Install JAva

sudo apt update

sudo apt install openjdk-11-jdk-headless

Download Hadoop

wget https://dlcdn.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz

tar -xvf hadoop-3.3.6.tar.gz

Add JAVA\_HOME

echo "export JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-amd64" >> .bashrc

Add Hadoop Home

echo "export HADOOP\_HOME=/home/vinodhtrainer/hadoop-3.3.6" >> .bashrc

note: replace /home/vinodhtrainer whith your path

add below four lines to .bashrc

export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_HOME=$HADOOP\_HOME

export HADOOP\_HDFS\_HOME=$HADOOP\_HOME

export YARN\_HOME=$HADOOP\_HOME

modify the environment variable path

echo "export PATH=$PATH:$HADOOP\_HOME/bin" >> .bashrc

Add Java Home to Hadoop-env.sh

echo "export JAVA\_HOME=$JAVA\_HOME" >> $HADOOP\_HOME/etc/hadoop/hadoop-env.sh

//add these lines within configuration elements in all files do not remove already existing commented lines

nano $HADOOP\_HOME/etc/hadoop/core-site.xml

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://0.0.0.0:9000</value>

</property>

</configuration>

nano $HADOOP\_HOME/etc/hadoop/hdfs-site.xml

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

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

<value>file://${hadoop.home.dir}/data/namenode</value>

</property>

<property>

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

<value>file://${hadoop.home.dir}/data/datanode</value>

</property>

</configuration>

nano $HADOOP\_HOME/etc/hadoop/hdfs-site.xml

<configuration>

<!-- Configuration for the ResourceManager -->

<property>

<name>yarn.resourcemanager.hostname</name>

<value>0.0.0.0</value>

</property>

<property>

<name>yarn.resourcemanager.address</name>

<value>0.0.0.0:8032</value>

</property>

<property>

<name>yarn.resourcemanager.scheduler.address</name>

<value>0.0.0.0:8030</value>

</property>

<property>

<name>yarn.resourcemanager.resource-tracker.address</name>

<value>0.0.0.0:8031</value>

</property>

<property>

<name>yarn.resourcemanager.admin.address</name>

<value>0.0.0.0:8033</value>

</property>

<property>

<name>yarn.resourcemanager.webapp.address</name>

<value>0.0.0.0:8088</value>

</property>

<property>

<name>yarn.resourcemanager.webapp.https.address</name>

<value>0.0.0.0:8090</value>

</property>

<!-- Configuration for the NodeManager -->

<property>

<name>yarn.nodemanager.address</name>

<value>0.0.0.0:45454</value>

</property>

<property>

<name>yarn.nodemanager.localizer.address</name>

<value>0.0.0.0:8040</value>

</property>

<property>

<name>yarn.nodemanager.webapp.address</name>

<value>0.0.0.0:8042</value>

</property>

</configuration>

Format name node

hdfs namenode -format

ssh-keygen -t rsa -P "" -f ~/.ssh/id\_rsa

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

chmod 600 ~/.ssh/id\_rsa

chmod 644 ~/.ssh/id\_rsa.pub

chmod 700 ~/.ssh

sudo chown -R vinodhtrainer:vinodhtrainer $HADOOP\_HOME

start-all.sh

check

<http://publicip:9870/>

in the browser

<http://publicip:8088/>

if working, Hadoop is ready

Spark Setup

wget <https://dlcdn.apache.org/spark/spark-3.5.1/spark-3.5.1-bin-hadoop3.tgz>

tar -xvf spark-3.5.1-bin-hadoop3.tgz

echo "export SPARK\_HOME=/home/vinodhtrainer/spark-3.5.1-bin-hadoop3" >> .bashrc

echo "PATH=$PATH:$SPARK\_HOME/bin" >> .bashrc

hdfs dfs -mkdir /spark

hdfs dfs -mkdir /spark/jars

hdfs dfs -put $SPARK\_HOME/jars/\* /sparks/jars/

hdfs dfs -put $SPARK\_HOME/jars/\* /spark/jars/

nano $SPARK\_HOME/conf/spark-defaults.conf

nano $SPARK\_HOME/conf/spark-defaults.conf

spark.master yarn

spark.submit.deployMode client

spark.executor.instances 2

spark.executor.memory 1G

spark.driver.memory 1G

spark.yarn.am.memory 1G

spark.yarn.jars hdfs:///spark/jars/\*

run

spark-shell

Note: if it works the spark setup is correct and run some

Code and check it behaviour

Upload the jar file from mac to linux using the following syntax

scp filepath/filename.jar username@publicip:/home/username

now run submit the jar file using following syntax

spark-submit --class SparkRDDReference --master yarn --deploy-mode cluster --executor-memory 2G --num-executors 3 spark-submittable-assembly-0.1.0-SNAPSHOT.jar

you can check this applications running in

<http://publicip:8088/>