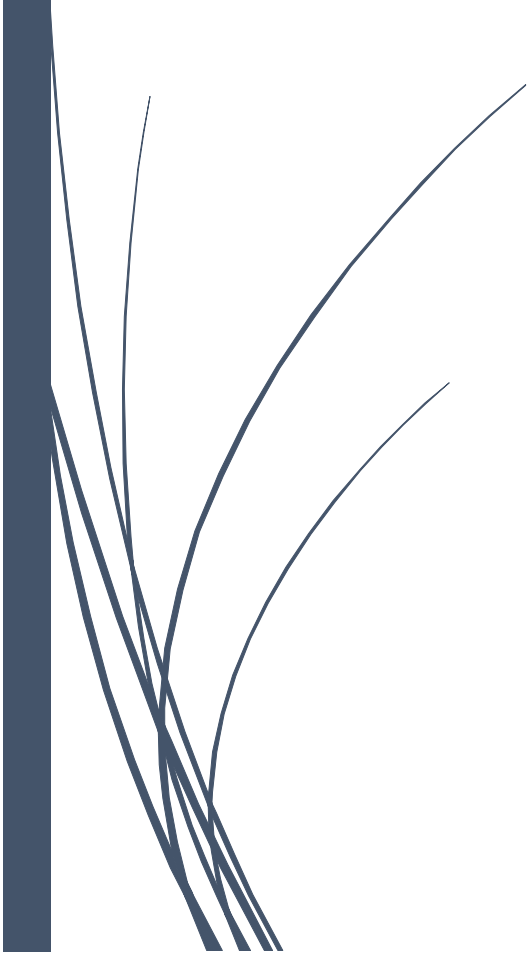
April 29, 2015

Installing Hadoop

Hortonworks Hadoop

VERSION 1.0



Mogulla, Deepak Reddy

Table of Contents

Get Linux platform ready

- Download a virtual machine software - either Oracle VirtualBox or VMWare Fusion if running windows, mac or other OSs.
- Install the virtual machine and make it ready to install Ubuntu or other Linux platforms on the VM or the machine, depending on the choice you make.
- If installing Ubuntu on a VM then download the .iso file from Ubuntu website, for the latest version, and move it to the Desktop (for easy access).
- Create a new VM and select the .iso file to install in the new VM. It automatically installs and Ubuntu is ready.

Update Linux

- Open the terminal application in Ubuntu
- Enter the commands to update Linux:
  ```
  $ sudo apt-get upgrade
  $ sudo apt-get dist-upgrade
  ```

Update/install Java:

- Check the Java version:
  ```
  $ java –version
  ```

- If there is no java then download and install it with the following command:
  ```
  $ sudo apt-get install default-jdk
  ```

- Check the java version with the same command as we did in the first step and setup the JAVA_HOME variable.

- To check where java is installed type in the below for the path:
  ```
  $ update-alternatives --config java
  ```

- JAVA_HOME is everything before the "/jre/bin/java". In this case JAVA_HOME is usr/lib/jvm/java-7-openjdk-amd64. So set JAVA_HOME using the below:
  ```
  $ export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
  ```

Setup SSH Certificates

- The below commands sets up SSH certificates so Hadoop can access the nodes with ssh without asking for passwords. Type in the following to set it up:

```
$ sudo apt-get install ssh
$ sudo apt-get install rsync
$ ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

Setup SSH connection from parent OS

- If using OS X, then the connection can be accomplished using ssh in the Terminal:

```
$ ssh <username>@<ip addr of Linux>
```

- For Windows, we may have to use Putty to setup a connection.

Download Hadoop in Linux

- Check the latest stable release and copy the mirror URL and see the contents inside to get the tar.gx file for hadoop. The current URL when this document was written is

```
http://apache.osuosl.org/hadoop/common/hadoop-2.6.0/hadoop-2.6.0.tar.gz
```

- So now we have to use the wget command to get the tar file and then untar the package.

```
$ wget http://apache.osuosl.org/hadoop/common/hadoop-2.6.0/hadoop-2.6.0.tar.gz

$ tar xfz hadoop-2.6.0.tar.gz
```

- Then we need to move the untared file to /usr/local location. We make a new folder in the local directory. Use the command:

```
$ mv hadoop-2.6.0 /usr/local/hadoop
```

- Add a dedicated user group to run this hadoop directory.

```
$ sudo addgroup hadoop
```

- Make the owner of all the files is the user chosen and the group is hadoop.

```
$ sudo chown -R hduser:hadoop hadoop
```

Configure files in Hadoop

- Open the .bashrc file in editing mode.
  `$ nano ~/.bashrc`

- Add the following at the end of the file

```
#HADOOP VARIABLES START
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
export HADOOP_INSTALL=/usr/local/hadoop
export PATH=$PATH:$HADOOP_INSTALL/bin
export PATH=$PATH:$HADOOP_INSTALL/sbin
export HADOOP_MAPRED_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_HOME=$HADOOP_INSTALL
export HADOOP_HDFS_HOME=$HADOOP_INSTALL
export YARN_HOME=$HADOOP_INSTALL
export
HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_INSTALL/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_INSTALL/lib"
#HADOOP VARIABLES END
```

- Save the file using the command CTRL + X.
- After that use the following command to acknowledge the updates variables:
  `$ source ~/.bashrc`

Updating Hadoop configuration files

Hadoop-env.sh

- Open the file in edit mode:
  `$ nano /usr/local/hadoop/etc/hadoop/hadoop-env.sh`

- Add the below line where it says '# The java implementation to use'
  `export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64`

- Save the file by pressing CTRL + X.

Core-site.xml

- Open the file in edit mode:
  `$ nano /usr/local/hadoop/etc/hadoop/core-site.xml`

- Add the lines to the file at the end:

```
<configuration>
     <property>
         <name>fs.default.name</name>
         <value>hdfs://localhost:9000</value>
     </property>
</configuration>
```

- Save the file using the command CTRL + X.

Hdfs-site.xml

- Add the namenode and datanode directory properties and then change owner for these directories.

```
$ sudo chown -R deepak:deepak namenode
$ sudo chown -R deepak:deepak datanode
```

- Open the file in edit mode:

```
$ nano /usr/local/hadoop/etc/hadoop/hdfs-site.xml
```

- Add the lines to the file at the end:

```
<configuration>
          <property>
              <name>dfs.replication</name>
              <value>1</value>
          </property>
     <property>
              <name>dfs.namenode.name.dir</name>
               <value>file:/usr/local/hadoop_dir/hdfs/nameno
de</value>
     </property>
      <property>
              <name>dfs.datanode.data.dir</name>
               <value>file:/usr/local/hadoop_dir/hdfs/datano
de</value>
       </property>
</configuration>
```

- Save the file using the command CTRL + X.

Mapred-site.xml

- Make the template and actual file.
  ```
  $ cp /usr/local/hadoop/etc/hadoop/mapred-site.xml.template
  /usr/local/hadoop/etc/hadoop/mapred-site.xml
  ```

- Open the file in edit mode:
  ```
  $ nano /usr/local/hadoop/etc/hadoop/mapred-site.xml
  ```

- Add the lines to the file at the end:
  ```
  <configuration>
       <property>
                 <name>mapred.job.tracker</name>
                 <value>localhost:9001</value>
            </property>
          <property>
             <name>mapreduce.framework.name</name>
             <value>yarn</value>
          </property>
  </configuration>
  ```

- Save the file using the command CTRL + X.

Fire up Hadoop

- Format namenode before starting hadoop:
  ```
  $ hdfs namenode -format
  ```

- Start hadoop:
  ```
  $ start-dfs.sh
  ```

- Check the instances running in hadoop:
  ```
  $ jps
  ```

Shutdown Hadoop

```
$ stop-all.sh
```