Mansoura University
Faculty of Computers and Information Sciences
Department of Computer Science
First Semester- 2020-2021

[CS412P] Distributed Systems

Grade : Fourth grade

By : Zeinab Awad

SINGLE NODE HADOOP DEPLOYMENT

-Hadoop excels when deployed in a **fully distributed mode** on a large cluster of networked servers. However, if you are new to Hadoop and want to explore basic commands or test applications, you can configure Hadoop on a single node.
This setup, also called **pseudo- distributed mode**, allows each Hadoop daemon to run as a single Java process.

-**A Hadoop environment is configured by editing a set of configuration files:**

- bashrc
- hadoop-env.sh
- core-site.xml
- hdfs-site.xml
- mapred-site-xml
- yarn-site.xml

CONFIGURE HADOOP ENVIRONMENT VARIABLES

.Bashrc :

- Edit the *.bashrc* shell configuration file using a text editor of your choice (we will be using nano):

sudo nano .bashrc

- Define the Hadoop environment variables by adding the following content to the end of the file:

```
#Hadoop Related Options
export HADOOP_HOME=/home/hdoop/hadoop-3.2.1
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
Export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS "-Djava.library.path=$HADOOP_HOME/lib/native"
```

CONFIGURE HADOOP ENVIRONMENT VARIABLES

Once you add the variables, save and exit the *.bashrc* file.
-It is vital to apply the changes to the current running environment by using the following command:
source ~/.bashrc
-Edit hadoop-env.sh File
The *hadoop-env.sh* file serves as a master file to configure YARN, HDFS, MapReduce, and Hadoop-related project settings.
-When setting up a **single node Hadoop cluster**, you need to define which Java implementation is to be utilized. Use the previously created **$HADOOP_HOME** variable to access the *hadoop-env.sh* file:
sudo nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh
-Uncomment the **$JAVA_HOME** variable (i.e., remove the **#** sign) and add the full path to the OpenJDK installation on your system. If you have installed the same version as presented in the first part of this tutorial, add the following line:
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64

APACHE HADOOP INSTALLATION

-Edit core-site.xml File

the *core-site.xml* file defines HDFS and Hadoop core properties.

To set up Hadoop in a pseudo-distributed mode, you need to **specify the URL** for your NameNode, and the temporary directory Hadoop uses for the map and reduce process.

-Open the *core-site.xml* file in a text editor by:

sudo nano $HADOOP_HOME/etc/hadoop/core-site.xml

-Add the following configuration to override the default values for the temporary directory and add your HDFS URL to replace the default local file system setting:

<configuration> <property> <name>hadoop.tmp.dir</name>
<value>/home/hadoopu/tmpdata</value> </property>
<property> <name>fs.default.name</name> <value>hdfs://127.0.0.1:9000</value>
</property> </configuration>

-Don't forget to make new directory for temporaray data on /home/hadoopu by:

Sudo Mkdir tmpdata

APACHE HADOOP INSTALLATION

-Edit hdfs-site.xml File:

The properties in the *hdfs-site.xml* file govern the location for storing node metadata, fsimage file, and edit log file. Configure the file by defining
the **NameNode** and **DataNode storage directories**.
Additionally, the default **dfs.replication** value of **3** needs to be changed to **1** to match the single node setup.
-Use the following command to open the *hdfs-site.xml* file for editing:
sudo nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml
-Add the following configuration to the file and, if needed, adjust the Name Node and Data Node directories to your custom locations:
<configuration> <property> <name>dfs.data.dir</name>
<value>/home/hdoop/dfsdata/namenode</value> </property>

APACHE HADOOP INSTALLATION

<property> <name>dfs.data.dir</name>
<value>/home/hdoop/dfsdata/datanode</value> </property>
<property> <name>dfs.replication</name> <value>1</value>
</property> </configuration>
-If necessary, create the specific directories you defined for the **dfs.data.dir** value.

-Edit mapred-site.xml File:

-Use the following command to access the *mapred-site.xml* file and **define MapReduce values**:
sudo nano $HADOOP_HOME/etc/hadoop/mapred-site.xml
-Add the following configuration to change the default MapReduce framework name value to **yarn**:
<configuration> <property>
<name>mapreduce.framework.name</name> <value>yarn</value>
</property> </configuration>

APACHE HADOOP INSTALLATION

- Edit yarn-site.xml File :

The *yarn-site.xml* file is used to define settings relevant to **YARN**. It contains configurations for the **Node Manager,**
Resource Manager, Containers, and **Application Master**.

-Open the *yarn-site.xml* file in a text editor:

sudo nano $HADOOP_HOME/etc/hadoop/yarn-site.xml

-Append the following configuration to the file:

<configuration>
<property> <name>yarn.nodemanager.aux-services</name> <value>mapreduce_shuffle</value>
</property>
<property> <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value> </property>
<property><name>yarn.resourcemanager.hostname</name> <value>127.0.0.1</value>
</property>
<property><name>yarn.acl.enable</name> <value>0</value> </property>

APACHE HADOOP INSTALLATION

<property><name>yarn.nodemanager.envwhitelist</name> <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PERPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME</value>
</property>
</configuration>

-Format HDFS NameNode :

-It is important to **format the NameNode** before starting Hadoop services for the first time by: hdfs namenode –format ,The shutdown notification signifies the end of the NameNode format process.

-Start Hadoop Cluster

Navigate to the *hadoop-3.2.1/sbin* directory and execute the following commands to start the NameNode and DataNode:

./start-dfs.shThe system takes a few moments to initiate the necessary nodes.

Once the namenode, datanodes, and secondary namenode are up and running, start the YARN resource and nodemanagers by typing:

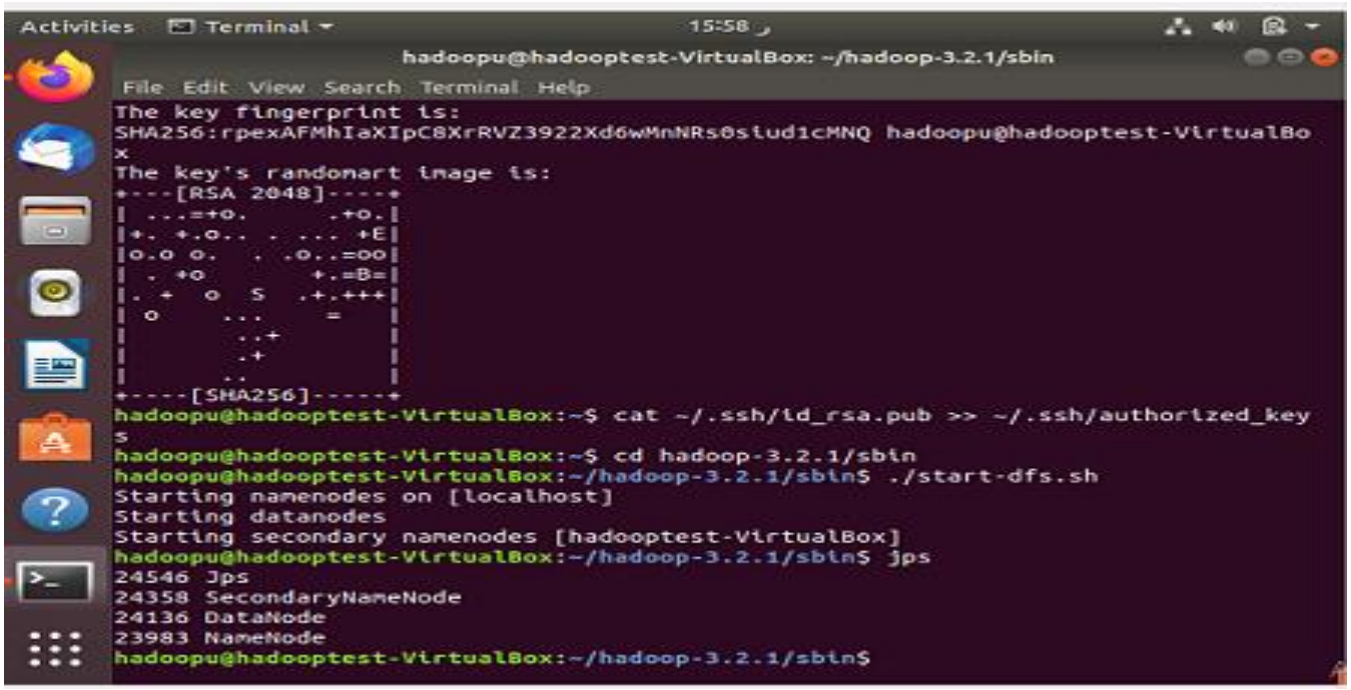
./start-yarn.sh As with the previous command, the output informs you that the processes are starting.

APACHE HADOOP INSTALLATION

-Type this simple command to check if all the daemons are active and running as Java processes:

Jps

If everything is working as intended, the resulting list of running Java processes contains all the HDFS and YARN daemons would appeared.



APACHE HADOOP INSTALLATION

Thanks