

How to install hadoop 2.7.3 single node cluster on ubuntu 16.04

Before installing or downloading anything, It is always better to update using following command:

```
$ sudo apt-get update
```

Step 1: Install oracle Java

Step 2: Add dedicated hadoop user

```
$ sudo addgroup hadoop
$ sudo adduser --ingroup hadoop hduser
```

NOTE: don't write password or any things here, Just press 'y' when it ask "Is the information correct?[Y|n]"

```
$ sudo adduser hduser sudo
```

Step 3: Install SSH

```
$ sudo apt-get install ssh
```

Step-4: Passwordless entry for localhost using SSH

```
$ su hduser
```

Now we are logged in in 'hduser'.

```
$ ssh-keygen -t rsa
NOTE: Leave file name and other things blank.
$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
$ chmod 0600 ~/.ssh/authorized_keys
$ ssh localhost
```

```
$ exit
```

Step 5: Install hadoop-2.7.3

```
$ wget http://www-us.apache.org/dist/hadoop/common/hadoop-2.7.3/hadoop-2.7.3.tar.gz
$ tar xvfz hadoop-2.7.3.tar.gz
$ sudo mkdir -p /usr/local/hadoop
$ cd hadoop-2.7.3/
$ sudo mv * /usr/local/hadoop
$ sudo chown -R hduser:hadoop /usr/local/hadoop
```

Step 6: Setup Configuration Files

The following files should to be modified to complete the Hadoop setup:

6.1 ~/.bashrc

6.2 hadoop-env.sh

6.3 core-site.xml

6.4 mapred-site.xml

6.5 hdfs-site.xml

6.6 yarn-site.xml

6.1 ~/.bashrc

First, we need to find the path where JAVA is installed in our system

```
$ update-alternatives --config java
```

It may possible vi will not work properly. If it does install vim

```
$ sudo apt-get install vim
```

Open bashrc file using command:

```
$ vim ~/.bashrc
```

Append following at the end. (Follow this process -> First append below content at the end by pressing 'INSERT' or 'i' key from keyboard-> Press 'esc' -> Press ':' (colon) -> Press 'wq' -> Press 'Enter' Key)

```
#HADOOP VARIABLES START
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
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=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
#HADOOP VARIABLES END
```

Update .bashrc file to apply changes

```
$ source ~/.bashrc
```

6.2 hadoop-env.sh

We need to modify JAVA_HOME path in hadoop-env.sh to ensure that the value of JAVA_HOME variable will be available to Hadoop whenever it is started up.

```
$ vim /usr/local/hadoop/etc/hadoop/hadoop-env.sh
```

Search JAVA_HOME variable in file. It may first variable in file. Do Change it by following:

```
export JAVA_HOME=/usr/lib/jvm/java-8-oracle
```

6.3 core-site.xml

core-site.xml file has configuration properties which are requires when Hadoop is started up.

```
$ sudo mkdir -p /app/hadoop/tmp
```

```
$ sudo chown hduser:hadoop /app/hadoop/tmp
```

Open the file and enter the following in between the <configuration></configuration> tag:

```
$ vim /usr/local/hadoop/etc/hadoop/core-site.xml
```

```
<configuration>
<property>
  <name>hadoop.tmp.dir</name>
  <value>/app/hadoop/tmp</value>
  <description>A base for other temporary directories.</description>
</property>
<property>
  <name>fs.default.name</name>
  <value>hdfs://localhost:54310</value>
</property>
</configuration>
```

6.4 mapred-site.xml

By default, the /usr/local/hadoop/etc/hadoop/ folder contains /usr/local/hadoop/etc/hadoop/mapred-site.xml.template file which has to be renamed/copied with the name mapred-site.xml:

```
$ cp /usr/local/hadoop/etc/hadoop/mapred-site.xml.template
/usr/local/hadoop/etc/hadoop/mapred-site.xml
```

The /usr/local/hadoop/etc/hadoop/mapred-site.xml file is used to specify which framework is being used for MapReduce.

We need to enter the following content in between the <configuration></configuration> tag:

```
$ vim /usr/local/hadoop/etc/hadoop/mapred-site.xml
```

```
<configuration>
<property>
  <name>mapred.job.tracker</name>
  <value>localhost:54311</value>
</property>
<property>
<name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>

</configuration>
```

6.5 hdfs-site.xml

We need to configure hdfs-site.xml for each host in the cluster which specifies two directories:

1. Name node
2. Data node

These can be done using the following commands:

```
$ sudo mkdir -p /usr/local/hadoop_store/hdfs/namenode
$ sudo mkdir -p /usr/local/hadoop_store/hdfs/datanode
$ sudo chown -R hduser:hadoop /usr/local/hadoop_store
```

Open hdfs-site.xml file and enter the following content in between the <configuration></configuration> tag:

```
$ vim /usr/local/hadoop/etc/hadoop/hdfs-site.xml
```

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

6.6 yarn-site.xml

Open hdfs-site.xml file and enter the following content in between the <configuration></configuration> tag:

```
$ vim /usr/local/hadoop/etc/hadoop/yarn-site.xml
```

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

Step7: Format hadoop file system

```
$ hadoop namenode -format
```

Step 8: Start Hadoop Daemons

```
$ cd /usr/local/hadoop/sbin
```

```
$ start-all.sh
```

```
$ jps
```

MAKE SURE NAMENODE IS RUNNING

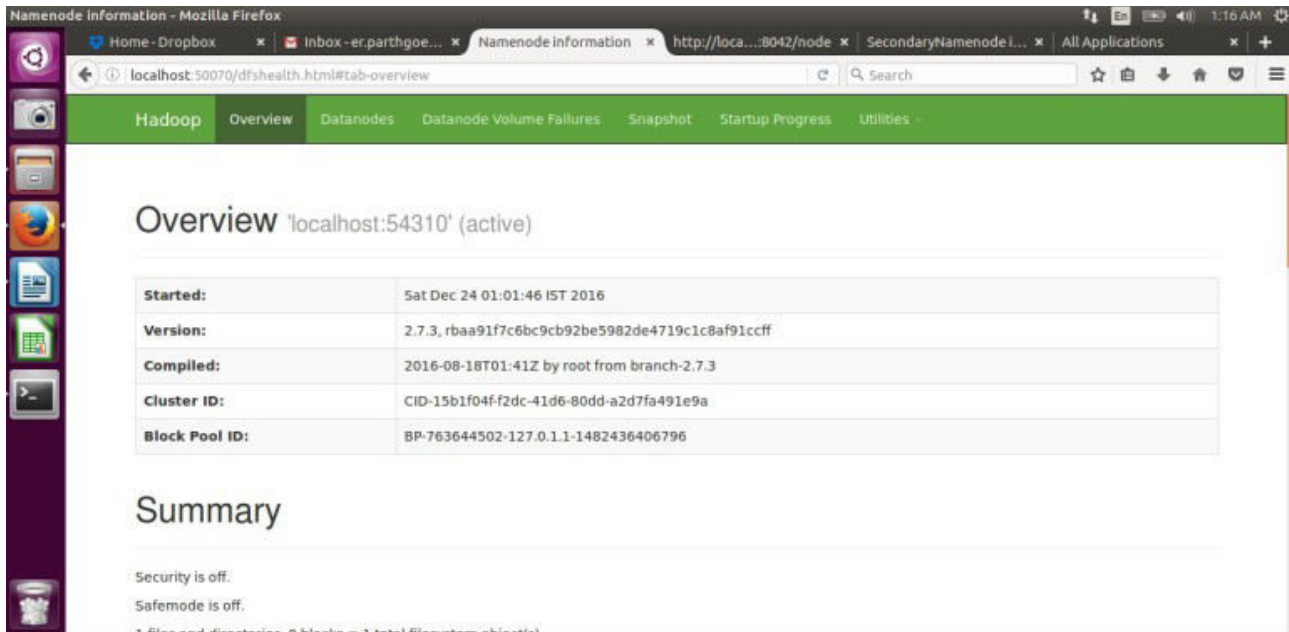
Step 9: Stop hadoop Daemons

\$ stop-all.sh

Congratulation...!! We have installed hadoop successfully..

Hadoop has Web Interfaces too. (Copy and paste following links in your browser)

NameNode daemon: <http://localhost:50070/>

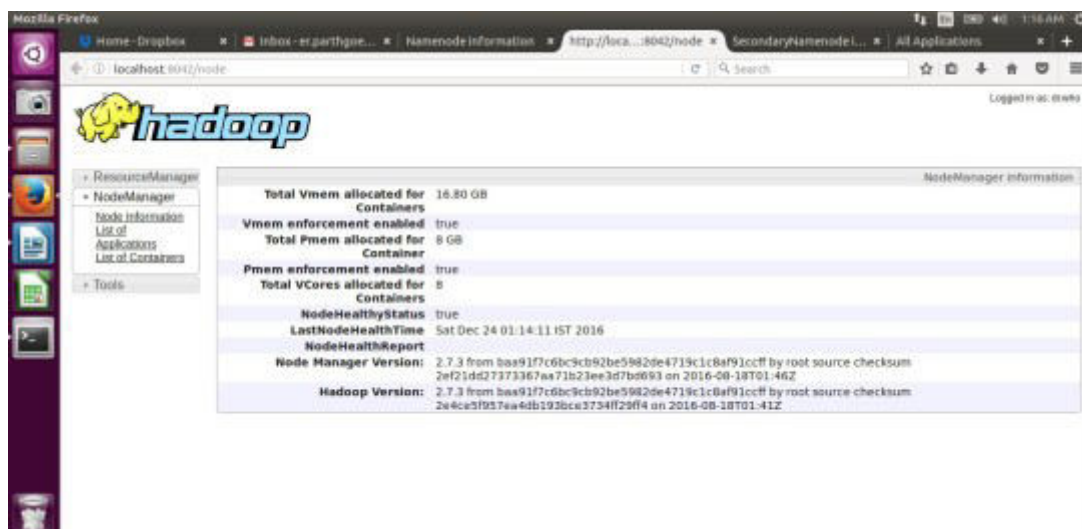


Property	Value
Started:	Sat Dec 24 01:01:46 IST 2016
Version:	2.7.3, rbaa91f7c6bc9cb92be5982de4719c1c8af91ccff
Compiled:	2016-08-18T01:41Z by root from branch-2.7.3
Cluster ID:	CID-15b1f04f-f2dc-41d6-80dd-a2d7fa491e9a
Block Pool ID:	BP-763644502-127.0.1.1-1482436406796

Summary

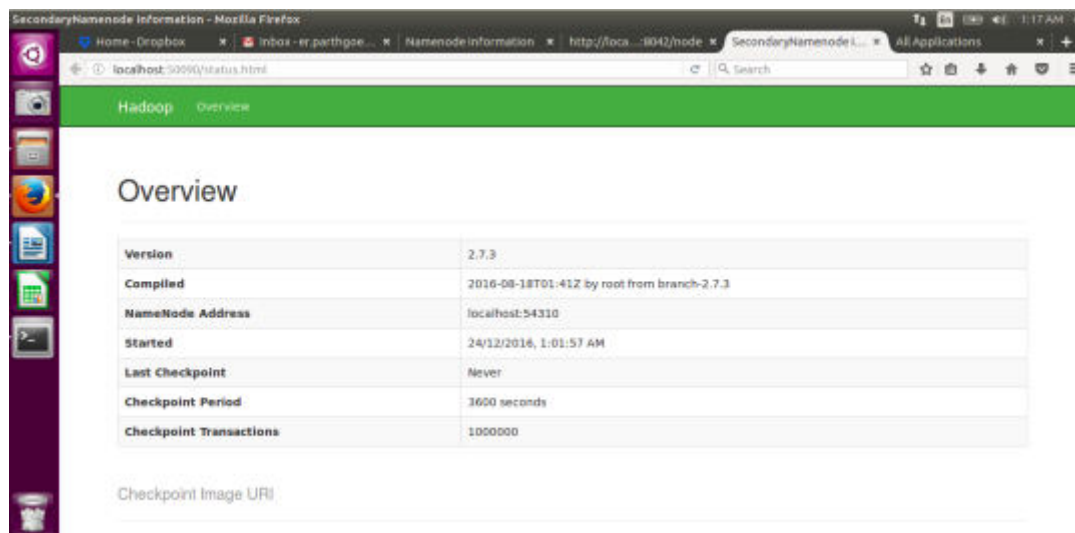
Security is off.
Safemode is off.
1 files and directories, 0 blocks = 1 total filesystem object(s)

mapreduce: <http://localhost:8042/>



Property	Value
Total Vmem allocated for Containers	16.80 GB
Vmem enforcement enabled	true
Total Pmem allocated for Container	8 GB
Pmem enforcement enabled	true
Total Vcores allocated for Containers	8
NodeHealthyStatus	true
LastNodeHealthTime	Sat Dec 24 01:14:11 IST 2016
NodeHealthReport	2.7.3 from baa91f7c6bc9cb92be5982de4719c1c8af91ccff by root source checksum 2ef21d427373367aa71b23ee3d7b0693 on 2016-08-18T01:46Z
Node Manager Version	2.7.3 from baa91f7c6bc9cb92be5982de4719c1c8af91ccff by root source checksum 2e4ca3f957aa4db193bca3734ff29ff4 on 2016-08-18T01:41Z
Hadoop Version	2.7.3 from baa91f7c6bc9cb92be5982de4719c1c8af91ccff by root source checksum 2e4ca3f957aa4db193bca3734ff29ff4 on 2016-08-18T01:41Z

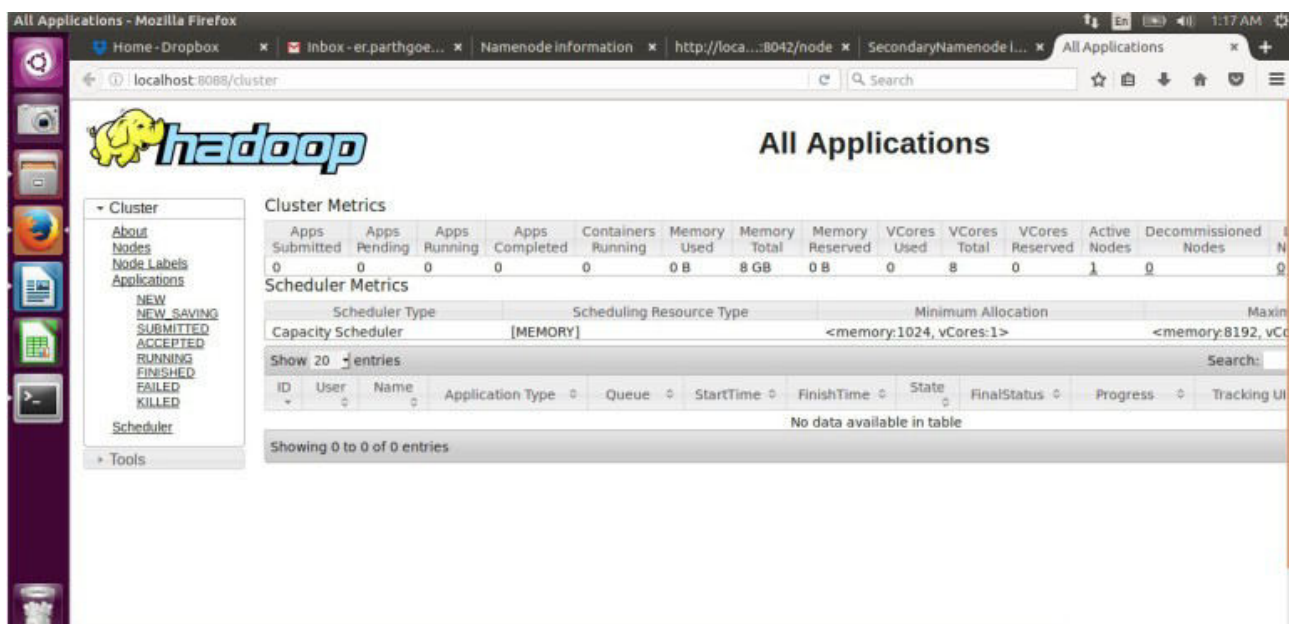
SecondaryNameNode:: <http://localhost:50090/status.html>



Metric	Value
Version	2.7.3
Compiled	2016-08-18T01:41Z by root from branch-2.7.3
NameNode Address	localhost:54310
Started	24/12/2016, 1:01:57 AM
Last Checkpoint	Never
Checkpoint Period	3600 seconds
Checkpoint Transactions	1000000

Checkpoint Image URI

Resource Manager: <http://localhost:8088/>



Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes
0	0	0	0	0	0 B	8 GB	0 B	0	8	0	1	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:8192, vCores:1>

Show 20 entries

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI
No data available in table										

Showing 0 to 0 of 0 entries

Now, we run mapreduce job on our newly created hadoop single node cluster setup.

```
hduser@parthgoel:/usr/local/hadoop$ hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.3.jar pi 2 5
```

```
hduser@parthgoel: /usr/local/hadoop
Reduce output records=0
Spilled Records=8
Shuffled Maps =2
Failed Shuffles=0
Merged Map outputs=2
GC time elapsed (ms)=892
CPU time spent (ms)=2260
Physical memory (bytes) snapshot=480915456
Virtual memory (bytes) snapshot=5702144000
Total committed heap usage (bytes)=307437568

Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0

File Input Format Counters
Bytes Read=236
File Output Format Counters
Bytes Written=97
Job Finished in 87.536 seconds
Estimated value of Pi is 3.60000000000000000000000000000000
hduser@parthgoel: /usr/local/hadoop$
```

NOTE: Whenever we login in Ubuntu, make sure you are in 'hduser'.

If you are not in hduser, use below command to login in 'hduser'

\$ su hduser

Hbase-1.2.5 Installation With Hadoop-2.7.3 In Ubuntu 16.04

for hadoop-2.7.3 ,hbase-1.2.x will support ,not less than this

1. HBASE Installation In Ubuntu

0) Running HDFS

\$ cd \$HADOOP_HOME

\$ bin/hdfs namenode -format

\$ sbin/start-dfs.sh

1) Download hbase-1.2.5.tar.gz In Apache Hbase (binary file)

<http://apache.mirror.cdnetworks.com/hbase/1.2.5/hbase-1.2.5-bin.tar.gz>

2) Make folder, Move and Extract tar file

\$ sudo mkdir /usr/local/hbase

\$ sudo tar -zxvf hbase-1.2.5-bin.tar.gz

\$ sudo mv apache-hbase-1.2.5-bin /usr/local/hbase

3) Set Environment Variables

```
$ cd ~
```

```
$ vi .bashrc
```

[.bashrc] : Below the Last Line

```
export HBASE_HOME=/usr/local/hbase/hbase-1.2.5
export PATH=$PATH:$HBASE_HOME/bin
export CLASSPATH=$CLASSPATH:$HBASE_HOME/lib/*:.
```

```
$ source ~/.bashrc
```

4) Configuration Check

4-1) Edit the hbase-env.sh about JAVA_HOME

```
$ cd $HBASE_HOME/conf
```

```
$ sudo vi hbase-env.sh
```

```
$ sudo chmod -R 777 /usr/local/hbase/
```

[hbase-env.sh]

```
export JAVA_HOME= /usr/local/java/jdk1.7.0_80/jre
```

```
export HBASE_MANAGES_ZK=true
```

4-2) Editing Configuration and Making Temporary Directories For Hbase Temp File

```
$ sudo mkdir hbaseinfra
```

```
$ sudo vi hbase-site.xml
```

[hbase-site.xml]

```
<configuration>
  <property>
    <name>hbase.rootdir</name>
    <value>hdfs://129.168.1.129:8030/hbase/hfile</value>
<!-- <value>/usr/local/hbase/hbaseinfra/hfile</value> -->
  </property>

  <property>
    <name>hbase.zookeeper.property.dataDir</name>
    <value>file:///usr/local/hbase/hbaseinfra/zookeeper</value>
  </property>
</configuration>
```

2. Running HBASE

```
$ cd $HBASE_HOME/bin
```

```
$ start-hbase.sh
```

```
$ hbase shell
```

```
> create 'hbase_test_table','id data'
```

```
> list
```

Submitted by:

Priyanak Gupta