-- Install single node hadoop cluster (Hadoop\_ver2.7.0)

Pre-requisites:

1. Azure Linux VM (Ubuntu 18.04/ Ubuntu 20.04) image with at least 4 GB memory

// Install jdk

sudo apt-get install default-jdk

sudo apt install openjdk-11-jdk

// check java version

Java -version

// check the installed java path

Update-alternatives --config java

//ssh install

sudo apt-get install ssh

// append the generated public keys from id\_rsa.pub to authorized\_keys

ssh-keygen -t dsa

// Copy the keygen into the authorized\_keys folder

cat ~/.ssh/id\_dsa.pub >> ~/.ssh/authorized\_keys

// Download Apache Hadoop

wget <https://archive.apache.org/dist/hadoop/core/hadoop-2.7.0/hadoop-2.7.0.tar.gz>

//extract the folder

sudo tar -xvf Hadoop-2.7.0.tar.gz

// Move the folder

sudo mv Hadoop-2.7.0 /usr/local/hadoop

Hadoop version 2.7.0

// check localhost

ssh localhost

// Install the configs

sudo vim ~/.bashrc

Go to the end of the file of bashrc and apply the following config

export JAVA\_HOME=”<your\_java\_path>”

export HADOOP\_HOME=”/usr/local/Hadoop

export PATH = $PATH:$HADOOP\_HOME/bin

export PATH = $PATH:$HADOOP\_HOME/sbin

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 HADOOP\_OPTS=”-Djava.library.path=$HADOOP\_HOME/lib”

//Save and exit

source ~/.bashrc

Perform the following tasks

// Move and copy the mapred-site.xml.template to mapred-site.xml

cd $HADOOP\_HOME/etc/Hadoop

Sudo cp mapred-site.xml.template mapred-site.xml

<configuration>

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

</configuration>

// open yarn-site.xml

<configuration>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

//update core-site.xml

<property>

<name>fs.default.name</name>

<value>hdfs://127.0.0.1:9000</value>

</property>

</configuration>

//update hdfs-site.xml

<configuration>

<property>

<name>dfs.replication</name>

<value>1 </value>

</property>

<property>

<name>dfs.name.dir</name>

<value>file://home/ani/hadoop/hdfs/namenode</value>

</property>

<property>

<name>dfs.data.dir</name>

<value>file://home/ani/hadoop/hdfs/namenode/datanode</value>

</property>

</configuration>

// Open the Hadoop env shell & update java\_home

Export JAVA\_HOME=<your\_java\_path>

// create the local directory for namenode and datanode

sudo mkdir -p /home/adminuser/hadoop/hdfs/namenode

sudo mkdir -p /home/adminuser/hadoop/hdfs/datanode

// Give permission

sudo chown -R adminuser:adminuser /usr/local/Hadoop

sudo chmod 777 -R /usr/local/Hadoop

sudo chmod 777 -R /home/adminuser/hadoop/hdfs/namenode

sudo chmod 777 -R /home/adminuser/hadoop/hdfs/datanode

// format the namenode

Hdfs namenode -format

// start the Hadoop services

// go to $HADOOP\_HOME/sbin folder

cd $HADOOP\_HOME/sbin

// then start the Hadoop services

Start-dfs.sh

Start-yarn.sh

//Check out the status

jps

// access hadoop UI from browser

http://localhost:50070

http://localhost:8088 - YARN resource manager

// Make HDFS directories required to execute Mapreduce jobs

hdfs dfs -mkdir /user/Hadoop/input

hdfs dfs -put data.txt /user/Hadoop/input

// Apply wordcount sample

cd #HADOOP\_HOME

hadoop jar share/Hadoop/mapreduce/Hadoop-mapreduce-examples-2.7.0.jar wordcount /user/Hadoop/input /user/Hadoop/output

hdfs dfs -ls /user/Hadoop/output

//check the contents

hdfs dfs -cat /user/Hadoop/output/part-r-00000

Create jar file from java

Jar cf wc.jar input-files