Hadoop Installation Guide - 2.6.0

Command to install Homebrew via OS X terminal:
$ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
Homebrew will be installed in the /usr/local directory

Useful Homebrew Commands for OSX:
-- Updates for OSX: $brew update # (updates list of available packages)
-- Upgrades for OSX: $brew upgrade # (installs newer versions of packages)
-- List all the packages installed by brew: $brew list
-- List broken packages/installers: $brew doctor

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 Homebrew packages: $brew update
-- Upgrade Homebrew packages: $brew upgrade
-- Install Java using the commands:
    $brew tap caskroom/cask
    $brew install brew-cask
    $brew cask install java
-- Check installed Java Version: $java -version

Configure 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. A RSA key pair is created with an empty password for Hadoop to interact with nodes without user intervention.
-- Enable Remote Login: Go to -> System Preferences in OSX -> Sharing -> Check “Remote Login”
-- Create the RSA key using the following command in the terminal: $ssh-keygen -t rsa -P ""
-- Save the key to the default folder when prompted (.../.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:
-- Go to the home directory: $cd ~
-- Download Hadoop using the command:
$curl -o hadoop-2.6.0.tar.gz 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 /Users/Username or similar
-- Move hadoop from pwd to usr/local/hadoop: $mv /home/hduser/Hadoop /usr/local/hadoop

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

File1: .bash_profile
-- Go to the home directory: $cd ~
-- Check if the .bash_profile file exists: $ls -al
-- Find the java home using the command: $/usr/libexec/java_home
-- Make note of the JAVA_HOME path.
-- Open the .bash_profile file using TextEdit: $open -a TextEdit .bash_profile
-- Append the following code to the end of the .bash_profile file:

```
#HADOOP VARIABLES START
export JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk1.8.0_66.jdk/Contents/Home
#append the JAVA_HOME path found above
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 .bash_profile file.
-- Make sure the above syntax is identical with the spacing between the variables and =
-- Save and close the .bash_profile file.
-- Run the following command in the terminal: $source ~/.bash_profile

File2: hadoop-env.sh
-- Go to the home directory: $cd ~
-- Find the java home using the command: $/usr/libexec/java_home
-- Copy the java home path.
-- Go to the following directory: $cd /usr/local/hadoop/etc/hadoop

-- Open the hadoop-env.sh file using TextEdit: $open -a TextEdit 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=/Library/Java/JavaVirtualMachines/jdk1.8.0_66.jdk/Contents/Home
#copy the java path found above
```

-- The JAVA_HOME path found previously is assigned the JAVA_HOME variable in the hadoop-env.sh file.
-- Save and close the hadoop-env.sh file.

File3: core-site.xml
The core-site.xml file contains configuration properties that Hadoop uses when starting up.
-- Go to the home directory: $cd ~
-- Make a temporary directory using the command: $mkdir -p /usr/local/hadoop/tmp
-- Go to the following directory: $cd /usr/local/hadoop/etc/hadoop
-- Open the core-site.xml file using TextEdit: $open -a TextEdit core-site.xml
-- Add the following code between the <configuration></configuration> xml tags:

```
<property>
        <name>hadoop.tmp.dir</name>
        <value>/usr/local/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

-- Go to the following directory: $cd /usr/local/hadoop/etc/hadoop
-- Make a copy of mapred-site.xml.template: $cp mapred-site.xml.template mapred-site.xml
-- Open the newly created file using TextEdit: $open -a TextEdit 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 that will be used as the namenode and 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.
-- Go to the directory: $cd /usr/local/hadoop/
-- Create a new folder "hadoop_store" in this directory: $mkdir -p hadoop_store
-- In hadoop_store folder, create folder hdfs: $mkdir -p /usr/local/hadoop/hadoop_store/hdfs
-- Create namenode folder in hdfs directory: $mkdir -p /usr/local/hadoop/hadoop_store/hdfs/namenode
-- Create a datanode folder in hdfs directory: $mkdir -p /usr/local/hadoop/hadoop_store/hdfs/datanode
-- Go to the following directory: $cd /usr/local/hadoop/etc/hadoop
-- Open the hdfs-site.xml file using TextEdit: $open -a TextEdit hdfs-site.xml
-- Add the following code between the <configuration></configuration> xml tags (continued on till the next page):

```
<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/hadoop_store/hdfs/namenode</value>
</property>

<property>
        <name>dfs.datanode.data.dir</name>
        <value>file:/usr/local/hadoop/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.
-- Go to the home directory: $cd ~
-- Run the following command to format Hadoop: $hadoop namenode -format

Start Hadoop and check is services are running:
-- 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 and 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:
-- Go to the home directory in hduser: $cd ~
-- Start the Hadoop services: $start-all.sh
-- Open a web-browser.
-- Enter the URL http://localhost:50070/

Install Eclipse:
-- Update OS X: $brew update
-- Upgrade OS X: $brew upgrade
-- Install Eclipse: $brew install Caskroom/cask/eclipse-ide