Hortonworks Data Platform

Ambari User's Guide

(Apr 13, 2015)

docs.hortonworks.com

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.



Except where otherwise noted, this document is licensed under Creative Commons Attribution ShareAlike 3.0 License. http://creativecommons.org/licenses/by-sa/3.0/legalcode

Table of Contents

1. Overview: Ambari User's Guide	. 1
1.1. Architecture	. 1
1.1.1. Sessions	. 1
1.2. Accessing Ambari Web	. 2
2. Monitoring and Managing your HDP Cluster with Ambari	3
2.1. Viewing Metrics on the Dashboard	3
2.2. Scanning System Metrics	3
2.3. Drilling Into Metrics for a Service	4
2.4. Viewing Cluster-Wide Metrics	. 5
2.5. Adding a Widget to the Dashboard	6
2.6. Resetting the Dashboard	. 6
2.7. Customizing Metrics Display	. 6
2.8. Viewing More Metrics for your HDP Stack	. 7
2.9. Viewing Heatmaps	. 8
2.10. Scanning Status	. 9
3. Managing Hosts	10
3.1. Working with Hosts	10
3.2. Determining Host Status	
3.3. Filtering the Hosts List	11
3.4. Performing Host-Level Actions	11
3.5. Viewing Components on a Host	
3.6. Decommissioning Masters and Slaves	13
3.6.1. How to Decommission a Component	13
3.7. How to Delete a Component	14
3.8. Deleting a Host from a Cluster	14
3.8.1. How to Delete a Host from a Cluster	15
3.9. Setting Maintenance Mode	15
3.9.1. Setting Maintenance Mode for Services, Components, and Hosts	15
3.9.2. How to Turn On Maintenance Mode for a Service	16
3.9.3. How to Turn On Maintenance Mode for a Host	16
3.9.4. How to Turn On Maintenance Mode for a Host (alternative using	
filtering for hosts)	16
3.9.5. Maintenance Mode Use Cases	17
3.10. Adding Hosts to a Cluster	18
4. Managing Services	19
4.1. Starting and Stopping All Services	19
4.2. Selecting a Service	
4.3. Adding a Service to your Hadoop cluster	20
4.4. Editing Service Config Properties	24
4.5. Viewing Summary, Alert, and Health Information	24
4.5.1. Alerts and Health Checks	
4.5.2. Analyzing Service Metrics	25
4.6. Performing Service Actions	
4.7. Monitoring Background Operations	
4.8. Using Quick Links	
4.9. Rolling Restarts	
4.9.1. Setting Rolling Restart Parameters	
4.9.2. Aborting a Rolling Restart	

4.10. Refreshing YARN Capacity Scheduler	28
4.10.1. How to refresh the YARN Capacity Scheduler	28
4.11. Rebalancing HDFS	29
4.11.1. How to rebalance HDFS	29
5. Managing Service High Availability	
5.1. NameNode High Availability	30
5.1.1. How To Configure NameNode High Availability	
5.1.2. How to Roll Back NameNode HA	35
5.2. ResourceManager High Availability	43
5.3. HBase High Availability	
5.3.1. Adding an HBase Master Component	44
5.4. Hive High Availability	
5.4.1. Adding a Hive Metastore Component	45
5.4.2. Adding a HiveServer2 Component	45
5.5. Oozie High Availability	45
5.5.1. Adding an Oozie Server Component	46
6. Managing Configurations	47
6.1. Configuring Services	47
6.1.1. Updating Service Properties	47
6.1.2. Restarting components	47
6.2. Using Host Config Groups	47
6.3. Customizing Log Settings	49
6.4. Downloading Client Configs	50
6.5. Service Configuration Versions	50
6.5.1. Basic Concepts	50
6.5.2. Terminology	51
6.5.3. Saving a Change	51
6.5.4. Viewing History	
6.5.5. Comparing Versions	53
6.5.6. Reverting a Change	54
6.5.7. Versioning and Host Config Groups	54
7. Administering the Cluster	56
7.1. Managing Stack and Versions	56
7.2. Register a Version	
7.3. Install the Version	
7.4. Perform Upgrade	57
7.4.1. Upgrade Prerequisites	58
7.5. Service Accounts	59
7.6. Kerberos	
7.6.1. How To Regenerate Keytabs	59
7.6.2. How To Disable Kerberos	60
8. Monitoring and Alerts	61
8.1. Managing Alerts	
8.1.1. Terms and Definitions	
8.1.2. Alert Definitions and Instances	
8.1.3. How To Change an Alert	
8.1.4. How To View a List of Alert Instances	
8.1.5. How To Enable or Disable an Alert	
8.2. Configuring Notifications	
8.3. List of Predefined Alerts	
8.3.1. HDFS Service Alerts	64

8.3.2. NameNode HA Alerts
8.3.3. YARN Alerts
8.3.4. MapReduce2 Alerts
8.3.5. HBase Service Alerts
8.3.6. Hive Alerts
8.3.7. Oozie Alerts
8.3.8. ZooKeeper Alerts
8.3.9. Ambari Alerts 69

1. Overview: Ambari User's Guide

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 manage the complexity, Apache Ambari collects a wide range of information from the cluster's nodes and services and presents it 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. You use Ambari Web to create and manage your HDP cluster and to perform basic operational tasks such as starting and stopping services, adding hosts to your cluster, and updating service configurations. You also use Ambari Web to perform administrative tasks for your cluster, such as managing users and groups and deploying Ambari Views.

For more information on administering Ambari users, groups and views, refer to the Ambari Administration Guide.

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.

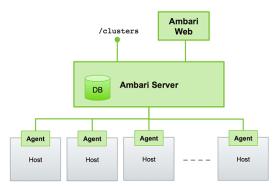


Figure - Ambari Server Architecture

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. Communication between the browser and server occurs asynchronously via the REST API.

Ambari Web sessions do not time out. The Ambari Server application constantly accesses the Ambari REST API, which resets the session timeout. During any period of Ambari Web inactivity, the Ambari Web user interface (UI) refreshes automatically. You must explicitly sign out of the Ambari Web UI to destroy the Ambari session with the server.

🝌 Ambari	My20Cluster 10 ops 0 alorts	Dashboard	Services	Hosts	Alerts	Admin		🛓 admin 👻
							-	About
Your Views								Manage Ambari
No views								Settings
								Sign out

1.2. Accessing Ambari Web

Typically, you start the Ambari Server and Ambari Web as part of the installation process. If Ambari Server is stopped, you can start it using a command line editor on the Ambari Server host machine. Enter the following command:

ambari-server start

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

http://<your.ambari.server>:8080

Enter your user name 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 Manage Ambari option.

🚕 Ambari My20Clu	ISter 0 ops 0 alerts	Dashboard Services Hosts Alerts A	Admin 🌐 🏝 admin 👻
• HDFS	Summary Configs	Quick Links	About Seri Manage Ambari
MapReduce2 YARN	Summary	No alerts Metrics	Settings L٤ Sign out

For more information about managing users and other administrative tasks, see Administering Ambari.

2. Monitoring and Managing your HDP Cluster with Ambari

This topic describes how to use Ambari Web features to monitor and manage your HDP cluster. To navigate, select one of the following feature tabs located at the top of the Ambari main window. The selected tab appears white.

🚕 Ambari	My20Cluster 10 ops 0 alorts	Dashboard	Services	Hosts	Alerts	Admin	 🛓 admin 🗸

- Viewing Metrics on the Dashboard
- Managing Services
- Managing Hosts
- Managing Service High Availabilty
- Managing Configurations
- Administering the Cluster

2.1. Viewing Metrics on the Dashboard

Ambari Web displays the **Dashboard** page as the home page. Use the **Dashboard** to view the operating status of your cluster in the following three ways:

- Scanning System Metrics
- Scanning Status
- Viewing Heatmaps

2.2. Scanning System Metrics

View **Metrics** that indicate the operating status of your cluster on the Ambari Dashboard. Each metrics widget displays status information for a single service in your HDP cluster. The Ambari Dashboard displays all metrics for the HDFS, YARN, HBase, and Storm services, and cluster-wide metrics by default.



Note

Metrics data for Storm is buffered and sent as a batch to Ambari every five minutes. After adding the Storm service, anticipate a five-minute delay for Storm metrics to appear.

You can add and remove individual widgets, and rearrange the dashboard by dragging and dropping each widget to a new location in the dashboard.

 YARN Tez Nagios Ganglia Hive 	HDFS Disk Usage	DataNodes Live	HDFS Links NameNode Secondary NameNode 1 DataNodes More	Memory Usage	Network Usage
 HBase Pig Sqoop Oozie ZooKeeper Falcon 	CPU Usage	Cluster Load	NameNode Heap	NameNode RPC 0.50 ms	NameNode CPU WIO
 Storm Flume Silder Knox Kafka 	NameNode Uptime 3.5 hr	HBase Master Heap	HBase Links HBase Master 1 RegionServers Master Web UI More	HBase Ave Load	HBase Master Uptime 3.5 hr
Actions -	ResourceManager Heap	ResourceManager Uptime 3.5 hr	NodeManagers Live 1/1	YARN Memory	Supervisors Live

Status information appears as simple pie and bar charts, more complex charts showing usage and load, sets of links to additional data sources, and values for operating parameters such as uptime and average RPC queue wait times. Most widgets display a single fact by default. For example, HDFS Disk Usage displays a load chart and a percentage figure. The Ambari Dashboard includes metrics for the following services:

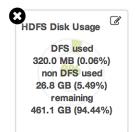
Metric:	Description:
HDFS	
HDFS Disk Usage	The Percentage of DFS used, which is a combination of DFS and non-DFS used.
Data Nodes Live	The number of DataNodes live, as reported from the NameNode.
NameNode Heap	The percentage of NameNode JVM Heap used.
NameNode RPC	The average RPC queue latency.
NameNode CPU WIO	The percentage of CPU Wait I/O.
NameNode Uptime	The NameNode uptime calculation.
YARN (HDP 2.1 or later Stacks)	
ResourceManager Heap	The percentage of ResourceManager JVM Heap used.
ResourceManager Uptime	The ResourceManager uptime calculation.
NodeManagers Live	The number of DataNodes live, as reported from the ResourceManager.
YARN Memory	The percentage of available YARN memory (used vs. total available).
HBase	
HBase Master Heap	The percentage of NameNode JVM Heap used.
HBase Ave Load	The average load on the HBase server.
HBase Master Uptime	The HBase Master uptime calculation.
Region in Transition	The number of HBase regions in transition.
Storm (HDP 2.1 or later Stacks)	
Supervisors Live	The number of Supervisors live, as reported from the Nimbus server.

Ambari Service Metrics and Descriptions

2.3. Drilling Into Metrics for a Service

• To see more detailed information about a service, hover your cursor over a Metrics widget.

More detailed information about the service displays, as shown in the following example:



- To remove a widget from the mashup, click the white X.
- To edit the display of information in a widget, click the pencil icon. For more information about editing a widget, see Customizing Metrics Display.

2.4. Viewing Cluster-Wide Metrics

Cluster-wide metrics display information that represents your whole cluster. The Ambari Dashboard shows the following cluster-wide metrics:

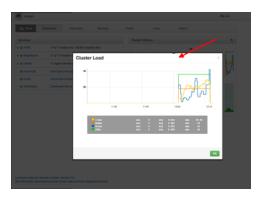
Metrics - Heatmap	os Config History			
Memory Usage	Network Usage	CPU Usage	Cluster Load	Region In Transition

Ambari Cluster-Wide Metrics and Descriptions

Metric:	Description:
Memory Usage	The cluster-wide memory utilization, including memory cached, swapped, used, 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.

- To remove a widget from the dashboard, click the white X.
- Hover your cursor over each cluster-wide metric to magnify the chart or itemize the widget display.
- To remove or add metric items from each cluster-wide metric widget, select the item on the widget legend.
- To see a larger view of the chart, select the magnifying glass icon.

Ambari displays a larger version of the widget in a pop-out window, as shown in the following example:



Use the pop-up window in the same ways that you use cluster-wide metric widgets on the dashboard.

To close the widget pop-up window, choose OK.

2.5. Adding a Widget to the Dashboard

To replace a widget that has been removed from the dashboard:

1. Select the Metrics drop-down, as shown in the following example:



- 2. Choose Add.
- 3. Select a metric, such as Region in Transition.
- 4. Choose Apply.

2.6. Resetting the Dashboard

To reset all widgets on the dashboard to display default settings:

1. Select the Metrics drop-down, as shown in the following example:

Metrics -	Heatmaps	Config I	History
	+ Add	- + I	
HDFS Dis	🗘 Edit	•	$\boldsymbol{\mathcal{Z}}~$ Reset all widgets to default
			C View metrics in Ganglia

- 2. Choose Edit.
- 3. Choose Reset all widgets to default.

2.7. Customizing Metrics Display

To customize the way a service widget displays metrics information:

- 1. Hover your cursor over a service widget.
- 2. Select the pencil-shaped, edit icon that appears in the upper-right corner.

The Customize Widget pop-up window displays properties that you can edit, as shown in the following example.

•	Custo	omize Wi	dget					×
	Edit the	a percentage thr	esholds to change	the color o	f current pie chart.	Enter two numbe	rs between 0 to 100	
	0	85			95		100	
							Cancel	Apply
<u>``</u>		1 DataNodes						ab.

3. Follow the instructions in the Customize Widget pop-up 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, choose Apply.
- 5. To close the editor without saving any changes, choose Cancel.



Not all widgets support editing.

2.8. Viewing More Metrics for your HDP Stack

The HDFS Links and HBase Links widgets list HDP components for which links to more metrics information, such as thread stacks, logs and native component UIs are available. For example, you can link to NameNode, Secondary NameNode, and DataNode components for HDFS, using the links shown in the following example:

Metrics -	Heatmaps				
HDFS Links					
Name Secondary 1 Data	NameNode				
More					
	NameNode UI NameNode logs NameNode JMX Thread Stacks				

Choose the More drop-down to select from the list of links available for each service. The Ambari Dashboard includes additional links to metrics for the following services:

Links to More Metrics for HDP Services

Service:	Metric:	Description:
HDFS		
	NameNode UI	Links to the NameNode UI.
	NameNode Logs	Links to the NameNode logs.
	NameNode JMX	Links to the NameNode JMX servlet.
	Thread Stacks	Links to the NameNode thread stack traces.
HBase		
	HBase Master UI	Links to the HBase Master UI.
	HBase Logs	Links to the HBase logs.
	ZooKeeper Info	Links to ZooKeeper information.
	HBase Master JMX	Links to the HBase Master JMX servlet.
	Debug Dump	Links to debug information.
	Thread Stacks	Links to the HBase Master thread stack traces.

2.9. Viewing Heatmaps

Heatmaps provides a graphical representation of your overall cluster utilization using simple color coding.



A colored block represents each host in your cluster. To see more information about a specific host, hover over the block representing the host in which you are interested. A popup window displays metrics about HDP components installed on that host. Colors displayed in the block represent usage in a unit appropriate for the selected set of metrics. If any data necessary to determine state is not available, the block displays "Invalid Data". Changing the default maximum values for the heatmap lets you fine tune the representation. Use the Select Metric drop-down to select the metric type.



Heatmaps supports the following metrics:

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

Metric	Uses
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

2.10. Scanning Status

Notice the color of the dot appearing next to each component name in a list of components, services or hosts. The dot color and blinking action indicates operating status of each component, service, or host. For example, in the Summary View, notice green dot next to each service name. The following colors and actions indicate service status:

Status Indicators

Color	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.

3. Managing Hosts

Use Ambari Hosts to manage multiple HDP 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 supports filtering your selection of host components, based on operating status, host health, and defined host groupings.

3.1. Working with Hosts

Use Hosts to view hosts in your cluster on which Hadoop services run. Use options on **Actions** 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.

Actions • Filter: All (3) •							
Name * Any	IP Address 0 Acy	Cores (CPU)	RAM 0 Ary	Disk Usage	Load Avg 0 Any	Versions Filter T	Components Filter Y
c6401.ambari.apache.org	10.0.2.15	1 (1)	1.83GB	1	0.04	2.2.1.1-20 (Current)	+ 7 Components
o6402.ambari.apacha.org	10.0.2.15	1 (1)	1.83GB	1	0.10	2.2.1.1-20 (Current)	+ 22 Components
c6403.ambari.apache.org	10.0.2.15	1 (1)	1.83GB		0.85	2.2.1.1-20 (Current)	21 Components

3.2. Determining Host Status

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

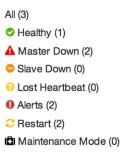
- Red At least one master component on that host is down. Hover to see a tooltip that lists affected components.
- Orange At least one slave component on that host is down. Hover to see a tooltip that lists affected components.
- Yellow Ambari Server has not received a heartbeat from that host for more 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.

Name ≑	
Any	
c6401.ambari.apache.org	C
c6402.ambari.apache.org	6
c6403.ambari.apache.org	1

3.3. Filtering the Hosts List

Use Filters 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 to apply specific search and sort criteria that limits the list of hosts appearing on the Hosts page.

3.4. Performing Host-Level Actions

Use Actions 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 option 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.

Actions - Filter:	All (3) -		
+ Add New Hosts			
Selected Hosts (1)	Hosts	•	
Filtered Hosts (3) →	DataNodes	•	Start
All Hosts (3)	NodeManagers	•	Stop
	RegionServers	->	Restart
	Supervisors	→ [Decommission
			Recommission

- 3. Choose OK to confirm starting the selected operation.
- 4. Optionally, use Monitoring Background Operations to follow, diagnose or troubleshoot the restart operation.

3.5. Viewing Components on a Host

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

Ambari My20Cluster 🛛 ops	0 alorts	Dashboard Services	Hosts Alerts	Admin 🔡	🛔 admin 👻
c6403.ambari.apache.org ← Back					
Summary Configs Alerts	Versions			Ho	st Actions -
Components	+ Add	Host Metrics			
Metric Collector / AMS	Started •	100%	F		
ZooKeeper Server / ZooKeeper	Started -		3	372.5 GB	
DataNode / HDFS	Started -	50%		186.2 GB	
S Flume / Flume	Started -				
RegionServer / HBase	Started -	CPU Usage		Disk Usag	le
Metric Monitor / AMS	Started -			1.8 GB	
NodeManager / YARN	Started -	1			
Supervisor / Storm	Started -		W	953.6 MB	
Clients / Falcon Client, HBase Client, HCat Client, HDFS Client, Hive Client, MapReduce2 Client, Oozle Client,	Installed -	Load	¥ 1	Memory Us	age
Pig, Silder, Sqoop, Tez Client, YARN Client, ZooKeeper Client		195.3 KB	4	50	
Summary		Network Usag	le .	Processe	s
Hostname: c6403.ambari.apach IP Address: 10.2.2.15 OS: centos6 (x86_64) Cores (CPU): 1(1) Disk: 30.9463/489.68GB (Memory: 1.83GB Load Avg: 0.78 Heartbaat: amoment ago Current Version: 2.2.1.1-20					

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 drop-down menu next to an individual component on a host. The drop-down menu shows current operation status for each component, For example, you can decommission, restart, or stop the DataNode component (started) for HDFS, by selecting one of the options shown in the following example:

Components		+ Add		
Metric Collector / AMS	Starte	d 🗸		
ZooKeeper Server / ZooKee	per Starte	d 🔻		
DataNode / HDFS	Starte	Started -		
Flume / Flume	Decommission			
RegionServer / HBase	Restart Stop			
Metric Monitor / AMS	Turn On Maintenan	ce Mode		
NodeManager / YARN	Delete	a •		
Supervisor / Storm	Starte	d ᠇		
Clients / Falcon Client, HBase Client, HCat Client, HDFS Client, Hive Client, MapReduce2 Client, MapReduce2 Client, Occie Client, Pig, Silder, Sqoop, Tez Client, YARN Client, ZooKeeper Client	Installe	ed 🔻		

3.6. Decommissioning Masters and Slaves

Decommissioning is a process that supports removing a component from the cluster. You must decommission a master or slave running on a host before removing the component or host from service. Decommissioning helps prevent potential loss of data or service disruption. 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.

3.6.1. How to Decommission a Component

To decommission a component using Ambari Web, browse **Hosts** to find the host FQDN on which the component resides.

Using Actions, select HostsComponent Type, then choose Decommission.

For example:

Add New Hosts				
Selected Hosts (1)	Þ	Hosts	•	
Filtered Hosts (3)	•	DataNodes	•	Start
All Hosts (3)	٠ſ	NodeManagers	•	Stop
	-	RegionServers	•	Restart
		Supervisors	->	Decommission
			- 5	Recommission

The UI shows "Decommissioning" status while steps process, then "Decommissioned" when complete.



3.7. How to Delete a Component

To delete a component using Ambari Web, on ${\tt Hosts}$ choose the host FQDN on which the component resides.

- 1. In Components, find a decommissioned component.
- 2. Stop the component, if necessary.



Note

A decommissioned slave component may restart in the decommissioned state.

3. For a decommissioned component, choose **Delete** from the component drop-down menu.



Note

Restarting services enables Ambari to recognize and monitor the correct number of components.

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.

3.8. 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.

3.8.1. 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:

Unable to Delete Host	[
▲ This host cannot be deleted since the following components are running: DataNode, Ganglia Monitor, NodeManager, ZooKeeper Server To delete this host, you must first stop all the running components listed above. If this host has a DataNode, it should be decommissioned first to prevent data loss.	
OK	

3.9. Setting 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.

3.9.1. 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.

3.9.2. 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.

3.9.3. 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.

3.9.4. 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.

	Name :	IP Address	Cores (CPU)	RAM :	Disk Usage	Load Avg	Components
5	Any	6 components in Maintenance Mode	Any	Any		Any	Filter T
1¢	c6401.ambari.apache.org	0 10.0.2.15	1 (1)	1.83GB	1	0.14	+ 6 Components
•	o6402.ambari.apache.org	10.0.2.15	1 (1)	1.83GB	1	0.11	+ 27 Components
¢	c6403.ambari.apache.org	C 10.0.2.15	1 (1)	1.83GB	1	0.02	+ 17 Components
3	of 3 hosts showing - clear 1	liters 1 host i	elected - clear s	election	Sh	ow: 10 0	1-3 of 3 + -

- 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; 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.

3.9.5. 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.

3.10. Adding Hosts to a Cluster

To add new hosts to your cluster, 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 Install Wizard. Follow the prompts, providing information similar to that provided to define the first set of hosts in your cluster.

DD HOST WIZARD	Install Options
Confirm Hosts	Enter the list of hosts to be included in the cluster and provide your SSH key.
ssign Slaves and Sients	Target Hosts
Configurations	Enter a list of hosts using the Fully Qualified Domain Name (FQDN), one per line. Or use Pattern Expressions
nstall, Start and Test Summary	o6405.ambari.apache.org
	Host Registration Information
	* Provide your SSH Private Key to automatically register hosts Chemitiki No file chosen
	MITEORIEANCOGENERALING/P200KTkyztvp5e6666A
	SSH user (root or passwordless sudo account) vagrant © Perform manual registration on hosts and do not use SSH

4. 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.

🚕 Ambari My20Cl	uster • 0 ops 0 alerts	Dashboard	Services Hosts Alerts	Admin 💷 🛓 admin 🗸
· HDFS	Summary Configs	Quick Links -		Service Actions -
MapReduce2 YARN	Summary	No alerts	Metrics	Last 1 hour -
Tor: Tor: Tor: Hive Hase Hose Code Code Code Code Code Socon Storn Flume	NameNode Uptime 8.9 NameNode Heap 205 Disk Usage (DFS Used) 575 Disk Usage (Non DFS Used) 29. Disk Usage (Remaining) 456 Blocks (tota) 404	Started V C dead / 0 decommissioning V C dead / 0 decommissioning ST MSI (964 A MB (20.3% used) ST MSI (962 C MB (20.3%) A GB / 488.2 C B (50.2%) 3.3 CB / 488.2 CB (50.2%) 3.3 CB / 488.2 CB (50.3%) 4.3 CB / 488.2 CB (50.3%) 4.3 CB / 488.2 CB (50.3%)	372.5 08 186.3 08 Total Space Utilization 400 200	18 epsile File Operations
C Knox	Upgrade Status No Safe Mode Status Not	pending upgrade	Block Status	HDFS VO
Kafka AMS Actions -			t ind C.5 ms PPC	20 ms 10 ms Garbage Collection
			953.6 MB 476.6 MB JVM Memory Status	100 50 JVM Threed Status

Services supports the following tasks:

- Starting and Stopping All Services
- Selecting a Service
- Adding a Service to your Hadoop cluster
- Editing Service Config Properties
- Viewing Summary, Alert, and Health Information
- Performing Service Actions
- Monitoring Background Operations
- Using Quick Links
- Rolling Restarts
- Refreshing YARN Capacity Scheduler
- Rebalancing HDFS

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, as shown in the following example:

Actions -
+ Add Service
Start All
Stop All

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.

4.3. Adding a Service to your Hadoop cluster

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 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 2.2 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.

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 Service wizard displays installed services highlighted green and check-marked, not available for selection.

Add Service Wizard

Choose Services			
Assian Masters	Choose which services	you want to in	stal on your cluster.
Assign Slaves and Clients			
Customize Services	Bervice	Version	Description
Configure Identities	2 HDFB	2.6.0.2.2	Apache Hadoop Distributed File System
Basiew	VARN + MapReduce2	2.6.0.2.2	Apache Hadoop NextGen MapReduce (YARN)
Install, Start and Test Summary	⊡ Tez	0.5.2.2.2	Tez is the next generation Hadoop Query Processing framework written on top of YARN.
	i⊴ Hive	0.14.0.2.2	Data warehouse system for ad-hoc queries & analysis of large datasets and table & storage management service
	HBase	0.98.4.2.2	Non-mistional distributed database and centralized service for configuration management & synchronization
	i≌ Pig	0.14.0.2.2	Scripting platform for analyzing large datasets
	i ⊗ Sqoop	1.4.5.2.2	Tool for transferring bulk data between Apache Hadoop and structured data stores such as relational databases
	I Oozie	4.1.0.2.2	System for workflow coordination and execution of Apache Hadoop jobs. Th also includes the installation of the optional Ocale Web Console which releas on and will install the ExtLS Library.
	2 ZooKeeper	3.4.6.2.2	Centralized service which provides highly reliable distributed coordination
	Falcon	0.6.0.2.2	Data management and processing platform
	Storm	0.9.3.2.2	Apache Hadoop Stream processing framework
	Pume	1.5.2.2.2	A distributed service for collecting, aggregating, and moving large amounts or streaming data into HDPS
	 Ambari Metrics 	0.1.0	A system for metrics collection that provides storage and retrieval capability for metrics collected from the cluster
	🗆 Kafka	0.8.1.2.2	A high-throughput distributed messaging system
	🖂 Клаж	0.5.0.2.2	Provides a single point of authentication and access for Apache Hadoop services in a cluster
	Renger	0.4.0	Comprehensive security for Hadoop
	⊴ Silder	0.60.0.2.2	A framework for deploying, managing and monitoring existing distributed applications on YAPN.
	M Spark	12.02.2	Apache Spark is a fast and general engine for large-scale data processing.



Note

Ambari 2.0 supports adding Ranger and Spark services, using the Add Services Wizard.

Ranger	0.4.0	Comprehensive security for Hadoop
Slider	0.60.0.2.2	A framework for deploying, managing and monitoring existing distributed applications on YARN.
Spark	1.2.0.2.2	Apache Spark is a fast and general engine for large-scale data processing.
🗌 Kafka	0.8.1.2.2	A high-throughput distributed messaging system
✓ Ambari Metrics	0.1.0	A system for metrics collection that provides storage and retrieval capability for metrics collected from the cluster

For more information about installing Ranger, see Installing Ranger.

For more information about Installing Spark, see Installing Spark.

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 shows a grey check mark.

Using the drop-down, choose an alternate host name, if necessary.

- 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.

ID SERVICE WIZARD Droose Services	Assign Ma	asters	
Issign Masters		penents to hosts you want to run them on. Metastore, and WebHCat Server will be ho	sted on the same server.
Dustomize Services	NameNode:	c6401.amberl.apache.org (1.8 🔅	o6401.ambari.apache.org (1.8 GB, 1 cores)
nstall, Start and Test	SNameNode:	c6402 ambari apache.org (1.8 🔅	NameNode HDase Master
ummary	History Server:	c6402.ambarl.epache.org (1.8 🕴	of402 amber, apache.org (1,8 GB, 1 cores)
	ResourceManager:	c6402.emberl.epache.org (1.8 2	Shamehode History Server
	App Timeline Server:	c6402.emberl.epeche.org (1.8	ResourceManager App Timetine Server Nagios Server
	Nagios Server:	c6402.emberl.epeche.org (1.8 2	Gangla Server HiveServer2 Hive Metastore WebHCat Server
	Ganglia Server:	c6402 ambari apache org (1.8 🔅	Occie Server Zookeeper
	HveServer2:	c6402 ambari apache.org (1.8 🔅	Falcon Server
	Hive Metastore:	c6402.ambari.apache.org #	c6403.ambari.apache.org (1.8 GB. 1 cores)
	WebHCat Server:	c6402.ambari.apache.org #	
	HBase Master:	c6401.emberl.epeche.org (1.8 2	
	Oozie Server:	c6402.emberl.epeche.org (1.8 2	
	ZooKeeper:	c6401.emberl.epeche.org (1.8 2	
	ZooKeeper:	c6402.emberl.apache.org (1.8	
	ZooKeeper:	o5403.emberl.apache.org (1.8	
	Falcon Server:	c6402.ambari.apache.org (1.8	

3. In Assign Slaves and Clients, accept the default assignment of slave and client components to hosts. Then, choose Next.

Alternatively, select hosts on which you want to install slave and client components. You must select at least one host for the slave of each service being added.

Host Roles Required for Added Services

Service Added	Host Role Required
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.

ssign Masters ssign Slaves and Olents ustomize Services	Assign slave and client com Hosts that are assigned mas "Client" will install HDFS Clie Client, Pig. Sqoop, Oozle Cli	ter components a nt, MapReduce 2	re shown with Olient, YARN Client,	Tez Clent, Hive Cler	r, HCat Clent, HB	850
eview stall. Start and Test	Host	al none	all none	all none	all none	all none
ummary	c6401.ambari.apache.org •	DataNode	NodeManager	BegionServer	Supervisor	8 Olerc
	c6402.ambari.apache.org •	DataNode	NodeManager	B RegionServer	Supervisor	8 Olerc
	c6403.ambarl.apache.org *	⊗ DataNode	2 NodeManager	RegionServer	8 Supervisor	R Olert
				Show: 25 0	1-3 of 3 H	

4. In Customize Services, accept the default configuration properties.

Alternatively, edit the default values for configuration properties, if necessary. Choose Override to create a configuration group for this service. Then, choose Next.

Droose Services Assign Masters	Customize S	ervices		
Assign Slaves and Clients	We have come up with ref.	commended configurations for the services y	ou selected. Customia	te them as you see
Noview Install, Start and Test Summary	HOFS YARN Mapfied Falcon Tez Miso	uce 2 Hive WebHCat HBase ZooKi	neper Oczie Nagio	s Ganglia Pig
	Group Falcon Default (Manage Config Groups	Fiter	•
	* Falcon Server			
	Falcon Server	c6402.ambari.apache.org		
	Falcon server port	15000		O Override
	Falcon data directory	/hadoop/falcon		O Override
	Falcon store URI	fiellihadoopifalcon/store		O Override
	Falcon - Occie Integri	ition		
	• Falcon startup proper	tes		
	Falcon runtime.prope	198		
	Advanced			
	Oustom startup.propr	rtes		
	Oustom runtime.prop	ertes		

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

D SERVICE WIZARD	Review
ssign Masters	Please review the configuration before installation
ssign Slaves and Clients	
ustomize Services	Admin Name : admin
stall, Start and Test	Cluster Name : mycluster
mary	Total Hosts : 3 (D new)
	Repositories:
	PHE, S/Cent06 S/Disole Linux 5 : http://s3.amazonaws.com/dev.hortonworks.com/HQP/centos5/2.x/updates/2.1.1.0
	PHE. 6/Cent06 6/Oracle Linux 6 : http://s3.amazonaws.com/dev.hortonworks.com/HOP/centos6/2.x/updates/2.1.1.0
	SLES 11/SUSE 11 : http://s3.amazonaws.com/dev.hortonworks.com/HOP/suse11/2.x/updates/2.1.1.0
	Services
	Paloon Server : 06402.ambari.apache.org
	- Back Desky

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

Masters	Please wait while the selector	d services are installed and started.		
Slaves and Clients				100 % overall
ize Services				
2		Show: A1 (0)	n Progress (0) Warning (0) Su	ccess.CO Eal
Start and Test	Host	Status	Mossage	
ry .	e6401.ambari.apache.org	100%	Success	
	c6402.ambari.apache.org	100%	Success	
	c6403.ambari.apache.org	100%	Success	
		SN	Dev: 25 0 1-3 of 3	H 4 4

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

Choose Services	Summary
Assign Masters Assign Slaves and Clients	Important Restarting Nagios service is required for alerts and notifications to work property. After clocking on the Complete button to dismiss this wated, go to Services -> Nagios to restart the Nagios service.
Oustomize Services Review	Here is the summary of the install process.
Install, Start and Test Summary	The cluster consists of 3 hosts Installed and stands devices successfully on 3 new hosts Install and start completed in 2 minutes and 58 seconds

8. Restart any other components having stale configurations.

4.4. Editing Service Config Properties

Select a service, then select Configs 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. For example, select General to configure Default virtual memory for a job's map task.

Summary Configs		Quick Links+		Service Actions *
Group MapReduce2 De	fault (3) 🔹 Manage	Config Groups	Filter	•
VI 17 hours ap Commit	admin 30			
X VI Current ad	min authored on Man, D	Dec 29, 2014 21:28		Discard Save
 History Server 				
* General				
Default virtual memory for a job's map-task	688	MB = 0 0		
Default virtual memory for a job's reduce-task	682	MB		
Map-side sort buffer memory	273	MB a O		
Advanced mapred-erv	v			
Advanced mapred-site	o l			
Custom mapred-site				
Add Property				

4.5. Viewing Summary, Alert, and Health Information

After you select a service, the Summary tab displays basic information about the selected service.



Select one of the View Host links, as shown in the following example, to view components and the host on which the selected service is running.

 NameNode
 Started

 SNameNode
 Started

 DataNodes
 1/1 DataNodes Live

4.5.1. Alerts and Health Checks

On each Service page, in the Summary area, click Alerts to see a list of all health checks and their status for the selected service. Critical alerts are shown first. Click the text title of each alert message in the list to see the alert definition. For example, On the HBase > Services, click Alerts. Then, in Alerts for HBase, click HBase Master Process.

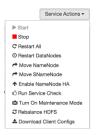
Alerts for HBase			
HBase Master Process Connection failed: [Errno 111] Connection refus,	CRIT	for 4 minutes	
Percent RegionServers Available affected: [0], total: [1]	ок	for 3 hours	
HBase RegionServer Process TCP OK - 0.000s response on port 60030	ок	for 3 hours	
RegionServers Health Summary <uriopen 111]="" [erron="" connection="" error="" refused=""></uriopen>	UNKWN	for 4 minutes	
		Cl	ose

4.5.2. Analyzing Service Metrics

Review visualizations in Metrics that chart common metrics for a selected service. Services > Summary displays metrics widgets for HDFS, HBase, Storm services. For more information about using metrics widgets, see Scanning System Metrics.

4.6. Performing Service Actions

Manage a selected service on your cluster by performing service actions. In Services, select the Service Actions drop-down menu, 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 generated by a 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 Setting Maintenance Mode.

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.

Operations	Start Time	Duration	Show: All (10)		٥
C Rolling Restart of DataNodes	Today 16:45	11.12 secs		35%	,
 Rolling Restart of NodeManagers - batch 2 of 2 	Today 10:06	38.45 secs		100%	•
 Rolling Restart of NodeManagers - batch 1 of 2 	Today 10:05	25.87 secs		100%	•

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.

Restart DataNo	des			х
Operations	Hosts	Show:	All (1)	\$
🗸 c6403.ambari.apache	o.org	100	1%	•
 Do not show this diale 	og again when starting a bao	ckground operation		ок

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

🗲 Tasks	Restart DataNode	🔁 Copy 🔀 Oper
stderr: /var/lib/ambari-	agent/data/errors-261.txt	
None		
stdout: /var/lib/ambari-	agent/data/output-261.txt	
<pre>BEXEC_DIR=/usr/lib/ hconfig /etc/had s'} 2014-02-27 21:58:01 .pid'] {'action': [2014-02-27 21:58:01 -datanode.pid'] 2014-02-27 21:58:01 fs', 'group': 'hado 2014-02-27 21:58:01</pre>	<pre>138 - Execute['ulimit - c unlimiter hodop/libexc && /usr/lib/hodop/s/ oop/conf stop datanode'] {'not_if'; /state:</pre>	<pre>bbin/hadoop-daemon.s None, 'user': 'hdf hadoop-hdfs-datanode bop/hdfs/hadoop-hdfs dfs'] {'owner': 'hd re} ''] {'owner': 'hdfs'</pre>

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

4.8. Using Quick Links

Select Quick Links options 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.



NameNode UI NameNode logs NameNode JMX Thread Stacks

4.9. 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, 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 Monitor Background Operations to monitor progress of rolling restarts.

4.9.1. 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 one component at a time, waits two 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.

Restart DataNodes	Х
This will restart a specified number of DataNodes at a time. Note: This will trigger alerts. To suppress alerts, turn on Maintenance Mode for HDFS prior to triggering a rolling restart	
Restart 1 DataNodes at a time Wait 120 seconds between batches Tolerate up to 1 restart failures Only restart DataNodes with stale config	gs
Cancel Trigger Rolling Restart	

Rolling restart parameter values must satisfy the following criteria:

Validation	Rules f	for Rollin	g Restart	Parameters
------------	----------------	------------	-----------	------------

Parameter	Required	Value	Description
Batch Size	Yes	Must be an integer > 0	Number of components to include in e
Wait Time	Yes	Must be an integer > = 0	Time (in seconds) to wait between que
Tolerate up to x failures	Yes	Must be an integer > = 0	Total number of restart failures to tole halting the restarts and not queuing b

4.9.2. Aborting a Rolling Restart

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

Operations	Hosts	Show: All (1)	:
Future operations of this	batch request can be aborted	Abort Rolling Restar	1
✓ c6403.ambari.apact	ne.org	100%	,

4.10. Refreshing YARN Capacity Scheduler

After you modify the Capacity Scheduler configuration, YARN supports refreshing the queues without requiring you to restart your ResourceManager. The "refresh" operation is valid if you have made no destructive changes to your configuration. Removing a queue is an example of a destructive change.

4.10.1. How to refresh the YARN Capacity Scheduler

This topic describes how to refresh the Capacity Scheduler in cases where you have added or modified existing queues.

- In Ambari Web, browse to Services > YARN > Summary.
- Select Service Actions, then choose Refresh YARN Capacity Scheduler.
- Confirm you would like to perform this operation.

The refresh operation is submitted to the YARN ResourceManager.



Important

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.

4.11. Rebalancing HDFS

HDFS provides a "balancer" utility to help balance the blocks across DataNodes in the cluster.

4.11.1. How to rebalance HDFS

This topic describes how you can initiate an HDFS rebalance from Ambari.

- 1. . In Ambari Web, browse to Services > HDFS > Summary.
- 2. Select Service Actions, then choose Rebalance HDFS.
- 3. Enter the Balance Threshold value as a percentage of disk capacity.

Rebalance HDFS	Х
Balancer threshold (percentage of disk capacity):	
10	
Cancel	Start

- 4. Click Start to begin the rebalance.
- 5. You can check rebalance progress or cancel a rebalance in process by opening the Background Operations dialog.

5. Managing Service High Availability

Ambari provides the ability to configure the High Availability features available with the HDP Stack services. This section describes how to enable HA for the various Stack services.

- NameNode High Availability
- ResourceManager High Availability
- HBase High Availability
- Hive High Availability
- Oozie High Availability

5.1. NameNode High Availability

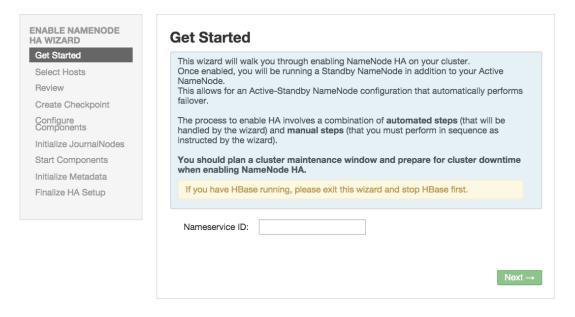
To ensure that a NameNode in your cluster is always available if the primary NameNode host fails, enable and set up NameNode High Availability on your cluster using Ambari Web.

Follow the steps in the Enable NameNode HA Wizard.

For more information about using the Enable NameNode HA Wizard, see How to Configure NameNode High Availability.

5.1.1. How To Configure NameNode High Availability

- 1. Check to make sure you have at least three hosts in your cluster and are running at least three ZooKeeper servers.
- 2. In Ambari Web, select Services > HDFS > Summary.
- 3. Select Service Actions and choose Enable NameNode HA.
- 4. The Enable HA Wizard launches. This wizard describes the set of automated and manual steps you must take to set up NameNode high availability.
- 5. **Get Started** : This step gives you an overview of the process and allows you to select a Nameservice ID. You use this Nameservice ID instead of the NameNode FQDN once HA has been set up. Click Next to proceed.



6. Select Hosts : Select a host for the additional NameNode and the JournalNodes. The wizard suggest options that you can adjust using the drop-down lists. Click Next to proceed.

	t will be running the additional NameNode. t the hosts to run JournalNodes, which store N	lameNode edit logs in a fault tolerant manner.
Current NameNode:	c6401.ambari.apache.org (1.8 GI 💠	c6401.ambari.apache.org (1.8 GB, 1 cores NameNode HBase Master ZooKeepe
Additional NameNode:	c6402.ambari.apache.org (1.8 Gi 🗘	JournalNode
JournalNode:	c6401.ambari.apache.org (1.8 Gi 💠	c6402.ambari.apache.org (1.8 GB, 1 cores SNameNode History Server
JournalNode:	c6402.ambari.apache.org (1.8 Gi 💠	ResourceManager App Timeline Server Nagios Server Ganglia Server
JournalNode:	c6403.ambari.apache.org (1.8 GI 💠	HiveServer2 Hive Metastore WebHCat Server Oozie Server
		ZooKeeper Falcon Server Nimbus Storm UI Server Logviewer Server DRPC Server Storm REST API Server
		DRPC Server Storm REST API Server JournalNode NameNode
		c6403.ambari.apache.org (1.8 GB, 1 cores ZooKeeper JournalNode

7. Review : Confirm your host selections and click Next.

Review			
Confirm your host select	ons.		
Current NameNode:	c6401.ambari.apache.org		
Secondary NameNode:	c6402.ambari.apache.org - TO BE DELETED		
Additional NameNode:	c6402.ambari.apache.org + TO BE INSTALLED		
JournalNode:	c6401.ambari.apache.org + TO BE INSTALLED c6402.ambari.apache.org + TO BE INSTALLED c6403.ambari.apache.org + TO BE INSTALLED		
	inges. guration changes that will be made by the Wizard to enable NameNode HA. This ay and is not editable except for the dfs.journalinode.edits.dir.property		
+ HDFS			
• HBase			
← Back	Next		

8. Create Checkpoints : Follow the instructions in the step. You need to log in to your current NameNode host to run the commands to put your NameNode into safe mode and create a checkpoint. When Ambari detects success, the message on the bottom of the window changes. Click Next.

Anomal Steps Required: Create Checkpoint on barrent of the state of t

9. Configure Components : The wizard configures your components, displaying progress bars to let you track the steps. Click Next to continue.

Configure Components		
Please proceed to the next step.		
V Stop All Services		
 Install Additional NameNode 		
✓ Install JournalNodes		
 Reconfigure HDFS 		
✓ Start JournalNodes		
Disable Secondary NameNode		
	Next	

10Jnitialize JournalNodes : Follow the instructions in the step. You need to login to your current NameNode host to run the command to initialize the JournalNodes. When Ambari detects success, the message on the bottom of the window changes. Click Next.

Manual Steps Required: Initialize JournalNodes
Login to the NameNode host 69401.ambarl.apache.org. Initiatize the JournalNodes by running: Imade set John 4 *** 04fa assessed -stitulizetManedRdits* Sou will be able to proceed once Ambari detects that the JournalNodes have been Initialized successfully.
JournalNodes Initialized Next

11 Start Components : The wizard starts the ZooKeeper servers and the NameNode, displaying progress bars to let you track the steps. Click Next to continue.

Start Components		
Please proceed to the next step.		
 Start ZooKeeper Servers 		
 Start NameNode 		
	No	xt

12 Initialize Metadata : Follow the instructions in the step. For this step you must log in to both the current NameNode and the additional NameNode. Make sure you are logged in to the correct host for each command. Click Next when you have completed the two commands. A Confirmation pop-up window displays, reminding you to do both steps. Click OK to confirm.

Manual Steps Required: Initialize NameNode HA Metadata 1. Login to the NameNode host of401.ambari.apasehe.org. 2. Initialize the metadata for NameNode subcrastic failower by nunning: redo re -1 Mds -e "Mds e atte -formatat"

In	portant! Be sure to login to the Additional NameNode host,
	s is a different host from the Steps 1 and 2 above.
Initia	ize the metadata for the Additional NameNode by running:
sude	su -1 hdfs -c 'hdfs namenode -bootstrapStandby'

13**Finalize HA Setup** : The wizard the setup, displaying progress bars to let you track the steps. Click Done to finish the wizard. After the Ambari Web GUI reloads, you may see some alert notifications. Wait a few minutes until the services come back up. If necessary, restart any components using Ambari Web.

Finalize HA Setup	
Please wait while the wizard finalizes the HA setup.	
Start Additional NameNode	
✓ Install Failover Controllers	
✓ Start Failover Controllers	
✓ Reconfigure HBase	
Delete Secondary NameNode	
Start All Services 72%	

- 14Jf you are using Hive, you must manually change the Hive Metastore FS root to point to the Nameservice URI instead of the NameNode URI. You created the Nameservice ID in the Get Started step.
 - a. Check the current FS root. On the Hive host:

```
hive --config /etc/hive/conf.server --service metatool -
listFSRoot
```

The output looks similar to the following: Listing FS Roots... hdfs:// <namenode-host>/apps/hive/warehouse

b. Use this command to change the FS root:

```
$ hive --config /etc/hive/conf.server --service metatool -
updateLocation <new-location><old-location>For example, where the
Nameservice ID is mycluster:$ hive --config /etc/hive/conf.server --
service metatool -updateLocation hdfs://mycluster/apps/hive/
warehouse hdfs://c6401.ambari.apache.org/apps/hive/warehouse
```

The output looks similar to the following:

```
Successfully updated the following locations...Updated X records in SDS table
```

15 Adjust the ZooKeeper Failover Controller retries setting for your environment.

• Browse to Services > HDFS > Configs >core-site.

• Set ha.failover-controller.active-standbyelector.zk.op.retries=120

5.1.2. How to Roll Back NameNode HA

To roll back NameNode HA to the previous non-HA state use the following step-by-step manual process, depending on your installation.

- 1. Stop HBase
- 2. Checkpoint the Active NameNode
- 3. Stop All Services
- 4. Prepare the Ambari Host for Rollback
- 5. Restore the HBase Configuration
- 6. Delete ZooKeeper Failover Controllers
- 7. Modify HDFS Configurations
- 8. Recreate the standby NameNode
- 9. Re-enable the standby NameNode
- 10.Delete All JournalNodes
- 11.Delete the Additional NameNode
- 12.Verify the HDFS Components
- 13Start HDFS

5.1.2.1. Stop HBase

- 1. From Ambari Web, go to the Services view and select HBase.
- 2. Choose Service Actions > Stop.
- 3. Wait until HBase has stopped completely before continuing.

5.1.2.2. Checkpoint the Active NameNode

If HDFS has been in use **after** you enabled NameNode HA, but you wish to revert back to a non-HA state, you must checkpoint the HDFS state before proceeding with the rollback.

If the Enable NameNode HA wizard failed and you need to revert back, you can skip this step and move on to Stop All Services.

• If Kerberos security has **not** been enabled on the cluster:

On the Active NameNode host, execute the following commands to save the namespace. You must be the HDFS service user to do this.

sudo su -l <HDFS_USER> -c 'hdfs dfsadmin -safemode enter' sudo su -l <HDFS_USER> -c 'hdfs dfsadmin -saveNamespace'

• If Kerberos security has been enabled on the cluster:

sudo su -l <HDFS_USER> -c 'kinit -kt /etc/security/keytabs/ nn.service.keytab nn/<HOSTNAME>@<REALM>;hdfs dfsadmin -safemode enter' sudo su -l <HDFS_USER> -c 'kinit -kt /etc/security/ keytabs/nn.service.keytab nn/<HOSTNAME>@<REALM>;hdfs dfsadmin saveNamespace'

Where <HDFS_USER> is the HDFS service user; for example hdfs, <HOSTNAME> is the Active NameNode hostname, and <REALM> is your Kerberos realm.

5.1.2.3. Stop All Services

Browse to Ambari Web > Services, then choose Stop All in the Services navigation panel. You must wait until all the services are completely stopped.

5.1.2.4. Prepare the Ambari Server Host for Rollback

Log into the Ambari server host and set the following environment variables to prepare for the rollback procedure:

Variable	Value
export AMBARI_USER=AMBARI_USERNAME	Substitute the value of the administrative user for Ambari Web. The default value is admin.
export AMBARI_PW=AMBARI_PASSWORD	Substitute the value of the administrative password for Ambari Web. The default value is admin.
export AMBARI_PORT=AMBARI_PORT	Substitute the Ambari Web port. The default value is 8080.
export	Substitute the value of the protocol for connecting to Ambari Web. Options are http or https.
AMBARI_PROTO=AMBARI_PROTOCOL	The default value is http.
export CLUSTER_NAME=CLUSTER_NAME	Substitute the name of your cluster, set during the Ambari Install Wizard process. For example: mycluster.
export	Substitute the FQDN of the host for the non-HA NameNode. For example:
NAMENODE_HOSTNAME=NN_HOSTNAM	Enn01.mycompany.com.
export	Substitute the FQDN of the host for the additional NameNode in your HA setup.
ADDITIONAL_NAMENODE_HOSTNAME=4	NN_HOSTNAME
export	Substitute the FQDN of the host for the standby NameNode for the non-HA setup.
SECONDARY_NAMENODE_HOSTNAME=S	NN_HOSTNAME
export	Substitute the FQDN of the host for the first Journal Node.
JOURNALNODE1_HOSTNAME=JOUR1_HC	STNAME
export	Substitute the FQDN of the host for the second Journal Node.
JOURNALNODE2_HOSTNAME=JOUR2_HC	STNAME
export	Substitute the FQDN of the host for the third Journal Node.
JOURNALNODE3_HOSTNAME=JOUR3_HC	STNAME

Double check that these environment variables are set correctly.

5.1.2.5. Restore the HBase Configuration

If you have installed HBase, you may need to restore a configuration to its pre-HA state.

1. To check if your current HBase configuration needs to be restored, on the Ambari Server host:

```
/var/lib/ambari-server/resources/scripts/configs.sh -u
<AMBARI_USER> -p <AMBARI_PW> -port <AMBARI_PORT> get localhost
<CLUSTER_NAME> hbase-site
```

Where the environment variables you set up in Prepare the Ambari Server Host for Rollback substitute for the variable names.

Look for the configuration property hbase.rootdir. If the value is set to the NameService ID you set up using the Enable NameNode HA wizard, you need to revert the hbase-site configuration set up back to non-HA values. If it points instead to a specific NameNode host, it does not need to be rolled back and you can go on to Delete ZooKeeper Failover Controllers.

For example:

"hbase.rootdir": "hdfs://<name-service-id>:8020/apps/hbase/data" The hbase.rootdir property points to the NameService ID and the value needs to be rolled back "hbase.rootdir": "hdfs://<nn01.mycompany.com>:8020/apps/ hbase/data" The hbase.rootdir property points to a specific NameNode host and not a NameService ID. This does not need to be rolled back.

2. If you need to roll back the hbase.rootdir value, on the Ambari Server host, use the config.sh script to make the necessary change:

```
/var/lib/ambari-server/resources/scripts/configs.sh -
u <AMBARI_USER> -p<AMBARI_PW> -port <AMBARI_PORT> set
localhost <CLUSTER_NAME> hbase-site hbase.rootdir hdfs://
<NAMENODE_HOSTNAME>:8020/apps/hbase/data
```

Where the environment variables you set up in Prepare the Ambari Server Host for Rollback substitute for the variable names.

3. Verify that the hbase.rootdir property has been restored properly. On the Ambari Server host:

```
/var/lib/ambari-server/resources/scripts/configs.sh -u
<AMBARI_USER> -p <AMBARI_PW> -port <AMBARI_PORT> get localhost
<CLUSTER_NAME> hbase-site
```

The hbase.rootdir property should now be set to the NameNode hostname, not the NameService ID.

5.1.2.6. Delete ZooKeeper Failover Controllers

You may need to delete ZooKeeper (ZK) Failover Controllers.

1. To check if you need to delete ZK Failover Controllers, on the Ambari Server host:

```
curl -u <AMBARI_USER>:<AMBARI_PW> -H "X-Requested-By: ambari"
-i <AMBARI_PROTO>://localhost:<AMBARI_PORT>/api/v1/clusters/
<CLUSTER_NAME>/host_components?HostRoles/component_name=ZKFC
```

If this returns an empty items array, you may proceed to Modify HDFS Configuration. Otherwise you must use the following DELETE commands:

2. To delete all ZK Failover Controllers, on the Ambari Server host:

```
curl -u <AMBARI_USER>:<AMBARI_PW> -H "X-Requested-By: ambari"
-i -X DELETE <AMBARI_PROTO>://localhost:<AMBARI_PORT>/
api/v1/clusters/<CLUSTER_NAME>/hosts/<NAMENODE_HOSTNAME>/
host_components/ZKFC curl -u <AMBARI_USER>:<AMBARI_PW> -
H "X-Requested-By: ambari" -i -X DELETE <AMBARI_PROTO>://
localhost:<AMBARI_PORT>/api/v1/clusters/<CLUSTER_NAME>/hosts/
<ADDITIONAL_NAMENODE_HOSTNAME>/host_components/ZKFC
```

3. Verify that the ZK Failover Controllers have been deleted. On the Ambari Server host:

```
curl -u <AMBARI_USER>:<AMBARI_PW> -H "X-Requested-By: ambari"
-i <AMBARI_PROTO>://localhost:<AMBARI_PORT>/api/v1/clusters/
<CLUSTER_NAME>/host_components?HostRoles/component_name=ZKFC
```

This command should return an empty items array.

5.1.2.7. Modify HDFS Configurations

You may need to modify your hdfs-site configuration and/or your core-site configuration.

1. To check if you need to modify your hdfs-site configuration, on the Ambari Server host:

```
/var/lib/ambari-server/resources/scripts/configs.sh -u
<AMBARI_USER> -p <AMBARI_PW> -port <AMBARI_PORT> get localhost
<CLUSTER_NAME> hdfs-site
```

If you see **any** of the following properties, you must delete them from your configuration.

- dfs.nameservices
- dfs.client.failover.proxy.provider.<NAMESERVICE_ID>
- dfs.ha.namenodes.<NAMESERVICE_ID>
- dfs.ha.fencing.methods
- dfs.ha.automatic-failover.enabled
- dfs.namenode.http-address.<NAMESERVICE_ID>.nnl
- dfs.namenode.http-address.<NAMESERVICE_ID>.nn2

- dfs.namenode.rpc-address.<NAMESERVICE_ID>.nn1
- dfs.namenode.rpc-address.<NAMESERVICE_ID>.nn2
- dfs.namenode.shared.edits.dir
- dfs.journalnode.edits.dir
- dfs.journalnode.http-address
- dfs.journalnode.kerberos.internal.spnego.principal
- dfs.journalnode.kerberos.principal
- dfs.journalnode.keytab.file

Where <NAMESERVICE_ID> is the NameService ID you created when you ran the Enable NameNode HA wizard.

2. To delete these properties, execute the following **for each property** you found. On the Ambari Server host:

/var/lib/ambari-server/resources/scripts/configs.sh -u
<AMBARI_USER> -p <AMBARI_PW> -port <AMBARI_PORT> delete
localhost <CLUSTER_NAME> hdfs-site property_name

Where you replace property_name with the name of each of the properties to be deleted.

3. Verify that all of the properties have been deleted. On the Ambari Server host: /var/ lib/ambari-server/resources/scripts/configs.sh -u <AMBARI_USER> -p <AMBARI_PW> -port <AMBARI_PORT> get localhost <CLUSTER_NAME> hdfs-site

None of the properties listed above should be present.

- 4. To check if you need to modify your core-site configuration, on the Ambari Server host: /var/lib/ambari-server/resources/scripts/configs.sh -u <AMBARI_USER> -p <AMBARI_PW> -port <AMBARI_PORT> get localhost <CLUSTER_NAME> core-site
- 5. If you see the property ha.zookeeper.quorum, it must be deleted. On the Ambari Server host:

/var/lib/ambari-server/resources/scripts/configs.sh -u
<AMBARI_USER> -p <AMBARI_PW> -port <AMBARI_PORT> delete
localhost <CLUSTER_NAME> core-site ha.zookeeper.quorum

6. If the property fs.defaultFS is set to the NameService ID, it must be reverted back to its non-HA value. For example:

```
"fs.defaultFS":"hdfs://<name-service-id>" The property
fs.defaultFS needs to be modified as it points to a NameService
ID "fs.defaultFS":"hdfs://<nn01.mycompany.com>" The property
```

 $\tt fs.defaultFS$ does not need to be changed as it points to a specific NameNode, not to a NameService ID

7. To revert the property fs.defaultFS to the NameNode host value, on the Ambari Server host:

```
/var/lib/ambari-server/resources/scripts/configs.sh -u
<AMBARI_USER> -p <AMBARI_PW> -port <AMBARI_PORT> set localhost
<CLUSTER_NAME> core-site fs.defaultFS hdfs://<NAMENODE_HOSTNAME>
```

8. Verify that the core-site properties are now properly set. On the Ambari Server host:

/var/lib/ambari-server/resources/scripts/configs.sh -u
<AMBARI_USER> -p <AMBARI_PW> -port <AMBARI_PORT> get localhost
<CLUSTER_NAME> core-site

The property fs.defaultFS should be set to point to the NameNode host and the property ha.zookeeper.quorum should not be there.

5.1.2.8. Recreate the Standby NameNode

You may need to recreate your standby NameNode.

1. To check to see if you need to recreate the standby NameNode, on the Ambari Server host:

```
curl -u <AMBARI_USER>:<AMBARI_PW> -H "X-Requested-By:
ambari" -i -X GET <AMBARI_PROTO>://localhost:<AMBARI_PORT>/
api/v1/clusters/<CLUSTER_NAME>/host_components?HostRoles/
component_name=SECONDARY_NAMENODE
```

If this returns an empty items array, you must recreate your standby NameNode. Otherwise you can go on to Re-enable Standby NameNode.

- 2. Recreate your standby NameNode. On the Ambari Server host: curl u <AMBARI_USER>:<AMBARI_PW> -H "X-Requested-By: ambari"
 -i -X POST -d '{"host_components" : [{"HostRoles":
 {"component_name":"SECONDARY_NAMENODE"}] }' <AMBARI_PROTO>://
 localhost:<AMBARI_PORT>/api/v1/clusters/<CLUSTER_NAME>/hosts?
 Hosts/host_name=<SECONDARY_NAMENODE_HOSTNAME>
- 3. Verify that the standby NameNode now exists. On the Ambari Server host:

```
curl -u <AMBARI_USER>:<AMBARI_PW> -H "X-Requested-By:
ambari" -i -X GET <AMBARI_PROTO>://localhost:<AMBARI_PORT>/
api/v1/clusters/<CLUSTER_NAME>/host_components?HostRoles/
component_name=SECONDARY_NAMENODE
```

This should return a non-empty items array containing the standby NameNode.

5.1.2.9. Re-enable the Standby NameNode

To re-enable the standby NameNode, on the Ambari Server host:

```
curl -u <AMBARI_USER>:<AMBARI_PW> -H "X-Requested-By: ambari" -i -
X '{"RequestInfo":{"context":"Enable Secondary NameNode"},"Body":
{"HostRoles":{"state":"INSTALLED"}}}'<AMBARI_PROTO>://
localhost:<AMBARI_PORT>/api/v1/clusters/<CLUSTER_NAME>/hosts/
<SECONDARY_NAMENODE_HOSTNAME}/host_components/SECONDARY_NAMENODE</pre>
```

- If this returns 200, go to Delete All JournalNodes.
- If this returns 202, wait a few minutes and run the following command on the Ambari Server host:

```
curl -u <AMBARI_USER>:${AMBARI_PW -H "X-Requested-By:
ambari" -i -X "<AMBARI_PROTO>://localhost:<AMBARI_PORT>/
api/v1/clusters/<CLUSTER_NAME>/host_components?HostRoles/
component_name=SECONDARY_NAMENODE&fields=HostRoles/state"
```

When "state" : "INSTALLED" is in the response, go on to the next step.

5.1.2.10. Delete All JournalNodes

You may need to delete any JournalNodes.

1. To check to see if you need to delete JournalNodes, on the Ambari Server host:

```
curl -u <AMBARI_USER>:<AMBARI_PW> -H "X-Requested-By:
ambari" -i -X GET <AMBARI_PROTO>://localhost:<AMBARI_PORT>/
api/v1/clusters/<CLUSTER_NAME>/host_components?HostRoles/
component_name=JOURNALNODE
```

If this returns an empty items array, you can go on to Delete the Additional NameNode. Otherwise you must delete the JournalNodes.

2. To delete the JournalNodes, on the Ambari Server host:

```
curl -u <AMBARI_USER>:<AMBARI_PW> -H "X-Requested-By: ambari"
-i -X DELETE <AMBARI_PROTO>://localhost:<AMBARI_PORT>/api/
v1/clusters/<CLUSTER_NAME>/hosts/<JOURNALNODE1_HOSTNAME>/
host_components/JOURNALNODE curl -u <AMBARI_USER>:<AMBARI_PW>
-H "X-Requested-By: ambari" -i -X DELETE <AMBARI_PROTO>://
localhost:<AMBARI_PORT>/api/v1/clusters/<CLUSTER_NAME>/hosts/
<JOURNALNODE2_HOSTNAME>/host_components/JOURNALNODE
curl -u <AMBARI_USER>:<AMBARI_PW> -H "X-Requested-By: ambari"
-i -X DELETE <AMBARI_PROTO>://localhost:<AMBARI_PORT>/api/
v1/clusters/<CLUSTER_NAME>/hosts/<JOURNALNODE3_HOSTNAME>/
host_components/JOURNALNODE
```

3. Verify that all the JournalNodes have been deleted. On the Ambari Server host:

```
curl -u <AMBARI_USER>:<AMBARI_PW> -H "X-Requested-By:
ambari" -i -X GET <AMBARI_PROTO>://localhost:<AMBARI_PORT>/
api/v1/clusters/<CLUSTER_NAME>/host_components?HostRoles/
component_name=JOURNALNODE
```

This should return an empty items array.

5.1.2.11. Delete the Additional NameNode

You may need to delete your Additional NameNode.

1. To check to see if you need to delete your Additional NameNode, on the Ambari Server host:

```
curl -u <AMBARI_USER>:<AMBARI_PW> -H "X-Requested-By: ambari" -i
-X GET <AMBARI_PROTO>://localhost:<AMBARI_PORT>/api/v1/clusters/
<CLUSTER_NAME>/host_components?HostRoles/component_name=NAMENODE
```

If the items array contains two NameNodes, the Additional NameNode must be deleted.

2. To delete the Additional NameNode that was set up for HA, on the Ambari Server host:

```
curl -u <AMBARI_USER>:<AMBARI_PW> -H "X-Requested-By: ambari"
-i -X DELETE <AMBARI_PROTO>://localhost:<AMBARI_PORT>/api/v1/
clusters/<CLUSTER_NAME>/hosts/<ADDITIONAL_NAMENODE_HOSTNAME>/
host_components/NAMENODE
```

3. Verify that the Additional NameNode has been deleted:

```
curl -u <AMBARI_USER>:<AMBARI_PW> -H "X-Requested-By: ambari" -i
-X GET <AMBARI_PROTO>://localhost:<AMBARI_PORT>/api/v1/clusters/
<CLUSTER_NAME>/host_components?HostRoles/component_name=NAMENODE
```

This should return an items array that shows only one NameNode.

5.1.2.12. Verify the HDFS Components

Make sure you have the correct components showing in HDFS.

- 1. Go to Ambari Web UI > Services, then select HDFS.
- 2. Check the Summary panel and make sure that the first three lines look like this:
 - NameNode
 - SNameNode
 - DataNodes

You should not see any line for JournalNodes.

5.1.2.13. Start HDFS

1. In the Ambari Web UI, select Service Actions, then choose Start.

Wait until the progress bar shows that the service has completely started and has passed the service checks.

If HDFS does not start, you may need to repeat the previous step.

2. To start all of the other services, select Actions > Start All in the Services navigation panel.

5.2. ResourceManager High Availability



Note

This feature is available with HDP Stack 2.2 or later.

- 1. Check to make sure you have at least three hosts in your cluster and are running at least three ZooKeeper servers.
- 2. In Ambari Web, browse to Services > YARN > Summary. Select Service Actions and choose Enable ResourceManager HA.
- 3. The Enable ResourceManager HA Wizard launches. The wizard describes a set of automated and manual steps you must take to set up ResourceManager High Availability.
- 4. Get Started: This step gives you an overview of enabling ResourceManager HA. Click ${\tt Next}$ to proceed.

ENABLE RESOURCEMANAGER HA WIZARD	Get Started
Get Started Select Host	This wizard will walk you through enabling ResourceManager HA on your cluster. Once enabled, you will be running a Standby ResourceManager in addition to your Active ResourceManager. This allows for an Active-Standby ResourceManager configuration that automatically performs failover.
Review Configure Components	You should plan a cluster maintenance window and prepare for cluster downtime when enabling ResourceManager HA.
	Next →

5. Select Host: The wizard shows you the host on which the current ResourceManager is installed and suggests a default host on which to install an additional ResourceManager. Accept the default selection, or choose an available host. Click Next to proceed.

ENABLE RESOURCEMANAGER HA WIZARD	Select Host
Get Started	Select a host that will be running the additional ResourceManager
Select Host Review	Current ResourceManager: 06402.ambari.apache.org (1.8 G ‡) 06401.ambari.apache.org (1.8 GB, 1 cores)
Configure Components	Additional ResourceManager: c6401.ambari.apachs.org (1.8 G 1 Httase Matter Zookkeppe Server Kota Botex, Kotos Catavey
	C6402 ambain apathe org 11.8 GB, 1 Corres) Biterina Stars Rog St
	o6403.ambari.apache.org (1.8 GB, 1 cores) Zooliesper Sarver
	- Back Next

6. **Review Selections**: The wizard shows you the host selections and configuration changes that will occur to enable ResourceManager HA. Expand YARN, if necessary, to review all the YARN configuration changes. Click Next to approve the changes and start automatically configuring ResourceManager HA.

NABLE ESOURCEMANAGER HA IZARD	Review
Get Started	Confirm your host selections.
Select Host	
Review	Current ResourceManager: c6402.ambari.apache.org
Configure Components	Additional ResourceManager: c6401.ambari.apache.org + TO BE INSTALLED
	Review Configuration Changes. The following lists the configuration changes that will be made by the Waterd to enable ResourceManager HK. This information is for enable ends and is not astitute.
	← Back Next →

7. **Configure Components**: The wizard configures your components automatically, displaying progress bars to let you track the steps. After all progress bars complete, click <code>Complete</code> to finish the wizard.

ENABLE RESOURCEMANAGER HA WIZARD	Configure Components
Get Started	ResourceManager HA has been enabled successfully.
Select Host	✓ Stop Required Services
Configure Components	 Install Additional ResourceManager
	✓ Reconfigure YARN
	✓ Start All Services
	Complete

5.3. HBase High Availability

During the HBase service install, depending on your component assignment, Ambari installs and configures one HBase Master component and multiple RegionServer components. To setup high availability for the HBase service, you can run two or more HBase Master components by adding an HBase Master component. Once running two or more HBase Masters, HBase uses ZooKeeper for coordination of the active Master.

5.3.1. Adding an HBase Master Component

- 1. In Ambari Web, browse to Services > HBase.
- 2. In Service Actions, select the + Add HBase Master option.
- 3. Choose the host to install the additional HBase Master, then choose Confirm Add.

Ambari installs the new HBase Master and reconfigure HBase to handle multiple Master instances.

5.4. Hive High Availability

The Hive service has multiple, associated components. The primary Hive components are: Hive Metastore and HiveServer2. To setup high availability for the Hive service, you can run two or more of each of those components.



Note

This feature is available with HDP 2.2 Stack.



Important

The relational database that backs the Hive Metastore itself should also be made highly available using best practices defined for the database system in use.

5.4.1. Adding a Hive Metastore Component

- 1. In Ambari Web, browse to Services > Hive.
- 2. In Service Actions, select the + Add Hive Metastore option.
- 3. Choose the host to install the additional Hive Metastore, then choose Confirm Add.
- 4. Ambari installs the component and reconfigures Hive to handle multiple Hive Metastore instances.

5.4.2. Adding a HiveServer2 Component

- 1. In Ambari Web, browse to the host where you would like to install another HiveServer2.
- 2. On the Host page, choose +Add.
- 3. Select HiveServer2 from the list.
- 4. Ambari installs the new HiveServer2.

Ambari installs the component and reconfigures Hive to handle multiple Hive Metastore instances.

5.5. Oozie High Availability

To setup high availability for the Oozie service, you can run two or more instances of the Oozie Server component.



Note

This capability is available with HDP 2.2 Stack.



Important

The relational database that backs the Oozie Server should also be made highly available using best practices defined for the database system in use. Using the default installed Derby database instance is not supported with multiple Oozie Server instances and therefore, you must use an existing relational database. When using Derby for the Oozie Server, you will not have an option to add Oozie Server components to your cluster.



Important

High availability for Oozie requires the use of an external Virtual IP Address or Load Balancer to direct traffic to the Oozie servers.

5.5.1. Adding an Oozie Server Component

- 1. In Ambari Web, browse to the host where you would like to install another Oozie Server.
- 2. On the Host page, click the "+Add" button.
- 3. Select "Oozie Server" from the list and Ambari will install the new Oozie Server.
- 4. After configuring your external Load Balancer, update the oozie configuration.
- 5. Browse to Services > Oozie > Configs and in oozie-site add the following:

Property	Value
oozie.zookeeper.connection.string	List of ZooKeeper hosts with ports. For example:
	c6401.ambari.apache.org:2181,c6402.ambari.apache.org:2181,c6403.ambari.apach
oozie.services.ext	org.apache.oozie.service.ZKLocksService,org.apache.oozie.service.ZKXLogStreamin
oozie.base.url	http:// <loadbalancer.hostname>:11000/oozie</loadbalancer.hostname>

6. In oozie-env, uncomment OOZIE_BASE_URL property and change value to point to the Load Balancer. For example:

```
export OOZIE_BASE_URL="http://<loadbalance.hostname>:11000/
oozie"
```

- 7. Restart Oozie service for the changes to take affect.
- 8. Update HDFS configs for the Oozie proxy user. Browse to Services > HDFS > Configs and in core-site update the hadoop.proxyuser.oozie.hosts property to include the newly added Oozie Server host. Hosts should be comma separated.
- 9. Restart all needed services.

6. Managing Configurations

Use Ambari Web to manage your HDP component configurations. Select any of the following topics:

- Configuring Services
- Using Host Config Groups
- Customizing Log Settings
- Downloading Client Configs
- Service Configuration Versions

6.1. Configuring Services

Select a service, then select Configs 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.

6.1.1. Updating Service Properties

- 1. Expand a configuration category.
- 2. Edit values for one or more properties that have the Override option.

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

3. Choose Save.

6.1.2. 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:



6.2. Using Host Config 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.

Manage HBase C	onfiguration Groups	Х
HBase Configuration Groups	of HBase configurations to groups of hosts by managing and their host membership. Hosts belonging to a HBase same set of configurations for HBase. Each host belongs to xup.	
HBase Default (3)	c6401.ambari.apache.org c6402.ambari.apache.org c6403.ambari.apache.org	
+ - 0-	Overrides 0 properties Description Default cluster level HBASE configuration	
	Cancel	ve

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.

	ct hosts that should belong to this HDFS p will have the same set of HDFS config		osts belonging to this
out o	f 2 hosts selected	Filter	Components •
	Host	IP Address	NameNode
	c6402.ambari.apache.org	10.0.2.15	SNameNode © DataNode
	c6403.ambari.apache.org	10.0.2.15	HDFS Client

- 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.

 DataNode 				
DataNode directories	/hadoop/hdfs/d	ata		Override
DataNode maximum Java heap size	1024	мв	Remove This is required	
DataNode volumes failure toleration	0			Override
DataNode directories	750			Override

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

Select or create a HDF	S Configuration Group where the configuration value will be overridden.
Select an existing HDF	S Configuration Group
custom log 4j hosts	•
Overridden property w	Il be changed for hosts belonging to the selected group.
Create a new HDFS Co	Infiguration Group
Create a new HDFS Co	· · ·

- 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.

6.3. Customizing Log Settings

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

# Define some custom values that can be overridden by custom properties hadoop.root.logger=INFO,console hadoop.log.dfra, hadoop.log.dfra,	0	
# Define the root logger to the system property "hadoop.root.logger". log4j.rootLogger=\${hadoop.root.logger}, EventCounter		
# Logging Threshold log4j,threshhold=ALL		Override
# # Daily Rolling File Appender #		
log4j.appender.DRFA=org.apache.log4j.DailyRollingFileAppender log4j.appender.DRFA.File=\${hadoop.log.dir}/\${hadoop.log.file}	6	

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 saving and overriding configuration settings, see Updating Service Config Properties.

6.4. Downloading Client Configs

For Services that include client components (for example Hadoop Client or Hive Client), you can download the client configuration files associated with that client from Ambari.

- In Ambari Web, browse to the Service with the client for which you want the configurations.
- Choose Service Actions.
- Choose Download Client Configs. You are prompted for a location to save the client configs bundle.

HDFS_CLIENT	-configs.tar.gz	
which is: tar a	archive (5.3 KB)	
from: http://	c6401.ambari.apache.org:8080	
What should Fi	refox do with this file?	
Open with	Archive Utility (default)	\$
 Save File 		
Do this out	emotionly for files like this from a	
	omatically for files like this from n	ow on.

• Save the bundle.

6.5. Service Configuration Versions

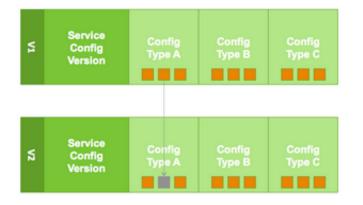
Ambari provides the ability to manage configurations associated with a Service. You can make changes to configurations, see a history of changes, compare + revert changes and push configuration changes to the cluster hosts.

- Basic Concepts
- Terminology
- Saving a Change
- Viewing History
- Comparing Versions
- Reverting a Change
- Versioning and Host Config Groups

6.5.1. Basic Concepts

It's important to understand how service configurations are organized and stored in Ambari. Properties are grouped into Configuration Types (config types). A set of config types makes up the set of configurations for a service. For example, the HDFS Service includes the following config types: hdfs-site, core-site, hdfs-log4j, hadoop-env, hadoop-policy. If you browse to Services > HDFS > Configs, the configuration properties for these config types are available for edit.

Versioning of configurations is performed at the service-level. Therefore, when you modify a configuration property in a service, Ambari will create a Service Config Version. The figure below shows V1 and V2 of a Service Configuration Version with a change to a property in Config Type A. After making the property change to Config Type A in V1, V2 is created.



6.5.2. Terminology

The following table lists configuration versioning terms and concepts that you should know.

Term	Description
Configuration Property	Configuration property managed by Ambari, such as NameNode heapsize or replication factor.
Configuration Type	Group of configuration properties. For example: hdfs-site is a Config Type.
(Config Type)	
Service Configurations	Set of configuration types for a particular service. For example: hdfs-site and core-site Config Types are part of the HDFS Service Configuration.
Change Notes	Optional notes to save with a service configuration change.
Service Config Version (SCV)	Particular version of configurations for a specific service. Ambari saves a history of service configuration versions.
Host Config Group (HCG)	Set of configuration properties to apply to a specific set of hosts. Each service has a default Host Config Group, and custom config groups can be created on top of the default configuration group to target property overrides to one or more hosts in the cluster. See Managing Configuration Groups for more information.

6.5.3. Saving a Change

- 1. Make the configuration property change.
- 2. Choose Save.
- 3. You are prompted to enter notes that describe the change.

Notes	What did you change?				1
-------	----------------------	--	--	--	---

4. Click Save to confirm your change. Cancel will not save but instead returns you to the configuration page to continuing editing.

To revert the changes you made and not save, choose Discard.

To return to the configuration page and continue editing without saving changes, choose Cancel.

6.5.4. Viewing History

Service Config Version history is available from Ambari Web in two places: On the Dashboard page under the Config History tab; and on each Service page under the Configs tab.

The Dashboard > Config History tab shows a list of all versions across services with each version number and the date and time the version was created. You can also see which user authored the change with the notes entered during save. Using this table, you can filter, sort and search across versions.

Service 0	Config Group 🕆	Created • Any •	Author 0	Notes 0 Any
W4 HDFS	HDFS Default Current	Thu, Nov 06, 2014 08:10	admin	Created from service config version V2
Va HDFS	HDFS Default	Thu, Nov 06, 2014 08:09	admin	adjusted NN heapsize
V7 ZooKeeper	deleted	Wed, Nov 05, 2014 19:02	admin	Created from service config version V4
vs ZooKeeper	ZooKeeper Default Current	Wed, Nov 05, 2014 19:01	admin	Created from service config version V1
VS ZooKeeper	deleted	Wed, Nov 05, 2014 19:00	admin	No notes
V4 ZooKeeper	deleted	Wed, Nov 05, 2014 19:00	admin	No notes
va ZooKeeper	ZooKeeper Default	Wed, Nov 05, 2014 19:00	admin	No notes
v2 ZooKeeper	ZooKeeper Default	Wed, Nov 05, 2014 19:00	admin	sync time
VI Pig	Pig Default Current	Wed, Nov 05, 2014 17:21	admin	Initial configurations for Pig
VI Hive	Hive Default Gurrent	Wed, Nov 05, 2014 17:21	admin	Initial configurations for Hive
18 of 18 versions showing	g - clear fibers			Show: 10 🔹 1 - 10 of 18 🔶 🔶

The most recent configuration changes are shown on the Service > Configs tab. Users can navigate the version scrollbar left-right to see earlier versions. This provides a quick way to access the most recent changes to a service configuration.

ZooKeeper Defau	It (1) Manage Config Groups	Filter
16 hours ag Current	dmin 16 hours ago 18 hours ago 19 hours ago	
:- V6 Current adr	nin authored on Wed, Nov 05, 2014 19:01	Discard Sav
ZooKeeper Server		
ooKeeper Server hosts	c6401.ambari.apache.org	
ooKeeper directory	/hadoop/zookeeper	•
ength of single Tick	2000 ms	
cks to allow for sync at it	10	
cks to allow for sync at untime	5	
ort for running ZK	2181	
arvar		
Advanced zookeeper-e		

Click on any version in the scrollbar to view, and hover to display an option menu which allows you compare versions and perform a revert. Performing a revert makes any config version that you select the current version.



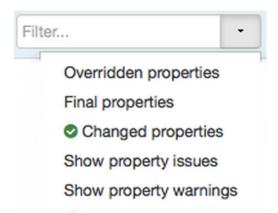
6.5.5. Comparing Versions

When navigating the version scroll area on the Services > Configs tab, you can hover over a version to display options to view, compare or revert.

ummary Configs		s	ervice Actions
ZooKeeper Defau	tt (1) • Manage Config Groups	Filter	•
V6 a 16 hours age Current	dmin V3 admin 16 hours ago 19 hours ago 19 hours ago		
Comparing V6	V2 admin authored on Wed, Nov 05, 2014 19:00		lake V2 Current
⊄ • V6 Current adm	nin authored on Wed, Nov 05, 2014 19:01	Dis	card Save
ZooKeeper Server			
icks to allow for sync at untime	٥	V6 Current	
unume	6	V2	

To perform a compare between two service configuration versions:

- 1. Navigate to a specific configuration version. For example "V6".
- 2. Using the version scrollbar, find the version would you like to compare against "V6". For example, if you want to compare V6 to V2, find V2 in the scrollbar.
- 3. Hover over the version to display the option menu. Click "Compare".
- 4. Ambari displays a comparison of V6 to V2, with an option to revert to V2.
- 5. Ambari also filters the display by only "Changed properties". This option is available under the Filter control.



6.5.6. Reverting a Change

You can revert to an older service configuration version by using the "Make Current" feature. The "Make Current" will actually create a new service configuration version with the configuration properties from the version you are reverting – it is effectively a "clone". After initiating the Make Current operation, you are prompted to enter notes for the new version (i.e. the clone) and save. The notes text will include text about the version being cloned.

Mal	ke Current	Confirmatio	n		Х
Notes Created from service config version V2				h	
		Ca	ncel	Discard	Make Current

There are multiple methods to revert to a previous configuration version:

• View a specific version and click the "Make V* Current" button.

Make V2 Current

• Use the version navigation dropdown and click the "Make Current" button.

			_			
V4	Thu, Nov 06, 2014 08:10	admin	•			
	Thu, Nov 06, 2014 06:09	edmin	Þ	V3		HDFS Default
V2	Wed, Nov 05, 2014 17:21	admin		admin authored on Thu, Nov 06, 2014 08:09 adjusted NN heapsize		
V1	Wed, Nov 05, 2014 16:00	admin		adjusted Nr	< neapsize	
			_	Q View	(P) Compare	Make Current

• Hover on a version in the version scrollbar and click the "Make Current" button.



• Perform a comparison and click the "Make V* Current" button.



6.5.7. Versioning and Host Config Groups

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. Same with custom config groups.

The following example describes a flow where you have multiple host config groups and create service configuration versions in each config group.



			nd save the change. Ambari increments the service compare and revert between these versions.
Default st Config Group lost1,host2,host3)	V1 heapsize=256	V2 heapsize=51:	
	Version cor ava	mpare + revert illable	
3 host gr	oup, override the	e heapsize to b	up and place host3 into the group. For this particular to 768, creating version V3. The host group (i.e. host3) nfig group and heapsize override from custom group.
Default st Config Group (host1,host2)	V1	V2	
(·····)	heapsize=256	heapsize=51	2
Custom st Config Group (host3)			V3 heapsize=768
			radjusting the heapsize to be 1024, creating version een these versions.
4 V4. USB	is call compare a	ind revent betw	
Default st Config Group (host1,host2)	V1 heapsize=256	V2 heapsize=512	
Custom at Config Group			V3 V4
			V3 V4 hespilar-748 hespilar-1024 Version compare + revert available
tt Config Group (host3)	he default host ars can compare a	config group by and revert betw	heapsize=768 heapsize=1024
t Config Group (host3) Modify t V5. Use Default st Config Group	he default host of rs can compare i	config group by and revert betw	bespilar-746 heapsilar-1024
ti Config Group (host3) Modify ti V5. Use	V1 heapsize=256	V2 heapsize=512	hespsize-78 hespsize-1024 Version compare + revert available adjusting the heapsize to be 256, creating version reven these versions inside of their config group.
Modify t (host3) Modify t V5. Use Default at Config Group	V1 heapsize=256	V2 heapsize=512	Adjusting the heapsize to be 256, creating version event with the original of the transmission of the tran

7. Administering the Cluster

From the cluster dashboard, use the Admin options to view information about Managing Stack and Versions, Service Accounts, and to Enable Kerberos security.



Note

For more information about administering your Ambari Server, see the Ambari Administration Guide.

7.1. Managing Stack and Versions

The **Stack** section includes information about the Services installed and available in the cluster Stack. Browse the list of Services and click **Add Service** to start the wizard to install Services into your cluster.

The **Versions** section shows what version of software is currently running and installed in the cluster. This section also exposes the capability to perform an automated cluster upgrade for maintenance and patch releases for the Stack. This capability is available for HDP 2.2 Stack only. If you have a cluster running HDP 2.2, you can perform Stack upgrades to later maintenance and patch releases. For example: you can upgrade from the GA release of HDP 2.2 (which is HDP 2.2.0.0) to the first maintenance release of HDP 2.2 (which is HDP 2.2.4.2).



Note

For more details on upgrading from HDP 2.2.0.0 to the latest HDP 2.2 maintenance release, see the Ambari Upgrade Guide.

The process for managing versions and performing an upgrade is comprised of three main steps:

- 1. Register a Version into Ambari
- 2. Install the Version into the Cluster
- 3. Perform Upgrade to the New Version

7.2. Register a Version

Ambari can manage multiple versions of Stack software.

To register a new version:

- 1. On the Versions tab, click Manage Versions.
- 2. Proceed to register a new version by clicking + Register Version.
- 3. Enter a two-digit version number. For example, enter **4.2**, (which makes the version HDP-2.2.4.2).

Versions / Register Version						
Details						
Name	HDP-2.2 \$					

- 4. Select one or more OS families and enter the respective Base URLs.
- 5. Click Save.
- 6. You can click "Install On..." or you can browse back to Admin > Stack and Versions > Versions tab. You will see the version current running and the version you just registered. Proceed to Install the Version.

7.3. Install the Version

To install a version in the cluster:

1. On the versions tab, click Install Packages.

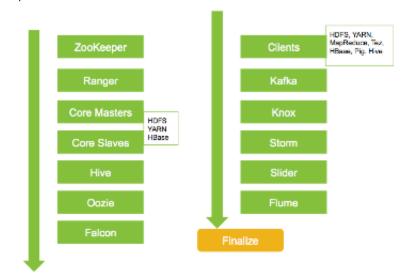


- 2. Click OK to confirm.
- 3. The Install version operation will start and the new version will be installed on all hosts.
- 4. You can browse to Hosts and to each Host > Versions tab to see the new version is installed. Proceed to Perform Upgrade.

7.4. Perform Upgrade

Once your target version has been registered into Ambari, installed on all hosts in the cluster and you meet the Prerequisites you are ready to perform an upgrade.

The perform upgrade process switches over the services in the cluster to a new version in a rolling fashion. The process follows the flow below. Starting with ZooKeeper and the Core Master components, ending with a Finalize step. To ensure the process runs smoothly, this process includes some manual prompts for you to perform cluster verification and testing along the way. You will be prompted when your input is required.





Important

This process can take some time to complete. You should validate the upgrade process in a dev/test environment prior to performing in production, as well, plan a block of time to monitor the progress. And as always, be sure to perform backups of your service metadata (you will be prompted during the first-stages of the upgrade process).

7.4.1. Upgrade Prerequisites

To perform an automated cluster upgrade from Ambari, your cluster must meet the following prerequisites:

Item	Requirement	Description
Cluster	Stack Version	Must be running HDP 2.2 Stack. This capability is not available for HDP 2.0 or 2.1 Stacks.
Version	New Version	All hosts must have the new version installed.
HDFS	NameNode HA	NameNode HA must be enabled and working properly. See Configuring NameNode High Availability in the Ambari User's Guide for more information.
HDFS	Decommission	No components should be in Decommissioning or Decommissioned state.
YARN	YARN WPR	Work Preserving Restart must be configured.
Hosts	Heartbeats	All Ambari Agents must be heartbeating to Ambari Server. Any hosts that are not heartbeating must be in Maintenance Mode.
Hosts	Maintenance Mode	Any hosts in Maintenance Mode must not be hosting any Service master components.
Services	Services Started	All Services must be started.
Services	Maintenance Mode	No Services can be in Maintenance Mode.

To perform an upgrade to a new version.

- 1. On the versions tab, click Perform Upgrade on the new version.
- 2. Follow the steps on the wizard.

7.5. Service Accounts

To view the list of users and groups used by the cluster services, choose Admin > Service Accounts.

i Ambari My2i	OCluster O ops 0 alorts Dashboard	d Services Hosts	Alerts Admin III 🔺 ad	ni
Stack and Versions	Service Users and Groups		Stack and Versions Service Accounts	ł
Service Accounts	Name	Value	Kerberos	1
Kerberos	Ams User	ams		1
	Smoke Test User	ambari-qa		
	Hadoop Group	hadoop		
	Falcon User	falcon		
	Flume User	flume		
	HDFS User	hdfs		
	Proxyuser Group	users		
	HBase User	hbase		
	HCat Client User	hcat		
	Hive User	hive		
	WebHCat User	hcat		
	Kafka User	kafka		
	Knox Group	knox		
	Knox User	knox		
	MapReduce User	mapred		
	Oozie User	oozie		
	Sqoop User	sqoop		
	Storm User	storm		
	Tez User	tez		
	YARN User	yarn		

7.6. Kerberos

If Kerberos has not been enabled in your cluster, click the Enable Kerberos button to launch the Kerberos wizard. For more information on configuring Kerberos in your cluster, see the Ambari Security Guide. Once Kerberos is enabled, you can:

- Regenerate Keytabs
- Disable Kerberos

7.6.1. How To Regenerate Keytabs

- 1. Browse to Admin > Kerberos.
- 2. Click the Regenerate Kerberos button.
- 3. Confirm your selection to proceed.
- 4. Optionally, you can regenerate keytabs for only those hosts that are missing keytabs. For example, hosts that were not online/available from Ambari when enabling Kerberos.
- 5. Once you confirm, Ambari will connect to the KDC and regenerate the keytabs for the Service and Ambari principals in the cluster.
- 6. Once complete, you must restart all services for the new keytabs to be used.



Note

Ambari requires the Kerberos Admin credentials in order to regenerate the keytabs. If the credentials are not available to Ambari, you will be prompted to enter the KDC Admin username and password. For more information on configuring Kerberos in your cluster, see the Ambari Security Guide.

7.6.2. How To Disable Kerberos

- 1. Browse to Admin > Kerberos.
- 2. Click the Disable Kerberos button.
- 3. Confirm your selection to proceed. Cluster services will be stopped and the Ambari Kerberos security settings will be reset.
- 4. To re-enable Kerberos, click Enable Kerberos and follow the wizard steps. For more information on configuring Kerberos in your cluster, see the Ambari Security Guide.

8. Monitoring and Alerts

Ambari monitors cluster health and can alert you in the case of certain situations to help you identify and troubleshoot problems. You manage how alerts are organized, under which conditions notifications are sent, and by which method. This section provides information on:

- Managing Alerts
- Configuring Notifications
- List of Predefined Alerts

8.1. Managing Alerts

Ambari predefines a set of alerts that monitor the cluster components and hosts. Each alert is defined by an **Alert Definition**, which specifies the checking interval and thresholds (which are dependent on the **Alert Type**). When a cluster is created or modified, Ambari reads the Alert Definitions and creates **Alert Instances** for the specific components to watch.

8.1.1. Terms and Definitions

The following basic terms help describe the key concepts associated with Ambari Alerts:

Term	Definition	
Alert Definition	Defines the alert including the description, check interval, type and thresholds.	
Туре	The type of alert, such as PORT or METRIC.	
State	Indicates the state of an alert definition. Enabled or disabled. When disabled, no alert instances are created.	
Alert Instance	tance Represents the specific alert instances based on an alert definition. For example, the alert definition for DataNode process will have an alert instance per DataNode in the cluster.	
Status	An alert instance status is defined by severity. The most common severity levels are OK, WARN, CRIT but there are also severities for UNKNOWN and NONE. See "Alert Instances" for more information.	
Threshold	The thresholds assigned to each status.	
Alert Group	Grouping of alert definitions, useful for handling notifications targets.	
Notification	A notification target for when an alert instance status changes. Methods of notification include EMAIL and SNMP.	

Terminology

8.1.2. Alert Definitions and Instances

An Alert Definition includes name, description and check interval, as well as configurable thresholds for each status (depending on the Alert Type).

The following table lists the types of alerts, their possible status and if the thresholds are configurable:

Alert Types

Туре	Description	Status	Thresholds Configurable	Units
PORT	Watches a port based on a configuration property as the URI. Example: Hive Metastore Process	OK, WARN, CRIT	Yes	seconds
METRIC	Watches a metric based on a configuration property. Example: ResourceManager RPC Latency	OK, WARN, CRIT	Yes	variable
AGGREGATE	Aggregate of status for another alert definition. Example: percentage NodeManagers Available	OK, WARN, CRIT	Yes	percentage
WEB	Watches a Web UI and adjusts status based on response. Example: App Timeline Web UI	OK, WARN, CRIT	No	n/a
SCRIPT	Uses a custom script to handle checking. Example: NodeManager Health Summary	OK, CRIT	No	n/a

8.1.3. How To Change an Alert

- 1. Browse to the Alerts section in Ambari Web.
- 2. Find the alert definition to modify and click to view the definition details.
- 3. Click to Edit the description, check interval or thresholds.
- 4. Changes will take effect on all alert instances at the next interval check.

8.1.4. How To View a List of Alert Instances

- 1. Browse to the Alerts section in Ambari Web.
- 2. Find the alert definition and click to view the definition details.
- 3. The list of alert instances is shown.
- 4. Alternatively, you can browse to a specific host via the Hosts section of Ambari Web to view the list of alert instances specific to that host.

8.1.5. How To Enable or Disable an Alert

- 1. Browse to the Alerts section in Ambari Web.
- 2. Find the alert definition. Click to enable/disable.
- 3. Alternatively, you can click to view the definition details and click to enable/disable.
- 4. When disabled, not alert instances are in effect, therefore no alerts will be reported or dispatched for the alert definition.

8.2. Configuring Notifications

With Alert Groups and Notifications, you can create groups of alerts and setup notification targets for each group. This way, you can notify different parties interested in certain sets of alerts via different methods. For example, you might want your Hadoop Operations team to receive all alerts via EMAIL, regardless of status. And at the same time, have your System Administration team receive all RPC and CPU related alerts that are Critical only via SNMP. To achieve this scenario, you would have an Alert Notification that handles Email

for all alert groups for all severity levels, and you would have a different Alert Notification group that handles SNMP on critical severity for an Alert Group that contains the RPC and CPU alerts.

Ambari defines a set of default Alert Groups for each service installed in the cluster. For example, you will see a group for HDFS Default. These groups cannot be deleted and the alerts in these groups are not modifiable. If you choose not to use these groups, just do not set a notification target for them.

1. Creating or Editing Notifications

- 2. Browse to the Alerts section in Ambari Web.
- 3. Under the Actions menu, click Manage Notifications.
- 4. The list of existing notifications is shown.
- 5. Click + to "Create new Alert Notification". The Create Alert Notification is displayed.
- 6. Enter the notification name, select that groups the notification should be assigned to (all or a specific set), select the Severity levels that this notification responds to, include a description, and choose the method for notification (EMAIL or SNMP).
 - For EMAIL: you will need to provide information about your SMTP infrastructure such as SMTP Server, Port, To/From address and if authentication is required to relay messages through the server. You can add custom properties to the SMTP configuration based on the Javamail SMTP options.
 - For SNMP: you will need to select the SNMP version, OIDs, community and port.
- 7. After completing the notification, click Save.
- 1. Creating or Editing Alert Groups
- 2. Browse to the Alerts section in Ambari Web.
- 3. From the Actions menu, choose Manage Alert Groups
- 4. The list of existing groups (default and custom) is shown.
- 5. Choose + to "Create Alert Group". Enter the Group a name and click Save.
- 6. By clicking on the custom group in the list, you can add or delete alert definitions from this group, and change the notification targets for the group.

8.3. List of Predefined Alerts

- HDFS Service Alerts
- NameNode HA Alerts
- YARN Alerts
- MapReduce2 Alerts

- HBase Service Alerts
- Hive Alerts
- Oozie Alerts
- ZooKeeper Alerts
- Ambari Alerts

8.3.1. HDFS Service Alerts

Description	Potential Causes	Possible Remedies
This service-level alert is triggered if the number of corrupt or missing blocks exceeds the configured critical threshold.	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.	For critical data, use a replication of 3. Bring up the failed DataNodes w missing or corrupt blocks. Identify the files associated with missing or corrupt blocks by runr Hadoop fsck command. Delete the corrupt files and recor from backup, if it exists.
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.	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).	Check for any errors in the logs (, log/hadoop/hdfs/)and restart th NameNode host/process using th Manage Services tab. Run the netstat-tuplpn command check if the NameNode process is to the correct network port.
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.	Cluster storage is full. If cluster storage is not full, DataNode is full.	If cluster still has storage, use Bal distribute the data to relatively le datanodes. If the cluster is full, delete unnece data or add additional storage by either more DataNodes or more disks to the DataNodes. After ad more storage run Balancer.
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.	DataNode process is down or not responding. DataNode are not down but is not listening to the correct network port/ address.	Check for dead DataNodes in Am Web. Check for any errors in the DataN logs (/var/log/hadoop/hdfs) and the DataNode, if necessary. Run the netstat-tuplpn command check if the DataNode process is to the correct network port.
This host-level alert is triggered if the DataNode Web UI is unreachable.	The DataNode process is not running.	Check whether the DataNode pro running.
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 only available if you are running JDK 1.7.	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.	Use the top command to determ which processes are consuming e CPU. Reset the offending 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. 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. 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.	This service-level alert is triggered if the number of corrupt or missing blocks exceeds the configured critical threshold.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.The NameNode screated because the only replica of the block is missing.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.The NameNode process is down on the HDFS master host.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.Cluster storage is full. the storage is not full, DataNode is full.This host-level alert is triggered if the nurividual DataNode processes cannot be established to be up and listening on the network for the configured critical threshold, given in seconds.DataNode process is down or not responding.This host-level alert is triggered if the pataNode web UI is unreachable.DataNode process is not running.This host-level alert is triggered if CPU utilization of the NameNode exceeds certain thresholds (200% warning, 250% critical). It checks the NameNode gMX Servlet for the System(CPULoad property.The DataNode process is not running.This host-level alert is triggered if CPU utilization of the NameNode exceeds certain thresholde (200% warning, 250% critical). It checks the NameNode JMXUnusually high CPU utilization: Can be caused by a very unusual job/query

Alert	Description	Potential Causes	Possible Remedies
NameNode Web UI	This host-level alert is triggered if the NameNode Web UI is unreachable.	The NameNode process is not running.	Check whether the NameNode p running.
Percent DataNodes with Available Space	This service-level alert is triggered if the storage if full on a certain percentage of DataNodes (10% warn, 30% critical). It aggregates the result from the check_datanode_storage.php plug-in.	Cluster storage is full. If cluster storage is not full, DataNode is full.	If cluster still has storage, use Bal distribute the data to relatively le DataNodes. If the cluster is full, delete unnece data or add additional storage by either more DataNodes or more disks to the DataNodes. After ad more storage run Balancer.
Percent DataNodesAvailable	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 plug-in to aggregate the results of Data node process checks.	DataNodes are down DataNodes are not down but are not listening to the correct network port/ address.	Check for dead DataNodes in Am Web. Check for any errors in the DataN logs (/var/log/hadoop/hdfs) and the DataNode hosts/processes. Run the netstat-tuplpn command check if the DataNode process is to the correct network port.
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.	A job or an application is performing too many NameNode operations.	Review the job or the application potential bugs causing it to performany NameNode operations.
NameNode Last Checkpoint	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.	Too much time elapsed since last NameNode checkpoint. Uncommitted transactions beyond threshold.	Set NameNode checkpoint. Review threshold for uncommitte transactions.
Secondary NameNode Process	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.	The Secondary NameNode is not running.	Check that the Secondary DataNo process is running.
NameNode Directory Status	This alert checks if the NameNode NameDirStatus metric reports a failed directory.	One or more of the directories are reporting as not healthy.	Check the NameNode UI for info about unhealthy directories.
HDFS capacity utilization	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.	Cluster storage is full.	Delete unnecessary data. Archive unused data. Add more DataNodes. Add more or larger disks to the DataNodes. After adding more storage, run B
DataNode Health Summary	This service-level alert is triggered if there are unhealthy DataNodes.	A DataNode is in an unhealthy state.	Check the NameNode UI for the dead DataNodes.

8.3.2. NameNode HA Alerts

Alert	Description	Potential Causes	Possible Remedies
•	55	The JournalNode process is down or not responding.	Check if the JournalNode process

Alert	Description	Potential Causes	Possible Remedies
	the network for the configured critical threshold, given in seconds.	The JournalNode is not down but is not listening to the correct network port/ address.	
NameNode High Availability Health	This service-level alert is triggered if either the Active NameNode or Standby NameNode are not running.	The Active, Standby or both NameNode processes are down.	On each host running NameNode check for any errors in the logs (/ log/hadoop/hdfs/) and restart th NameNode host/process using A Web. On each host running NameNode the netstat-tuplpn command to o the NameNode process is bound correct network port.
ZooKeeper Failover Controller process	This alert is triggered if the ZooKeeper Failover Controller process cannot be confirmed to be up and listening on the network.	The ZKFC process is down or not responding.	Check if the ZKFC process is runn

8.3.3. YARN Alerts

Alert	Description	Potential Causes	Possible Remedies
Percent NodeManagersAvai	This alert is triggered if the number of abde /n NodeManagers in the cluster is greater than the configured critical threshold. It aggregates the results of DataNode process alert checks.	NodeManagers are down. NodeManagers are not down but are not listening to the correct network port/ address.	Check for dead NodeManagers. Check for any errors in the Node logs (/var/log/hadoop/yarn) and the NodeManagers hosts/proces necessary. Run the netstat-tuplpn command check if the NodeManager proces bound to the correct network po
ResourceManager Web UI	This host-level alert is triggered if the ResourceManager Web UI is unreachable.	The ResourceManager process is not running.	Check if the ResourceManager pr running.
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.	A job or an application is performing too many ResourceManager operations.	Review the job or the applicatior potential bugs causing it to perfo many ResourceManager operatio
ResourceManager CPU utilization	This host-level alert is triggered if CPU utilization of the ResourceManager exceeds certain thresholds (200% warning, 250% critical). It checks the ResourceManager JMX Servlet for the SystemCPULoad property. This information is only available if you are running JDK 1.7.	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.	Use the top command to determ which processes are consuming e CPU. Reset the offending process.
NodeManager Web UI	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.	NodeManager process is down or not responding. NodeManager is not down but is not listening to the correct network port/ address.	Check if the NodeManager is run Check for any errors in the Node logs (/var/log/hadoop/yarn) and the NodeManager, if necessary.
NodeManager health	This host-level alert checks the node health property available from the NodeManager component.	Node Health Check script reports issues or is not configured.	Check in the NodeManager logs log/hadoop/yarn) for health che and restart the NodeManager, a restart if necessary.

Alert	Description	Potential Causes	Possible Remedies
			Check in the ResourceManager U var/log/hadoop/yarn) for health errors.

8.3.4. MapReduce2 Alerts

Alert	Description	Potential Causes	Possible Remedies
HistoryServer Web UI	This host-level alert is triggered if the HistoryServer Web UI is unreachable.	The HistoryServer process is not running.	Check if the HistoryServer proces running.
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.	A job or an application is performing too many HistoryServer operations.	Review the job or the application potential bugs causing it to perfor many HistoryServer operations.
HistoryServer CPU utilization	This host-level alert is triggered if the percent of CPU utilization on the HistoryServer exceeds the configured critical threshold.	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.	Use the top command to determ which processes are consuming e CPU. Reset the offending process.
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.	HistoryServer process is down or not responding. HistoryServer is not down but is not listening to the correct network port/ address.	Check the HistoryServer is runnin Check for any errors in the Histor logs (/var/log/hadoop/mapred) restart the HistoryServer, if neces

8.3.5. HBase Service Alerts

Alert	Description	Potential Causes	Possible Remedies
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 aggregates the results of RegionServer process down checks.	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 Zookeeper.	Check the dependent services to sure they are operating correctly Look at the RegionServer log file (usually /var/log/hbase/*.log) fo information. If the failure was associated with particular workload, try to under the workload better. Restart the RegionServers.
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.	The HBase master process is down. The HBase master has shut itself down because there were problems in the dependent services, ZooKeeper or HDFS.	Check the dependent services. Look at the master log files (usua log/hbase/*.log) for further info Look at the configuration files (/ hbase/conf). Restart the master.
HBase Master Web UI	This host-level alert is triggered if the HBase Master Web UI is unreachable.	The HBase Master process is not running.	Check if the Master process is rur
HBase Master CPU utilization	This host-level alert is triggered if CPU utilization of the HBase Master exceeds certain thresholds (200% warning, 250%	Unusually high CPU utilization: Can be caused by a very unusual job/query	Use the top command to determ which processes are consuming e CPU

Alert	Description	Potential Causes	Possible Remedies
	critical). It checks the HBase Master JMX Servlet for the SystemCPULoad property. This information is only available if you are running JDK 1.7.	workload, but this is generally the sign of an issue in the daemon.	Reset the offending process.
RegionServer process	This host-level alert is triggered if the RegionServer processes cannot be	The RegionServer process is down on the host.	Check for any errors in the logs (/ log/hbase/) and restart the Regio
	confirmed to be up and listening on the network for the configured critical threshold, given in seconds.	The RegionServer process is up and running but not listening on the correct network port (default 60030).	process using Ambari Web. Run the netstat-tuplpn command check if the RegionServer process bound to the correct network po

8.3.6. Hive Alerts

Alert	Description	Potential Causes	Possible Remedies
HiveServer2 Process	This host-level alert is triggered if the HiveServer cannot be determined to be up and responding to client requests.	HiveServer2 process is not running. HiveServer2 process is not responding.	Using Ambari Web, check status HiveServer2 component. Stop an restart.
HiveMetastore Process	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.	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.	Using Ambari Web, stop the Hive and then restart it.
WebHCat Server status	This host-level alert is triggered if the WebHCat server cannot be determined to be up and responding to client requests.	The WebHCat server is down. The WebHCat server is hung and not responding. The WebHCat server is not reachable over the network.	Restart the WebHCat server using Web.

8.3.7. Oozie Alerts

Alert	Description	Potential Causes	Possible Remedies
Oozie status	This host-level alert is triggered if the Oozie server cannot be determined to be up and responding to client requests.	The Oozie server is down. The Oozie server is hung and not responding. The Oozie server is not reachable over the network.	Restart the Oozie service using A Web.

8.3.8. ZooKeeper Alerts

Alert	Description	Potential Causes	Possible Remedies
Percent ZooKeeper ServersAvailable	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.	The majority of your ZooKeeper servers are down and not responding.	Check the dependent services to sure they are operating correctly Check the ZooKeeper logs (/var/ hadoop/zookeeper.log) for furth information. If the failure was associated with particular workload, try to under the workload better. Restart the ZooKeeper servers fro Ambari UI.

Alert	Description	Potential Causes	Possible Remedies
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.	The ZooKeeper server process is up and running but not listening on the correct	Check for any errors in the ZooK logs (/var/log/hbase/) and resta ZooKeeper process using Ambar Run the netstat-tuplpn comman check if the ZooKeeper server pr bound to the correct network po

8.3.9. Ambari Alerts

Alert	Description	Potential Causes	Possible Remedies
Ambari Agent Disk Usage	This host-level alert is triggered if the amount of disk space used on a host goes above specific thresholds. The default values are 50% for WARNING and 80% for CRITICAL.		Check logs and temporary directo items to remove. Add more disk space.