Hortonworks Data Platform

Reference

(Mar 20, 2013)

docs.hortonworks.com

Hortonworks Data Platform : Reference

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

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

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

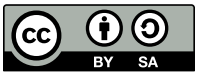


Table of Contents

List of Tables

1. Hadoop Service Accounts

This topic provides information about the service users for Hadoop.

HDP creates the following service users:

Note

You must always execute the `createUsers.sh` script file to ensure that these users are created by the installer.

The user names for these service users cannot be modified.

Table 1.1. Hadoop Service Accounts

HDP Service	User Name	Notes
HDFS	`hdfs`	NameNode, Secondary NameNode, and the DataNodes run as the `hdfs` user.
MapReduce	`mapred`	JobTracker, Job HistoryServer, and the TaskTrackers run as the `mapred` user.
HBase	`hbase`	HBase Master and the RegionServers run as the `hbase` user.
Hive	`hive`	Hive Metastore and HiveServer2 run as the `hive` user.
HCatalog	`hcat`	HCatalog runs as the `hcat` user. WebHCat Server also runs as the `hcat` user.
ZooKeeper	`zookeeper`	ZooKeeper server runs as the `zookeeper` user.
Oozie	`oozie`	Oozie server runs as the `oozie` user.

2. Configuring Ports

The tables below specify which ports must be opened for which ecosystem components to communicate with each other. Make sure the appropriate ports are opened before you install HDP.

2.1. HDFS Ports

The following table lists the default ports used by the various HDFS services.

Table 2.1. HDFS Ports

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
NameNode WebUI	Master Nodes (NameNode and any back-up NameNodes)	50070	http	Web UI to look at current status of HDFS, explore file system	Yes (Typically admins, Dev/ Support teams)	`dfs.http.address`
		50470	https	Secure http service		`dfs.https.address`
NameNode metadata service	Master Nodes (NameNode and any back-up NameNodes)	8020/9000	IPC	File system metadata operations	Yes (All clients who directly need to interact with the HDFS)	Embedded in URI specified by `fs.default.name`
DataNode	All Slave Nodes	50075	http	DataNode WebUI to access the status, logs etc.	Yes (Typically admins, Dev/ Support teams)	`dfs.datanode.http.address`
		50475	https	Secure http service		`dfs.datanode.https.address`
		50010		Data transfer		`dfs.datanode.address`
		50020	IPC	Metadata operations	No	`dfs.datanode.ipc.address`
Secondary NameNode	Secondary NameNode and any backup Secondanry NameNode	50090	http	Checkpoint for NameNode metadata	No	`dfs.secondary.http.address`

2.2. MapReduce Ports

The following table lists the default ports used by the various MapReduce services.

Table 2.2. MapReduce Ports

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
JobTracker WebUI	Master Nodes (JobTracker Node and any back-up Job-Tracker node)	50030	http	Web UI for JobTracker	Yes	`mapred.job.tracker.http.address`
JobTracker	Master Nodes (JobTracker Node)	8021	IPC	For job submissions	Yes (All clients who need to submit the MapReduce jobs including Hive, Hive server, Pig)	Embedded in URI specified by `mapred.job.tracker`
Task-Tracker Web UI and Shuffle	All Slave Nodes	50060	http	DataNode Web UI to access status, logs, etc.	Yes (Typically admins, Dev/ Support teams)	`mapred.task.tracker.http.address`
History Server WebUI		51111	http	Web UI for Job History	Yes	`mapreduce.history.server.http.address`

2.3. Hive Ports

The following table lists the default ports used by the various Hive services.

Note

Neither of these services are used in a standard HDP installation.

Table 2.3. Hive Ports

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
Hive Server2	Hive Server machine (Usually a utility machine)	10000	thrift	Service for programatically (Thrift/ JDBC) connecting to Hive	Yes (Clients who need to connect to Hive either programatically or through UI SQL tools that use JDBC)	`ENV Variable HIVE_PORT`

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
Hive Metastore		9083	thrift	Yes (Clients that run Hive, Pig and potentially M/R jobs that use HCatalog)	`hive.metastore.uris`	

2.4. HBase Ports

The following table lists the default ports used by the various HBase services.

Table 2.4. HBase Ports

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
HMaster	Master Nodes (HBase Master Node and any back-up HBase Master node)	60000			Yes	`hbase.master.port`
HMaster Info Web UI	Master Nodes (HBase master Node and back up HBase Master node if any)	60010	http	The port for the HBase-Master web UI. Set to -1 if you do not want the info server to run.	Yes	`hbase.master.info.port`
Region Server	All Slave Nodes	60020			Yes (Typically admins, dev/ support teams)	`hbase.regionserver.port`
Region Server	All Slave Nodes	60030	http		Yes (Typically admins, dev/ support teams)	`hbase.regionserver.info.port`
	All ZooKeeper Nodes	2888		Port used by ZooKeeper peers to talk to each other.See	No	`hbase.zookeeper.peerport`

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
				here for more information.		
	All ZooKeeper Nodes	3888		Port used by ZooKeeper peers to talk to each other.See here for more information.		`hbase.zookeeper.leaderport`
		2181		Property from ZooKeeper's config `zoo.cfg`. The port at which the clients will connect.		`hbase.zookeeper.property.clientPort`

2.5. WebHCat Port

The following table lists the default ports used by the WebHCat service.

Table 2.5. WebHCat Port

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
WebHCat Server	Any utility machine	50111	http	Web API on top of HCatalog and other Hadoop services	Yes	`templeton.port`

2.6. Ganglia Ports

The following table lists the default ports used by the various Ganglia services.

Table 2.6. Ganglia Ports

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
	Ganglia server	8660/61/62/63		For gmond collectors		
	All Slave Nodes	8660		For gmond agents		
	Ganglia server	8651		For ganglia gmetad		

2.7. MySQL Ports

The following table lists the default ports used by the various MySQL services.

Table 2.7. MySQL Ports

Service	Servers	Default Ports Used	Protocol	Description	Need End User Access?	Configuration Parameters
MySQL	MySQL database server	3306				

3. Controlling HDP Services Manually

This section describes how to start and stop HDP services manually.

3.1. Starting HDP services

Start all the Hadoop services in the following order:

- HDFS
- MapReduce
- ZooKeeper
- HBase
- Hive Metastore
- HiveServer2
- WebHCat
- Oozie
- Ganglia
- Nagios

Instructions

1. Start HDFS

   a. Execute these commands on the NameNode host machine:

   ```
   su -l hdfs -c "/usr/lib/hadoop/bin/hadoop-daemon.sh --config /etc/hadoop/
   conf start namenode"
   ```

   b. Execute these commands on the Secondary NameNode host machine:

   ```
   su -l hdfs -c "/usr/lib/hadoop/bin/hadoop-daemon.sh --config /etc/hadoop/
   conf start secondarynamenode”
   ```

   c. Execute these commands on all DataNodes:

   ```
   su -l hdfs -c "/usr/lib/hadoop/bin/hadoop-daemon.sh --config /etc/hadoop/
   conf start datanode"
   ```

2. Start MapReduce

   a. Execute these commands on the JobTracker host machine:

   ```
   su -l mapred -c "/usr/lib/hadoop/bin/hadoop-daemon.sh --config /etc/
   hadoop/conf start jobtracker; sleep 25"
   ```

   b. Execute these commands on the JobTracker host machine:

```
su -l mapred -c "/usr/lib/hadoop/bin/hadoop-daemon.sh --config /etc/
hadoop/conf start historyserver"
```

c. Execute these commands on all TaskTrackers:

```
su -l mapred -c "/usr/lib/hadoop/bin/hadoop-daemon.sh --config /etc/
hadoop/conf start tasktracker"
```

3. Start ZooKeeper. On the ZooKeeper host machine, execute the following command:

```
su - zookeeper -c "export  ZOOCFGDIR=/etc/zookeeper/conf ; export ZOOCFG=
zoo.cfg ; source /etc/zookeeper/conf/zookeeper-env.sh ; /usr/lib/zookeeper/
bin/zkServer.sh start"
```

4. Start HBase

a. Execute these commands on the HBase Master host machine:

```
su -l hbase -c "/usr/lib/hbase/bin/hbase-daemon.sh --config /etc/hbase/
conf start master"
```

b. Execute these commands on all RegionServers:

```
su -l hbase -c "/usr/lib/hbase/bin/hbase-daemon.sh --config /etc/hbase/
conf start regionserver"
```

5. Start Hive Metastore. On the Hive Metastore host machine, execute the following command:

```
 su -l hive -c "nohup hive --service metastore > $HIVE_LOG_DIR/hive.out 2>
$HIVE_LOG_DIR/hive.log   &"
```

where *$HIVE_LOG_DIR* is the directory where Hive server logs are stored (example: /var/log/hive).

6. Start HiveServer2. On the Hive Server2 host machine, execute the following command:

```
sudo su hive -c "nohup /usr/lib/hive/bin/hiveserver2 -hiveconf hive.
metastore.uris=\" \" > $HIVE_LOG_DIR /hiveServer2.out 2>$HIVE_LOG_DIR/
hiveServer2.log &"
```

where *$HIVE_LOG_DIR* is the directory where Hive server logs are stored (example: /var/log/hive).

7. Start WebHCat. On the WebHCat host machine, execute the following command:

```
su -l hcat -c "/usr/lib/hcatalog/sbin/webhcat_server.sh  start"
```

8. Start Oozie. On the Oozie server host machine, execute the following command:

```
 sudo su -l oozie -c "cd $OOZIE_LOG_DIR/log; /usr/lib/oozie/bin/oozie-start.
sh"
```

where *$OOZIE_LOG_DIR* is the directory where Oozie log files are stored (for example: `/var/log/oozie`).

9. Start Ganglia.

a. Execute this command on the Ganglia server host machine:

```
/etc/init.d/hdp-gmetad start
```

b. Execute this command on all the nodes in your Hadoop cluster:

```
/etc/init.d/hdp-gmond start
```

10. Start Nagios.

```
service nagios start
```

3.2. Stopping HDP services

Before trying any upgrades or uninstalling software, stop all Hadoop services in the following order:

- Nagios
- Ganglia
- Oozie
- WebHCat
- Hive Metastore
- ZooKeeper
- HBase
- MapReduce
- HDFS

1. Stop Nagios. On the Nagios host machine, execute the following command:

```
service nagios stop
```

2. Stop Ganglia.

a. Execute this command on the Ganglia server host machine:

```
/etc/init.d/hdp-gmetad stop
```

b. Execute this command on all the nodes in your Hadoop cluster:

```
/etc/init.d/hdp-gmond stop
```

3. Stop Oozie.

4. Stop WebHCat. On the WebHCat host machine, execute the following command:

```
su -l hcat -c "/usr/lib/hcatalog/sbin/webhcat_server.sh  stop"
```

5. Stop Hive. On the Hive Metastore host machine and Hive Server2 host machine, execute the following command:

```
ps aux | awk '{print $1,$2}' | grep hive | awk '{print $2}' | xargs kill >/
dev/null 2>&1
```

6. Stop ZooKeeper. On the ZooKeeper host machine, execute the following command:

```
su - zookeeper -c "export ZOOCFGDIR=/etc/zookeeper/conf ; export ZOOCFG=zoo.
cfg ;source /etc/zookeeper/conf/zookeeper-env.sh ; /usr/lib/zookeeper/bin/
zkServer.sh stop"
```

7. Stop HBase.

   a. Execute these commands on all RegionServers:

   ```
   su -l hbase -c "/usr/lib/hbase/bin/hbase-daemon.sh --config /etc/hbase/
   conf stop regionserver"
   ```

   b. Execute these commands on the HBase Master host machine:

   ```
   su -l hbase -c "/usr/lib/hbase/bin/hbase-daemon.sh --config /etc/hbase/
   conf stop master"
   ```

8. Stop MapReduce

   a. Execute these commands on all TaskTrackers:

   ```
   su -l mapred -c "/usr/lib/hadoop/bin/hadoop-daemon.sh --config /etc/
   hadoop/conf stop tasktracker"
   ```

   b. Execute these commands on the JobTracker host machine:

   ```
   su -l mapred -c "/usr/lib/hadoop/bin/hadoop-daemon.sh --config /etc/
   hadoop/conf stop historyserver"
   ```

   c. Execute theses commands on the JobTracker host machine:

   ```
   su -l mapred -c "/usr/lib/hadoop/bin/hadoop-daemon.sh --config /etc/
   hadoop/conf stop jobtracker"
   ```

9. Stop HDFS

   a. Execute these commands on all DataNodes:

   ```
   su -l hdfs -c "/usr/lib/hadoop/bin/hadoop-daemon.sh --config /etc/hadoop/
   conf stop datanode"
   ```

   b. Execute these commands on the Secondary NameNode host machine:

   ```
   su -l hdfs -c "/usr/lib/hadoop/bin/hadoop-daemon.sh --config /etc/hadoop/
   conf stop secondarynamenode”
   ```

   c. Execute these commands on the NameNode host machine:

   ```
   su -l hdfs -c "/usr/lib/hadoop/bin/hadoop-daemon.sh --config /etc/hadoop/
   conf stop namenode"
   ```

4. Deploying HDP In Production Data Centers with Firewalls

This section describes mechanisms for deploying HDP in situations where a connection to the Internet is not possible or desirable.

4.1. Deployment Strategies for Data Centers with Firewalls

A typical Hortonworks Data Platform (HDP) install requires access to the Internet in order to fetch software packages from a remote repository. Since corporate networks typically have various levels of firewalls, these firewalls may limit or restrict Internet access, making it impossible for your cluster nodes to access the HDP repository during the install process.

The solution for this is to either:

- Create a local mirror repository inside your firewall hosted on a local mirror server inside your firewall; or
- Provide a trusted proxy server inside your firewall that can access the hosted repositories.

This document will cover these two options in detail, discuss the trade-offs, provide configuration guidelines, and will also provide recommendations for your deployment strategy.

In general, before installing Hortonworks Data Platform in a production data center, it is best to ensure that both the Data Center Security team and the Data Center Networking team are informed and engaged to assist with these aspects of the deployment.

4.1.1. Terminology

Table 4.1. Terminology

Item	Description
Yum Package Manager (yum)	A package management tool that fetches and installs software packages and performs automatic dependency resolution. See http://yum.baseurl.org/ for more information.
Local Mirror Repository	The yum repository hosted on your Local Mirror Server that will serve the HDP software.
Local Mirror Server	The server in your network that will host the Local Mirror Repository. This server must be accessible from all hosts in your cluster where you will install HDP.
HDP Repositories	A set of repositories hosted by Hortonworks that contains the HDP software packages. HDP software packages include the HDP Repository and the HDP-UTILS Repository.
HDP Repository Tarball	A tarball image that contains the complete contents of the HDP Repositories.

4.1.2. Options for Mirroring or Proxying

HDP uses yum to install software, and this software is obtained from the: HDP Repositories, and the Extra Packages for Enterprise Linux (EPEL) repository.

If your firewall prevents Internet access, it will be necessary to mirror and/or proxy both the HDP repository and the Extra Packages for Enterprise Linux (EPEL) repository. Many Data Centers already mirror or proxy the EPEL repository, so discuss with your Data Center team whether EPEL is already available from within your firewall.

Mirroring a repository involves copying the entire repository and all its contents onto a local server and enabling an HTTPD service on that server to serve the repository locally. Once the local mirror server setup is complete, the `*.repo` configuration files on every repository client (i.e. cluster nodes) must be updated, so that the given package names are associated with the local mirror server instead of the remote repository server.

There are three options for creating a local mirror server. Each of these options is explained in detail in a later section.

- **Option I:** Mirror server has no access to Internet at all

  Use a web browser on your workstation to download the HDP Repository Tarball, move the tarball to the selected mirror server using scp or an USB drive, and extract it to create the repository on the local mirror server.

- **Option II:** Mirror server has temporary access to Internet

  Temporarily configure a server to have Internet access, download a copy of the HDP Repository to this server using the **reposync** command, then reconfigure the server so that it is back behind the firewall.

- **Option III:** Mirror server has permanent access to Internet (modified form of Option II)

  Establish a "trusted host", by permanently configuring a server to have Internet access, but still be accessible from within the firewall. Download a copy of the HDP Repository to this server using the **reposync** command.

  ### Note

  Option I is probably the least effort, and in some respects, is the most secure deployment option.

  Option III is best if you want to be able to update your Hadoop installation periodically from the Hortonworks Repositories.

  However, if you are considering Option III, you should also consider the fourth option, which is to proxy the HDP Repositories through a trusted proxy server. If you have a network administrator who has expertise in setting up proxies, and if the proxy option is acceptable within your Data Center Security policies, this can be the easiest of all the options.

- **Option IV:** Trusted proxy server

Proxying a repository involves setting up a standard HTTP proxy on a local server to forward repository access requests to the remote repository server and route responses back to the original requestor. Effectively, the proxy server makes the repository server accessible to all clients, by acting as an intermediary.

Once the proxy is configured, change the `/etc/yum.conf` file on every repository client (i.e. cluster nodes), so that when the client attempts to access the repository during installation, the request will go through the local proxy server instead of going directly to the remote repository server.

4.1.3. Considerations for choosing a Mirror or Proxy solution

The following table lists some benefits provided by these alternative deployment strategies:

Table 4.2. Comparison - HDP Deployment Strategies

Advantages of repository mirroring (Options I, II, and III)	Advantages of creating a proxy (Options IV)
• Minimizes network access (after the initial investment of copying the repository to local storage). The install process is therefore faster, reliable, and more cost effective (reduced WAN bandwidth minimizes the data center costs). • Allows security-conscious data centers to qualify a fixed set of repository files. It also ensures that the remote server will not change these repository files. • Large data centers may already have existing repository mirror servers for the purpose of OS upgrades and software maintenance. You can easily add the HDP Repositories to these existing servers.	• Avoids the need for long term management of the repository files (including periodic updates for upgrades, new versions, and bug fixes). • Almost all data centers already have a setup of well-known proxies. In such cases, you can simply add the local proxy server to the existing proxies' configurations. This approach is easier compared to creating local mirror servers in data centers with no mirror server setup. • The network access is same as that required when using a mirror repository, but the source repository handles file management.

However, each of the above approaches are also known to have the following disadvantages:

- Mirrors have to be managed for updates, upgrades, new versions, and bug fixes.
- Proxy servers rely on the repository provider to not change the underlying files without notice.
- Caching proxies are necessary, because non-caching proxies do not decrease WAN traffic and do not speed up the install process.

4.2. Recommendations for Deploying HDP

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

In many data centers, the following deployment strategy may be optimal:

- Use a mirror for the HDP Repositories. The HDP Repositories are small and easily mirrored, thereby allowing secure control over the contents of the Hadoop packages accepted for use in your data center.

- Use a caching proxy for the EPEL repository, which is a well-known and trustworthy repository managed by the Fedora Project team, and which may be too large to mirror in your data center. If your data center already mirrors or proxies EPEL, use that mirror or proxy.

Note

The installer pulls many packages from the base OS repositories (repos). If you do not have a complete base OS available to all your machines at the time of installation, you may run into issues. For example, if you are using RHEL 6 your hosts must be able to access the "Red Hat Enterprise Linux Server 6 Optional (RPMs)" repo. If this repo is disabled, the installation is unable to access the rubygems package.

If you encounter problems with base OS repos being unavailable, please contact your system administrator to arrange for these additional repos to be proxied or mirrored.

4.2.1. RPMs in the HDP repository

In the HDP repository, you will find two differenct source RPM for each component.

For example, for Hadoop, you should find the following two RPMs:

- `hadoop-x.x.x.x.el6.src.rpm`
- `hadoop-source-x.x.x.x.el6.i386.rpm`

The `src` and `source` are two different packages that serve the following purpose:

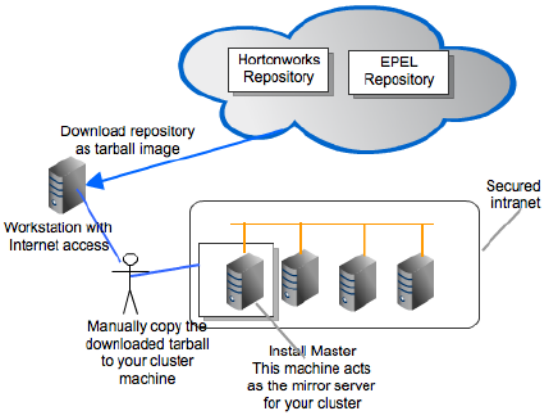
- The `src` package is used to re-create the binary in a given environment. You can use the `src` package of a particular component if you want to rebuild RPM for that component.
- The `source` package on the other hand, is used for reference or debuging purpose. The `source` package is particularly useful when you want to examine the source code of a particular component in a deployed cluster.

4.3. Detailed Instructions for Creating Mirrors and Proxies

In this section:

4.3.1. Option I - Mirror server has no access to the Internet

The local mirror setup for Option I is shown in the following illustration:



4.3.1.1. Prerequisites

To configure local repositories for HDP deployment, your system must meet the following minimum requirements:

- Your mirror server host must run on one of the following operating systems:
  - 64 bit Red Hat Enterprise Linux (RHEL)
  - 64 bit CentOS v5.x, v6.x
  - 64 bit SUSE Linux Enterprise Server (SLES) 11 SP1

  Ensure that this server and the cluster nodes are all running the same OS.

Note

To support repository mirroring for heterogeneous clusters requires a more complex procedure than the one documented here.

- The mirror server host must have several GB of storage available.
- The firewall allows all cluster nodes (the servers on which you want to install HDP) to access this server.
- If your mirror host uses SLES, execute the following command to install the required packages:

```
zypper -n --no-gpg-checks install apache2 apache2-prefork apache2-utils
 python-curl python-gobject2 python-gpgme python-urlgrabber python-iniparse
 wget curl yum-metadata-parser yum yum-updatesd yum-utils createrepo
```

4.3.1.2. Instructions

1. Use a workstation with access to the Internet and download the tarball image of the appropriate Hortonworks yum repository.

Table 4.3. Deploying HDP - Option I

Cluster OS	HDP Repository Tarballs

RHEL/ CentOS 5.x	• **HDP Repository:**
	`wget http://public-repo-1.hortonworks.com/HDP-1.2.0/repos/centos5/HDP-1.2.0-centos5.tar.gz`
	• **HDP-Utils Repository:**
	`wget http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.15/repos/centos5/HDP-UTILS-1.1.0.15-centos5.tar.gz`
	• **Ambari Repository (Optional):**
	`wget http://public-repo-1.hortonworks.com/ambari/centos5/ambari-1.2.2.3-centos5.tar.gz`

RHEL/ CentOS 6.x	• **HDP Repository:**
	`wget http://public-repo-1.hortonworks.com/HDP-1.2.0/repos/centos6/HDP-1.2.0-centos6.tar.gz`
	• **HDP-Utils Repository (Optional):**
	`wget http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.15/repos/centos6/HDP-UTILS-1.1.0.15-centos6.tar.gz`
	• **Ambari Repository (Optional):**
	`wget http://public-repo-1.hortonworks.com/ambari/centos6/ambari-1.2.2.3-centos5.tar.gz`

SLES 11	• **HDP Repository:**
	`wget http://public-repo-1.hortonworks.com/HDP-1.2.0/repos/suse11/HDP-1.2.0-suse11.tar.gz`
	• **HDP-Utils Repository (Optional):**
	`wget http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.15/repos/suse11/HDP-UTILS-1.1.0.15-suse11.tar.gz`
	• **Ambari Repository (Optional):**
	`wget http://public-repo-1.hortonworks.com/ambari/suse11/ambari-1.2.2.3-centos5.tar.gz`

Note

The EPEL repository is not available as a tarball currently. Use one of Options II through IV to provide access to the EPEL repository.

2. Create an HTTP server.

   - On the mirror server, install an HTTP server (such as Apache httpd) using the instructions provided here.
   - Activate this web server.
   - Ensure that the firewall settings (if any) allow inbound HTTP access from your cluster nodes to your mirror server.

Note

If you are using EC2, make sure that SELinux is disabled.

3. On your mirror server, create a directory for your web server.

- For example, from a shell window, type:
  - **For RHEL/CentOS:** `mkdir –p /var/www/html/hdp/`
  - **For SLES:** `mkdir –p /srv/www/htdocs/rpms`
- If you are using a symlink, enable the `followsymlinks` on your web server.

4. Copy the HDP Repository Tarballs to the directory created in step 3, and untar these tarballs.
5. Verify the configuration.
   - The configuration is successful, if you can access the above directory through your web browser.

     To test this out, browse to the following URLs:
     - **HDP Repository:**

       `http://yourwebserver/hdp/HDP-1.2.0/repos/$os`
     - **HDP-Utils Repository (Optional):**

       `http://yourwebserver/hdp/HDP-UTILS-1.1.0.15/repos/$os`
     - **Ambari Repository (Optional):**

       `http://yourwebserver/hdp/ambari/$os/1.x/GA`

     where, *$os* can be centos5, centos6, or suse11.

     For each repository, you should see directory listing for the HDP or HDP-UTILS, or Ambari components along with the RPMs.
6. Configure the `yum` or `zypper` clients on all the nodes in your cluster.
   a. Fetch the yum configuration file from your mirror server.
      - **HDP Repository:**

        `http://yourwebserver/hdp/HDP-1.2.0/repos/$os/hdp.repo`
      - **HDP-Utils Repository (Optional):**

        `http://yourwebserver/hdp/HDP-UTILS-1.1.0.15/repos/$os/hdp-utils.repo`
      - **Ambari Repository (Optional):**

        `http://yourwebserver/hdp/ambari/$os/1.x/GA`

      where, *$os* can be centos5, centos6, or suse11.

   b. Store the repo file(s) in a temporary location.

   c. Edit the repo file(s), changing the value of the **baseurl** property to the local mirror URL.

- Edit the `/etc/yum.repos.d/hdp.repo` file changing the **baseurl** property as shown below:

```
http://yourwebserver/hdp/HDP-1.2.0/repos/$os/hdp.repo
```

- Edit the `/etc/yum.repos.d/hdp-utils.repo` file changing the **baseurl** property as shown below:

```
http://yourwebserver/hdp/HDP-UTILS-1.1.0.15/repos/$os/hdp-utils.repo
```

- Edit the `/etc/yum.repos.d/ambari.repo` file changing the **baseurl** property as shown below:

```
[ambari-1.x]
name=Ambari 1.x
baseurl=http://yourwebserver/hdp/ambari/$os/1.x/GA/ambari.repo
gpgcheck=1
gpgkey=http://public-repo-1.hortonworks.com/ambari/centos5/RPM-GPG-KEY/
RPM-GPG-KEY-Jenkins
enabled=1
priority=1

[HDP-UTILS-1.1.0.15]
name=Hortonworks Data Platform Utils Version - HDP-UTILS-1.1.0.15
baseurl=http://yourwebserver/HDP-UTILS-1.1.0.15/repos/centos5
gpgcheck=0
gpgkey=http://public-repo-1.hortonworks.com/ambari/centos5/RPM-GPG-KEY/
RPM-GPG-KEY-Jenkins
enabled=1
priority=1

[Updates-ambari-1.x]
name=ambari-1.x - Updates
baseurl=http://yourwebserver/ambari/centos5/1.x/updates
gpgcheck=1
gpgkey=http://public-repo-1.hortonworks.com/ambari/centos5/RPM-GPG-KEY/
RPM-GPG-KEY-Jenkins
enabled=1
priority=1
```

where, *$os* can be centos5, centos6, or suse11.

d. Copy the yum/zypper client configuration file to all nodes in your cluster.

- **For RHEL and CentOS:** Use **scp** or **pdsh** to copy the client yum configuration file to `/etc/yum.repos.d/` directory on every node in the cluster.

- **For SLES:**

  i. Store the repo file(s) back into the repository location in the web server.

  - Store the `/etc/yum.repos.d/hdp.repo` file:

```
cp /tmp/hdp.repo /srv/www/htdocs/rpms/hdp/HDP-1.2.0/repos/suse11
```

  - Store the `/etc/yum.repos.d/hdp-utils.repo` file:

```
cp /tmp/hdp-utils.repo /srv/www/htdocs/rpms/hdp/HDP-UTILS-1.1.0.15/
repos/suse11
```

- Store the `/etc/yum.repos.d/ambari.repo` file:

```
cp /tmp/ambari.repo /srv/www/htdocs/rpms/hdp/ambari/suse11/1.x/GA
```

ii. On every node, invoke the following command:

- **HDP Repository:**

```
zypper addrepo -r http://yourwebserver/hdp/HDP-1.2.0/repos/suse11/
hdp.repo
```

- **HDP-Utils Repository (Optional):**

```
zypper addrepo -r http://yourwebserver/hdp/HDP-UTILS-1.1.0.15/
repos/suse11/hdp-utils.repo
```

- **Ambari Repository (Optional):**

```
zypper addrepo -r http://yourwebserver/hdp/ambari/suse11/1.x/GA/
ambari.repo
```

7. If your cluster runs CentOS or RHEL, and if you have multiple repositories configured in your environment, deploy the following plugin on all the nodes in your cluster.

   a. Install the plugin.

   - **For RHEL and CentOS v5.x**

```
yum install yum-priorities
```

   - **For RHEL and CentOS v6.x**

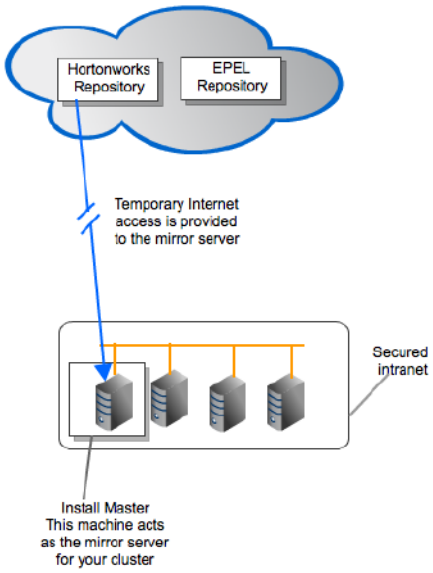
```
yum install yum-plugin-priorities
```

   b. Edit the `/etc/yum/pluginconf.d/priorities.conf` file to add the following:

```
[main]
enabled=1
gpgcheck=0
```

4.3.2. Option II - Mirror server has temporary access to the Internet

The local mirror setup for Option II is shown in the following illustration:



4.3.2.1. Prerequisites

To configure local repositories for HDP deployment, your system must meet the following minimum requirements:

- Your mirror server host must run on one of the following operating systems:
  - 64 bit Red Hat Enterprise Linux (RHEL)
  - 64 bit CentOS v5.x, v6.x
  - 64 bit SUSE Linux Enterprise Server (SLES) 11 SP1

  Ensure that this server and the cluster nodes are all running the same OS.

  Note

  To support repository mirroring for heterogeneous clusters requires a more complex procedure than the one documented here.
- The mirror server host must have several GB of storage available.
- The firewall allows all cluster nodes (the servers on which you want to install HDP) to access this server.
- Ensure that the mirror server has **yum** installed.
- Add the **yum-utils** and **createrepo** packages on the mirror server.

```
yum install yum-utils createrepo
```

- If your mirror host uses SLES, execute the following command to install the required packages:

```
zypper -n --no-gpg-checks install apache2 apache2-prefork apache2-utils
 python-curl python-gobject2 python-gpgme python-urlgrabber python-iniparse
 wget curl yum-metadata-parser yum yum-updatesd yum-utils createrepo
```

4.3.2.2. Instructions

1. Temporarily reconfigure your firewall to allow Internet access from your mirror server host.

2. Execute the following command to download the appropriate Hortonworks yum client configuration file and save it in `/etc/yum.repos.d/` directory on the mirror server host.

Table 4.4. Deploying HDP - Option II

Cluster OS	HDP Repository Tarballs

RHEL/ CentOS 5.x	• **HDP Repository:**
	`wget http://public-repo-1.hortonworks.com/HDP-1.2.0/repos/centos5/hdp.repo -O /etc/yum.repos.d/hdp.repo`
	• **HDP-Utils Repository (Optional):**
	`wget http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.15/repos/centos5/hdp-util.repo -O /etc/yum.repos.d/hdp-util.repo`
	• **Ambari Repository (Optional):**
	`wget http://public-repo-1.hortonworks.com/ambari/centos5/1.x/GA/ambari.repo -O /etc/yum.repos.d/ambari.repo`

RHEL/ CentOS 6.x	• **HDP Repository:**
	`wget http://public-repo-1.hortonworks.com/HDP-1.2.0/repos/centos6/hdp.repo -O /etc/yum.repos.d/hdp.repo`
	• **HDP-Utils Repository (Optional):**
	`wget http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.15/repos/centos6/hdp-util.repo -O /etc/yum.repos.d/hdp-util.repo`
	• **Ambari Repository (Optional):**
	`wget http://public-repo-1.hortonworks.com/ambari/centos6/1.x/GA/ambari.repo -O /etc/yum.repos.d/ambari.repo`

SLES 11	• **HDP Repository:**
	`wget http://public-repo-1.hortonworks.com/HDP-1.2.0/repos/suse11/hdp.repo -O /etc/zypp/repos.d/hdp.repo`
	• **HDP-Utils Repository (Optional):**
	`wget http://public-repo-1.hortonworks.com/HDP-UTILS-1.1.0.15/repos/suse11/hdp-util.repo -O /etc/zypp/repos.d/hdp-util.repo`
	• **Ambari Repository (Optional):**
	`wget http://public-repo-1.hortonworks.com/ambari/suse11/1.x/GA/ambari.repo -O /etc/zypp/repos.d/ambari.repo`

Note

If you are using Ambari to perform the HDP installation, you will need to setup the Ambari repository using the information provided above.

3. Create an HTTP server.

   - On the mirror server, install an HTTP server (such as Apache `httpd`) using the instructions provided here.

- Activate this web server.
- Ensure that the firewall settings (if any) allow inbound HTTP access from your cluster nodes to your mirror server.

Note

If you are using EC2, make sure that SELinux is disabled.

- [Optional]: If your mirror server uses SLES, modify the `default-server.conf` file to enable the docs root folder listing.

```
sed -e "s/Options None/Options Indexes MultiViews/ig" /etc/apache2/
default-server.conf > /tmp/tempfile.tmp
mv /tmp/tempfile.tmp /etc/apache2/default-server.conf
```

4. On your mirror server, create a directory for your web server.
   - For example, from a shell window, type:
     - **For RHEL/CentOS:** `mkdir –p` *`/var/www/html/hdp/`*
     - **For SLES:** `mkdir –p` *`/srv/www/htdocs/rpms`*
   - If you are using a symlink, enable the `followsymlinks` on your web server.
5. Copy the contents of entire HDP repository from the remote yum server to your local mirror server.

   Continuing the previous example, from a shell window, type:

   - **For RHEL/CentOS:**
     - **HDP Repository:**

```
cd /var/www/html/hdp
reposync -r HDP-1.2.0
```

     - **HDP-Utils Repository (Optional):**

```
cd /var/www/html/hdp
reposync -r HDP-UTILS-1.1.0.15
```

     - **Ambari Repository (Optional):**

```
cd /var/www/html/hdp
reposync –r ambari-1.x
reposync –r Updates-ambari-1.x
```

   - **For SLES:**

     HDP Repository:

```
cd /srv/www/htdocs/rpms
reposync -r HDP-1.2.0
```

- **HDP-Utils Repository (Optional):**

```
cd /srv/www/htdocs/rpms
reposync -r HDP-UTILS-1.1.0.15
```

- **Ambari Repository (Optional):**

```
cd /srv/www/htdocs/rpms
reposync –r ambari-1.x
reposync –r Updates-ambari-1.x
```

6. Generate appropriate metadata.

   This step defines each directory as a yum repository.

   From a shell window, type:

   - **For RHEL/CentOS:**
     - **HDP Repository:**

```
createrepo /var/www/html/hdp/HDP-1.2.0
```

     - **HDP-Utils Repository (Optional):**

```
createrepo /var/www/html/hdp/HDP-UTILS-1.1.0.15
```

     - **Ambari Repository (Optional):**

```
createrepo /var/www/html/hdp/ambari-1.x
createrepo /var/www/html/hdp/Updates-ambari-1.x
```

   - **For SLES:**

     HDP Repository:

```
createrepo /srv/www/htdocs/rpms/hdp/HDP-1.2.0
```

     - **HDP-Utils Repository (Optional):**

```
createrepo /srv/www/htdocs/rpms/hdp/HDP-UTILS-1.1.0.15
```

     - **Ambari Repository (Optional):**

```
createrepo /srv/www/htdocs/rpms/hdp/ambari-1.x
createrepo/srv/www/htdocs/rpms/hdp/Updates-ambari-1.x
```

   You should see a new folder called `repodata` inside the HDP directories.

   If using Ambari, you should also see the `repodata` directory under the `ambari-1.x` and `Updates-ambari-1.x` directories.

7. Verify the configuration.

   - The configuration is successful, if you can access the above directory through your web browser.

     To test this out, browse to the following URLs:

- **HDP Repository:**

```
http://yourwebserver/hdp/HDP-1.2.0/repos/$os
```

- **HDP-Utils Repository (Optional):**

```
http://yourwebserver/hdp/HDP-UTILS-1.1.0.15/repos/$os
```

- **Ambari Repository (Optional):**

```
http://yourwebserver/hdp/ambari/$os/1.x/GA
```

where, *$os* can be centos5, centos6, or suse11.

- You should now see directory listing for all the HDP components.

8. At this point, it is okay to disable external Internet access for the mirror server, so that the mirror server is once again entirely within your data center firewall.

9. Configure the `yum` or `zypper` clients on all the nodes in your cluster.

   a. Edit the repo file(s), changing the value of the **baseurl** property to the local mirror URL.

   - Edit the `/etc/yum.repos.d/hdp.repo` file changing the **baseurl** property as shown below:

```
http://yourwebserver/hdp/HDP-1.2.0/repos/$os/hdp.repo
```

   - Edit the `/etc/yum.repos.d/hdp-utils.repo` file changing the **baseurl** property as shown below:

```
http://yourwebserver/hdp/HDP-UTILS-1.1.0.15/repos/$os/hdp-utils.repo
```

   - Edit the `/etc/yum.repos.d/ambari.repo` file changing the **baseurl** property as shown below:

```
[ambari-1.x]
name=Ambari 1.x
baseurl=http://yourwebserver/hdp/ambari/$os/1.x/GA/ambari.repo
gpgcheck=1
gpgkey=http://public-repo-1.hortonworks.com/ambari/centos5/RPM-GPG-KEY/
RPM-GPG-KEY-Jenkins
enabled=1
priority=1

[HDP-UTILS-1.1.0.15]
name=Hortonworks Data Platform Utils Version - HDP-UTILS-1.1.0.15
baseurl=http://yourwebserver/HDP-UTILS-1.1.0.15/repos/centos5
gpgcheck=0
gpgkey=http://public-repo-1.hortonworks.com/ambari/centos5/RPM-GPG-KEY/
RPM-GPG-KEY-Jenkins
enabled=1
priority=1

[Updates-ambari-1.x]
name=ambari-1.x - Updates
baseurl=http://yourwebserver/ambari/centos5/1.x/updates
gpgcheck=1
```

```
gpgkey=http://public-repo-1.hortonworks.com/ambari/centos5/RPM-GPG-KEY/
RPM-GPG-KEY-Jenkins
enabled=1
priority=1
```

where, *$os* can be centos5, centos6, or suse11.

b. Copy the yum/zypper client configuration file to all nodes in your cluster.

- **For RHEL and CentOS:** Use **scp** or **pdsh** to copy the client yum configuration file to `/etc/yum.repos.d/` directory on every node in the cluster.
- **For SLES:** On every node, invoke the following command:
  - **HDP Repository:**

    ```
    zypper addrepo -r http://yourwebserver/hdp/HDP-1.2.0/repos/suse11/
    hdp.repo
    ```

  - **HDP-Utils Repository (Optional):**

    ```
    zypper addrepo -r http://yourwebserver/hdp/HDP-UTILS-1.1.0.15/repos/
    suse11/hdp-utils.repo
    ```

  - **Ambari Repository (Optional):**

    ```
    zypper addrepo -r http://yourwebserver/hdp/ambari/suse11/1.x/GA/
    ambari.repo
    ```

- If using Ambari, verify the configuration by deploying Ambari server on one of the cluster nodes.

  ```
  yum install ambari-server
  ```

10. If your cluster runs CentOS or RHEL, and if you have multiple repositories configured in your environment, deploy the following plugin on all the nodes in your cluster.

a. Install the plugin.

- **For RHEL and CentOS v5.x**

  ```
  yum install yum-priorities
  ```

- **For RHEL and CentOS v6.x**

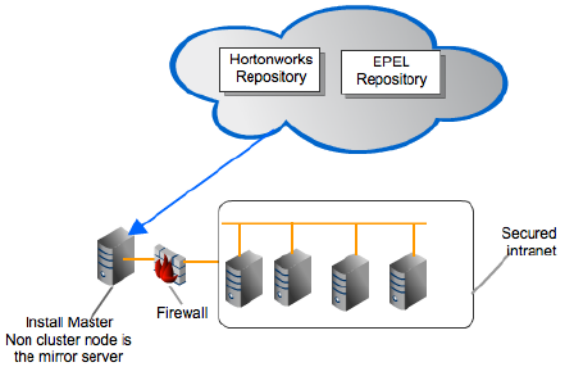
  ```
  yum install yum-plugin-priorities
  ```

b. Edit the `/etc/yum/pluginconf.d/priorities.conf` file to add the following:

```
[main]
enabled=1
gpgcheck=0
```

4.3.3. Option III - Mirror server has permanent access to the Internet

The local mirror setup for Option III is shown in the following illustration:



4.3.3.1. Prerequisites

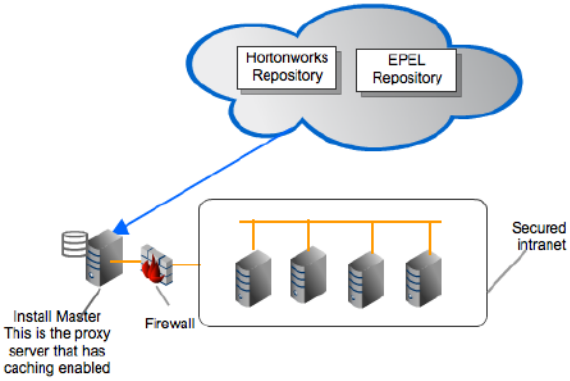
Same as Option II.

4.3.3.2. Instructions

Same as Option II but Step 1 and 8 are unnecessary and may be skipped.

4.3.4. Option IV - Trusted proxy server

The local mirror setup for Option IV is shown in the following illustration:



4.3.4.1. Prerequisites

Select a mirror server host with the following characteristics:

- This server runs on either CentOS (v5.x, v6.x) or RHEL (v5.x, v6.x) and has several GB of storage available.
- The firewall allows all cluster nodes (the servers on which you want to install HDP) to access this server, and allows this server to access the Internet (at least those Internet servers for the repositories to be proxied).

4.3.4.2. Instructions

1. Create a caching HTTP PROXY server on the selected host.

a. It is beyond the scope of this document to show how to set up an HTTP PROXY server, given the many variations that may be required, depending on your data center's network security policy. If you choose to use the Apache HTTPD server, it starts by installing **httpd**, using the instructions provided here, and then adding the **mod_proxy** and **mod_cache modules**, as stated here. Please engage your network security specialists to correctly set up the proxy server.

b. Activate this proxy server and configure its cache storage location.

c. Ensure that the firewall settings (if any) allow inbound HTTP access from your cluster nodes to your mirror server, and outbound access to the desired repo sites, including **public-repo-1.hortonworks.com**.

Note

If you are using EC2, make sure that SELinux is disabled.

2. Depending on your cluster OS, configure the `yum` or `zypper` clients on all the nodes in your cluster.

- **For RHEL and CentOS:**

Note

The following description is taken from the CentOS documentation here.

a. On each cluster node, add the following lines to the **/etc/yum.conf** file.

(As an example, the settings below will enable **yum** to use the proxy server **mycache.mydomain.com**, connecting to port **3128**, with the following credentials **yum-user/qwerty**.

```
# proxy server:port number
proxy=http://mycache.mydomain.com:3128
```

```
# account details for secure yum proxy connections
proxy_username=yum-user
proxy_password=qwerty
```

b. Once all nodes have their **/etc/yum.conf** file updated with appropriate configuration info, you can proceed with the HDP installation just as though the nodes had direct access to the Internet repositories.

c. If this proxy configuration does not seem to work, try adding a **/** at the end of the proxy URL. For example:

```
proxy=http://mycache.mydomain.com:3128/
```

- **For SLES:**

a. Configure the zypper clients on all the nodes in your cluster, to use the proxy server.

b. Please consult with your system administrator to do this correctly in your environment.

5. Installing the JDK

HDP requires that the Java SE Development Kit (JDK) v1.6 update 31 or later must be installed on all the nodes in your cluster. Follow the instructions listed below to deploy the JDK manually:

1. Obtain the version of JDK currently installed on your host machine.

```
java -version
```

2. Uninstall the Java package if JDK version is less than v 1.6 update 31.

```
rpm -qa | grep java
yum remove java-1.x.0-jdk-1.x.0.0-1.45.1.11.1.el6.x86_64
```

Step 3: Verify that the default Java package is uninstalled.

```
which java
```

3. Download Oracle JDK jdk-6u31-linux-x64.binon all the host machines.

4. Change directory to the location where you downloaded the JDK and run the install.

```
mkdir /usr/jdk1.6.0_31
cd /usr/jdk1.6.0_31
chmod u+x $JDK_download_directory/jdk-6u31-linux-x64.
bin
$JDK_download_directory/jdk-6u31-linux-x64.bin -noregister
```

5. Create symbolic links (symlinks) to the JDK.

```
mkdir /usr/java
ln -s /usr/jdk1.6.0_31/jdk1.6.0_31 /usr/java/default
ln -s /usr/java/default/bin/java /usr/bin/java
```

6. Set up your environment to define JAVA_HOME to put the Java Virtual Machine and the Java compiler on your path.

```
export JAVA_HOME=/usr/java/default
export PATH=$JAVA_HOME/bin:$PATH
```

6. Manually Add Slave Nodes to HDP Cluster

This section provides instructions to manually add slave nodes to your HDP cluster.

6.1. Prerequisites

Ensure that the new slave nodes meet the following prerequisites:

- The following operating systems are supported:
  - 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
- On each of your hosts:
  - yum (RHEL)
  - zypper (SLES)
  - rpm
  - scp
  - curl
  - wget
  - unzip
  - tar
  - pdsh
- Ensure that all the ports listed here are available to the Installer.
- To install Hive metastore or to use external database for Oozie metastore, ensure that you deploy either a MySQL or an Oracle database in your cluster. For instructions, see here.
- Your system must have the correct JDK installed on all the nodes of the cluster. HDP requires Oracle JDK 1.6 update 31. For more information, see Install the Java Development Kit.

6.2. Add DataNodes or TaskTrackers

Use the following instructions to manually add a DataNode or a TaskTracker hosts:

1. On each of the newly added slave nodes, add the HDP repository to yum:

```
wget -nv http://public-repo-1.hortonworks.com/HDP-1.2.0/repos/centos6/hdp.
repo -O
/etc/yum.repos.d/hdp.repo
yum clean all
```

2. On each of the newly added slave nodes, install HDFS and MapReduce.

   - On RHEL and CentOS:

```
yum install hadoop hadoop-libhdfs hadoop-native
yum install hadoop-pipes hadoop-sbin openssl
```

   - On SLES:

```
zypper install hadoop hadoop-libhdfs hadoop-native
zypper install hadoop-pipes hadoop-sbin openssl
```

3. On each of the newly added slave nodes, install Snappy compression/decompression library:

   a. Check if Snappy is already installed:

```
rpm-qa | grep snappy
```

   b. Install Snappy on the new nodes:

      - For RHEL/CentOS:

```
yum install snappy snappy-devel
```

      - For SLES:

```
zypper install snappy snappy-devel
```

```
ln -sf /usr/lib64/libsnappy.so
/usr/lib/hadoop/lib/native/Linux-amd64-64/.
```

4. Optional - Install the LZO compression library.

   - On RHEL and CentOS:

```
yum install lzo-devel hadoop-lzo-native
```

   - On SLES:

```
zypper install lzo-devel hadoop-lzo-native
```

5. Copy the Hadoop configurations to the newly added slave nodes and set appropriate permissions.

   - **Option I:** Copy Hadoop config files from an existing slave node.

     a. On an existing slave node, make a copy of the current configurations:

```
tar zcvf hadoop_conf.tgz /etc/hadoop/conf
```

     b. Copy this file to each of the new nodes:

```
rm -rf /etc/hadoop/conf
cd /
tar zxvf $location_of_copied_conf_tar_file/hadoop_conf.tgz
chmod -R 755 /etc/hadoop/conf
```

- **Option II:** Manually add Hadoop configuration files.

  a. Download core Hadoop configuration files from here and extract the files under `configuration_files -> core_hadoop` directory to a temporary location.

  b. In the temporary directory, locate the following files and modify the properties based on your environment. Search for TODO in the files for the properties to replace.

Table 6.1. core-site.xml

Property	Example	Description
fs.default.name	`hdfs://{namenode.full.host-name}:8020`	Enter your NameNode hostname
fs.checkpoint.dir	`/grid/hadoop/hdfs/snn`	A comma separated list of paths. Use the list of directories from `$FS_CHECKPOINT_DIR.` .

Table 6.2. hdfs-site.xml

Property	Example	Description
dfs.name.dir	`/grid/hadoop/hdfs/nn,/grid1/hadoop/hdfs/nn`	Comma separated list of paths. Use the list of directories from `$DFS_NAME_DIR`
dfs.data.dir	`/grid/hadoop/hdfs/dn,grid1/hadoop/hdfs/dn`	Comma separated list of paths. Use the list of directories from `$DFS_DATA_DIR`
dfs.http.address	`{namenode.full.hostname}:50070`	Enter your NameNode hostname for http access
dfs.secondary.http.address	`{secondary.namenode.full.host-name}:50090`	Enter your SecondaryNameNode hostname
dfs.https.address	`{namenode.full.hostname}:50470`	Enter your NameNode hostname for https access.

Table 6.3. mapred-site.xml

Property	Example	Description
mapred.job.tracker	`{jobtracker.full.host-name}:50300`	Enter your JobTracker hostname
mapred.job.tracker.http.address	`{jobtracker.full.host-name}:50030`	Enter your JobTracker hostname
mapred.local.dir	`/grid/hadoop/mapred,/grid1/hadoop/mapred`	Comma separated list of paths. Use the list of directories from `$MAPREDUCE_LOCAL_DIR`
mapreduce.tasktracker.group	hadoop	Enter your group. Use the value of `$HADOOP_GROUP`
mapreduce.history.server.http.address	`{jobtracker.full.hostname}:51111`	Enter your JobTracker hostname

Table 6.4. taskcontroller.cfg

Property	Example	Description
mapred.local.dir	`/grid/hadoop/mapred,/grid1/hadoop/mapred`	Comma separated list of paths. Use the list of directories from *$MAPREDUCE_LOCAL_DIR*

c. Create the config directory on all hosts in your cluster, copy in all the configuration files, and set permissions.

```
rm -r $HADOOP_CONF_DIR
mkdir -p $HADOOP_CONF_DIR
```

```
 <copy the all the config files to $HADOOP_CONF_DIR>
```

```
chmod a+x $HADOOP_CONF_DIR/
chown -R $HDFS_USER:$HADOOP_GROUP $HADOOP_CONF_DIR/../
chmod -R 755 $HADOOP_CONF_DIR/../
```

6. On each of the newly added slave nodes, start HDFS:

```
su -hdfs
/usr/lib/hadoop/bin/hadoop-daemon.sh --config
$HADOOP_CONF_DIR start datanode
```

7. On each of the newly added slave nodes, start MapReduce:

```
su -mapred
/usr/lib/hadoop/bin/hadoop-daemon.sh --config
$HADOOP_CONF_DIR start tasktracker
```

8. Add new slave nodes.

- To add a new NameNode slave (DataNode):

a. On the NameNode host machine, edit the `/etc/hadoop/conf/dfs.include` file and add the list of slave nodes' hostnames (separated by newline character).

Important

Ensure that you create a new `dfs.include` file, if the NameNode host machine does not have an existing copy of this file.

b. On the NameNode host machine, execute the following command:

```
su – hdfs –c “hadoop dfsadmin –refreshNodes”
```

- To add a new JobTracker slave (TaskTracker):

a. One the JobTracker host machine, edit the `/etc/hadoop/conf/mapred.include` file and add the list of slave nodes' hostnames (separated by newline character).

Important

Ensure that you create a new `mapred.include` file, if the JobTracker host machine does not have an existing copy of this file.

b. On the JobTracker host machine, execute the following command:

```
su – mapred –c “hadoop mradmin –refreshNodes”
```

9. Optional - Enable monitoring on the newly added slave nodes using the instructions provided here.

10. Optional - Enable cluster alerting on the newly added slave nodes using the instructions provided here.

6.3. Adding HBase RegionServer

Use the following instructions to manually add HBase RegionServer hosts:

1. On each of the newly added slave nodes, install HBase and ZooKeeper.

   - For RHEL/CentOS:

     ```
     yum install zookeeper hbase
     ```

   - For SLES:

     ```
     zypper install zookeeper hbase
     ```

2. On each of the newly added slave nodes, add the HDP repository to yum:

   ```
   wget -nv http://public-repo-1.hortonworks.com/HDP-1.2.0/repos/centos6/hdp.
   repo -O
   /etc/yum.repos.d/hdp.repo
   yum clean all
   ```

3. Copy the HBase configurations to the newly added slave nodes and set appropriate permissions.

   - **Option I:** Copy HBase config files from an existing slave node.

     a. On any existing slave node, make a copy of the current configurations:

     ```
     tar zcvf hbase_conf.tgz /etc/hbase/conf
     tar zcvf zookeeper_conf.tgz /etc/zookeeper/conf
     ```

     b. Copy this file to each of the new nodes:

     ```
     rm -rf /etc/hbase/conf
     mkdir -p /etc/hbase/conf
     cd /
     tar zxvf $location_of_copied_conf_tar_file/hbase_conf.tgz
     chmod -R 755 /etc/hbase/conf
     ```

```
rm -rf /etc/zookeeper/conf
mkdir -p /etc/zookeeper/conf
cd /
tar zxvf $location_of_copied_conf_tar_file/zookeeper_conf.tgz
chmod -R 755 /etc/zookeeper/conf
```

- **Option II:** Manually add Hadoop configuration files.

  a. Download the HBase/ZooKeeper config files from here and extract these files under `configuration_files -> hbase` and `configuration_files -> zookeeper` directories to two temporary locations.

  b. Modify the configuration files:

Table 6.5. zoo.cfg

Variable	Example	Description
server.1	*$zookeeper.server1.full.hostname* :2888:3888	Enter the 1st ZooKeeper hostname
server.2	*$zookeeper.server1.full.hostname* :2888:3888	Enter the 2nd ZooKeeper hostname
server.3	*$zookeeper.server3.full.hostname* :2888:3888	Enter the 3rd ZooKeeper hostname

Table 6.6. hbase-site.xml

Variable	Example	Description
hbase.rootdir	hdfs:// {$namenode.full.hostname}:8020/ apps/hbase/data	Enter the HBase NameNode server
hbase.master.info.bindAddress	{hbase.master.full.hostname}	Enter the HBase Master server hostname
hbase.zookeeper.quorum	server1.full.hostname, server2.full.hostname, server3.full.hostname	Comma separated list of Zookeeper servers (match to what is specified in zoo.cfg but without portnumbers)

4. On all the new slave nodes create the config directory, copy all the config files, and set the permissions:

```
rm -r $HBASE_CONF_DIR ;
mkdir -p $HBASE_CONF_DIR ;

copy all the config files to $HBASE_CONF_DIR

chmod a+x $HBASE_CONF_DIR/;
chown -R $HBASE_USER:$HADOOP_GROUP $HBASE_CONF_DIR/../  ;
chmod -R 755 $HBASE_CONF_DIR/../

rm -r $ZOOKEEPER_CONF_DIR ;
mkdir -p $ZOOKEEPER_CONF_DIR ;

copy all the config files to $ZOOKEEPER_CONF_DIR

chmod a+x $ZOOKEEPER_CONF_DIR/;
chown -R $ZOOKEEPER_USER:$HADOOP_GROUP $ZOOKEEPER_CONF_DIR/../  ;
chmod -R 755 $ZOOKEEPER_CONF_DIR/../
```

5. Start HBase RegionServer node:

```
<login as $HBASE_USER>
/usr/lib/hbase/bin/hbase-daemon.sh --config $HBASE_CONF_DIR start
 regionserver
```

6. On the HBase Master host machine, edit the `/usr/lib/hbase/conf` file and add the list of slave nodes' hostnames (separated by newline character).

7. Optional - Enable monitoring on the newly added slave nodes using the instructions provided here.

8. Optional - Enable cluster alerting on the newly added slave nodes using the instructions provided here.

6.4. Optional - Configure Monitoring Using Ganglia

For each of the newly added slave node, complete the following instructions in order to use Ganglia for monitoring and metrics collection:

1. On each new host, install the Ganglia gmond:

   - On RHEL and CentOS:

     ```
     yum install ganglia-gmond-3.2.0-99
     ```

   - On SLES:

     ```
     zypper install ganglia-gmond-3.2.0-99
     ```

2. Copy the HDP Ganglia scripts from any existing slave node to the newly added node:

```
mkdir -p /usr/libexec/hdp/ganglia
scp root@$FQDN_of_any_existing_node:/usr/libexec/hdp/ganglia/*
/usr/libexec/hdp/ganglia
```

3. Configure the gmond emitter:

```
/usr/libexec/hdp/ganglia/setupGanglia.sh -c HDPSlaves
```

4. Edit the `/etc/ganglia/hdp/HDPSlaves/conf.d/gmond.slave.conf`to validate the slave configuration files.

   Ensure that `host` points to the Ganglia server master node as shown below:

```
host = $Ganglia_server_hostname
```

5. Copy Ganglia service start file from any existing slave node to each of the newly added node:

```
scp root@$existing_node:/etc/init.d/hdp-gmond /etc/init.d
/etc/init.d/hdp-gmond start
```

   This step will start Ganglia monitoring for your Hadoop cluster.

6.5. Optional - Configure Cluster Alerting Using Nagios

For each of the newly added slave node, complete the following instructions in order to use Nagios for cluster alerting:

1. On the Nagios server, add the host definition to the `/etc/nagios/objects/hadoop-hosts.cfg` file:
2. On the Nagios server, edit the `/etc/nagios/objects/hadoop-hostsgroups.cfg` file.

   Add hostnames of the newly added slave nodes to members list under the `slaves` host group.
3. Restart the Nagios server:

   ```
   service nagios start
   ```

7. Supported Database Matrix for Hortonworks Data Platform

This page contains certification information on supported databases for Hortonworks Data Platform (HDP).

The following table identifies the supported databases for HDP.

Table 7.1. Supported Databases

Operating System	Component	Database			
		Postgres 8.x	**MySQL 5.x**	**Oracle 11gr2**	**Other**
RHEL/ CentOS 5.x, 6.x. SLES	Hive / HCatalog		Default. For instructions on configuring this database for Hive metastore, see: Installing HDP manually or Installing HDP using shell scripts.	Supported. For instructions on configuring this database for Hive metastore, see: Database requirements for installing HDP manually or Database requirements for installing HDP using shell scripts.	
	Oozie		Supported. For instructions on configuring this database for Oozie metastore, see: Installing HDP manually or Installing HDP using shell scripts.	Supported. For instructions on configuring this database for Oozie metastore, see: Installing HDP manually or Installing HDP using shell scripts.	Derby (default).
	Ambari	Default.			

8. Appendix: Tarballs

Individual links to the Apache structured tarball files for the projects included with Hortonworks Data Platform are listed in the following sections.

8.1. RHEL 5 and CentOS 5

Table 8.1. RHEL/CentOS 5

Project	Download
Hadoop	hadoop-1.1.2.21.tar.gz
Pig	pig-0.10.1.21.tar.gz
Hive and HCatalog	hive-0.10.0.21.tar.gz hcatalog-0.5.0.21.tar.gz
Oozie	oozie-3.2.0.21-distro.tar.gz
HBase and ZooKeeper	hbase-0.94.2.21.tar.gz zookeeper-3.4.5.21.tar.gz
Sqoop	sqoop-1.4.2.21.bin__hadoop-1.1.2.21.tar.gz
Flume	apache-flume-1.3.0.21-bin.tar.gz
Mahout	mahout-distribution-0.7.0.21.tar.gz

8.2. RHEL 6 and CentOS 6

Table 8.2. RHEL/CentOS 6

Project	Download
Hadoop	hadoop-1.1.2.21.tar.gz
Pig	pig-0.10.1.21.tar.gz
Hive and HCatalog	hive-0.10.0.21.tar.gz hcatalog-0.5.0.21.tar.gz
Oozie	oozie-3.2.0.21-distro.tar.gz
HBase and ZooKeeper	hbase-0.94.2.21.tar.gz zookeeper-3.4.5.21.tar.gz
Sqoop	sqoop-1.4.2.21.bin__hadoop-1.1.2.21.tar.gz
Flume	apache-flume-1.3.0.21-bin.tar.gz
Mahout	mahout-distribution-0.7.0.21.tar.gz

8.3. SUSE Enterprise Linux 11

Table 8.3. SLES 11

Project	Download
Hadoop	hadoop-1.1.2.21.tar.gz
Pig	pig-0.10.1.21.tar.gz
Hive and HCatalog	hive-0.10.0.21.tar.gz

Project	Download
	hcatalog-0.5.0.21.tar.gz
Oozie	oozie-3.2.0.21-distro.tar.gz
HBase and ZooKeeper	hbase-0.94.2.21.tar.gz zookeeper-3.4.5.21.tar.gz
Sqoop	sqoop-1.4.2.21.bin__hadoop-1.1.2.21.tar.gz
Flume	apache-flume-1.3.0.21-bin.tar.gz
Mahout	mahout-distribution-0.7.0.21.tar.gz