Hadoop Multi-node cluster setup in GCP
Session-4

Step 1: Download java and Hadoop

1. Get into the namenode -1 instance
2. Switch as root user using the command
   ```
   sudo -i
   ```
3. Download java1.8 and Hadoop 2.7 by using the below command

   ```
   cd /opt/

   wget --no-cookies --no-check-certificate --header "Cookie: gpw_e24=http%3A%2F%2Fwww.oracle.com%2F; oraclelicense=accept-securebackup-cookie" http://download.oracle.com/otn-pub/java/jdk/8u131-b11/d54c1d3a095b4ff2b6607d096fa80163/jdk-8u131-linux-x64.tar.gz

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

Step2: Install Java

1. Extract java tar file and configure java using the below command

   ```
   tar xzf jdk-8u131-linux-x64.tar.gz

   cd /opt/jdk1.8.0_131

   alternatives --install /usr/bin/java java /opt/jdk1.8.0_131/bin/java 2

   alternatives --config java

   java -version
   ```

Step3: Configure Hadoop

1. Extract Hadoop tar file and update Hadoop configuration files using the below command

   ```
   cd /opt/
   tar xzf hadoop-2.7.0.tar.gz
   mv hadoop-2.7.0 /usr/local/hadoop
   chown -R hduser:hduser /usr/local/hadoop

   mkdir -p /usr/local/hadoop_store/tmp
   mkdir -p /usr/local/hadoop_store/hdfs/namenode
   mkdir -p /usr/local/hadoop_store/hdfs/datanode
   mkdir -p /usr/local/hadoop_store/hdfs/secondarynamenode
   chown -R hduser:hduser /usr/local/hadoop_store
   su hduser
   ```

2. Edit hdfs-site.xml file and add the given configuration

```
vi /usr/local/hadoop/etc/hadoop/hdfs-site.xml
```

```
 <configuration>
<property>
<name>dfs.replication</name>
<value>3</value>
<description>Default block replication.
The actual number of replications can be specified when the file is created.
The default is used if replication is not specified in create time.
</description>
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>file:/usr/local/hadoop_store/hdfs/namenode</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>file:/usr/local/hadoop_store/hdfs/datanode</value>
</property>
<property>
<name>dfs.namenode.checkpoint.dir</name>
<value>file:/usr/local/hadoop_store/hdfs/secondarynamenode</value>
</property>
<property>
<name>dfs.namenode.checkpoint.period</name>
<value>3600</value>
</property>
</configuration>
```

3. Edit core-site.xml file and add the given configuration

```
vi /usr/local/hadoop/etc/hadoop/core-site.xml
```

```
<configuration>
<property>
<name>hadoop.tmp.dir</name>
<value>/usr/local/hadoop_store/tmp</value>
<description>A base for other temporary directories.</description>
</property>
<property>
<name>fs.default.name</name>
<value>hdfs://namenode-1:54310</value>
<description>
The name of the default file system. A URI whose scheme and authority determine the FileSystem
implementation. The uri's scheme determines the config property fs.SCHEME.impl) naming the
FileSystem implementation class. The uri's authority is used to determine the host, port, etc. for a
filesystem.
</description>
</property>
</configuration>
```

4. Edit map-site.xml file and add the given configuration

```
vi /usr/local/hadoop/etc/hadoop/map-site.xml
```

```
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
<property>
  <name>mapred.job.tracker</name>
  <value>namenode-1:54311</value>
  <description>The host and port that the MapReduce job tracker runs
  at.  If "local", then jobs are run in-process as a single map
  and reduce task.
  </description>
</property>
</configuration>
```

5. Edit yarn-site.xml file and add the given configuration

```
vi /usr/local/hadoop/etc/hadoop/yarn-site.xml
```

```
<configuration>
<!-- Site specific YARN configuration properties -->
<property>
    <name>yarn.resourcemanager.hostname</name>
    <value>namenode-1</value>
</property>
<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>
</configuration>
```

6. Add java path in hadoop-env.sh file

```
echo 'export JAVA_HOME=/opt/jdk1.8.0_131' >> /usr/local/hadoop/etc/hadoop/hadoop-env.sh
```

Step4: Configure environment variable

1. To run Hadoop services, we need to add hadoop and java path in the .basrhc file

```
su hduser
vi /home/hduser/.bashrc

export HADOOP_PREFIX=/usr/local/hadoop
export HADOOP_HOME=/usr/local/hadoop
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_CONF_DIR=${HADOOP_HOME}/etc/hadoop
# Native Path
export HADOOP_COMMON_LIB_NATIVE_DIR=${HADOOP_PREFIX}/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_PREFIX/lib"
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
export JAVA_HOME=/opt/jdk1.8.0_131
export JRE_HOME=/opt/jdk1.8.0_131/jre
export PATH=$PATH:/opt/jdk1.8.0_131/bin:/opt/jdk1.8.0_131/jre/bin
```

Step5: Copy the Java tar file and Hadoop folder into another 2 instances

1. Copy java tar file into another 2 instances

```
su hduser
cd /opt/
scp jdk-8u131-linux-x64.tar.gz datanode-1:/home/hduser/
scp jdk-8u131-linux-x64.tar.gz datanode-2:/home/hduser/
```

2. Copy Hadoop folder into another 2 instances

```
cd /usr/local
scp -r hadoop datanode-1:/home/hduser/
scp -r hadoop_store datanode-1:/home/hduser/
scp -r /home/hduser/.bashrc datanode-1:/home/hduser/

scp -r /usr/local/hadoop datanode-2:/home/hduser/
scp -r /usr/local/hadoop_store datanode-2:/home/hduser/
scp -r /home/hduser/.bashrc datanode-2:/home/hduser/
```

Step6: Get into another 2 instances , Install Java and create folders

1. Extract java tar file and configure java using the below command

```
sudo -i

tar xzf jdk-8u131-linux-x64.tar.gz

cd /opt/jdk1.8.0_131

alternatives --install /usr/bin/java java /opt/jdk1.8.0_131/bin/java 2

alternatives --config java
```

```
java -version
```

2. Create and give permissions to the folders for Hadoop disk storage and processing

```
mv /home/hduser/hadoop /usr/local/
mv /home/hduser/hadoop_store /usr/local/

chown -R hduser:hduser /usr/local/hadoop
chown -R hduser:hduser /usr/local/hadoop_store
```

Note: Do step6 in both datanode-1 and datnode-2

Step7: Format namenode and start all Hadoop services

1. Get into the namenode instance and format namenode

```
su hduser
hadoop namenode -format
```

2. Start storage and yarn services

```
start-dfs.sh
start-yarn.sh
```