Hadoop Development Setup on Ubuntu

Gopalakrishnan Subramani / 2020-10-10

Hadoop Ubuntu Setup?

This setup demonstrate easy development setup for Hadoop, should not be used for production environment. Production environment deployment shall be discussed later with clusters.

Requirements

1. Java 1.8 JDK/JRE
2. Apache Hadoop 2.7.3
3. Basic Linux Skills

Setup

Install Java 8 from OPEN JDK

open terminal and run below command.

Copy

```
sudo apt install openjdk-8-jdk -y
```

Set system path for JAVA_HOME and JRE_HOME

Copy

```
sudo nano /etc/environment
```

paste below content

Copy

```
JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
JRE_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

Download Hadoop Binaries

Copy

```
wget https://archive.apache.org/dist/hadoop/common/hadoop-2.7.7/hadoop-2.7.7.tar.gz

tar xf hadoop-2.7.7.tar.gz

sudo mv hadoop-2.7.7 /opt
```

Assign read/write/execute permission for complete hadoop folder. [not recommended for production]

Copy

```
sudo chmod 777 /opt/hadoop-2.7.7
```

Copy

```
sudo nano /etc/environment
```

paste below content

Copy

```
HADOOP_HOME=/opt/hadoop-2.7.7
```

```
HADOOP_INSTALL=/opt/hadoop-2.7.7
HADOOP_MAPRED_HOME=/opt/hadoop-2.7.7
HADOOP_COMMON_HOME=/opt/hadoop-2.7.7
HADOOP_HDFS_HOME=/opt/hadoop-2.7.7
YARN_HOME=/opt/hadoop-2.7.7
HADOOP_COMMON_LIB_NATIVE_DIR=/opt/hadoop-2.7.7/lib/native
```

Update profile (per user) environment.

no sudo here

Copy

```
nano ~/.profile
```

paste below to end of the file

Copy

```
export HADOOP_HOME=/opt/hadoop-2.7.7

export PATH=\$PATH:\$HADOOP_HOME/sbin:\$HADOOP_HOME/bin
```

Close existing terminal and open new terminal

Backup original hadoop config files, mapred-site.xml won't be there by default, ignore the error

move or copy..

Copy

```
mv $HADOOP_HOME/etc/hadoop/core-site.xml $HADOOP_HOME/etc/hadoop/core-site.xml.origina
mv $HADOOP_HOME/etc/hadoop/hdfs-site.xml $HADOOP_HOME/etc/hadoop/hdfs-site.xml.origina
mv $HADOOP_HOME/etc/hadoop/mapred-site.xml $HADOOP_HOME/etc/hadoop/mapred-site.xml.ori
mv $HADOOP_HOME/etc/hadoop/yarn-site.xml $HADOOP_HOME/etc/hadoop/yarn-site.xml.origina
```

Truncate the existing content in the conf file [for easy editing, optional]

Copy

```
cat /dev/null > $HADOOP_HOME/etc/hadoop/core-site.xml
cat /dev/null > $HADOOP_HOME/etc/hadoop/hdfs-site.xml
cat /dev/null > $HADOOP_HOME/etc/hadoop/mapred-site.xml
cat /dev/null > $HADOOP_HOME/etc/hadoop/yarn-site.xml
```

edit the config files

Copy

```
sudo nano $HADOOP_HOME/etc/hadoop/core-site.xml
```

Replace with below content

https://raw.githubusercontent.com/nodesense/kafka-workshop/master/hadoop/core-site.xml

or

Copy

```
<configuration>

<property>
<name>fs.defaultFS</name>
<value>hdfs://localhost:9000</value>

</property>

<property>
<name>hadoop.tmp.dir</name>
<value>/data/hdfs</value>
</property>

  <property>
    <name>hadoop.proxyuser.hive.hosts</name>
    <value>*</value>
</property>
```

```
<property>
    <name>hadoop.proxyuser.hive.groups</name>
    <value>*</value>
</property>
</configuration>
```

Copy

```
sudo nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml
```

Replace with below content

https://raw.githubusercontent.com/nodesense/kafka-workshop/master/hadoop/hdfs-site.xml

or

Copy

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

    <property>
<name>dfs.datanode.use.datanode.hostname</name>
<value>true</value>
</property>
</configuration>
```

Copy

```
sudo nano $HADOOP_HOME/etc/hadoop/mapred-site.xml
```

Replace with below content

https://raw.githubusercontent.com/nodesense/kafka-workshop/master/hadoop/mapred-site.xml

or

Copy

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

Copy

```
sudo nano $HADOOP_HOME/etc/hadoop/yarn-site.xml
```

Replace with below content

https://raw.githubusercontent.com/nodesense/kafka-workshop/master/hadoop/yarn-site.xml

Copy

```
<configuration>
    <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>

    <property>
    <name>yarn.log-aggregation-enable</name>
    <value>true</value>
    </property>

</configuration>
```

Restart the linux system once

For running hadoop cluster with ssh, as hadoop user needs ssh permission.

Install ssh server if not there

Check status

Copy

```
sudo systemctl status ssh
```

if not installed,

Copy

```
sudo apt install openssh-server
sudo systemctl enable ssh
sudo systemctl start ssh
```

if firewall installed,

Copy

```
sudo ufw allow ssh
```

Check again

Copy

```
sudo systemctl status ssh
```

Test if all ok,

Copy

```
ssh username@ip_address
```

Copy

```
ssh-keyscan localhost,0.0.0.0 > ~/.ssh/known_hosts
chmod +x $HADOOP_HOME/sbin/start-all.sh
```

Prepare data directory for HDFS

Copy

```
sudo mkdir -p /data/hdfs
sudo chmod 777 /data/hdfs
```

format the namenode

Copy

```
hdfs namenode -format
```

Then start all services

Copy

```
start-all.sh
```

Open browser and check all is well..

Copy

```
http://hostname:50070

or

http://localhost:50070

http://localhost:50070/explorer.html#/


yarn


http://localhost:8088/cluster
```

Default port references **https://kontext.tech/column/hadoop/265/default-ports-used-by-hadoop-services-hdfs-mapreduce-yarn**

Examples

To run sample pi application

Number of Maps = 4 Samples per Map = 4

Copy

```
hadoop jar $HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.7.jar pi
```

To list out all the application

Copy

```
yarn application -list -appStates ALL
```

works only when HA enabled, we will discuss in later sessions

Copy

```
yarn rmadmin -checkHealth
```

To get application ID use yarn application -list

Copy

```
yarn application -status application_XXXYYYZZZKKKK_0002
```

To view logs of application,

Copy

```
yarn logs -applicationId application_XXXYYYZZZKKKK_0002
```

Copy

```
yarn application -kill application_XXXYYYZZZKKKK_0002
```

Copy

```
yarn application -list

yarn application -list -appStates FINISHED
yarn application -list -appStates ALL
```

more **https://docs.cloudera.com/runtime/7.0.0/yarn-monitoring-clusters-applications/topics/yarn-use-cli-view-logs-applications.html**