Hortonworks Data Platform 2.0

Installing HDP Manually

(Mar 18, 2013)

docs.hortonworks.com

Hortonworks Data Platform 2.0 : Installing HDP Manually

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.

Hortonworks Data Platform (HDP) and any of its components are not anticipated to be combined with any hardware, software or data, except as expressly recommended in this documentation.

Table of Contents

List of Tables

1. Getting Ready to Install

This section describes the information and materials you need to get ready to install the Hortonworks Data Platform (HDP) manually. Use the following instructions to prepare your cluster before deploying HDP:

1. Understand the Basics
2. Meet Minimum System Requirements
3. Decide on Deployment Type
4. Collect Information
5. Prepare the Environment
6. Configure the Local Repositories
7. Optional - Install MySQL
8. Download Companion Files
9. Create System Users and Groups
10. Define Environment Parameters

1.1. Understand the Basics

The Hortonworks Data Platform consists of three layers.

- **Core Hadoop**:The basic components of Apache Hadoop.
  - **Hadoop Distributed File System (HDFS)**: A special purpose file system that is designed to work with the MapReduce engine. It provides high-throughput access to data in a highly distributed environment.
  - **Apache Hadoop YARN**: YARN is a general-purpose, distributed, application management framework that supersedes the classic Apache Hadoop MapReduce framework for processing data in Hadoop clusters. The fundamental idea of YARN is to split up the two major responsibilities of the JobTracker i.e. resource management and job scheduling/monitoring, into separate daemons: a global **ResourceManager** and per-application **ApplicationMaster** (AM). The ResourceManager and per-node slave, the **NodeManager** (NM), form the new, and generic, system for managing applications in a distributed manner. The ResourceManager is the ultimate authority that arbitrates resources among all the applications in the system. The per-application ApplicationMaster is, in effect, a framework specific entity and is tasked with negotiating resources from the ResourceManager and working with the NodeManager(s) to execute and monitor the component tasks.
  - **MapReduce**: A framework for performing high volume distributed data processing using the MapReduce programming paradigm.

- **Essential Hadoop**A set of Apache components designed to ease working with Core Hadoop.
  - **Apache Pig**: A platform for creating higher level data flow programs that can be compiled into sequences of MapReduce programs, using Pig Latin, the platform's native language.
  - **Apache Hive**: A tool for creating higher level SQL-like queries using HiveQL, the tool's native language, that can be compiled into sequences of MapReduce programs.
  - **Tez**: A general-purpose, highly customizable framework that creates simplifies data-processing tasks across both small scale (low-latency) and large-scale (high throughput) workloads in Hadoop.
  - **Apache HCatalog**: A metadata abstraction layer that insulates users and scripts from how and where data is physically stored.
  - **Apache HBase**: A distributed, column-oriented database that provides the ability to access and manipulate data randomly in the context of the large blocks that make up HDFS.
  - **Apache ZooKeeper**:A centralized tool for providing services to highly distributed systems. ZooKeeper is necessary for HBase installations.

You must always install Core Hadoop, but you can select the components from the other layers based on your needs. For more information on the structure of the HDP, see Understanding Hadoop Ecosystem.

1.2. Meet Minimum System Requirements

To run the Hortonworks Data Platform, your system must meet minimum requirements.

- Hardware Recommendations
- Operating System Requirements
- Software Requirements
- Database Requirements
- JDK Recommendations

1.2.1. Hardware Recommendations

Although there is no single hardware requirement for installing HDP, there are some basic guidelines. You can see sample setups here: Suggested Hardware for a Typical Hadoop Cluster.

1.2.2. Operating Systems Requirements

The following operating systems are supported:

- 64-bit Red Hat Enterprise Linux (RHEL) 5 or 6
- 64-bit CentOS 5 or 6

1.2.3. Software Requirements

On each of your hosts:

- yum [for RHEL or CentOS]
- rpm
- scp
- curl
- wget
- unzip
- tar
- pdsh

1.2.4. Database Requirements

By default, Hive use Derby database for its metastore. To use external database for Hive metastore, ensure that a MySQL database is deployed and available. Hive or HCatalog requires a MySQL database for its use.

- You can choose to use a current instance of MySQL or install a new instance for its use. For more information, see Install MySQL (Optional).
- Ensure that your database administrator creates the following databases and users.
    - For Hive, ensure that your database administrator creates `hive_dbname`, `hive_dbuser`, and `hive_dbpasswd`.

Note

For instructions on creating users for MySQL, see here.

1.2.5. 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:

1. Check the version. From a terminal window, type:

```
java -version
```

2. (Optional) Uninstall the Java package if the JDK version is less than v1.6 update 31.

   ```
   rpm -qa | grep java
   yum remove {java-1.*}
   ```

3. (Optional) Verify that the default Java package is uninstalled.

   ```
   which java
   ```

4. Download the Oracle 64-bit JDK (jdk-6u31-linux-x64.bin) from the Oracle download site. From your browser window, go to `http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-javase6-419409.html#jdk-6u31-oth-JPR`.

   Accept the license agreement and download `jdk-6u31-linux-x64.bin`.

   Download the JDK to a temporary directory (***$JDK_download_directory***).

5. Change directory to *$JDK_download_directory* 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
   ```

6. 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
   ```

7. 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
   ```

8. Verify if Java is installed in your environment. Execute the following from the command line console:

   ```
   java -version
   ```

   You should see the following output:

   ```
   java version "1.6.0_31"
   Java(TM) SE Runtime Environment (build 1.6.0_31-b04)
   Java HotSpot(TM) 64-Bit Server VM (build 20.6-b01, mixed mode)
   ```

1.2.6. Virtualization and Cloud Platforms

HDP is certified and supported when running on virtual or cloud platforms (for example, VMware vSphere or Amazon Web Services EC2) as long as the respective guest operating system (OS) is supported by HDP and any issues detected on these platforms are reproducible on the same supported OS installed on bare metal.

See Operating Systems Requirements for the list of supported operating systems for HDP.

1.3. Decide on Deployment Type

While it is possible to deploy all of HDP on a single host, this is appropriate only for initial evaluation. In general you should use at least three hosts: one master host and two slaves.

1.4. Collect Information

To deploy your HDP installation, you need to collect the following information:

- The fully qualified domain name (FQDN) for each host in your system, and which component(s) you wish to set up on which host. You can use `hostname -f` to check for the FQDN if you do not know it.
- The hostname (for an existing instance), database name, username, and password for the MySQL instance, if you install Hive/HCatalog.

Note

If you are using an existing instance, the dbuser you create for HDP’s use must be granted ALL PRIVILEGES on that instance.

1.5. Prepare the Environment

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

- Enable NTP on the Cluster
- Check DNS
- Disable SELinux
- Set Up Password-less SSH

1.5.1. Enable NTP on the Cluster

The clocks of all the nodes in your cluster must be able to synchronize with each other. If your system does not have access to the Internet, set up a master node as an NTP xserver. Use the following instructions to enable NTP for your cluster:

1. Configure NTP clients. Execute the following command on all the nodes in your cluster:

   ```
   yum install ntp
   ```

2. Enable the service. Execute the following command on all the nodes in your cluster:

   ```
   chkconfig ntpd on
   ```

3. Start the NTP. Execute the following command on all the nodes in your cluster:

```
/etc/init.d/ntpd start
```

4. For using existing NTP server in your environment. Configure firewall on local NTP server to enable UDP input traffic on port `123`. See the following sample rule:

   ```
   -A RH-Firewall-1-INPUT -s 192.168.1.0/24 -m state --state NEW -p udp --dport
    123 -j ACCEPT
   ```

   Restart iptables. Execute the following command on all the nodes in your cluster:

   ```
   service iptables restart
   ```

   Configure clients to use the local NTP server. Edit the `/etc/ntp.conf` and add the following line:

   ```
   server $LOCAL_SERVER_IP OR HOSTNAME
   ```

1.5.2. Check DNS

All hosts in your system must be configured for DNS and Reverse DNS.

Note

If you are unable to configure DNS and Reverse DNS, you must edit the hosts file on every host in your cluster to contain each of your hosts.

Use the following instructions to check DNS for all the host machines in your cluster:

1. Forward lookup checking.

   For example, for domain `localdomain` that contains host with name `host01` and IP address `192.168.0.10`, execute the following command:

   ```
   nslookup host01
   ```

   You should see a message similar to the following:

   ```
   Name: host01.localdomain
   Address: 192.168.0.10
   ```

2. Reverse lookup checking.

   For example, for domain `localdomain` that contains host with name `host01` and IP address `192.168.0.10`, execute the following command:

   ```
   nslookup 192.168.0.10
   ```

   You should see a message similar to the following:

   ```
    10.0.168.192.in-addr.arpa name = host01.localdomain.
   ```

If you do not receive valid responses (as shown above), you should set up DNS zone in your cluster or configure host files on each host of the cluster using one of the following options:

- **Option I:** Configure hosts file on each node of the cluster.

  For all nodes of cluster, add to the `/etc/hosts` file key-value pairs like the following:

```
192.168.0.11 host01
```

- **Option II:** Configuring DNS using BIND nameserver.

  The following instructions, use the example values given below:

  ```
  Example values:
  domain name: “localdomain”
  nameserver: “host01”/192.168.0.11
  hosts: “host02”/192.168.0.12, “host02”/192.168.0.12
  ```

  1. Install BIND packages:

     ```
     yum install bind
     yum install bind-libs
     yum install bind-utils
     ```

  2. Initiate service

     ```
     chkconfig named on
     ```

  3. Configure files. Add the following lines for the example values given above (ensure that you modify these for your environment) :

     - Edit the `/etc/resolv.conf` (for all nodes in cluster) and add the following lines:

       ```
       domain localdomain
       search localdomain
       nameserver 192.168.0.11
       ```

     - Edit the `/etc/named.conf` (for all nodes in cluster) and add the following lines:

       ```
       listen-on port 53 { any; };//by default it is opened only for localhost
        ...
       zone "localdomain" {
        type master;
        notify no;
        allow-query { any; };
        file "named-forw.zone";
        };
        zone "0.168.192.in-addr.arpa" {
         type master;
         notify no;
         allow-query { any; };
         file "named-rev.zone";
       };
       ```

     - Edit the `named-forw.zone` as shown in the following sample forward zone configuration file:

       ```
        $TTL 3D
        @ SOA   host01.localdomain.root.localdomain
        (201306030;3600;3600;3600;3600)
        NS host01            ; Nameserver Address
        localhost IN A 127.0.0.1
        host01  IN A 192.168.0.11
        host02  IN A 192.168.0.12
        host03  IN A 192.168.0.13
       ```

- Edit the `named-rev.zone` as shown in the following sample reverse zone configuration file:

  ```
  $TTL 3D
  @ SOA host01.localdomain.root.localdomain. (201306031;28800;2H;4W;1D);
  NS host01.localdomain.; Nameserver Address
  11 IN PTR host01.localdomain.
  12 IN PTR host02.localdomain.
  13 IN PTR host03.localdomain.
  ```

4. Restart bind service.

   ```
   /etc/init.d/named restart
   ```

5. Add rules to firewall.

   ```
   iptables -A INPUT -p udp -m state --state NEW --dport 53 -j ACCEPT
   iptables -A INPUT -p tcp -m state --state NEW --dport 53 -j ACCEPT
   service iptables save
   service iptables restart
   ```

   Alternatively, you can also allow traffic over DNS port (53) using `system-config-firewall` utility.

1.5.3. Disable SELinux

Security-Enhanced (SE) Linux feature should be disabled during installation process.

1. Check state of SELinux. On all the host machines, execute the following command:

   ```
   getenforce
   ```

   If the result is `permissive` or `disabled`, no further actions are required. Else, proceed to step 2.

2. Disable SELinux either temporarily for each session or permanently.

   - Option I: Disable SELinux temporarily by executing the following command:

     ```
     setenforce 0
     ```

   - Option II: Disable SELinux permanently in the `/etc/sysconfig/selinux` file by changing the value of `SELINUX` field to `permissive` or `disabled`. Restart your system.

1.5.4. Set Up Password-less SSH

To have automatically deploy HDP in all your cluster hosts, you must set up password-less SSH connections between the master installation host and all other machines.

Note

You can choose to install the HDP on each cluster host manually. In this case you do not need to setup SSH.

1. Generate public and private SSH keys on the master install machine:

   ```
   ssh-keygen
   ```

2. Copy the SSH Public Key `id_rsa.pub` to the root account on your target hosts.

   ```
   .ssh/id_rsa
   .ssh/id_rsa.pub
   ```

3. Depending on your version of SSH, you may need to set permissions on your `.ssh` directory (to `700`) and the `authorized_keys` file in that directory (to `600`).

   ```
   chmod 700 ~/.ssh
   chmod 600 ~/. ssh /authorized_keys
   ```

4. Add the SSH Public Key to the `authorized_keys` file.

   ```
   cat id_rsa.pub >> authorized_keys
   ```

5. From the master install host machine, make sure you can connect to each host in the cluster using SSH.

   ```
   ssh root@{$remote.target.host}
   ```

   You may see this warning. This happens on your first connection and is normal.

   ```
   Are you sure you want to continue connecting (yes/no)?
   ```

6. Retain a copy of the SSH Private Key on the machine from which you will run HDP Installer.

1.6. Configure the Remote Repositories

The standard 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.

Note

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 more information, see Deployment Strategies for Data Centers with Firewalls, a separate document in this set.

1. Download the yum repo configuration file `hdp.repo`. On your local mirror server, execute the following command:

   - For RHEL/CentOS 5:

     ```
     wget http://public-repo-1.hortonworks.com/HDP-2.0.0.2/repos/centos5/hdp.
     repo
     ```

   - For RHEL/CentOS 6:

     ```
     wget http://public-repo-1.hortonworks.com/HDP-2.0.0.2/repos/centos6/hdp.
     repo
     ```

2. On all hosts, copy the `hdp.repo` file you just downloaded to your yum repo list.

   ```
   cp ./hdp.repo /etc/yum.repos.d/hdp.repo
   ```

3. Confirm the HDP repository is configured by checking the repo list.

   ```
   yum repolist
   ```

   You should see something like this. Ensure that you have HDP-2.0.0.2 directory:

   ```

   Loaded plugins: fastestmirror, security
   Loading mirror speeds from cached hostfile
   * base: mirrors.cat.pdx.edu
   * extras: linux.mirrors.es.net
   * updates: mirrors.usc.edu
   repo id repo name                                          status
   HDP-2.0.0.2 Hortonworks Data Platform Version - HDP-2.0.0.2 enabled: 53
   ```

1.7. Optional - Install MySQL

If you are installing Hive and HCatalog services, you need a MySQL database instance to store metadata information. You can either use an existing MySQL instance or install a new instance of MySQL manually. To install a new instance:

1. Connect to the host machine you plan to use for Hive and HCatalog.
2. Install MySQL server. From a terminal window, type:

   ```
   yum install mysql-server
   ```

3. Start the instance.

   ```
   /etc/init.d/mysqld start
   ```

4. Set the `root` user password.

   ```
   mysqladmin -u root password ‘{password}’
   ```

5. Remove unnecessary information from log and STDOUT.

   ```
   mysqladmin -u root 2>&1 >/dev/null
   ```

6. As `root,` use mysql (or other client tool) to create the “dbuser” and grant it adequate privileges. This user provides access to the Hive metastore.

   ```
   CREATE USER '$dbusername'@'%' IDENTIFIED BY '$dbuserpassword';
   GRANT ALL PRIVILEGES ON *.* TO '$dbusername'@'%';
   flush privileges;
   ```

7. See if you can connect to the database as that user. You are prompted to enter the *`$dbuserpassword`* password above.

   ```
   mysql -u $dbusername -p
   ```

8. Install the MySQL connector JAR file:

   ```
   yum install mysql-connector-java-5.0.8-1
   ```

1.8. Download Companion Files

We have provided a set of companion files, including script files (`scripts.zip`) and configuration files (`configuration_files.zip`), that you should download and use throughout this process.

Download and extract the files:

```
wget http://dev.hortonworks.com.s3.amazonaws.com/HDP-2.0.0.2/tools/
hdp_manual_install_rpm_helper_files-2.0.0.22.tar.gz
```

1.9. Create System Users and Groups

In general Hadoop services should be owned by specific users and not by root or application users.The table below shows the typical users for Hadoop services. Identify the users that you want for your Hadoop services and the common Hadoop group and create these accounts on your system.

Table 1.1. Typical System Users and Groups

Hadoop Service	User	Group
HDFS	hdfs	hadoop
YARN	yarn	hadoop
MapReduce	mapred	hadoop
Hive	hive	hadoop
Pig	pig	hadoop
HCatalog/WebHCatalog	hcat	hadoop
HBase	hbase	hadoop
ZooKeeper	zookeeper	hadoop

1.10. Define Environment Parameters

You need to set up the following specific users and directories for your HDP installation:

1. Define users and groups
2. Define directories

1.10.1. Define Users and Groups

The following table describes system user account and groups. Use this table to define what you are going to use in setting up your environment. These users and groups should reflect the accounts you created in Create System Users and Groups.

Note

The `scripts` directory you downloaded in Download Companion Files includes a script, `usersAndGroups.sh`, for setting user and group environment parameters.

We strongly suggest you edit and source (alternatively, you can also copy the contents to your ~/.bash_profile) to set up these environment variables in your environment.

Table 1.2. Define Users and Groups for Systems

Parameter	Definition
HDFS_USER	User owning the HDFS services. For example, hdfs.
YARN_USER	User owning the YARN services. For example, yarn.
ZOOKEEPER_USER	User owning the ZooKeeper services. For example, zookeeper.
HIVE_USER	User owning the Hive services. For example, hive.
WEBHCAT_USER	User owning the WebHCat services. For example, hcat.
HBASE_USER	User owning the HBase services. For example, hbase.
PIG_USER	User owning the Pig services. For example, pig.
HADOOP_GROUP	A common group shared by services. For example, hadoop.

1.10.2. Define Directories

The following table describes the directories for install, configuration, data, process IDs and logs based on the Hadoop Services you plan to install. Use this table to define what you are going to use in setting up your environment.

Note

The scripts directory you downloaded in Download Companion Files includes a script, directories.sh, for setting directory environment parameters.

We strongly suggest you edit and source (alternatively, you can also copy the contents to your ~/.bash_profile) to set up these environment variables in your environment.

Table 1.3. Define Directories for Core Hadoop

Hadoop Service	Parameter	Definition
HDFS	DFS_NAME_DIR	Space separated list of directories where NameNode should store the file system image. For example, /grid/hadoop/hdfs/nn /grid1/hadoop/hdfs/nn
HDFS	DFS_DATA_DIR	Space separated list of directories where DataNodes should store the blocks. For example, /grid/hadoop/hdfs/dn /grid1/hadoop/hdfs/dn

Hadoop Service	Parameter	Definition
		`/grid2/hadoop/hdfs/dn`

HDFS	`FS_CHECKPOINT_DIR`	Space separated list of directories where SecondaryNameNode should store the checkpoint image.
		For example,
		`/grid/hadoop/hdfs/snn`
		`/grid1/hadoop/hdfs/snn`
		`/grid2/hadoop/hdfs/snn`

HDFS	`HDFS_LOG_DIR`	Directory for storing the HDFS logs. This directory name is a combination of a directory and the *$HDFS_USER*. For example, `/var/log/hadoop/hdfs` where `hdfs` is the *$HDFS_USER*.
HDFS	`HDFS_PID_DIR`	Directory for storing the HDFS process ID. This directory name is a combination of a directory and the *$HDFS_USER*. For example, `/var/run/hadoop/hdfs` where `hdfs` is the *$HDFS_USER*
HDFS	`HADOOP_CONF_DIR`	Directory for storing the Hadoop configuration files. For example, `/etc/hadoop/conf`

YARN	`YARN_LOCAL_DIR`	Space separated list of directories where YARN should store temporary data.
		For example,
		`/grid/hadoop/yarn`
		`/grid1/hadoop/yarn`
		`/grid2/hadoop/yarn`.

YARN	`YARN_LOG_DIR`	Directory for storing the YARN logs. For example, `/var/log/hadoop/yarn`. This directory name is a combination of a directory and the *$YARN_USER*. In the example `yarn` is the *$YARN_USER*.
YARN	`YARN_PID_DIR`	Directory for storing the YARN process ID. For example, `/var/run/hadoop/yarn`. This directory name is a combination of a directory and the *$YARN_USER*.

Hadoop Service	Parameter	Definition
		In the example, `yarn` is the *$YARN_USER*.
MapReduce	`MAPRED_LOG_DIR`	Directory for storing the JobHistory Server logs. For example, `/var/log/hadoop/mapred`. This directory name is a combination of a directory and the *$MAPRED_USER*. In the example `mapred` is the *$MAPRED_USER*

Table 1.4. Define Directories for Ecosystem Components

Hadoop Service	Parameter	Definition
Pig	`PIG_CONF_DIR`	Directory to store the Pig configuration files. For example, `/etc/pig/conf`.
Pig	`PIG_LOG_DIR`	Directory to store the Pig logs. For example, `/var/log/pig`.
Pig	`PIG_PID_DIR`	Directory to store the Pig process ID. For example, `/var/run/pig`.
Hive	`HIVE_CONF_DIR`	Directory to store the Hive configuration files. For example, `/etc/hive/conf`.
Hive	`HIVE_LOG_DIR`	Directory to store the Hive logs. For example, `/var/log/hive`.
Hive	`HIVE_PID_DIR`	Directory to store the Hive process ID. For example, `/var/run/hive`.
WebHCat	`WEBHCAT_CONF_DIR`	Directory to store the WebHCat configuration files. For example, `/etc/hcatalog/conf/webhcat`.
WebHCat	`WEBHCAT_LOG_DIR`	Directory to store the WebHCat logs. For example, `var/log/webhcat`.
WebHCat	`WEBHCAT_PID_DIR`	Directory to store the WebHCat process ID. For example, `/var/run/webhcat`.
HBase	`HBASE_CONF_DIR`	Directory to store the HBase configuration files. For example, `/etc/hbase/conf`.
HBase	`HBASE_LOG_DIR`	Directory to store the HBase logs. For example, `/var/log/hbase`.
HBase	`HBASE_PID_DIR`	Directory to store the HBase process ID. For example, `/var/run/hbase`.
ZooKeeper	`ZOOKEEPER_DATA_DIR`	Directory where ZooKeeper will store data. For example, `/grid/hadoop/zookeeper/data`
ZooKeeper	`ZOOKEEPER_CONF_DIR`	Directory to store the ZooKeeper configuration files. For example, `/etc/zookeeper/conf`.
ZooKeeper	`ZOOKEEPER_LOG_DIR`	Directory to store the ZooKeeper logs. For example, `/var/log/zookeeper`.
ZooKeeper	`ZOOKEEPER_PID_DIR`	Directory to store the ZooKeeper process ID. For example, `/var/run/zookeeper`.

Hadoop Service	Parameter	Definition
ZooKeeper	`myid`	Every machine that is part of the ZooKeeper ensemble should know about every other machine in the ensemble. Create a file named `myid` (one for each server) which resides in that server's data directory *`$ZOOKEEPER_DATA_DIR`*. The myid file consists of a single line containing only the text of that machine's id. So myid of server 1 would contain the string `"1"` and nothing else. The id must be unique within the ensemble and should have a value between 1 and 255.

2. Installing HDFS and YARN

This section describes how to install the Hadoop Core components, HDFS, YARN, and MapReduce.

Complete the following instructions to install Hadoop Core components:

1. Set Default File and Directory Permissions
2. Install the Hadoop RPMs
3. Install Compression Libraries
4. Create Directories

2.1. Set Default File and Directory Permissions

Set the default file and directory permissions to 0022 (022). This is typically the default for most Linux distributions.

Use the umask command to confirm and set as necessary.

Ensure that the umask is set for all terminal sessions that you use during installation.

2.2. Install the Hadoop RPMs

Execute the following command on all cluster nodes.

From a terminal window, type:

```
yum install hadoop hadoop-hdfs hadoop-libhdfs hadoop-yarn hadoop-mapreduce
 hadoop-client openssl
```

2.3. Install Compression Libraries

Make the following compression libraries available on all the cluster nodes.

2.3.1. Install Snappy

Complete the following instructions on all the nodes in your cluster:

1. Install Snappy.

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

2. Make the Snappy libraries available to Hadoop:

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

2.3.2. Install LZO

Execute the following command on all the nodes in your cluster. From a terminal window, type:

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

2.4. Create Directories

Create directories and configure ownership + permissions on the appropriate hosts as described below.

If any of these directories already exist, we recommend deleting and recreating them.

Use the following instructions to create appropriate directories:

1. We strongly suggest that you edit and source the files included in `scripts.zip` file (downloaded in Download Companion Files).

   Alternatively, you can also copy the contents to your `~/.bash_profile`) to set up these environment variables in your environment.

2. Create the NameNode directories

3. Create the Secondary NameNode directories

4. Create the DataNode and YARN NodeManager local directories

5. Create the log and PID directories

2.4.1. Create the NameNode Directories

On the node that hosts the NameNode service, execute the following commands:

```
mkdir -p $DFS_NAME_DIR;
chown -R $HDFS_USER:$HADOOP_GROUP $DFS_NAME_DIR;
chmod -R 755 $DFS_NAME_DIR;
```

where:

- *$DFS_NAME_DIR* is the space separated list of directories where NameNode stores the file system image. For example, `/grid/hadoop/hdfs/nn /grid1/hadoop/hdfs/nn`.

- *$HDFS_USER* is the user owning the HDFS services. For example, `hdfs`.

- *$HADOOP_GROUP* is a common group shared by services. For example, `hadoop`.

2.4.2. Create the SecondaryNameNode Directories

On all the nodes that can potentially run the SecondaryNameNode service, execute the following commands:

```
mkdir -p $FS_CHECKPOINT_DIR;
```

```
chown -R $HDFS_USER:$HADOOP_GROUP $FS_CHECKPOINT_DIR;
chmod -R 755 $FS_CHECKPOINT_DIR;
```

where:

- *$FS_CHECKPOINT_DIR* is the space separated list of directories where SecondaryNameNode should store the checkpoint image. For example, `/grid/hadoop/hdfs/snn /grid1/hadoop/hdfs/snn /grid2/hadoop/hdfs/snn`.
- *$HDFS_USER* is the user owning the HDFS services. For example, `hdfs`.
- *$HADOOP_GROUP* is a common group shared by services. For example, `hadoop`.

2.4.3. Create DataNode and YARN NodeManager Local Directories

On all DataNodes, execute the following commands:

```
mkdir -p $DFS_DATA_DIR;
chown -R $HDFS_USER:$HADOOP_GROUP $DFS_DATA_DIR;
chmod -R 750 $DFS_DATA_DIR;
```

where:

- *$DFS_DATA_DIR* is the space separated list of directories where DataNodes should store the blocks. For example, `/grid/hadoop/hdfs/dn /grid1/hadoop/hdfs/dn /grid2/hadoop/hdfs/dn`.
- *$HDFS_USER* is the user owning the HDFS services. For example, `hdfs`.
- *$HADOOP_GROUP* is a common group shared by services. For example, `hadoop`.

On the ResourceManager and all Datanodes, execute the following commands:

```
mkdir -p $YARN_LOCAL_DIR;
chown -R $YARN_USER:$HADOOP_GROUP $YARN_LOCAL_DIR;
chmod -R 755 $YARN_LOCAL_DIR;
```

where:

- *$YARN_LOCAL_DIR* is the space separated list of directories where YARN should store temporary data. For example, `/grid/hadoop/yarn /grid1/hadoop/yarn /grid2/hadoop/yarn`.
- *$YARN_USER* is the user owning the YARN services. For example, `yarn`.
- *$HADOOP_GROUP* is a common group shared by services. For example, `hadoop`.

2.4.4. Create the Log and PID Directories

On all nodes, execute the following commands:

1. Create *$HDFS_LOG_DIR* and *$HDFS_PID_DIR* directories. Set appropriate permissions

   Execute the following commands on all the nodes in your cluster:

```
mkdir -p $HDFS_LOG_DIR;
chown -R $HDFS_USER:$HADOOP_GROUP $HDFS_LOG_DIR;
chmod -R 755 $HDFS_LOG_DIR;
```

```
mkdir -p $HDFS_PID_DIR;
chown -R $HDFS_USER:$HADOOP_GROUP $HDFS_PID_DIR;
chmod -R 755 $HDFS_PID_DIR
```

where:

- *$HDFS_LOG_DIR* is the directory for storing the HDFS logs.

  This directory name is a combination of a directory and the *$HDFS_USER*.

  For example, `/var/log/hadoop/hdfs` where `hdfs` is the *$HDFS_USER*.

- *$HDFS_PID_DIR* is the directory for storing the HDFS process ID.

  This directory name is a combination of a directory and the *$HDFS_USER*.

  For example, `/var/run/hadoop/hdfs` where `hdfs` is the *$HDFS_USER*.

- *$HDFS_USER* is the user owning the HDFS services. For example, `hdfs`.

- *$HADOOP_GROUP* is a common group shared by services. For example, `hadoop`.

2. Create *$YARN_LOG_DIR* and *$YARN_PID_DIR* directories. Set appropriate permissions.

   Execute the following commands on all the nodes in your cluster:

```
mkdir -p $YARN_LOG_DIR;
chown -R $YARN_USER:$HADOOP_GROUP $YARN_LOG_DIR;
chmod -R 755 $YARN_LOG_DIR;
```

```
mkdir -p $YARN_PID_DIR;
chown -R $YARN_USER:$HADOOP_GROUP $YARN_PID_DIR;
chmod -R 755 $YARN_PID_DIR;
```

where:

- *$YARN_LOG_DIR* is the directory for storing the YARN logs.

  This directory name is a combination of a directory and the *$YARN_USER*.

  For example, `/var/log/hadoop/yarn` where `yarn` is the *$YARN_USER*.

- *$YARN_PID_DIR* is the directory for storing the YARN process ID.

  This directory name is a combination of a directory and the *$YARN_USER*.

  For example, `/var/run/hadoop/yarn` where `yarn` is the *$YARN_USER*.

- *$YARN_USER* is the user owning the YARN services. For example, `yarn`.

- *$HADOOP_GROUP* is a common group shared by services. For example, `hadoop`.

3. Create *$MAPRED_LOG_DIR* for JobHistory Server and set appropriate permissions:

```
mkdir -p $MAPRED_LOG_DIR;
chown -R $MAPRED_USER:$HADOOP_GROUP $MAPRED_LOG_DIR;
chmod -R 755 $MAPRED_LOG_DIR;
```

where:

- *$MAPRED_PID_DIR* is the directory for storing the JobHistory Server logs.
- This directory name is a combination of a directory and the *$MAPRED_USER*.

  For example, `/var/run/hadoop/mapred` where `mapred` is the *$MAPRED_USER*.
- *$MAPRED_USER* is the user owning the MAPRED services. For example, `mapred`.
- *$HADOOP_GROUP* is a common group shared by services. For example, `hadoop`.

3. Setting Up the Hadoop Configuration

This section describes how to set up and edit the deployment configuration files for HDFS and MapReduce.

Use the following instructions to set up Hadoop configuration files:

1. We strongly suggest that you edit and source the files included in `scripts.zip` file (downloaded in Download Companion Files).

   Alternatively, you can also copy the contents to your `~/.bash_profile`) to set up these environment variables in your environment.

2. From the downloaded `scripts.zip` file, extract the files from the `configuration_files/core_hadoop` directory to a temporary directory.

3. Modify the configuration files.

   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. See Define Environment Parameters for more information.

   a. Edit the `core-site.xml` and modify the following properties:

   ```
   <property>
    <name>fs.default.name</name>
    <value>hdfs://$namenode.full.hostname:8020</value>
    <description>Enter your NameNode hostname</description>
   </property>
   ```

   ```
   <property>
    <name>fs.checkpoint.dir</name>
    <value>/grid/hadoop/hdfs/snn,/grid1/hadoop/hdfs/snn,/grid2/hadoop/hdfs/
   snn</value>
    <description>A comma separated list of paths. Use the list of
    directories from $FS_CHECKPOINT_DIR.
                    For example, /grid/hadoop/hdfs/snn,sbr/grid1/hadoop/hdfs/
   snn,sbr/grid2/hadoop/hdfs/snn </description>
   </property>
   ```

   b. Edit the `hdfs-site.xml` and modify the following properties:

   ```
   <property>
    <name>dfs.name.dir</name>
    <value>/grid/hadoop/hdfs/nn,/grid1/hadoop/hdfs/nn</value>
    <description>Comma separated list of paths. Use the list of directories
    from $DFS_NAME_DIR.
                    For example, /grid/hadoop/hdfs/nn,/grid1/hadoop/hdfs/nn.
   </description>
   </property>
   ```

```
<property>
 <name>dfs.data.dir</name>
 <value>/grid/hadoop/hdfs/dn,/grid1/hadoop/hdfs/dn</value>
 <description>Comma separated list of paths. Use the list of directories
 from $DFS_DATA_DIR.
                For example, /grid/hadoop/hdfs/dn,/grid1/hadoop/hdfs/dn.
</description>
</property>
```

```
<property>
 <name>dfs.http.address</name>
 <value>$namenode.full.hostname:50070</value>
 <description>Enter your NameNode hostname for http access.</description>
</property>
```

```
<property>
 <name>dfs.secondary.http.address</name>
 <value>$secondary.namenode.full.hostname:50090</value>
 <description>Enter your Secondary NameNode hostname.</description>
</property>
```

```
<property>
 <name>dfs.https.address</name>
 <value>$namenode.full.hostname:50470</value>
 <description>Enter your NameNode hostname for https access.</
description>
</property>
```

Note

The value of NameNode new generation size should be 1/8 of maximum heap size (`-Xmx`). Please check, as the default setting may not be accurate.

To change the default value, edit the `/etc/hadoop/conf/hadoop-env.sh` file and change the value of the `-XX:MaxnewSize` parameter to 1/8th the value of the maximum heap size (`-Xmx`) parameter.

c. Edit the `yarn-site.xml` and modify the following properties:

```
<property>
 <name>yarn.resourcemanager.resourcetracker.address</name>
 <value>$resourcemanager.full.hostname:8025</value>
 <description>Enter your ResourceManager hostname.</description>
</property>
```

```
<property>
 <name>yarn.resourcemanager.scheduler.address</name>
 <value>$resourcemanager.full.hostname:8030</value>
 <description>Enter your ResourceManager hostname.</description>
</property>
```

```
<property>
 <name>yarn.resourcemanager.address</name>
 <value>$resourcemanager.full.hostname:8050</value>
 <description>Enter your ResourceManager hostname.</description>
</property>
```

```
<property>
 <name>yarn.resourcemanager.admin.address</name>
 <value>$resourcemanager.full.hostname:8041</value>
 <description>Enter your ResourceManager hostname.</description>
</property>
```

```
<property>
 <name>yarn.nodemanager.local-dirs</name>
 <value>/grid/hadoop/hdfs/yarn,/grid1/hadoop/hdfs/yarn</value>
 <description>Comma separated list of paths. Use the list of directories
 from $YARN_LOCAL_DIR.
                For example, /grid/hadoop/hdfs/yarn,/grid1/hadoop/hdfs/
yarn.</description>
</property>
```

```
<property>
 <name>yarn.nodemanager.log-dirs</name>
 <value>/var/log/hadoop/yarn</value>
 <description>Use the list of directories from $YARN_LOG_DIR.
                For example, /var/log/hadoop/yarn.</description>
</property>
```

d. Edit the `mapred-site.xml` and modify the following properties:

```
<property>
 <name>mapreduce.jobhistory.address</name>
 <value>$jobhistoryserver.full.hostname:10020</value>
 <description>Enter your JobHistoryServer hostname.</description>
</property>
```

```
<property>
 <name>mapreduce.jobhistory.webapp.address</name>
 <value>$jobhistoryserver.full.hostname:19888</value>
 <description>Enter your JobHistoryServer hostname.</description>
</property>
```

4. Copy the configuration files.

a. On all hosts in your cluster, create the Hadoop configuration directory:

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

where *$HADOOP_CONF_DIR* is the directory for storing the Hadoop configuration files.

For example, `/etc/hadoop/conf`.

b. Copy all the configuration files to *$HADOOP_CONF_DIR*.

c. Set appropriate permissions:

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

where:

- *$HDFS_USER* is the user owning the HDFS services. For example, `hdfs`.

- *$HADOOP_GROUP* is a common group shared by services. For example, `hadoop`.

4. Validating the Core Hadoop Installation

This section describes starting Core Hadoop and doing simple smoke tests.

4.1. Format and Start HDFS

1. Execute these commands on the NameNode host machine:

```
su $HDFS_USER
/usr/lib/hadoop/bin/hadoop namenode -format
/usr/lib/hadoop/sbin/hadoop-daemon.sh --config $HADOOP_CONF_DIR start
 namenode
```

2. Execute these commands on the SecondaryNameNode:

```
su $HDFS_USER
/usr/lib/hadoop/sbin/hadoop-daemon.sh --config $HADOOP_CONF_DIR start
 secondarynamenode
```

3. Execute these commands on all DataNodes:

```
su $HDFS_USER
/usr/lib/hadoop/sbin/hadoop-daemon.sh --config $HADOOP_CONF_DIR start
 datanode
```

where:

- *$HDFS_USER* is the user owning the HDFS services. For example, `hdfs`.
- *$HADOOP_CONF_DIR* is the directory for storing the Hadoop configuration files. For example, `/etc/hadoop/conf`.

4.2. Smoke Test HDFS

1. See if you can reach the NameNode server with your browser:

```
http://$namenode.full.hostname:50070
```

2. Try copying a file into HDFS and listing that file:

```
su $HDFS_USER
/usr/lib/hadoop/sbin/hadoop dfs -copyFromLocal /etc/passwd passwd-test
/usr/lib/hadoop/sbin/hadoop dfs -ls
```

3. Test browsing HDFS:

```
http://$datanode.full.hostname:50075/browseDirectory.jsp?namenodeInfoPort=
50070&dir=/&nnaddr=$datanode.full.hostname:8020
```

4.3. Start YARN

1. Execute these commands from the ResourceManager server:

```
<login as $YARN_USER and source the directories.sh companion script>
/usr/lib/hadoop-yarn/sbin/yarn-daemon.sh --config $HADOOP_CONF_DIR start
 resourcemanager
```

2. Execute these commands from all NodeManager nodes:

```
<login as $YARN_USER and source the directories.sh companion script>
/usr/lib/hadoop-yarn/sbin/yarn-daemon.sh --config $HADOOP_CONF_DIR start
 nodemanager
```

```

hadoop fs -mkdir /app-logs
hadoop fs -chown $YARN_USER /app-logs
hadoop fs -chmod 1777 /app-logs
```

where:

- *$YARN_USER* is the user owning the YARN services. For example, `yarn`.
- *$HADOOP_CONF_DIR* is the directory for storing the Hadoop configuration files. For example, `/etc/hadoop/conf`.

4.4. Start MapReduce JobHistory Server

1. Execute these commands from the JobHistory server to set up directories on HDFS :

```
su $HDFS_USER
/usr/lib/hadoop/sbin/hadoop fs -mkdir -p /mapred/history/done_intermediate
/usr/lib/hadoop/sbin/hadoop fs -chmod -R 1777 /mapred/history/
done_intermediate
/usr/lib/hadoop/sbin/hadoop fs -mkdir -p /mapred/history/done
/usr/lib/hadoop/sbin/hadoop fs -chmod -R 1777 /mapred/history/done
/usr/lib/hadoop/sbin/hadoop fs -chown -R mapred /mapred
```

2. Execute these commands from the JobHistory server:

```
export HADOOP_LIBEXEC_DIR=/usr/lib/hadoop/libexec/
export HADOOP_MAPRED_HOME=/usr/lib/hadoop-mapreduce
export HADOOP_MAPRED_LOG_DIR=/var/log/hadoop/mapred
```

```

<login as $MAPRED_USER and source the directories.sh companion script>
/usr/lib/hadoop-mapreduce/sbin/mr-jobhistory-daemon.sh start historyserver
 --config $HADOOP_CONF_DIR
```

where:

- *$HDFS_USER* is the user owning the HDFS services. For example, `hdfs`.
- *$MAPRED_USER* is the user owning the MapRed services. For example, `mapred`.
- *$HADOOP_CONF_DIR* is the directory for storing the Hadoop configuration files. For example, `/etc/hadoop/conf`.

4.5. Smoke Test MapReduce

1. Try browsing to the ResourceManager:

```
http://$resourcemanager.full.hostname:8088/
```

2. Smoke test using Terasort and sort 10GB of data.

```
su $HDFS_USER
/usr/lib/hadoop/bin/hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-
examples-2.0.2.1-alpha.jar teragen 100 /test/10gsort/input
/usr/lib/hadoop/bin/hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-
examples-2.0.2.1-alpha.jar terasort /test/10gsort/input /test/10gsort/output
```

5. Installing Apache Pig

This section describes installing and testing Apache Pig, a platform for creating higher level data flow programs that can be compiled into sequences of MapReduce programs, using Pig Latin, the platform's native language.

Complete the following instructions to install Pig:

1. Install the Pig RPMs
2. Set Directories and Permissions
3. Set Up Configuration Files

5.1. Install the Pig RPMs

On all the hosts where Pig programs will be executed, install the RPMs. From a terminal window, type:

```
yum install pig
```

5.2. Set Directories and Permissions

Create directories and configure ownership + permissions on the appropriate hosts as described below.

If any of these directories already exist, we recommend deleting and recreating them. Use the following instructions to set up Pig configuration files :

1. We strongly suggest that you edit and source the files included in `scripts.zip` file (downloaded in Download Companion Files).

   Alternatively, you can also copy the contents to your `~/.bash_profile`) to set up these environment variables in your environment.

2. Execute the following commands on all the hosts where Pig programs will be executed:

   ```
   mkdir -p $PIG_LOG_DIR
   chown -R $PIG_USER:$HADOOP_GROUP $PIG_LOG_DIR
   chmod 755 -R $PIG_LOG_DIR
   ```

   where:

   - *`$PIG_LOG_DIR`* is the directory to store the Pig logs. For example, `/var/log/pig`.
   - *`$PIG_USER`* is the user owning the Pig services. For example, `pig`.
   - *`$HADOOP_GROUP`* is a common group shared by services. For example, `hadoop`.

5.3. Set Up Configuration Files

Use the following instructions to set up configuration files for Pig:

1. Extract the Pig configuration files.

   From the downloaded `scripts.zip` file, extract the files from the `configuration_files/pig` directory to a temporary directory.

2. Copy the configuration files.

   a. On all hosts where Pig will be executed, create the Pig configuration directory:

   ```
   rm -r $PIG_CONF_DIR
   mkdir -p $PIG_CONF_DIR
   ```

   b. Copy all the configuration files to *`$PIG_CONF_DIR`*.

   c. Set appropriate permissions:

   ```
   chown -R $PIG_USER:$HADOOP_GROUP $PIG_CONF_DIR/../
   chmod -R 755 $PIG_CONF_DIR/../
   ```

   where:

   - *`$PIG_CONF_DIR`* is the directory to store Pig configuration files. For example, `/etc/pig/conf`.

   - *`$PIG_USER`* is the user owning the Pig services. For example, `pig`.

   - *`$HADOOP_GROUP`* is a common group shared by services. For example, `hadoop`.

5.4. Validate the Installation

Use the following steps to validate your installation:

1. On the host machine where Pig is installed execute the following commands:

   ```
   login as $HDFS_USER
   /usr/lib/hadoop/bin/hadoop dfs -copyFromLocal /etc/passwd passwd
   ```

2. Create the pig script file `/tmp/id.pig` with the following contents:

   ```
   echo "A = load 'passwd' using PigStorage(':'); " > /tmp/id.pig
   echo "B = foreach A generate \$0 as id; store B into '/tmp/id.out'; " >> /
   tmp/id.pig
   ```

3. Execute the Pig script:

   ```
   pig -l /tmp/pig.log /tmp/id.pig
   ```

6. Installing Apache Hive and Apache HCatalog

This section describes installing and testing Apache Hive, a tool for creating higher level SQL-like queries using HiveQL, the tool's native language that can then be compiled into sequences of MapReduce programs.

It also describes installing and testing Apache HCatalog, a metadata abstraction layer that insulates users and scripts from how and where data is physically stored.

Complete the following instructions to install Hive and HCatalog:

1. Install the Hive and HCatalog RPMs
2. Set Directories and Permissions
3. Set Up the Hive/HCatalog Configuration Files
4. Create Directories on HDFS
5. Optional - Download the Database Connector
6. Validate the Installation

6.1. Install the Hive and HCatalog RPMs

On all client/gateway nodes (on which Hive programs will be executed), Hive Metastore Server, and HiveServer2 machine, install the Hive RPMs. TOdo - hcat??

```
yum install hive hcatalog
```

6.2. Set Directories and Permissions

Create directories and configure ownership + permissions on the appropriate hosts as described below.

If any of these directories already exist, we recommend deleting and recreating them. Use the following instructions to set up Pig configuration files :

1. We strongly suggest that you edit and source the files included in `scripts.zip` file (downloaded in Download Companion Files).

   Alternatively, you can also copy the contents to your `~/.bash_profile`) to set up these environment variables in your environment.

2. Execute these commands on the Hive server machine:

   ```
   mkdir -p $HIVE_LOG_DIR;
   ```

   ```
   chown -R $HIVE_USER:$HADOOP_GROUP $HIVE_LOG_DIR;
   chmod -R 755 $HIVE_LOG_DIR;
   ```

where:

- *$HIVE_LOG_DIR* is the directory for storing theHive Server logs.

  This directory name is a combination of a directory and the *$HIVE_USER*.

- *$HIVE_USER* is the user owning the Hive services. For example, hive.

- *$HADOOP_GROUP* is a common group shared by services. For example, hadoop.

6.3. Set Up the Hive/HCatalog Configuration Files

Use the following instructions to set up the Hive/HCatalog configuration files:

1. Extract the Hive/HCatalog configuration files.

   From the downloaded scripts.zip file, extract the files in configuration_files/hive directory to a temporary directory.

2. Modify the configuration files.

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

   a. Edit hive-site.xml and modify the following properties:

```
<property>
 <name>javax.jdo.option.ConnectionURL</name>
 <value>jdbc:mysql://$mysql.full.hostname:3306/$database.name?
createDatabaseIfNotExist=true</value>
 <description>Enter your JDBC connection string. </description>
</property>
```

```
<property>
 <name>javax.jdo.option.ConnectionUserName</name>
 <value>$dbusername</value>
 <description>Enter your MySQL credentials. </description>
</property>
```

```
<property>
 <name>javax.jdo.option.ConnectionPassword</name>
 <value>$dbuserpassword</value>
 <description>Enter your MySQL credentials. </description>
</property>
```

   Enter your MySQL credentials from Install MySQL (Optional).

```
<property>
 <name>hive.metastore.uris</name>
 <value>thrift://$metastore.server.full.hostname:9083</value>
 <description>URI for client to contact metastore server. To enable
 HiveServer2, leave the property value empty. </description>
</property>
```

3. Copy the configuration files.

a. On all Hive hosts create the Hive configuration directory.

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

b. Copy all the configuration files to `$HIVE_CONF_DIR` directory.

c. Set appropriate permissions:

```
chown -R $HIVE_USER:$HADOOP_GROUP $HIVE_CONF_DIR/../ ;
chmod -R 755 $HIVE_CONF_DIR/../ ;
```

where:

- `$HIVE_CONF_DIR` is the directory to store the Hive configuration files. For example, `/etc/hive/conf`.
- `$HIVE_USER` is the user owning the Hive services. For example, `hive`.
- `$HADOOP_GROUP` is a common group shared by services. For example, `hadoop`.

6.4. Create Directories on HDFS

1. Create Hive user home on HDFS.

```
Login as $HDFS_USER
hadoop fs -mkdir /user/$HIVE_USER
hadoop fs -chown $HIVE_USER:$HIVE_USER/user/$HIVE_USER
```

2. Create warehouse directory on HDFS.

```
hadoop fs -mkdir /apps/hive/warehouse
 hadoop fs -chown -R $HIVE_USER:users /apps/hive/warehouse
 hadoop fs -chmod -R 775 /apps/hive/warehouse
```

where:

- `$HDFS_USER` is the user owning the HDFS services. For example, `hdfs`.
- `$HIVE_USER` is the user owning the Hive services. For example, `hive`.

6.5. [Optional] - Download the Database Connector

By default, Hive uses embedded Derby database for its metastore. However, you can choose to enable remote database (MySQL) for Hive metastore.

1. Ensure that you complete the instructions provided here.
2. Unzip and copy the downloaded JAR file the `/usr/lib/hive/lib/` directory on your Hive host machine.
3. Ensure that the JAR file has appropriate permissions.

6.6. Validate the Installation

Use the following steps to validate your installation:

1. Start Hive Metastore service.

   ```
    Login as $HIVE_USER
   nohup hive --service metastore>$HIVE_LOG_DIR/hive.out 2>$HIVE_LOG_DIR/hive.
   log &
   ```

2. Smoke Test Hive.

   a. Open Hive command line shell.

      ```
      hive
      ```

   b. Run sample commands.

      ```
      show databases;
      create table test(col1 int, col2 string);
      show tables;
      ```

3. Start HiveServer2.

   ```
    /usr/lib/hive/bin/hiveserver2 -hiveconf hive.metastore.uris=" " >
   $HIVE_LOG_DIR/hiveserver2.out 2> $HIVE_LOG_DIR/hiveserver2.log &
   ```

4. Smoke Test HiveServer2.

   a. Open Beeline command line shell to interact with HiveServer2.

      ```
      /usr/lib/hive/bin/beeline
      ```

   b. Establish connection to server.

      ```
      !connect jdbc:hive2://$hive.server.full.hostname:10000 $HIVE_USER
       password org.apache.hive.jdbc.HiveDriver
      ```

   c. Run sample commands.

      ```
      show databases;
      create table test2(a int, b string);
      show tables;
      ```

where:

- *$HDFS_USER* is the user owning the HDFS services. For example, `hdfs`.
- *$HIVE_LOG_DIR* is the directory for storing theHive Server logs. This directory name is a combination of a directory and the *$HIVE_USER*.

7. Installing and Configuring Apache Tez

Tez is the next generation Hadoop Query Processing framework written on top of YARN.

Warning

These instructions are now obsolete and no longer provide a Supported version of Tez. Use the information found in HDP 2.1 regarding Installing and Configuring Tez.

Tez AM is a new and improved implementation of the MapReduce application that supports container reuse. This allows jobs to run faster on clusters that have limited resources per job. On smaller clusters, it reduces the time for a job to finish by efficiently using a container to run more than one task.

The **Tez AMPoolService** or **Tez Service** is a service that launches and makes available a pool of pre-launched MapReduce AMs (Tez AMs). These AMs in the pool can, in turn, be configured to pre-allocate a number of containers to allow jobs to be launched and completed faster. To use the Tez Service, the clients must submit the jobs to this service instead of the ResourceManager.

Use the following instructions to install and configure Tez:

1. Install the Tez RPMs
2. Enable Tez AM
3. Enable and Use Tez Service
4. Optional: Disable Tez
5. Troubleshooting

7.1. Install the Tez RPMs

Warning

These instructions are now obsolete and no longer provide a Supported version of Tez. Use the information found in HDP 2.1 regarding Installing and Configuring Tez.

Execute the following command on all cluster nodes. From a terminal window, type:

```
yum install tez
```

This will install the Tez RPM and create the following directories on all the cluster nodes:

Table 7.1. Default Directories for Tez

Hadoop Service	Parameter	Description	Default
Tez	TEZ_HOME	Directory that contains all the Tez JAR files.	/usr/lib/tez

Hadoop Service	Parameter	Description	Default
Tez	TEZ_CONF_DIR	Directory that contains all the Tez configuration files.	/etc/tez/conf
Tez	TEZ_LOG_DIR	Directory to store the Tez logs.	*$TEZ_HOME*/logs
Tez	TEZ_PID_DIR	Directory to store the Tez process ID.	/tmp

7.1.1. Optional: Modify Default Directories

Warning

These instructions are now obsolete and no longer provide a Supported version of Tez. Use the information found in HDP 2.1 regarding Installing and Configuring Tez.

By default, the Tez RPM creates the following directories:

Table 7.2. Default Directories for Tez

Hadoop Service	Parameter	Description	Default
Tez	TEZ_HOME	Directory that contains all the Tez JAR files.	/usr/lib/tez
Tez	TEZ_CONF_DIR	Directory that contains all the Tez configuration files.	/etc/tez/conf
Tez	TEZ_LOG_DIR	Directory to store the Tez logs.	*$TEZ_HOME*/logs
Tez	TEZ_PID_DIR	Directory to store the Tez process ID.	/tmp

To change these default locations, execute the following instructions:

- Create new directories for those parameters that you want to override.
- Ensure that Tez Service user has appropriate permissions to these directories.

  For example, if Hive user is responsible for submitting queries to the Tez Service, this user should have appropriate permissions to the newly created directories.
- On all the client nodes and Tez Service host machine, edit /etc/tez/conf/tez-env.sh file and modify those environment variables that you want to override.

  For example:

  ```
  export TEZ_LOG_DIR="$TEZ_LOG_DIR"
  export TEZ_PID_DIR="$TEZ_PID_DIR"
  export TEZ_HOME="$TEZ_HOME"
  export TEZ_CONF_DIR="$TEZ_CONF_DIR"
  ```

7.2. Enable Tez AM

Warning

These instructions are now obsolete and no longer provide a Supported version of Tez. Use the information found in HDP 2.1 regarding Installing and Configuring Tez.

Use the following instructions to enable Tez AM:

1. On all the client nodes and Tez Service host machine, edit `/etc/tez/conf/tez-env.sh` file and modify the following environment variables:

   ```
   export HADOOP_HOME="$HADOOP_HOME"
   export JAVA_HOME="$JAVA_HOME"
   ```

   where

   - *$HADOOP_HOME* is the location of the directory that contains all core Hadoop JAR files. For example, `/usr/lib/hadoop`.
   - *$JAVA_HOME* is the location of the directory that contains JDK.

2. Ensure that the `/`*$HADOOP_HOME*`/bin/hadoop` file exists on the Tez Service host machine.

3. On all the client nodes and Tez host machine, edit `mapred-site.xml` and modify the following properties:

   a. Enable Tez AM:

   ```
   <property>
       <name>mapreduce.framework.name</name>
       <value>yarn-tez</value>
       <description>Name of the MapReduce framework. Default value is yarn.
   </description>
   </property>
   ```

   b. Set MapReduce CLASSPATH to a CLASSPATH that contains all the Tez JAR files:

   ```
   <property>
       <name>mapreduce.application.classpath</name>
       <value>$TEZ_HOME/*,$TEZ_HOME/lib/*</value>
       <description>Classpath for MapReduce applications.</description>
   </property>
   ```

   where *$TEZ_HOME* is the location of the directory that contains all the Tez JAR files. By default, *$TEZ_HOME* is set to **`/usr/lib/tez`**.

   c. Enable container reuse across task attempts:

   ```
   <property>
       <name>yarn.app.mapreduce.am.scheduler.reuse.enable</name>
       <value>true</value>
       <description>Enable container reuse across task attempts. Default is
    set to false.</description>
   </property>
   ```

d. Define number of task attempts to be run on a single container before the container is released. Use -1 to disable this limit.

```
<property>
    <name>yarn.app.mapreduce.am.scheduler.reuse.max-attempts-per-
container</name>
    <value>-1</value>
    <description>Defines number of task attempts to be run on a single
 container before the container is
            released. To disable this limit, set the value of this
 property to -1.</description>
</property>
```

Note

For certain workloads, some jobs tend to have memory leaks and so we recommend that you set the container reuse property to a manageable value (for example 5 or 10).

4. On all the client nodes and Tez hostmachine, edit `hadoop-env.sh` and set HADOOP_CLASSPATH as shown below:

```
export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$TEZ_HOME/*:$TEZ_HOME/lib/*
```

where, *$TEZ_HOME* is the location of the directory that contains all the Tez JAR files. By default, *$TEZ_HOME* is set to **`/usr/lib/tez`**.

7.3. Enable and Use Tez Service

Use the following steps to use Tez Service:

1. Enable Tez Service

2. Start Tez Service

3. Submit Jobs to Tez Service

7.3.1. Enable Tez Service

Warning

These instructions are now obsolete and no longer provide a Supported version of Tez. Use the information found in HDP 2.1 regarding Installing and Configuring Tez.

1. Create directories and configure ownership + permissions on the appropriate hosts as described below. If any of these directories already exist, we recommend deleting and recreating them.

   On all the client nodes, create the following directory:

   a. Create Hadoop configuration directory for Tez. For example, `/etc/hadoop-tez/conf/`.

```
mkdir -p $HADOOP_TEZ_DIR
```

b. Copy the contents from the Hadoop configuration directory (`etc/hadoop/conf`) to the Hadoop-Tez configuration directory.

c. Set appropriate permissions for the Hadoop-Tez configuration directory.

For example, if the Hive user is responsible for submitting the queries to Tez Service, the permissions should be set as shown below:

```
chown -R $HIVE_USER:$HADOOP_GROUP $HADOOP_TEZ_DIR;
chmod -R 755 $HADOOP_TEZ_DIR;
```

where:

- *$HADOOP_TEZ_DIR* is the Hadoop configuration directoy for Tez. For example, `/etc/hadoop-tez/conf/`.
- *$HIVE_USER* is the user owning the Hive services. For example, `hive`.
- *$HADOOP_GROUP* is a common group shared by services. For example, `hadoop`.

2. Enable Tez AM using the instructions provided here.

3. Enable Tez Service for Hive.

a. Create a directory to store the Hive JAR files (for example, `/apps/hive/tez-ampool-jars`).

```
hadoop dfs -mkdir -p $HIVE_JAR_DIR
hadoop dfs -put $HIVE_HOME/lib/hive*.jar $HIVE_JAR_DIR
```

Set appropriate permissions for the Tez Service user. For example, if the Hive user is responsible for submitting the queries to Tez Service, the permissions should be set as shown below:

```
hadoop fs chown -R $HIVE_USER:$HADOOP_GROUP $HIVE_JAR_DIR;
hadoop fs chmod -R 755 $HIVE_JAR_DIR;
```

where:

- *$HIVE_JAR_DIR* is the directory that contains Hive JAR files. For example, `/apps/hive/tez-ampool-jars` and is used by the `tez.ampool.mr-am.job-jar-path` property.

Note

User submitting jobs should have appropriate access permissions to the files listed in `tez.ampool.mr-am.job-jar-path` property.

- *$HIVE_HOME* is the location of the Hive JAR files. For example, `/usr/lib/hive`.
- *$HIVE_USER* is the user owning the Hive services. For example, `hive`.
- *$HADOOP_GROUP* is a common group shared by services. For example, `hadoop`.

b. Create a comma-spearated list of all the file paths in the uploaded directory (*$HIVE_JAR_DIR*) on HDFS.

Continuing with the previous example, create comma-separated list of file paths from the `/apps/hive/tez-ampool-jars` directory.

These file paths would be in the form of `/apps/hive/tez-ampool/hive*.jar`.

4. On the Tez Service host machine, edit *$TEZ_CONF_DIR*`/tez-ampool-site.xml` and modify the following properties:

(where *$TEZ_CONF_DIR* is the directory that contains all the Tez configuration files and by default is set to `/etc/tez/conf`)

```
<property>
    <name>tez.ampool.ws.port</name>
    <value>12999</value>
    <description>Port to use for AMPoolService status.</description>
</property>
```

```
<property>
    <name>tez.ampool.am-pool-size</name>
    <value>3</value>
    <description>Minimum size of AM Pool.</description>
</property>
```

```
<property>
    <name>tez.ampool.max.am-pool-size</name>
    <value>5</value>
    <description>Maximum size of AM Pool.</description>
</property>
```

```
<property>
    <name>tez.ampool.launch-new-am-after-app-completion</name>
    <value>true</value>
    <description>This property determines the time to launch new AM.
                 If set to true, new AM is launched after an existing AM in
 the pool completes execution. Otherwise,
      AM is launched as soon as a job is assigned to an AM from the  pool.</
description>
</property>
```

```
<property>
    <name>tez.ampool.max-am-launch-failures</name>
    <value>10</value>
    <description>Number of launch failures to account for unassigned AMs
 before shutting down AMPoolService.</description>
</property>
```

```
<property>
    <name>tez.ampool.address</name>
    <value>$Tez_Host_Machine:10030</value>
    <description>Address on which to run the ClientRMProtocol proxy.</
description>
</property>
```

```
<property>
    <name>tez.ampool.mr-am.memory-allocation-mb</name>
    <value>1536</value>
    <description>Memory to use when launching the lazy MR AM.</
description>
</property>
```

```
<property>
    <name>tez.ampool.mr-am.queue-name</name>
    <value>default</value>
    <description>Queue to which the Lazy MRAM is to be submitted to.</
description>
</property>
```

The value of the following `tez.ampool.mr-am.job-jar-path` property will be the file path of the uploaded directory (*`$HIVE_JAR_DIR`*) on HDFS (from Step - 4 above) .

For example,

```
<property>
    <name>tez.ampool.mr-am.job-jar-path</name>
    <value>
hadoop dfs -mkdir -p $HIVE_JAR_DIR
hadoop dfs -put $HIVE_HOME/hive*.jar $HIVE_JAR_DIR
hadoop dfs -put $HIVE_HOME/hive*.war $HIVE_JAR_DIR
</value>
    <description>Location of the Hive JAR files on HDFS.</description>
</property>
```

where *`$HIVE_JAR_DIR`* is the directory that contains Hive JAR files. For example, `/apps/hive/tez-ampool-jars`.

User submitting jobs should have appropriate access permissions to the files listed in `tez.ampool.mr-am.job-jar-path` property.

```
<property>
    <name>tez.ampool.tmp-dir-path</name>
    <value>/tmp/ampoolservice/</value>
    <description>Local filesystem path for staging local data used by
 AMPoolClient/AMPoolService.</description>
</property>
```

```
<property>
    <name>tez.ampool.am.staging-dir</name>
    <value>/tmp/tez/ampool/staging/</value>
    <description>Path on HDFS used by AMPoolService to upload lazy-mr-am
 config.</description>
</property>
```

Important

The user starting the Tez Service must have appropriate permissions to the `tez.ampool.am.staging-dir` directory.

5. On all the client nodes and the Tez Service host machine, edit *`$TEZ_CONF_DIR`*`/lazy-mram-site.xml` and modify the following property:

(where *$TEZ_CONF_DIR* is the directory that contains all the Tez configuration files and by default is set to /etc/tez/conf)

```
<property>
    <name>yarn.app.mapreduce.am.lazy.prealloc-container-count</name>
    <value>1</value>
    <description>Number of containers to pre-allocate after starting up. To
 use preallocation, the value for this property must be set to a non-zero
 value.</description>
</property>
```

Important

The tez.ampool.am-pool-size, tez.ampool.max-am-pool-size, and yarn.app.mapreduce.am.lazy.prealloc-container-count parameters affect the cluster resources utilized by the Tez Service.

The tez.ampool.am-pool-size parameter determines the minimum number of YARN containers utilized and is equal to the number of Tez AMs launched. Each Tez AM, in turn, will allocate at the most N containers where N is defined by yarn.app.mapreduce.am.lazy.prealloc-container-count.

The above two together define the resource utilization and therefore should be set carefully to ensure that the Tez Service does not occupy all the resources in your cluster.

7.3.2. Start Tez Service

Warning

These instructions are now obsolete and no longer provide a Supported version of Tez. Use the information found in HDP 2.1 regarding Installing and Configuring Tez.

Use the following instructions to start the Tez Service:

1. Start the Tez Service.

   Execute the following command on the Tez Service host machine:

   ```
   $TEZ_HOME/sbin/tez-daemon.sh start ampoolservice
   ```

   where, *$TEZ_HOME* is the location of the directory that contains all the Tez JAR files. By default, *$TEZ_HOME* is set to **/usr/lib/tez**.

   ### Important

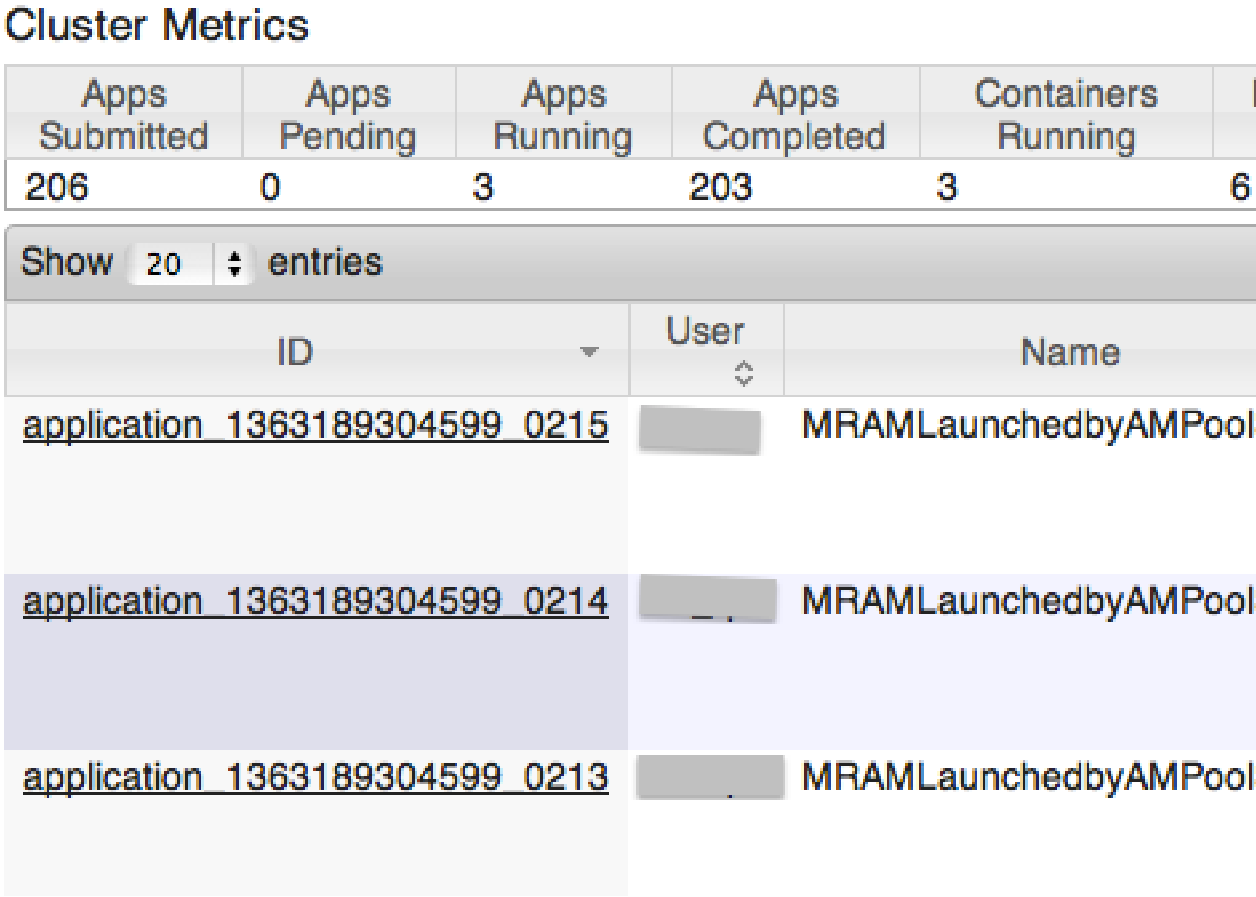
   Ensure that the user submitting the jobs and the user starting the Tez Service are identical.

   For Tez Service that is used with Hive, you must start the Tez Service as user hive.

2. Validate if Tez AM is enabled successfully.

   Browse to the ResourceManager (RM) web user interface at `http://$resourcemanager.full.hostname:8088/cluster` .

   Your RM web UI should have the following status message as shown in the screenshot below:



Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	
206	0	3	203	3	6

Show 20 entries

ID	User	Name
application_1363189304599_0215	[redacted]	MRAMLaunchedbyAMPool
application_1363189304599_0214	[redacted]	MRAMLaunchedbyAMPool
application_1363189304599_0213	[redacted]	MRAMLaunchedbyAMPool

Note

If any of the applications remain in an **ACCEPTED** or **SUBMITTED** state, this implies that your existing configuration is overutilizing the cluster resources.

Ensure that you tune the configuration parameters as instructed here to avoid cluster over-utilization and restart the service.

To stop the service, execute the following command on the Tez Service host machine:

```
$TEZ_HOME/sbin/tez-daemon.sh stop ampoolservice
```

7.3.3. Submit Hive Queries to Tez Service

Warning

These instructions are now obsolete and no longer provide a Supported version of Tez. Use the information found in HDP 2.1 regarding Installing and Configuring Tez.

Use the following instructions to submit Hive queries to Tez Service:

1. On the Tez Service host machine, browse to the Hadoop-Tez configuration directory created here and open the `yarn-site.xml` file.

   Modify the following property:

   ```
   <property>
       <name>yarn.resourcemanager.address</name>
      <value>$Tez_Host_Machine:10030</value>
       <description>Match the value specified in the tez.ampool.address
    property.</description>
   </property>
   ```

2. Submit Hive queries to the Tez Service.

   You can use either one of the following options:

   - **Option I:** From command line

     ```
     hive -e '$HIVE_QUERY' -hiveconf yarn.resourcemanager.
     address=$Tez_Host_Machine:10030
     ```

   - **Option II:**

     a. Edit `etc/hive/conf/hive-env.sh` and add the following environment variables:

        ```
        export HADOOP_CONF_DIR="$HADOOP_TEZ_DIR"
        export YARN_CONF_DIR="$HADOOP_TEZ_DIR"
        ```

        where *$HADOOP_TEZ_DIR* is the Hadoop configuration directory for Tez created here. For example, `etc/hadoop-tez/conf`.

     b. From the Hive client, execute the following command:

        ```
        hive -e "$HIVE_QUERY"
        ```

   where *$HIVE_QUERY* is your Hive query. For example, `select count(*) from employee`.

7.4. Optional: Disable Tez

Warning

These instructions are now obsolete and no longer provide a Supported version of Tez. Use the information found in HDP 2.1 regarding Installing and Configuring Tez.

Use the following instructions to disable Tez.

On all the client nodes and Tez Service host machine, edit `mapred-site.xml` and modify the following properties as shown below:

1. Disable Tez AM:

```
<property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
    <description>Name of the MapReduce framework. Default value is yarn.</
description>
</property>
```

2. Remove value for the Mapreduce CLASSPATH property:

```
<property>
    <name>mapreduce.application.classpath</name>
    <value></value>
    <description>Classpath for MapReduce applications.</description>
</property>
```

3. Stop the service. Execute the following command on the Tez Service host machine:

```
$TEZ_HOME/sbin/tez-daemon.sh stop ampoolservice
```

   where, *$TEZ_HOME* is the location of the directory that contains all the Tez JAR files. By default, *$TEZ_HOME* is set to **`/usr/lib/tez`**.

7.5. Troubleshooting

The following information can help you troubleshoot issues you may run into with your Tez AM and Tez Service configuration.

Warning

These instructions are now obsolete and no longer provide a Supported version of Tez. Use the information found in HDP 2.1 regarding Installing and Configuring Tez.

7.5.1. Getting the Logs

The first thing to do if you run into trouble is to find the logs.Tez AM and Tez Service logs are found at `/var/log/tez` on your Tez Service host machine.

7.5.2. Quick Checks

- Ensure that the /*$HADOOP_HOME*/`bin/hadoop` file exists on the Tez Service host machine.
- User submitting jobs should have appropriate access permissions to the files listed in `tez.ampool.mr-am.job-jar-path` property.
- The user starting the Tez Service must have appropriate permissions to the `tez.ampool.am.staging-dir` directory.

- Ensure that you restart the Tez service, each time the ResourceManager is restarted.

  Execute the following commands on the Tez Service host machine:

  ```
  $TEZ_HOME/sbin/tez-daemon.sh stop ampoolservice
  $TEZ_HOME/sbin/tez-daemon.sh start ampoolservice
  ```

  where, *$TEZ_HOME* is the location of the directory that contains all the Tez JAR files. By default, *$TEZ_HOME* is set to **/usr/lib/tez**.

  ### Important

  Ensure that the user submitting the jobs and the user starting the Tez Service are identical.

  For Tez Service that is used with Hive, you must start the Tez Service as user `hive`.

7.5.3. Specific Issues

Warning

These instructions are now obsolete and no longer provide a Supported version of Tez. Use the information found in HDP 2.1 regarding Installing and Configuring Tez.

7.5.3.1. Problem: AMs fail to launch.

Solution:

- If your `tez.ampool.am-pool-size`, `tez.ampool.max-am-pool-size`, and `yarn.app.mapreduce.am.lazy.prealloc-container-count` parameters are incorrectly configured, the AMs will fail to launch because these parameters affect the cluster resources utilized by the Tez Service.

- The `tez.ampool.am-pool-size` parameter determines the minimum number of YARN containers utilized and is equal to the number of Tez AMs launched.

  Each Tez AM, in turn, will allocate at the most N containers where N is defined by `yarn.app.mapreduce.am.lazy.prealloc-container-count`.

  The above two together define the resource utilization and therefore should be set carefully to ensure that the Tez Service does not occupy all the resources in your cluster.

- We recommend that you configure these parameters such that they should be 20% of your cluster resources.

- On the Tez Service host machine, edit `tez-ampool-site.xml` and reduce the values for the following properties:

  ```
  <property>
      <name>tez.ampool.am-pool-size</name>
      <value>3</value>
      <description>Minimum size of AM Pool.</description>
  </property>
  ```

```
<property>
    <name>tez.ampool.max.am-pool-size</name>
    <value>5</value>
    <description>Maximum size of AM Pool.</description>
</property>
```

- On all the client nodes and the Tez Service host machine, edit `lazy-mram-site.xml` and reduce the values for the following property:

```
<property>
    <name>yarn.app.mapreduce.am.lazy.prealloc-container-count</name>
    <value>1</value>
    <description>Number of containers to pre-allocate after starting up. To
 use preallocation, the value for this property must be set to a non-zero
 value.</description>
</property>
```

7.5.3.2. Problem: "Cannot initialize Cluster" exception.

If you see an error similar to this during job submission, it indicates that the HADOOP_CLASSPATH is incorrectly configured:

```
ERROR security.UserGroupInformation: PriviledgedActionException  as:<user>
 (auth:SIMPLE)
   cause:java.io.IOException: Cannot  initialize Cluster. Please check your
 configuration formapreduce.framework.name and
   the correspond server addresses. java.io.IOException: Cannot initialize
 Cluster. Please
   check your configuration for mapreduce.framework.name and the correspond
 server addresses.
```

Solution: This issue is caused when HADOOP_CLASSPATH is incorrectly configured. To set HADOOP_CLASSPATH:

- On the Tez Service host machine, edit `hadoop-env.sh` and modify the following parameter:

```
export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$TEZ_HOME/*:$TEZ_HOME/lib/*
```

where, *$TEZ_HOME* is the location of the directory that contains all the Tez JAR files. By default, *$TEZ_HOME* is set to **`/usr/lib/tez`**.

8. Installing WebHCat

This section describes installing and testing WebHCat, which provides a REST interface to Apache HCatalog services like job submission and eventing.

Use the following instructions to install WebHCat:

1. Install the WebHCat RPMs
2. Set Directories and Permissions
3. Modify WebHCat Configuration Files
4. Set Up HDFS User and Prepare WebHCat Directories On HDFS
5. Validate the Installation

8.1. Install the WebHCat RPMs

On the WebHCat server machine, install the necessary RPMs.

```
yum install hcatalog webhcat-tar-hive webhcat-tar-pig
```

8.2. Set Directories and Permissions

Create directories and configure ownership + permissions on the appropriate hosts as described below.

If any of these directories already exist, we recommend deleting and recreating them. Use the following instructions to set up Pig configuration files :

1. We strongly suggest that you edit and source the files included in `scripts.zip` file (downloaded in Download Companion Files).

   Alternatively, you can also copy the contents to your `~/.bash_profile`) to set up these environment variables in your environment.

2. Execute these commands on your WebHCat server machine to create log and pid directories.

   ```
   mkdir -p $WEBHCAT_LOG_DIR
   chown -R $WEBHCAT_USER:$HADOOP_GROUP $WEBHCAT_LOG_DIR
   chmod -R 755 $WEBHCAT_LOG_DIR
   ```

   ```
   mkdir -p $WEBHCAT_PID_DIR
   chown -R $WEBHCAT_USER:$HADOOP_GROUP $WEBHCAT_PID_DIR
   chmod -R 755 $WEBHCAT_PID_DIR
   ```

   where:

   - *$WEBHCAT_LOG_DIR* is the directory to store the WebHCat logs. For example, `var/log/webhcat`.

- *$WEBHCAT_PID_DIR* is the directory to store the WebHCat process ID. For example, `/var/run/webhcat`.
- *$WEBHCAT_USER* is the user owning the WebHCat services. For example, `hcat`.
- *$HADOOP_GROUP* is a common group shared by services. For example, `hadoop`.

8.3. Modify WebHCat Configuration Files

Use the following instructions to modify the WebHCat config files:

1. Extract the WebHCat configuration files

   From the downloaded `scripts.zip` file, extract the files in `configuration_files/webhcat` directory to a temporary location.

2. Modify the configuration files

   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. See Define Environment Parameters for more information.

   a. Edit the `webhcat-site.xml` and modify the following properties:

   ```
   <property>
    <name>templeton.hive.properties</name>
    <value>hive.metastore.local=false, hive.metastore.uris=thrift:/
   /$metastore.server.full.hostname:9083,hive.metastore.sasl.enabled=no,
   hive.metastore.execute.setugi=true</value>
    <description>Properties to set when running Hive.</description>
   </property>
   ```

   ```
   <property>
    <name>templeton.zookeeper.hosts</name>
    <value>$zookeeper1.full.hostname:2181,$zookeeper1.full.hostname:2181,..
   </value>
    <description>ZooKeeper servers, as comma separated HOST:PORT pairs.</
   description>
   </property>
   ```

   ```
   <property>
   <name>templeton.controller.map.mem</name>
   <value>1600</value>
   <description>Total virtual memory available to map tasks.</description>
   </property>
   ```

3. Set up the WebHCat configuration files.

   a. Delete any existing WebHCat configuration files:

   ```
    rm -rf  $WEBHCAT_CONF_DIR/*
   ```

   b. Copy all the config files to *$WEBHCAT_CONF_DIR* and set appropriate permissions:

```
chown -R $WEBHCAT_USER:$HADOOP_GROUP $WEBHCAT_CONF_DIR
chmod -R 755 $WEBHCAT_CONF_DIR
```

where:

- *$WEBHCAT_CONF_DIR* is the directory to store theWebHCat configuration files. For example, `/etc/hcatalog/conf/webhcat`.
- *$WEBHCAT_USER* is the user owning the WebHCat services. For example, `hcat`.
- *$HADOOP_GROUP* is a common group shared by services. For example, `hadoop`.

8.4. Set Up HDFS User and Prepare WebHCat Directories On HDFS

1. Set up the HDFS user.

   ```
   Login as $HDFS_USER
   hadoop fs -mkdir /user/$WEBHCAT_USER
   hadoop fs -chown -R $WEBHCAT_USER:$WEBHCAT_USER /user/$WEBHCAT_USER
   hadoop fs -mkdir /apps/webhcat
   ```

2. Prepare WebHCat directories on HDFS.

   ```
   hadoop dfs -copyFromLocal /usr/share/HDP-webhcat/pig.tar.gz /apps/webhcat/
   hadoop dfs -copyFromLocal /usr/share/HDP-webhcat/hive.tar.gz /apps/webhcat/
   hadoop dfs -copyFromLocal /usr/lib/hadoop/contrib/streaming/hadoop-
   streaming*.jar /apps/webhcat/
   ```

3. Set appropriate permissions for the HDFS user and the webhcat directory.

   ```
   hadoop fs -chown -R $WEBHCAT_USER:users /apps/webhcat
   hadoop fs -chmod -R 755 /apps/webhcat
   ```

   where:

   - *$HDFS_USER* is the user owning the HDFS services. For example, `hdfs`.
   - *$WEBHCAT_USER* is the user owning the WebHCat services. For example, `hcat`.

8.5. Validate the Installation

1. Start the WebHCat server.

   ```
   <login as $WEBHCAT_USER>
   /usr/lib/hcatalog/sbin/webhcat_server.sh start
   ```

2. From the browser, type:

   ```
   http://$WebHCat.server.full.hostname:50111/templeton/v1/status
   ```

   You should see the following output:

   ```

   {"status":"ok","version":"v1"}
   ```

9. Installing HBase and Zookeeper

This section describes installing and testing Apache HBase, a distributed, column-oriented database that provides the ability to access and manipulate data randomly in the context of the large blocks that make up HDFS. It also describes installing and testing Apache ZooKeeper, a centralized tool for providing services to highly distributed systems. Use the following instructions to deploy HBase and ZooKeeper RPMs:

1. Install the HBase and ZooKeeper RPMs
2. Set Directories and Permissions
3. Set Up the Configuration Files
4. Validate the Installation

9.1. Install the HBase and ZooKeeper RPMs

In a terminal window, type:

```
yum install zookeeper hbase
```

9.2. Set Directories and Permissions

Create directories and configure ownership + permissions on the appropriate hosts as described below.

If any of these directories already exist, we recommend deleting and recreating them. Use the following instructions to create appropriate directories:

1. We strongly suggest that you edit and source the files included in `scripts.zip` file (downloaded in Download Companion Files).

   Alternatively, you can also copy the contents to your `~/.bash_profile`) to set up these environment variables in your environment.

2. Execute the following commands on all nodes:

```
mkdir -p $HBASE_LOG_DIR;
chown -R $HBASE_USER:$HADOOP_GROUP $HBASE_LOG_DIR;
chmod -R 755 $HBASE_LOG_DIR;
```

```
mkdir -p $HBASE_PID_DIR;
chown -R $HBASE_USER:$HADOOP_GROUP $HBASE_PID_DIR;
chmod -R 755 $HBASE_PID_DIR;
```

```
mkdir -p $ZOOKEEPER_LOG_DIR;
chown -R $ZOOKEEPER_USER:$HADOOP_GROUP $ZOOKEEPER_LOG_DIR;
chmod -R 755 $ZOOKEEPER_LOG_DIR;
```

```
mkdir -p $ZOOKEEPER_PID_DIR;
chown -R $ZOOKEEPER_USER:$HADOOP_GROUP $ZOOKEEPER_PID_DIR;
chmod -R 755 $ZOOKEEPER_PID_DIR;
```

```
mkdir -p $ZOOKEEPER_DATA_DIR;
```

```
chmod -R 755 $ZOOKEEPER_DATA_DIR;
chown -R $ZOOKEEPER_USER:$HADOOP_GROUP $ZOOKEEPER_DATA_DIR;
```

where:

- *$HBASE_LOG_DIR* is the directory to store the HBase logs. For example, /var/log/hbase.
- *$HBASE_PID_DIR* is the directory to store the HBase process ID. For example, /var/run/hbase.
- *$HBASE_USER* is the user owning the HBase services. For example, hbase.
- *$ZOOKEEPER_USER* is the user owning the ZooKeeper services. For example, zookeeper.
- *$ZOOKEEPER_LOG_DIR* is the directory to store the ZooKeeper logs. For example, /var/log/zookeeper.
- *$ZOOKEEPER_PID_DIR* is the directory to store the ZooKeeper process ID. For example, /var/run/zookeeper.
- *$ZOOKEEPER_DATA_DIR* is the directory where ZooKeeper will store data. For example, /grid/hadoop/zookeeper/data.
- *$HADOOP_GROUP* is a common group shared by services. For example, hadoop.

9.3. Set Up the Configuration Files

There are several configuration files that need to be set up for HBase and ZooKeeper.

- Extract the HBase and ZooKeeper configuration files.

  From the downloaded scripts.zip file, extract the files in configuration_files/hbase and configuration_files/zookeeper directory to separate temporary directories.

- Modify the configuration files.

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

  1. Edit the zoo.cfg and modify the following properties:

     ```
     <property>
      <name>server.1</name>
      <value>$zk.server1.full.hostname:2888:3888</value>
      <description>Enter the 1st ZooKeeper hostname</description>
     </property>
     ```

     ```
     <property>
      <name>server.2</name>
      <value>$zk.server2.full.hostname:2888:3888</value>
      <description>Enter the 2nd ZooKeeper hostname</description>
     </property>
     ```

```
<property>
 <name>server.3</name>
 <value>$zk.server3.full.hostname:2888:3888</value>
 <description>Enter the 3rd ZooKeeper hostname</description>
</property>
```

2. Edit the `hbase-site.xml` and modify the following properties:

```
<property>
 <name>hbase.rootdir</name>
 <value>hdfs://$hbase.namenode.full.hostname:8020/apps/hbase/data</value>

 <description>Enter the HBase NameNode server hostname</description>
</property>
```

```
<property>
 <name>hbase.zookeeper.quorum</name>
 <value>$zk.server1.full.hostname,$zk.server2.full.hostname,$zk.server3.
full.hostname</value>
 <description>Comma separated list of Zookeeper servers (match to what is
 specified in zoo.cfg but without portnumbers)</description>
</property>
```

3. Edit the `regionservers` file and list all the RegionServers hostnames (separated by newline character) in your environment. For example, see the sample `regionservers` file with hostnames `RegionServer1` through `RegionServer9`.

```
RegionServer1
RegionServer2
RegionServer3
RegionServer4
RegionServer5
RegionServer6
RegionServer7
RegionServer8
RegionServer9
```

- Copy the configuration files

  1. On all hosts create the config directory:

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

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

  2. Copy all the HBase configuration files to *$HBASE_CONF_DIR* and the ZooKeeper configuration files to *$ZOOKEEPER_CONF_DIR* directory.

  3. Set appropriate permissions:

```

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

```

chmod a+x $ZOOKEEPER_CONF_DIR/;
```

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

where:

- *$HBASE_CONF_DIR* is the directory to store the HBase configuration files. For example, `/etc/hbase/conf`.
- *$HBASE_USER* is the user owning the HBase services. For example, `hbase`.
- *$ZOOKEEPER_CONF_DIR* is the directory to store the ZooKeeper configuration files. For example, `/etc/zookeeper/conf`.
- *$ZOOKEEPER_USER* is the user owning the ZooKeeper services. For example, `zookeeper`.

9.4. Validate the Installation

Use these steps to validate your installation.

1. Start HBase and ZooKeeper.

   a. Execute this command from the each ZooKeeper node:

   ```

   <login as $ZOOKEEPER_USER>
   /usr/lib/zookeeper/bin/zkServer.sh start $ZOOKEEPER_CONF_DIR/zoo.cfg
   ```

   b. Execute this command from the HBase Master node:

   ```
   <login as $HDFS_USER>
   /usr/lib/hadoop/bin/hadoop fs -mkdir -p /apps/hbase
   /usr/lib/hadoop/bin/hadoop fs -chown -R hbase /apps/hbase
   <login as $HBASE_USER>
   /usr/lib/hbase/bin/hbase-daemon.sh --config $HBASE_CONF_DIR start master
   ```

   c. Execute this command from each HBase Region Server node:

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

   where:

   - *$HBASE_CONF_DIR* is the directory to store the HBase configuration files. For example, `/etc/hbase/conf`.
   - *$HBASE_USER* is the user owning the HBase services. For example, `hbase`.
   - *$HDFS_USER* is the user owning the HDFS services. For example, `hdfs`.
   - *$ZOOKEEPER_CONF_DIR* is the directory to store the ZooKeeper configuration files. For example, `/etc/zookeeper/conf`.
   - *$ZOOKEEPER_USER* is the user owning the ZooKeeper services. For example, `zookeeper`.

2. Smoke Test HBase and ZooKeeper.

   From a terminal window, enter:

   ```
   echo "echo status | hbase shell" > /tmp/hbasesmoke.sh
   echo "echo disable 'usertable' | hbase shell" >> /tmp/hbasesmoke.sh
   echo "echo drop 'usertable' | hbase shell" >> /tmp/hbasesmoke.sh
   echo "echo create 'usertable', 'family' | hbase shell" >> /tmp/hbasesmoke.sh
   echo "echo put 'usertable', 'row01', 'family:col01', 'value1' | hbase shell"
    >> /tmp/hbasesmoke.sh
   echo "echo scan 'usertable' | hbase shell" >> /tmp/hbasesmoke.sh
   sh /tmp/hbasesmoke.sh
   ```

10. Manual Install Appendix: Tarballs

Individual links to the Apache structured tarball files for the projects included with Hortonworks Data Platform are listed below:

- RHEL 5 and CentOS 5
- RHEL 6 and CentOS 6

10.1. RHEL 5 and CentOS 5

Table 10.1. RHEL/CentOS 5

Project	Download
Hadoop	hadoop-2.0.3.22-alpha.tar.gz
Pig	pig-0.10.1.22.tar.gz
Hive and HCatalog	hive-0.10.0.22.tar.gz hcatalog-0.5.0.22.tar.gz
Tez	tez-0.1.0.22.tar.gz
HBase and ZooKeeper	hbase-0.94.5.22-security.tar.gz zookeeper-3.4.5.22.tar.gz

10.2. RHEL 6 and CentOS 6

Table 10.2. RHEL/CentOS 6

Project	Download
Hadoop	hadoop-2.0.3.22-alpha.tar.gz
Pig	pig-0.10.1.22.tar.gz
Hive and HCatalog	hive-0.10.0.22.tar.gz hcatalog-0.5.0.22.tar.gz
Tez	tez-0.1.0.22.tar.gz
HBase and ZooKeeper	hbase-0.94.5.22-security.tar.gz zookeeper-3.4.5.22.tar.gz

11. Uninstalling HDP

Use the following instructions to uninstall HDP:

1. Stop all the services using the instructions provided here.
2. If HBase and ZooKeeper are installed, execute the following commands on all the cluster nodes:

   ```
   rm -f /usr/share/hbase/lib/zookeeper-$version.jar
   rm -rf $ZOOKEEPER_PID_DIR/*.pid
   rm -rf $HBASE_PID_DIR/*.pid
   ```

3. If HCatalog is installed, execute the following command on all the cluster nodes:

   ```
   yum remove hcatalog\*
   ```

4. If Hive is installed, execute the following command on all the cluster nodes:

   ```
   yum remove hive\*
   ```

5. If Tez is installed, execute the following command on all the cluster nodes:

   ```
   yum remove tez
   ```

6. If HBase is installed, execute the following command on all the cluster nodes:

   ```
   yum remove hbase\*
   ```

7. If ZooKeeper is installed, execute the following command on all the cluster nodes:

   ```
   yum remove zookeeper\*
   ```

8. If Pig is installed, execute the following command on all the cluster nodes:

   ```
   yum remove pig\*
   ```

9. If compression libraries are installed, execute the following command on all the cluster nodes:

   ```
   yum remove snappy\*
   yum remove hadoop-lzo\*
   ```

10. Uninstall Hadoop. Execute the following command on all the cluster nodes:

    ```
    yum remove hadoop\*
    ```

11. Uninstall ExtJS libraries and MySQL connector. Execute the following command on all the cluster nodes:

    ```
    yum remove extjs-2.2-1 mysql-connector-java-5.0.8-1\*
    ```

12. Delete Hadoop directories.

    ```
    rm -rf $HADOOP_HOME
    ```