Setting up Hadoop 2.7.4 Standalone and Pseudo-Distributed Modes on Ubuntu 16.04

Ubuntu Installation

If you have not installed Ubuntu 16.04, you can find instructions here.

2. Downloads

Download Hadoop 2.7.4 binary and JDK 8u92 to ~/Downloads

Apache Hadoop (binary version 2.7.4)

JavaSE JDK (version 8u92)

Pre-Installation

3.1 Configure SSH server

This is to update the repository and then install the openssh server program.

```
sudo apt-get update


sudo apt-get install openssh-server
```

3.2 Configure passworld-less ssh login

The concept is simple: to generate a private and a public keys, then add the public key to the authorised list. You can read this for more information.

```
ssh-keygen -t rsa -P ""
cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys

## THEN
sudo service ssh restart
-- OR --
```

```
sudo reboot
```

3.2 Extract downloaded files to your home directory

```
cd Downloads
```

```
tar xzvf hadoop-2.7.2.tar.gz
```

4.3 Configure .bashrc

This is to modify PATH variable.

```
cd /home<your-use-name>
```

```
gedit ./.bashrc
```

and append following lines to file

```
export HADOOP_HOME=/home/<your-user_name>/hadoop
```

```
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
```

4.4 Configure Hadoop

Configure Hadoop’s hadoop-env.sh file

```
gedit ./hadoop/etc/hadoop/hadoop-env.sh
```

and change JAVA_HOME

```
export JAVA_HOME=/home/<your-user-name>/jdk
```

4.5 Exit and re-open terminal

4.6 Run a Hadoop job on Standalone cluster

Test the *hadoop* command. The usage message should be displayed.

Test run a Hadoop job—This step 1) creates *testhadoop* directory, 2) create *input* directory inside *testhadoop*, 3) copy some input files (the .xml files), 4) run MapReduce example job, and 5) view the *output* directory using *cat*command.

```
mkdir testhadoop
```

```
cd testhadoop
```

```
mkdir input
```

```
cp ~/hadoop/etc/hadoop/*.xml input
hadoop jar ~/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.2.jar grep input output 'dfs[a-z.]+'
cat output/*
```

5.1 Configure core-site.xml and hdfs-site.xml

```
gedit ./hadoop/etc/hadoop/core-site.xml
## add these lines to the file ##
<configuration>
<property>
<name>fs.defaultFS</name>
<value>hdfs://localhost:9000</value>
</property>
</configuration>

gedit ./hadoop/etc/hadoop/hdfs-site.xml
## add these lines to the file ##
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
</configuration>
```

5.2 Format the namenode

```
hdfs namenode -format
```

5.3 Start / Stop Hadoop cluster

```
## to start hadoop cluster
```

```
start-dfs.sh
## open browser → http://localhost:50070/
## to shutdown hadoop cluster
stop-dfs.sh
```

6. YARN Setup

```
gedit ./hadoop/etc/hadoop/mapred-site.xml.template
```

save it as mapred-site.xml and then add these lines

```
## adding these lines to the file ##
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>
```

```
gedit ./hadoop/etc/hadoop/yarn-site.xml
## adding these lines to the file ##
<configuration>
<property>
<name>yarn.nodemanager.aux-services</name><value>mapreduce_shuffle</value>
</property>
</configuration>
```

6.2 Start and Stop YARN cluster

```
## to start YARN
```

```
start-yarn.sh
## open browser → http://localhost:8088/
## to start YARN
stop-yarn.sh
```

5.4 Run a Hadoop job on the cluster and view outputs

```
## create /user/<username> on HDFS


hdfs dfs -mkdir /user


hdfs dfs -mkdir /user/luck
## copy some test input files from Ubuntu to HDFS
## in this case, from Ubuntu's ~/hadoop/etc/hadoop
## to HDFS's /user/<your-user-name>/input
hdfs dfs -put ~/hadoop/etc/hadoop input
## run a Hadoop job
hadoop jar ~/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.4.jar grep input output ‘d[a-z.]+’
## display output
## view outputs straight from HDFS
hdfs dfs -cat output/*
-- OR --
## copy data from Hadoop to local filesystem first
## in this case, to testhadoop/output,
## then view the output using cat
cd testhadoop
hdfs dfs -get output output
cat output/*
```