Hadoop Installation Guide for Ubuntu OS

This is a guide to install Hadoop on Ubuntu OS. If you want to install it on a PC or MAC please follow instructions at Apache Hadoop website https://hadoop.apache.org/

1 HADOOP Installation Steps

1.1 Download Hadoop

One can visit Appache Hadoop website https://hadoop.apache.org/ to download the latest version. For this tutorial we will use *version* = 2.7.2 download

1.2 Prerequisites

- Make sure Java is installed on your machine i.e. `java` and `javac` are working
- if not installed, you can install it using `sudo apt-get install java`
- set `JAVA_HOME` environment variable
  - to check if it is already set, execute command `echo $JAVA_HOME`, it should print the path to jdk installation directory
  - if nothing is printed, you have to set it. The best place is the `.bashrc` file. It can be edited using a text editor i.e. `gedit ~/.bashrc`.
    (a) Set `JAVA_HOME` path with `<PATH-TO-JDK>` i.e. add this line in your `.bashrc`
    `export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64/`.
    (b) Once you have set the path you can run `source ~/.bashrc` to make the environment variable visible in your currect session.
- passwordless authentication is required for the communication between different Hadoop nodes. To check if it is enabled run `ssh localhost`
  - if it ask for password than you need to follow this http://www.linuxproblem.org/art_9.html or

```
ssh-keygen -t dsa -P '' -f ~/.ssh/id_dsa
cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
chmod 0600 ~/.ssh/authorized_keys
```

1.3 Install Hadoop

- extract hadoop-2.7.2.tar.gz in a director probably (/home/<username>/hadoop) `tar -zxf hadoop-2.7.2.tar.gz`
- change director to `cd hadoop-2.7.2`
- setting Hadoop configuration files
  - add `JAVA_HOME` in the etc/hadoop/hadoop-env.sh using editor `gedit etc/hadoop/hadoop-env.sh`
    - add/update the following

```
...
# set to the root of your Java installation
export JAVA_HOME=/usr/java/latest
...
```

    - check if Hadoop path is correctly set `bin/hadoop`
  - add following to etc/hadoop/core-site.xml using editor `gedit etc/hadoop/core-site.xml`

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

`</property>`

`</configuration>`

– add following to etc/hadoop/hdfs-site.xml `gedit etc/hadoop/hdfs-site.xml`

```
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
</configuration>
```

– Now check if the daemons are running

`jps`

output

```
9547 DataNode
9388 NameNode
9745 SecondaryNameNode
16160 Jps
```

2 HADOOP Execution Steps

2.1 Setting for executing a job

- First you need to format the filesystem... (this is usually required once)
  `./bin/hadoop namenode -format`
- Check for NameNode daemon and DataNode daemon `./sbin/start-dfs.sh`
  Output: `NameNode information available at http://localhost:50070/`
- Creating HDFS directories which are required by MapReduce jobs for execution: `./bin/hdfs dfs -mkdir /user`
  `./bin/hdfs dfs -mkdir /user/STUDENT`

2.2 Adding data to HDFS

whenever you want to run a job first you need to put your data on hdfs for that you can do. (for example)
`./bin/hdfs dfs -put <yourfile or folder> <path to hdfs>`
example: (put folder /etc/hadoop into worddata folder of hdfs
`./bin/hdfs dfs -put etc/hadoop worddata`

2.3 Executing a job

- For job execution on the Hadoop cluster you have to provide as an input a) jar file (your code), b) classname to be executed, c)input directory name and d) output directory. If your program needs additional inputs, you can add them as well.
  `./bin/hadoop jar hadoop-mapreduce-examples-2.7.2.jar grep worddata results 'dfs[a-z.]+'`
  here `input = worddata` and `output = results`
- To view the output you can execute `./bin/hdfs dfs -cat output/*`

NOTE: you can replace `./bin/hdfs` with `hdfs` if you have correct setup environment variables.

2.4 Tutorial: WordCount Source

```
./bin/hdfs dfs -mkdir wordcountinput
./bin/hdfs dfs -put ../example/Hadoop-WordCount/input/ wordcountinput
./bin/hdfs dfs -ls wordcountdata
./bin/hadoop jar ../example/Hadoop-WordCount/wordcount.jar WordCount wordcountinput
  wordcountoutput
./bin/hdfs dfs -cat wordcountoutput/*
```