Hadoop Installation Guide - 2.6.0

Install Java:

The Hadoop framework is written in Java and thus the installation of Java is a pre-requisite.
-- Go to the home directory: $cd ~
-- Update Ubuntu packages: $sudo apt-get update
-- Upgrade Ubuntu packages: $sudo apt-get upgrade
-- Install Java using the command: $sudo apt-get install default-jdk
-- Check installed Java Version: $java -version

Adding a dedicated Hadoop user:

This step is not required by recommended because it helps to separate the Hadoop installation process from other software applications and user accounts running on the same machine.
-- Go to the home directory: $cd ~
-- Add a new group "hadoop" for Hadoop users: $sudo addgroup hadoop
-- Add new user "hduser" to the Hadoop users group: $sudo adduser --ingroup hadoop hduser
-- Add hduser as a sudo user: $sudo adduser hduser sudo

Install SSH:

SSH access is required to manage nodes (connect to remote machines). For a single-node setup of Hadoop, we need to configure SSH access to localhost.
-- Go to the home directory: $cd ~
-- Install ssh and ssh daemon: $sudo apt-get install ssh
-- Check installed ssh location: $which ssh
-- Check installed sshd (daemon) location: $which sshd

Create and Setup SSH Certificates:

A RSA key pair is created with an empty password for Hadoop to interact with nodes without user intervention.
-- Switch to the new user "hduser": $su hduser
-- Go to the home directory inside hduser: $cd ~
-- Create the RSA key using the following command: $ssh-keygen -t rsa -P ""
-- Save the key to the default folder when prompted (/home/hduser/.ssh/id_rsa): $[press enter]
-- Add the newly created key to the list of authorized keys so that Hadoop can use SSH without prompting for a password: $cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
-- Run the SSH localhost to check if no password prompt: $ssh localhost

Install Hadoop:

-- Switch to the new user "hduser": $su hduser
-- Go to the home directory inside hduser: $cd ~
-- Download Hadoop using the command:
$wget http://www.apache.org/dist/hadoop/common/hadoop-2.6.0/hadoop-2.6.0.tar.gz
-- If the above link fails to work, download Hadoop from the apache website (tar.gz file).
-- Unzip the folder with the command: $tar -xvzf hadoop-2.6.0.tar.gz
-- The above command will create a new Hadoop file (hadoop-2.6.0), to check use the command: $ls -l
-- Rename the unzipped file to "hadoop": $mv hadoop-2.6.0 hadoop
-- Check the present working director: $pwd #should be /home/hduser or similar
-- Move hadoop from pwd to /usr/local/hadoop: $sudo mv /home/hduser/hadoop /usr/local/hadoop
-- Change owner of hadoop to hduser: $sudo chown -R hduser:hadoop /usr/local/hadoop
-- To check permissions of the Hadoop folder, go to /usr/local and enter the command: $ls -al

Setup Hadoop Configuration Files:

A total of 5 configurations files have to be modified to complete the Hadoop setup.

File1: .bashrc
-- Switch to the new user "hduser": $su hduser
-- Go to the home directory inside hduser: $cd ~
-- Check if the .bashrc file exists: $ls -al
-- Get JAVA_HOME environmental variable path using command: $update-alternatives --config java
-- Make note of the JAVA_HOME path (copy it and save for later use).
-- Open the .bashrc file using gedit/nano/vi: $sudo gedit .bashrc #Used gedit in this case
-- Append the following code to the end of the .bashrc file:

```
#HADOOP VARIABLES START
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
#copy the java path found above excluding jre
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
```

-- Use the JAVA_HOME path found previously and assign it to the JAVA_HOME variable in the .bashrc file
-- Make sure rest of the above syntax is identical with the spacing between the variables and =
-- Save and close the .bashrc file
-- Run the following command in the terminal: $source ~/.bashrc

File2: hadoop-env.sh
-- Switch to the new user "hduser": $su hduser
-- Go to the following directory inside hduser: $cd /usr/local/hadoop/etc/hadoop
-- Open the hadoop-env.sh file using gedit/nano/vi: $sudo gedit hadoop-env.sh
-- Inside the file, search for the variable export JAVA_HOME
-- replace the export JAVA_HOME with the following line:

```
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
#copy the java path found excluding jre
```

-- The JAVA_HOME path found previously is assigned to the JAVA_HOME variable in the hadoop-env.sh file.
-- Save and close the hadoop-env.sh file.
NOTE: The JAVA_HOME environmental path was found using the command:
$update-alternatives --config java

File3: core-site.xml
The core-site.xml file contains configuration properties that Hadoop uses when starting up.
-- Switch to the new user "hduser": $su hduser
-- Go to the home directory inside hduser: $cd ~
-- Make a temporary directory using the command: $sudo mkdir -p /app/hadoop/tmp
-- Make hduser owner of temporary directory: $sudo chown hduser:hadoop /app/hadoop/tmp
-- Go to the following directory: $cd /usr/local/hadoop/etc/hadoop
-- Open the hadoop-env.sh file using gedit/nano/vi: $sudo gedit core-site.xml
-- Add the following code between the <configuration></configuration> xml tags (continued on till the next page):

```
<property>
        <name>hadoop.tmp.dir</name>
        <value>/app/hadoop/tmp</value>
        <description>A base for other temporary directories.</description>
</property>

<property>
        <name>fs.default.name</name>
        <value>hdfs://localhost:54310</value>
```

```
        <description>The name of the default file system.  A URI whose
        scheme and authority determine the FileSystem implementation.  The
        uri's scheme determines the config property (fs.SCHEME.impl) naming
        the FileSystem implementation class.  The uri's authority is used to
        determine the host, port, etc. for a filesystem.</description>
    </property>
```

-- Make sure the above syntax is identical.
-- Save and close the core-site.xml file.

File4: mapred-site.xml
The mapred-site.xml file is used to specify which framework is being used for MapReduce. By default, the /usr/local/hadoop/etc/hadoop/ folder contains the mapred-site.xml.template file. This file has to be renamed/copied with the name mapred-site.xml
-- Switch to the new user "hduser": $su hduser
-- Go to the following directory: $cd /usr/local/hadoop/etc/hadoop
-- Make a copy of the mapred-site.xml.template: $cp mapred-site.xml.template mapred-site.xml
-- Open the newly created file using gedit/nano/vi: $sudo gedit mapred-site.xml
-- Add the following code between the <configuration></configuration> xml tags:

```
    <property>
        <name>mapred.job.tracker</name>
        <value>localhost:54311</value>
        <description>The host and port that the MapReduce job tracker runs
        at.  If "local", then jobs are run in-process as a single map
        and reduce task.
        </description>
    </property>
```

-- Make sure the above syntax is identical.
-- Save and close the mapred-site.xml file.

File5: hdfs-site.xml
The hdfs-site.xml file needs to be configured for each host in the cluster. It is used for specifying the directories which will be used as the namenode and the datanode on that host. Before editing this file, we need to create two directories which will contain the namenode and the datanode for this Hadoop installation.
-- Switch to the new user "hduser": $su hduser
-- Go to the home directory in hduser: $cd ~
-- Create a namenode folder: $sudo mkdir -p /usr/local/hadoop_store/hdfs/namenode
-- Create a datanode folder: $sudo mkdir -p /usr/local/hadoop_store/hdfs/datanode
-- Change owner of the folders to hduser: $sudo chown -R hduser:hadoop /usr/local/hadoop_store

-- Go to the following directory: $cd /usr/local/hadoop/etc/hadoop
-- Open the hdfs-site.xml file using gedit/nano/vi: $sudo gedit hdfs-site.xml
-- Add the following code between the <configuration></configuration> xml tags:

```
<property>
        <name>dfs.replication</name>
        <value>1</value>
        <description>Default block replication.
        The actual number of replications can be specified when the file is created.
        The default is used if replication is not specified in create time.
        </description>
</property>

<property>
        <name>dfs.namenode.name.dir</name>
        <value>file:/usr/local/hadoop_store/hdfs/namenode</value>
</property>

<property>
        <name>dfs.datanode.data.dir</name>
        <value>file:/usr/local/hadoop_store/hdfs/datanode</value>
</property>
```

-- Make sure the above syntax is identical.
-- Save and close the hdfs-site.xml file.

Format the new Hadoop File System:

The Hadoop file system needs to be formatted so that we can start to use it. The format command should be executed only once before we start using Hadoop. If this command is used again, it will destroy all the data in the Hadoop file system.

-- Switch to the new user "hduser": $su hduser
-- Go to the directory: $cd /usr/local/hadoop_store
-- Give read/write permissions to the folder hdfs using the command: $sudo chmod -R 777 hdfs
-- To check the permissions in the hadoop_store folder: $ls -al
-- Go to the home directory in hduser: $cd ~
-- Run the following command to format Hadoop: $hadoop namenode -format

Start Hadoop and check is services are running:

-- Switch to the new user "hduser": $su hduser
-- Go to the home directory in hduser: $cd ~

-- To start Hadoop use the following command: $start-all.sh
-- To check if Hadoop services are running, use the command: $jps
-- The above command should show a minimum of 5 services running, namely - DataNode, NodeManager, SecondaryNameNode, ResourceManager, NameNode
-- The commands to start/stop Hadoop services are in the folder: $cd /usr/local/hadoop/sbin
-- To stop Hadoop services use the command: $stop-all.sh

Hadoop Web Interface:
-- Switch to the new user "hduser": $su hduser
-- Start the Hadoop services: $start-all.sh
-- Open a web-browser.
-- Enter the URL http://localhost:50070/

Install Eclipse:
-- Use the following command to install Eclipse: $sudo apt-get install eclipse