2 Installing  and  configuring  HDP

**Hardware/Software selection Winodws**

The following operating systems are supported:

* Windows Server 2008 R2 (64-bit)
* Windows Server 2012 (64-bit)

Your system must have the correct JDK installed on all the nodes of the cluster. HDP supports the following JDKs.

* Oracle JDK 1.6 update 31 64-bit
* Oracle JDK 7 64-bit

**Database Requirements**

* By default, Hive and Oozie use an embedded Derby database for its metastore.

To use an external database for Hive and Oozie metastores, ensure that Microsoft SQL Server database is deployed and available in your environment.

Wednesday, June 11, 2014

1:48 PM

Understand  minimum  hardware  and  software  requirements

•Understand  how  to  setup  a  local  repository  for  HDP  installation

•Understand  How  to  install  HDP  using  Apache  Ambari

•Understand  difference  between  master  and  slave  services

•Understand  complete  deployment  layout

•Understand  how  to  configure  and manage  different  services

•Understand  different  configuration  parameters

**Operating Systems Requirements**

* + 64-bit Red Hat Enterprise Linux (RHEL) 5 or 6
  + 64-bit CentOS 5 or 6
  + 64-bit SUSE Linux Enterprise Server (SLES) 11, SP1

**Software Requirements**

On each of your hosts:

* + yum [for RHEL or CentOS]
  + zypper [for SLES]
  + rpm
  + scp [for multiple node installs]
  + curl
  + wget
  + unzip
  + tar
  + pdsh [for multiple node installs over many hosts]

**Configure the Remote Repository**

HDP install fetches the software from a remote yum repository over the Internet. To use this option, you must set up access to the remote repository and have an available Internet connection for each of your hosts.

If your cluster does not have access to the Internet, or you are creating a large cluster and you want to conserve bandwidth, you can instead provide a local copy of the HDP repository that your hosts can access.

* + For RHEL and CentOS 5  
    wget -nv <http://public-repo-1.hortonworks.com/HDP/centos5/1.x/updates/1.3.7.0/hdp.repo> -O /etc/yum.repos.d/hdp.repo

**Database Requirements**

* To use external database for Hive or Oozie metastore, ensure that a MySQL or Oracle or PostgreSQL database is deployed and available.  
  (By default, Hive and Oozie use Derby database for the metastore.)

**JDK Requirements**

Your system must have the correct JDK installed on all the nodes of the cluster. HDP requires Oracle JDK 1.6 update 31.

Use the following instructions to manually install JDK 1.6 update 31:

**Prepare the Environment**

To deploy your HDP instance, you need to prepare your deploy environment:

* [Enable NTP on the Cluster](http://docs.hortonworks.com/HDPDocuments/HDP1/HDP-1.3.7/bk_installing_manually_book/content/rpm-chap1-6.html#rpm-chap1-6-1)
* [Check DNS](http://docs.hortonworks.com/HDPDocuments/HDP1/HDP-1.3.7/bk_installing_manually_book/content/rpm-chap1-6.html#rpm-chap1-6-2)
* [Disable SELinux](http://docs.hortonworks.com/HDPDocuments/HDP1/HDP-1.3.7/bk_installing_manually_book/content/rpm-chap1-6.html#rpm-chap1-6-3)

Configure the Local Repositories

If your cluster does **not** have access to the Internet, or you are creating a large cluster and you want to conserve bandwidth, you need to provide access to the bits using an alternative method.

Configure Ambari Server so that it knows to connect to the mirrored repositories during installation.

1. On Ambari Server, browse to the stacks definitions directory  
   cd /var/lib/ambari-server/resources/stacks  
   There are two stack definitions in this directory: HDP and HDPLocal. The HDP definition points to the publicly hosted HDP software packages. You must modify the HDPLocal definition to point to the local repositories you have set up.
2. Browse to the stack HDPLocal 1.2.0 repos directory.  
   cd HDPLocal/1.2.0/repos
3. Edit the repo info file:  
   vi repoinfo.xml

You must update the <baseurl> value to point to your local repositories for each operating system that your cluster includes. So, for example, if your system includes hosts running CentOS 6, to point to the HDP and HDP-EPEL repositories, you would update stanzas to look something like this:

<os type="centos6">  
     <repo>         
 <baseurl>**http://{your.hosted.local.repository}/HDP-1.2.0/repos/centos6**</baseurl>         
 <repoid>HDP-1.2.0</repoid>  
       <reponame>HDP</reponame>  
     </repo>  
     <repo>  
       <baseurl>**http://{your.hosted.local.repository}/HDP-1.2.0/repos/centos6**</baseurl>  
       <repoid>HDP-epel</repoid>  
       <reponame>HDP-epel</reponame>  
       <mirrorslist><![CDATA[http://mirrors.fedoraproject.org/mirrorlist?repo=epel-6&arch=$basearch]]></mirrorslist>  
     </repo>  
   </os>

**Imstallation Steps using AMbari**

1. Install the epel repository:

yum install epel-release

1. Run the Ambari Server setup:

ambari-server setup

1. To start the Ambari Server:  
   ambari-server start
2. Point your browser to http://{main.install.hostname}:8080.
3. If you do not wish to have Ambari automatically install the Ambari Agent on all your hosts using SSH, you have the option of doing this work manually. Uncheck **Provide your SSH Private Key**and **OK** out of the **Warning**

**Configuration Parameters**

**HDFS Settings:NameNode**

|  |  |
| --- | --- |
| **Name** | **Notes** |
| NameNode host | This value is prepopulated based on your choices on previous screens |
| NameNode directories | NameNode directories for HDFS to store the file system image. |
| NameNode Java heap size | Initial and maximum Java heap size for NameNode (Java options -Xms and -Xmx) |
| NameNode new generation size | Default size of Java new generation for NameNode (Java option -XX:NewSize) |

**HDFS Settings:SNameNode**

|  |  |
| --- | --- |
| **Name** | **Notes** |
| SNameNode host | This value is prepopulated based on your choices on previous screens |
| Secondary NameNode Checkpoint Directory | Directory on the local filesystem where the Secondary NameNode should store the temporary images to merge |

**HDFS Settings:General**

|  |  |
| --- | --- |
| **Name** | **Notes** |
| WebHDFS enabled | Check to enable WebHDFS |
| Hadoop maximum Java heap size | Maximum Java heap size for daemons such as Balancer (Java option -Xmx) [[a](http://docs.hortonworks.com/HDPDocuments/HDP1/HDP-1.2.0/bk_using_Ambari_book/content/ambari-chap3-7-1.html#ftn.d6e803)] |
| Reserved space for HDFS | Space in GB per volume reserved for HDFS |
| HDFS Maximum Checkpoint Delay | Maximum delay between two consecutive checkpoints for HDFS in seconds |
| HDFS Maximum Edit Log Size for Checkpointing | Maximum size of the edits log file that forces an urgent checkpoint even if the maximum checkpoint delay is not reached |
| [[a](http://docs.hortonworks.com/HDPDocuments/HDP1/HDP-1.2.0/bk_using_Ambari_book/content/ambari-chap3-7-1.html#d6e803)]The default value for this property is 1 GB. This value may need to be reduced for a VM-based installation. On the other hand, for significant work using Hive Server, 2GB is a more realistic value. |  |

**HDFS Settings:DataNodes**

|  |  |
| --- | --- |
| **Name** | **Notes** |
| DataNode hosts | The hostnames of the hosts on which this group's DataNodes run. |
| DataNode directories | The directories where HDFS should store the data blocks for this group. |
| DataNode maximum Java heap size | Maximum Java heap size for DataNode (Java option -Xmx) |
| DataNode volumes failure toleration | The number of volumes that are allowed to fail before a DataNode stops offering services. |