Managing and Monitoring a Cluster

Date of Publish: 2018-11-15
# Contents

## Introducing Ambari operations

- Understanding Ambari architecture ................................................................. 5
- Access Ambari ..................................................................................................... 6
- Access Ambari Admin page ............................................................................... 7

## Working with the cluster dashboard

- Open the cluster dashboard ............................................................................. 9
- Scan operating status ....................................................................................... 10
- Monitor service metrics ................................................................................... 12
- View cluster health .......................................................................................... 12
- View cluster heatmaps ...................................................................................... 14
- Finding current stack and versions information .............................................. 15
- Viewing service account user names ................................................................. 18

## Modifying the cluster dashboard

- Replace a removed widget .............................................................................. 20
- Reset the dashboard ......................................................................................... 20
- Customize metrics display .............................................................................. 21

## Managing Hosts

- View host status ................................................................................................ 21
- Find a host in the cluster .................................................................................. 22
- Perform host level actions ............................................................................... 24
- Add a host to a cluster ...................................................................................... 25
- Manage components on a host ........................................................................ 26
- Decommission a host ....................................................................................... 28
- Decommission a NodeManager ....................................................................... 28
- Decommission a RegionServer ........................................................................ 29
- Decommission a DataNode ............................................................................. 30
- Delete a component ........................................................................................ 31
- Delete a host from a cluster ............................................................................ 32
- Bulk add or delete hosts .................................................................................. 33
- Setting Maintenance Mode ............................................................................. 34
- Set Maintenance Mode for a host .................................................................. 35
- Set Maintenance Mode for a service ............................................................... 36

## Establishing Rack Awareness

- Set the rack id for Ambari ............................................................................... 37
- Set the rack id on a host ................................................................................... 38
- Set the rack id using a custom topology script .............................................. 39

## Managing Services

- View service summary ..................................................................................... 40
- Find quick links to service information ......................................................... 41
Managing service configuration settings ................................. 72
   Change configuration settings ............................................. 72
   Adjust Smart Config settings .............................................. 73
   Edit specific configuration properties ................................ 74
   Review and confirm configuration changes .......................... 74
   Restart components ......................................................... 76
   Download client configuration files for a service .................. 76
   Download all client configuration files for a cluster .............. 77

Managing service configuration versions .................................. 78
   Understanding service configuration versions ...................... 78
   Service configuration terminology .................................... 79
   Save a service configuration change ................................... 79
   View service configuration history .................................... 80
   Compare service configuration versions ............................ 82
   Make a previous service version current ............................ 84

Managing HDFS ..................................................................... 84
   Rebalance HDFS blocks .................................................. 84
   Tune HDFS garbage collection ....................................... 85
   Customize the HDFS home directory ............................... 85
Introducing Ambari operations

Apache Ambari collects a wide range of information from cluster nodes and services and displays that information in an easy-to-use, centralized interface known as Ambari Web.

Hadoop is a large-scale, distributed data storage and processing infrastructure using clusters of commodity hosts networked together. Monitoring and managing such complex distributed systems is not simple. To help you manage the complexity, Ambari server, agent and infrastructure components provide you operating control of hosts in the cluster as well as administrative control of cluster access.

Ambari Web displays information such as service-specific summaries, graphs, and alerts. You create and manage your cluster using Ambari Web to perform basic operational tasks, such as starting and stopping services, adding hosts to your cluster, and updating service configurations. You also can use Ambari Web to perform administrative tasks for your cluster, such as enabling Kerberos security and performing Stack upgrades. Any user can view Ambari Web features. Users with administrator-level roles can access more options than those users with operator-level or view-only roles. For example, an Ambari administrator can manage cluster security, an operator user can monitor the cluster, but a view-only user can only access features to which an administrator grants required permissions.

Related Information
Understanding cluster roles and access

Understanding Ambari architecture

Simple overview of the Ambari Server, Agents and Web architecture.

The Ambari Server collects data from across your cluster. Each host has a copy of the Ambari Agent, which allows the Ambari Server to control each host.
Ambari Web is a client-side, JavaScript application that calls the Ambari REST API (accessible from the Ambari Server) to access cluster information and perform cluster operations. After authenticating to Ambari Web, the application authenticates to the Ambari Server. Communication between the browser and server occurs asynchronously using the REST API.

The Ambari Web UI periodically accesses the Ambari REST API, which resets the session timeout. Therefore, Ambari Web sessions do not timeout automatically by default. You can configure Ambari to timeout after a period of inactivity.

Access Ambari

Access an Ambari Server using a web browser and your default credentials.

Before you begin
Install, set up and start Ambari using a command line editor.

About this task
After installing and starting Ambari Server from the command line, you access Ambari using a web browser directed to the Ambari Server host's fully qualified domain name (FQDN).

Procedure
1. Open a supported web browser.
2. Enter the Ambari Web URL in the browser address bar.
   http://[YOUR_AMBARI_SERVER_FQDN]:8080
   The Ambari Web Sign In page displays in your browser.
3. Type your user name and password in the Sign In page.
   If you are an Ambari administrator accessing the Ambari Web UI for the first time, use the default Ambari administrator credentials.
   admin/admin

4. Click **Sign In**.
   If Ambari Server is stopped, you can restart it using a command line editor.

5. If necessary, start Ambari Server on the Ambari Server host machine.
   `ambari-server start`
   Typically, you start the Ambari Server and Ambari Web as part of the installation process.

**Related Information**
Installing, configuring, and deploying a cluster

**Access Ambari Admin page**
As an Ambari administrator, you use the Admin page to perform tasks that require Ambari-level permissions.

**About this task**
Only an Ambari administrator can access the **Ambari Admin** page from Ambari Web. The Ambari Admin page supports tasks such as creating a cluster, managing users, groups, roles, and permissions, and managing stack versions.
Procedure

1. From the user menu, click the Manage Ambari option.

The user menu displays the user name of the current user. The Manage Ambari option appears only for users assigned Ambari Administrator access to the cluster.

If Ambari Server is stopped, you can restart it using a command line editor.

2. If necessary, start Ambari Server on the Ambari Server host machine.

   ambari-server start

   Typically, you start the Ambari Server and Ambari Web as part of the installation process.

Working with the cluster dashboard

Use Dashboard to view metrics and configuration history for your cluster.

You monitor your Hadoop cluster using Ambari Web. Dashboard provides Metrics and Heatmaps visualizations and cluster configuration history options. You access the cluster dashboard by clicking Dashboard at the top left of the Ambari Web UI main window.
Open the cluster dashboard

Use the Ambari Web Dashboard to monitor operating status of your cluster.

About this task
Ambari Web displays the Dashboard page as the home page.

Procedure
• In Ambari Web, click Dashboard to view the operating status of your cluster.
  Dashboard includes three options:
  • Metrics
  • Heatmaps
  • Config History

The Metrics option displays by default. On the Metrics page, multiple widgets represent operating status information of services in your cluster. Most widgets display a single metric; for example, HDFS Disk Usage represented by a load chart and a percentage figure:

Example

Example
Metrics Widgets and Descriptions:
HDFS metrics
<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDFS Disk Usage</td>
<td>The percentage of distributed file system (DFS) used, which is a combination of DFS and non-DFS used</td>
</tr>
<tr>
<td>Data Nodes Live</td>
<td>The number of DataNodes operating, as reported from the NameNode</td>
</tr>
<tr>
<td>NameNode Heap</td>
<td>The percentage of NameNode Java Virtual Machine (JVM) heap memory used</td>
</tr>
<tr>
<td>NameNode RPC</td>
<td>The average RPC queue latency</td>
</tr>
<tr>
<td>NameNode CPU WIO</td>
<td>The percentage of CPU wait I/O</td>
</tr>
<tr>
<td>NameNode Uptime</td>
<td>The NameNode uptime calculation</td>
</tr>
<tr>
<td>YARN metrics (HDP 2.1 or later stacks)</td>
<td></td>
</tr>
<tr>
<td>ResourceManager Heap</td>
<td>The percentage of ResourceManager JVM heap memory used</td>
</tr>
<tr>
<td>ResourceManager Uptime</td>
<td>The ResourceManager uptime calculation</td>
</tr>
<tr>
<td>NodeManagers Live</td>
<td>The number of DataNodes operating, as reported from the ResourceManager</td>
</tr>
<tr>
<td>YARN Memory</td>
<td>The percentage of available YARN memory (used versus. total available)</td>
</tr>
<tr>
<td>HBase metrics</td>
<td></td>
</tr>
<tr>
<td>HBase Master Heap</td>
<td>The percentage of NameNode JVM heap memory used</td>
</tr>
<tr>
<td>HBase Ave Load</td>
<td>The average load on the HBase server</td>
</tr>
<tr>
<td>HBase Master Uptime</td>
<td>The HBase master uptime calculation</td>
</tr>
<tr>
<td>Region in Transition</td>
<td>The number of HBase regions in transition</td>
</tr>
<tr>
<td>Storm metrics (HDP 2.1 or later stacks)</td>
<td></td>
</tr>
<tr>
<td>Supervisors Live</td>
<td>The number of supervisors operating as reported by the Nimbus server</td>
</tr>
</tbody>
</table>

**Scan operating status**

Expand Services > to see top-level metrics about service operating status.

**About this task**

Services v on the left side of Ambari Web lists all of the Apache component services that are currently installed and monitored.
The icon shape, color, and action to the left of each item indicates the operating status of that service.

**Table 1: Status Indicators**

<table>
<thead>
<tr>
<th>Color</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>no icon</td>
<td>All masters are running.</td>
</tr>
<tr>
<td>blinking green</td>
<td>Starting up</td>
</tr>
<tr>
<td>solid red</td>
<td>At least one master is down.</td>
</tr>
<tr>
<td>blinking red</td>
<td>Stopping</td>
</tr>
</tbody>
</table>

**Procedure**

- Use these icons to scan current operating status for the cluster.
- Hover your cursor over the restart indicator to scan the status of stale components.
  - Click Services v to collapse Ambari Web > Services.
Monitor service metrics

Use the widgets on Dashboard Metrics to monitor cluster-wide metrics.

About this task

On the Metrics page, multiple widgets represent operating status information of your cluster. Most widgets display a single metric, such as HDFS Disk Usage:

Procedure

• Hover your cursor over a Metrics widget.
  A pop up window displays more details, if available.

  ![HDFS Disk Usage](image)

• Click the edit widget icon to export, modify or remove a widget from the Metrics page.

• For cluster-wide metrics, such as Memory, Network or CPU Usage, click Save as CSV or Save as JSON to export metrics data.

• Click Edit to modify the display of information in a widget.

• Click Delete to remove the widget from the Dashboard.

View cluster health

Filter Dashboard > Metrics to see only metrics that reflect overall cluster health.
About this task

From **Dashboard > Metrics**, you can filter the dashboard to display only cluster-wide metrics. For example, you can customize the dashboard to hide all except the following widgets:

![Dashboard Metrics](image)

Additional options include:

**Procedure**

- From a widget legend, click **Delete** to remove the widget from **Metrics**. Deleting a widget from Metrics hides the widget from view. To replace it, add the widget, using Metric Actions.
- For cluster-wide metrics, hover the cursor over the chart, then click the magnifying glass icon to zoom in.
- Hover your cursor over a widget to magnify the chart or itemize the widget display. A larger version of the widget displays in a separate window. You can use the larger view in the same ways that you use the dashboard.
- Click **OK** to close the larger view.

**Example**

![Cluster Load](image)
Example
Cluster-wide metrics include:

- **Memory usage**: Cluster-wide memory used, including memory that is cached, swapped, used, and shared
- **Network usage**: The cluster-wide network utilization, including in-and-out
- **CPU Usage**: Cluster-wide CPU information, including system, user and wait IO
- **Cluster Load**: Cluster-wide Load information, including total number of nodes, total number of CPUs, number of running processes and 1-min Load

View cluster heatmaps

The Ambari Web Heatmaps page provides a graphical representation of your overall cluster utilization, using simple color coding known as a heatmap.

**About this task**
A colored block represents each host in your cluster. Colors displayed in the block represent usage in a unit appropriate for the selected set of metrics.

**Procedure**
- Hovering over a block that represents a host displays more detailed information about that specific host. A separate window displays metrics about components installed on that host.

Example

If any data necessary to determine usage is not available, the block displays Invalid data.
Example

You can select a different heatmap, using the Select Metric menu.

Example

Heatmaps supports the following metrics:

<table>
<thead>
<tr>
<th>Metric</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host/Disk Space Used %</td>
<td>Host/Disk Space Used %</td>
</tr>
<tr>
<td>Host/Memory Used %</td>
<td>Host/Memory Used %</td>
</tr>
<tr>
<td>Host/CPU Wait I/O %</td>
<td>Host/CPU Wait I/O %</td>
</tr>
<tr>
<td>HDFS/Bytes Read</td>
<td>HDFS/Bytes Read</td>
</tr>
<tr>
<td>HDFS/Bytes Written</td>
<td>HDFS/Bytes Written</td>
</tr>
<tr>
<td>HDFS/Garbage Collection Time</td>
<td>HDFS/Garbage Collection Time</td>
</tr>
<tr>
<td>HDFS/JVM Heap MemoryUsed</td>
<td>HDFS/JVM Heap MemoryUsed</td>
</tr>
<tr>
<td>YARN/Garbage Collection Time</td>
<td>YARN/Garbage Collection Time</td>
</tr>
<tr>
<td>YARN / JVM Heap Memory Used</td>
<td>YARN / JVM Heap Memory Used</td>
</tr>
<tr>
<td>YARN / Memory used %</td>
<td>YARN / Memory used %</td>
</tr>
<tr>
<td>HBase/RegionServer read request count</td>
<td>HBase/RegionServer read request count</td>
</tr>
<tr>
<td>HBase/RegionServer write request count</td>
<td>HBase/RegionServer write request count</td>
</tr>
<tr>
<td>HBase/RegionServer compaction queue size</td>
<td>HBase/RegionServer compaction queue size</td>
</tr>
<tr>
<td>HBase/RegionServer regions</td>
<td>HBase/RegionServer regions</td>
</tr>
<tr>
<td>HBase/RegionServer memstore sizes</td>
<td>HBase/RegionServer memstore sizes</td>
</tr>
</tbody>
</table>

Finding current stack and versions information

Use the Stack and Versions tabs add services to the stack, and register new version information before performing an upgrade.
About this task

The **Stack** tab includes information about the services installed and available in the cluster stack. The **Versions** tab includes information about which versions of software are currently installed and running in the cluster.

Procedure

- In Ambari Web, browse to **Cluster Admin > Stack and Versions**
- Click **Stack** to browse all services installed in your current stack.
- As an Ambari administrator you can click **Add Service** to start the wizard that installs each service into your cluster.
<table>
<thead>
<tr>
<th>Service</th>
<th>Version</th>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDFS</td>
<td>2.7.3</td>
<td>Installed</td>
<td>Apache Hadoop Distributed File System</td>
</tr>
<tr>
<td>YARN</td>
<td>2.7.3</td>
<td>Installed</td>
<td>Apache Hadoop NextGen MapReduce (YARN)</td>
</tr>
<tr>
<td>MapReduce2</td>
<td>2.7.3</td>
<td>Installed</td>
<td>Apache Hadoop NextGen MapReduce (YARN)</td>
</tr>
<tr>
<td>Tez</td>
<td>0.7.0</td>
<td>Installed</td>
<td>Tez is the next generation Hadoop Query Processing framework written on top of YARN.</td>
</tr>
<tr>
<td>Hive</td>
<td>1.2.1000</td>
<td>Installed</td>
<td>Data warehouse system for ad-hoc queries &amp; analysis of large datasets and table &amp; storage management service</td>
</tr>
<tr>
<td>HBase</td>
<td>1.1.2</td>
<td>Installed</td>
<td>A Non-relational distributed database, plus Phoenix, a high performance SQL layer for low latency applications.</td>
</tr>
<tr>
<td>Pig</td>
<td>0.16.0</td>
<td>Installed</td>
<td>Scripting platform for analyzing large datasets</td>
</tr>
<tr>
<td>Snoop</td>
<td>1.4.6</td>
<td>Installed</td>
<td>Tool for transferring bulk data between Apache Hadoop and structured data stores such as relational databases</td>
</tr>
<tr>
<td>Oozie</td>
<td>4.2.0</td>
<td>Installed</td>
<td>System for workflow coordination and execution of Apache Hadoop jobs. This also includes the installation of the optional Oozie Web Console which relies on and will install the ExtJIS Library.</td>
</tr>
<tr>
<td>ZooKeeper</td>
<td>3.4.6</td>
<td>Installed</td>
<td>Centralized service which provides highly reliable distributed coordination</td>
</tr>
<tr>
<td>Falcon</td>
<td>0.10.0</td>
<td>Add Service</td>
<td>Data management and processing platform</td>
</tr>
<tr>
<td>Storm</td>
<td>1.0.1</td>
<td>Add Service</td>
<td>Apache Hadoop Stream processing framework</td>
</tr>
<tr>
<td>Flume</td>
<td>1.5.2</td>
<td>Add Service</td>
<td>A distributed service for collecting, aggregating, and moving large amounts of streaming data into HDFS</td>
</tr>
<tr>
<td>Accumulo</td>
<td>1.7.0</td>
<td>Add Service</td>
<td>Robust, scalable, high performance distributed key/value store.</td>
</tr>
<tr>
<td>Ambari Infra</td>
<td>0.1.0</td>
<td>Installed</td>
<td>Core shared service used by Ambari managed components.</td>
</tr>
<tr>
<td>Ambari Metrics</td>
<td>0.1.0</td>
<td>Installed</td>
<td>A system for metrics collection that provides storage and retrieval capability for metrics collected from the cluster</td>
</tr>
<tr>
<td>Atlas</td>
<td>0.7.0</td>
<td>Add Service</td>
<td>Atlas Metadata and Governance platform</td>
</tr>
<tr>
<td>Kafka</td>
<td>0.10.0</td>
<td>Add Service</td>
<td>A high-throughput distributed messaging system</td>
</tr>
<tr>
<td>Knox</td>
<td>0.9.0</td>
<td>Add Service</td>
<td>Provides a single point of authentication and access for Apache Hadoop services in a cluster</td>
</tr>
<tr>
<td>Log Search</td>
<td>0.5.0</td>
<td>Add Service</td>
<td>Log aggregation, analysis, and visualization for Ambari managed services. This service is Technical Preview.</td>
</tr>
<tr>
<td>Ranger</td>
<td>0.6.0</td>
<td>Add Service</td>
<td>Comprehensive security for Hadoop</td>
</tr>
<tr>
<td>Ranger KMS</td>
<td>0.6.0</td>
<td>Add Service</td>
<td>Key Management Server</td>
</tr>
<tr>
<td>SmartSense</td>
<td>1.3.0.0-50</td>
<td>Installed</td>
<td>SmartSense - Hortonworks SmartTool (HST) helps quickly gather configuration, metrics, logs from common HDP services that aids to quickly troubleshoot support cases and receive cluster-specific recommendations.</td>
</tr>
<tr>
<td>Spark</td>
<td>1.6.2</td>
<td>Add Service</td>
<td>Apache Spark is a fast and general engine for large-scale data processing.</td>
</tr>
<tr>
<td>Spark2</td>
<td>2.0.0</td>
<td>Add Service</td>
<td>Apache Spark 2.0 is a fast and general engine for large-scale data processing. This service is Technical Preview.</td>
</tr>
<tr>
<td>Zeppelin Notebook</td>
<td>0.6.0</td>
<td>Add Service</td>
<td>A web-based notebook that enables interactive data analytics. It enables you to make beautiful data-driven, interactive and collaborative documents with SQL, Scala and more.</td>
</tr>
<tr>
<td>Kerberos</td>
<td>1.10.3-10</td>
<td>Add Service</td>
<td>A computer network authentication protocol which works on the basis of ‘tickets’ to allow nodes communicating over a non-secure network to prove their identity to one another in a secure manner.</td>
</tr>
<tr>
<td>Mahout</td>
<td>0.9.0</td>
<td>Add Service</td>
<td>Project of the Apache Software Foundation to produce free implementations of distributed or otherwise scalable machine learning algorithms focused primarily in the areas of collaborative filtering, clustering and classification</td>
</tr>
<tr>
<td>Slider</td>
<td>0.91.0</td>
<td>Installed</td>
<td>A framework for deploying, managing and monitoring existing distributed applications on YARN.</td>
</tr>
</tbody>
</table>

- Click **Versions** to see specific version information about each service.
- As an Ambari administrator, click register version to initiate an automated cluster upgrade from the **Versions** page.
Viewing service account user names

As a cluster administrator, you can view the list of Service Users and Group accounts used by the cluster services.

Procedure

• In Ambari Web > Cluster Admin, click Service Accounts.
### Service Users and Groups

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ambari Metrics User</td>
<td>ams</td>
</tr>
<tr>
<td>Metadata User</td>
<td>atlas</td>
</tr>
<tr>
<td>Smoke User</td>
<td>ambari-qa</td>
</tr>
<tr>
<td>Hadoop Group</td>
<td>hadoop</td>
</tr>
<tr>
<td>HDFS User</td>
<td>hdfs</td>
</tr>
<tr>
<td>Proxy User Group</td>
<td>users</td>
</tr>
<tr>
<td>Hive User</td>
<td>hive</td>
</tr>
<tr>
<td>Infra Solr User</td>
<td>infra-solr</td>
</tr>
<tr>
<td>Kafka User</td>
<td>kafka</td>
</tr>
<tr>
<td>Kms group</td>
<td>kms</td>
</tr>
<tr>
<td>Kms User</td>
<td>kms</td>
</tr>
<tr>
<td>Knox Group</td>
<td>Knox</td>
</tr>
<tr>
<td>Knox User</td>
<td>Knox</td>
</tr>
<tr>
<td>Livy2 Group</td>
<td>Livy</td>
</tr>
<tr>
<td>Livy2 User</td>
<td>Livy</td>
</tr>
<tr>
<td>Log Search User</td>
<td>Logsearch</td>
</tr>
<tr>
<td>Mapreduce User</td>
<td>Mapred</td>
</tr>
<tr>
<td>Oozie User</td>
<td>Oozie</td>
</tr>
<tr>
<td>Ranger Group</td>
<td>Ranger</td>
</tr>
<tr>
<td>Ranger User</td>
<td>Ranger</td>
</tr>
<tr>
<td>Spark2 Group</td>
<td>Spark</td>
</tr>
<tr>
<td>Spark2 User</td>
<td>Spark</td>
</tr>
<tr>
<td>Tez User</td>
<td>Tez</td>
</tr>
<tr>
<td>Yarn ATS User</td>
<td>Yarn-ats</td>
</tr>
</tbody>
</table>
Modifying the cluster dashboard

You can modify the content of the Ambari Cluster dashboard in the following ways:

Replace a removed widget

Replace metrics widgets not currently displayed on the Metrics page using Metrics Actions > +Add.

About this task
You can remove and replace metrics widgets on the Metrics page. Metrics widgets that have been removed from the Metrics page appear in a list you can access from the Metrics Actions menu, using the +Add option.

Procedure
1. On the Metrics page, click Metrics Actions > +Add.
2. Click the checkbox next to metrics that you want to appear on the Metrics page. For example, click Region in Transition.
3. Click Add.
   The Region in Transition widget appears on the Metrics page.

Reset the dashboard

Reset the Metrics page to its default settings, using Metrics Actions > Edit.

About this task
If you have customized the Metrics page by removing widgets or changing display settings, you may reset the default display settings.

Procedure
1. On the Metrics page, click Metrics Actions > Edit.
2. Click Reset all widgets to default.
   The Metrics page resets to its default settings.
Customize metrics display

Change the way in which a widget displays metrics information, using the Edit icon, if available.

About this task
Although not all widgets can be edited, you can customize the way that some of them display metrics by using the Edit icon, if one is displayed.

Procedure
1. On the Metrics page, click the edit widget icon
   ![](image)
   to customize or remove a widget from the Metrics page.
2. Click Edit.
   The Customize Widget window appears.
3. Follow the instructions in Customize Widget to customize widget appearance.
   In this example, you can adjust the thresholds at which the HDFS Capacity bar chart changes color, from green to orange to red.
4. To save your changes and close the editor, click Apply.
5. To close the editor without saving any changes, click Cancel.

Managing Hosts

Mange one or multiple hosts in your cluster from the Hosts page in Ambari Web.

About this task
As a cluster administrator or cluster operator, you need to react when the operating status of a host indicates issues that require action. You can use the Ambari Web Hosts page to manage multiple components, such as DataNodes, NameNodes, NodeManagers, and RegionServers, running on hosts throughout your cluster. For example, you can restart all DataNode components, optionally controlling that task with rolling restarts. Ambari Hosts enables you to filter your selection of host components to manage host health based on operating status and defined host groupings. The Hosts tab enables you to perform the following tasks:

View host status

You can view the individual hosts in your cluster on Ambari Web > Hosts.
About this task

Hosts lists cluster hosts by fully qualified domain name (FDQN) and accompanied by a colored icon that indicates the host’s operating status.

Procedure

• Observe the icon beside each listed host name.

Example

Red Triangle
At least one master component on that host is down. You can hover your cursor over the host name to see a tooltip that lists affected components.

Orange
At least one slave component on that host is down. Hover to see a tooltip that lists affected components.

Yellow
Ambari Server has not received a heartbeat from that host for more than 3 minutes.

Green
Normal running state.

Maintenace Mode
Black, medical-bag icon indicates a host in maintenance mode.

Alert
Red square with white number indicates the number of alerts generated on a host.

Example

A red icon overrides an orange icon, which overrides a yellow icon. In other words, a host that has a master component down is accompanied by a red icon, even though it might have slave component or connection issues as well. Hosts in maintenance mode or are experiencing alerts, are accompanied by an icon to the right of the host name.

The following example Hosts page shows three hosts, one having a master component down, one having a slave component down, one running normally, and two with alerts:

Find a host in the cluster

Use a filtered search to find only those hosts in the cluster that match specific criteria.

About this task

The Hosts page lists all hosts in the cluster. You can search the full list of hosts, filtering your search by host name, component attribute, and component operating status. You can also search by keyword, simply by typing a word in the search box. The filtered search tool appears left of the Actions menu on the Hosts page.

Procedure

1. On Ambari Web > Hosts, click the search icon.
A filtered search field appears above the listed hosts.

2. Click the filtered search field.

Available search types appear, including:

**Search by Host Attribute**

Search by host name, IP, host status, and other attributes, including:

![Host Attributes]

**Search by Service**

Find hosts that are hosting a component from a given service.

**Search by Component**

Find hosts that are hosting components in a given state, such as started, stopped, maintenance mode, and so on.

**Search by keyword**

Type any word that describes what you are looking for in the search box. This becomes a text filter.

3. Click a Search type.

A list of available options appears, depending on your first selection. For example, if you click Service, current services appear:
4. Click one or multiple options.
   In this example, Service: YARN, then Host Status: Master Down.
   The list of hosts that match your current search criteria display on the Hosts page.

5. You can click more options or type keywords to further refine your search.

   **Example**

   Examples of searches that you can perform, based on specific criteria, and which interface controls to use:

   - **Find all hosts with a DataNode**

   - **Find all the hosts with a DataNode that are stopped**

   - **Find all the hosts with an HDFS component**

   - **Find all the hosts with an HDFS or HBase component**

   **Perform host level actions**

   Use **Hosts > Actions** options to start bulk operations across multiple hosts in your cluster.
About this task
Use the Actions UI control to act on hosts in your cluster. Actions that you perform that comprise more than one operation, possibly on multiple hosts, are also known as bulk operations. The Actions control comprises a workflow that uses a sequence of three menus to refine your search: a hosts menu, a menu of objects based on your host choice, and a menu of actions based on your object choice. For example, if you want to restart the RegionServers on any host in your cluster on which a RegionServer exists:

Procedure
1. In the Hosts page, select or search for hosts running a RegionServer:
2. Using the Actions control, click Filtered Hosts > RegionServers > Restart >
3. Click OK to start the selected operation.

What to do next
Optionally, monitor background operations to follow, diagnose, or troubleshoot the restart operation.

Related Information
Monitor background operations

Add a host to a cluster
Use the +Add New Hosts wizard to add new host machines to your cluster.

Procedure
1. Browse to the Hosts page and select Actions > +Add New Hosts. The Add Host wizard provides a sequence of prompts similar to those in the Ambari Cluster Install wizard.
2. Follow the prompts, providing information similar to that provided to define the first set of hosts in your cluster.
What to do next
Review and confirm all recommended configuration changes.

Related Information
Review and confirm recommended configuration changes
Install options

Manage components on a host
Click a host FQDN on the Hosts page to manage components running on that host.

Procedure
• Click one of the FQDNs listed on the Hosts page.
  For example, if you click c7404.ambari.apache.org, that host's page appears.
• Click Summary.
  Components lists all components installed on that host.
• To manage all of the components on a single host, you can use the **Host Actions** control at the top right of the display to start, stop, restart, delete, or turn on maintenance mode for all components installed on the selected host.

• Alternatively, you can manage components individually, by using the **Action** drop-down menu shown next to an individual component in the **Components** pane. Each component's menu is labeled with the component's current operating status. Opening the menu displays your available management options, based on that status.

  For example, you can decommission, restart, stop, or put in maintenance mode the standby NameNode component for HDFS.
Decommission a host

Decommissioning is a process that supports removing components and their hosts from the cluster.

You must decommission a master or slave running on a host before removing it or its host from service.

Decommissioning helps you to prevent potential loss of data or disruption of service. Decommissioning is available for the following component types:

- DataNodes
- NodeManagers
- RegionServers

Decommissioning executes the following tasks:

**For DataNodes**

Safely replicates the HDFS data to other DataNodes in the cluster

**For NodeManagers**

Stops accepting new job requests from the masters and stops the component

**For RegionServers**

Turns on drain mode and stops the component

Decommission a NodeManager

Use Ambari Web > Hosts > Actions to decommission a NodeManager component.

Procedure

1. Using Ambari Web, browse the Hosts page.
2. Find and click the FQDN of the host on which the NodeManager component resides.
3. Using the Actions menu, click Selected Hosts > NodeManagers > Decommission.
The UI shows Decommissioning status while in process.

**Results**
When this NodeManager decommissioning process is finished, the Action option shows Recommission.

**Decommission a RegionServer**

Use [Ambari Web](https://ambari.apache.org/) > [Hosts](https://ambari.apache.org/) > [Actions](https://ambari.apache.org/) to decommission a RegionServer component.

**Procedure**
1. Using Ambari Web, browse the Hosts page.
2. Find and click the FQDN of the host on which the RegionServer component resides.
3. Using the Actions menu, click **Selected Hosts** > **RegionServers** > **Decommission**.
Ambari

Managing Hosts

The UI shows Decommissioning status while in process.

Results
When this RegionServer decommissioning process is finished, the Action option shows Recommission.

Decommission a DataNode
Use Ambari Web > Hosts > Actions to decommission a DataNode component.

Procedure
1. Using Ambari Web, browse the Hosts page.
2. Find and click the FQDN of the host on which the DataNode component resides.
3. Using the Actions control, click Selected Hosts > DataNodes > Decommission.
Ambari

Managing Hosts

The UI shows Decommissioning status while in process.

Results
When this DataNode decommissioning process is finished, the Action option shows Recommission.

Delete a component

Use Component > Action to delete a decommissioned component:

Before you begin
Decommission the component that you intend to delete.

Procedure
1. Using Ambari Web, browse the Hosts page.
2. Find and click the FQDN of the host on which the component resides.
3. In Components, find a decommissioned component.
4. If the component status is Started, stop it.
   A decommissioned slave component may restart in the decommissioned state.
5. Click Delete from the Action drop-down menu.
Deleting a slave component, such as a DataNode does not automatically inform a master component, such as a NameNode, to remove the slave component from its exclusion list. Adding a deleted slave component back into the cluster presents the following issue: The added slave remains decommissioned from the master's perspective. Restart the master component, as a work-around.

6. Restart services.

Results
Ambari recognizes and monitors only the remaining components.

Delete a host from a cluster

Use Host Actions > Delete Host to remove a host from the cluster.

About this task
Deleting a host removes the host from the cluster.

Before you begin
Before deleting a host, you must:

• Stop all components running on the host.
• Decommission any DataNodes running on the host.
• Move from the host any master components, such as NameNode or ResourceManager, running on the host.
• Turn off host Maintenance Mode, if it is on.

Procedure
1. Using Ambari Web, browse the hosts page to find and click the FQDN of the host that you want to delete.
2. On the Host page, click Host Actions.
3. Click Delete Host.

The host is removed from the cluster.
Bulk add or delete hosts

Use the Hosts > Actions > Delete Hosts option to delete multiple hosts in your cluster.

Before you begin
Decommission and stop all components on the target hosts. There are guardrails in place to prevent the accidental deletion of hosts containing singleton master instances. To delete such hosts, first move all the masters to a different host.

About this task
Use the Actions UI control to act on hosts in your cluster. Actions that you perform that comprise more than one operation, possibly on multiple hosts, are also known as bulk operations. After you request that Ambari perform a bulk operation, Ambari validates whether or not it can successfully perform the action. There are two possible scenarios where the operation cannot be performed:

- The state of the component makes it impossible to perform the operation. For example, asking to delete service components not in the stopped state. Expand each host to see the details on why the operation cannot be performed.

- Number of components remaining after the operation completes is less than the minimum number of components required by the stack. For example, asking to delete all the nodemanagers, where the minimum number of nodemanagers required by the service is 1.

Procedure
1. In the Hosts page, select or search for multiple hosts.
2. Using the Actions control, click Selected Hosts > Hosts > Delete Hosts.
3. Click **OK** to confirm the bulk operation.

**What to do next**
Optionally, monitor background operations to follow, diagnose, or troubleshoot the bulk operation.

**Related Information**
Review and confirm recommended configuration changes

### Setting Maintenance Mode

Set Maintenance Mode to suppress alerts when performing hardware or software maintenance, changing configuration settings, troubleshooting, decommissioning, or removing cluster nodes.

Setting Maintenance Mode enables you to suppress alerts and omit bulk operations for specific services, components, and hosts in an Ambari-managed cluster. Explicitly setting Maintenance Mode for a service implicitly sets Maintenance Mode for components and hosts that run the service. While Maintenance Mode prevents bulk operations being performed on the service, component, or host, you may explicitly start and stop a service, component, or host while in Maintenance Mode.

Four common instances in which you might want to set Maintenance Mode are; to perform maintenance, to test a configuration change, to delete a service completely, and to address alerts. Specific use-case examples:

<table>
<thead>
<tr>
<th>You want to perform hardware, firmware, or OS maintenance on a host.</th>
<th>You want to test a service configuration change.</th>
</tr>
</thead>
<tbody>
<tr>
<td>While performing maintenance, you want to be able to do the following:</td>
<td></td>
</tr>
<tr>
<td>• Prevent alerts generated by all components on this host.</td>
<td></td>
</tr>
<tr>
<td>• Be able to stop, start, and restart each component on the host.</td>
<td></td>
</tr>
<tr>
<td>• Prevent host-level or service-level bulk operations from starting, stopping, or restarting components on this host.</td>
<td></td>
</tr>
<tr>
<td>To achieve these goals, explicitly set Maintenance Mode for the host. Putting a host in Maintenance Mode implicitly puts all components on that host in Maintenance Mode.</td>
<td></td>
</tr>
<tr>
<td>To test configuration changes, you want to ensure the following conditions:</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td></td>
</tr>
</tbody>
</table>
• No alerts are generated by any components in this service.
• No host-level or service-level bulk operations start, stop, or restart components in this service.

To achieve these goals, explicitly set Maintenance Mode for the service. Putting a service in Maintenance Mode implicitly turns on Maintenance Mode for all components in the service.

You will stop, start, and restart the service using a rolling restart to test whether restarting activates the change.

You want to stop a service.

To stop a service completely, you want to ensure the following conditions:
• No warnings are generated by the service.
• No components start, stop, or restart due to host-level actions or bulk operations.

To achieve these goals, explicitly set Maintenance Mode for the service. Putting a service in Maintenance Mode implicitly turns on Maintenance Mode for all components in the service.

You want to stop a host component from generating alerts.

To stop a host component from generating alerts, you must be able to do the following:
• Check the component.
• Assess warnings and alerts generated for the component.
• Prevent alerts generated by the component while you check its condition.

Set Maintenance Mode for a host

How to set Maintenance Mode for a host and what happens when you do.

About this task
This task provides examples of how to use Maintenance Mode in a three-node, Ambari-managed cluster installed using default options and having one data node, on host c7403. The example describes how to explicitly turn on Maintenance Mode for the HDFS service, alternative procedures for explicitly turning on Maintenance Mode for a host, and the implicit effects of turning on Maintenance Mode for a service, a component, and a host.

Procedure

1. On the Hosts page, select a host FQDN.
   c7401.ambari.apache.org

2. In Host Actions, click Turn On Maintenance Mode.

3. Click OK to confirm.

   Notice, on Components, that Maintenance Mode turns on for all components.

4. To set Maintenance Mode for a multiple hosts, using the Actions control.


6. In Actions > Selected Hosts > Hosts, click Turn On Maintenance Mode.
7. Choose OK.

Results
Your list of hosts shows that Maintenance Mode is set for hosts c7401 and c7403:

Example
If you hover your cursor over each Maintenance Mode icon that appears in the hosts list, you see the following information:

- Hosts c7401 and c7403 are in Maintenance Mode.
- On host c7401, HBaseMaster, HDFS client, NameNode, and ZooKeeper Server are also in Maintenance Mode.
- On host c7403, 15 components are in Maintenance Mode.
- On host c7402, HDFS client and Secondary NameNode are in Maintenance Mode, even though the host is not.

Notice also how the DataNode is affected by setting Maintenance Mode on this host:

- Alerts are suppressed for the DataNode.
- DataNode is omitted from HDFS Start/Stop/Rolling Restart.
- DataNode is omitted from all Bulk Operations except Turn Maintenance Mode ON/OFF.
- DataNode is omitted from Start All and / Stop All components.
- DataNode is omitted from a host-level restart/restart all/stop all/start.

Set Maintenance Mode for a service
How to set Maintenance Mode for a service and what happens when you do.

About this task
Setting Maintenance Mode enables you to suppress alerts and omit bulk operations for specific services, components, and hosts in an Ambari-managed cluster. Explicitly setting Maintenance Mode for a service implicitly sets
Establishing Rack Awareness

The term Rack Awareness means defining the physical rack on which a cluster host resides. Establishing Rack awareness can increase availability of data blocks and improve cluster performance. Co-locating data replication blocks on one physical rack speeds replication operations. The HDFS balancer and DataNode decommissioning are rack-aware operations. Rack awareness is not established by default, when a cluster is deployed or a new host is added, using Ambari. Instead the entire cluster is assigned to one, default rack.

You can establish rack awareness in two ways. Either you can set the rack ID using Ambari or you can set the rack ID using a custom topology script.

Set the rack id for Ambari

About this task
By setting the Rack ID, you can enable Ambari to manage rack information for hosts, including displaying the hosts in heatmaps by Rack ID, enabling users to filter and find hosts on the Hosts page, using that Rack ID. If HDFS is installed in your cluster, Ambari passes this Rack ID information to HDFS using a topology script. Ambari generates a topology script at /etc/hadoop/conf/topology.py and sets the net.topology.script.file.name property in core-site automatically. This topology script reads a mappings file /etc/hadoop/conf/topology_mappings.data that Ambari automatically generates. When you make changes to Rack ID assignment using Ambari, this mappings file will be updated when you push out the HDFS configuration. HDFS uses this topology script to obtain Rack information about the DataNode hosts.

Procedure
You can set the Rack ID using Ambari Web in two ways.

- Make multiple hosts aware of a rack simultaneously, using Actions.
- Make each host aware of a rack individually, using Host Actions.

1. To set the Rack ID for multiple hosts, from the Hosts page, click Actions, then click Selected Hosts, Filtered Hosts, or All Hosts.
2. Expand the menu, and click Hosts.
3. Then, expand the menu and click Set Rack.
4. Alternatively, for a specific host, from the Hosts page, click Host Actions > Set Rack.
5. In Set Rack, type the rack name, then click OK.

What to do next
Browse to Ambari Web > Dashboard, and restart any services that display the Restart Required icon.

Set the rack id on a host

Procedure
1. In Ambari Web, browse to the Hosts page.
2. Click Host Actions.
3. Click Set Rack.
What to do next
Verify rack awareness.

Set the rack id using a custom topology script

About this task
If you do not want to have Ambari manage the rack information for hosts, you can use a custom topology script. To do this, you must create your own topology script and manage distributing the script to all hosts.

Note: Ambari Web will not display heatmaps by rack, because Ambari will have no access to host rack information.

Procedure
1. Browse to Services > HDFS > Configs.
2. Modify net.topology.script.file.name to your own custom topology script.
   For example: /etc/hadoop/conf/topology.sh.
3. Distribute that topology script to your hosts.

What to do next
You can now manage the rack mapping information for your script outside of Ambari.

Managing Services

You use Ambari Web > Services to monitor and manage selected services running in your Hadoop cluster.
All services installed in your cluster are listed in the leftmost panel:

![Ambari Services](image)

**View service summary**

Use **Services > Summary** to view the status of service components and metrics.

**About this task**

In Ambari Web, the **Services > Summary** pane displays basic information about the operational status of that service, including any current alerts.

**Procedure**

1. In **Ambari Web**, click a service name from the list of displayed services.
   
   In **Summary**, basic information about the operational status of that service, including any alerts displays.
2. In Ambari Web, click a different service name from the list of displayed services. Summary refreshes to display information about the different service.

What to do next

click one of the View Host links.

to view components and the host FQDN on which the selected service is running.

Find quick links to service information

Use Services > [SERVICE_NAME] > Summary > Quick Links to find more information for each service.
About this task
The Summary tab for each service lists Quick Links to more information, such as thread stacks, logs, and native component UIs. For example, you can link to NameNode UI, NameNode Logs, NameNode JMX and Thread Stacks for HDFS.

Procedure
- Browse the Quick Links listed at the right side of Summary to select from the list of links available for each service.

Example

Link to the native user interface
Select Quick Links options to access additional sources of information about a selected service.

About this task
Some services provide a native user interface. For most services, Ambari Web provides links to the native UI and to other information such as component log files and thread stacks.

Procedure
1. For a service, browse to Services > Summary > Quick Links.
   - Quick Links are not available for every service.
2. On Quick Links, click a native user interface name.

Example
For example, HDFS Quick Links options include the following:
Add or remove a service widget

Use options on **Ambari Web > Services > Metrics** to manage the metrics visible for a specific service.

**About this task**
The **Ambari Web > Services > Metrics** tab displays metrics information using multiple widgets. You can add and remove widgets from the **Metrics** tab. HDP core services currently support this functionality, including HDFS, Hive, HBase, and YARN services.
Procedure

1. On the Metrics tab, either click + to launch the Widget Browser, or click Actions > Browse Widgets. The Widget Browser displays service widgets available to add to your Services tab, widgets already added to your dashboard, shared widgets, and widgets you have created. Widgets that are shared are identified by the Shared icon.

![Widget Browser Image]

2. In the Widget Browser, click the Show only my widgets check box. Only the widgets you have created display on the Metrics tab.

3. In the Widget Browser, click a green, ADDED checkmark for any widget that displays one. The widget disappears from the the Metrics tab and no longer shows a green, ADDED checkmark in the Widget Browser.

4. In the Widget Browser, for any available widget not already added to your Metrics tab, click Add. The widget appears on the the Metrics tab and shows a green, ADDED checkmark in the Widget Browser.

What to do next
To remove a widget from the Metrics tab, click the edit widget icon, then click Delete.

Create a service widget

Use the Widget Browser and the Create Widget wizard to create new widgets that display metrics for a service.

About this task
The Widget Browser displays the widgets available to add to your Services tab, widgets already added to your dashboard, shared widgets, and widgets you have created. Widgets that are shared are identified by the Shared icon. The Widget Browser also supports creating and deleting new service widgets.
Procedure

1. On the Services tab, either click + to launch the Widget Browser, or click Metrics > Actions menu > Create Widget.

2. If the Widget browser is open, click + Create Widget.
   a) In the Create Widget wizard, select a widget type.
   b) Select metrics and operators to create an expression that defines the metrics to be displayed in the widget.
      A preview of the widget is displayed as you build the expression.
   c) Enter a name and description for the widget.
   d) Optionally, click Share to share the widget.
      Sharing the widget makes the widget available to all Ambari users for this cluster. After a widget is shared, other Ambari Admins or Cluster Operators can modify or delete the widget.
      
      Note: Share cannot be undone.

Delete a service widget

Use the Widget Browser or the edit widget icon options to delete service widgets from a service Metrics tab.

About this task
The Widget Browser displays the widgets available to add to your Metrics tab, widgets already added to your dashboard, shared widgets, and widgets you have created. Multiple users can view widgets identified by the Shared icon. The Widget Browser also supports creating and deleting new service widgets.

Procedure

1. On the Metrics tab, either click + to launch the Widget Browser, or click Metrics > Actions > Create Widget.
The **Widget Browser** displays all current service widgets.

2. In the **Widget Browser**, click a green, Added checkmark for any widget that displays one.
   This removes the widget from the **Metrics** tab. This does not delete the widget from the **Widget Browser**.
   The widget disappears from the **Metrics** tab and no longer shows a green, Added checkmark in the **Widget Browser**.

3. Only for a widget that you created, you can click **More... > Delete**.
   For a shared widget, only an Ambari Admin or Cluster Operator has the **Delete** option. Deleting a shared widget removes the widget from all users. Deleting a shared widget cannot be undone.
   The widget disappears from the **Widget Browser**.

**What to do next**
To remove a widget from the **Metrics** tab, click the edit widget icon

then click **Delete**.

**Export widget graph data**
Use export options on the **Metrics** widget dashboard to save service metrics data locally.

**About this task**
Many of the service metrics widgets support exporting the data represented on the widget graph. You can export the metrics data from those widget graphs using the **Export** option.
Procedure
1. For a widget on Service > Metrics, click the Edit Widget icon
   or click the magnify icon.

2. If the Export icon appears, click it.

3. Click either Save as CSV or Save as JSON.
   A file of the selected format containing metrics data downloads to your local drive.

Set display timezone
Each user can set a locale in User Settings.

About this task
You can set a locale that defines the time zone used for displaying metrics data in widget graphs.

Procedure
1. In the Ambari Web operator menu, click your user name, then click Settings.
2. In Locale > Timezone, click a time zone option.
3. Click Save.

Results
The Ambari Web UI reloads. Graphs display the timezone you have saved.

Modify the service metrics dashboard
Use [SERVICE_NAME] > Metrics to scan metrics data for services that display metrics.
Depending on the service, the Metrics tab includes widgets populated by default with important service metrics information to monitor:
If you have Ambari Metrics service (AMS) installed and are using Apache HDFS, Apache Hive, Apache HBase, or Apache YARN, you can view a Metrics dashboard for that service. You can add and remove widgets from the Metrics dashboard, and you can create new widgets and delete widgets. Widgets can be private to you and your dashboard, or they can be shared in a Widget Browser library. You must have AMS installed to view Metrics for AMS service.

**Related Information**
- Understanding Ambari Metrics Service

**Performing service actions**

Use Service > Actions to manage service operations on your cluster.

**About this task**

Manage a selected service on your cluster by performing service actions.

**Procedure**

1. In the Services tab, click Actions.
2. In Actions, click an option.
   - Available options depend on the service you have selected
   - For example, HDFS service action options include:
Clicking **Turn On Maintenance Mode** suppresses alerts and status indicator changes generated by the service, while allowing you to start, stop, restart, move, or perform maintenance tasks on the service.

### Start all services

Use **Ambari Web > Services > Start All** to start all services at once.

**About this task**

In **Ambari Web > Services** you can start, stop, and restart all listed services simultaneously.

**Procedure**

- In **Services**, click ... and then click **Start All**.
All installed services start.

**Stop all services**

Use Ambari Web > Services > Stop All to stop all services at once.

**About this task**

In Ambari Web > Services you can start, stop, and restart all listed services simultaneously.

**Procedure**

- In Services, click ... and then click Stop All.
All installed services stop.

Add a service

Use Add Service to deploy additional services in your cluster without interrupting operations.

About this task
The Ambari installation wizard installs all available Hadoop services by default. You can choose to deploy only some services initially, and then add other services as you need them. For example, many customers deploy only core Hadoop services initially. The Add Service option enables you to deploy additional services without interrupting operations in your Hadoop cluster. When you have deployed all available services, the Add Service control display dims, indicating that it is unavailable. To add a service, follow the steps shown in this example.

Procedure
1. In Services, click … > +Add Service.
   The Add Service wizard opens.
2. In the Add Service wizard, click Choose Services.
   The Choose Services pane lists those services already added in a shaded background and with checked boxes.
3. In **Choose Services**, click the empty check box next to the service that you want to add. Then, click **Next**. Notice that you can also select all services listed by clicking the checkbox next to the Service table column heading.

The Add Services Wizard indicates hosts on which the master components for a chosen service will be installed. A service chosen for addition shows a grey check mark.

4. In **Assign Masters**, confirm the default host assignment. Alternatively, use the drop-down menu to choose a different host machine to which master components for your selected service will be added. Then, click **Next**.
5. If you are adding a service that requires slaves and clients, in **Assign Slaves and Clients**, accept the default assignment of slave and client components to hosts. Alternatively, select hosts on which you want to install slave and client components (at least one host for the slave of each service being added). Then, click **Next**.
Table 2: Host Roles Required for Added Services

<table>
<thead>
<tr>
<th>Service Added</th>
<th>Host Role Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>YARN</td>
<td>NodeManager</td>
</tr>
<tr>
<td>HBase</td>
<td>RegionServer</td>
</tr>
</tbody>
</table>

6. In **Customize Services**, accept the default configuration properties. If necessary, edit any property values indicated.

a) If necessary, click **Override** to create a configuration group for this service.

b) Make sure all configurations have been addressed.
Then, click **Next**.

7. In **Review**, verify that the configuration settings match your intentions. Then, click **Deploy**.

8. Monitor the progress of installing, starting, and testing the service. When that finishes successfully, click **Next**.
9. Review the summary display of installation results. Then, click Complete.

10. Review and confirm recommended configuration changes.

What to do next
Restart any other components that have stale configurations as a result of adding services.

Related Information
Review and confirm recommended configuration changes

Restart multiple components
Use a rolling restart to restart multiple components.

About this task
When you restart multiple components, use rolling restarts to distribute the task. A rolling restart stops and then starts multiple running slave components, such as DataNodes, NodeManagers, RegionServers, or Supervisors, using a batch sequence. You set rolling restart parameter values to control the number, time between, tolerance for failures, and limits for restarts of many components across large clusters.
**Important:** Rolling restarts of DataNodes should be performed only during cluster maintenance.

### Procedure

1. In **Services**, click a service name.
2. In **Summary**, click a link, such as DataNodes or RegionServers, of any components that you want to restart. The **Hosts** page lists any host names in your cluster on which that component resides.
3. Using the host-level **Actions** menu, click the name of a slave component option. Then, click **Restart**.
4. Review and set values for **Rolling Restart Parameters**.
5. Optionally, reset the flag to restart only components with changed configurations.
6. Click **Trigger Rolling Restart**.

### What to do next

Monitor the progress of the background operations.

### Related Information

**Monitor background operations**

### Set rolling restart parameters

You can adjust the default parameter values before starting a rolling restart operation.

### About this task

When you choose to restart slave components, you should use parameters to control how restarts of components roll. Parameter values based on ten percent of the total number of components in your cluster are set as default values. For example, default settings for a rolling restart of components in a three-node cluster restarts one component at a time, waits two minutes between restarts, proceeds if only one failure occurs, and restarts all existing components that run this service.

### Before you begin

Initiate a rolling restart.
Procedure

• In the **Restart** dialog, enter integer, non-zero values for all parameters.

  **Restart (Batch Size)**
  Number of components to include in each restart batch.

  **Wait (Time)**
  Time (in seconds) to wait between queuing each batch of components.

  **Tolerate up to x failures**
  Total number of restart failures to tolerate, across all batches, before halting the restarts and not queuing batches.

If you trigger a rolling restart of components, the default value of **Only restart components with stale configs** is true.

If you trigger a rolling restart of services, the default value of **Only restart services with stale configs** is false.

• Change the default value of **Only restart components with stale configs**, if necessary.

What to do next
Click **Trigger Rolling Restart**.

Monitor background operations

Use **Background Operations** to monitor tasks during execution and examine log details, post-completion.

About this task
You can use the Background Operations window to monitor progress and completion of a task that comprises multiple operations, such as a rolling restart of components. The Background Operations window opens by default when you run such a task. This topic shows monitoring the progress of a rolling restart as an example.

Procedure

1. In **Background Operations**, click the text that describes each operation to show restart operation progress on each host.
2. After operations complete, you can click an operation name link to view log files and any error messages generated on the selected host.

3. Optionally, click a host name link to view tasks on that host.

4. Optionally, click a task name link to view the task log generated during the operation.
   You can use the Copy, Open, or Host Logs icons located at the upper-right of the [HOST_NAME] dialog to copy, open, or view log information generated during the operation.
This example demonstrates the task log that contains error and output information for the restart NodeManagers task run on host c7403.ambari.apache.org.

**What to do next**
You can also select the check box at the bottom of the Background Operations window to hide the Background Operations dialog during future operations.

### Abort a rolling restart

Use the **Abort Operation** option to prevent future operations in a batch from running.

**About this task**

In-progress batch operations display an **Abort Operation** option when you hover the cursor over the operation displayed in **Background Operations**.

**Procedure**

1. In the active **Background Operations** dialog, hover your cursor over a task in-progress.
   - For example, the in-progress rolling restart batch operation shades blue and displays the Abort operation option when you hover the cursor over it.

2. Click **Abort operation** to stop any future operations in the batch from occurring.
3. Click **OK** to confirm that you want to stop future operations.

### Enable Service Auto Start from Ambari Web

Use **Cluster Admin > Service Auto Start** to control whether components in a stopped state re-start automatically.

**About this task**

As a Cluster Administrator or Cluster Operator, you can enable each service in your stack to re-start automatically. Enabling auto-start for a service causes the ambari-agent to attempt re-starting service components in a stopped state without manual effort by a user. As a first step, you should enable auto-start for the worker nodes in the core Hadoop services, the DataNode and NameNode components in YARN and HDFS, for example. You should also enable auto-start for all components in the SmartSense service. After enabling auto-start, monitor the operating status of your services on the Ambari Web dashboard. Auto-start attempts do not display as background operations. To manage the auto-start status for components in a service:

**Procedure**

1. In **Ambari Web > Cluster Admin**, click **Service Auto Start**.

   ![Ambari Web Service Auto Start](image)

   Auto-Start Settings is enabled by default, but only the Ambari Metrics Collector component is set to auto-start by default.

2. To toggle auto-start settings from Enabled to Disabled and back, click the Auto-Start settings button.

3. To set a component to restart automatically, click the **Auto-Start** checkbox for a component.

4. To set all components to auto-start, click the **Auto Start** checkbox.
5. To clear all pending status changes before saving them, click **Cancel**.
6. When you complete changes to your auto-start settings, click **Save**.

**What to do next**
To diagnose issues with service components that fail to start, check the ambari agent logs, located at: /var/log/ambari-agent.log on the component host.

### Disable service auto start settings from Ambari Web

Disable Auto Start Settings to prevent components in a stopped state in your cluster from re-starting automatically.

**About this task**
As a Cluster Administrator or Cluster Operator, you can enable each component in your cluster to re-start automatically. Enabling auto-start for a service causes the ambari-agent to attempt re-starting service components in a stopped state without manual effort by a user. Auto-start attempts do not display as background operations. To disable Auto-Start Services:

**Procedure**
1. In **Ambari Web > Cluster Admin**, click **Service Auto Start**.
Auto-Start Settings is enabled by default, but only the Ambari Metrics Collector component is set to auto-start by default.

2. In **Auto-Start Settings**, click the green **Enabled** button.

   ![Auto Start settings button toggles to Disabled.](image)

3. To clear all pending status changes before saving them, click **Cancel**.

4. When you complete changes to your auto-start settings, click **Save**.
Remove a service

Use Services > Actions > Delete Service to remove a service from your cluster.

About this task

**Important:** Removing a service is not reversible and all configuration history will be lost.

Procedure

1. In Ambari Web, click a service name listed in Services.
2. Click Actions > Delete Service.
3. Remove any dependent services, as prompted.
4. Stop all components for the service, as prompted.
5. Confirm the removal.
   
   After the service is stopped, you must confirm the removal to proceed.

Bulk add or delete service components

Use the Hosts > Actions > [COMPONENT_NAME] > Delete option to delete multiple components in your cluster.

Before you begin

Before adding new components, make sure all hosts have sufficient memory and disk space.

Before deleting service components, you must stop them. Services that support decommissioning should be first decommissioned, then stopped, then deleted.

**Note:**

Warning!! Bulk-deleting services that store data, such as datanodes, regionservers, without decommissioning them first may result in data loss. Deleting more datanodes than the configured dfs.replication.factor, without first decommissioning them completely, will result in permanent data loss.

About this task

Use the Actions UI control to act on hosts in your cluster. Actions that you perform that comprise more than one operation, possibly on multiple hosts, are also known as bulk operations. The bulk component management behavior is similar to individual service component management, so usual best practices apply.

Procedure

1. In the Hosts page, select or search for multiple hosts.
2. Using the Actions control, click Selected Hosts > [COMPONENT_NAME] > Delete.
3. Click **OK** to confirm the bulk operation.

**What to do next**
After adding components, consider adjusting the resource allocations of other components on the same hosts to accommodate the new component.

**Read audit log files**
Read the audit logs to assess details of Ambari-managed operations and configuration changes.

**About this task**
When you use Ambari to perform an operation, such as user login or logout, stopping or starting a service, and adding or removing a service, Ambari creates an entry in an audit log. By reading the audit logs, you can determine who performed the operation, when the operation occurred, and other, operation-specific information. When you change configuration for a service, Ambari creates an entry in the audit log, and creates a specific log file.

**Procedure**
- Find the Ambari audit log on your Ambari server host, at: 
  `/var/log/ambari-server/ambari-audit.log`
- Find the Ambari config changes log on your Ambari server host, at: 
  `ambari-config-changes.log`

By reading the configuration change log, you can find out even more information about each change. For example:

```
  service_name='HDFS' config_group='default' config_group_id='-1'
  version='2'
```

**Enable the Oozie UI**
You must manually create an Ooze UI when using the latest versions of Ambari and HDP.
About this task
Ext JS is GPL licensed software and is no longer included in builds of HDP 2.6 and higher. Because of this, the Oozie WAR file is not built to include the Ext JS-based user interface unless Ext JS is manually installed on the Oozie server. If you add Oozie using Ambari 2.6.1.0 to an HDP 2.6.4 or greater stack, no Oozie UI will be available by default. If you want an Oozie UI, you must manually install Ext JS on the Oozie server host, then restart Oozie. During the restart operation, Ambari re-builds the Oozie WAR file and will include the Ext JS-based Oozie UI.

Procedure
1. Log in to the Oozie Server host.
2. Download and install the Ext JS package.

CentOS RHEL Oracle Linux 6:
```
wget http://public-repo-1.hortonworks.com/HDP-UTILS-GPL-1.1.0.22/repos/centos6/extjs/extjs-2.2-1.noarch.rpm
rpm -ivh extjs-2.2-1.noarch.rpm
```

CentOS RHEL Oracle Linux 7:
```
wget http://public-repo-1.hortonworks.com/HDP-UTILS-GPL-1.1.0.22/repos/centos7/extjs/extjs-2.2-1.noarch.rpm
rpm -ivh extjs-2.2-1.noarch.rpm
```

CentOS RHEL Oracle Linux 7 (PPC):
```
wget http://public-repo-1.hortonworks.com/HDP-UTILS-GPL-1.1.0.22/repos/centos7-ppc/extjs/extjs-2.2-1.noarch.rpm
rpm -ivh extjs-2.2-1.noarch.rpm
```

SUSE11SP3:
```
wget http://public-repo-1.hortonworks.com/HDP-UTILS-GPL-1.1.0.22/repos/suse11sp3/extjs/extjs-2.2-1.noarch.rpm
rpm -ivh extjs-2.2-1.noarch.rpm
```

SUSE11SP4:
```
wget http://public-repo-1.hortonworks.com/HDP-UTILS-GPL-1.1.0.22/repos/suse11sp4/extjs/extjs-2.2-1.noarch.rpm
rpm -ivh extjs-2.2-1.noarch.rpm
```

SLES12:
```
wget http://public-repo-1.hortonworks.com/HDP-UTILS-GPL-1.1.0.22/repos/sles12/extjs/extjs-2.2-1.noarch.rpm
rpm -ivh extjs-2.2-1.noarch.rpm
```

Ubuntu12:
```
wget http://public-repo-1.hortonworks.com/HDP-UTILS-GPL-1.1.0.22/repos/sles12-extjs/extjs-2.2-1.noarch.rpm
rpm -ivh extjs-2.2-1.noarch.rpm
```
3. Remove the .war file prep file.
   `rm /usr/hdp/current/oozie-server/.prepare_war_cmd`

4. Restart Oozie Server from the Ambari UI.
   Ambari rebuilds the Oozie WAR file.

Enable the Oozie UI on CentOS RHEL Oracle Linux 7 PPC

You must manually create an Oozie UI when using the latest versions of Ambari and HDP.

About this task
Ext JS is GPL licensed software and is no longer included in builds of HDP 2.6. Because of this, the Oozie WAR file is not built to include the Ext JS-based user interface unless Ext JS is manually installed on the Oozie server. If you add Oozie using Ambari 2.6.1.0 to an HDP 2.6.4 or greater stack, no Oozie UI will be available by default. If you want an Oozie UI, you must manually install Ext JS on the Oozie server host, then restart Oozie. During the restart operation, Ambari re-builds the Oozie WAR file and will include the Ext JS-based Oozie UI.

Procedure
1. Log in to the Oozie Server host.
2. Download and install the Ext JS package.

   wget http://public-repo-1.hortonworks.com/HDP-UTILS-GPL-1.1.0.22/repos/centos7-ppc/extjs/extjs-2.2-1.noarch.rpm
   rpm -ivh extjs-2.2-1.noarch.rpm

3. Remove the following file:
   rm /usr/hdp/current/oozie-server/.prepare_war_cmd
   Ambari rebuilds the Oozie WAR file.

**Enable the Oozie UI on CentOS RHEL Oracle Linux 7**

You must manually create an Ooze UI when using the latest versions of Ambari and HDP.

**About this task**

Ext JS is GPL licensed software and is no longer included in builds of HDP 2.6. Because of this, the Oozie WAR file is not built to include the Ext JS-based user interface unless Ext JS is manually installed on the Oozie server. If you add Oozie using Ambari 2.6.1.0 to an HDP 2.6.4 or greater stack, no Oozie UI will be available by default. If you want an Oozie UI, you must manually install Ext JS on the Oozie server host, then restart Oozie. During the restart operation, Ambari re-builds the Oozie WAR file and will include the Ext JS-based Oozie UI.

**Procedure**

1. Log in to the Oozie Server host.
2. Download and install the Ext JS package.

   wget http://public-repo-1.hortonworks.com/HDP-UTILS-GPL-1.1.0.22/repos/centos7/extjs/extjs-2.2-1.noarch.rpm
   rpm -ivh extjs-2.2-1.noarch.rpm

3. Remove the following file:
   rm /usr/hdp/current/oozie-server/.prepare_war_cmd
   Ambari rebuilds the Oozie WAR file.

**Enable the Oozie UI on Suse11 sp4**

You must manually create an Ooze UI when using the latest versions of Ambari and HDP.

**About this task**

Ext JS is GPL licensed software and is no longer included in builds of HDP 2.6. Because of this, the Oozie WAR file is not built to include the Ext JS-based user interface unless Ext JS is manually installed on the Oozie server. If you add Oozie using Ambari 2.6.1.0 to an HDP 2.6.4 or greater stack, no Oozie UI will be available by default. If you want an Oozie UI, you must manually install Ext JS on the Oozie server host, then restart Oozie. During the restart operation, Ambari re-builds the Oozie WAR file and will include the Ext JS-based Oozie UI.

**Procedure**

1. Log in to the Oozie Server host.
2. Download and install the Ext JS package.

   wget http://public-repo-1.hortonworks.com/HDP-UTILS-GPL-1.1.0.22/repos/suse11sp4/extjs/extjs-2.2-1.noarch.rpm
   rpm -ivh extjs-2.2-1.noarch.rpm

3. Remove the following file:
   rm /usr/hdp/current/oozie-server/.prepare_war_cmd
4. Restart Oozie Server from **Ambari Web**.
   Ambari rebuilds the Oozie WAR file.

### Enable the Oozie UI on Suse 11 sp3
You must manually create an Ooze UI when using the latest versions of Ambari and HDP.

**About this task**
Ext JS is GPL licensed software and is no longer included in builds of HDP 2.6. Because of this, the Oozie WAR file is not built to include the Ext JS-based user interface unless Ext JS is manually installed on the Oozie server. If you add Oozie using Ambari 2.6.1.0 to an HDP 2.6.4 or greater stack, no Oozie UI will be available by default. If you want an Oozie UI, you must manually install Ext JS on the Oozie server host, then restart Oozie. During the restart operation, Ambari re-builds the Oozie WAR file and will include the Ext JS-based Oozie UI.

**Procedure**
1. Log in to the Oozie Server host.
2. Download and install the Ext JS package.
   ```
   wget http://public-repo-1.hortonworks.com/HDP-UTILS-GPL-1.1.0.22/repos/suse11sp3/extjs/extjs-2.2-1.noarch.rpm
   rpm -ivh extjs-2.2-1.noarch.rpm
   ```
3. Remove the following file:
   ```
   rm /usr/hdp/current/oozie-server/.prepare_war_cmd
   ```
4. Restart Oozie Server from **Ambari Web**.
   Ambari rebuilds the Oozie WAR file.

### Enable the Oozie UI on SLES 12
You must manually create an Ooze UI when using the latest versions of Ambari and HDP.

**About this task**
Ext JS is GPL licensed software and is no longer included in builds of HDP 2.6. Because of this, the Oozie WAR file is not built to include the Ext JS-based user interface unless Ext JS is manually installed on the Oozie server. If you add Oozie using Ambari 2.6.1.0 to an HDP 2.6.4 or greater stack, no Oozie UI will be available by default. If you want an Oozie UI, you must manually install Ext JS on the Oozie server host, then restart Oozie. During the restart operation, Ambari re-builds the Oozie WAR file and will include the Ext JS-based Oozie UI.

**Procedure**
1. Log in to the Oozie Server host.
2. Download and install the Ext JS package.
   ```
   wget http://public-repo-1.hortonworks.com/HDP-UTILS-GPL-1.1.0.22/repos/sles12/extjs/extjs-2.2-1.noarch.rpm
   rpm -ivh extjs-2.2-1.noarch.rpm
   ```
3. Remove the following file:
   ```
   rm /usr/hdp/current/oozie-server/.prepare_war_cmd
   ```
4. Restart Oozie Server from **Ambari Web**.
   Ambari rebuilds the Oozie WAR file.

### Enable the Oozie UI on Ubuntu 14
You must manually create an Ooze UI when using the latest versions of Ambari and HDP.
About this task
Ext JS is GPL licensed software and is no longer included in builds of HDP 2.6. Because of this, the Oozie WAR file is not built to include the Ext JS-based user interface unless Ext JS is manually installed on the Oozie server. If you add Oozie using Ambari 2.6.1.0 to an HDP 2.6.4 or greater stack, no Oozie UI will be available by default. If you want an Oozie UI, you must manually install Ext JS on the Oozie server host, then restart Oozie. During the restart operation, Ambari re-builds the Oozie WAR file and will include the Ext JS-based Oozie UI.

Procedure
1. Log in to the Oozie Server host.
2. Download and install the Ext JS package.
   ```
   wget http://public-repo-1.hortonworks.com/HDP-UTILS-GPL-1.1.0.22/repos/ubuntu14/pool/main/e/extjs/extjs_2.2-2_all.deb
dpkg -i extjs_2.2-2_all.deb
   ```
3. Remove the following file:
   ```
   rm /usr/hdp/current/oozie-server/.prepare_war_cmd
   ```
   Ambari rebuilds the Oozie WAR file.

Enable the Oozie UI on Ubuntu 16
You must manually create an Ooze UI when using the latest versions of Ambari and HDP.

About this task
Ext JS is GPL licensed software and is no longer included in builds of HDP 2.6. Because of this, the Oozie WAR file is not built to include the Ext JS-based user interface unless Ext JS is manually installed on the Oozie server. If you add Oozie using Ambari 2.6.1.0 to an HDP 2.6.4 or greater stack, no Oozie UI will be available by default. If you want an Oozie UI, you must manually install Ext JS on the Oozie server host, then restart Oozie. During the restart operation, Ambari re-builds the Oozie WAR file and will include the Ext JS-based Oozie UI.

Procedure
1. Log in to the Oozie Server host.
2. Download and install the Ext JS package.
   ```
   wget http://public-repo-1.hortonworks.com/HDP-UTILS-GPL-1.1.0.22/repos/ubuntu16/pool/main/e/extjs/extjs_2.2-2_all.deb
dpkg -i extjs_2.2-2_all.deb
   ```
3. Remove the following file:
   ```
   rm /usr/hdp/current/oozie-server/.prepare_war_cmd
   ```
   Ambari rebuilds the Oozie WAR file.

Enable the Oozie UI on Debian 9
You must manually create an Ooze UI when using the latest versions of Ambari and HDP.

About this task
Ext JS is GPL licensed software and is no longer included in builds of HDP 2.6. Because of this, the Oozie WAR file is not built to include the Ext JS-based user interface unless Ext JS is manually installed on the Oozie server. If you add Oozie using Ambari 2.6.1.0 to an HDP 2.6.4 or greater stack, no Oozie UI will be available by default. If you want an Oozie UI, you must manually install Ext JS on the Oozie server host, then restart Oozie. During the restart operation, Ambari re-builds the Oozie WAR file and will include the Ext JS-based Oozie UI.
Procedure
1. Log in to the Oozie Server host.
2. Download and install the Ext JS package.
   
   ```
   wget http://public-repo-1.hortonworks.com/HDP-UTILS-GPL-1.1.0.22/repos/debian9/pool/main/e/extjs/extjs_2.2-2_all.deb
   dpkg -i extjs_2.2-2_all.deb
   ```
3. Remove the following file:
   ```
   rm /usr/hdp/current/oozie-server/.prepare_war_cmd
   ```
   Ambari rebuilds the Oozie WAR file.

Refresh YARN Capacity Scheduler

Use Ambari Web to refresh the YARN Capacity Scheduler when you add or modify existing queues.

About this task
After you modify the Capacity Scheduler configuration, YARN enables you to refresh the queues without restarting your ResourceManager, if you have made no destructive changes (such as completely removing a queue) to your configuration. The Refresh operation will fail with the following message: Failed to re-init queues if you attempt to refresh queues in a case where you performed a destructive change, such as removing a queue. In cases where you have made destructive changes, you must perform a ResourceManager restart for the capacity scheduler change to take effect. To refresh the Capacity Scheduler, follow these steps:

Procedure
1. In Ambari Web, browse to Services > YARN > Summary.
2. Click Actions > Refresh YARN Capacity Scheduler.
3. Confirm that you want to perform this operation.
   The refresh operation is submitted to the YARN ResourceManager.

Restart all required services

As prompted in Ambari Web, restart all required services whenever you update configuration properties.

About this task
When you update service configuration settings in Ambari, Ambari marks the services that require a restart with a prompt similar to this:

To restart all required services:

Procedure
1. In Ambari Web, click Services....
2. Click Restart All Required.
Important: Apache Oozie requires a restart after an Atlas configuration update, but might not be marked as requiring restart in Ambari. If Oozie is not included, follow these steps to restart Oozie:

a) In Ambari Web, click Oozie in the services summary pane on the left of the display.
b) Click Service Actions > Restart All.

Managing service configuration settings

You can optimize performance of Hadoop components in your cluster by adjusting configuration settings and property values. You can also use Ambari Web to set up and manage groups and versions of configuration settings in the following ways:

Related Information
Review and confirm configuration changes
Restart all required services

Change configuration settings

Use Services > [SERVICE_NAME] > Configs to optimize service performance for the service.

About this task
The Configs page includes several tabs you can use to manage configuration versions, groups, settings, properties and values. You can adjust settings called Smart Configs that control at a macro-level, memory allocation for each service. Adjusting Smart Configs requires related configuration settings to change throughout your cluster. Ambari prompts you to review and confirm all recommended changes and restart affected services.

Procedure

1. In Ambari Web, click a service name in the service summary list on the left.
2. From the the service Summary page, click the Configs tab, then use one of the following tabs to manage configuration settings.
   - Use the Configs tab to manage configuration versions and groups.
   - Use the Settings tab to manage Smart Configs by adjusting the green, slider buttons.
   - Use the Advanced tab to edit specific configuration properties and values.
3. Click **Save**.
4. Enter a description for this configuration version that includes your current changes.
5. Review and confirm each recommended change.

**What to do next**
Restart all affected services.

---

**Adjust Smart Config settings**

Use **Configs > Settings** to manage "Smart Configs" by adjusting the green, slider buttons.

**Procedure**
1. On the **Settings** tab, click and drag a green-colored slider button to the desired value.
2. Edit values for any properties that display the Override option.
   
   Edited values, also called *stale configs*, show an Undo option.
3. Click **Save**.
4. Enter a description for this configuration version that includes your current changes.
5. Review and confirm each recommended change.

**What to do next**
Restart all affected services.
Edit specific configuration properties

Use **Configs > Advanced** for each service to access groups of individual properties that affect performance of that service.

**Procedure**

1. For a service, click **Configs > Advanced**, and expand a category.
2. Edit the value for any property.
   
   Edited values, also called *stale configs*, show an Undo option.
3. Click **Save**.
4. Enter a description for this configuration version that includes your current changes.
5. Review and confirm each recommended change.

**What to do next**

Restart all affected services.

Review and confirm configuration changes

When you change a configuration setting, use **Dependent Configurations** to confirm other, related configuration changes.

**Before you begin**

Change a configuration property. As prompted, click Show Details.

![There are 11 configuration changes in 3 services Show Details](image)

**About this task**

When you change a configuration property value, the Ambari Stack Advisor captures and recommends changes to all related configuration properties affected by your original change. Changing a single property, a Smart Configuration, and other actions; such as adding or deleting a service, host, or ZooKeeper Server, moving a master, or enabling high availability for a component, all require that you review and confirm related configuration changes. For example, if you increase the Minimum Container Size (Memory) setting for YARN, **Dependent Configurations** lists all recommended changes that you must review and (optionally) accept.
Types of changes are highlighted in the following colors:

<table>
<thead>
<tr>
<th>Type</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value Changes</td>
<td>Yellow</td>
</tr>
<tr>
<td>Added Properties</td>
<td>Green</td>
</tr>
<tr>
<td>Deleted properties</td>
<td>Red</td>
</tr>
</tbody>
</table>

To review and confirm changes to configuration properties:

**Procedure**

1. In **Dependent Configurations**, for each listed property review the summary information.
2. If the change is acceptable, proceed to review the next property in the list.
3. If the change is not acceptable, click the check mark in the blue box to the right of the listed property change. Clicking the check mark clears the box. Changes for which you clear the box are not confirmed and will not occur.
4. After reviewing all listed changes, click **OK** to confirm that all marked changes occur.

**What to do next**

Restart any components marked for restart to utilize the changes you confirmed.
**Restart components**

Use Restart options to start service components using new configuration properties.

**Before you begin**
Edit and save configurations for one or more services.

**About this task**
After editing and saving configuration changes, a Restart indicator appears next to components that require restarting. You must restart components affected by a configuration change so that the component uses the updated configuration values.

**Procedure**

1. Click the indicated Components or Hosts links to view details about the requested restart.
2. Click Restart and then click the appropriate action.
   For example, options to restart YARN components include the following:

   ![Restart components](image)

**Download client configuration files for a service**

Use the Services > Actions > Download Client Configs option to download client configurations for a single service.

**About this task**
Client configuration files include .xml files, env-sh scripts, and log4j properties used to configure Hadoop services. For services that include client components (most services except SmartSense and Ambari Metrics Service), you can download the client configuration files associated with that service. You can also download the client configuration files for your entire cluster as a single archive. To download client configuration files for a single service:

**Procedure**

1. In Ambari Web > Services, click the service for which you want to download configurations.
2. Click Actions.
3. Click Download Client Configs.
   Your browser downloads a tarball archive containing only the client configuration files for that service to your default, local downloads directory.
4. If prompted to save or open the client configs bundle,
Click **Save File**, then click **OK**.

**Download all client configuration files for a cluster**

Use the **Ambari Web > Services > ... > Download All Client Configs** option to download all client configurations for your cluster.

**About this task**

Client configuration files include; .xml files, env-sh scripts, and log4j properties used to configure Hadoop services. For services that include client components (most services except SmartSense and Ambari Metrics Service), you can download the client configuration files associated with that service. You can also download the client configuration files for your entire cluster as a single archive. To download client configuration files for your entire cluster:

**Procedure**

1. In **Ambari Web > Services > ...**, click **Download All Client Configs**.
Your browser downloads a tarball archive containing all client configuration files for your cluster to your default, local downloads directory.

2. If prompted to save or open the cluster configs bundle, click **Save File**, then click **OK**.

## Managing service configuration versions

Use **Ambari Web > Services > [SERVICE_NAME] > Configs**, to manage configuration versions.

Ambari enables you to manage configurations associated with a service. You can make changes to configurations, see a history of changes, compare and revert changes, and push configuration changes to the cluster hosts.

**Related Information**
- Review and confirm configuration changes
- Restart all required services

## Understanding service configuration versions

Ambari creates and stores a unique configuration version when you change any property value in a config type.

It is important to understand how service configurations are organized and stored in Ambari. Properties are grouped into configuration types. A set of config types composes the set of configurations for a service.

For example, the Hadoop Distributed File System (HDFS) service includes the hdfs-site, core-site, hdfs-log4j, hadoop-env, and hadoop-policy config types. Using **Ambari Web**, browse to **Services > HDFS > Configs**, to edit the configuration properties for these config types.

Ambari performs configuration versioning at the service level. Therefore, when you modify a configuration property in a service, Ambari creates a *service config version*. The following figure shows version 1 (V1) and version 2 (V2) of a service configuration with a change to a property in Config Type A. When you change a property value in Config Type A in V1, Ambari creates V2.
Service configuration terminology

Terms and concepts that describe configuration versioning.

configuration property
Configuration property managed by Ambari, such as NameNode heap size or replication factor

configuration type (config type)
Group of configuration properties: for example, hdfs-site

service configurations
Set of configuration types for a particular service: for example, hdfs-site and core-site as part of the HDFS service configuration

change notes
Optional notes to save with a service configuration change

service config version (SCV)
A particular version of a configuration for a specific service

host config group (HCG)
A set of configuration properties to apply to a specific set of hosts

Save a service configuration change

Describe any changes you make to service configs as you save the changes, to store descriptive information in the service configuration version history.

About this task
Any change that you make to a configuration property creates a new configuration version.

Procedure
1. For a service, in Configs, change the value of a configuration property.
2. Choose Save.
   Save Configuration prompts you to describe and save the change.
3. Optionally, enter text that describes the change.

4. Then, click one of the following options:
   - **Option**  **To:**
     - Cancel  continue editing
     - Discard  leave the control without making any changes
     - Save  confirm your change

**Results**
After saving your change(s), a new service configuration version appears in **Configs**.

**What to do next**
Restart affected components, as prompted.

**View service configuration history**
Ambari saves previous versions of configurations and provides access to them for comparison and revert.

**About this task**
*Ambari Web* provides two ways to view your configuration change history. The **Dashboard** page includes the **Config History** tab. Each service page provides a **Configs** tab, specific to that service. Using **Configs** enables you to quickly access the most recent changes to a service configuration.

**Procedure**
- From the **Dashboard**, click **Config History**
Config History shows all versions across all services, with each version number and the date and time the version was created. You can also see which user authored the change, and any notes about the change. Use controls on Config History to filter, sort, and search across versions.

- For a selected service, click Configs, then expand Version:

Configs shows you details of the most recent configuration version by default. Use the version scrollbar to see previous config versions.

- From the Version list, click any version in to view detailed information.
Managing service configuration versions

• Click one of the following options:
  Select Option
  To:
  Click a Version
  display detailed information for the selected version in Configs
  display and compare two versions
  apply configs from the currently selected version to the cluster

What to do next
If you choose Make Current, restart any affected services after changing versions.

Compare service configuration versions

Use Compare Versions to display differences between the current configuration version and an older one.

About this task
Services > [SERVICE_NAME] > Configs displays specific property values from the current service configuration. Configs also supports comparing details from the current version with a previous one. All versions of a service configuration history display as version-numbered rows in the Versions menu. When browsing multiple config versions displayed in the Versions menu, hover your cursor over a version block to display options to view, compare, or make current.

Procedure
1. While details of the current version display on Configs, find a version you want to compare with the current one, using the Versions menu.
   For example, if you want to compare the current version 7 to 6, find version 6 in the Versions menu.
2. In the Versions menu, hover your cursor over the V2 block to display options, then click

   Configs displays a comparison of V6 to V7. Specific differences appear in the main window.
In this example, V6 differs from V7 by nine properties.

3. To filter the comparison, expand the Filter menu, and select an option.
   For example, to display only properties with changes between two versions, in Filters, click Changed Properties.
4. To close **Comparing Changes**, click the X.

![Comparing Changes](image)

**What to do next**
If you want to revert the service to use the older version settings, click **Make [PREVIOUS_VERSION] Current**.

### Make a previous service version current

Use **Make Current** to revert a cluster to operate using a previous configuration version.

**About this task**

[SERVICE_NAME] > **Configs** displays specific property values from the current service configuration. **Configs** also supports comparing details from the current version with a previous one. All versions in a service configuration history display as version-numbered rows across the **Versions** menu. When browsing multiple config versions displayed in the **Versions** menu, click a version row to display options to view, compare, or make current. The Make Current option reverts the cluster to operate using a previous config version. **Make Current** creates (clones) a new, current service configuration version using the configuration properties from the older version you are comparing.

**Procedure**

- From the Version list, click any version, then click **Make Current**.

![Make Current](image)

**What to do next**
After initiating the Make Current operation, **Make Current Confirmation** prompts you to enter text that describes the new version. To confirm and save the new version after typing descriptive text, click **Make Current**.

### Managing HDFS

This section contains information specific to rebalancing and tuning garbage collection in Hadoop Distributed File System (HDFS).

### Rebalance HDFS blocks
About this task
HDFS provides a balancer utility to help balance the blocks across DataNodes in the cluster. To initiate a balancing process, follow these steps:

Procedure
1. In Ambari Web, browse to Services > HDFS > Summary.
2. Click Service Actions > Rebalance HDFS.
3. Enter the Balance Threshold value as a percentage of disk capacity.
4. Click Start.

What to do next
You can monitor or cancel a rebalance process by opening Background Operations.

Tune HDFS garbage collection

About this task
The Concurrent Mark Sweep (CMS) garbage collection (GC) process includes a set of heuristic rules used to trigger garbage collection. This makes garbage collection less predictable and tends to delay collection until capacity is reached, creating a Full GC error (which might pause all processes). Ambari sets default parameter values for many properties during cluster deployment. Within the export HADOOP_NameNode_Opts= clause of the hadoop-env template, two parameters that affect the CMS GC process have the following default settings:

- -XX:+UseCMSInitiatingOccupancyOnly
  prevents the use of GC heuristics.
- -XX:CMSInitiatingOccupancyFraction=<percent>
  tells the Java VM when the CMS collector should be triggered.

If this percent is set too low, the CMS collector runs too often; if it is set too high, the CMS collector is triggered too late, and concurrent mode failure might occur. The default setting for -XX:CMSInitiatingOccupancyFraction is 70, which means that the application should utilize less than 70% of capacity. To tune garbage collection by modifying the NameNode CMS GC parameters, follow these steps:

Procedure
1. In Ambari Web, browse to Services > HDFS.
2. Open the Configs tab and browse to Advanced > Advanced hadoop-env.
3. Edit the hadoop-env template.
4. Save your configurations and restart, as prompted.

Customize the HDFS home directory

About this task
By default, the HDFS home directory is set to /user/[USER_NAME]. You can use the dfs.user.home.base.dir property to customize the HDFS home directory.

Procedure
1. In Ambari Web, browse to Services > HDFS > Configs > Advanced.
2. Click Custom hdfs-site, then click Add Property.
3. On the Add Property pop-up, add the following property:
   dfs.user.home.base.dir=[HOME_DIRECTORY]
   Where [HOME_DIRECTORY] is the path to the new home directory.
4. Click Add, then save the new configuration and restart, as prompted.

**Configure HDFS Federation**

An HDFS federation allows you to scale a cluster horizontally by configuring multiple namespaces and NameNodes. The DataNodes in the cluster are available as a common block of storage for every NameNode in the federation.

**Before you begin**

- Ensure that you have planned for a cluster maintenance window before configuring the HDFS federation because all the cluster services are restarted during the process of configuration.
- Verify that you have configured HA for all the NameNodes that you want to include in the federation.

**About this task**

You must associate every NameNode you want to include in a federation with a namespace. You can configure a maximum of four namespaces in a federated environment.

**Procedure**

1. In Ambari Web, select Services > HDFS > Summary.
2. Click Actions > Add New HDFS Namespace.
   The Add New HDFS Namespace wizard launches. The wizard describes the set of automated and manual steps you must perform to add the new namespace.
3. On the Get Started page, type in a NameService ID and click Next.

   ![Add New HDFS Namespace Wizard](image)

4. On the Select Hosts page, select a host for the additional NameNodes and JournalNodes, and click Next.
5. On the **Review** page, confirm your host selections and click **Next**.

6. On the Configure Components page, monitor the progress bars as the wizard completes adding the new namespace, then click **Complete**.
After the Ambari Web UI reloads, you may see some alert notifications. Wait a few minutes until all the services restart.

You can navigate to Services > HDFS > Summary to view details of the newly added namespace.

Note: If the process of adding the new namespace does not complete, then you must update the configuration properties of the NameNodes so that the cluster reverts to its earlier functional state.

**Configure ViewFs**

Configure ViewFs to create common views for file and directory paths corresponding to locations on different NameNodes of a federated HDFS cluster by specifying mount table entries. You can use Ambari Web UI to specify ViewFs as the default file system, and specify mount table entries for the file and directory paths.

**Before you begin**

Verify that you have configured HDFS federation for the cluster.

**About this task**

- You can either specify the ViewFs mount table entries as key-value pairs in core-site.xml, or add a separate configuration file containing the mount paths.
- ViewFs is not supported on Hive clusters.

**Procedure**

1. In Ambari Web, select Services > HDFS > Configs, and navigate to Advanced core-site.
2. Update the value of the fs.defaultFS property to viewfs://<clustername>.
   For example, if your cluster is named Clusterx, set the value to viewfs://ClusterX.
   This sets ViewFs as the default file system.
   
   **Note:** Updating fs.defaultFS to ViewFs also updates the default root to point to the ViewFs root. Therefore, you must remap any directory paths mounted earlier to point to the namespaces that contain those directories.

3. Use either of the following options to specify the mount table entries for the file and directory paths.
   Key-value pairs in core-site.xml:
   
   a. Navigate to Custom core-site.
b. Click **Add property**.
c. Add a path name property as the key, its corresponding mount point as the value, and click **Add**.

Add similar key-value pairs for all the file and directory paths you want to mount.

For example, if ClusterX has a directory named /tmp in the namespace ns1 and another directory named /foo in the namespace ns2, you can define the following key-value pairs:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>fs.viewfs.mounttable.ClusterX.link./tmp</td>
<td>hdfs://ns1/tmp</td>
</tr>
<tr>
<td>fs.viewfs.mounttable.ClusterX.link./foo</td>
<td>hdfs://ns2/foo</td>
</tr>
</tbody>
</table>

d. Click **Save**.
e. Add a description of the configuration changes, and click **Save**.
f. Restart HDFS and other services as applicable.

Mount table configuration file:

a. Navigate to **Advanced viewfs-mount-table**.
b. Enter the mount table entries to the various file and directory paths as properties in the text box.

For the previous example, you can specify mount table entries for the directories /foo and /tmp as follows:

```xml
<configuration>
  <property>
    <name>fs.viewfs.mounttable.ClusterX.link./tmp</name>
    <value>hdfs://ns1/tmp</value>
  </property>
  <property>
    <name>fs.viewfs.mounttable.ClusterX.link./foo</name>
    <value>hdfs://ns2/foo</value>
  </property>
</configuration>
```
c. Click **Save**.
d. Add a description of the configuration changes, and click **Save**.
e. Restart HDFS and other services as applicable.

---

**Start Kerberos wizard from Ambari Web**

As a cluster administrator, use **Ambari Web > Cluster Admin > Kerberos** to enable and manage Kerberos security in your cluster.

**Before you begin**
Before enabling Kerberos in your cluster, you must prepare the cluster, as described in Configuring Kerberos.

**Procedure**
- In **Ambari Web > Cluster Admin > Kerberos**, click **Enable Kerberos**.
  The **Enable Kerberos** wizard launches.
- Complete all steps in the **Enable Kerberos** wizard.

**What to do next**
After Kerberos is enabled, regenerate key tabs. You can disable Kerberos, using **Ambari Web > Cluster Admin > Kerberos**.
Regenerate Kerberos keytabs from Ambari Web

As a cluster administrator, use Ambari Web > Cluster Admin > Kerberos to regenerate the key tabs required to maintain Kerberos security in your cluster.

Before you begin
Before regenerating key tabs in your cluster:

- your cluster must be Kerberos-enabled
- you must have KDC Admin credentials

Procedure

1. In Ambari Web > Cluster Admin > Kerberos, click Regenerate Keytabs.
2. Confirm your selection to proceed.

Results

Ambari connects to the Kerberos Key Distribution Center (KDC) and regenerates the key tabs for the service and Ambari principals in the cluster. Optionally, you can regenerate key tabs for only those hosts that are missing key tabs: for example, hosts that were not online or available from Ambari when enabling Kerberos.

What to do next

Restart all services.

Disable Kerberos from Ambari Web

As a cluster administrator, use Ambari Web > Cluster Admin > Kerberos to disable Kerberos security in your cluster.

Before you begin

your cluster must be Kerberos-enabled.

Procedure

1. In Ambari Web > Cluster Admin > Kerberos, click Disable Kerberos.
2. Confirm your selection.
3. Confirm your selection to proceed.

Results

Cluster services are stopped and the Ambari Kerberos security settings are reset.

What to do next

To re-enable Kerberos, click Enable Kerberos and complete all steps in the wizard.

Configuring log settings

Configure the log4j property group to control logging activities for each service running in your Hadoop cluster.
About this task
Ambari uses sets of properties called log4j properties to control logging activities for each service running in your Hadoop cluster. Initially, default values for each property reside in Advanced[SERVICE_NAME]-log4j in a [SERVICE_NAME]-log4j template file, and in custom[SERVICE_NAME]-log4j in a custom[SERVICE_NAME]-log4j properties file. Log4j properties and values that limit the size and number of backup log files for each service appear in Advanced[SERVICE_NAME]-log4j with the log4j template file. Together, the log4j template, custom properties file and backup properties are called the log4j property group. To configure log settings for a service, first access the log4j property group for the service.

Procedure
• In Ambari Web, browse to Services > [SERVICE_NAME] > Configs > Advanced[SERVICE_NAME]-log4j. For example, Advanced yarn-log4j in the YARN log4k property group for the YARN service looks like:

![Advanced yarn-hbase-log4j](image)

• In Ambari Web, browse to Services > [SERVICE_NAME] > Configs > custom[SERVICE_NAME]-log4j and browse to custom [SERVICE_NAME]log4j properties.

What to do next
Use the template and custom properties files for a selected service to edit and customize specific properties and values that control log activities for that service, as necessary.
Related Information
- Review and confirm configuration changes
- Restart all required services

Limit the size and number of backup log files for a service

Edit properties in the `log4j property group` for a service to limit backup log files.

About this task
Log4j properties and values that limit the size and number of backup log files for each service appear above the log4j template file in the log4j property group. To limit backup log files for a service, first access the Log4j property group for the service.

Procedure
1. In Ambari Web, browse to Services > [SERVICE_NAME] > Configs > Advanced[SERVICE_NAME]-log4j.
2. Edit the values for the [SERVICE_NAME] backup file size and [SERVICE_NAME] # of backup files properties.
3. In Configs, click Save.

Customize log4j settings for a service

Edit the `log4j property template` in which default log4j properties reside, and use those customizations to overwrite the custom log4j properties for each service to control logging activities for that service.

About this task
Ambari uses sets of properties called `Log4j properties` to control logging activities for each service running in your Hadoop cluster. Initially, default values for each property reside in the log4j property group. To customize log settings, first access the Log4j property group for the service. Then, edit the log4j template, and save a service-specific version of property values appropriate to control log activities for the service, in custom [SERVICE_NAME]log4j properties.

Procedure
1. In Ambari Web, browse to Services > [SERVICE_NAME] > Configs > Advanced[SERVICE_NAME]-log4j.
2. Edit values of any properties in the [SERVICE_NAME] log4j template.
3. Copy the content of the log4j template file.
5. Paste the copied content into custom [SERVICE_NAME]log4j properties, overwriting, the default content.
6. In Configs, click Save.
7. Review and confirm any recommended, related configuration changes, as prompted.
8. Restart affected components and services, as prompted.
   - Restarting components in the service pushes the configuration properties displayed in Custom log4j.properties to each host running components for that service.

What to do next
If you have customized logging properties that define how activities for each service are logged, you see refresh indicators next to each service name. Ensure that logging properties displayed in Custom log4j.properties include any customization.
Managing host configuration groups

Manage configurations across multiple hosts by creating host configuration groups.

Ambari initially assigns all hosts in your cluster to one default configuration group for each service you install. For example, after deploying a three-node cluster with default configuration settings, each host belongs to one configuration group that has default configuration settings for the HDFS service.

Create a new host configuration group

To create new groups, reassign hosts, and override default settings for host components, you can use the Manage Configuration Groups control:

About this task
To create a new configuration group:

Procedure
1. Click a service name, then click Configs.
2. In Configs, click Manage Config Groups.

A default config group appears as follows:

4. Name and describe the group; then choose OK.

**Add a host to a configuration group**

Use Manage Config Groups to add hosts to a configuration group.

**Procedure**

1. In Manage Config Groups, click a configuration group name.
2. Click Add Hosts to selected Configuration Group.
3. Using **Select Configuration Group Hosts**, click **Components**, then click a component name from the list. Choosing a component filters the list of hosts to only those on which that component exists for the selected service. To further filter the list of available host names, use the Filter drop-down list. The host list is filtered by IP address, by default.

![Select Configuration Group Hosts](image)

4. After filtering the list of hosts, click the check box next to each host that you want to include in the configuration group.

5. Choose **OK**.

6. In **Manage Configuration Groups**, choose **Save**.

**Edit settings for a configuration group**

Use **Configs** to edit settings for a configuration group.

**Procedure**

1. In **Configs**, click a group name.
2. Click a Config Group; then expand components to expose settings that allow Override.
3. Provide a non-default value; then click **Override** or **Save**.

Configuration groups enforce configuration properties that allow override, based on installed components for the selected service and group.
4. In [SERVICE] Configuration Group complete one of the two options.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Click an existing</td>
<td>to which the property value override provided in Step 3 applies</td>
</tr>
<tr>
<td>configuration group</td>
<td></td>
</tr>
<tr>
<td>Create a new</td>
<td>which includes default properties, plus the property override provided in</td>
</tr>
<tr>
<td>configuration group</td>
<td>Step 3</td>
</tr>
</tbody>
</table>

5. In [SERVICE] Configuration Group click OK.

6. In Configs, choose Save.

**Host configuration groups example workflow**

Shows multiple host config groups and creates service configuration versions in each config group.

Service configuration versions are scoped to a host config group. For example, changes made in the default group can be compared and reverted in that config group. The same applies to custom config groups.

The following example workflow shows multiple host config groups and creates service configuration versions in each config group:
2. Modify the heapsize property to 512 and save the change. Ambari increments the service configuration version to V2. Users can compare and revert between these versions.

Default
Host Config Group
(host1, host2, host3)

V1
heapsize=256

V2
heapsize=512

Version compare + revert available

3. Create a new custom host config group and place host3 into the group. For this particular host group, override the heapsize to be 768, creating version V3. The host group (i.e. host3) inherits all properties from default config group and heapsize override from custom group.

Default
Host Config Group
(host1, host2)

V1
heapsize=256

V2
heapsize=512

Custom
Host Config Group
(host3)

V3
heapsize=768

4. Modify the custom host config group by adjusting the heapsize to be 1024, creating version V4. Users can compare and revert between these versions.

Default
Host Config Group
(host1, host2)

V1
heapsize=256

V2
heapsize=512

Custom
Host Config Group
(host3)

V3
heapsize=768

V4
heapsize=1024

Version compare + revert available
Managing Alerts and Notifications

Use customized notifications to communicate important Ambari alerts to the appropriate recipient.

Ambari uses a predefined set of seven types of alerts; web, port, metric, aggregate, script, server, and recovery, for each cluster component and host. You can use these alerts to monitor cluster health and to alert other users to help you identify and troubleshoot problems. You can modify alert names, descriptions, and check intervals, and you can disable and re-enable alerts. You can also create groups of alerts and setup notification targets for each group so that you can notify different parties interested in certain sets of alerts in different ways.

Understanding alerts

Use Ambari Web > Alerts to scan, enable or disable, or open an alert definition for customization.

Ambari predefines a set of alerts that monitor the cluster components and hosts. Each alert is defined by an alert definition, which specifies the alert type, check interval, and thresholds. When a cluster is created or modified, Ambari reads the alert definitions and creates alert instances for the specific items to monitor in the cluster. For example, if your cluster includes Hadoop Distributed File System (HDFS), there is an alert definition to monitor "DataNode Process". An instance of that alert definition is created for each DataNode in the cluster.

Using Ambari Web > Alerts, you can browse the list of alerts defined for your cluster. You can search and filter alert definitions by current status, alert definition name, last status change, alert state, and by the service with which the alert definition is associated. You can click alert definition name to view details about that alert, to modify the alert properties (such as check interval and thresholds), and to see the list of alert instances associated with that alert definition.

Each alert instance reports an alert status, defined by severity. The most common severity levels are OK, WARNING, and CRITICAL, but there are also severities for UNKNOWN and NONE. Alert notifications are sent when alert status changes (for example, status changes from OK to CRITICAL).

Alert types

Describes seven types of Ambari alerts: web, port, metric, aggregate, script, server, and recovery.

Alert thresholds and the threshold units depend on the type of the alert. The following list describes alert types, possible status for each type, and to what units thresholds can be configured if the thresholds are configurable.
WEB Alert Type

WEB alerts watch a web URL on a given component; the alert status is determined based on the HTTP response code. Therefore, you cannot change which HTTP response codes determine the thresholds for WEB alerts. You can customize the response text for each threshold and the overall web connection timeout. A connection timeout is considered a CRITICAL alert. Threshold units are based on seconds. The status values and response codes for WEB alerts are:

- OK status if the web URL responds with a code under 400.
- WARNING status if the web URL responds with code 400 and above.
- CRITICAL status if Ambari cannot connect to the web URL.

PORT Alert Type

PORT alerts check the response time to connect to a given a port; the threshold units are based on seconds.

METRIC Alert Type

METRIC alerts check the value of a single or multiple metrics, if a calculation is performed. The metric is accessed from a URL endpoint available on a given component. A connection timeout is considered a CRITICAL alert. The thresholds are adjustable and the units for each threshold depend on the metric. For example, in the case of CPU utilization alerts, the unit is percentage; in the case of RPC latency alerts, the unit is milliseconds.

AGGREGATE Alert Type

AGGREGATE alerts aggregate the alert status as a percentage of the alert instances affected. For example, the Percent DataNode Process alert aggregates the DataNode Process alert.

SCRIPT Alert Type

SCRIPT alerts execute a script that determines status such as OK, WARNING, or CRITICAL. You can customize the response text and values for the properties and thresholds for the SCRIPT alert.

SERVER Alert Type

SERVER alerts execute a server-side runnable class that determines the alert status such as OK, WARNING, or CRITICAL.

RECOVERY Alert Type

RECOVERY alerts are handled by the Ambari Agents that are monitoring for process restarts. Alert status OK, WARNING, and CRITICAL are based on the number of times a process is restarted automatically. This is useful to know when processes are terminating and Ambari is automatically restarting.

Find alerts for a service

A service with current alerts displays red, numbered indicators next to the service name, and on the service's Summary page.
**About this task**

The **Summary** tab provides access to all alerts and their status for the selected service. Critical alerts cause a red indicator to display on **Summary**.

![Summary tab](image)

**Procedure**

1. For a service, in **Summary**, click the bell icon next to the alert indicator. For example, in **Services > HDFS** click the bell icon. Alerts for HDFS opens.

![Alerts for HDFS](image)

2. Click the text title of each alert message in the list to see the alert definition.

**What to do next**

Modify the alert definition, if necessary.

**Modify an alert**

Edit the configuration properties in each alert definition to modify alert behavior.
About this task
General properties for an alert include name, description, check interval, and thresholds. The check interval defines the frequency with which Ambari checks alert status. For example, a 1 minute value means that Ambari checks the alert status every minute. The configuration options for thresholds depend on the alert type. To modify alert behavior, edit the general properties in the alert definition.

Procedure
1. In Ambari Web, click Alerts.
2. In Alerts, find an alert definition name, and click it. The alert definition page for that alert opens.
3. Click the pencil icon next to the alert name to modify the name of the alert.
4. Click Edit to modify configuration properties such as description, check interval, and thresholds, as necessary. Configuration properties differ for each alert.
5. Click Save.
   Changes take effect on all alert instances at the next check interval.

Modify the global alert check count
Use Alerts > Actions > Manage Alert Settings to set a global control for how many times Ambari checks an alert before dispatching a notification.

About this task
If the alert state changes during a check, Ambari attempts to check the condition a number of times. That number is called the check count. You can set the value of check count. Alert check counts are not applicable to AGGREGATE alert types. A state change for an AGGREGATE alert results in a notification dispatch. If your environment experiences transient issues resulting in false alerts, you can increase the check count. In this case, the alert state change is still recorded, but as a soft state change. If the alert condition is still triggered after the specified number of checks, the state change is considered a hard state change, and notifications dispatch. Generally, you should set the check count value globally for all alerts. You can override the global check count value for any alerts experiencing transient issues.

Procedure
1. In Ambari Web, browse to Alerts > Actions > Manage Alert Settings.
2. In Check Count, edit the value, as necessary.
3. Click Save.

What to do next
Wait several seconds for changes made to the global alert check count to appear for individual alerts in Ambari Web.

Override the global alert check count
Edit the check count in each alert definition to override the global control for how many times Ambari checks an alert before dispatching a notification with a unique value specified per alert.

About this task
If the alert state changes during a check, Ambari attempts to check the condition a number of times. That number is called the check count. You can set the value of check count both globally and for each alert. Alert check counts are not applicable to AGGREGATE alert types. A state change for an AGGREGATE alert results in a notification dispatch. If your environment experiences transient issues resulting in false alerts, you can increase the check count. In this case, the alert state change is still recorded, but as a soft state change. If the alert condition is still triggered after the
specified number of checks, the state change is considered a hard state change, and notifications dispatch. Generally, you should set the check count value globally for all alerts. You can override the global check count value for any alerts experiencing transient issues by modifying the check count value specifically for each alert.

Procedure
1. In Ambari Web, browse to Alerts.
2. Select the alert for which you want to set a check count value different from the global check count value.
3. In Alert Info, click the Edit icon next to the Check Count property.
4. In Edit Alert Count, update the Check Count value.
5. Click Save.

Enabling an alert
Edit the Alert Info > State for each alert definition to control whether the alert dispatches notifications.

About this task
You can optionally disable an alert. When an alert is disabled, no alert instances are in effect and Ambari will no longer perform the checks for the alert. Therefore, no alert status changes will be recorded and no notifications (i.e. no emails or SNMP traps) will be dispatched.

Procedure
1. In Ambari Web, browse to Alerts.
2. Browse the list of alert names to find a specific alert name.
3. Optionally, you can click an alert name to view the definition details.
4. In State, click Disabled text to enable the alert.
5. Confirm the enable action, as prompted.

Disabling an alert
Edit the Alert Info > State for each alert definition to control whether the alert dispatches notifications.

About this task
You can optionally disable alerts. When an alert is disabled, no alert instances are in effect and Ambari will no longer perform the checks for the alert. Therefore, no alert status changes will be recorded and no notifications (i.e. no emails or SNMP traps) will be dispatched.

Procedure
1. In Ambari Web, browse to Alerts.
2. Browse the list of alert names to find a specific alert name.
3. Optionally, you can click an alert name to view the definition details.
4. In State, click Enabled text to disable the alert.
5. Confirm the disable action, as prompted.

View the alert status log
Using a command line editor, view the alert status log on the Ambari server host.
About this task
Whether you have configured Ambari to send alert notifications or not, it writes alert status changes to a log on the Ambari Server host. You can view the alert status log using a command line editor.

Procedure

1. On the Ambari server host, browse to the log directory.
   
   ```bash
   cd /var/log/ambari-server/
   ```

2. View the ambari-alerts.log file.
   
   Log entries include the time of the status change, the alert status, the alert definition name, and the response text.

   ```plaintext
   2015-08-10 22:47:37,120 [OK] [HARD] [STORM] (Storm Server Process) TCP OK - 0.000s response on port 8744
   2015-08-11 11:06:18,479 [CRITICAL] [HARD] [AMBARI]
   [ambari_server_agent_heartbeat] (Ambari Agent Heartbeat)
   c6401.ambari.apache.org is not sending heartbeats
   2015-08-11 11:08:18,481 [OK] [HARD] [AMBARI]
   [ambari_server_agent_heartbeat] (Ambari Agent Heartbeat)
   c6401.ambari.apache.org is healthy
   ```

Understanding notifications

Use alerts, notifications, and groups to inform unique sets of users about only those issues important to them, using an appropriate channel.

Using alert groups and notifications enables you to create groups of alerts and set up notification targets for each group in such a way that you can notify different parties interested in certain sets of alerts by using different methods. For example, you might want your Hadoop Operations team to receive all alerts by email, regardless of status, while at the same time you want your System Administration team to receive only RPC- and CPU-related alerts that are in Critical state, and only by simple network management protocol (SNMP).

To achieve these different results, you can have one alert notification that manages email for all alert groups for all severity levels, and a different alert notification group that manages SNMP on critical-severity alerts for an alert group that contains the RPC and CPU alerts.

Create an alert notification

Use Alerts > Actions > Manage Notifications to create or edit alert notifications.

Procedure

1. In Ambari Web, browse to Alerts > Actions > Manage Notifications.
2. In Manage Alert Notifications, click +.
3. In Create Alert Notification, complete the fields.
   a) In Name, enter a name for the notification.
   b) In Groups, click All or Custom to assign the notification to every or set of groups that you specify.
   c) In Description, type a phrase that describes the notification.
   d) In Method, click EMAIL, SNMP (for MIB-based) or Custom SNMP as the method by which Ambari server handles delivery of this notification.
4. Complete the fields for the notification method you selected.
   
   For email notification - provide information about your SMTP infrastructure, such as SMTP server, port, to and from addresses, and whether authentication is required to relay messages through the server. You can add custom properties to the SMTP configuration based on Javamail SMTP options.

103
Email To
A comma-separated list of one or more email addresses to which to send the alert email

SMTP Server
The FQDN or IP address of the SMTP server to use to relay the alert email

SMTP Port
The SMTP port on the SMTP server

Email From
A single email address to be the originator of the alert email

Use Authentication
Determine whether your SMTP server requires authentication before it can relay messages. Be sure to also provide the username and password credentials

For MIB-based SNMP notification - provide the version, community, host, and port to which the SNMP trap should be sent.

Version
SNMPv1 or SNMPv2c, depending on the network environment

Hosts
A comma-separated list of one or more host FQDNs to which to send the trap

Port
The port on which a process is listening for SNMP traps

For SNMP notifications - Ambari uses a MIB, a text file manifest of alert definitions, to transfer alert information from cluster operations to the alerting infrastructure. A MIB summarizes how object IDs map to objects or attributes. For example, MIB file content looks like this:

```
apacheAmbiariAlertEntry OBJECT-TYPE
SYNTAX AlertEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"Each Alert Event"
INDEX {alertDefinitionId}
::= {apacheAmbiariAlertTable 1}

alertDefinitionId OBJECT-TYPE
SYNTAX Integer32 (-2147483648..2147483647)
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"ID of the Alert"
::= {apacheAmbiariAlertEntry 1}

alertDefinitionName OBJECT-TYPE
SYNTAX OCTET STRING
MAX-ACCESS accessible-for-notify
STATUS current
DESCRIPTION
"Alert Definition Name"
::= {apacheAmbiariAlertEntry 2}
```

You can find the MIB file for your cluster on the Ambari Server host, at /var/lib/ambari-server/resources/APACHE-AMBARI-MIB.txt.
For **Custom SNMP notification** - provide the version, community, host, and port to which the SNMP trap should be sent. Also, the OID parameter must be configured properly for SNMP trap context. If no custom, enterprise-specific OID is used, you should use the following:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Version</strong></td>
<td>SNMPv1 or SNMPv2c, depending on the network environment</td>
</tr>
<tr>
<td><strong>OID</strong></td>
<td>1.3.6.1.4.1.18060.16.1.1</td>
</tr>
<tr>
<td><strong>Hosts</strong></td>
<td>A comma-separated list of one or more host FQDNs to which to send the trap</td>
</tr>
<tr>
<td><strong>Port</strong></td>
<td>The port on which a process is listening for SNMP traps</td>
</tr>
</tbody>
</table>

5. In **Create Alert Notification**, click **Save**.

### Create an alert group

Use **Alerts** > **Actions** > **Manage Alert Groups** to create or edit alert groups.

#### Procedure

1. In **Ambari Web**, click **Alerts** > **Actions** > **Manage Alert Groups**.
2. In Manage Alert Groups, click + to create a new alert notification.
3. In **Create Alert Group**, enter a group name and click **Save**.
4. Click the custom group, you can add + or delete - alert definitions from this group, and change the notification targets for the group.
5. When you finish your assignments, click **Save**.

### Understanding dispatch notifications

Use dispatch notifications to communicate alert status changes.

When an alert is enabled and the alert status changes (for example, from OK to CRITICAL or CRITICAL to OK), Ambari sends either an email or SNMP notification, depending on how notifications are configured.

For email notifications, Ambari sends an email digest that includes all alert status changes. For example, if two alerts become critical, Ambari sends one email message that Alert A is CRITICAL and Ambari B alert is CRITICAL. Ambari does not send another email notification until status changes again.

For SNMP notifications, Ambari sends one SNMP trap per alert status change. For example, if two alerts become critical, Ambari sends two SNMP traps, one for each alert, and then sends two more when the two alerts change.

### Customize notification templates

The notification template content produced by Ambari is tightly coupled to a notification type. Email and SNMP notifications both have customizable templates that you can use to generate content. You can change the template used by Ambari when creating alert notifications.

#### About this task

**Alert Template XML Location**

An alert-templates.xml file ships with Ambari. This file contains all of the templates for every known type of notification, such as EMAIL and SNMP. This file is bundled in the Ambari server .jar file so that the template is not
exposed on the disk; however, that file is used in the following example. When you customize the alert template, you are effectively overriding the default alert template's XML.

**Procedure**

1. On the Ambari server host, browse to `/etc/ambari-server/conf` directory.
2. Edit the `ambari.properties` file.
3. Add an entry for the location of your new template.
   
   ```
   alerts.template.file=/foo/var/alert-templates-custom.xml
   ```
4. Save the file.
5. Restart Ambari server.

**Example**

After you restart Ambari, any notification types defined in the new template override those bundled with Ambari. If you choose to provide your own template file, you only need to define notification templates for the types that you wish to override. If a notification template type is not found in the customized template, Ambari will default to the templates that ship with the JAR.

**Example**

**Alert Template XML Structure**

The structure of the template file is defined as follows. Each `<alert-template>` element declares what type of alert notification it should be used for.

```xml
<alert-templates>
  <alert-template type="EMAIL">
    <subject>Subject Content</subject>
    <body>Body Content</body>
  </alert-template>
  <alert-template type="SNMP">
    <subject>Subject Content</subject>
    <body>Body Content</body>
  </alert-template>
</alert-templates>
```

**Example**

**Template Variables**

The template uses Apache Velocity to render all tokenized content. The following variables are available for use in your template:

- `$alert.getAlertDefinition()` - The definition of which the alert is an instance.
- `$alert.getAlertText()` - The specific alert text.
- `$alert.getAlertName()` - The name of the alert.
- `$alert.getAlertState()` - The alert state (OK, WARNING, CRITICAL, or UNKNOWN)
- `$alert.getServiceName()` - The name of the service that the alert is defined for.
- `$alert.hasComponentName()` - True if the alert is for a specific service component.
$alert.getComponentName() The component, if any, that the alert is defined for.
$alert.hasHostName() True if the alert was triggered for a specific host.
$alert.getHostName() The hostname, if any, that the alert was triggered for.
$ambari.getServerUrl() The Ambari Server URL.
$ambari.getServerVersion() The Ambari Server version.
$ambari.getServerHostName() The Ambari Server hostname.
$dispatch.getTargetName() The notification target name.
$dispatch.getTargetDescription() The notification target description.
$summary.getAlerts(service,alertState) A list of all alerts for a given service or alert state (OK|WARNING|CRITICAL|UNKNOWN)
$summary.getServicesByAlertState(alertState) A list of all services for a given alert state (OK|WARNING|CRITICAL|UNKNOWN)
$summary.getServices() A list of all services that are reporting an alert in the notification.
$summary.getCriticalCount() The CRITICAL alert count.
$summary.getOkCount() The OK alert count.
$summary.getTotalCount() The total alert count.
$summary.getUnknownCount() The UNKNOWN alert count.
$summary.getWarningCount() The WARNING alert count.
$summary.getAlerts() A list of all of the alerts in the notification.

Example

Example: Modify Alert EMAIL Subject

The following example illustrates how to change the subject line of all outbound email notifications to include a hard-coded identifier:

1. Download the alert-templates.xml code as your starting point.
2. On the Ambari Server, save the template to a location, such as:
   /var/lib/ambari-server/resources/alert-templates-custom.xml
3. Edit the alert-templates-custom.xml file and modify the subject link for the <alert-template type=”EMAIL”> template:

   <subject>
   <![CDATA[Petstore Ambari has $summary.getTotalCount() alerts!]]>
   </subject>

4. Save the file.
5. Browse to /etc/ambari-server/conf directory.
6. Edit the ambari.properties file.
7. Add an entry for the location of your new template file.

   alerts.template.file=/var/lib/ambari-server/resources/alert-templates-custom.xml

8. Save the file.
Predefined Alerts

HDFS alerts

Descriptions, potential causes and possible remedies for alerts triggered by HDFS.

Table 3: HDFS service alerts

<table>
<thead>
<tr>
<th>Alert</th>
<th>Alert Type</th>
<th>Description</th>
<th>Potential Causes</th>
<th>Possible Remedies</th>
</tr>
</thead>
<tbody>
<tr>
<td>NameNode Blocks Health</td>
<td>METRIC</td>
<td>This service-level alert is triggered if the number of corrupt or missing blocks exceeds the configured critical threshold.</td>
<td>Some DataNodes are down and the replicas that are missing blocks are only on those DataNodes. The corrupt or missing blocks are from files with a replication factor of 1. New replicas cannot be created because the only replica of the block is missing.</td>
<td>For critical data, use a replication factor of 3. Bring up the failed DataNodes with missing or corrupt blocks. Identify the files associated with the missing or corrupt blocks by running the Hadoop fsck command. Delete the corrupt files and recover them from backup, if one exists.</td>
</tr>
<tr>
<td>NFS Gateway Process</td>
<td>PORT</td>
<td>This host-level alert is triggered if the NFS Gateway process cannot be confirmed as active.</td>
<td>NFS Gateway is down.</td>
<td>Check for a non-operating NFS Gateway in Ambari Web.</td>
</tr>
<tr>
<td>DataNode Storage</td>
<td>METRIC</td>
<td>This host-level alert is triggered if storage capacity is full on the DataNode (90% critical). It checks the DataNode JMX Servlet for the Capacity and Remaining properties.</td>
<td>Cluster storage is full. If cluster storage is not full, DataNode is full.</td>
<td>If the cluster still has storage, use the load balancer to distribute the data to relatively less-used DataNodes. If the cluster is full, delete unnecessary data or add additional storage by adding either more DataNodes or more or larger disks to the DataNodes. After adding more storage, run the load balancer.</td>
</tr>
<tr>
<td>DataNode Process</td>
<td>PORT</td>
<td>This host-level alert is triggered if the individual DataNode processes cannot be established to be up and listening on the network for the configured critical threshold, in seconds.</td>
<td>DataNode process is down or not responding. DataNode are not down but is not listening to the correct network port/address.</td>
<td>Check for non-operating DataNodes in Ambari Web. Check for any errors in the DataNode logs /var/log/hadoop/hdfs and restart the DataNode, if necessary. Run the netstat-tulpn command to check if the DataNode process is bound to the correct network port.</td>
</tr>
<tr>
<td>DataNode Web UI</td>
<td>WEB</td>
<td>This host-level alert is triggered if the DataNode web UI is unreachable.</td>
<td>The DataNode process is not running.</td>
<td>Check whether the DataNode process is running.</td>
</tr>
<tr>
<td>Alert</td>
<td>Alert Type</td>
<td>Description</td>
<td>Potential Causes</td>
<td>Possible Remedies</td>
</tr>
<tr>
<td>-------</td>
<td>------------</td>
<td>-------------</td>
<td>------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>NameNode Host CPU Utilization</td>
<td>METRIC</td>
<td>This host-level alert is triggered if CPU utilization of the NameNode exceeds certain thresholds (200% warning, 250% critical). It checks the NameNode JMX Servlet for the SystemCPULoad property. This information is available only if you are running JDK 1.7.</td>
<td>Unusually high CPU utilization might be caused by a very unusual job or query workload, but this is generally the sign of an issue in the daemon.</td>
<td>Use the top command to determine which processes are consuming excess CPU. Reset the offending process.</td>
</tr>
<tr>
<td>NameNode Web UI</td>
<td>WEB</td>
<td>This host-level alert is triggered if the NameNode web UI is unreachable.</td>
<td>The NameNode process is not running.</td>
<td>Check whether the NameNode process is running.</td>
</tr>
<tr>
<td>Percent DataNodes with Available Space</td>
<td>AGGREGATE</td>
<td>This service-level alert is triggered if the storage is full on a certain percentage of DataNodes (10% warn, 30% critical).</td>
<td>Cluster storage is full. If cluster storage is not full, DataNode is full.</td>
<td>If the cluster still has storage, use the load balancer to distribute the data to relatively less-used DataNodes. If the cluster is full, delete unnecessary data or increase storage by adding either more DataNodes or more or larger disks to the DataNodes. After adding more storage, run the load balancer.</td>
</tr>
<tr>
<td>Percent DataNodes Available</td>
<td>AGGREGATE</td>
<td>This alert is triggered if the number of non-operating DataNodes in the cluster is greater than the configured critical threshold. This aggregates the DataNode process alert.</td>
<td>DataNodes are down. DataNodes are not down but are not listening to the correct network port/address.</td>
<td>Check for non-operating DataNodes in Ambari Web. Check for any errors in the DataNode logs /var/log/hadoop/hdfs and restart the DataNode hosts/processes. Run the netstat-tuplpn command to check if the DataNode process is bound to the correct network port.</td>
</tr>
<tr>
<td>NameNode RPC Latency</td>
<td>METRIC</td>
<td>This host-level alert is triggered if the NameNode operations RPC latency exceeds the configured critical threshold. Typically an increase in the RPC processing time increases the RPC queue length, causing the average queue wait time to increase for NameNode operations.</td>
<td>A job or an application is performing too many NameNode operations.</td>
<td>Review the job or the application for potential bugs causing it to perform too many NameNode operations.</td>
</tr>
<tr>
<td>NameNode Last Checkpoint</td>
<td>SCRIPT</td>
<td>This alert will trigger if the last time that the NameNode performed a checkpoint was too long ago or if the number of uncommitted transactions is beyond a certain threshold.</td>
<td>Too much time elapsed since last NameNode checkpoint. Uncommitted transactions beyond threshold.</td>
<td>Set NameNode checkpoint. Review threshold for uncommitted transactions.</td>
</tr>
<tr>
<td>Secondary NameNode Process</td>
<td>WEB</td>
<td>If the Secondary NameNode process cannot be confirmed to be up and listening on the network. This alert is not applicable when NameNode HA is configured.</td>
<td>The Secondary NameNode is not running.</td>
<td>Check that the Secondary DataNode process is running.</td>
</tr>
<tr>
<td>NameNode Directory Status</td>
<td>METRIC</td>
<td>This alert checks if the NameNode NameDirStatus metric reports a failed directory.</td>
<td>One or more of the directories are reporting as not healthy.</td>
<td>Check the NameNode UI for information about unhealthy directories.</td>
</tr>
<tr>
<td>Alert</td>
<td>Alert Type</td>
<td>Description</td>
<td>Potential Causes</td>
<td>Possible Remedies</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>HDFS Capacity Utilization</td>
<td>METRIC</td>
<td>This service-level alert is triggered if the HDFS capacity utilization exceeds the configured critical threshold (80% warn, 90% critical). It checks the NameNode JMX Servlet for the CapacityUsed and CapacityRemaining properties.</td>
<td>Cluster storage is full.</td>
<td>Delete unnecessary data. Archive unused data. Add more DataNodes. Add more or larger disks to the DataNodes. After adding more storage, run the load balancer.</td>
</tr>
<tr>
<td>DataNode Health Summary</td>
<td>METRIC</td>
<td>This service-level alert is triggered if there are unhealthy DataNodes.</td>
<td>A DataNode is in an unhealthy state.</td>
<td>Check the NameNode UI for the list of non-operating DataNodes.</td>
</tr>
<tr>
<td>HDFS Pending Deletion Blocks</td>
<td>METRIC</td>
<td>This service-level alert is triggered if the number of blocks pending deletion in HDFS exceeds the configured warning and critical thresholds. It checks the NameNode JMX Servlet for the PendingDeletionBlock property.</td>
<td>Large number of blocks are pending deletion.</td>
<td></td>
</tr>
<tr>
<td>HDFS Upgrade Finalized State</td>
<td>SCRIPT</td>
<td>This service-level alert is triggered if HDFS is not in the finalized state.</td>
<td>The HDFS upgrade is not finalized.</td>
<td>Finalize any upgrade you have in progress.</td>
</tr>
<tr>
<td>DataNode Unmounted Data Dir</td>
<td>SCRIPT</td>
<td>This host-level alert is triggered if one of the data directories on a host was previously on a mount point and became unmounted.</td>
<td>If the mount history file does not exist, then report an error if a host has one or more mounted data directories as well as one or more unmounted data directories on the root partition. This may indicate that a data directory is writing to the root partition, which is undesirable.</td>
<td>Check the data directories to confirm they are mounted as expected.</td>
</tr>
<tr>
<td>DataNode Heap Usage</td>
<td>METRIC</td>
<td>This host-level alert is triggered if heap usage goes past thresholds on the DataNode. It checks the DataNode JMXServlet for the MemHeapUsedM and MemHeapMaxM properties. The threshold values are percentages.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NameNode Client RPC Queue Latency</td>
<td>SCRIPT</td>
<td>This service-level alert is triggered if the deviation of RPC queue latency on client port has grown beyond the specified threshold within a given period. This alert will monitor Hourly and Daily periods.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NameNode Client RPC Processing Latency</td>
<td>SCRIPT</td>
<td>This service-level alert is triggered if the deviation of RPC latency on client port has grown beyond the specified threshold within a given period. This alert will monitor Hourly and Daily periods.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NameNode Service RPC Queue Latency</td>
<td>SCRIPT</td>
<td>This service-level alert is triggered if the deviation of RPC latency on the DataNode port has grown beyond the specified threshold within a given period. This alert will monitor Hourly and Daily periods.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alert</td>
<td>Alert Type</td>
<td>Description</td>
<td>Potential Causes</td>
<td>Possible Remedies</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NameNode Service RPC Processing Latency</td>
<td>SCRIPT</td>
<td>This service-level alert is triggered if the deviation of RPC latency on the DataNode port has grown beyond the specified threshold within a given period. This alert will monitor Hourly and Daily periods.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HDFS Storage Capacity Usage</td>
<td>SCRIPT</td>
<td>This service-level alert is triggered if the increase in storage capacity usage deviation has grown beyond the specified threshold within a given period. This alert will monitor Daily and Weekly periods.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NameNode Heap Usage</td>
<td>SCRIPT</td>
<td>This service-level alert is triggered if the NameNode heap usage deviation has grown beyond the specified threshold within a given period. This alert will monitor Daily and Weekly periods.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### HDFS high availability alerts

Descriptions, potential causes and possible remedies for alerts related to HDFS high availability.

**Table 4: HDFS HA Alerts**

<table>
<thead>
<tr>
<th>Alert</th>
<th>Alert Type</th>
<th>Description</th>
<th>Potential Causes</th>
<th>Possible Remedies</th>
</tr>
</thead>
<tbody>
<tr>
<td>JournalNode Web UI</td>
<td>WEB</td>
<td>This host-level alert is triggered if the individual JournalNode process cannot be established to be up and listening on the network for the configured critical threshold, given in seconds.</td>
<td>The JournalNode process is down or not responding.</td>
<td>Check if the JournalNode process is running.</td>
</tr>
<tr>
<td>NameNode High Availability Health</td>
<td>SCRIPT</td>
<td>This service-level alert is triggered if either the Active NameNode or Standby NameNode are not running.</td>
<td>The Active, Standby or both NameNode processes are down.</td>
<td>On each host running NameNode, check for any errors in the logs (/var/log/hadoop/hdfs/) and restart the NameNode host/process using Ambari Web.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>On each host running NameNode, run the netstat-tuplpn command to check if the NameNode process is bound to the correct network port.</td>
</tr>
<tr>
<td>Percent JournalNodes Available</td>
<td>AGGREGATE</td>
<td>This service-level alert is triggered if the number of down JournalNodes in the cluster is greater than the configured critical threshold (33% warn, 50% crit). It aggregates the results of JournalNode process checks.</td>
<td>JournalNodes are down.</td>
<td>Check for dead JournalNodes in Ambari Web.</td>
</tr>
<tr>
<td>ZooKeeper Failover Controller Process</td>
<td>PORT</td>
<td>This alert is triggered if the ZooKeeper Failover Controller process cannot be confirmed to be up and listening on the network.</td>
<td>The ZKFC process is down or not responding.</td>
<td>Check if the ZKFC process is running.</td>
</tr>
</tbody>
</table>
NameNode high availability alerts

Descriptions, potential causes and possible remedies for alerts related to NameNode high availability.

Table 5: NameNode HA Alerts

<table>
<thead>
<tr>
<th>Alert</th>
<th>Alert Type</th>
<th>Description</th>
<th>Potential Causes</th>
<th>Possible Remedies</th>
</tr>
</thead>
<tbody>
<tr>
<td>JournalNode Process</td>
<td>WEB</td>
<td>This host-level alert is triggered if the individual JournalNode process cannot be established to be up and listening on the network for the configured critical threshold, given in seconds.</td>
<td>The JournalNode process is down or not responding. The JournalNode is not down but is not listening to the correct network port/address.</td>
<td>Check if the JournalNode process is running.</td>
</tr>
<tr>
<td>NameNode High Availability Health</td>
<td>SCRIPT</td>
<td>This service-level alert is triggered if either the Active NameNode or Standby NameNode are not running.</td>
<td>The Active, Standby or both NameNode processes are down.</td>
<td>On each host running NameNode, check for any errors in the logs /var/log/hadoop/hdfs/ and restart the NameNode host/process using Ambari Web. On each host running NameNode, run the netstat-tuplpn command to check if the NameNode process is bound to the correct network port.</td>
</tr>
<tr>
<td>Percent JournalNodes Available</td>
<td>AGGREGATE</td>
<td>This service-level alert is triggered if the number of down JournalNodes in the cluster is greater than the configured critical threshold (33% warn, 50% crit). It aggregates the results of JournalNode process checks.</td>
<td>JournalNodes are down. JournalNodes are not down but are not listening to the correct network port/address.</td>
<td>Check for non-operating JournalNodes in Ambari Web.</td>
</tr>
<tr>
<td>ZooKeeper Failover Controller Process</td>
<td>PORT</td>
<td>This alert is triggered if the ZooKeeper Failover Controller process cannot be confirmed to be up and listening on the network.</td>
<td>The ZKFC process is down or not responding.</td>
<td>Check if the ZKFC process is running.</td>
</tr>
</tbody>
</table>

YARN alerts

Descriptions, potential causes and possible remedies for alerts triggered by YARN.

Table 6: YARN Alerts

<table>
<thead>
<tr>
<th>Alert</th>
<th>Alert Type</th>
<th>Description</th>
<th>Potential Causes</th>
<th>Possible Remedies</th>
</tr>
</thead>
<tbody>
<tr>
<td>App Timeline Web UI</td>
<td>WEB</td>
<td>This host-level alert is triggered if the App Timeline Server Web UI is unreachable.</td>
<td>The App Timeline Server is down. App Timeline Service is not down but is not listening to the correct network port/address.</td>
<td>Check for non-operating App Timeline Server in Ambari Web.</td>
</tr>
<tr>
<td>Percent NodeManagers Available</td>
<td>AGGREGATE</td>
<td>This alert is triggered if the number of down NodeManagers in the cluster is greater than the configured critical threshold. It aggregates the results of DataNode process alert checks.</td>
<td>NodeManagers are down. NodeManagers are not down but are not listening to the correct network port/address.</td>
<td>Check for non-operating NodeManagers. Check for any errors in the NodeManager logs /var/log/hadoop/yarn and restart the NodeManagers hosts/processes, as necessary. Run the netstat-tuplpn command to check if the NodeManager process is bound to the correct network port.</td>
</tr>
</tbody>
</table>
### MapReduce2 alerts

Descriptions, potential causes and possible remedies for alerts triggered by MapReduce2.

#### Table 7: MapReduce2 Alerts

<table>
<thead>
<tr>
<th>Alert</th>
<th>Alert Type</th>
<th>Description</th>
<th>Potential Causes</th>
<th>Possible Remedies</th>
</tr>
</thead>
<tbody>
<tr>
<td>History Server Web UI</td>
<td>WEB</td>
<td>This host-level alert is triggered if the HistoryServer Web UI is unreachable.</td>
<td>The HistoryServer process is not running.</td>
<td>Check if the HistoryServer process is running.</td>
</tr>
</tbody>
</table>

---

113
### HBase service alerts

Descriptions, potential causes and possible remedies for alerts triggered by HBase.

**Table 8: HBase Service Alerts**

<table>
<thead>
<tr>
<th>Alert</th>
<th>Description</th>
<th>Potential Causes</th>
<th>Possible Remedies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percent RegionServers</td>
<td>This service-level alert is triggered if the configured percentage of Region Server processes cannot be determined to be up and listening on the network for the configured critical threshold. The default setting is 10% to produce a WARNING alert and 30% to produce a CRITICAL alert. It aggregates the results of RegionServer process down checks.</td>
<td>Misconfiguration or less-than-ideal configuration caused the RegionServers to crash. Cascading failures brought on by some workload caused the RegionServers to crash. The RegionServers shut themselves down because there were problems in the dependent services, ZooKeeper or HDFS. GC paused the RegionServer for too long and the RegionServers lost contact with ZooKeeper.</td>
<td>Check the dependent services to make sure they are operating correctly. Look at the RegionServer log files (usually /var/log/hbase/*.log) for further information. If the failure was associated with a particular workload, try to understand the workload better. Restart the RegionServers.</td>
</tr>
<tr>
<td>Available</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HBase Master Process</td>
<td>This alert is triggered if the HBase master processes cannot be confirmed to be up and listening on the network for the configured critical threshold, given in seconds.</td>
<td>The HBase master process is down. The HBase master has shut itself down because there were problems in the dependent services, ZooKeeper or HDFS.</td>
<td>Check the dependent services. Look at the master log files (usually /var/log/hbase/*.log) for further information. Look at the configuration files /etc/hbase/conf. Restart the master.</td>
</tr>
<tr>
<td>HBase Master CPU Utilization</td>
<td>This host-level alert is triggered if CPU utilization of the HBase Master exceeds certain thresholds (200% warning, 250% critical). It checks the HBase Master JMX Servlet for the SystemCPULoad property. This information is only available if you are running JDK 1.7.</td>
<td>Unusually high CPU utilization: Can be caused by a very unusual job/query workload, but this is generally the sign of an issue in the daemon.</td>
<td>Use the top command to determine which processes are consuming excess CPU. Reset the offending process.</td>
</tr>
</tbody>
</table>
### Predefined Alerts

<table>
<thead>
<tr>
<th>Alert</th>
<th>Description</th>
<th>Potential Causes</th>
<th>Possible Remedies</th>
</tr>
</thead>
<tbody>
<tr>
<td>RegionServers Health Summary</td>
<td>This service-level alert is triggered if there are unhealthy RegionServers.</td>
<td>The RegionServer process is down on the host. The RegionServer process is up and running but not listening on the correct network port (default 60030).</td>
<td>Check for dead RegionServer in Ambari Web.</td>
</tr>
<tr>
<td>HBase RegionServer Process</td>
<td>This host-level alert is triggered if the RegionServer processes cannot be confirmed to be up and listening on the network for the configured critical threshold, given in seconds.</td>
<td>The RegionServer process is down on the host. The RegionServer process is up and running but not listening on the correct network port (default 60030).</td>
<td>Check for any errors in the logs /var/log/hbase/ and restart the RegionServer process using Ambari Web. Run the netstat-tuplpn command to check if the RegionServer process is bound to the correct network port.</td>
</tr>
</tbody>
</table>

### Hive alerts

Descriptions, potential causes and possible remedies for alerts triggered by Hive.

**Table 9: Hive Alerts**

<table>
<thead>
<tr>
<th>Alert</th>
<th>Description</th>
<th>Potential Causes</th>
<th>Possible Remedies</th>
</tr>
</thead>
<tbody>
<tr>
<td>HiveServer2 Process</td>
<td>This host-level alert is triggered if the HiveServer cannot be determined to be up and responding to client requests.</td>
<td>HiveServer2 process is not running. HiveServer2 process is not responding.</td>
<td>Using Ambari Web, check status of HiveServer2 component. Stop and then restart.</td>
</tr>
<tr>
<td>Hive Metastore Process</td>
<td>This host-level alert is triggered if the Hive Metastore process cannot be determined to be up and listening on the network for the configured critical threshold, given in seconds.</td>
<td>The Hive Metastore service is down. The database used by the Hive Metastore is down. The Hive Metastore host is not reachable over the network.</td>
<td>Using Ambari Web, stop the Hive service and then restart it.</td>
</tr>
<tr>
<td>WebHCat Server Status</td>
<td>This host-level alert is triggered if the WebHCat server cannot be determined to be up and responding to client requests.</td>
<td>The WebHCat server is down. The WebHCat server is hung and not responding. The WebHCat server is not reachable over the network.</td>
<td>Restart the WebHCat server using Ambari Web.</td>
</tr>
</tbody>
</table>

### Oozie alerts

Descriptions, potential causes and possible remedies for alerts triggered by Oozie.

**Table 10: Oozie Alerts**

<table>
<thead>
<tr>
<th>Alert</th>
<th>Description</th>
<th>Potential Causes</th>
<th>Possible Remedies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oozie Server Web UI</td>
<td>This host-level alert is triggered if the Oozie server Web UI is unreachable.</td>
<td>The Oozie server is down. Oozie Server is not down but is not listening to the correct network port/address.</td>
<td>Check for dead Oozie Server in Ambari Web.</td>
</tr>
<tr>
<td>Oozie Server Status</td>
<td>This host-level alert is triggered if the Oozie server cannot be determined to be up and responding to client requests.</td>
<td>The Oozie server is down. The Oozie server is hung and not responding. The Oozie server is not reachable over the network.</td>
<td>Restart the Oozie service using Ambari Web.</td>
</tr>
</tbody>
</table>
ZooKeeper alerts

Descriptions, potential causes and possible remedies for alerts triggered by ZooKeeper.

Table 11: ZooKeeper Alerts

<table>
<thead>
<tr>
<th>Alert</th>
<th>Alert Type</th>
<th>Description</th>
<th>Potential Causes</th>
<th>Possible Remedies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percent ZooKeeper Servers</td>
<td>AGGREGATE</td>
<td>This service-level alert is triggered if the configured percentage of ZooKeeper processes cannot be determined to be up and listening on the network for the configured critical threshold, given in seconds. It aggregates the results of ZooKeeper process checks.</td>
<td>The majority of your ZooKeeper servers are down and not responding.</td>
<td>Check the dependent services to make sure they are operating correctly.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Check the ZooKeeper logs /var/log/hadoop/zookeeper.log for further information.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>If the failure was associated with a particular workload, try to understand the workload better.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Restart the ZooKeeper servers from the Ambari UI.</td>
</tr>
<tr>
<td>ZooKeeper Server Process</td>
<td>PORT</td>
<td>This host-level alert is triggered if the ZooKeeper server process cannot be determined to be up and listening on the network for the configured critical threshold, given in seconds.</td>
<td>The ZooKeeper server process is down on the host. The ZooKeeper server process is up and running but not listening on the correct network port (default 2181).</td>
<td>Check for any errors in the ZooKeeper logs /var/log/hbase/ and restart the ZooKeeper process using Ambari Web.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Run the netstat-tulpn command to check if the ZooKeeper server process is bound to the correct network port.</td>
</tr>
</tbody>
</table>

Ambari alerts

Descriptions, potential causes and possible remedies for alerts triggered by Ambari.

Table 12: Ambari Alerts

<table>
<thead>
<tr>
<th>Alert</th>
<th>Alert Type</th>
<th>Description</th>
<th>Potential Causes</th>
<th>Possible Remedies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host Disk Usage</td>
<td>SCRIPT</td>
<td>This host-level alert is triggered if the amount of disk space used on a host goes above specific thresholds (50% warn, 80% crit ).</td>
<td>The amount of free disk space left is low.</td>
<td>Check host for disk space to free or add more storage.</td>
</tr>
<tr>
<td>Ambari Agent Heartbeat</td>
<td>SERVER</td>
<td>This alert is triggered if the server has lost contact with an agent.</td>
<td>Ambari Server host is unreachable from Agent host. Ambari Agent is not running.</td>
<td>Check connection from Agent host to Ambari Server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Check Agent is running.</td>
</tr>
<tr>
<td>Ambari Server Alerts</td>
<td>SERVER</td>
<td>This alert is triggered if the server detects that there are alerts which have not run in a timely manner.</td>
<td>Agents are not reporting alert status. Agents are not running.</td>
<td>Check that all Agents are running and heartbeating.</td>
</tr>
<tr>
<td>Ambari Server Performance</td>
<td>SERVER</td>
<td>This alert is triggered if the Ambari Server detects that there is a potential performance problem with Ambari.</td>
<td>This type of issue can arise for many reasons, but is typically attributed to slow database queries and host resource exhaustion.</td>
<td>Check your Ambari Server database connection and database activity. Check your Ambari Server host for resource exhaustion such as memory.</td>
</tr>
</tbody>
</table>

Ambari metrics alerts

Descriptions, potential causes and possible remedies for alerts triggered by Ambari metrics.
### Table 13: Ambari Metrics Alerts

<table>
<thead>
<tr>
<th>Alert</th>
<th>Description</th>
<th>Potential Causes</th>
<th>Possible Remedies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metrics Collector Process</td>
<td>This alert is triggered if the Metrics Collector cannot be confirmed to be up and listening on the configured port for number of seconds equal to threshold.</td>
<td>The Metrics Collector process is not running.</td>
<td>Check the Metrics Collector is running.</td>
</tr>
<tr>
<td>Metrics Collector – ZooKeeper Server Process</td>
<td>This host-level alert is triggered if the Metrics Collector ZooKeeper Server Process cannot be determined to be up and listening on the network.</td>
<td>The Metrics Collector process is not running.</td>
<td>Check the Metrics Collector is running.</td>
</tr>
<tr>
<td>Metrics Collector – HBase Master Process</td>
<td>This alert is triggered if the Metrics Collector HBase Master Processes cannot be confirmed to be up and listening on the network for the configured critical threshold, given in seconds.</td>
<td>The Metrics Collector process is not running.</td>
<td>Check the Metrics Collector is running.</td>
</tr>
<tr>
<td>Metrics Collector – HBase Master CPU Utilization</td>
<td>This host-level alert is triggered if CPU utilization of the Metrics Collector exceeds certain thresholds.</td>
<td>Unusually high CPU utilization generally the sign of an issue in the daemon configuration.</td>
<td>Tune the Ambari Metrics Collector.</td>
</tr>
<tr>
<td>Metrics Monitor Status</td>
<td>This host-level alert is triggered if the Metrics Monitor process cannot be confirmed to be up and running on the network.</td>
<td>The Metrics Monitor is down.</td>
<td>Check whether the Metrics Monitor is running on the given host.</td>
</tr>
<tr>
<td>Percent Metrics Monitors Available</td>
<td>This is an AGGREGATE alert of the Metrics Monitor Status.</td>
<td>Metrics Monitors are down.</td>
<td>Check the Metrics Monitors are running.</td>
</tr>
<tr>
<td>Metrics Collector -Auto-Restart Status</td>
<td>This alert is triggered if the Metrics Collector has been auto-started for number of times equal to start threshold in a 1 hour timeframe. By default if restarted 2 times in an hour, you will receive a Warning alert. If restarted 4 or more times in an hour, you will receive a Critical alert.</td>
<td>The Metrics Collector is running but is unstable and causing restarts. This could be due to improper tuning.</td>
<td>Tune the Ambari Metrics Collector.</td>
</tr>
<tr>
<td>Percent Metrics Monitors Available</td>
<td>This is an AGGREGATE alert of the Metrics Monitor Status.</td>
<td>Metrics Monitors are down.</td>
<td>Check the Metrics Monitors.</td>
</tr>
<tr>
<td>Grafana Web UI</td>
<td>This host-level alert is triggered if the AMS Grafana Web UI is unreachable.</td>
<td>Grafana process is not running.</td>
<td>Check whether the Grafana process is running. Restart if it has gone down.</td>
</tr>
</tbody>
</table>

### SmartSense alerts

Descriptions, potential causes and possible rememdies for alerts triggered by SmartSense.

### Table 14: SmartSense Alerts

<table>
<thead>
<tr>
<th>Alert</th>
<th>Description</th>
<th>Potential Causes</th>
<th>Possible Remedies</th>
</tr>
</thead>
<tbody>
<tr>
<td>SmartSense Server Process</td>
<td>This alert is triggered if the HST server process cannot be confirmed to be up and listening on the network for the configured critical threshold, given in seconds.</td>
<td>HST server is not running.</td>
<td>Start HST server process. If startup fails, check the hst-server.log.</td>
</tr>
<tr>
<td>SmartSense Bundle Capture Failure</td>
<td>This alert is triggered if the last triggered SmartSense bundle is failed or timed out.</td>
<td>Some nodes are timed out during capture or fail during data capture. It could also be because upload to Hortonworks fails.</td>
<td>From the Bundles page check the status of bundle. Next, check which agents have failed or timed out, and review their logs. You can also initiate a new capture.</td>
</tr>
<tr>
<td>Alert</td>
<td>Description</td>
<td>Potential Causes</td>
<td>Possible Remedies</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SmartSense Long Running Bundle</td>
<td>This alert is triggered if the SmartSense in-progress bundle has possibility of not completing successfully on time.</td>
<td>Service components that are getting collected may not be running. Or some agents may be timing out during data collection/upload.</td>
<td>Restart the services that are not running. Force-complete the bundle and start a new capture.</td>
</tr>
<tr>
<td>SmartSense Gateway Status</td>
<td>This alert is triggered if the SmartSense Gateway server process is enabled but is unable to reach.</td>
<td>SmartSense Gateway is not running.</td>
<td>Start the gateway. If gateway start fails, review hst-gateway.log</td>
</tr>
</tbody>
</table>