oracle Database 11g	Oracle Database 11g Release 2 drivers	ojdbc6.jar
Oracle Database 12c	Oracle Database 12c Release 1 drivers	ojdbc7.jar

Procedure

1. On the Ambari Server host, stage the appropriate JDBC driver file for later deployment.
 - a) Download the Oracle JDBC (OJDBC) driver from <https://www.oracle.com/technetwork/database/application-development/jdbc/downloads/index.html>.
 - b) Make sure the .jar file has the appropriate permissions.
chmod 644 ojdbc7.jar
 - c) Add the path to the downloaded .jar file.
ambari-server setup --jdbc-db=oracle --jdbc-driver=/path/to/downloaded/ojdbc7.jar
2. Create a user for OOZIE and grant it permissions.
using the Oracle database admin utility:

```
# sqlplus sys/root as sysdba
CREATE USER [OOZIE_USER] IDENTIFIED BY [OOZIE_PASSWORD];
GRANT ALL PRIVILEGES TO [OOZIE_USER];
GRANT CONNECT, RESOURCE TO [OOZIE_USER];
QUIT;
```

Where [OOZIE_USER] is the Oozie user name and [OOZIE_PASSWORD] is the Oozie user password.

Using Oozie with MySQL

Before using Oozie with a new or existing MySQL database; obtain the appropriate driver and .jar files, create a Oozie user with sufficient permissions, and create the Oozie database.

Before you begin

Determine the appropriate database version and obtain the release drivers and .jar file.

Procedure

1. On the Ambari Server host, stage the appropriate MySQL connector for later deployment.
 - a) On the Ambari Server host, [Download the MySQL Connector/JDBC driver from MySQL](#).
 - b) Run `ambari-server setup --jdbc-db=mysql --jdbc-driver=/path/to/mysql/mysql-connector-java.jar`
 - c) Confirm that `mysql-connector-java.jar` is in the Java share directory.
`ls /usr/share/java/mysql-connector-java.jar`
 - d) Make sure the .jar file has the appropriate permissions - 644.
 - e) Execute the following command:
`ambari-server setup --jdbc-db=mysql --jdbc-driver=/usr/share/java/mysql-connector-java.jar`
2. Create a user for Oozie and grant it permissions.
using the MySQL database admin utility:

```
# mysql -u root -p
CREATE USER '<OOZIEUSER>'@'%' IDENTIFIED BY '<OOZIEPASSWORD>';
GRANT ALL PRIVILEGES ON *.* TO '<OOZIEUSER>'@'%' ;
FLUSH PRIVILEGES;
```

Where [OOZIE_USER] is the Oozie user name and [OOZIE_PASSWORD] is the Oozie user password.

3. Create the Oozie database.

The Oozie database must be created before loading the Oozie database schema.

```
# mysql -u root -p
CREATE DATABASE [OOZIE_DATABASE]
```

Where [OOZIE_DATABASE] is the Oozie database name.

Using Oozie with PostgreSQL

Before using Oozie with a new or existing PostgreSQL database; obtain the appropriate driver and .jar files, and create a Oozie user with sufficient permissions.

Before you begin

Determine the appropriate database version and obtain the release drivers and .jar file.

Procedure

1. On the Ambari Server host, stage the appropriate PostgreSQL connector for later deployment.
 - a) On the Ambari Server host, [Download the PostgreSQL JDBC Driver from PostgreSQL](#).
 - b) Confirm that .jar is in the Java share directory.
`ls /usr/share/java/postgresql-jdbc.jar`
 - c) Change the access mode of the .jar file to 644.
`chmod 644 /usr/share/java/postgresql-jdbc.jar`
 - d) Execute the following command:
`ambari-server setup --jdbc-db=postgres --jdbc-driver=/usr/share/java/postgresql-jdbc.jar`
2. Create a user for Oozie and grant it permissions.

using the PostgreSQL database admin utility:

```
echo "CREATE DATABASE [OOZIE_DATABASE];" | psql -U postgres
echo "CREATE USER [OOZIE_USER] WITH PASSWORD '[OOZIE_PASSWORD]';" | psql -
U postgres
echo "GRANT ALL PRIVILEGES ON DATABASE [OOZIE_DATABASE] TO [OOZIE_USER];"
| psql -U postgres
```

Where [OOZIE_USER] is the Oozie user name, [OOZIE_PASSWORD] is the Oozie user password and [OOZIE_DATABASE] is the Oozie database name.

Example: Install MariaDB for use with multiple components

Before deploying an Ambari-managed cluster, set up a secure MariaDB database and db users for each component with sufficient permissions.

Before you begin

Determine the appropriate database version and obtain the release drivers and .jar file.

About this task

This example is specific for Centos/RHEL 7 OS systems. For production setups, please consider installing the database instance on a dedicated host and configuring master-slave replication.

Procedure

1. On a dedicated host, [Download the MySQL Connector/JDBC driver from MySQL](#).
2. Install mysql packages and configure to start on boot.

```
yum install mariadb-server -y
systemctl start mariadb
systemctl enable mariadb
```

3. Secure the installation.
/usr/bin/mysql_secure_installation

4. Precreate databases and users.

% matches any host on your domain, so we add localhost explicitly

```
mysql -uroot -p

create database hive;
grant all privileges on hive.* to 'hive'@'localhost' identified by
'[YOUR_PASSWORD]';
grant all privileges on hive.* to 'hive'@'%.[YOUR_DOMAIN_NAME]'
identified by '[YOUR_PASSWORD]';

create database ranger;
grant all privileges on ranger.* to 'ranger'@'localhost' identified by
'[YOUR_PASSWORD]';
grant all privileges on ranger.* to 'ranger'@'%.[YOUR_DOMAIN_NAME]'
identified by '[YOUR_PASSWORD]';

create database rangerkms;
grant all privileges on rangerkms.* to rangerkms@'localhost' identified
by '[YOUR_PASSWORD]';
grant all privileges on rangerkms.* to rangerkms@'%.[YOUR_DOMAIN_NAME]'
identified by '[YOUR_PASSWORD]';

create database oozie;
```

```
grant all privileges on oozie.* to 'oozie'@'localhost' identified by
'[YOUR_PASSWORD]';
grant all privileges on oozie.* to 'oozie'@'%.[YOUR_DOMAIN_NAME]'
identified by '[YOUR_PASSWORD]';

create database superset DEFAULT CHARACTER SET utf8;;
grant all privileges on superset.* to 'superset'@'localhost' identified
by '[YOUR_PASSWORD]';
grant all privileges on superset.* to 'superset'@'%.[YOUR_DOMAIN_NAME]'
identified by '[YOUR_PASSWORD]';

create database druid DEFAULT CHARACTER SET utf8;;
grant all privileges on druid.* to 'druid'@'localhost' identified by
'[YOUR_PASSWORD]';
grant all privileges on druid.* to 'druid'@'%.[YOUR_DOMAIN_NAME]'
identified by '[YOUR_PASSWORD]';

exit;
```

5. Install driver on ambari host.

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

Configuring network port numbers

Ambari install wizard creates default port number assignments for Ambari Server and Ambari Agents which you can customize.

Ambari creates default port number assignments required to maintain communication between Ambari Server, Ambari Agents, and Ambari Web. An Ambari Administrator can customize port number assignments for Ambari Server and Ambari Agents.

Related Information

[Configuring Ports](#)

Default network port numbers for Ambari

Reference information related to Ambari port default settings.

Table 15: Default Ports Information for Ambari Server and Ambari Agent Services

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
Ambari Server	Ambari Server host	8080	http	Interface to Ambari Web and Ambari REST API	No	
Ambari Server	Ambari Server host	8440	https	Handshake Port for Ambari Agents to Ambari Server	No	
Ambari Server	Ambari Server host	8441	https	Registration and Heartbeat Port for Ambari Agents to Ambari Server	No	

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
Ambari Agent	All hosts running Ambari Agents	8670 You can change the Ambari Agent ping port in the Ambari Agent configuration.	tcp	Ping port used for alerts to check the health of the Ambari Agent	No	

Change the default Ambari server port

To change the default port number, 8080, that Ambari Server uses to access Ambari Web, edit the Ambari properties file.

About this task

By default, Ambari Server uses port 8080 to access Ambari Web and the REST API. To change the port number, you must edit the Ambari properties file.

Before you begin

Ambari Server should not be running when you change port numbers. Edit `ambari.properties` before you start Ambari Server the first time or stop Ambari Server before editing properties.

Procedure

1. On the Ambari Server host, open `/etc/ambari-server/conf/ambari.properties` with a text editor.
2. Add the client API port property and set it to your desired port value:
`client.api.port=[PORT_NUMBER]`
3. Start or re-start the Ambari Server.
`ambari-server start`
`http://[YOUR_AMBARI_SERVER_HOST]:[PORT_NUMBER]`

Results

Ambari Server now accesses Ambari Web via the newly configured port.

Change the default Ambari server-agent port

To change the default port number, 8187, that Ambari Server uses to access Ambari Agents, edit the Ambari properties file.

About this task

By default, Ambari Server uses port 8187 to communicate with Ambari Agents. To change the port number, you must edit the Ambari properties file.

Before you begin

Ambari Server should not be running when you change port numbers. Edit `ambari.properties` before you start Ambari Server the first time or stop Ambari Server before editing properties.

Procedure

1. On the Ambari Server host, open `/etc/ambari-server/conf/ambari.properties` with a text editor.
2. Add the following properties and set them to your desired port values:
`security.server.two_way_ssl.port=5222 security.server.one_way_ssl.port=5223`
3. On every Ambari Agent host, open `/etc/ambari-agent/conf/ambari-agent.ini` with a text editor.

4. Add the following properties and set them to your desired port values:
`url_port=5223 secured_url_port=5222`
5. Start or re-start the Ambari Server.
`ambari-server start`
`http://[YOUR_AMBARI_SERVER_HOST]:[PORT_NUMBER]`

Results

Ambari Server now accesses Ambari Agents via the newly configured port.

Tuning Ambari performance

Ambari-managed clusters larger than 100 nodes may require tuning to perform optimally.

For clusters larger than 100 nodes, consider the following tuning options:

- Increase available memory by adjusting heap size based on the number of cluster nodes.
- Set a new, larger cache size.
- Adjust the JDBC connection pool settings.
- Increase Wait Timeout and Interactive Timeout Settings.
- Increase available memory by adjusting heap size based on the number of cluster nodes.
- For clusters larger than 1500 nodes, configure Ambari agents for optimum performance

After performing one or more of these options, restart Ambari server for the option(s) to take effect.

`ambari-server restart`

If you are using the Ambari Metrics service, you might want to consider switching from the default embedded mode to distributed mode, as well as other tuning options.

Related Information

[Tuning performance for AMS](#)

Adjust Ambari server heap size

Adjust the heap size on the Ambari server host to increase available memory.

Procedure

1. On the Ambari server host, edit the `ambari-env.sh` file:
`vi /var/lib/ambari-server/ambari-env.sh`
2. For the `AMBARI_JVM_ARGS` variable, replace the default `-Xmx2048m` with a value such as: `-Xmx4GB -Xmn2GB` based on the number of nodes in your cluster. Use the following recommendations as guidance:

Table 16: Recommended Ambari Server heap size settings for large clusters

Number of Cluster Nodes	Xmx value	Xmn value
100 - 400	4 GB	2 GB
400 - 800	4 GB	2 GB
800 - 1200	8 GB	2 GB
1200 - 1600	16 GB	2.4 GB

Increase Ambari server cache size

Increase the Ambari Server cache size to accommodate a large number of hosts.

Procedure

1. Calculate a new, larger cache size, using the following relationship:
 $ecCacheSizeValue=60*[CLUSTER_SIZE]$
 Where [CLUSTER_SIZE] is the number of nodes in the cluster.
2. On the Ambari Server host, in /etc/ambari-server/conf/ambari-properties, add the following property and value:
`server.ecCacheSize=[EC_CACHE_SIZE_VALUE]`
 where [EC_CACHE_SIZE_VALUE] is the value calculated in the previous step, based on the number of nodes in the cluster.

Adjust jdbc connection pool settings

Add the following properties to adjust the jdbc connection pool settings:
 On the Ambari Server host, in /etc/ambari-server/conf/ambari-properties, add the following properties and values:

Table 17: JDBC Connection Pool Settings

Property	Setting
server.jdbc.connection-pool.acquisition-size	5
server.jdbc.connection-pool.max-age	0
server.jdbc.connection-pool.max-idle-time	14400
server.jdbc.connection-pool.max-idle-time-excess	0
server.jdbc.connection-pool.idle-test-interval	7200

Increase MySQL wait_timeout and interactive_timeout settings

Adjust the following properties to increase the timeout values for a MySQL Ambari backend database:

About this task

If using MySQL as the Ambari database:



Important:

It is critical that the Ambari configuration for `server.jdbc.connection-pool.max-idle-time` and `server.jdbc.connection-pool.idle-test-interval` are lower than the MySQL `wait_timeout` and `interactive_timeout` set on the MySQL side.

Procedure

1. In your MySQL configuration, increase the `wait_timeout`, `interactive_timeout`, and `max. connections` settings.

Table 18: Recommended MySQL Timeout and Connection Settings

Setting	Default	Recommended
wait_timeout		8 hrs (28800)
interactive_timeout		8 hrs (28800)
max. connections	32	128

- If you choose to decrease these timeout values, adjust connection pool settings in the Ambari configuration so that they are less than the adjusted wait_timeout and interactive_timeout values.

Table 19: Recommended MySQL Connection Pool Settings

Connection Pool Setting	Value Must Be	Timeout Setting
downserver.jdbc.connection-pool.max-idle-time	<	wait_timeout
server.jdbc.connection-pool.idle-test-interval	<	interactive_timeout

Purge Ambari server database history

To reduce performance degradation, use an Ambari server CLI command to automate removal of historical records from the Ambari server database.

About this task

After months of operating a larger cluster, an Ambari server may begin to accrue a large amount of historical data in its database. Data accrual can cause UI performance degradation. The db-purge-history command takes two arguments, the name of the cluster, and the date of the earliest record to purge. The following example steps demonstrate purging history records created before August 1st, 2017 for the cluster named [PROD].

Procedure

- Stop the Ambari Server by using ambari-server stop.

```
# ambari-server stop
Using python /usr/bin/python
Stopping ambari-server
Waiting for server stop
Ambari Server stopped
```

- Run db-purge-history .

```
# ambari-server db-purge-history --cluster-name [PROD] --from-date
2017-08-01
Using python /usr/bin/python
Purge database history...
Ambari Server configured for Embedded Postgres. Confirm you have made a
backup of the Ambari Server database [y/n] y
Ambari server is using db type Embedded Postgres. Cleanable database
entries older than 2017-08-01 will be purged. Proceed [y/n] y
Purging historical data from the database ...
Purging historical data completed. Check the ambari-server.log for
details.
Ambari Server 'db-purge-history' completed successfully.
```

- Start the Ambari Server: by using ambari-server start .

```
# ambari-server start
Using python /usr/bin/python
Starting ambari-server
Ambari Server running with administrator privileges.
Organizing resource files at /var/lib/ambari-server/resources...
Ambari database consistency check started...
Server PID at: /var/run/ambari-server/ambari-server.pid
Server out at: /var/log/ambari-server/ambari-server.out
Server log at: /var/log/ambari-server/ambari-server.log
Waiting for server start.....
Server started listening on 8080
```

```
DB configs consistency check: no errors and warnings were found.
Ambari Server 'start' completed successfully.
```

Results

The db-purge-history command analyzes tables in the Ambari Server database and removes those rows that can be deleted which have a --create-date after the --from-date specified when the command runs. In this example, the rows deleted would be:

- AlertCurrent
- AlertNotice
- ExecutionCommand
- HostRoleCommand
- Request
- RequestOperationLevel
- RequestResourceFilter
- RoleSuccessCriteria
- Stage
- TopologyHostRequest
- TopologyHostTask
- TopologyLogicalTask

Optimize Ambari agent performance

For clusters larger than 1500 nodes, you should configure Ambari agents to optimize performance by decreasing their reporting frequency.

About this task

For clusters larger than 1,500 nodes, you should set two properties for optimal performance:

Property/Value

command_update_output

When set to 0, the Ambari Agent only reports output of component tasks, such as install/start, after the task completes.

send_alert_changes_only

When set to 1, the Ambari Agent only sends alert status if the status changes. For example, alert status changes from OK to Warning, or changes from Critical to OK. Side Effect: When set to 1, no live text output of the alert displays in the Ambari Web UI.

On each cluster host, in the agent section of the ambari-agent.ini file:

Procedure

1. Set `command_update_output=0`
2. Set `send_alert_changes_only=1`
3. Restart the Ambari Agent.
`ambari-agent restart`

Customizing Ambari log and pid directories

Ambari Server and Agents write log activity output to .log files and use a .pid file that contains the process identification number for their running process.

More Information

<https://linuxconfig.org/logrotate-8-manual-page>

Related Information

<http://linuxconfig.org/logrotate-8-manual-page>

Finding Ambari log files

Default locations for Ambari Server and Agent log files.

Ambari Server writes log files and .pid files on it's host in following, default locations:

Table 20: Ambari Server Log File Locations

Host	File Path	File Name
[AMBAR_SERVER_FQDN]	/var/log/ambari-server/	ambari-server.log
[AMBAR_SERVER_FQDN]	/var/run/ambari-server/	ambari-server.pid

Ambari Agents write log files and .pid files on each, respective agent host in following, default locations:

Table 21: Ambari Agent Log File Locations

Host	File Path	File Name
[AMBAR_AGENT_FQDN]	/var/log/ambari-agent/	ambari-agent.log
[AMBAR_AGENT_FQDN]	/var/run/ambari-agent/	ambari-agent.pid

Configure Ambari logging level

You can set the level at which Ambari Server and Ambari Agents create log files that record cluster operations activity.

Procedure

1. Using a command line editor on the Ambari Server host, browse to `/etc/ambari-server/conf/log4j.properties`.
2. Modify the logging level for Ambari Server by editing `ambari-server.log`.
3. Using a command line editor on the Ambari Agent host, browse to `/etc/ambari-agent/conf/ambari-agent.ini`.
4. Modify the logging level for each Agent by editing `loglevel`

You can also modify these locations. Generally, you should also consider setting log-rotate policies for your systems. Refer to your operating system documentation for more information on setting up log-rotate in your environment.

Customizing Ambari agent log and pid directories

To modify the Ambari agent log and pid locations:

Procedure

1. Using a command line editor on each host running an Ambari agent, stop the Ambari agent.
ambari-agent stop
2. Edit the Ambari agent properties file.
vi /etc/ambari-agent/conf/ambari-agent.ini
3. In the [AGENT] section, modify the piddir and logdir properties.

```
[AGENT]
logdir=/var/log/ambari-agent
piddir=/var/run/ambari-agent
```

4. Save the ambari-agent.inifile.
5. Create the new directories and be sure to set the directory ownership and permissions to allow the Ambari agent process access.
6. Restart the Ambari agent.
ambari-agent start

Managing host participation for HDFS and YARN

You can configure Ambari to manage which hosts in a cluster are included and excluded from participating in cluster operations for HDFS and YARN services.

Both HDFS and YARN can control which hosts in a cluster are included and excluded from participating in the cluster. HDFS uses the dfs.hosts and dfs.hosts.exclude properties to define those datanode hosts allowed and those not allowed to connect to the NameNode. YARN uses user-definable files configured through the yarn.resourcemanager.nodes.include-path and yarn.resourcemanager.nodes.exclude-path properties to define those hosts running the NodeManager component allowed and those not allowed to communicate with the ResourceManager. When the contents of these files are modified, both the HDFS NameNode and YARN ResourceManager must be notified of these changes by invoking the -refreshNodes commands through [dfsadmin](#) for HDFS and [rmadmin](#) for YARN. You can configure Ambari to manage host participation.

Related Information

- [dfsadmin](#)
- [rmadmin](#)

Ambari-managed host participation

When you use Ambari to configure manage.include.file properties for HDFS and YARN, Ambari manages how hosts for each component participate in the cluster when you add, delete, or decommission a component.

You can configure Ambari to manage the hosts that participate in a cluster for both YARN and HDFS services. You can enable this feature only for HDFS or for YARN, or for both services. When enabled, Ambari manages files included and excluded and updates their contents based on the state of hosts as displayed in Ambari Web. When files changes occur, Ambari calls the necessary refreshNodes commands to update the DataNode, NameNode, and ResourceManager host states as necessary.

Table 22: User Actions, Ambari Operations, and Automated Tasks

User Action	Ambari Operation	Include File Action	Exclude File Action	Refresh Nodes Call	Triggers Master Restart Indicator
Add a NodeManager or DataNode	Add Component	Add hostname	Remove hostname	Yes	No

User Action	Ambari Operation	Include File Action	Exclude File Action	Refresh Nodes Call	Triggers Master Restart Indicator
Remove a NodeManager or DataNode	Delete Component	Remove hostname	Remove hostname	No	No
Decommission a NodeManager or DataNode	Decommission Component	Remove hostname (YARN only)	Add hostname	Yes	No
Recommission a NodeManager or DataNode	Recommission Component	Add hostname (YARN only)	Remove hostname	Yes	No

Enable manage.include.files for HDFS

Using Ambari Web, enable automated host participation by adding manage.include.files properties to the HDFS configuration.

About this task

When manage.include.files is configured and enabled, Ambari manages files that include and exclude hosts participating in cluster operations. Ambari also updates these files based on the state of hosts as displayed in Ambari Web. Add the manage.include.files=true and dfs.hosts properties to the HDFS configuration to enable automated host file participation for the HDFS service.

Procedure

1. Using **Ambari Web**, browse to **Services > HDFS**.
2. In the **Advanced hdfs-site** configuration section, add the manage.include.files=true property.
3. In the **Custom hdfs-site** configuration section, ensure that the dfs.hosts property appears and that it is set to a valid location of the HDFS NameNode on the filesystem .
Also, ensure that the file exists and is owned by the user account being used to run the NameNode.
dfs.hosts=/etc/hadoop/conf/dfs.include
4. Restart services, as prompted by Ambari.

Results

When manage.include.files is configured and enabled, Ambari manages files that include and exclude hosts participating in cluster operations.

Enable manage.include.files for Yarn

Using Ambari Web, enable automated host participation by adding manage.include.files properties to the YARN configuration.

About this task

When configured and enabled, Ambari manages files that include and exclude hosts participating in cluster operations. Ambari also updates these files based on the state of hosts as displayed in Ambari Web. Add the manage.include.files=true and yarn.resourcemanager.nodes.include-path to the YARN configuration to enable automated host file participation for the YARN service.

Procedure

1. Using **Ambari Web**, browse to **Services > YARN**.
2. In the **Advanced yarn-site** configuration section, add the manage.include.files=true property.

3. In the **Custom yarn-site** configuration section, ensure that the `yarn.resourcemanager.nodes.include-path` property appears and that it is set to a valid location of the YARN Resource Manager on the filesystem .
`yarn.resourcemanager.nodes.include-path=/etc/hadoop/conf/yarn.include`
4. Restart services, as prompted by Ambari.

Results

When `manage.include.files` is configured and enabled, Ambari manages files that include and exclude hosts participating in cluster operations.

Disable `manage.include.files` for HDFS

Using Ambari Web, disable automated host participation by setting `manage.include.files` properties to false in the HDFS configuration.

About this task

About this task

When `manage.include.files` is disabled, Ambari does not manage the files that include and exclude hosts participating in cluster operations. Ambari will not update these files based on the state of hosts as displayed in Ambari Web. Set the `manage.include.files=false` and remove the `dfs.hosts` property from the HDFS configuration to disable automated host file participation for the HDFS service.

Procedure

1. Using **Ambari Web**, browse to **Services > HDFS**.
2. In the **Advanced hdfs-site** configuration section, set the `manage.include.files` property to false.
3. In the **Custom hdfs-site** configuration section, remove the `dfs.hosts` property.
4. Restart services, as prompted by Ambari.

Results

When `manage.include.files` is disabled, Ambari does not manage the files that include and exclude hosts participating in cluster operations.

Disable `manage.include.files` for Yarn

Using Ambari Web, disable automated host participation by setting `manage.include.files` properties to false in the YARN configuration.

About this task

When `manage.include.files` is disabled, Ambari does not manage the files that include and exclude hosts participating in cluster operations. Ambari will not update these files based on the state of hosts as displayed in Ambari Web. Set the `manage.include.files=false` and remove the `yarn.resourcemanager.nodes.include-path` property from the YARN configuration to disable automated host file participation for the YARN service.

Procedure

1. Using **Ambari Web**, browse to **Services > YARN**.
2. In the **Advanced yarn-site** configuration section, set the `manage.include.files` property to false.
3. In the **Custom yarn-site** configuration section, remove the `yarn.resourcemanager.nodes.include-path` property.
4. Restart services, as prompted by Ambari.

Results

When `manage.include.files` is disabled, Ambari does not manage the files that include and exclude hosts participating in cluster operations.