Commands Used:

sudo apt install default-jdk

java -version

sudo apt install openssh-server openssh-client pdsh

sudo useradd -m -s /bin/bash hadoop

sudo passwd hadoop

sudo usermod -aG sudo hadoop

su - hadoop

ssh-keygen -t rsa

ls ~/.ssh/

cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

chmod 600 ~/.ssh/authorized_keys

ssh localhost

wget https://dlcdn.apache.org/hadoop/commo...

tar -xvzf hadoop-3.3.4.tar.gz

sudo mv hadoop-3.3.4 /usr/local/hadoop

sudo chown -R hadoop:hadoop /usr/local/hadoop

vi ~/.bashrc

Hadoop environment variables

export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64

export HADOOP_HOME=/usr/local/hadoop

export HADOOP_INSTALL=$HADOOP_HOME

export HADOOP_MAPRED_HOME=$HADOOP_HOME

export HADOOP_COMMON_HOME=$HADOOP_HOME

export HADOOP_HDFS_HOME=$HADOOP_HOME

export HADOOP_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"

source ~/.bashrc
echo $JAVA_HOME
echo $HADOOP_HOME
echo $HADOOP_OPTS

vi $HADOOP_HOME/etc/hadoop/hadoop-env.sh
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
hadoop version

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

configuration
   property
      name fs.defaultFS /name
      value hdfs://IP:9000 /value
   /property
/configuration

sudo mkdir -p /home/hadoop/hdfs/{namenode,datanode}
sudo chown -R hadoop:hadoop /home/hadoop/hdfs
sudo vi $HADOOP_HOME/etc/hadoop/hdfs-site.xml

configuration

   property
      name dfs.replication /name
```

```
      value 1 /value
   /property

  property
    name dfs.name.dir /name
    value file:///home/hadoop/hdfs/namenode /value
  /property

  property
    name dfs.data.dir /name
    value file:///home/hadoop/hdfs/datanode /value
  /property

/configuration

hdfs namenode -format
start-dfs.sh

//IF YOU HAVE AN ERROR
sudo apt-get remove pdsh
start-dfs.sh
IF IT DOESN'T HELP
ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
cat ~/.ssh/id_rsa.pub >>  ~/.ssh/authorized_keys

//YARN MANAGER
sudo vi  $HADOOP_HOME/etc/hadoop/mapred-site.xml
configuration
   property
```

name mapreduce.framework.name /name

value yarn /value

/property

property

name mapreduce.application.classpath /name

value
$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/lib/* /value

/property

/configuration

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

configuration

property

name yarn.nodemanager.aux-services /name

value mapreduce_shuffle /value

/property

property

name yarn.nodemanager.env-whitelist /name

value
JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PREPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_HOME,PATH,LANG,TZ,HADOOP_MAPRED_HOME /value

/property

/configuration

start-yarn.sh

//Copy Data from local to hdfs

hdfs dfs -put /home/hadoop/global_data_latest.csv ./

//Navigating to HDFS root directory

hdfs dfs -ls /

//Connecting spark to YARN

cd ~/spark-3.5.0-bin-hadoop3/bin/

./spark-shell --master yarn --queue default --name interaction

//Connect and read data from CSV file

val df = spark.read.format("csv").option("header",true).option("Separator",",").load("hdfs://localhost:9000/global_data_latest.csv")

//Display first 10 rows

df.show(10,false)

//Extracting data using pyspark(Python)

cd ~/spark-3.5.0-bin-hadoop3/bin/

./pyspark --master yarn --queue default --name interaction

//connect to CSV File

df = spark.read.format("csv").option("header",True).option("Separator",",").load("hdfs://localhost:9000/global_data_latest.csv")

//Show Schema

df.printSchema()

```
//Select Specific columns

df.select("Country","Age","Gender").show(20,false)


//Use filter function

import pyspark.sql.functions as f


//Display the first 10 males


df.filter(f.col("Gender")=="Male").select("Country","Age","Gender").show(20,false)


//Query data frame  using SQL Commands

df.createOrReplaceTempView("covid_data")

spark.sql("select sum(New_deaths) as new_death_count from covid_data where
Country_code='KE'").show()
```