Program – 1

AIM: Perform setting up and install HADOOP in its two operating modes, Pseudo-distributed and fully distributed.

About HADOOP

The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.

Setting up HADOOP

Pre-requisites:

1. Java
2. SSH

Before any other steps, we need to set the java environment variable, this can be done in windows from the system variables window or on linux by adding the following to the variables file:

```
export JAVA_HOME=/usr/java/latest
```

Download and extract the HADOOP binaries.

```
wget http://apache.claz.org/hadoop/common/hadoop-3.1.2/
hadoop-3.1.2.tar.gz
tar xzf hadoop-3.1.2.tar.gz
hadoop-3.1.2/* to hadoop/
```

Pseudo-distributed mode

1. Add the following variables to the system variable file

```
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_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_INSTALL=$HADOOP_HOME
```

2. Configure HADOOP files
    a. Change to the Hadoop directory/etc/Hadoop
    b. Add the following to the ***hadoop-env.sh*** file

```
export JAVA_HOME=/usr/local/jdk1.7.0_71
```

    c. Edit the following config files

Program – 1

core-site.xml

```
<configuration>
   <property>
      <name>fs.default.name</name>
      <value>hdfs://localhost:9000</value>
   </property>
</configuration>
```

hdfs-site.xml

```
<configuration>
   <property>
      <name>dfs.replication</name>
      <value>1</value>
   </property>
   <property>
      <name>dfs.name.dir</name>
      <value>file:///home/<user_name>/hadoopinfra/hdfs/namenode
</value>
   </property>
   <property>
      <name>dfs.data.dir</name>
      <value>file:///home/<user_name>/hadoopinfra/hdfs/datanode
</value>
   </property>
</configuration>
```

yarn-site.xml

```
<configuration>
   <property>
      <name>yarn.nodemanager.aux-services</name>
      <value>mapreduce_shuffle</value>
   </property>
</configuration>
```

mapred-site.xml

```
<configuration>
   <property>
      <name>mapreduce.framework.name</name>
      <value>yarn</value>
   </property>
</configuration>
```

d. Verifying the installation
   i. Formatting the namenodes

Program – 1

```
Terminal  File  Edit  View  Search  Terminal  Help
2019-04-01 01:30:51,517 INFO util.GSet: VM type       = 64-bit
2019-04-01 01:30:51,517 INFO util.GSet: 0.25% max memory 3.9 GB = 10.0 MB
2019-04-01 01:30:51,517 INFO util.GSet: capacity      = 2^20 = 1048576 entries
2019-04-01 01:30:51,523 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.window.num.buckets = 10
2019-04-01 01:30:51,523 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.num.users = 10
2019-04-01 01:30:51,523 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.windows.minutes = 1,5,25
2019-04-01 01:30:51,526 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
2019-04-01 01:30:51,527 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry time is 600000 millis
2019-04-01 01:30:51,528 INFO util.GSet: Computing capacity for map NameNodeRetryCache
2019-04-01 01:30:51,528 INFO util.GSet: VM type       = 64-bit
2019-04-01 01:30:51,528 INFO util.GSet: 0.029999999329447746% max memory 3.9 GB = 1.2 MB
2019-04-01 01:30:51,528 INFO util.GSet: capacity      = 2^17 = 131072 entries
2019-04-01 01:30:51,544 INFO namenode.FSImage: Allocated new BlockPoolId: BP-1004355672-127.0.1.1-1554062451539
2019-04-01 01:30:51,717 INFO common.Storage: Storage directory /home/rinzler/hadoopinfra/hdfs/namenode has been successfully formatted.
2019-04-01 01:30:51,726 INFO namenode.FSImageFormatProtobuf: Saving image file /home/rinzler/hadoopinfra/hdfs/namenode/current/fsimage.ckpt_0000000000000000000
using no compression
2019-04-01 01:30:51,796 INFO namenode.FSImageFormatProtobuf: Image file /home/rinzler/hadoopinfra/hdfs/namenode/current/fsimage.ckpt_0000000000000000000 of size
 394 bytes saved in 0 seconds .
2019-04-01 01:30:51,847 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
2019-04-01 01:30:51,850 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at jarvis/127.0.1.1
************************************************************/
rinzler@jarvis:/usr/local/hadoop$
```

ii. Verifying the HDFS File system

```
Terminal  File  Edit  View  Search  Terminal  Help
rinzler@jarvis:/usr/local/hadoop$ start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [jarvis]
2019-04-01 01:52:05,401 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
rinzler@jarvis:/usr/local/hadoop$
```

iii. Starting YARN

```
Terminal  File  Edit  View  Search  Terminal  Help
rinzler@jarvis:/usr/local/hadoop$ start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [jarvis]
2019-04-01 01:52:05,401 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
rinzler@jarvis:/usr/local/hadoop$ start-yarn.sh
Starting resourcemanager
Starting nodemanagers
rinzler@jarvis:/usr/local/hadoop$
```

iv. Accessing the HADOOP bowser and verifying everything.

Hadoop | Overview | Datanodes | Datanode Volume Failures | Snapshot | Startup Progress | Utilities

Overview 'localhost:9000' (active)

Started:	Mon Apr 01 02:15:38 +0530 2019
Version:	3.1.2, r1019dde65bcf12e05ef48ac71e84550d589e5d9a
Compiled:	Tue Jan 29 07:09:00 +0530 2019 by sunilg from branch-3.1.2
Cluster ID:	CID-cbaacdc4-bd40-479f-948e-27a273a9086d
Block Pool ID:	BP-1674154990-127.0.1.1-1554064917150

Summary

Security is off.

Safemode is off.

7 files and directories, 0 blocks (0 replicated blocks, 0 erasure coded block groups) = 7 total filesystem object(s).

Heap Memory used 84.27 MB of 137 MB Heap Memory. Max Heap Memory is 3.9 GB.

Non Heap Memory used 47.25 MB of 50.94 MB Commited Non Heap Memory. Max Non Heap Memory is <unbounded>.

Program – 1

Fully distributed mode

1 Configure system and create host files on each node

   a. For each node, edit eh /etc/hosts/ file and add the IP addresses of the servers e.g.

```
192.0.2.1    node-master
192.0.2.2    node1
192.0.2.3    node2
```

2 Distribute the authentication key-pairs to the users

   a. Login to the node-master and generate ssh-keys

   b. Copy the keys tot the other nodes.

3 Download and extract the HADOOP binaries

4 Set the environment variables (same as pseudo-distributed)

5 Edit the core-site.xml file to set NameNode location

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
    <configuration>
        <property>
            <name>fs.default.name</name>
            <value>hdfs://node-master:9000</value>
        </property>
    </configuration>
```

6 Set the HDFS Paths in hdfs-site.xml

```
<configuration>
    <property>
            <name>dfs.namenode.name.dir</name>
            <value>/home/hadoop/data/nameNode</value>
    </property>

    <property>
            <name>dfs.datanode.data.dir</name>
            <value>/home/hadoop/data/dataNode</value>
    </property>

    <property>
            <name>dfs.replication</name>
            <value>1</value>
    </property>
</configuration>
```

7 Set the Job scheduler (same as pseudo-distributed)

8 Configure YARN in yarn-site.xml

```
<configuration>
    <property>
            <name>yarn.acl.enable</name>
            <value>0</value>
    </property>

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

Program – 1

```
 9                <value>node-master</value>
10        </property>
11
12        <property>
13                <name>yarn.nodemanager.aux-services</name>
14                <value>mapreduce_shuffle</value>
15        </property>
16  </configuration>
```

9 Duplicate the config files to each node.
10 Format the HDFS (same as pseudo-distributed).
11 Start the HDFS (same as pseudo-distributed).
12 Run YARN (same as pseudo-distributed).

Findings and Learnings:

1. We have installed HADOOP in both pseudo-distributed and fully-distributed modes.