Hortonworks Data Platform

Quick Start Guide

1.0 (Nov 30, 2012)

Hortonworks Data Platform : Quick Start Guide

1.0 (2012-11-30)

Hortonworks Data Platform (HDP) is a 100% open source data management platform based on Apache Hadoop. It allows you to load, store, process and manage data in virtually any format and at any scale. As the foundation for the next generation enterprise data architecture, HDP includes all of the necessary components to begin uncovering business insights from the quickly growing streams of data flowing into and throughout your business – TODO - verify

Table of Contents

List of Figures

1. High Availability for Hadoop

This section provides information on the various components of the Apache Hadoop ecosystem.

HDPs Full-Stack HA Architecture

Hortonworks Data Platform (HDP) is an open source distribution powered by Apache Hadoop. HDP provides you with the actual Apache-released versions of the stack with all the necessary bug fixes to make all the components in the stack interoperable in your production environments. This stack uses multiple ‘master’ services whose failure would cause functionality outage in your Hadoop cluster. Hortonworks’ Full-Stack High Availability architecture provides a common framework to make all the master services resilient to failures.

HDP uses industry proven HA technologies in order to provide a reliable HA solution.

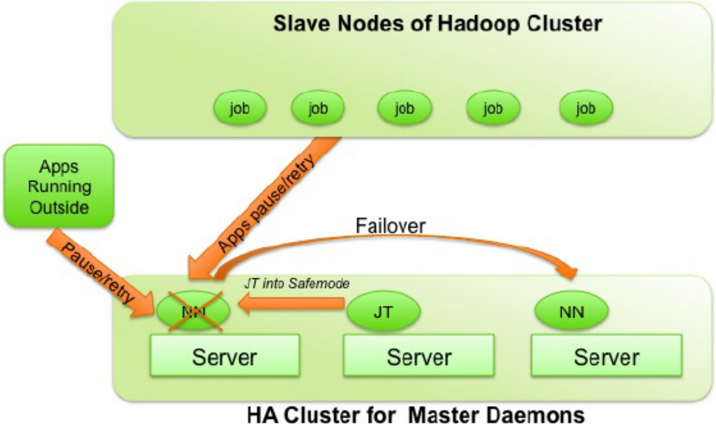
The Hadoop stack contains multiple services (HDFS, MapReduce, HBase, etc.) and each of these services have their own co-dependencies. A client application, that interacts with Hadoop, can depend on one or more of these services. A highly available Hadoop platform must ensure that the NameNode master service as well as client applications are resilient to critical failure services. Hortonworks’ Full-Stack HA architecture considers this global view.

Also see. Hortonworks blog on NameNode HA with Hadoop 1.0 The HDP HA architecture has the following key properties:

- It provides high availability for the NameNode master daemon service.
- When the NameNode master daemon fails over, the HA solution initiates the following actions:
    - Dependent services (like JobTracker) automatically detect the failure or fail over of the co-dependent component (NameNode) and these dependent services pause, retry, and recover the failed service. (For example, the JobTracker does not launch new jobs or kill jobs that have been waiting for the NameNode.)
    - Applications running inside and outside the Hadoop cluster also automatically pause and retry their connection to the failed service.

The above actions are highlighted in the following illustration. This illustration shows how HDFS clients and MapReduce services (Jobtracker daemon) handle the NameNode fail over.

Figure 1.1. FullStackHA.jpg



To configure High Availability for your Hadoop cluster see:

2. High Availability for HDFS NameNode Using VMware

This document is intended for system administrators for configuring the Hadoop NameNode (NN) service for High Availability and also for configuring clients and other Hadoop services to be resilient while the NameNode fails over.

The Hortonworks Data Platform (HDP) High Availability (HA) kit for VMware provides enterprise grade high availability of the NameNode Service as part of the HDP cluster.

To learn more, see the following: High Availability for Hadoop

In this document:

2.1. Use Cases and Fail Over Scenarios

In this section:

2.1.1. Use Cases

This solution enables the following Hadoop system administration use cases:

- Planned failure of the NameNode (for maintenance tasks like software or hardware upgrade)
- Unplanned failure of the NameNode

2.1.2. Fail over scenarios

The solution deals with the following faults:

- NameNode service failure
- NameNode JVM failure
- Hung NameNode daemon or hung operating system
- NameNode operating system failure
- Virtual machine failure
- ESXi host failure
- Failure of the NIC cards on ESXi hosts.
- Network failure between ESXi hosts.

Note

Some double faults are not handled (such as failure of multiple ESXi hosts).

2.2. Supported Operating Systems

The following operating systems are supported:

- Red Hat Enterprise Linux (RHEL) v5.7, v5.8, v6.2, v6.3 and later
- CentOS v5.7, v5.8, v6.2, v6.3 and later

2.3. Configuration For Physical Servers

Ensure that your setup meets the following requirements:

- The NameNode must run inside a virtual machine which is hosted on the vSphere HA cluster.
- The vSphere HA cluster must include a minimum of two ESXi server machines. See the NameNode hardware recommendations available (TODO - add xlink:href to h/w recommendation) here
- The vSphere HA cluster must use shared storage.

Note

Shared storage stores the state of the NameNode edit journal and checkpointed image. It provides ability to start the NameNode on another physical machine in the vSphere HA Cluster. Additionally, vSphere uses the shared state for the NameNode’s root VM disk.

2.4. Software Configuration

Complete the following tasks to configure NameNode HA solution:

TODO - add itemized list of subsections in this section

- Configure a vSphere HA cluster
- Install HDP
- Configure NameNode for automatic fail over
- Validate autostart configuration for NameNode and the NameNode monitoring agent

2.4.1. Configure a vSphere HA cluster

Complete the following configurations to configure your vSphere HA cluster:

TODO - add itemized list of subsections in this section

Configure vSphere cluster using vSphere Management Console

Configure shared storage

2.4.1.1. Configure vSphere cluster using vSphere Management Console

Use the vSphere Management Console to complete the following instructions. Also see the VMware vSphere documentation.:

Step 1: Create a vSphere HA cluster.

- In the vCenter Client, select Inventory -> Hosts and Clusters.
- In the left column, right-click the Datacenter and select New Cluster.

This step will create a vSphere HA cluster. (The vSphere HA cluster hosts the NameNode's VM.)

Step 2: Configure a VM to host the NameNode

- Within the vSphere HA cluster, create the virtual machine. This VM will host the NameNode process.
- Store the virtual machine disk (VMDK) on the data store located on the shared storage device.
- Specify the exact machine hardware configurations as required for configuring a physical server for the NameNode process. Use the NameNode hardware recommendations available {todo - add hardware reco link here} here.
- Start this virtual machine and configure the network settings. You should now have the IP address and FQDN for the virtual machine.
- The virtual machine should be present on the same network where the rest of the nodes of your Hadoop cluster are located.

2.4.1.2. Configure shared storage

- Shared storage is required to host both the NameNode data directory and the VM storage contents.
- Use standard enterprise practices (such as configuring the shared storage to be RAIDed) to ensure that your shared storage is reliable.

2.4.2. Install HDP

Use the following instructions to install HDP on your cluster hardware. Ensure that you specify the virtual machine (configured in the previous section) as your NameNode.

Step 1: Download Hortonworks Management Console (HMC) using the instructions provided here (TODO: Add xlink to HDP1-HMC doc-set).

Do not start the HMC server until you have configured the relevant templates as outlined in the following steps.

Step 2: Add the configuration parameters for Full-Stack HA.

1. Edit the`<master-install-machine-for-HDP>/etc/puppet/master/modules/hdp-hadoop/templates/hdfs-site.xml.erb` file to add the following properties:

- Enable the HDFS client retry policy.

```
<property>
 <name>dfs.client.retry.policy.enabled</name>
 <value>true</value>
 <description> Enables HDFS client retry in case of NameNode failure.</
description>
</property>
```

- Configure protection for NameNode edit log.

```
<property>
 <name>dfs.namenode.edits.toleration.length</name>
 <value>8192</value>
 <description> Prevents corruption of NameNode edit log.</description>
</property>
```

- Configure safe mode extension time.

```
<property>
 <name>dfs.safemode.extension</name>
 <value>10</value>
 <description> The default value (30 seconds) is applicable for very large
 clusters. For small to large clusters (upto 200 nodes), recommended value
 is 10 seconds.</description>
</property>
```

- Ensure that the allocated DFS blocks persist across multiple fail overs.

```
<property>
 <name>dfs.persist.blocks</name>
 <value>true</value>
 <description> Ensure that the allocated DFS blocks persist across
 multiple fail overs.</description>
</property>
```

- Configure delay for first block report.

```
<property>
 <name>dfs.blockreport.initialDelay</name>
 <value>10</value>
 <description> Delay (in seconds) for first block report. </description>
</property>
```

2. Edit the`<master-install-machine-for-HDP>/etc/puppet/master/modules/hdp-hadoop/templates/mapred-site.xml.erb` file to add the following properties:

- Enable the JobTracker’s safe mode functionality.

```
<property>
 <name>mapreduce.jt.hdfs.monitor.enable</name>
 <value>true</value>
 <description> Enable the JobTracker to go into safe mode when the
 NameNode is unresponsive.</description>
</property>
```

- Enable retry for JobTracker clients (when the JobTracker is in safe mode).

```
<property>
 <name>mapreduce.jobclient.retry.policy.enabled</name>
 <value>true</value>
 <description> Enable the MapReduce job client to retry job submission
 when the JobTracker is in safe
    mode.</description>
</property>
```

- Enable recovery of JobTracker’s queue after it is restarted.

```
<property>
 <name>mapred.jobtracker.restart.recover</name>
 <value>true</value>
 <description>  Enable the JobTracker to recover its queue after
 JobTracker is restarted.</description>
</property>
```

3. Edit the`<master-install-machine-for-HDP>/etc/puppet/master/modules/hdp-hadoop/templates/core-site.xml.erb` file to add the following properties:

   - Configure checkpoint interval so that the checkpoint is performed on an hourly basis.

```
<property>
 <name>fs.checkpoint.period</name>
 <value>3600</value>
 <description> The number of seconds between two periodic checkpoints.</
description>
</property>
```

Step 3: Complete HDP installation.

- Continue the HMC installation process using the instructions provided here. TODO add link here
- Ensure that you also follow the instructions listed below:
  - Use the fully qualified domain name (FQDN) of the virtual machine for configuring the host names (see: TODO add link here Prepare the Environment - Create hostdetail.txt). Note that HMC might not identify the NameNode VM automatically and it is therefore important to note the FQDN (IP address and DNS name) of the NameNode VM.
  - Specify shared storage for the NameNode’s directories (see: HMC - Step 6). TODO add link here
  - Do not use the NameNode VM for running any other master daemon.
- Complete the HMC installation. Ensure that the installation was successful.

Note

To modify the parameters in the template files after you have installed HDP, ensure that you follow the instructions listed below:

- Change the template files as required.

- Stop and start the respective service through the HMC console GUI.

  For example, stop and restart the MapReduce service if the `mapreduce-site.xml.erb` file is modified. Stop and restart HDFS service if either the `core-site.xml.erb` or the `hdfs-site.xml.erb` file is modified.

2.4.3. Configure NameNode for automatic fail over

Complete the following instructions to configure automatic fail over: TODO add itemized list for subheadings

2.4.3.1. Install and configure vSphere Monitoring Agent

The NameNode's monitoring agent monitors the NameNode daemon and notifies the vSphere if the NameNode daemon fails or becomes unstable. As soon as the vSphere receives the notification, it triggers HA solution to restart NameNode VM on either the same or a different ESXi host.

Follow the steps listed below to install and configure the monitoring agent:

Step 1: On the NameNode VM, download the HDP HA kit for VMware.

- Download the Hortonworks Data Platform High Availability Kit for VMware from here.
- Extract the following RPM files from the kit:
  - For RHEL/CentOS v5.x:

    ```
    hmonitor-vsphere-monitoring-1.1.0.15-1.el5.x86_64.rpm
    ```

  - For RHEL/CentOS v6.x:

    ```
    hmonitor-vsphere-monitoring-1.1.0.15-1.el6.x86_64.rpm
    ```

- Install the NameNode monitoring functionality:

  ```
  yum install hmonitor
  ```

  ```
  rpm -ivh hmonitor-vsphere-monitoring-1.1.0.15-1*
  ```

  ```
  yum install hmonitor-vsphere-namenode-daemon
  ```

Step 2: Configure the monitoring agent to point to the NameNode configurations.

- Edit the `/usr/lib/hadoop/monitor/vm-namenode.xml` file and provide the FQDN for your NameNode VM here:

  ```
  <property>
   <name>service.monitor.portprobe.host</name>
   <value>$NameNode_FQDN</value>
   <description> Port to probe via a telnet operation.</description>
  </property>
  ```

- Provide the HDFS port number.

```
<property>
 <name>service.monitor.portprobe.port</name>
 <value>$HDFS_filesystem_port</value>
 <description> Port to probe via a telnet operation.</description>
</property>
```

- Provide the FQDN and port information for the HDFS filesystem.

```
<property>
 <name>service.monitor.webprobe.url</name>
 <value>$http://NameNode_FQDN:NameNode_web_ui_port/</value>
 <description> URL to get.</description>
</property>
```

2.4.3.2. Configure the NameNode to start automatically on OS boot

The monitoring agent requires that the NameNode process should start automatically once the virtual machine is bootstrapped.

To configure this, use the following command:

```
yum install hadoop-namenode
```

2.4.4. Validate autostart configuration for NameNode and the NameNode monitoring agent

Use the following instructions to validate that the NameNode and the NameNode monitoring agent are started automatically on server restart:

Step 1: Reboot the NameNode machine.

Step 2: Verify that the NameNode service is up and running.

- On the NameNode machine, execute the following command:

  ```
  service hadoop-namenode status
  ```

- If the service is up and running, you should see a message similar to the one shown in the following example:

  ```
  namenode (pid 4651) is running...
  ```

Step 3: Verify that the NameNode monitoring agent is up and running.

- On the NameNode machine, execute the following command:

  ```
  service hmonitor-namenode-monitor status
  ```

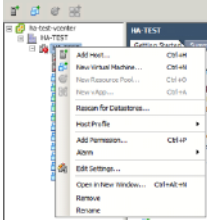
- If the service is up and running, you should see a message similar to the one shown in the following example:

  ```
  java (pid 4870) is running...
  ```

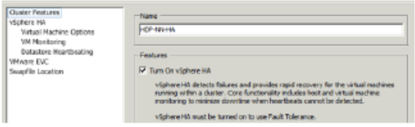
2.4.5. Enable vSphere for HA

Use the following instructions to modify the settings for the vSphere HA cluster created in the previous section:

Step 1: Right click on the cluster in vCenter and select ‘Edit Settings’.



Step 2: In the “Cluster Features” panel, enable the “Turn on vSphere HA” option.



Name: HDP-NN-HA

Features

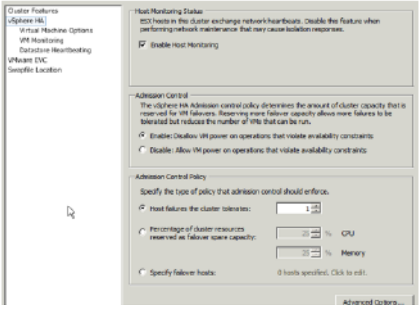
☑ Turn On vSphere HA

vSphere HA detects failures and provides rapid recovery for the virtual machines running within a cluster. Core functionality includes host and virtual machine monitoring to minimize downtime when heartbeats cannot be detected.

vSphere HA must be turned on to use Fault Tolerance.

Step 3: In the left navigation menu, select the “vSphere HA” option and configure the following parameters.

- Enable the “Enable Host Monitoring” option.
- In the same panel, enable Admission Control.
- Under “Admission Control Policy” section, select “Host failures the cluster tolerates” and set the number of failures as 1.



Host Monitoring Status

ESX hosts in this cluster exchange network heartbeats. Disable this feature when performing network maintenance that may cause isolation responses.

☑ Enable Host Monitoring

Admission Control

The vSphere HA Admission control policy determines the amount of cluster capacity that is reserved for VM failovers. Reserving more failover capacity allows more failures to be tolerated but reduces the number of VMs that can be run.

● Enable: Disallow VM power on operations that violate availability constraints

○ Disable: Allow VM power on operations that violate availability constraints

Admission Control Policy

Specify the type of policy that admission control should enforce.

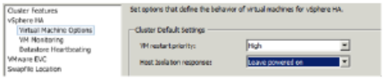
● Host failures the cluster tolerates: 1

○ Percentage of cluster resources reserved as failover spare capacity: 25 % CPU 25 % Memory

○ Specify failover hosts: 0 hosts specified. Click to edit.

Step 4: In the left navigation menu, select the “Virtual Machine Options” and configure the following parameters.

- Set the “VM restart priority” to ‘High’.
- Set the “Host Isolation response” to ‘Leave powered on’.



Set options that define the behavior of virtual machines for vSphere HA.

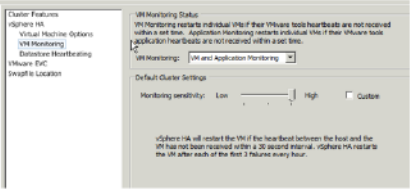
Cluster Default Settings

VM restart priority: High

Host Isolation response: Leave powered on

Step 5: In the left navigation menu, select the “VM Monitoring” option and configure the following parameters:

- Set the “VM Monitoring” to ‘VM and Application Monitoring’.
- Set the “Monitoring sensitivity” slider to ‘High’.



VM Monitoring restarts individual VMs if their VMware tools heartbeats are not received within a set time. Application Monitoring restarts individual VMs if their VMware tools application heartbeats are not received within a set time.

VM Monitoring: VM and Application Monitoring

Monitoring sensitivity: High □ Custom

vSphere HA will restart the VM if the heartbeat between the host and the VM has not been received within a 30 second interval. vSphere HA restarts the VM after each of the first 3 failures every hour.

2.4.6. Validate NameNode High Availability

It is critical to verify that the NameNode HA infrastructure is functional.

To commission a NameNode HA cluster, we recommend performing explicit tests of vSphere's ability to detect a failed NameNode and to react to that failure. These tests should be performed in addition to any other commissioning tests.

To aid in this commissioning process, the HA monitor RPM includes the Hortonworks Application Monitor (HAM) application. HAM can be used to monitor and restart a remote NameNode. To validate NameNode High Availability, ensure that you follow the instructions as outlined below:

2.4.6.1. Install and configure HAM

Typically, HAM can be run on the NameNode. However, we recommend that you also install HAM on other nodes in the cluster because the NameNode will be restarted as part of the validation tests. Use the following command to install HAM:

- For RHEL v5.x:

  ```
  rpm -ivh hmonitor-1.1.0.15-1.el5.x86_64.rpm
  ```

- For RHEL v6.x:

  ```
  rpm -ivh hmonitor-1.1.0.15-1.el6.x86_64.rpm
  ```

2.4.6.2. Invoke HAM application

Use the following command to invoke the HAM application:

```
/usr/lib/hadoop/monitor/ham.sh -conf hadoop-site.xml
```

This displays a GUI that lists the current values and state of the NameNode and Job Tracker.

You can use this GUI to trigger blocking and non-blocking HDFS operations.

Blocking operations will block until the NameNode has failed over. The blocking operations use the `dfs.retry` properties. Non-blocking operations will fail if the NameNode is down, but these operations start succeeding once the failover NameNode is live.

2.4.6.3. Validate the fail over behavior

Use the following tests to verify fail over behavior. (These tests can also be used to verify that the availability monitoring can be suspended for administrative tasks.) TODO add xml:id for subsections here

2.4.6.3.1. Verify that NameNode failure triggers the fail over

1. Start the NameNode VM and run the HAM application configured to work with this NameNode.

2. In HAM, start blocking LS operations.

3. SSH to the NameNode VM and terminate the NameNode process.

   ```
   service hadoop-namenode stop
   ```

   Alternatively, identify the NameNode process (**jps -v**) and issue **kill -9** command.

4. Ensure that you see the following expected results:

   - In HAM, the NameNode status area (at the top of the application) should display offline status for NameNode. The main area should also stop displaying any new text (this indicates that the file system operations are blocked).

   - In the vSphere Management UI, the vSphere should terminate the NameNode VM within 60-90 seconds and must start a new instance.

   - Once the NameNode service restarts, its status must be displayed in both the vSphere UI and in the status indicator of HAM.

   - The blocking operations started in HAM must now continue. The fail over should not affect the client except for the pause during fail over.

   - SSH to the NameNode VM again and verify that the host name, IP address, and SSH host key have not changed.

2.4.6.3.2. Verify that a hung NameNode triggers the fail over

This test verifies that the VM does not fail immediately after the NameNode process is hung. The monitor considers this time period as a Garbage Collection-related pause. The monitor provides a (configurable) period of grace time before it terminates the hung NameNode process.

1. Start the NameNode VM and run the HAM application configured to work with this NameNode.

2. In HAM, start non-blocking operations.

3. SSH to the NameNode VM and identify the NameNode process.

   ```
   jps -v| grep namenode
   ```

4. Suspend the NameNode process.

   ```
   kill -19 namenode-process-id-here
   ```

5. Ensure that you see the following expected results:

   - In HAM, the NameNode status area must indicate hung status. The non-blocking operations, that are initiated, will now appear to be blocked (the hung NameNode prevents these operations from completing).

- In the vSphere Management UI, the vSphere should terminate the NameNode VM and start a new instance within a delay of approximately 2-3 minutes.
- In HAM, the NameNode status area should indicate offline status. The non-blocking operations should now report failure.
- Once the NameNode service restarts, its status must be displayed in both the vSphere UI and in the status indicator of HAM.
- The operations started in HAM will now start succeeding.

This test can be repeated when HAM performs blocking operations.

In this case, the active filesystem operation (the operation when the NameNode was suspended) will fail when the NameNode is restarted and reported as such. This failure happens because the open socket connection breaks and these connections are not preserved during a fail over.

2.4.6.3.3. Verify that ESXi server failure triggers the fail over

This test verifies that the HA solution detects the failures of the physical hardware and also trigger fail overs.

1. Start the NameNode VM in an ESXi server that is not running any other VMs.
2. Run the HAM application configured to work against this NameNode.
3. In the HAM, start blocking LS operations.
4. Initiate a power down of the ESXi server.
5. Ensure that you see the following expected results:
   - The main area should stop displaying new text - this indicates that the file system operations are blocked.
   - In the vSphere management UI, once the loss of the ESXi server is detected, the NameNode VM is re-instantiated on one of the remaining ESXi servers.
   - Once the NameNode service restarts, its status must be displayed in both the vSphere UI and in the status indicator of HAM.
   - The blocked LS operation started in HAM should now continue without failures.

2.4.6.3.4. Verify that no fail over is triggered on planned shutdown of the monitor service

This test verifies that if the monitor service is shut down the fail over is not triggered. The NameNode can now be manipulated as part of planned management operations.

1. Start the NameNode VM and SSH to the NameNode VM.
2. Terminate the monitor process.

```
service hmonitor-namenode-monitor stop
```

> **Note**
>
> A **kill -9** command for the monitor is not a graceful shutdown and will trigger fail over.

3. Terminate the NameNode process.

    ```
    service hadoop-namenode stop
    ```

4. Ensure that you see the following expected results:

    - In the vSphere Management UI, the NameNode VM should be live.
    - The fail over should not be initiated by vSphere.
    - In HAM, the NameNode status area should indicate offline status. The non-blocking operations should now report failure.
    - The SSH connection must not be broken and the VM should be live.

5. Restart the monitor process.

    ```
    service hmonitor-namenode-monitor start
    ```

6. Restart the NameNode process.

    ```
    service hadoop-namenode start
    ```

7. The NameNode health should be monitored and failures should trigger fail over.

2.4.6.3.5. Verify that the monitor provides a bootstrap period before reporting that the NameNode is not live

This test verifies that the monitoring process includes a bootstrap period.

The bootstrap period ensures that the monitor will immediately not report a failure and trigger a restart. Instead, the monitor provides the service a bootstrap period in which probes are allowed to (initially fail). This bootstrap period is configurable (see: TODO:xlink here Tuning the bootstrap timeout).

1. Start the NameNode VM and SSH to the NameNode VM.
2. Terminate the monitor process.

    ```
    service hmonitor-namenode-monitor stop
    ```

3. Terminate the NameNode process.

    ```
    service hadoop-namenode stop
    ```

4. Restart the monitor process.

    ```
    service hmonitor-namenode-monitor start
    ```

2.4.6.3.6. Verify that no fail over is triggered when the NameNode enters the safe mode

This test verifies that the VM is not restarted if the NameNode enters safe mode. This allows administration operations to be performed on a file system in safemode without having to disable HA services.

1. Start the NameNode VM and SSH to the NameNode VM.
2. Enter safe mode.

   ```
   hadoop dfsadmin -safemode enter
   ```

3. Ensure that you see the following expected results:
   - In the vSphere UI, the NameNode VM should be live.
   - The SSH session should exist and the VM should be live.
4. Terminate the NameNode process.

   ```
   service hadoop-namenode stop
   ```

5. The vSphere should identify the NameNode failure and should restart the VM.

> **Note**
>
> This test shows that even in safe mode the fail over is triggered if the NameNode process is terminated. To avoid automatic restart for NameNode after performing safe mode operations, use the **service hmonitor-namenode-monitor restart** to restart the monitor service.

2.4.7. Administration Guide for Highly Available NameNode

2.4.7.1. NameNode shutdown for planned maintenance

Use the following steps to ensure controlled shutdown of NameNode server:

Step 1: Shutdown the NameNode monitoring agent.

```
service stop hmonitor-namenode-monitor
```

Step 2: Shutdown the NameNode process.

```
service stop hadoop-namenode
```

2.4.7.2. Starting the NameNode

Use the following instructions to start the NameNode:

Step 1: Start NameNode process.

```
service start hadoop-namenode
```

Step 2: Start the NameNode monitoring agent and register it with vSphere.

```
service start hmonitor-namenode-monitor
```

Step 3: Verify that the monitoring process has started.

```
service status hmonitor-namenode-monitor
```

2.4.7.3. Reconfiguring HDFS

Use the following instructions to reconfigure HDFS:

Step 1: Disable HA on vSphere.

Step 2: Shut down HDFS using the instructions available here. TODO point to Manually Controlling Services doc.

Step 3: Update the HDFS parameters using HMC. TODO point to new HMC doc.

Step 4: Restart HDFS using HMC. TODO point to new HMC doc.

Step 5: Enable HA on vSphere.

2.4.7.4. Tuning parameters for your environment

TODO add xml id for subsections here.

2.4.7.4.1. Tuning parameters for your environment

When the VM starts, the HA Monitor waits for the NameNode to begin responding to file system operations. During this "bootstrap phase”, the HA monitor does not report startup failures of NameNode probes to the HA infrastructure. The HA monitor exits the bootstrap phase once all the probes succeed (from that point, the failure of a probe is reported as a service failure).

The time limit of the bootstrap phase can be configured using the **service.monitor.bootstrap.timeout** property:

```

<property>
<name>service.monitor.bootstrap.timeout</name>
<value>120000</value>
<description>
The time in milliseconds for the monitor to wait for the service to bootstrap
 and
become available before it reports a failure to the management infrastructure
</description>
</property>
```

The timeout must be sufficiently long so that the monitored service is able to open its network ports for external interaction. For the NameNode, the web page and IPC port must be open.

The bootstrap time also needs to include the time required for the HDFS journal replay operations. The bootstrap timeout value should be kept high if the filesystem is large and if the secondary NameNode checkpointing time intervals are longer.

2.4.7.4.2. Tuning the reporting rate

The internal VM Monitor daemon sends “heartbeat” messages to vSphere to indicate that the VM is alive. Use the following property, to modify the rate at which these heartbeats are sent.

```
<property>
<name>service.monitor.report.interval</name>
<value>7000</value>
<description>
Interval in milliseconds for sending heartbeats to vSphere.
</description>
</property>
```

It is essential that the live VM sends a heartbeats to vSphere at least every thirty seconds.

A smaller reporting interval reduces the risk of missed heartbeats in case an Operating System or Java related process hangs. However, a smaller reporting interval can also have adverse effects - especially if the VM is overloaded.

It is therefore strongly recommended to address the root cause of the VM overload. If your VM is overloaded and becomes unresponsive, we recommend that you either add more CPUs and RAM or rebalance VMs across the cluster.

2.4.7.4.3. Tuning the probing rate

The Monitor daemon probes the health of the NameNode at a regular interval, and stops sending heartbeats to vSphere when any of the probes fail. Use the following property to change the rate of probes:

```
<property>
<name>service.monitor.probe.interval</name>
<value>11000</value>
<description>
Time in milliseconds between the last probe cycle ending and the new one
 beginning.
The shorter this cycle, the faster failures are detected, but more CPU,
 network,
and server load can be generated.
</description>
</property>
```

The smaller the interval between probes, the faster it becomes to detect and report service failures. Although, this might increase the load on the service and the CPU slightly, but even with a very short probing interval, vSphere will not trigger VM restart for at least thirty seconds after the probe failure.

2.4.7.4.4. Tuning for NameNode Garbage Collection

The NameNode process can appear hung during Garbage Collection event. To prevent this from triggering immediate failover, a grace period is provided to the NameNode to resume its operation. You can configure this grace period using the following property:

```
<property>
<name>service.monitor.probe.timeout</name>
<value>60000</value>
<description>
Duration in milliseconds for the probe loop to be blocked, before it is
 considered a liveness failure
</description>
</property>
```

A smaller value will cause the VM (where the hung NameNode process is running) faster, but it increases the risk of incorrectly identifying a long GC-related pause as a hung process. On larger clusters (with longer GC pauses), you can increase the value of this property.

2.4.8. References

TODO add link to the latest VMWare White paper.http://www.vmware.com/files/pdf/VMwareHA_twp.pdf