Hortonworks Data Platform

Ambari User's Guide

Hortonworks Data Platform : Ambari User's Guide

Copyright © 2012-2014 Hortonworks, Inc. Some rights reserved.

The Hortonworks Data Platform, powered by Apache Hadoop, is a massively scalable and 100% open source platform for storing, processing and analyzing large volumes of data. It is designed to deal with data from many sources and formats in a very quick, easy and cost-effective manner. The Hortonworks Data Platform consists of the essential set of Apache Hadoop projects including MapReduce, Hadoop Distributed File System (HDFS), HCatalog, Pig, Hive, HBase, Zookeeper and Ambari. Hortonworks is the major contributor of code and patches to many of these projects. These projects have been integrated and tested as part of the Hortonworks Data Platform release process and installation and configuration tools have also been included.

Unlike other providers of platforms built using Apache Hadoop, Hortonworks contributes 100% of our code back to the Apache Software Foundation. The Hortonworks Data Platform is Apache-licensed and completely open source. We sell only expert technical support, training and partner-enablement services. All of our technology is, and will remain free and open source.

Please visit the Hortonworks Data Platform page for more information on Hortonworks technology. For more information on Hortonworks services, please visit either the Support or Training page. Feel free to Contact Us directly to discuss your specific needs.

Table of Contents

List of Figures

List of Tables

1. Introducing 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 a non-trivial task. To help you deal with the complexity, Apache Ambari collects a wide range of information from the cluster's nodes and services and presents them to you in an easy to read and use centralized web interface, Ambari Web. Ambari Web displays information such as service-specific summaries, graphs, and alerts. It also allows you to perform basic management tasks such as starting and stopping services, adding hosts to your cluster, and updating service configurations.

> **Note**
>
> At this time, Ambari Web is supported only in deployments made using the Ambari Install Wizard.

1.1. Architecture

The Ambari Server serves as the collection point for data from across your cluster. Each host has a copy of the Ambari Agent - either installed automatically by the Install wizard or manually - which allows the Ambari Server to control each host. In addition, each host has a copy of Ganglia Monitor (`gmond`), which collects metric information that is passed to the Ganglia Connector, and then on to the Ambari Server.

**Figure 1.1. Architectural Overview**

1.1.1. Sessions

Ambari Web is a client-side JavaScript application, which 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 and communication between the browser and server occur asynchronously via the REST API.

> **Note**
>
> Ambari Web sessions do not timeout since the application is constantly accessing the REST API, which resets the session timeout. As well, if there is a period of Ambari Web inactivity, the Ambari Web interface is automatically refreshed. Therefore you must **explicitly sign out** of the Ambari Web interface to destroy the Ambari session with the server.
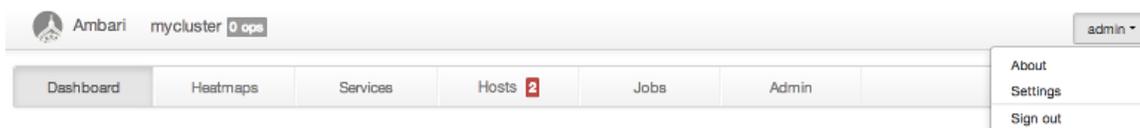
1.2. Starting and Accessing Ambari Web

Generally the Ambari Server and Ambari Web are started as part of the installation process. If for some reason the server is not running, on the Ambari Server machine, type:

```
ambari-server start
```

To access Ambari Web, open a supported browser and enter the Ambari Web URL:

```
http://{your.ambari.server}:8080
```

Enter your username and password. If this is the first time Ambari Web is accessed, use the default values, `admin/admin`. These values can be changed, and new users provisioned, using the **Admin** view in Ambari Web itself.

2. Navigating Ambari Web

This section gives you an overview of using the Ambari Web GUI for monitoring and managing your Hadoop 2 cluster.

2.1. The Navigation Header

At the top of every screen is the Navigation Header.

| Dashboard | Heatmaps | Services | Hosts | Jobs | Admin |

This header appears in all views in Ambari Web. Use this bar to move from view to view. The active view is indicated with a slightly darker gray.
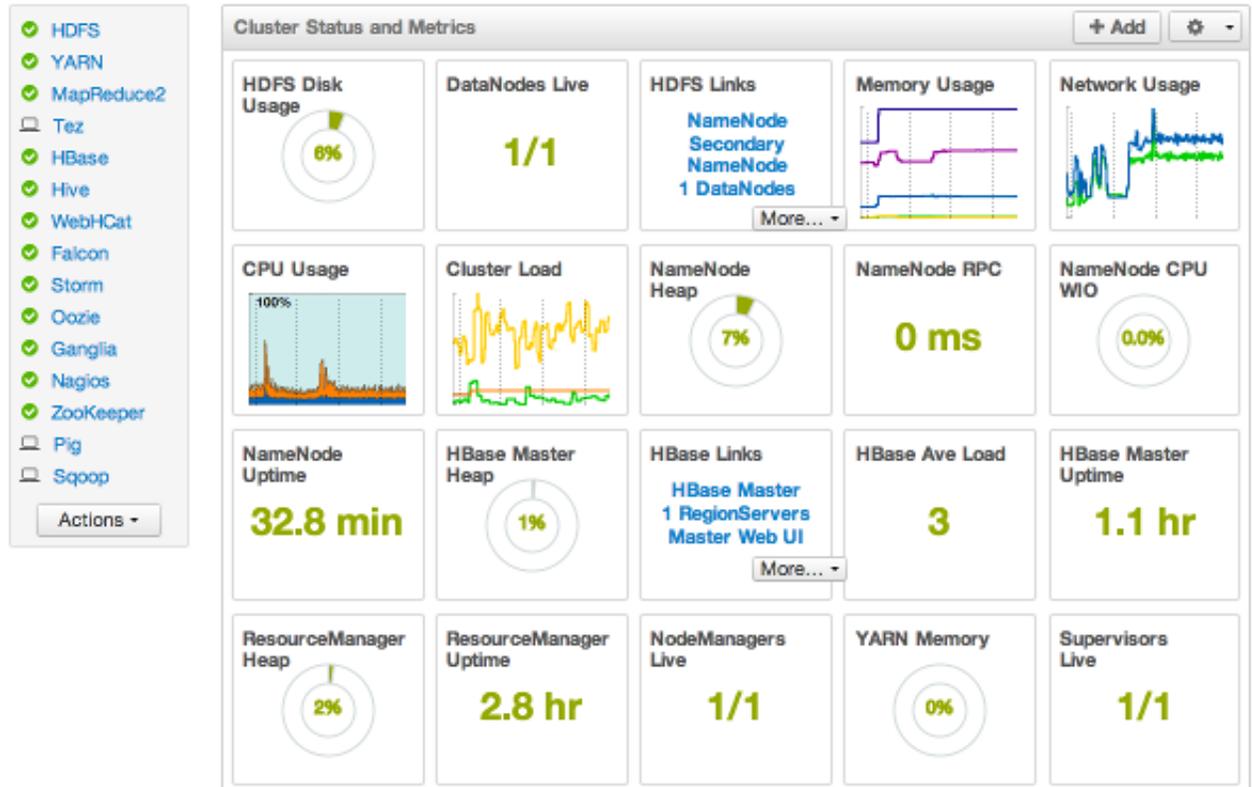
2.2. The Dashboard View

When you open Ambari Web, you are placed in the **Dashboard** view. This view is divided into two versions:

• The Widget version

• The Classic version

2.2.1. The Widget Version

The **Dashboard** Widget version opens first. This view gives you a customizable overview of the state of your cluster as a whole.

The Widget version is divided into two main sections:

- Services Summary

- Cluster Status and Metrics

2.2.1.1. Services Summary

Notice the color of the dot appearing next to each service name in the list of **Services** . The dot color and blinking action indicates operating status of each service. For example, in the Dashboard Widget [3] image, notice green dot next to each service name. The following colors and actions indicate service status:

Table 2.1. Service Status

| Color	Name	Status
	Solid Green	All masters are running
	Blinking Green	Starting up
	Solid Red	At least one master is down
	Blinking Red	Stopping

Click the service name to open the **Services** screen, where you can see more detailed information on each service.

2.2.1.2. Cluster Status and Metrics

On the main section of the screen [3], a customizable set of widget tiles presents information on the status of your cluster. There are simple pie and bar charts, more complex usage and load charts, and sets of links to additional data sources, as well as status information like uptime and average RPC queue wait times.

Table 2.2. Widget Interactions

To:	Do:
Move widgets around on the screen	Drag and drop to desired position
See more detailed information	Hover over the widget box
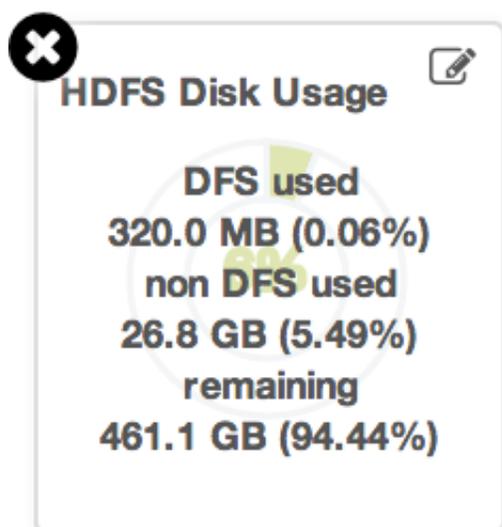
Currently, Ambari displays Metrics for the following services:
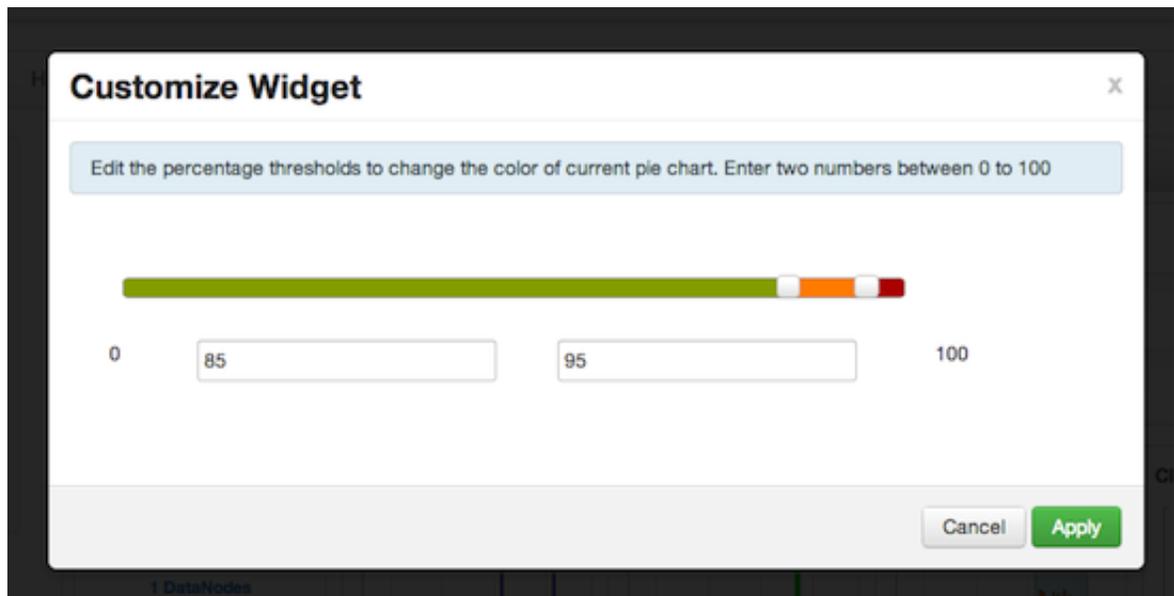
- HDFS

- YARN

- HBase

- Storm

> **Note**
>
> Metrics data for Storm is buffered and sent as a batch to Ambari every 5 minutes. Anticipate a five-minute delay for appearance of Storm metrics after adding the Storm service.

Table 2.3. Widget Interactions 2

To:	Do:
Delete a widget	Click the X, marked in red

To:	Do:
Edit a widget	Click the small edit icon, marked in blue. The **Customize Widget** popup appears.

Follow the instructions in the popup to customize the widget display. In this case, you can adjust the thresholds at which the **HDFS Capacity** bar chart changes color, from green to orange to red. Click **Apply** to save your changes. Not all widgets have an edit icon.

Table 2.4. Widget Interactions 3

To:	Do:
Add back a deleted widget	Click **+Add** and use the checkbox to select the widget you want from the dropdown menu
Use quick links to other information, like thread stacks, logs and native component GUIs	Click **More** and select from the dropdown menu
Zoom into more complex charts, like the Network Usage chart.	Click the chart. A larger version displays. You can choose to add and remove information from the chart by selecting or deselecting the metric in the legend
Switch to the Classic version, return widgets to default setting, or view metrics in the native Ganglia Server interface	Use the gear icon
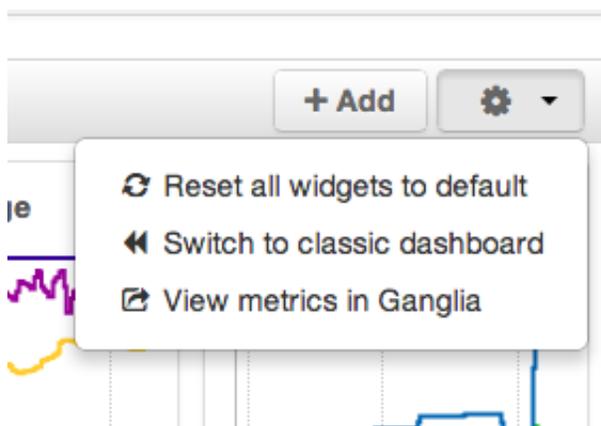
2.2.2. The Classic Version

The **Dashboard** Classic version provides a slightly different view of high-level cluster information.
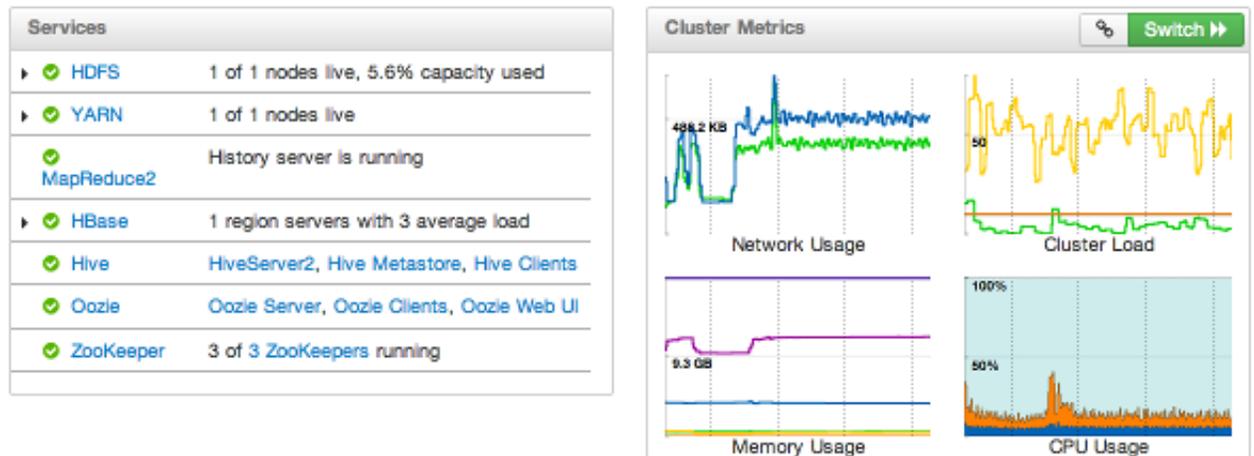
The Classic version is also divided into two main sections:

• Services Status

• Cluster Metrics

2.2.3. Services Status

Notice the color of the dot appearing next to each service name in the list of **Services** . The dot color and blinking action indicates operating status of each service. For example, in the Classic Dashboard image [7], notice the green dots next to **HDFS**, **YARN**, **MapReduce2, Hive**, **Oozie**, and **Zookeeper**. Notice the red dot next to **HBase**. The following colors and actions indicate service status:

Table 2.5. Service Status Indicators

Color	Action	Status
✅	Solid Green	All masters are running
✅	Blinking Green	Starting up
⚠	Solid Red	At least one master is down
⚠	Blinking Red	Stopping

Notice also in the Classic Dashboard image [7], the small triangles next to HDFS, YARN, and HBase services. These triangles point to services deployed in master/slave sets. Select the triangle to open a panel showing more detailed status information about the service.

Notice especially a small, red, numbered rectangle, such as the following example highlighted blue, appearing next to a service name.

A red, numbered rectangle shows how many current alerts a service generates, indicating necessary actions.


Select a service name to open a more detailed **Services** screen that displays more detailed alert information.

2.2.4. Cluster Metrics

On the right side of the screen is the **Cluster Metrics** section. This section gives you charts for a snapshot of the important cluster-wide metrics: **Network Usage**, **Cluster Load**, **Memory Usage**, and **CPU Usage**. To see a legend for the chart, hover over it. To remove a metric from the chart, click on the legend in the metric to remove the checkmark and deselect it.

To see a larger view of the chart, click the chart. Ambari displays a larger version of the chart in a pop-out window.
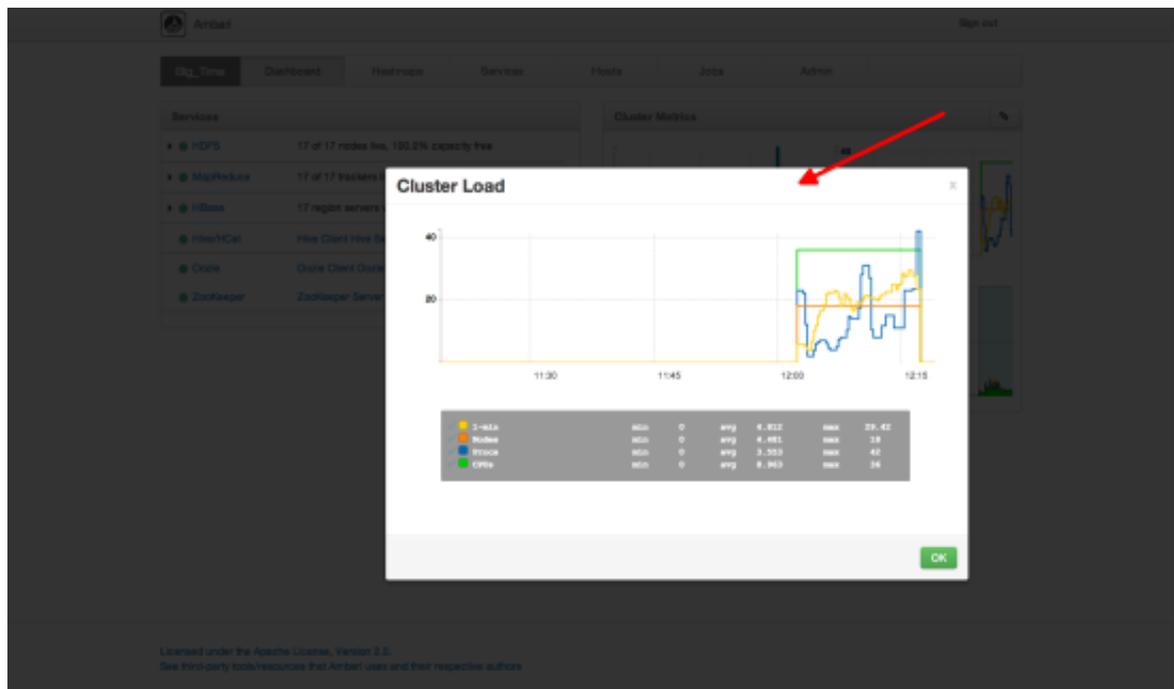
Notice the link symbol on the upper right side of the Cluster Metric section, outlined in blue in the overview screenshot [7] above. This is a link to the GUI for the Ganglia Server itself, where you can find much more detailed information on your cluster. You can also use this to switch back to the Widget version.

2.3. The Heatmaps View

The **Heatmaps** view gives you a graphic representation of the overall utilization of your cluster using simple color coding.
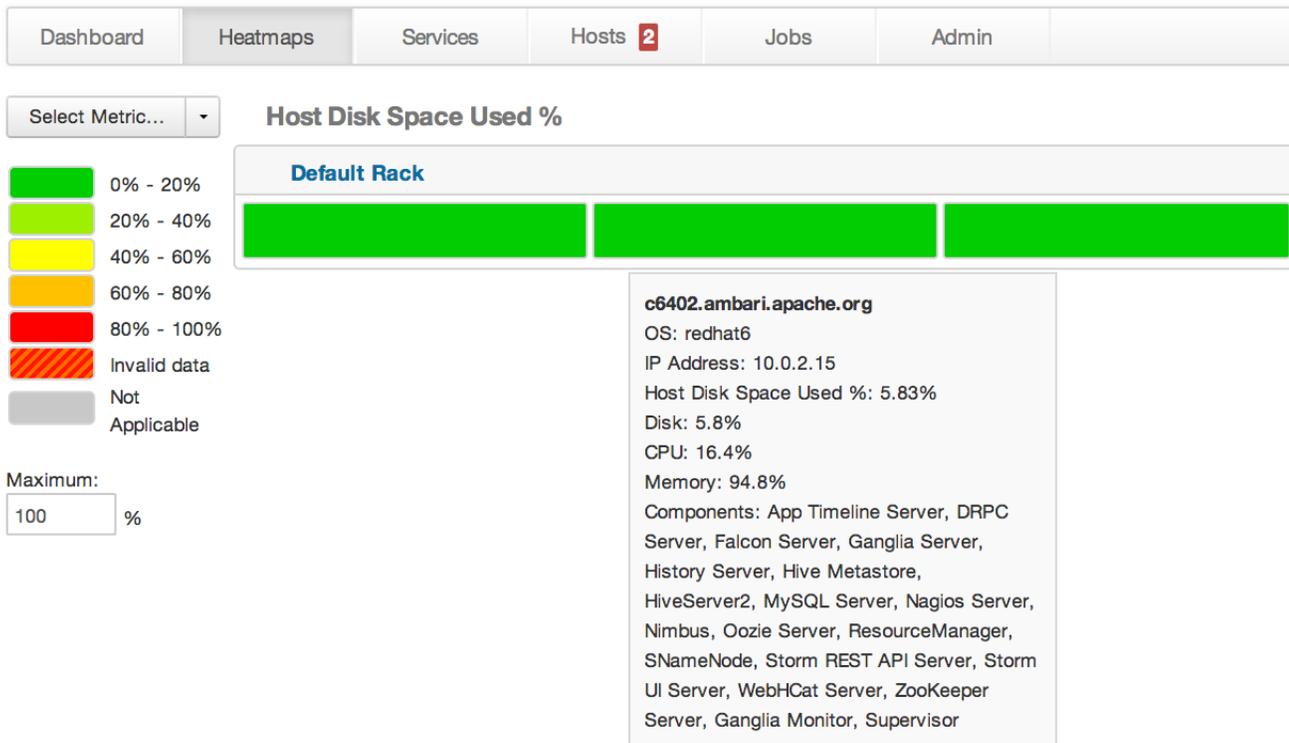
Each host in the cluster is represented by a block. To see more information on a specific host, hover over the block you are interested in, and a popup with key host data appears. The color of the blocks represents usage in an appropriate unit based on a selectable set of metrics. If the data necessary to determine state are not all available, the block is marked as having Invalid Data. Changing the default maximum values for the heatmap lets you fine tune the representation. Use the Select Metric dropdown to select the metric type.
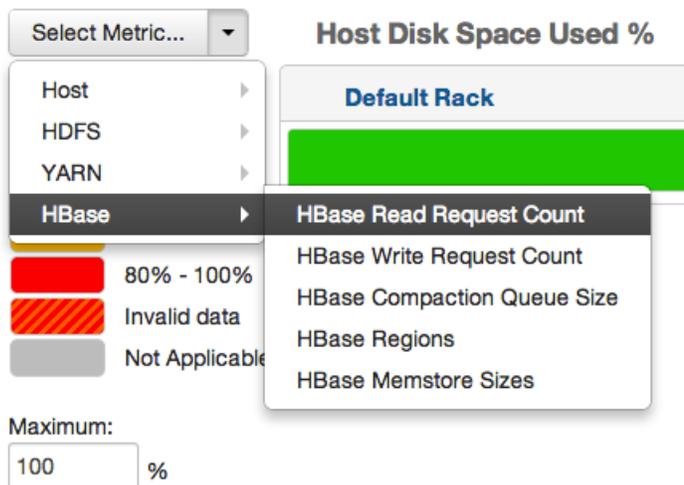
Currently the following metrics are supported:

Metric	Uses
Host/Disk Space Used %	disk.disk_free and disk.disk_total

Host/Memory Used %	memory.mem_free and memory.mem_total
Host/CPU Wait I/O %	cpu.cpu_wio
HDFS/Bytes Read	dfs.datanode.bytes_read
HDFS/Bytes Written	dfs.datanode.bytes_written
HDFS/Garbage Collection Time	jvm.gcTimeMillis
HDFS/JVM Heap MemoryUsed	jvm.memHeapUsedM
YARN/Garbage Collection Time	jvm.gcTimeMillis
YARN / JVM Heap Memory Used	jvm.memHeapUsedM
YARN / Memory used %	UsedMemoryMB and AvailableMemoryMB
HBase/RegionServer read request count	hbase.regionserver.readRequestsCount
HBase/RegionServer write request count	hbase.regionserver.writeRequestsCount
HBase/RegionServer compaction queue size	hbase.regionserver.compactionQueueSize
HBase/RegionServer regions	hbase.regionserver.regions
HBase/RegionServer memstore sizes	hbase.regionserver.memstoreSizeMB

2.4. Monitoring and Managing Services

Use **Services** to monitor and manage selected services running in your Hadoop cluster.

All services installed in your cluster are listed in the leftmost **Services** panel.

Services supports the following tasks:

- Starting and Stopping All Services

- Selecting a Service

- Viewing Summary, Alert, and Health Information

- Configuring Services

- Rolling Restarts

- Using Quick Links

- Analyzing Service Metrics

2.4.1. Starting and Stopping All Services

To start or stop all listed services at once, select **Actions**, then choose **Start All** or **Stop All**. Actions is outlined red in the Services [11] image.

2.4.2. Selecting a Service

Selecting a service name from the list shows current summary, alert, and health information for the selected service. To refresh the monitoring panels and show information about a different service, select a different service name from the list.

Notice the colored dot next to each service name, indicating service operating status and a small, red, numbered rectangle indicating any alerts generated for the service.

2.4.2.1. Adding a Service

The Ambari install wizard installs all available Hadoop services by default. You may choose to deploy only some services initially, then add other services at later times. For example, many customers deploy only core Hadoop services initially. **Add Service**, outlined red in Services [11], supports deploying additional services without interrupting operations in your Hadoop cluster. When you have deployed all available services, **Add Service** displays disabled.

For example, if you are using HDP 2.1 Stack and did not install Falcon or Storm, you can use the **Add Service** capability to add those services to your cluster.

To add a service, select **Actions** -> **Add Service**, then complete the following procedure using the Add Service Wizard.

2.4.2.1.1. Adding a Service to your Hadoop cluster

This example shows the Falcon service selected for addition.

1. Choose Services.

   Choose an available service. Alternatively, choose all to add all available services to your cluster. Then, choose Next.

   The Add Services wizard displays installed services highlighted green and marked unavailable for selection.

2. In **Assign Masters**, confirm the default host assignment. Alternatively, choose a different host machine to which master components for your selected service will be added. Then, choose Next.

The Add Services Wizard indicates hosts on which the master components for a chosen service will be installed. A service chosen for addition displays as:

• A green label located on the host to which its master components will be added, or

• An active drop-down list on which available host names appear.
Using the drop-down, choose an alternate host name, if necessary.
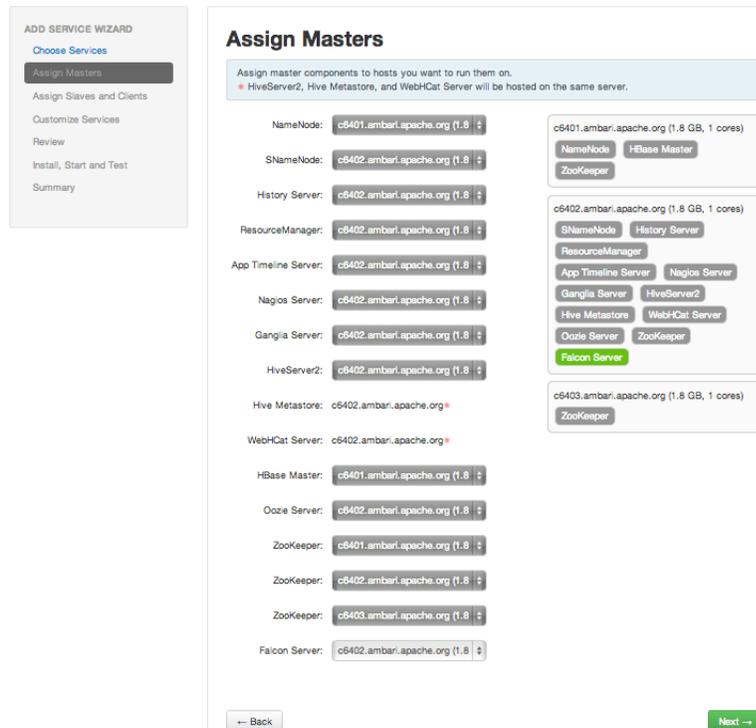
3. In Assign Slaves and Clients, accept the default assignment of slave and client components to hosts. Then, choose Next.

   Alternatively, select hosts to which you want to assign slave and client components.

   You must select at least one host for the slave of each service being added.

   Table 2.6. Host Roles Required for Added Services

Service Added	Host Role Required
MapReduce	TaskTracker
YARN	NodeManager
HBase	RegionServer

   The Add Service Wizard skips and disables the Assign Slaves and Clients step for a service requiring no slave nor client assignment.
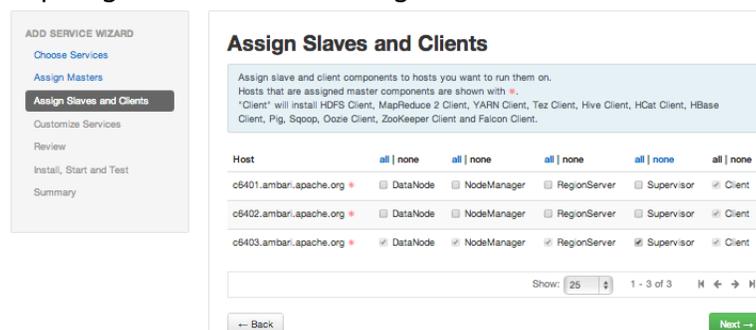
4. In Customize Services, accept the default configuration properties. Then, choose Next.

Alternatively, edit the default values for configuration properties, if necessary.

Choose Override to create a configuration group for this service.

5. In Review, make sure the configuration settings match your intentions. Then, choose Deploy.

6. Monitor the progress of installing, starting, and testing the service. When the service installs and starts successfully, choose Next.

7. Summary displays the results of installing the service. Choose Complete.

8. Restart the Nagios service and any other components having stale configurations.

![Important icon]

Important

If you do not restart Nagios service after completing the Add Service Wizard, alerts and notifications may not work properly.

2.4.3. Viewing Summary, Alert, and Health Information

After you select a service, **Summary** tab displays basic information about the selected service. Select a **View Host** link, outlined gray in the Summary section of Services [11] to open the **Host Details** view of the host on which the selected service is running.

2.4.3.1. Alerts and Health Checks

View results of the health checks performed on your cluster by Nagios in the **Alerts and Health Checks** panel. Alerts list a brief summary of each issue and its rating, sorted first by descending severity, then by descending time. To access more detailed information, select the native Nagios GUI link outlined yellow in the upper right corner of the Alerts and Health checks [11] panel. Use the Nagios credentials you set up during installation to log in.

2.4.4. Configuring Services

Select a service, then select **Configs**, outlined orange in Services [11] to view and update configuration properties for the selected service. For example, select MapReduce2, then select Configs. Expand a config category to view configurable service properties.

2.4.4.1. Updating Service Properties

1. Expand a configuration category.

2. Edit values for one or more properties having the Override option.

   Edited values, also called stale configs, show an Undo option.

3. Choose Save.

2.4.4.2. Customizing Logging Properties

Ambari Web 1.5.0 displays default logging properties in -> Service Configs -> Custom log 4j Properties. Log 4j properties control logging activities for the selected service.

Restarting components in the service pushes the configuration properties displayed in Custom log 4j Properties to each host running components for that service. If you have

customized logging properties that define how activities for each service are logged, you will see refresh indicators next to each service name after upgrading to Ambari 1.5.0 or higher. Make sure that logging properties displayed in Custom log 4j Properties include any customization. Optionally, you can create configuration groups that include custom logging properties. For more information about about saving and overriding configuration settings, see Managing Configuration Groups.

2.4.4.3. Managing 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. In Configs, select **Manage Config Groups**, to create new groups, re-assign hosts, and override default settings for host components you assign to each group.
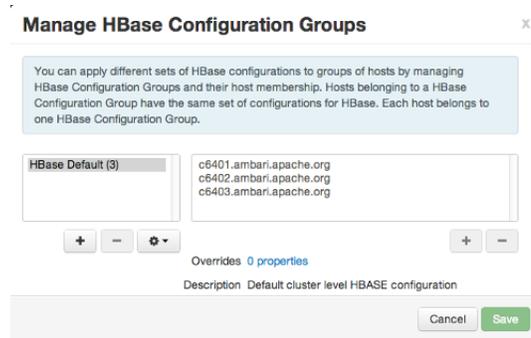
To create a Configuration Group:

1. Choose Add New Configuration Group.

2. Name and Describe the group, then choose Save.

3. Select a Config Group, then choose Add Hosts to Config Group.

4. Select Components and choose from available Hosts to add hosts to the new group.

   Select Configuration Group Hosts enforces host membership in each group, based on installed components for the selected service.

5. Choose OK.

6. In Manage Configuration Groups, choose Save.

To edit settings for a configuration group:

1. In Configs, choose a Group.

2. Select a Config Group, then expand components to expose settings that allow Override.

3. Provide a non-default value, then choose Override or Save.

   Configuration groups enforce configuration properties that allow override, based on installed components for the selected service and group.

4. Override prompts you to choose one of the following options:

   a. Select an existing configuration group (to which the property value override provided in step 3 will apply), or

   b. Create a new configuration group (which will include default properties, plus the property override provided in step 3).

   c. Then, choose OK.

5. In Configs, choose Save.

2.4.4.4. Restarting components

After editing and saving a service configuration, Restart indicates components that you must restart.

Select the Components or Hosts links to view details about components or hosts requiring a restart.

Then, choose an option appearing in Restart. For example, options to restart YARN components include:
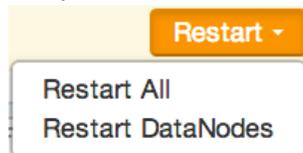
2.4.4.5. Performing Service Actions

Manage a selected service on your cluster by performing service actions. Select the Service Actions drop-down menu, outlined blue in Services [11], then choose an option. Available options depend on the service you have selected. For example, HDFS service action options include:
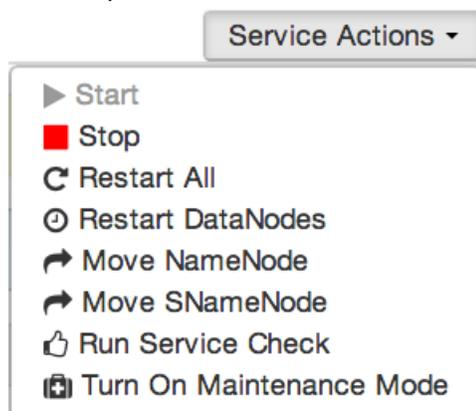
Optionally, choose **Turn On Maintenance Mode** to suppress alerts about this service before performing a service action. Maintenance Mode suppresses alerts and status indicator changes generated by the service, while allowing you to start, stop, restart, move, or perform maintenance tasks on the service. For more information about how Maintenance Mode affects bulk operations for host components, see Maintenance Mode.

2.4.4.6. Rolling Restarts

When you restart multiple services, components, or hosts, use rolling restarts to distribute the task; minimizing cluster downtime and service disruption. A rolling restart stops, then starts multiple, running slave components such as DataNodes, TaskTrackers, NodeManagers, RegionServers, or Supervisors, using a batch sequence. You set rolling restart parameter values to control the number of, time between, tolerance for failures, and limits for restarts of many components across large clusters.

To run a rolling restart:

1. Select a Service, then link to a lists of specific components or hosts that Require Restart.

2. Select Restart, then choose a slave component option.

3. Review and set values for Rolling Restart Parameters.

4. Optionally, reset the flag to only restart components with changed configurations.

5. Choose Trigger Restart.

Use Background Operations to monitor progress of rolling restarts.

2.4.4.6.1. Setting Rolling Restart Parameters

When you choose to restart slave components, 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 3-node cluster restarts 1 component at a time, waits 2 minutes between restarts, will proceed if only one failure occurs, and restarts all existing components that run this service.

If you trigger a rolling restart of components, Restart components with stale configs defaults to true. If you trigger a rolling restart of services, Restart services with stale configs defaults to false.

Rolling restart parameter values must satisfy the following criteria:

Table 2.7. Validation Rules for Rolling Restart Parameters

Parameter	Required	Value	Description
Batch Size	Yes	Must be an integer > 0	Number of components to include in each restart batch.
Wait Time	Yes	Must be an integer > = 0	Time (in seconds) to wait between queuing each batch of components.
Tolerate up to x failures	Yes	Must be an integer > = 0	Total number of restart failures to tolerate, across all batches, before halting the restarts and not queuing batches.

2.4.4.6.2. Aborting a Rolling Restart

To abort future restart operations in the batch, choose Abort Rolling Restart.

2.4.4.7. Monitoring Background Operations

Optionally, use Background Operations to monitor progress and completion of bulk operations such as rolling restarts.

Background Operations opens by default when you run a job that executes bulk operations.

1. Select the right-arrow for each operation to show restart operation progress on each host.

2. After restarts complete, Select the right-arrow, or a host name, to view log files and any error messages generated on the selected host.

3. Select links at the upper-right to copy or open text files containing log and error information.

Optionally, select the option to not show the bulk operations dialog.

2.4.4.8. Using Quick Links

Select **Quick Links** options, outlined in light green in the Services image [11], to access additional sources of information about a selected service. For example, HDFS Quick Links options include the native NameNode GUI, NameNode logs, the NameNode JMX output, and thread stacks for the HDFS service. Quick Links are not available for every service.

2.4.4.9. Analyzing Service Metrics

Review visualizations in **Metrics** that chart common metrics for the service. Hover your cursor over a chart to view the chart legend. Select a chart to view a larger version and legend together. To get more information, select the link that opens he native Ganglia GUI, outlined in purple in the Services image [11].

2.5. Managing Hosts

Use Ambari Hosts to manage multiple Hadoop components such as DataNodes, NameNodes, TaskTrackers 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 supports filtering your selection of host components, based on operating status, host health, and defined host groupings.

2.5.1. Working with Hosts

Use Hosts to view hosts in your cluster on which Hadoop services run. Use Actions, outlined green in the Hosts [25] image, to perform actions on one or more hosts in your cluster.

View individual hosts, listed by fully-qualified domain name, on the Hosts landing page.

2.5.2. Determining Host Status

A colored dot beside each host name indicates operating status of each host, as follows:

• Red - At least one master component on that host is down. Hover 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.

A red condition flag overrides an orange condition flag, which overrides a yellow condition flag. In other words, a host having a master component down may also have other issues. The following example shows three hosts, one having a master component down, one

having a slave component down, and one healthy. Warning indicators appear next to hosts having a component down.
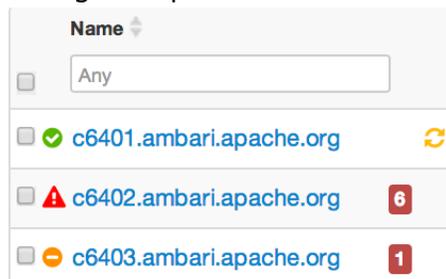
2.5.3. Filtering the Hosts List

Use Filters, outlined red in the Hosts [25] image, to limit listed hosts to only those having a specific operating status. The number of hosts in your cluster having a listed operating status appears after each status name, in parenthesis. For example, the following cluster has one host having healthy status and three hosts having Maintenance Mode turned on.

For example, to limit the list of hosts appearing on Hosts home to only those with Healthy status, select Filters, then choose the Healthy option. In this case, one host name appears on Hosts home. Alternatively, to limit the list of hosts appearing on Hosts home to only those having Maintenance Mode on, select Filters, then choose the Maintenance Mode option. In this case, three host names appear on Hosts home.

Use the general filter tool, outlined blue in the Hosts image, to apply specific search and sort criteria that limits the list of hosts appearing on Hosts home.

2.5.4. Performing Host-Level Actions

Use Actions, outlined green in the Hosts image, to act on one, or multiple hosts in your cluster. Actions performed on multiple hosts are also known as bulk operations.

Actions comprises three menus that list the following options types:

• Hosts - lists selected, filtered or all hosts options, based on your selections made using Hosts home and Filters.

• Objects - lists component objects that match your host selection criteria.

• Operations - lists all operations available for the component objects you selected.

For example, to restart DataNodes on one host:

1. In Hosts, select a host running at least one DataNode.

2. In Actions, choose Selected Hosts -> DataNodes -> Restart, as shown in the following image.
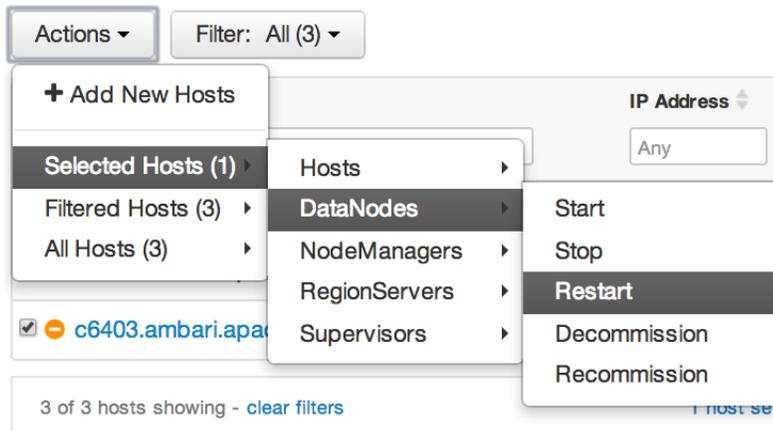
3. Choose OK to confirm starting the selected operation.

4. Optionally, use Background Operations to monitor progress of the restart operation.

2.5.5. Viewing Components on a Host

To manage components running on a specific host, choose a FQDN on Hosts Home. For example, choose c6403.ambari.apache.org in the default example shown. Summary-Components lists all components installed on that host.

Choose options in **Host Actions**, to start, stop, restart, delete, or turn on maintenance mode for all components installed on the selected host.

Alternatively, choose action options from the dropdown next to an individual component on a host. The dropdown shows current operation status for each component, For example, you can restart, stop, decommission, or turn on Maintenance Mode for the DataNode component (started) for HDFS, by selecting one of the following options:
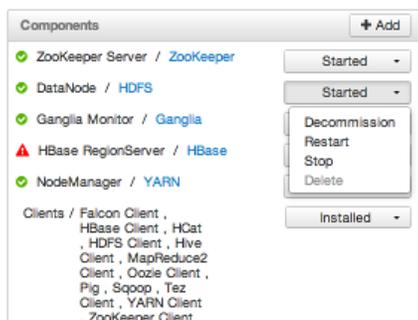
2.6. Maintenance Mode

Maintenance Mode supports suppressing alerts and skipping bulk operations for specific services, components and hosts in an Ambari-managed cluster. You typically turn on Maintenance Mode when performing hardware or software maintenance, changing configuration settings, troubleshooting, decommissioning, or removing cluster nodes. You may place a service, component, or host object in Maintenance Mode before you perform necessary maintenance or troubleshooting tasks.

Maintenance Mode affects a service, component, or host object in the following two ways:

• maintenance mode suppresses alerts, warnings and status change indicators generated for the object

• maintenance mode exempts an object from host-level or service-level bulk operations

Explicitly turning on Maintenance Mode for a service implicitly turns on Maintenance Mode for components and hosts that run the service. While Maintenance Mode On prevents bulk operations being performed on the service, component, or host, you may explicitly start and stop a service, component, or host having Maintenance Mode On.

2.6.1. Setting Maintenance Mode for Services, Components, and Hosts

For example, examine using Maintenance Mode in a 3-node, Ambari-managed cluster installed using default options. This cluster has one data node, on host c6403. This 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.

2.6.1.1. How to Turn On Maintenance Mode for a Service

1. Using Services, select HDFS.

2. Select Service Actions, then choose Turn On Maintenance Mode.

3. Choose OK to confirm.

   Notice, on Services Summary that Maintenance Mode turns on for the NameNode and SNameNode components.

2.6.1.2. How to Turn On Maintenance Mode for a Host

1. Using Hosts, select c6401.ambari.apache.org.

2. Select Host Actions, then choose Turn On Maintenance Mode.

3. Choose OK to confirm.

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

2.6.1.3. How to Turn On Maintenance Mode for a Host (alternative using filtering for hosts)

1. Using Hosts, select c6403.ambari.apache.org.

2. In Actions -> Selected Hosts -> Hosts choose Turn On Maintenance Mode.

3. Choose OK to confirm.

   Notice that Maintenance Mode turns on for host c6403.ambari.apache.org.

Your list of Hosts now shows Maintenance Mode On for hosts c6401 and c6403.

- Hover your cursor over each Maintenance Mode icon appearing in the Hosts list.

  - Notice that hosts c6401 and c6403 have Maintenance Mode On.

  - Notice that on host c6401; Ganglia Monitor, HbaseMaster, HDFS client, NameNode, and Zookeeper Server have Maintenance Mode turned On.

  - Notice on host c6402, that HDFS client and Secondary NameNode have Maintenance Mode On.

  - Notice on host c6403, that 15 components have Maintenance Mode On.

- The following behavior also results:

  - Alerts are suppressed for the DataNode.

- DataNode is skipped from HDFS Start/Stop/Restart All, Rolling Restart.

- DataNode is skipped from all Bulk Operations except Turn Maintenance Mode ON/ OFF.

- DataNode is skipped from Start All and / Stop All components.

- DataNode is skipped from a host-level restart/restart all/stop all/start.

2.6.1.4. Maintenance Mode Use Cases

Four common Maintenance Mode Use Cases follow:

1. You want to perform hardware, firmware, or OS maintenance on a host.

   You want to:

   - Prevent alerts generated by all components on this host.

   - Be able to stop, start, and restart each component on the host.

   - Prevent host-level or service-level bulk operations from starting, stopping, or restarting components on this host.

   To achieve these goals, turn On Maintenance Mode explicitly for the host. Putting a host in Maintenance Mode implicitly puts all components on that host in Maintenance Mode.

2. You want to test a service configuration change. You will stop, start, and restart the service using a rolling restart to test whether restarting picks up the change.

   You want:

   - No alerts generated by any components in this service.

   - To prevent host-level or service-level bulk operations from starting, stopping, or restarting components in this service.

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

3. You turn off a service completely.

   You want:

   - The service to generate no warnings.

   - To ensure that no components start, stop, or restart due to host-level actions or bulk operations.

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

4. A host component is generating alerts.

You want to:

- Check the component.

- Assess warnings and alerts generated for the component.

- Prevent alerts generated by the component while you check its condition.

To achieve these goals, turn on Maintenance Mode explicitly for the host component. Putting a host component in Maintenance Mode prevents host-level and service-level bulk operations from starting or restarting the component. You can restart the component explicitly while Maintenance Mode is on.

2.6.2. Decommissioning Master and Slave Nodes

Decommissioning is a process that supports removing a slave component from the cluster. You must decommission the slave running on a host before removing the component or host from service to avoid potential loss of data or disruption on processing. Decommissioning is available for the following component types:

- DataNodes

- NodeManagers

- TaskTrackers

- RegionServers

Decommissioning executes the following steps:

- For DataNodes, safely replicates the HDFS data to other DataNodes in the cluster.

- For NodeManagers and TaskTrackers, stops accepting new job requests from the masters and stops the component.

- For RegionServers, turns on drain mode and stops the component.

To initiate a decommission, browse the host details page and select the Decommission option from the Actions menu next to the component you want to decommission. The UI shows "Decommissioning" status while steps process, then "Decommissioned" when complete.

For example, to move a DataNode slave from one host to another, decommission the DataNode.

2.6.3. Deleting a Host from a Cluster

Deleting a host removes the host from the cluster. Before deleting a host, you must complete the following prerequisites:

• 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 Maintenance Mode, if necessary, for the host.

2.6.3.1. How to Delete a Host from a Cluster

1. In Hosts, click on a host name.

2. On the Host-Details page, select Host Actions drop-down menu.

3. Choose Delete.

If you have not completed prerequisite steps, a warning message similar to the following one appears:
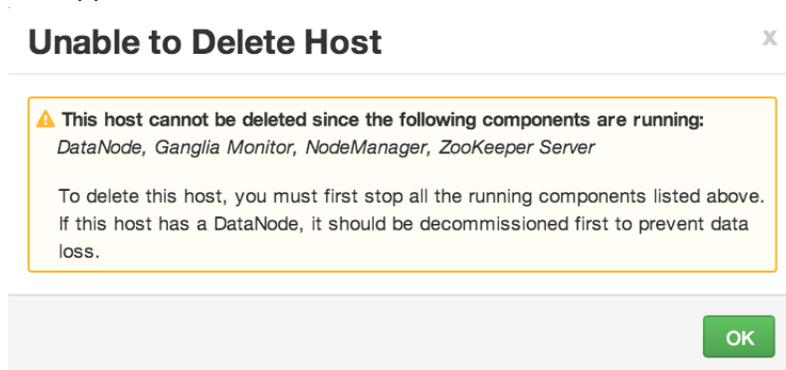
2.6.4. Adding Hosts to a Cluster

To add new hosts to your cluster, browse to the Hosts page and select **+Add New Hosts**, from the Actions menu. The **Add Host Wizard** provides a sequence of prompts similar to those in the Ambari Install Wizard. Follow the prompts, providing information simliar to that provided to define the first set of hosts in your cluster.

To review the Ambari Install Wizard procedure, see Installing, Configuring, and Deploying the Cluster in the Ambari Installation Guide.

2.6.5. Admin View

The **Admin** view allows you to manage Ambari Web users and check for general information about the cluster.

2.6.5.1. Managing Ambari Web Users

By default, Ambari is configured for a lcoal user store for authentication. This section describes managing users in the local user store. Optionally, you can configure Ambari to use LDAP or Active Directory for authentication. For more information, see Setting Up LDAP or Active Directory Authentication.

Select Users in the left nav bar to add Ambari Web users, make them administrators, delete them, or change their passwords. There are two user roles: **User** and **Admin**. **Users** can view metrics, view service status and configuration, and browse job information. **Admins** can do all User tasks and in addition can start or stop services, modify configurations, and run smoke tests.
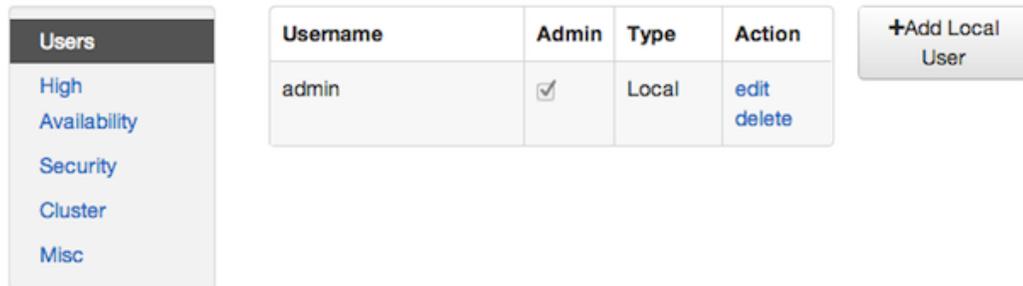
To make a change in a current user's password, click **edit**, in red above. To add an additional Ambari Web user, click **+Add Local User**. In both cases, a variant of the username pop up appears.
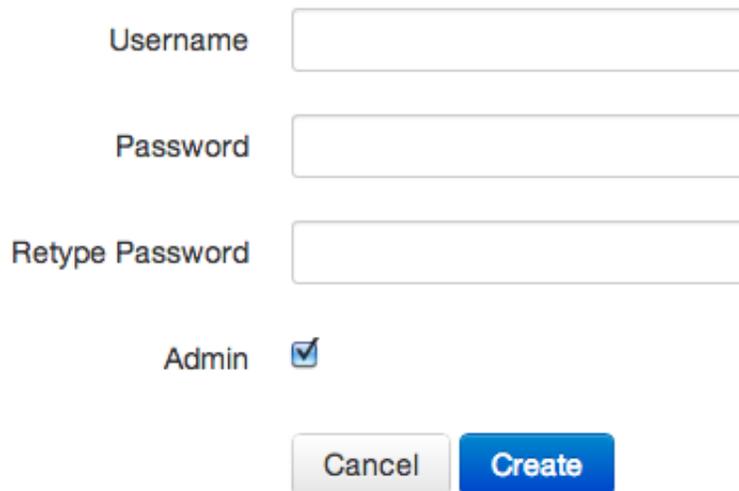
For a new Ambari Web user, enter the desired **Username** and **Password**. Check **Admin** to make this user an administrator. Click **Create** to make the user.

For an Ambari Web existing user, enter the old and new **Passwords**. Make your changes and click **Save**.

To delete an existing Ambari Web user, click **delete** and fill in the **Username**.

2.6.5.2. Managing NameNode High Availabilty for Hadoop 2.x

To mange availability of NameNodes, you must first enable NameNode High Availability on your cluster. For instructions to enable NameNode High Availability, see Setting Up NameNode High Availability.

2.6.5.3. Enabling Kerberos Security

To turn on Kerberos-based security you must:

1. Have already set up Kerberos for your cluster. For more information on setting up Kerberos, see Setting Up Kerberos for Use with Ambari

2. Click **Enable Security** and follow the Enable Security Wizard.
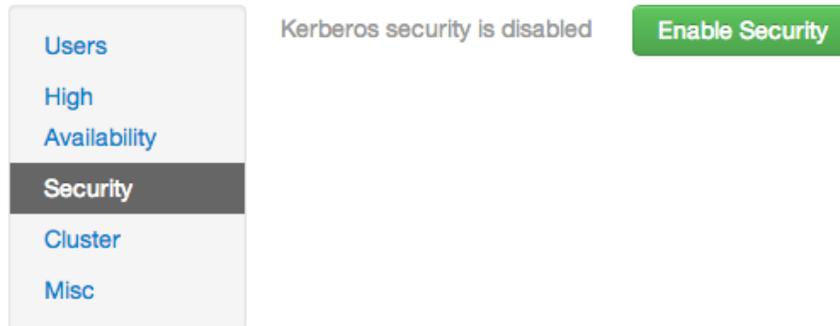
   a. **Get Started**: This step just reminds you that you need to set up Kerberos before you start.

   b. **Configure Services**: This step asks you for your Kerberos information: principals and paths to keytabs, path to your Kerberos tools, realm names and so on. For more information about a specific field, hover over it, and a popup with a definition appears.

   c. **Create Principals and Keytabs**: Use this step to check that all your information is correct. Click **Back** to make any changes. Click **Apply** when you are satisfied with the assignments.

      > **Note**
      >
      > If you have a large cluster, you may want to go to the **Create Principals and Keytabs** step first. Step through the wizard accepting the defaults to get to the appropriate page. On the page, use the **Download CSV** button to get a list of all the necessary principals and keytabs in CSV form, which can be used to set up a script. The list includes hostname, principal description, principal name, keytab user, keytab group, keytab permissions, absolute keytab path, and keytab filename.

d. **Save and Apply Configuration**: This step displays a bar showing the progress of integrating the Kerberos information into your Ambari Server.

2.6.5.4. Checking Stack and Component Versions

To check which version of the Stack you are using and to see the component versions that are included in that Stack, click **Cluster**.

2.6.5.5. Checking Service User Accounts and Groups

To check the service user accounts and groups that have been assigned to various Hadoop services, click the **Misc** tab.

Service Users and Groups
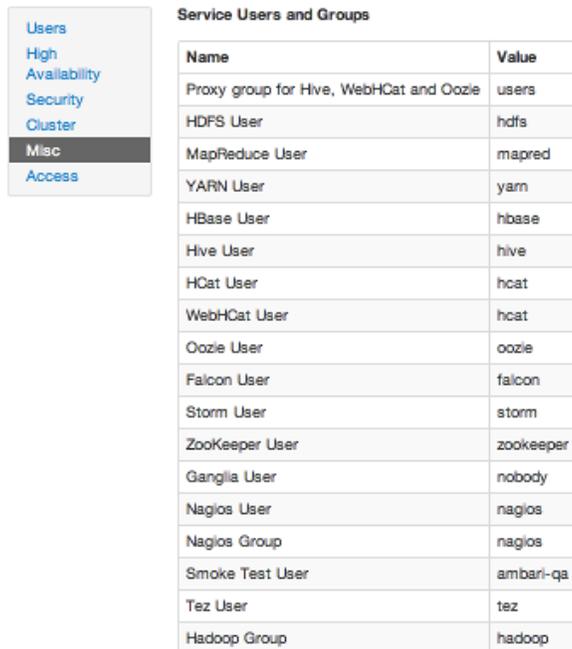
Name	Value
Proxy group for Hive, WebHCat and Oozie	users
HDFS User	hdfs
MapReduce User	mapred
YARN User	yarn
HBase User	hbase
Hive User	hive
HCat User	hcat
WebHCat User	hcat
Oozie User	oozie
Falcon User	falcon
Storm User	storm
ZooKeeper User	zookeeper
Ganglia User	nobody
Nagios User	nagios
Nagios Group	nagios
Smoke Test User	ambari-qa
Tez User	tez
Hadoop Group	hadoop

Users
High Availability
Security
Cluster
Misc
Access

3. Using Nagios With Hadoop

Nagios is an open source network monitoring system designed to monitor all aspects of your Hadoop cluster (such as hosts, services, and so forth) over the network. It can monitor many facets of your installation, ranging from operating system attributes like CPU and memory usage to the status of applications, files, and more. Nagios provides a flexible, customizable framework for collecting data on the state of your Hadoop cluster.

Nagios is primarily used for the following kinds of tasks:

• Getting instant information about your organization's Hadoop infrastructure

• Detecting and repairing problems, and mitigating future issues, before they affect end-users and customers

• Leveraging Nagios' event monitoring capabilities to receive alerts for potential problem areas

• Analyzing specific trends; for example: what is the CPU usage for a particular Hadoop service weekdays between 2 p.m. and 5 p.m

For more information, see the Nagios website at http://www.nagios.org.

> **Note**
>
> Nagios is an optional component of Ambari. During cluster install you can choose to install and configure Nagios. When selected, out-of-the-box Ambari provides a set of Nagios plugins specially designed for monitoring important aspects of your Hadoop cluster, based on your Stack selection.

3.1. Basic Nagios Architecture

Using the open source monitoring system Nagios, Ambari gathers information on the status of both of the hosts and the services that run on them.

• **Host and System Information**: Ambari monitors basic host and system information such as CPU utilization, disk I/O bandwidth and operations per second, average memory and swap space utilization, and average network latency.

• **Service Information**: Ambari monitors the health and performance status of each service by presenting information generated by that service. Because services that run in master/slave configurations (HDFS, MapReduce, and HBase) are fault tolerant in regard to service slaves, master information is presented individually, whereas slave information is presented largely in aggregate.

• **Alert Information**: Using Nagios with Hadoop-specific plugins and configurations, Ambari Web can issue alerts based on service states defined on three basic levels:

   • OK

   • Warning

- Critical

The thresholds for these alerts can be tuned using configuration files, and new alerts can be added. For more details on Nagios architecture, see the Nagios Overview at the `nagios.org` web site.

3.2. Installing Nagios

The Ambari Installation Wizard gives you the option of installing and configuring Nagios, including the out-of-the-box plugins for Hadoop-specific alerts. The Nagios server, Nagios plugins, and the web-based user interface are installed on the Nagios server host, as specified during the installation procedure.

Some of the Nagios plugins also require the Simple Network Management Protocol (SNMP) daemon to work, so the daemon is automatically installed on all cluster nodes if Nagios is installed. For example, the SNMP daemon is used to collect information about system resources (like memory, storage, swap space, and CPU utilization).

> **Note**
>
> By default the Nagios server does not use any authentication or encryption scheme while communicating with SNMP daemons. However you can enable secure communication with SNMP daemons by setting the appropriate SNMP configuration.

3.3. Configuration File Locations

All Hadoop-specific configurations are added to Nagios through files prefixed with "hadoop-" located in the `/etc/nagios/objects` directory of the Nagios Server host. The default general Nagios configuration file, `nagios.cfg` (in `/etc/nagios`), is set up to pick up the new Hadoop specific configurations from this directory.

Hadoop-specific plugins are stored in the Nagios plugins directory,`/usr/lib64/nagios/plugins/`.

By default, the Nagios server runs as a user named "nagios" which is in a group also named "nagios". The user and group can be customized during the Ambari Cluster Install (Cluster Install Wizard > Customize Services > Misc). Once Nagios is installed, use Ambari Web to start and stop the Nagios server.

3.4. Configuring Nagios Alerts For Hadoop Services

For each alert, the out of the box Hadoop Nagios configuration file defines default values for the following Nagios directives:

Warning threshold   The value that produces a warning alert.

Critical threshold   The value that produces a critical alert.

Check interval	The number of minutes between regularly scheduled checks on the host as long as the check does not change the state.
Retry interval	The number of minutes between "retries" When a service changes state, Nagios can confirm that state change by retrying the check multiple times. This retry interval can be different than the original check interval.
Maximum number of check attempts	The max number of retry attempts. Usually when the state of a service changes, this change is considered "soft" until multiple retries confirm it. Once the state change is confirmed, it is considered "hard". Ambari Web displays hard states for all the Nagios Hadoop specific checks.

3.5. Nagios Alerts For Hadoop Services

The following section provides more information on the various Hadoop alerts provided by Ambari. All these alerts are displayed in Ambari Web and in the native Nagios web interface.

There are two types of alerts configured out-of-the-box with Ambari:

- **Host-level Alerts**

  These alerts are related to a specific host and specific component running on that host. These alerts check a component and system-level metrics to determine health of the host.

- **Service-level Alerts**

  These alerts are related to a Hadoop Service and do not pertain to a specific host. These alerts check one or more components of a service as well as system-level metrics to determine overall health of a Hadoop Service.

3.5.1. HDFS Service Alerts

These alerts are used to monitor the HDFS service.

3.5.1.1. Blocks health

This service-level alert is triggered if the number of corrupt or missing blocks exceeds the configured critical threshold. This alert uses the `check_hdfs_blocks` plugin.

3.5.1.1.1. Potential causes

- Some DataNodes are down and the replicas that are missing blocks are only on those DataNodes

- The corrupt/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

3.5.1.1.2. Possible remedies

- 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 it exists

3.5.1.2. NameNode process

This host-level alert is triggered if the NameNode process cannot be confirmed to be up and listening on the network for the configured critical threshold, given in seconds. It uses the Nagios `check_tcp`

3.5.1.2.1. Potential causes

- The NameNode process is down on the HDFS master host

- The NameNode process is up and running but not listening on the correct network port (default 8201)

- The Nagios server cannot connect to the HDFS master through the network.

3.5.1.2.2. Possible remedies

- Check for any errors in the logs (`/var/log/hadoop/hdfs/`)and restart the NameNode host/process using the **HMC Manage Services** tab.

- Run the `netstat-tuplpn` command to check if the NameNode process is bound to the correct network port

- Use `ping` to check the network connection between the Nagios server and the NameNode

3.5.1.3. DataNode space

This host-level alert is triggered if storage capacity is full on the DataNodes. It uses the `check_snmp_storage` plugin.

3.5.1.3.1. Potential causes

- Cluster storage is full

- If cluster storage is not full, DataNode is full

3.5.1.3.2. Possible remedies

- If cluster still has storage, use 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 Balancer

3.5.1.4. DataNode process

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, given in seconds. It uses the Nagios `check_tcp` plugin.

3.5.1.4.1. Potential causes

- DataNode process is down or not responding

- DataNode are not down but is not listening to the correct network port/address

- Nagios server cannot connect to the DataNodes

3.5.1.4.2. Possible remedies

- Check for dead 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-tuplpn` command to check if the DataNode process is bound to the correct network port

- Use `ping` to check the network connection between the Nagios server and the DataNode

3.5.1.5. NameNode host CPU utilization

This ost-level alert is triggered if the percent of CPU utilization on the master host exceeds the configured critical threshold. It uses the Nagios `check_snmp_load` plugin.

3.5.1.5.1. Potential causes

- 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.

- A down SNMP daemon on the master node, producing an unknown status

3.5.1.5.2. Possible remedies

- Use the `top` command to determine which processes are consuming excess CPU.

- Reset the offending process

- Check the status of the SNMP daemon

3.5.1.6. NameNode edit logs directory status

This host-level alert is triggered if the NameNode cannot write to one of its configured edit log directories.

3.5.1.6.1. Potential causes

- At least one of the multiple edit log directories is mounted over NFS and has become unreachable

- The permissions on at least one of the multiple edit log directories has become read-only

3.5.1.6.2. Possible remedies

- Check permissions on all edit log directories

- Use the `dfs.name.dir` parameter in the `hdfs-site.xml` file on the NameNode to identify the locations of all the edit log directories for the NameNode. Check whether the NameNode can reach all those locations.

3.5.1.7. NameNode Web UI

This host-level alert is triggered if the NameNode Web UI is unreachable.

3.5.1.7.1. Potential causes

- The NameNode Web UI is unreachable from the Nagios Server.

- The NameNode process is not running

3.5.1.7.2. Possible remedies

- Check whether the NameNode process is running

- Check whether the Nagios Server can ping the NameNode server.

- Using a browser, check whether the Nagios Server can reach the NameNode Web UI.

3.5.1.8. Percent DataNodes with space available

This service-level alert is triggered if storage capacity is full on the DataNodes. All the local data partitions storing HDFS data are checked against the total capacity across all the partitions. It uses the `check_snmp_storage` plugin.

3.5.1.8.1. Potential causes

- Cluster storage is full

- If cluster storage is not full, DataNode is full

3.5.1.8.2. Possible remedies

- If cluster still has storage, use 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 Balancer

3.5.1.9. Percent DataNodes live

This alert is triggered if the number of down DataNodes in the cluster is greater than the configured critical threshold. It uses the `check_aggregate` plugin to aggregate the results of `Data node process` checks.

3.5.1.9.1. Potential causes

- DataNodes are down

- DataNodes are not down but are not listening to the correct network port/address

- Nagios server cannot connect to one or more DataNodes

3.5.1.9.2. Possible remedies

- Check for dead 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.

- Use `ping` to check the network connection between the Nagios server and the DataNodes.

3.5.1.10. NameNode RPC latency

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. It uses the Nagios `check_rpcq_latency` plugin.

3.5.1.10.1. Potential causes

- A job or an application is performing too many NameNode operations.

3.5.1.10.2. Possible remedies

- Review the job or the application for potential bugs causing it to perform too many NameNode operations.

3.5.1.11. HDFS capacity utilization

This service-level alert is triggered if the HDFS capacity utilization exceeds the configured critical threshold. It uses the `check_hdfs_capacity` plugin.

3.5.1.11.1. Potential causes

- Cluster storage is full

3.5.1.11.2. Possible remedies

- Delete unnecessary data.

- Archive unused data.

- Add more DataNodes.

- Add more or larger disks to the DataNodes.

- After adding more storage, run Balancer.

3.5.2. NameNode HA Alerts (Hadoop 2 only)

These alerts are available only when you are using Hadoop 2.x and you have enabled NameNode HA.

3.5.2.1. JournalNode process

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. It uses the Nagios `check_tcp` plugin.

3.5.2.1.1. Potential causes

- The JournalNode process is down or not responding.

- The JournalNode is not down but is not listening to the correct network port/address.

- The Nagios server cannot connect to the JournalNode.

3.5.2.1.2. Possible remedies

- Check if the JournalNode process is dead.

- Use `ping` to check the network connection between the Nagios server and the JournalNode host.

3.5.2.2. NameNode HA Healthy process

This service-level alert is triggered if either the Active NameNode or Standby NameNode are not running.

3.5.2.2.1. Potential causes

- The Active, Standby or both NameNode processes are down.

- The Nagios Server cannot connect to one or both NameNode hosts.

3.5.2.2.2. Possible remedies

- 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.

- Use `ping` to check the network connection between the Nagios server and the hosts running NameNode.

3.5.3. YARN Alerts (Hadoop 2 only)

These alerts are used to monitor YARN.

3.5.3.1. ResourceManager process

This host-level alert is triggered if the individual ResourceManager process cannot be established to be up and listening on the network for the configured critical threshold, given in seconds. It uses the Nagios `check_tcp` plugin.

3.5.3.1.1. Potential causes

- The ResourceManager process is down or not responding.

- The ResourceManager is not down but is not listening to the correct network port/ address.

- Nagios Server cannot connect to the ResourceManager

3.5.3.1.2. Possible remedies

- Check for a dead ResourceManager.

- Check for any errors in the ResourceManager logs (`/var/log/hadoop/yarn`) and restart the ResourceManager, if necessary.

- Use `ping` to check the network connection between the Nagios Server and the ResourceManager host.

3.5.3.2. Percent NodeManagers live

This alert is triggered if the number of down NodeManagers in the cluster is greater than the configured critical threshold. It uses the `check_aggregate` plugin to aggregate the results of DataNode process alert checks.

3.5.3.2.1. Potential causes

- NodeManagers are down.

- NodeManagers are not down but are not listening to the correct network port/address .

- Nagios server cannot connect to one or more NodeManagers.

3.5.3.2.2. Possible remedies

- Check for dead 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.

- Use `ping` to check the network connection between the Nagios Server and the NodeManagers host.

3.5.3.3. ResourceManager Web UI

This host-level alert is triggered if the ResourceManager Web UI is unreachable.

3.5.3.3.1. Potential causes

- The ResourceManager Web UI is unreachable from the Nagios Server.

- The ResourceManager process is not running.

3.5.3.3.2. Possible remedies

- Check if the ResourceManager process is running.

- Check whether the Nagios Server can ping the ResourceManager server.

- Using a browser, check whether the Nagios Server can reach the ResourceManager Web UI.

3.5.3.4. ResourceManager RPC latency

This host-level alert is triggered if the ResourceManager 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 ResourceManager operations. It uses the Nagios `check_rpcq_latency` plugin.

3.5.3.4.1. Potential causes

- A job or an application is performing too many ResourceManager operations.

3.5.3.4.2. Possible remedies

- Review the job or the application for potential bugs causing it to perform too many ResourceManager operations.

3.5.3.5. ResourceManager CPU utilization

This host-level alert is triggered if the percent of CPU utilization on the ResourceManager exceeds the configured critical threshold. This alert uses the Nagios `check_snmp_load` plugin.

3.5.3.5.1. Potential causes

- 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.

- A down SNMP daemon on the ResourceManager node, producing an unknown status.

3.5.3.5.2. Possible remedies

- Use the `top` command to determine which processes are consuming excess CPU.

- Reset the offending process.

- Check the status of the SNMP daemon.

3.5.3.6. NodeManager process

This host-level alert is triggered if the NodeManager process cannot be established to be up and listening on the network for the configured critical threshold, given in seconds. It uses the Nagios `check_tcp` plugin.

3.5.3.6.1. Potential causes

- NodeManager process is down or not responding.

- NodeManager is not down but is not listening to the correct network port/address.

- Nagios Server cannot connect to the NodeManager

3.5.3.6.2. Possible remedies

- Check if the NodeManager is running.

- Check for any errors in the NodeManager logs (`/var/log/hadoop/yarn`) and restart the NodeManager, if necessary.

- Use `ping` to check the network connection between the Nagios Server and the NodeManager host.

3.5.3.7. NodeManager health

This host-level alert checks the node health property available from the NodeManager component.

3.5.3.7.1. Potential causes

- Node Health Check script reports issues or is not configured.

3.5.3.7.2. Possible remedies

- Check in the NodeManager logs (`/var/log/hadoop/yarn`) for health check errors and restart the NodeManager, and restart if necessary.

- Check in the ResoruceManager UI logs (`/var/log/hadoop/yarn`) for health check errors.

3.5.4. MapReduce2 Alerts (Hadoop 2 only)

These alerts are used to monitor MR2.

3.5.4.1. HistoryServer Web UI

This host-level alert is triggered if the HistoryServer Web UI is unreachable.

3.5.4.1.1. Potential causes

- The HistoryServer Web UI is unreachable from the Nagios Server.

- The HistoryServer process is not running.

3.5.4.1.2. Possible remedies

- Check if the HistoryServer process is running.

- Check whether the Nagios Server can ping the HistoryServer server.

- Using a browser, check whether the Nagios Server can reach the HistoryServer Web UI.

3.5.4.2. HistoryServer RPC latency

This host-level alert is triggered if the HistoryServer 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. It uses the Nagios `check_rpcq_latency` plugin.

3.5.4.2.1. Potential causes

- A job or an application is performing too many HistoryServer operations

3.5.4.2.2. Possible remedies

- Review the job or the application for potential bugs causing it to perform too many HistoryServer operations.

3.5.4.3. HistoryServer CPU utilization

This host-level alert is triggered if the percent of CPU utilization on the HistoryServer exceeds the configured critical threshold. This alert uses the Nagios `check_snmp_load` plug-in.

3.5.4.3.1. Potential causes

- 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.

- A down SNMP daemon on the HistoryServer node, producing an unknown status

3.5.4.3.2. Possible remedies

- Use the `top` command to determine which processes are consuming excess CPU.

- Reset the offending process.

- Check the status of the SNMP daemon.

3.5.4.4. HistoryServer process

This host-level alert is triggered if the HistoryServer process cannot be established to be up and listening on the network for the configured critical threshold, given in seconds. It uses the Nagios `check_tcp` plugin.

3.5.4.4.1. Potential causes

- HistoryServer process is down or not responding.

- HistoryServer is not down but is not listening to the correct network port/address.

- Nagios Server cannot connect to the HistoryServer,

3.5.4.4.2. Possible remedies

- Check the HistoryServer is running.

- Check for any errors in the HistoryServer logs (`/var/log/hadoop/mapred`) and restart the HistoryServer, if necessary

- Use `ping` to check the network connection between the Nagios Server and the HistoryServer host.

3.5.5. MapReduce Service Alerts (Hadoop 1 only)

These alerts are used to monitor the MapReduce service.

3.5.5.1. JobTracker RPC latency alert

This host-level alert is triggered if the JobTracker 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 JobTracker operations. This alert uses the Nagios `check_rpcq_latency` plugin.

3.5.5.1.1. Potential causes

- A job or an application is performing too many JobTracker operations.

3.5.5.1.2. Possible remedies

- Review the job or the application for potential bugs causing it to perform too many JobTracker operations

3.5.5.2. JobTracker process

This host-level alert is triggered if the individual JobTracker process cannot be confirmed to be up and listening on the network for the configured critical threshold, given in seconds. It uses the Nagios `check_tcp` plugin.

3.5.5.2.1. Potential causes

- JobTracker process is down or not responding.

- JobTracker is not down but is not listening to the correct network port/address.

- The Nagios server cannot connect to the JobTracker

3.5.5.2.2. Possible remedies

- Check if the JobTracker process is running.

- Check for any errors in the JobTracker logs (`/var/log/hadoop/mapred`) and restart the JobTracker, if necessary

- Use `ping` to check the network connection between the Nagios Server and the JobTracker host.

3.5.5.3. JobTracker Web UI

This Host-level alert is triggered if the JobTracker Web UI is unreachable.

3.5.5.3.1. Potential causes

- The JobTracker Web UI is unreachable from the Nagios Server.

- The JobTracker process is not running.

3.5.5.3.2. Possible remedies

- Check if the JobTracker process is running.

- Check whether the Nagios Server can ping the JobTracker server.

- Using a browser, check whether the Nagios Server can reach the JobTracker Web UI.

3.5.5.4. JobTracker CPU utilization

This host-level alert is triggered if the percent of CPU utilization on the JobTracker exceeds the configured critical threshold. This alert uses the Nagios `check_snmp_load` plug-in.

3.5.5.4.1. Potential causes

- 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.

- A down SNMP daemon on the JobTracker node, producing an unknown status.

3.5.5.4.2. Possible remedies

- Use the `top` command to determine which processes are consuming excess CPU

- Reset the offending processor.

- Check the status of the SNMP daemon.

3.5.5.5. HistoryServer Web UI

This host-level alert is triggered if the HistoryServer Web UI is unreachable.

3.5.5.5.1. Potential causes

- The HistoryServer Web UI is unreachable from the Nagios Server.

- The HistoryServer process is not running.

- Using a browser, check whether the Nagios Server can reach the HistoryServer Web UI.

3.5.5.5.2. Possible remedies

- Check the HistoryServer process is running.

- Check whether the Nagios Server can ping the HistoryServer server.

- Check the status of the SNMP daemon.

3.5.5.6. HistoryServer process

This host-level alert is triggered if the HistoryServer process cannot be established to be up and listening on the network for the configured critical threshold, given in seconds. It uses the Nagios `check_tcp` plugin.

3.5.5.6.1. Potential causes

- The HistoryServer process is down or not responding.

- The HistoryServer is not down but is not listening to the correct network port/address.

- The Nagios Server cannot connect to the HistoryServer.

3.5.5.6.2. Possible remedies

- Check for any errors in the HistoryServer logs (`/var/log/hadoop/mapred`) and restart the HistoryServer, if necessary.

- Use `ping` to check the network connection between the Nagios Server and the HistoryServer host.

3.5.6. HBase Service Alerts

These alerts are used to monitor the HBase service.

3.5.6.1. Percent RegionServers live

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 WARN alert and 30% to produce a CRITICAL alert. It uses the `check_aggregate` plugin to aggregate the results of `RegionServer process down` checks.

3.5.6.1.1. Potential causes

- 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 own because there were problems in the dependent services, ZooKeeper or HDFS

- GC paused the RegionServer for too long and the RegionServers lost contact with Zoo-keeper

3.5.6.1.2. Possible remedies

- 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

- Look at the configuration files (`/etc/hbase/conf`)

- If the failure was associated with a particular workload, try to understand the workload better

- Restart the RegionServers

3.5.6.2. HBase Master process

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. It uses the Nagios `check_tcp` plugin.

3.5.6.2.1. Potential causes

- The HBase master process is down

- The HBase master has shut itself down because there were problems in the dependent services, ZooKeeper or HDFS

- The Nagios server cannot connect to the HBase master through the network

3.5.6.2.2. Possible remedies

- 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`)

  Use `ping` to check the network connection between the Nagios server and the HBase master

- Restart the master

3.5.6.3. HBase Master Web UI

This host-level alert is triggered if the HBase Master Web UI is unreachable.

3.5.6.3.1. Potential causes

- The HBase Master Web UI is unreachable from the Nagios Server.

- The HBase Master process is not running.

3.5.6.3.2. Possible remedies

- Check if the Master process is running.

- Check whether the Nagios Server can ping the HBase Master server.

- Using a browser, check whether the Nagios Server can reach the HBase Master Web UI.

3.5.6.4. HBase Master CPU utilization

This host-level alert is triggered if the percent of CPU utilization on the master host exceeds the configured critical threshold. This alert uses the Nagios `check_snmp_load` plugin.

3.5.6.4.1. Potential causes

- 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.

- A down SNMP daemon on the master node, producing an unknown status.

3.5.6.4.2. Possible remedies

- Use the `top` command to determine which processes are consuming excess CPU

- Reset the offending process.

- Check the status of the SNMP daemon.

3.5.6.5. RegionServer process

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. It uses the Nagios `check_tcp` plugin.

3.5.6.5.1. Potential causes

- 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).

- The Nagios server cannot connect to the RegionServer through the network.

3.5.6.5.2. Possible remedies

- 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.

- Use `ping` to check the network connection between the Nagios Server and the RegionServer.

3.5.7. Hive Alerts

These alerts are used to monitor the Hive service.

3.5.7.1. Hive-Metastore status

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. It uses the Nagios `check_tcp` plugin.

3.5.7.1.1. Potential causes

- 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.

3.5.7.1.2. Possible remedies

- Using **Ambari Web**, stop the Hive service and then restart it.

- Use `ping` to check the network connection between the Nagios server and the Hive Metastore server.

3.5.8. WebHCat Alerts

These alerts are used to monitor the WebHCat service.

3.5.8.1. WebHCat Server status

This host-level alert is triggered if the WebHCat server cannot be determined to be up and responding to client requests.

3.5.8.1.1. Potential causes

- The WebHCat server is down.

- The WebHCat server is hung and not responding.

- The WebHCat server is not reachable over the network.

3.5.8.1.2. Possible remedies

- Restart the WebHCat server using **Ambari Web**.

3.5.9. Oozie Alerts

These alerts are used to monitor the Oozie service.

3.5.9.1. Oozie status

This host-level alert is triggered if the Oozie server cannot be determined to be up and responding to client requests.

3.5.9.1.1. Potential causes

- The Oozie server is down.

- The Oozie server is hung and not responding.

- The Oozie server is not reachable over the network.

3.5.9.1.2. Possible remedies

- Restart the Oozie service using **Ambari Web**.

3.5.10. Ganglia Alerts

These alerts are used to monitor the Ganglia service.

3.5.10.1. Ganglia Server status

This host-level alert determines if the Ganglia server is running and listening on the network port. It uses the Nagios `check_tcp` plugin.

3.5.10.1.1. Potential causes

- The Ganglia server process is down.

- The Ganglia server process is hanging.

- The network connection is down between the Nagios and Ganglia servers

3.5.10.1.2. Possible remedies

- Check the Ganglia server (`gmetad`) related log in `/var/log/messages` for any errors.

- Restart the Ganglia server.

- Check if `ping` works between Nagios and Ganglia servers.

3.5.10.2. Ganglia Monitor process

These host-level alerts check if the Ganglia monitor daemons (`gmond`) on the Ganglia server are running and listening on the network port. This alert uses the Nagios `check_tcp` plugin.

Ganglia Monitoring daemons run for the following collections:

- Slaves

- NameNode

- HBase Master

- JobTracker (Hadoop 1 only)

- ResourceManager (Hadoop 2 only)

- HistoryServer (Hadoop 2 only)

3.5.10.2.1. Potential causes

- A `gmond` process is down.

- A `gmond` process is hanging.

- The network connection is down between the Nagois and Ganglia servers.

3.5.10.2.2. Possible remedies

- Check the `gmond` related log in `/var/log/messages` for any errors.

- Check if `ping` works between Nagios and Ganglia servers.

3.5.11. Nagios Alerts

These alerts are used to monitor the Nagios service.

3.5.11.1. Nagios status log freshness

This host-level alert determines if the Nagios server is updating its status log regularly. Ambari depends on the status log (`/var/nagios/status.dat`) to receive all the Nagios alerts.

3.5.11.1.1. Potential causes

- The Nagios server is hanging and thus not scheduling new alerts

- The file `/var/nagios/status.dat` does not have appropriate write permissions for the Nagios user.

3.5.11.1.2. Possible remedies

- Restart the Nagios server

- Check the permissions on `/var/nagios/status.dat`.

- Check `/var/log/messages` for any related errors.

3.5.12. ZooKeeper Alerts

These alerts are used to monitor the Zookeeper service.

3.5.12.1. Percent ZooKeeper servers live

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 uses the `check_aggregate` plugin to aggregate the results of `Zookeeper process` checks.

3.5.12.1.1. Potential causes

- The majority of your ZooKeeper servers are down and not responding.

3.5.12.1.2. Possible remedies

- Check the dependent services to make sure they are operating correctly.

- Check at the ZooKeeper logs files (`/var/log/hadoop/zookeeper.log`) for further information.

- If the failure was associated with a particular workload, try to understand the workload better.

- Restart the ZooKeeper servers from the Ambari UI.

3.5.12.2. Zookeeper Server process

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. It uses the Nagios `check_tcp` plugin.

3.5.12.2.1. Potential causes

- 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).

- The Nagios server cannot connect to the ZooKeeper server through the network.

3.5.12.2.2. Possible remedies

- Check for any errors in the ZooKeeper logs (`/var/log/hbase/`) and restart the ZooKeeper process using **Ambari Web**.

- Run the `netstat-tuplpn` command to check if the ZooKeeper server process is bound to the correct network port.

- Use `ping` to check the network connection between the Nagios server and the ZooKeeper server.

3.5.13. Ambari Alerts

This alert is used to monitor the Ambari Agent service.

3.5.13.1. Ambari Agent process.

This host-level alert is triggered if the Ambari Agent process cannot be confirmed to be up and listening on the network for the configured critical threshold, given in seconds. It uses the Nagios `check_tcp` plugin.

3.5.13.1.1. Potential causes

- The Ambari Agent process is down on the host.

- The Ambari Agent process is up and running but heartbeating to the Ambari Server.

- The Ambari Agent process is up and running but is unreachable through the network from the Nagios Server.

- The Ambari Agent cannot connect to the Ambari Server through the network.

3.5.13.1.2. Possible remedies

- Check for any errors in the logs (`/var/log/ambari-agent/ambari-agent.log`) and restart the Ambari Agent process.

- Use `ping` to check the network connection between the Ambari Agent host and the Ambari Servers.