BIG DATA ANALYTICS Name: Lovedeep Singh SID: 16103104

Hadoop Installation on Mac OS X

Wed, Aug 21, 2019

Install Homebrew and Cask

```
$ ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
$ brew install caskroom/cask/brew-cask
```

Install Java

```
$ brew update
$ brew cask install java
```

Configure SSH

In order to keep the safety of Hadoop remote administration as well as user sharing among Hadoop nodes, Hadoop requires SSH protocol. First, go to `System Preferences -> Sharing`, change `Allow access for`: **All Users**. Then open Terminal, input `ssh localhost`, if terminal returns `Last login: Sun Jul 2 16:57:36 2017`, which means that you have configured SSH Keys successfully before.

If you suffer the problem of `ssh: connect to host localhost port 22: Connection refused`, it happens since the remote login is closed.

```
$ sudo systemsetup -getremotelogin
Remote Login: off
```

You need to open port 22 in Mac OS X:

To verify if SSH Localhost is working check for files ~/.ssh/id_rsa and the ~/.ssh/id_rsa.pub files. If they don’t exist generate the keys using below command

```
$ ssh-keygen -t rsa
```

Enable Remote Login: “System Preferences” -> “Sharing”. Check “Remote Login” Authorize SSH Keys: To allow your system to accept login, we have to make it aware of the keys that will be used

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

Test login.

```
$ ssh localhost
Last login: Fri Mar 6 20:30:53 2015
$ exit
```

Install Hadoop

First, install Hadoop via Homebrew: `brew install hadoop`, it will install the hadoop under `/usr/local/Cellar/hadoop`. Then, you need to modify the configuration files.

Go to `usr/local/Cellar/hadoop/2.8.0/libexec/etc/hadoop`, then open `hadoop-env.sh`

export HADOOP_OPTS="$HADOOP_OPTS -Djava.net.preferIPv4Stack=true” change to

```
export HADOOP_OPTS="$HADOOP_OPTS -
Djava.net.preferIPv4Stack=true -Djava.security.krb5.realm= -
Djava.security.krb5.kdc="
export JAVA_HOME="/Library/Java/JavaVirtualMachines/
jdk1.7.0_79.jdk/Contents/Home"
```

Then configure HDFS address and port number, open `core-site.xml`, input following content in `<configuration></configuration>` tag

```
<!-- Put site-specific property overrides in this file. -->
 <configuration>
    <property>
        <name>hadoop.tmp.dir</name>
        <value>/usr/local/Cellar/hadoop/hdfs/tmp</value>
        <description>A base for other temporary 
directories.</description>
    </property>
    <property>
        <name>fs.default.name</name>
        <value>hdfs://localhost:8020</value>
    </property>
</configuration>
```

Configure `jobtracker` address and port number in map-reduce, first `sudo cp mapred-site.xml.template mapred-site.xml` to make a copy of `mapred-site.xml`, and open `mapred-site.xml`, add

```
<configuration>
    <property>
        <name>mapred.job.tracker</name>
        <value>localhost:8021</value>
    </property>
</configuration>
```

Set HDFS default backup, the default value is 3, we should change to 1, open `hdfs-site.xml`, add

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

Before running background program, we should format the installed HDFS first, executing command `hdfs namenode -format`, when terminal returns a long inforamtion like:

```
17/07/02 16:11:05 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
......
17/07/02 16:11:07 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at haodemacbook-
pro.local/192.168.1.4
************************************************************/
```

It means that we finish HDFS configuration, and Hadoop is ready to launch. Besides, maybe you will get a warning

```
$ ... WARN util.NativeCodeLoader: Unable to load native-
hadoop library for your platform... using builtin-java
classes where applicable
```

It happens since you are running on 64-bit system but Hadoop native library is based on 32-bit. This is not a big issue. If it appears, you can fixed by refering this link: here.

Launch Hadoop

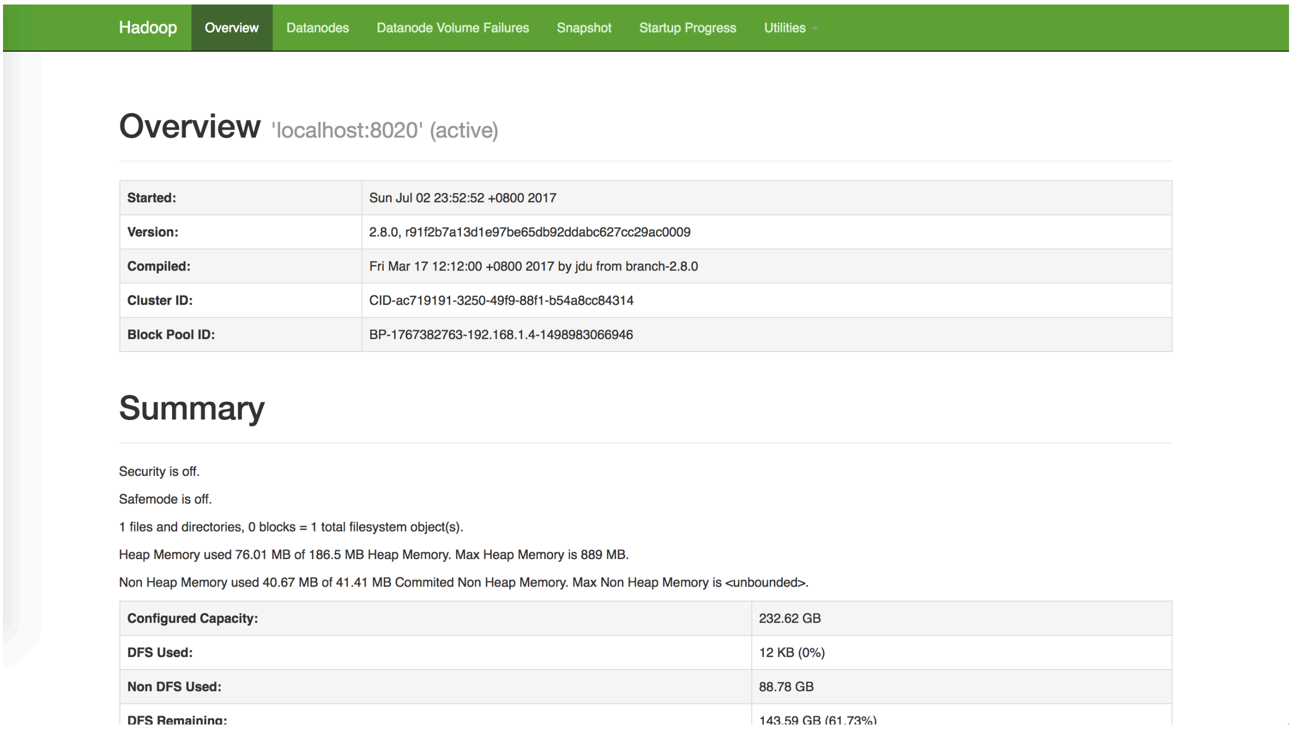
Go to `/usr/local/Cellar/hadoop/2.8.0/sbin`, execute:

```
$ ./start-dfs.sh # start HDFS service
$ ./stop-dfs.sh # stop HDFS service
```

Ternimal will return the following information:

```
Starting namenodes on [localhost]
localhost: starting namenode, logging to /usr/local/Cellar/
hadoop/2.8.0/libexec/logs/hadoop-zhanghao-namenode-
HaodeMacBook-Pro.local.out
localhost: starting datanode, logging to /usr/local/Cellar/
hadoop/2.8.0/libexec/logs/hadoop-zhanghao-datanode-
HaodeMacBook-Pro.local.out
Starting secondary namenodes [0.0.0.0]
```

It means the local service launched successfully, then open Resource Manager in browser through the link `http://localhost:9870`, you can see the following page



Overview 'localhost:8020' (active)

Started:	Sun Jul 02 23:52:52 +0800 2017
Version:	2.8.0, r91f2b7a13d1e97be65db92ddabc627cc29ac0009
Compiled:	Fri Mar 17 12:12:00 +0800 2017 by jdu from branch-2.8.0
Cluster ID:	CID-ac719191-3250-49f9-88f1-b54a8cc84314
Block Pool ID:	BP-1767382763-192.168.1.4-1498983066946

Summary

Security is off.

Safemode is off.

1 files and directories, 0 blocks = 1 total filesystem object(s).

Heap Memory used 76.01 MB of 186.5 MB Heap Memory. Max Heap Memory is 889 MB.

Non Heap Memory used 40.67 MB of 41.41 MB Commited Non Heap Memory. Max Non Heap Memory is <unbounded>.

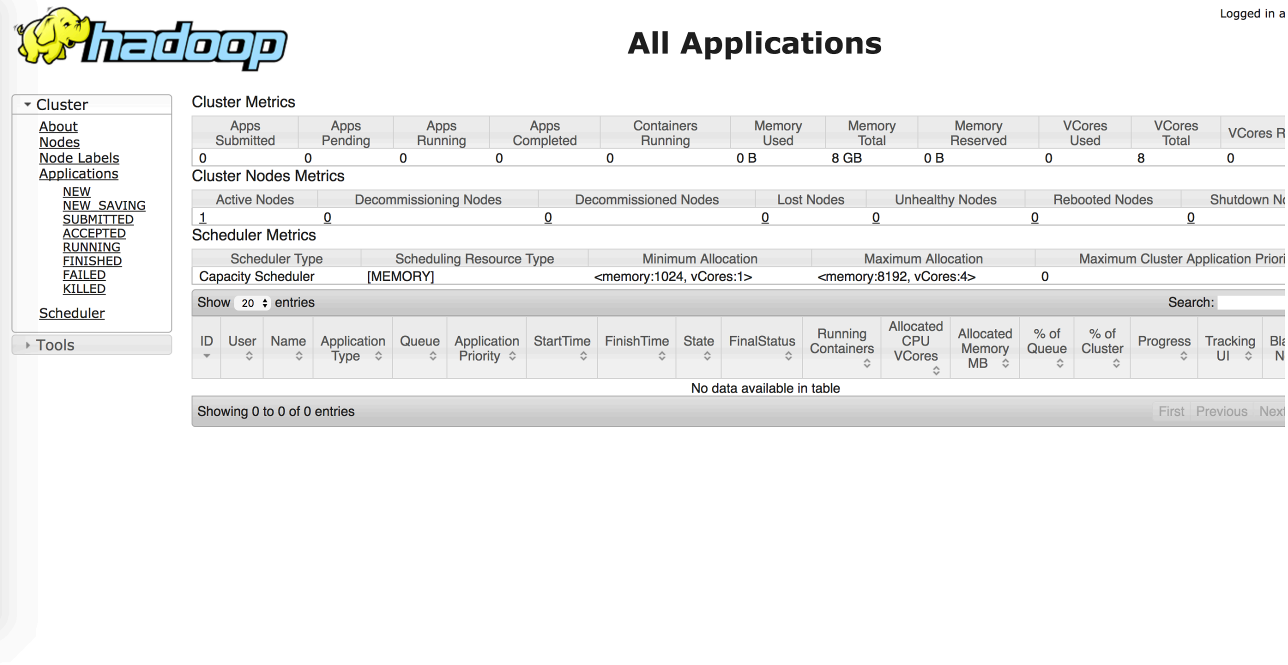
Configured Capacity:	232.62 GB
DFS Used:	12 KB (0%)
Non DFS Used:	88.78 GB
DFS Remaining:	143.59 GB (61.73%)

Note: earlier Hadoop versions of 2.x.x had 50070, after 3.0.0 it was changed to 9870 port number

Samely, under current diretory, you can start the JobTracker through the commands:

```
$ ./start-yarn.sh # start yarn, MapReduce framework
$ ./stop-yarn.sh # stop yarn
```

Then open browser and go to the page `http://localhost:8088`, Specific Node Information `http://localhost:8042`, you will see



All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores R
0	0	0	0	0	0 B	8 GB	0 B	0	8	0

Cluster Nodes Metrics

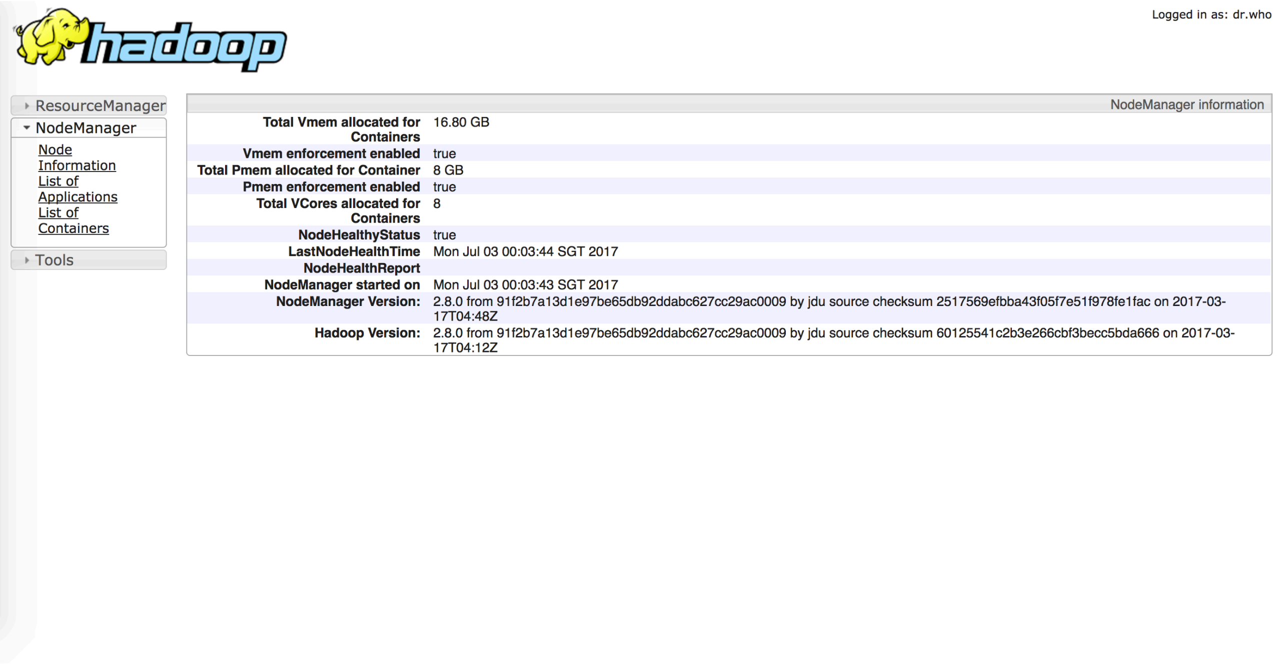
Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	Shutdown No
1	0	0	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priori
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0

ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	% of Queue	% of Cluster	Progress	Tracking UI	Bla N
No data available in table																	

Showing 0 to 0 of 0 entries



NodeManager information	
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	Mon Jul 03 00:03:44 SGT 2017
NodeHealthReport	
NodeManager started on	Mon Jul 03 00:03:43 SGT 2017
NodeManager Version:	2.8.0 from 91f2b7a13d1e97be65db92ddabc627cc29ac0009 by jdu source checksum 2517569efbba43f05f7e51f978fe1fac on 2017-03-17T04:48Z
Hadoop Version:	2.8.0 from 91f2b7a13d1e97be65db92ddabc627cc29ac0009 by jdu source checksum 60125541c2b3e266cbf3becc5bda666 on 2017-03-17T04:12Z

Simply, you can execute `./start-all.sh` and `./stop-all.sh` to start or close all the hadoop service. Finally, open `/etc/`

`profile` and add the configuration information of Hadoop environment variables.

```
export HADOOP_HOME=/usr/local/Cellar/hadoop/2.8.0
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
```

Then you can start and close Hadoop under the user directory rather than go to `/usr/local/Cellar/hadoop/2.8.0/sbin` every time.

Note: Since 3.0.0 version, lot of port numbers changed. We have:

In fact, lots of others ports changed too. Look:

```
Namenode ports: 50470 --> 9871, 50070 --> 9870, 8020 --> 9820
Secondary NN ports: 50091 --> 9869, 50090 --> 9868
Datanode ports: 50020 --> 9867, 50010 --> 9866, 50475 -->
9865, 50075 --> 9864
```