

Engineering Excellence J1

(Hadoop Cluster Setup – Enabling HA)

Enabling HA on Namenode and ResourceManger

TABLE OF CONTENT

[Engineering Excellence J1](#)

[NameNode HA Overview](#)

[Configure NameNode High Availability](#)

[Review Settings](#)

[Configure Components](#)

[Start Components](#)

[Finalize HA Setup](#)

[Resource HA Overview](#)

[Configure ResourceManger High Availability](#)

[Review Settings](#)

[Configure & Start Components](#)

Namenode High Availability

The HDFS NameNode High Availability feature enables you to run redundant NameNodes in the same cluster in an Active/Passive configuration with a hot standby. This eliminates the NameNode as a potential single point of failure (SPOF) in an HDFS cluster.

Formerly, if a cluster had a single NameNode, and that machine or process became unavailable, the entire cluster would be unavailable until the NameNode was either restarted or started on a separate machine. This situation impacted the total availability of the HDFS cluster in two major ways:

- In the case of an unplanned event such as a machine crash, the cluster would be unavailable until an operator restarted the NameNode.
- Planned maintenance events such as software or hardware upgrades on the NameNode machine would result in periods of cluster downtime.

HDFS NameNode HA avoids this by facilitating either a fast failover to the new NameNode during machine crash, or a graceful administrator-initiated failover during planned maintenance.

This guide provides an overview of the HDFS NameNode High Availability (HA) feature, instructions on how to deploy Hue with an HA cluster, and instructions on how to enable HA on top of an existing HDP cluster using the Quorum Journal Manager (QJM) and Zookeeper Failover Controller for configuration and management. Using the QJM and Zookeeper Failover Controller enables the sharing of edit logs between the Active and Standby NameNodes.

Note : This guide assumes that an existing HDP cluster has been manually installed and deployed. If your existing HDP cluster was installed using Ambari, configure NameNode HA using the Ambari wizard, as described in the [Ambari User's Guide](#)

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.

How to Configure NameNode High Availability

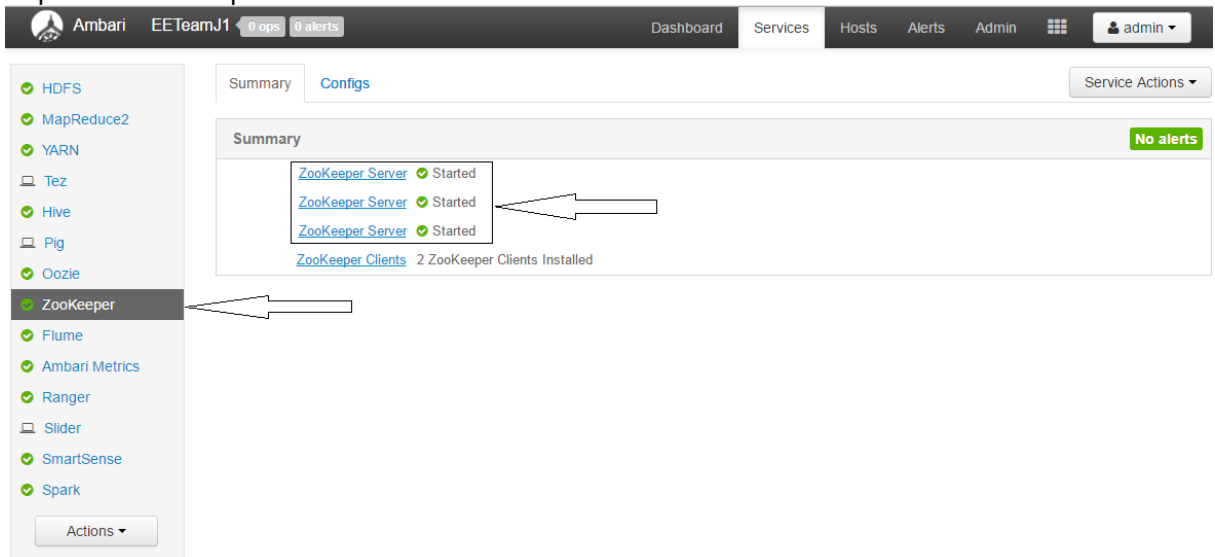
1. Check to make sure you have at least three hosts in your cluster and are running at least three ZooKeeper servers.

In our case we are running ZooKeeper servers on the following box.

impetus-i0161.impetus.co.in

impetus-I0163.impetus.co.in

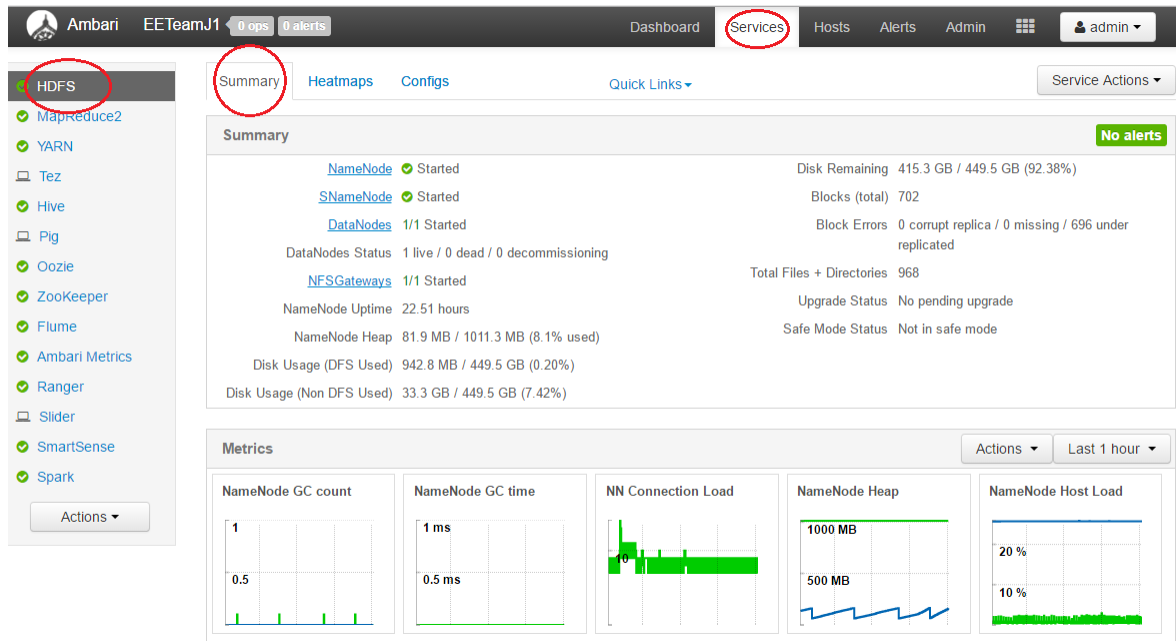
impetus-i0095.impetus.co.in



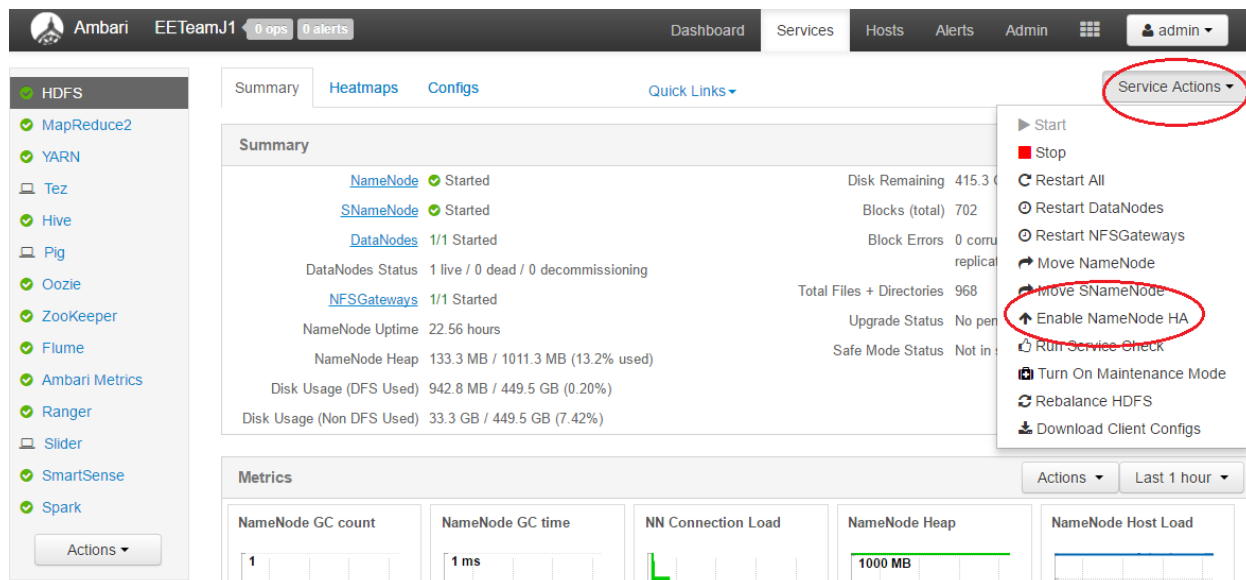
2. Check to make sure that the HDFS and ZooKeeper services are not in Maintenance Mode.

These services will be stopped and started when enabling NameNode HA. Maintenance Mode will prevent those start and stop operations from occurring. If the HDFS or ZooKeeper services are in Maintenance Mode the NameNode HA wizard will not complete successfully.

3. In Ambari Web, select Services > HDFS > Summary.



4. Select Service Actions and choose Enable **NameNode HA**.



5. The Enable HA Wizard launches. This wizard describes the set of automated and manual steps you must take to set up NameNode high availability.

6. **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.

Enable NameNode HA Wizard

ENABLE NAMEDNODE HA WIZARD

Get Started

Select Hosts

Review

Create Checkpoint

Configure Components

Initialize JournalNodes

Start Components

Initialize Metadata

Finalize HA Setup

Get Started

This wizard will walk you through enabling NameNode HA on your cluster. Once enabled, you will be running a Standby NameNode in addition to your Active NameNode. This allows for an Active-Standby NameNode configuration that automatically performs failover.

The process to enable HA involves a combination of **automated steps** (that will be handled by the wizard) and **manual steps** (that you must perform in sequence as instructed by the wizard).

You should plan a cluster maintenance window and prepare for cluster downtime when enabling NameNode HA.

If you have HBase running, please exit this wizard and stop HBase first.

Nameservice ID:

Nameservice ID
This will be the ID for the NameNode HA cluster. For example, if you set Nameservice ID to **mycluster**, the logical URI for HDFS will be **hdfs://mycluster**.

Next →

- 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.

Enable NameNode HA Wizard

ENABLE NAMEDNODE HA WIZARD

Get Started

Select Hosts

Review

Create Checkpoint

Configure Components

Initialize JournalNodes

Start Components

Initialize Metadata

Finalize HA Setup

Select Hosts

Select a host that will be running the additional NameNode. In addition, select the hosts to run JournalNodes, which store NameNode edit logs in a fault tolerant manner.

Current NameNode:

Additional NameNode:

JournalNode:

JournalNode:

JournalNode:

impetus-i0095.impetus.co.in (7.7 GB, 4 cores)
JournalNode NameNode
ZooKeeper Server Grafana
Spark History Server
SmartSense HST Server

impetus-i0161.impetus.co.in (7.7 GB, 4 cores)
JournalNode SNameNode
ResourceManager WebHCat Server
HiveServer2 Hive Metastore
Oozie Server ZooKeeper Server
Ranger Usersync Ranger Admin

impetus-i0163.impetus.co.in (7.7 GB, 4 cores)
NameNode History Server
App Timeline Server ZooKeeper Server
Metrics Collector

impetus-i0203.impetus.co.in (7.7 GB, 4 cores)
JournalNode

8. **Review:** Confirm your host selections and click Next.

Enable NameNode HA Wizard

ENABLE NAMENODE HA WIZARD

[Get Started](#)

[Select Hosts](#)

Review

[Create Checkpoint](#)

[Configure Components](#)

[Initialize JournalNodes](#)

[Start Components](#)

[Initialize Metadata](#)

[Finalize HA Setup](#)

Review

Confirm your host selections.

Current NameNode: impetus-i0095.impetus.co.in

Secondary NameNode: impetus-i0161.impetus.co.in **TO BE DELETED**

Additional NameNode: impetus-i0163.impetus.co.in **TO BE INSTALLED**

JournalNode:
impetus-i0161.impetus.co.in **TO BE INSTALLED**
impetus-i0095.impetus.co.in **TO BE INSTALLED**
impetus-i0203.impetus.co.in **TO BE INSTALLED**

Review Configuration Changes.
The following lists the configuration changes that will be made by the Wizard to enable NameNode HA. This information is for **review only** and is not editable except for the **dfs.journalnode.edits.dir** property

HDFS

dfs.journalnode.edits.dir

/hadoop/hdfs/journal

C

fs.defaultFS

hdfs://EETeamJ1

ha.zookeeper.quorum

impetus-i0095.impetus.co.in:2181,impetus-i0163.impetus.co.in

dfs.nameservices

EETeamJ1

dfs.ha.namenodes.
EETeamJ1

nn1,nn2

dfs.namenode.rpc-
address.EETeamJ1.nn1

impetus-i0095.impetus.co.in:8020

9. **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.

Enable NameNode HA Wizard

ENABLE NAMENODE HA WIZARD

Get Started

Select Hosts

Review

Create Checkpoint

Configure Components

Initialize JournalNodes

Start Components

Initialize Metadata

Finalize HA Setup

Manual Steps Required: Create Checkpoint on NameNode

1. Login to the NameNode host **impetus-i0095.impetus.co.in**.
2. Put the NameNode in Safe Mode (read-only mode):

```
sudo su hdfs -l -c 'hdfs dfsadmin -safemode enter'
```
3. Once in Safe Mode, create a Checkpoint:

```
sudo su hdfs -l -c 'hdfs dfsadmin -saveNamespace'
```
4. You will be able to proceed once Ambari detects that the NameNode is in Safe Mode and the Checkpoint has been created successfully.

If the **Next** button is enabled before you run the "**Step 4: Create a Checkpoint**" command, it means there is a recent Checkpoint already and you may proceed without running the "**Step 4: Create a Checkpoint**" command.

Checkpoint not created yet **Next →**

```
impetus-i0095:$ sudo su hdfs -l -c 'hdfs dfsadmin -safemode enter'
```

Safe mode is ON

```
impetus-i0095:~$ sudo su hdfs -l -c 'hdfs dfsadmin -saveNamespace'
```

Save namespace successful

Enable NameNode HA Wizard

X

ENABLE NAMENODE HA WIZARD

Get Started

Select Hosts

Review

Create Checkpoint

Configure Components

Initialize JournalNodes

Start Components

Initialize Metadata

Finalize HA Setup

Manual Steps Required: Create Checkpoint on NameNode

1. Login to the NameNode host **impetus-i0095.impetus.co.in**.
2. Put the NameNode in Safe Mode (read-only mode):

```
sudo su hdfs -l -c 'hdfs dfsadmin -safemode enter'
```
3. Once in Safe Mode, create a Checkpoint:

```
sudo su hdfs -l -c 'hdfs dfsadmin -saveNamespace'
```
4. You will be able to proceed once Ambari detects that the NameNode is in Safe Mode and the Checkpoint has been created successfully.

If the **Next** button is enabled before you run the "**Step 4: Create a Checkpoint**" command, it means there is a recent Checkpoint already and you may proceed without running the "**Step 4: Create a Checkpoint**" command.

Checkpoint created **Next →**

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

Enable NameNode HA Wizard

ENABLE NAMEDNODE HA WIZARD

Get Started

Select Hosts

Review

Create Checkpoint

Configure Components

Initialize JournalNodes

Start Components

Initialize Metadata

Finalize HA Setup

Configure Components

Please proceed to the next step.

✓ Stop All Services

✓ Install Additional NameNode

✓ Install JournalNodes

✓ Reconfigure HDFS

✓ Start JournalNodes

✓ Disable Secondary NameNode

Next

11. **Initialize 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**.

ENABLE NAMEDNODE HA WIZARD

Get Started

Select Hosts

Review

Create Checkpoint

Configure Components

Initialize JournalNodes

Start Components

Initialize Metadata

Finalize HA Setup

Manual Steps Required: Initialize JournalNodes

1. Login to the NameNode host **impetus-i0095.impetus.co.in**.

2. Initialize the JournalNodes by running:

```
sudo su hdfs -l -c 'hdfs namenode -initializeSharedEdits'
```

3. You will be able to proceed once Ambari detects that the JournalNodes have been initialized successfully.

JournalNodes not initialized yet

Next →

```
impetus-i0095:~$ sudo su hdfs -l -c 'hdfs namenode -initializeSharedEdits'
```

```
16/06/02 19:08:31 INFO namenode.NameNode: STARTUP_MSG:
/*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG:  host = impetus-i0095.impetus.co.in/172.26.60.19
STARTUP_MSG:  args = [-initializeSharedEdits]
STARTUP_MSG:  version = 2.7.1.2.4.2.0-258
epoch 1
```



```

16/06/02 19:08:32 INFO namenode.EditLogInputStream: Fast-forwarding stream
'/hadoop/hdfs/namenode/current/edits_0000000000000062342-0000000000000062342'
to transaction ID 62342
16/06/02 19:08:32 INFO namenode.FSEditLog: Starting log segment at 62342
16/06/02 19:08:32 INFO namenode.FSEditLog: Ending log segment 62342
16/06/02 19:08:32 INFO namenode.FSEditLog: Number of transactions: 1 Total time
for transactions(ms): 1 Number of transactions batched in Syncs: 0 Number of
syncs: 1 SyncTimes(ms): 29
16/06/02 19:08:32 INFO util.ExitUtil: Exiting with status 0
16/06/02 19:08:32 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at impetus-
i0095.impetus.co.in/172.26.60.19
*****/

```

Enable NameNode HA Wizard

ENABLE NAMENODE HA WIZARD

- Get Started
- Select Hosts
- Review
- Create Checkpoint
- Configure Components
- Initialize JournalNodes**
- Start Components
- Initialize Metadata
- Finalize HA Setup

Manual Steps Required: Initialize JournalNodes

1. Login to the NameNode host **impetus-i0095.impetus.co.in**.
2. Initialize the JournalNodes by running:

```
sudo su hdfs -l -c 'hdfs namenode -initializeSharedEdits'
```
3. You will be able to proceed once Ambari detects that the JournalNodes have been initialized successfully.

JournalNodes initialized
Next →

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

Enable NameNode HA Wizard

ENABLE NAMENODE HA WIZARD

- Get Started
- Select Hosts
- Review
- Create Checkpoint
- Configure Components
- Initialize JournalNodes
- Start Components**
- Initialize Metadata
- Finalize HA Setup

Start Components

Please proceed to the next step.

- ✓ Start ZooKeeper Servers
- ✓ Start NameNode

Next

13. **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.

1. Login to the NameNode host **impetus-i0095.impetus.co.in**.
2. Initialize the metadata for NameNode automatic failover by running:

```
sudo su hdfs -l -c 'hdfs zkfc -formatZK'
```
3. Login to the Additional NameNode host **impetus-i0163.impetus.co.in**.
Important! Be sure to login to the Additional NameNode host. This is a different host from the Steps 1 and 2 above.
4. Initialize the metadata for the Additional NameNode by running:

```
sudo su hdfs -l -c 'hdfs namenode -bootstrapStandby'
```

Enable NameNode HA Wizard X

ENABLE NAMENODE HA WIZARD

Get Started

Select Hosts

Review

Create Checkpoint

Configure Components

Initialize JournalNodes

Start Components

Initialize Metadata

Finalize HA Setup

Manual Steps Required: Initialize NameNode HA Metadata

1. Login to the NameNode host **impetus-i0095.impetus.co.in**.

2. Initialize the metadata for NameNode automatic failover by running:

```
sudo su hdfs -l -c 'hdfs zkfc -formatZK'
```

3. Login to the Additional NameNode host **impetus-i0163.impetus.co.in**.

Important! Be sure to login to the Additional NameNode host.
This is a different host from the Steps 1 and 2 above.

4. Initialize the metadata for the Additional NameNode by running:

```
sudo su hdfs -l -c 'hdfs namenode -bootstrapStandby'
```

Please proceed once you have completed the steps above.

Next →

14. **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.

Enable NameNode HA Wizard



ENABLE NAMENODE HA WIZARD

Get Started

Select Hosts

Review

Create Checkpoint

Configure Components

Initialize JournalNodes

Start Components

Initialize Metadata

Finalize HA Setup

Finalize HA Setup

You can click on the Retry button to retry failed tasks.

✔ Start Additional NameNode

✔ Install Failover Controllers

✔ Start Failover Controllers

✔ Delete Secondary NameNode

! Start All Services

Retry

Done

Enable NameNode HA Wizard



ENABLE NAMENODE HA WIZARD

Get Started

Select Hosts

Review

Create Checkpoint

Configure Components

Initialize JournalNodes

Start Components

Initialize Metadata

Finalize HA Setup

Finalize HA Setup

NameNode HA has been enabled successfully.

✔ Start Additional NameNode

✔ Install Failover Controllers

✔ Start Failover Controllers

✔ Delete Secondary NameNode

✔ Start All Services

Done

Ambari
 EETeamJ1
 0 ops
 0 alerts

Dashboard
 Services
 Hosts
 Alerts
 Admin

 admin

HDFS
 MapReduce2
 YARN
 Tez
 Hive
 Pig
 Oozie
 ZooKeeper
 Flume
 Ambari Metrics
 Ranger
 Slider
 SmartSense
 Spark

Summary
 Heatmaps
 Configs
 Quick Links
 Service Actions

Active NameNode

Started

ZKFailoverController

Started

Standby NameNode

Started

ZKFailoverController

Started

DataNodes

1/1 Started

DataNodes Status

1 live / 0 dead / 0 decommissioning

JournalNodes

3/3 JournalNodes Live

NFSGateways

1/1 Started

NameNode Uptime

29.90 mins

NameNode Heap

48.0 MB / 1011.3 MB (4.7% used)

Disk Usage (DFS Used)

943.4 MB / 449.5 GB (0.20%)

Disk Usage (Non DFS Used)

33.1 GB / 449.5 GB (7.37%)

Disk Remaining

415.5 GB / 449.5 GB (92.43%)

Blocks (total)

703

Block Errors

0 corrupt replica / 0 missing / 696 under replicated

Total Files + Directories

979

Upgrade Status

No pending upgrade

Safe Mode Status

Not in safe mode

No alerts

Metrics

Actions
 Last 1 hour

- Adjust the ZooKeeper Failover Controller retries setting for your environment.
Browse to Services > HDFS > Configs > core-site.
Set `ha.failover-controller.active-standby-elector.zk.op.retries=120`

HDFS

MapReduce2

YARN

Tez

Hive

Pig

Oozie

ZooKeeper

Flume

Ambari Metrics

Ranger

Slider

SmartSense

Spark

Actions

Summary

Heatmaps

Configs

Check Links

Service Actions

Group: Default (4) Manage Config Groups

Filter...

V5 admin about an hour ago HDP-2.4

V4 admin about a day ago HDP-2.4

V3 admin about a day ago HDP-2.4

V2 admin 2 days ago HDP-2.4

V1 admin 3 days ago HDP-2.4

admin authored on Thu, Jun 02, 2016 19:04

Discard

Save

Settings

Advanced

NameNode

NameNode hosts impetus-i0095.impetus.co.in and 1 other

NameNode new generation size 128 MB

NameNode maximum new generation size 128 MB

NameNode permanent generation size 128 MB

NameNode maximum permanent generation size 256 MB

Secondary NameNode

SNameNode host

NFS Gateway

NFSGateway host impetus-i0095.impetus.co.in

NFSGateway maximum Java heap size 1024 MB

NFSGateway dump directory /tmp/.hdfs-nfs

Allowed hosts * rw

Advanced core-site

fs.defaultFS hdfs://EETeamJ1

fs.trash.interval 360

ha.failover-controller active-standby-elector

zk.op.retries 120

hadoop.http.authentication.simple.anonymous.allowed true

hadoop.security.auth_to_local DEFAULT

ResourceManager High Availability

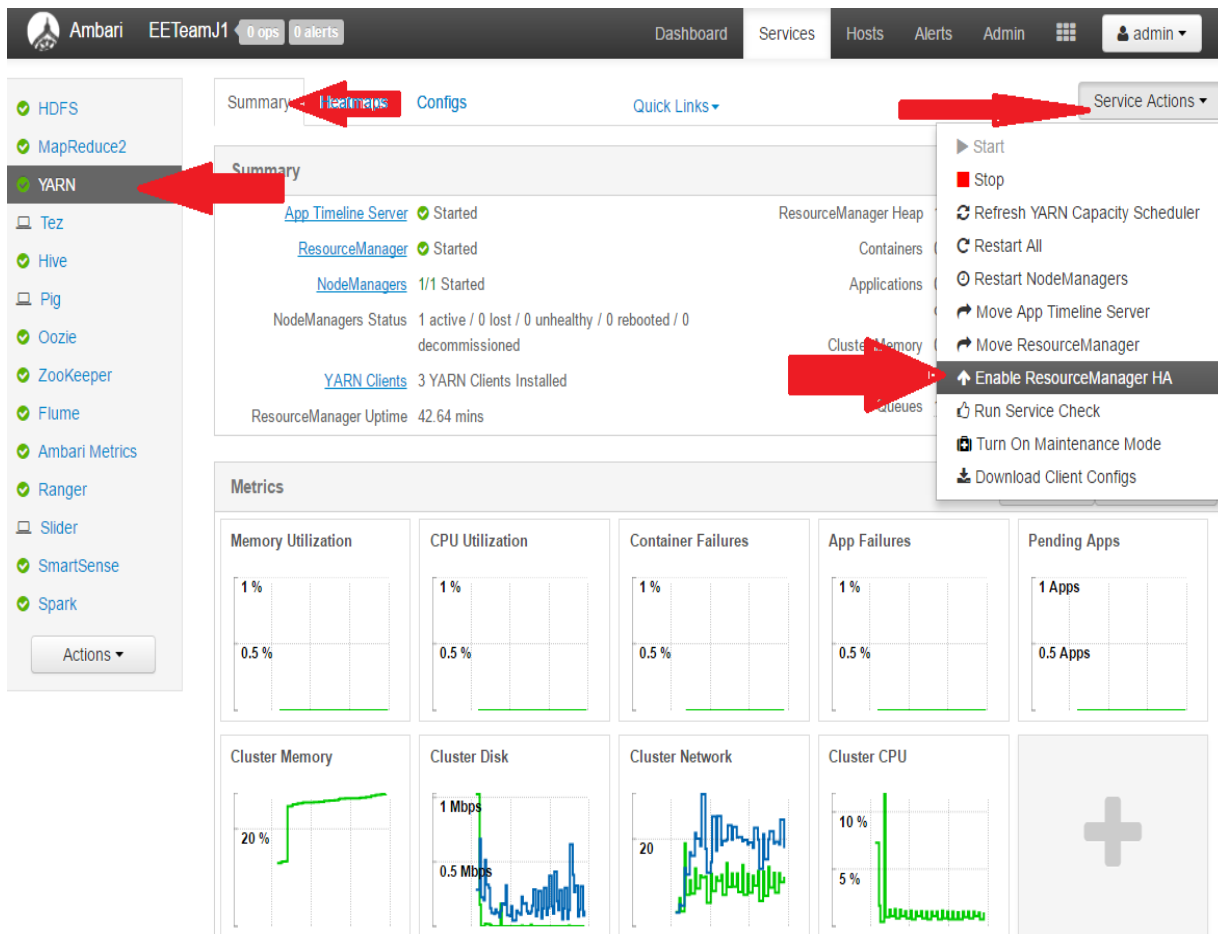
The ResourceManager (RM) is responsible for tracking the resources in a cluster, and scheduling applications (e.g., MapReduce jobs). Prior to Hadoop 2.4, the ResourceManager is the single point of failure in a YARN cluster. The High Availability feature adds redundancy in the form of an Active/Standby ResourceManager pair to remove this otherwise single point of failure.

Note: This feature is available with HDP Stack 2.2 or later.

The following topic explains How to Configure ResourceManager High Availability.

How to Configure ResourceManager 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, 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 Next to proceed.
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

ENABLE RESOURCEMANAGER
HA WIZARD

Get Started

Select Host

Review

Configure Components

Select Host

Select a host that will be running the additional ResourceManager

Current ResourceManager:

Additional ResourceManager:

impetus-i0095.impetus.co.in (7.7 GB, 4 cores)

NameNode

ZooKeeper Server

Spark History Server

Grafana

SmartSense HST Server

impetus-i0161.impetus.co.in (7.7 GB, 4 cores)

ResourceManager

Hive Metastore

HiveServer2

WebHCat Server

Oozie Server

ZooKeeper Server

Ranger Admin

Ranger Usersync

impetus-i0163.impetus.co.in (7.7 GB, 4 cores)

NameNode

History Server

App Timeline Server

ZooKeeper Server

Metrics Collector

impetus-i0203.impetus.co.in (7.7 GB, 4 cores)

ResourceManager

← Back

Next →

- 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.

Enable ResourceManager HA Wizard

ENABLE RESOURCEMANAGER
HA WIZARD

[Get Started](#)
[Select Host](#)
Review
[Configure Components](#)

Review

Confirm your host selections.

Current ResourceManager: impetus-i0161.impetus.co.in
Additional ResourceManager: impetus-i0203.impetus.co.in **+ TO BE INSTALLED**

Review Configuration Changes.
The following lists the configuration changes that will be made by the Wizard to enable ResourceManager HA.
This information is for **review only** and is not editable.

YARN

yarn.resourcemanager.
ha.enabled

☒

yarn.resourcemanager.
ha.rm-ids

rm1,rm2

yarn.resourcemanager.
hostname.rm1

impetus-i0161.impetus.co.in

yarn.resourcemanager.
webapp.address.rm1

impetus-i0161.impetus.co.in:8088

yarn.resourcemanager.
webapp.address.rm2

impetus-i0203.impetus.co.in:8088

yarn.resourcemanager.
webapp.https.address.
rm1

impetus-i0161.impetus.co.in:8090

yarn.resourcemanager.
webapp.https.address.
rm2

impetus-i0203.impetus.co.in:8090

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

Enable ResourceManager HA Wizard

ENABLE RESOURCEMANAGER
HA WIZARD

[Get Started](#)
[Select Host](#)
[Review](#)
Configure Components

Configure Components

ResourceManager HA has been enabled successfully.

✔ Stop Required Services

✔ Install Additional ResourceManager

✔ Reconfigure YARN

✔ Reconfigure HDFS

✔ Start All Services

Complete

✓ HDFS

✓ MapReduce2

✓ YARN

Tez

✓ Hive

Pig

✓ Oozie

✓ ZooKeeper

✓ Flume

✓ Ambari Metrics

✓ Ranger

Slider

✓ SmartSense

✓ Spark

Actions

SummaryHeatmapsConfigsQuick LinksService Actions

Summary

No alerts

App Timeline Server

Started

Standby ResourceManager

Started

Active ResourceManager

Started

NodeManagers

1/1 Started

NodeManagers Status

1 active / 0 lost / 0 unhealthy / 0 rebooted / 0 decommissioned

YARN Clients

3 YARN Clients Installed

ResourceManager Uptime

17.34 mins

ResourceManager Heap

173.8 MB / 910.5 MB (19.1% used)

Containers

0 allocated / 0 pending / 0 reserved

Applications

7 submitted / 0 running / 0 pending / 7 completed / 0 killed / 0 failed

Cluster Memory

0 Bytes used / 0 Bytes reserved / 5.0 GB available

Queues

1 Queues

Metrics

ActionsLast 1 hour

Memory Utilization

CPU Utilization

Container Failures

App Failures

Pending Apps