Hadoop Distributed File System (HDFS) + Spark Integration Guidance

Please follow the steps below exactly and read the explanations carefully

Note:

All the commands should be entered and run in Ubuntu Terminal (In Fedora commands may vary) if no special reminders. Assume we have three machines which are in the same local network: worker1 (IP: 192.168.0.1), worker2 (IP: 192.168.0.2) and worker3 (IP: 192.168.0.3). Among them, worker1 is the master, worker2 and worker3 are workers/slaves.

Prerequisite:

1. Amazon EC2 Machines:

Each group should have at least 2 Amazon EC2 Ubuntu Instances. You can try amazon free tier here for one year: https://aws.amazon.com/free/.

When you create your instances, at step 6, make sure you add one more rule wit Type: All traffic, source: anywhere. Without doing this, you will not be able to check status of the cluster in the web interface.

After you create your instances with the required security group setting, make sure you can access the machines.

If you are using Linux system, use ssh –i pem_filepath ubuntu@public_ip to log into the EC2 instances.

If you are using Windows system, use putty to access the EC2 by following the instruction here: https://dzone.com/articles/how-set-multi-node-hadoop.

2. java installation and JAVA_HOME environment variable setup

Use "sudo apt-get install default-jre".

Check value of JAVA_HOME. If it is empty, you need to setup it.
https://askubuntu.com/questions/175514/how-to-set-java-home-for-java

1. Password-less SSH login (Required by both of Hadoop and Spark)

1) Generate the public key and private key (On worker1 and worker2)

```
ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
```

2) Append contents of the file `id_rsa.pub` of worker2 and worker1 to the file authorized_keys in both worker2 and worker1 (master).

3) On worker2, test you can log in master and worker 2 with no password by using both public ip and private ip of the master instance.

```
ssh 192.168.0.1
ssh 192.168.1.2
```

4) On master, test that we can log in master and worker2 with no password by using the public and private ip

```
ssh 192.168.0.1
ssh 192.168.0.2
```

5) After the steps above, we have a bi-direction password-less SSH login from worker2 to master (worker1). Please implement the password-less SSH among your cluster based on the following topology.

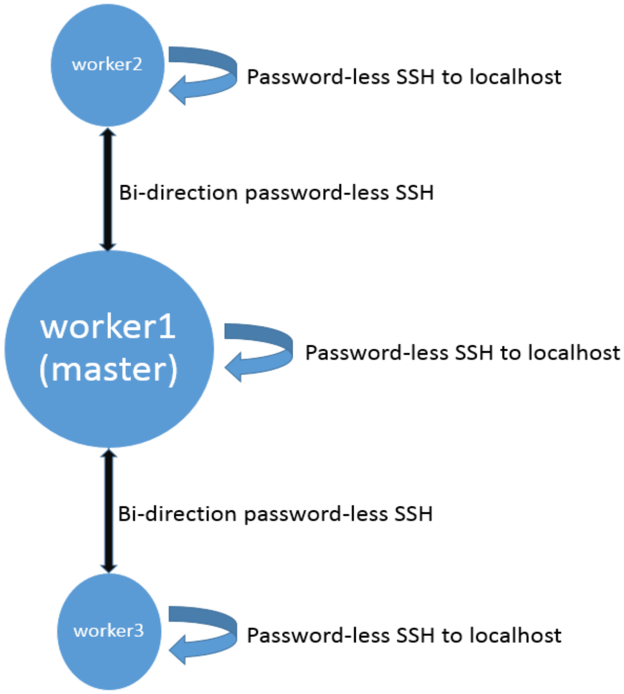

worker2
Password-less SSH to localhost
Bi-direction password-less SSH
worker1
(master)
Password-less SSH to localhost
Bi-direction password-less SSH
worker3
Password-less SSH to localhost

Figure 1 Password-less SSH Topology

2. Hadoop Configuration

1) Download Hadoop 2.7 from http://apache.mirrors.lucidnetworks.net/hadoop/common/hadoop-2.7.7/hadoop-2.7.7.tar.gz on all machines and unzip it to the same file path.

2) Append the following lines in HadoopFolder/etc/hadoop/core-site.xml inside the configuration. (modify core-site.xml file in both master and workers, replace the ip with your master private ip)

```
<property>
  <name>fs.default.name</name>
  <value>hdfs://192.168.0.1:54310</value>
</property>
```

3) Clear content and add the following lines in HadoopFolder/etc/hadoop/slaves. This step gives the list of all the machines who can be workers to Hadoop master. (master and workers, use private ip)

```
192.168.0.1
192.168.0.2
192.168.0.3
```

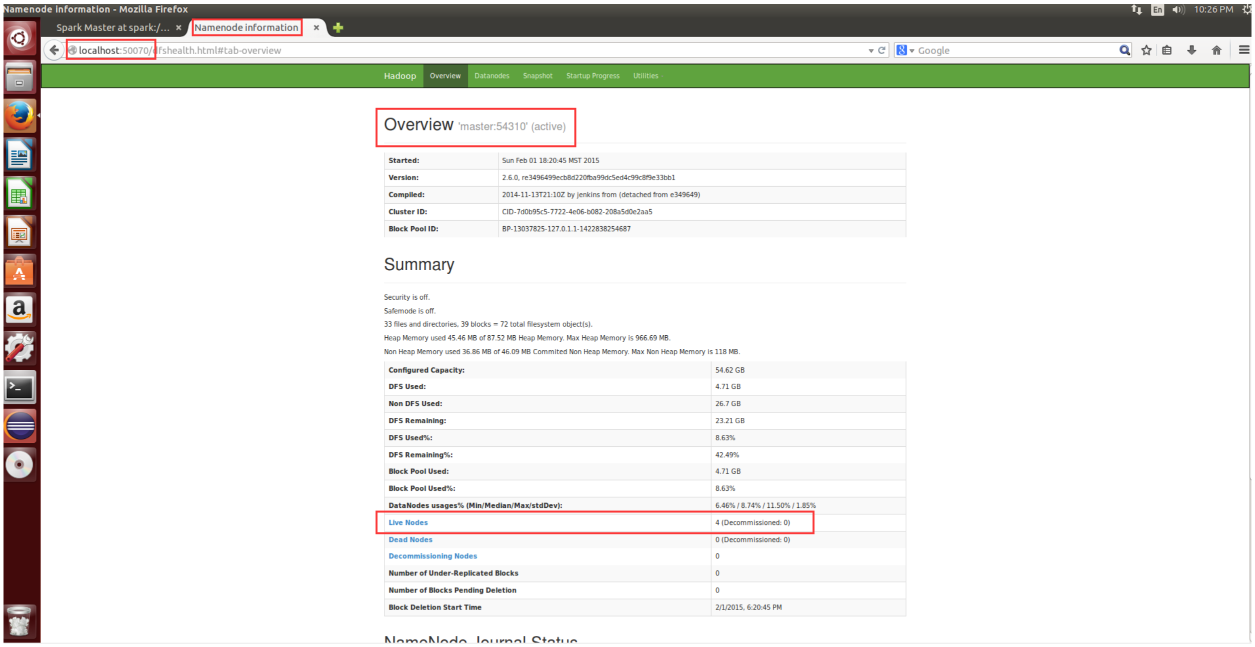
4) Run the following line in HadoopFolder/bin/ folder to format the HDFS system if this is the first time to run this Hadoop or after you change any machine attributes such as IP address. (Master only).

```
./hadoop namenode -format
```

5) Run the following line in HadoopFolder/sbin/ folder to start your cluster. It will take a while. (Master only)

```
./start-dfs.sh
```

6) Check the status of Hadoop cluster by entering “master_ public_ip:50070” in your browser. If you reach a page like the following, that means you have started Hadoop successfully. You should have 3 live nodes if you use three machines.


Namenode information - Mozilla Firefox
Spark Master at spark:/...
Namenode information
localhost:50070/dfshealth.html#tab-overview
Google
Hadoop
Overview
Datanodes
Snapshot
Startup Progress
Utilities

Overview 'master:54310' (active)

Started:	Sun Feb 01 18:20:45 MST 2015
Version:	2.6.0, re3496499ecb8d220fba99dc5ed4c99c8f9e33bb1
Compiled:	2014-11-13T21:10Z by jenkins from (detached from e349649)
Cluster ID:	CID-7d0b95c5-7722-4e06-b082-208a5d0e2aa5
Block Pool ID:	BP-13037825-127.0.1.1-1422838254687

Summary

Security is off.

Safemode is off.

33 files and directories, 39 blocks = 72 total filesystem object(s).

Heap Memory used 45.46 MB of 87.52 MB Heap Memory. Max Heap Memory is 966.69 MB.

Non Heap Memory used 36.86 MB of 46.09 MB Commited Non Heap Memory. Max Non Heap Memory is 118 MB.

Configured Capacity:	54.62 GB
DFS Used:	4.71 GB
Non DFS Used:	26.7 GB
DFS Remaining:	23.21 GB
DFS Used%:	8.63%
DFS Remaining%:	42.49%
Block Pool Used:	4.71 GB
Block Pool Used%:	8.63%
DataNodes usages% (Min/Median/Max/stdDev):	6.46% / 8.74% / 11.50% / 1.85%
Live Nodes	4 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)
Decommissioning Nodes	0
Number of Under-Replicated Blocks	0
Number of Blocks Pending Deletion	0
Block Deletion Start Time	2/1/2015, 6:20:45 PM

NameNode Journal Status

Figure 2 Hadoop Namenode Web UI

Tip: Sometimes you will have errors when setting up Hadoop. It is recommended to delete all the files related to Hadoop in /tmp in master and all workers. Then formant the node and start-dfs.sh.

3. Spark Configuration

1) Download Hadoop 2.7 from on all machines http://www-us.apache.org/dist/spark/spark-2.3.2/spark-2.3.2-bin-hadoop2.7.tgz and unzip it to the same file path.

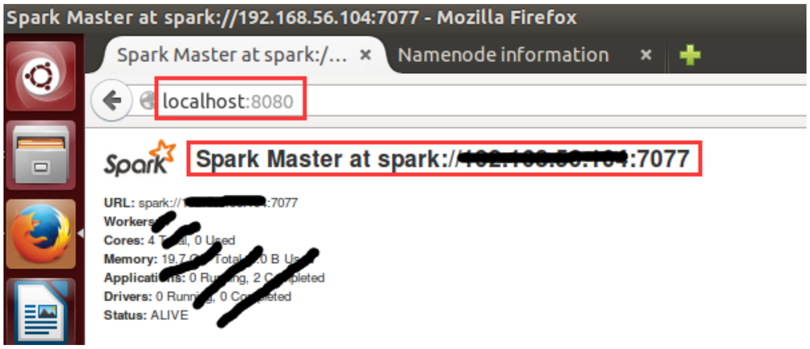
2) Add the following line in SparkFolder/conf/slaves.template only in Master Node. Then rename slaves.template to /slaves to make it come into effect. In our case, worker 2 has IP address 192.168.0.2 (public ip, if does not work then try the private ip).

```
192.168.0.1
192.168.0.2
```

3) Run the following line in SparkFolder/sbin/ folder to start Spark master. (Master only)

```
./start-all.sh
```

4) Check the status of Spark master by entering "master_public_ip:8080" in your linux browser. If you reach a page like the following, that means your master has been started. (Master only)



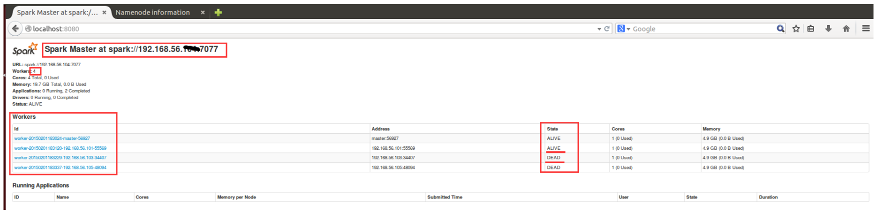
Spark Master at spark://192.168.56.104:7077 - Mozilla Firefox
Spark Master at spark:/...
Namenode information
localhost:8080
Spark

Spark Master at spark://[redacted]:7077

URL: spark://[redacted]:7077
Workers[redacted]
Cores: 4 [redacted], 0 Used
Memory: 19.7 [redacted] Total[redacted] 0 B U[redacted]
Applicati[redacted]**ns:** 0 R[redacted]ng, 2 C[redacted]pleted
Drivers: 0 Runn[redacted]g, 0 Co[redacted]eted
Status: ALIVE

Figure 3 Spark Master Web UI

5) There should be 3 workers in total in our scenario.



Spark Master at spark://192.168.56.[redacted]7077

URL: spark://192.168.56.104:7077
Workers: 4
Cores: 4 Total, 0 Used
Memory: 19.7 GB Total, 0.0 B Used
Applications: 0 Running, 2 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers

Id	Address	State	Cores	Memory
worker-20150201183024-master-56927	master:56927	ALIVE	1 (0 Used)	4.9 GB (0.0 B Used)
worker-20150201183120-192.168.56.101-55569	192.168.56.101:55569	ALIVE	1 (0 Used)	4.9 GB (0.0 B Used)
worker-20150201183229-192.168.56.103-34407	192.168.56.103:34407	DEAD	1 (0 Used)	4.9 GB (0.0 B Used)
worker-20150201183337-192.168.56.105-48094	192.168.56.105:48094	DEAD	1 (0 Used)	4.9 GB (0.0 B Used)

Running Applications

ID	Name	Cores	Memory per Node	Submitted Time	User	State	Duration

Figure 4 Spark Master Web UI after configuration