

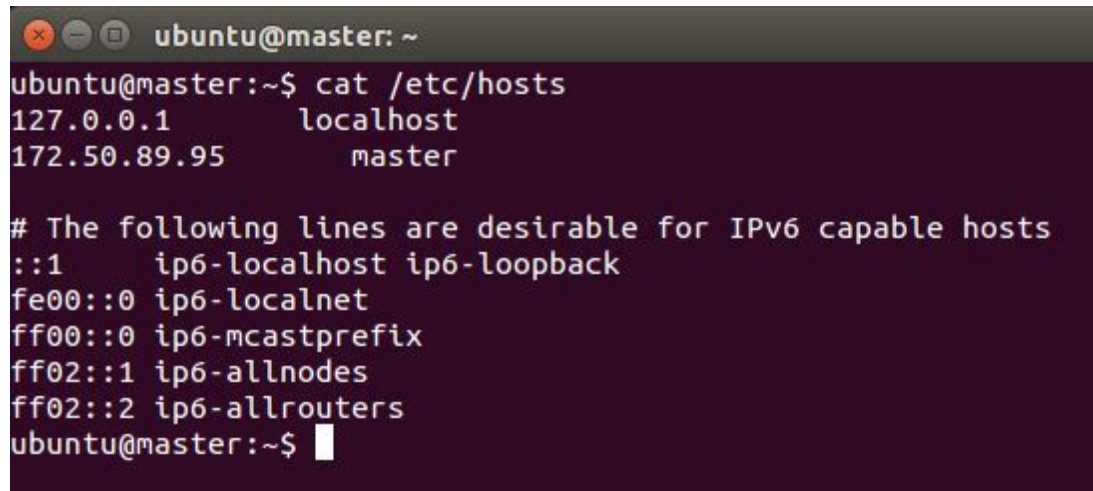
# Apache Spark

## Setting up Apache Spark cluster of N=3 nodes in standalone mode

### Pre-Setup:

- ❖ Setting up ubuntu 14.04.5 on all the three nodes having same user name
- ❖ Editing the /etc/hostname on one pc as “master” and other two as “worker1”, “worker2”
  - Making appropriate changes in /etc/hosts and instead of binding the hostname to 127.0.1.1 bind it to actual ip.

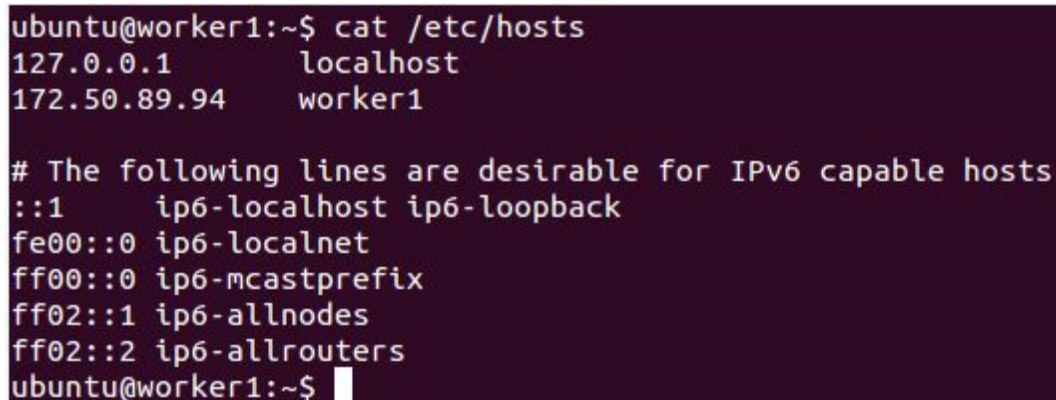
### Supporting screenshots

A terminal window titled 'ubuntu@master: ~' showing the contents of the /etc/hosts file. The file lists 127.0.0.1 as localhost and 172.50.89.95 as master. It also includes a section for IPv6 capable hosts with entries for ip6-localhost, ip6-loopback, ip6-localnet, ip6-mcastprefix, ip6-allnodes, and ip6-allrouters.

```
ubuntu@master:~$ cat /etc/hosts
127.0.0.1        localhost
172.50.89.95     master

# The following lines are desirable for IPv6 capable hosts
::1             ip6-localhost ip6-loopback
fe00::0         ip6-localnet
ff00::0         ip6-mcastprefix
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters
ubuntu@master:~$
```

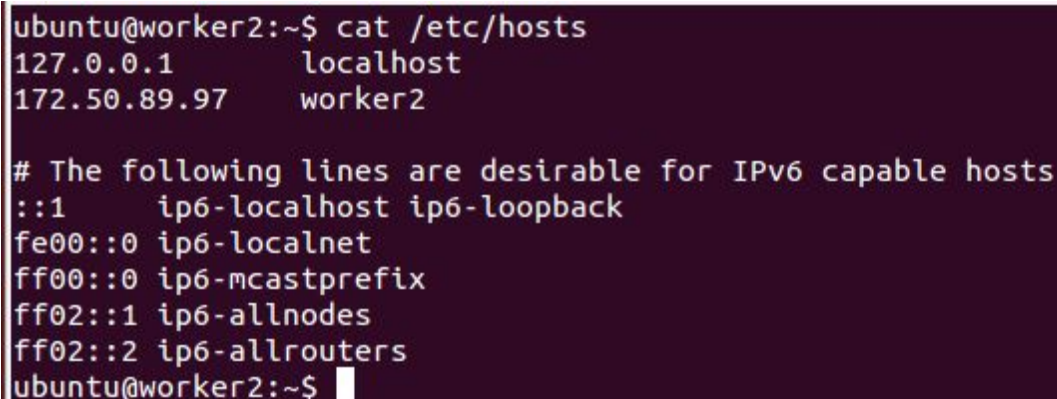
*/etc/hosts on master also see the changes done to replace 127.0.1.1 by actual ip*

A terminal window titled 'ubuntu@worker1:~\$' showing the contents of the /etc/hosts file. The file lists 127.0.0.1 as localhost and 172.50.89.94 as worker1. It also includes a section for IPv6 capable hosts with entries for ip6-localhost, ip6-loopback, ip6-localnet, ip6-mcastprefix, ip6-allnodes, and ip6-allrouters.

```
ubuntu@worker1:~$ cat /etc/hosts
127.0.0.1        localhost
172.50.89.94     worker1

# The following lines are desirable for IPv6 capable hosts
::1             ip6-localhost ip6-loopback
fe00::0         ip6-localnet
ff00::0         ip6-mcastprefix
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters
ubuntu@worker1:~$
```

*/etc/hosts on worker1*

A terminal window titled 'ubuntu@worker2:~\$' showing the contents of the /etc/hosts file. The file lists 127.0.0.1 as localhost and 172.50.89.97 as worker2. It also includes a section for IPv6 capable hosts with entries for ip6-localhost, ip6-loopback, ip6-localnet, ip6-mcastprefix, ip6-allnodes, and ip6-allrouters.

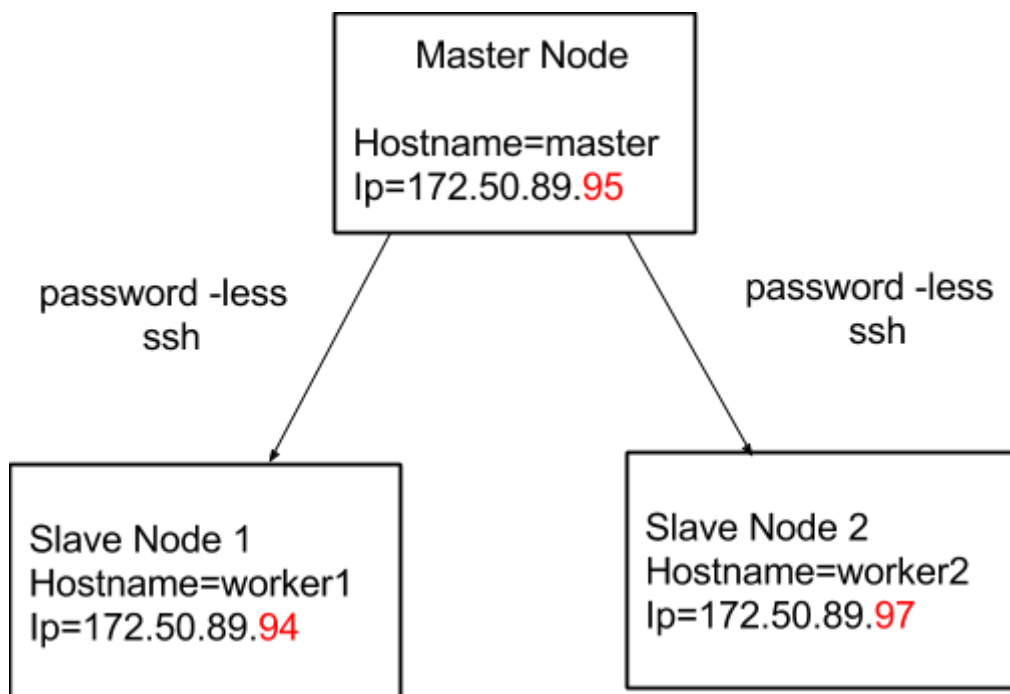
```
ubuntu@worker2:~$ cat /etc/hosts
127.0.0.1        localhost
172.50.89.97     worker2

# The following lines are desirable for IPv6 capable hosts
::1             ip6-localhost ip6-loopback
fe00::0         ip6-localnet
ff00::0         ip6-mcastprefix
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters
ubuntu@worker2:~$
```

*/etc/hosts on worker2*

## Architecture of cluster

Ip address	Status(Master or Worker)
172.50.89.95	Master and Worker
172.50.89.94	Worker1
172.50.89.97	Worker2



Now to install spark on all the nodes run the install.sh script on all nodes with sudo permissions

➤ **sudo sh [install.sh](#)**

It will install **spark** (version-2.0.0), **jdk7**, **scala** (version-2.11.7), and also sets up path as required

Now for password-less ssh so that master and worker can communicate easily. In order to start worker services and interact with workers, master node should have login access to worker nodes. Generate private SSH key on master node and add the same to workers node.

Go into the home directory of master

➤ **ssh-keygen**

➤ **ssh-copy-id -i ~/.ssh/id\_rsa.pub 172.50.89.94**

(on prompting enter worker1 password)

➤ **ssh-copy-id -i ~/.ssh/id\_rsa.pub 172.50.89.97**

(on prompting enter worker2 password)

➤ **cat ~/.ssh/id\_rsa.pub >> ~/.ssh/.authorized\_keys**

## Supporting screenshots

```
ubuntu@master:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
/home/ubuntu/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa.
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub.
The key fingerprint is:
0e:26:86:7d:d9:77:11:46:d2:9b:a5:f6:a7:40:ed:de ubuntu@master
The key's randomart image is:
+--[ RSA 2048 ]-----+
|           .o+       |
|            o...     |
|             *.      |
|      o   o   *..    |
|    . + = S .o.o     |
|    . + o . . . o .  |
|             o +     |
|              o E    |
+-----+
ubuntu@master:~$
```

*Ssh-keygen on master*

```
ubuntu@master:~$ cd ~/.ssh/
ubuntu@master:~/.ssh$ ssh-copy-id -i ./id_rsa.pub 172.50.89.94
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
ubuntu@172.50.89.94's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh '172.50.89.94'"
and check to make sure that only the key(s) you wanted were added.

ubuntu@master:~/.ssh$
```

*ssh-copy-id -i ~/.ssh/id\_rsa.pub 172.50.89.94*

```

ubuntu@master:~/ssh$ ssh-copy-id -i ./id_rsa.pub 172.50.89.97
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
Agent admitted failure to sign using the key.
ubuntu@172.50.89.97's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh '172.50.89.97'"
and check to make sure that only the key(s) you wanted were added.

ubuntu@master:~/ssh$ █

```

*ssh-copy-id -i ~/.ssh/id\_rsa.pub 172.50.89.97*

```

ubuntu@master:~$ ssh 172.50.89.94
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 4.4.0-38-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

17 packages can be updated.
14 updates are security updates.

New release '16.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2019.
Last login: Sat Oct  8 11:37:39 2016 from 172.50.89.95
ubuntu@worker1:~$ █

```

*Test to login to worker1 without password*

Now we update some conf files on master to make our cluster run as standalone cluster

Goto /usr/local/spark/conf

➤ **cd /usr/local/spark/conf**

➤ **cp slaves.template slaves**

➤ **sudo nano slaves**

In this file add

localhost

172.50.89.94

172.50.89.97



Now edit spark-env.sh

➤ **cp spark-env.sh.template spark-env.sh**

➤ **sudo nano spark-env.sh**

(you can use spark-env.sh.template file to edit)

In this file add

SPARK\_WORKER\_INSTANCES=2

SPARK\_MASTER\_IP=172.50.89.95

SPARK\_LOCAL\_IP=172.50.88.95 # This ip should be corresponding to each node.

(In order to start N number of worker instances we need to update spark-env.sh file)

Now edit spark-env.sh in each of worker nodes

➤ **cp spark-env.sh.template spark-env.sh**

➤ **sudo nano spark-env.sh**

(you can use spark-env.sh.template file to edit)

In this file add

SPARK\_WORKER\_INSTANCES=2

SPARK\_MASTER\_IP=172.50.89.95

SPARK\_LOCAL\_IP=172.50.88.94 # This ip should be corresponding to each node.

Spark comes with inbuilt scripts to start

Just go to sbin directory on master and start

➤ **cd /usr/local/spark/sbin**

➤ **./start-all.sh**

```

ubuntu@master:/usr/local/spark/sbin$ ./start-all.sh
starting org.apache.spark.deploy.master.Master, logging to /usr/local/spark/logs
/spark-ubuntu-org.apache.spark.deploy.master.Master-1-master.out
localhost: starting org.apache.spark.deploy.worker.Worker, logging to /usr/local
/spark/logs/spark-ubuntu-org.apache.spark.deploy.worker.Worker-1-master.out
172.50.89.94: starting org.apache.spark.deploy.worker.Worker, logging to /usr/lo
cal/spark/logs/spark-ubuntu-org.apache.spark.deploy.worker.Worker-1-worker1.out
172.50.89.97: starting org.apache.spark.deploy.worker.Worker, logging to /usr/lo
cal/spark/logs/spark-ubuntu-org.apache.spark.deploy.worker.Worker-1-worker2.out
localhost: starting org.apache.spark.deploy.worker.Worker, logging to /usr/local
/spark/logs/spark-ubuntu-org.apache.spark.deploy.worker.Worker-2-master.out
172.50.89.94: starting org.apache.spark.deploy.worker.Worker, logging to /usr/lo
cal/spark/logs/spark-ubuntu-org.apache.spark.deploy.worker.Worker-2-worker1.out
172.50.89.97: starting org.apache.spark.deploy.worker.Worker, logging to /usr/lo
cal/spark/logs/spark-ubuntu-org.apache.spark.deploy.worker.Worker-2-worker2.out
172.50.89.94: starting org.apache.spark.deploy.worker.Worker, logging to /usr/lo
cal/spark/logs/spark-ubuntu-org.apache.spark.deploy.worker.Worker-3-worker1.out
ubuntu@master:/usr/local/spark/sbin$ jps
8936 Jps
8621 Master
8773 Worker
8850 Worker
ubuntu@master:/usr/local/spark/sbin$

```

We can see it on web gui also  
172.50.89.95:8080

Spark Master at spark://172.50.89.95:7077

URL: spark://172.50.89.95:7077  
REST URL: spark://172.50.89.95:8080 (cluster mode)  
Alive Workers: 7  
Cores in use: 36 Total, 0 Used  
Memory in use: 133.5 GB Total, 0.0 B Used  
Applications: 0 Running, 0 Completed  
Drivers: 0 Running, 0 Completed  
Status: ALIVE

Worker Id	Address	State	Cores	Memory
worker-20161008125048-172.50.89.94-39272	172.50.89.94:39272	ALIVE	4 (0 Used)	14.6 GB (0.0 B Used)
worker-20161008125631-172.50.89.97-40435	172.50.89.97:40435	ALIVE	8 (0 Used)	14.5 GB (0.0 B Used)
worker-20161008125634-172.50.89.97-40332	172.50.89.97:40332	ALIVE	8 (0 Used)	14.5 GB (0.0 B Used)
worker-20161008125659-172.50.89.94-38616	172.50.89.94:38616	ALIVE	4 (0 Used)	14.6 GB (0.0 B Used)
worker-20161008125702-172.50.89.94-38072	172.50.89.94:38072	ALIVE	4 (0 Used)	14.6 GB (0.0 B Used)
worker-20161009125824-172.50.89.95-32923	172.50.89.95:32923	ALIVE	4 (0 Used)	30.3 GB (0.0 B Used)
worker-20161009125826-172.50.89.95-45494	172.50.89.95:45494	ALIVE	4 (0 Used)	30.3 GB (0.0 B Used)

Running Applications

Application ID	Name	Cores	Memory per Node	Submitted Time	User	State	Duration
----------------	------	-------	-----------------	----------------	------	-------	----------

Completed Applications

Application ID	Name	Cores	Memory per Node	Submitted Time	User	State	Duration
----------------	------	-------	-----------------	----------------	------	-------	----------

Now to run our search-query

```

spark-submit --master spark://172.50.89.95:7077 <python file>
<input-folder> <query-file>

```

➤ **spark-submit --master spark://172.50.89.95:7077 [spark.py](#) raw.en  
[query](#)**

Currently our cluster supports only single word search and returns no. of hits it received of that particular word in that text file.

```
ubuntu@master:~$ spark-submit --master spark://172.50.89.95:7077 spark.py raw.en query

16/10/09 13:35:29 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes
16/10/09 13:35:30 WARN SparkConf:
SPARK_WORKER_INSTANCES was detected (set to '2').
This is deprecated in Spark 1.0+.

Please instead use:
- ./spark-submit with --num-executors to specify the number of executors
- Or set SPARK_EXECUTOR_INSTANCES
- spark.executor.instances to configure the number of instances in the spark config.

Time to build inverted index is 1.41080093384
[Stage 0:>                                     (0 + 2) / 2]
```

Time to build whole inverted index is 1.4108 sec  
And then system comes to halt.

So it was built again with only three docs from original corpus and ran again with same set of queries .

➤ **spark-submit --master spark://172.50.89.95:7077 [spark.py](#) corpus [query](#)** > output  
Time to build inverted index was 0.568 secs

The screenshot shows the Spark Master web interface at `spark://172.50.89.95:7077`. The interface displays cluster health, worker information, and application status.

**Cluster Status:**

- URL: `spark://172.50.89.95:7077`
- REST URL: `spark://172.50.89.95:6066` (cluster mode)
- Alive Workers: 7
- Cores in use: 36 Total, 36 Used
- Memory in use: 133.5 GB Total, 7.0 GB Used
- Applications: 1 Running, 11 Completed
- Drivers: 0 Running, 0 Completed
- Status: ALIVE

**Workers Table:**

Worker Id	Address	State	Cores	Memory
worker-20161008125046-172.50.89.94-39272	172.50.89.94:39272	ALIVE	4 (4 Used)	14.6 GB (1024.0 MB Used)
worker-20161008130832-172.50.89.97-34692	172.50.89.97:34692	ALIVE	8 (8 Used)	14.5 GB (1024.0 MB Used)
worker-20161008130834-172.50.89.97-44621	172.50.89.97:44621	ALIVE	8 (8 Used)	14.5 GB (1024.0 MB Used)
worker-20161008130900-172.50.89.94-43137	172.50.89.94:43137	ALIVE	4 (4 Used)	14.6 GB (1024.0 MB Used)
worker-20161008130903-172.50.89.94-33069	172.50.89.94:33069	ALIVE	4 (4 Used)	14.6 GB (1024.0 MB Used)
worker-20161009131024-172.50.89.95-42626	172.50.89.95:42626	ALIVE	4 (4 Used)	30.3 GB (1024.0 MB Used)
worker-20161009131026-172.50.89.95-36852	172.50.89.95:36852	ALIVE	4 (4 Used)	30.3 GB (1024.0 MB Used)

**Running Applications Table:**

Application ID	Name	Cores	Memory per Node	Submitted Time	User	State	Duration
app-20161009134936-0011	(kill) Inverted Index	36	1024.0 MB	2016/10/09 13:49:36	ubuntu	RUNNING	3.1 min

**Completed Applications Table:**

Application ID	Name	Cores	Memory per Node	Submitted Time	User	State	Duration
app-20161009134843-0010	Inverted Index	36	1024.0 MB	2016/10/09 13:48:43	ubuntu	FINISHED	41 s
app-20161009134825-0009	Inverted Index	36	1024.0 MB	2016/10/09 13:48:25	ubuntu	FINISHED	9 s
app-20161009133829-0008	Inverted Index	36	1024.0 MB	2016/10/09 13:38:29	ubuntu	FINISHED	9.6 min
app-20161009133530-0007	Inverted Index	36	1024.0 MB	2016/10/09 13:35:30	ubuntu	FINISHED	2.9 min
app-20161009132622-0006	Inverted Index	36	1024.0 MB	2016/10/09 13:26:22	ubuntu	FINISHED	42 s
app-20161009132139-0005	Inverted Index	36	1024.0 MB	2016/10/09 13:21:39	ubuntu	FINISHED	46 s
app-20161009132005-0004	Inverted Index	36	1024.0 MB	2016/10/09 13:20:05	ubuntu	FINISHED	41 s
app-20161009131726-0003	Inverted Index	36	1024.0 MB	2016/10/09 13:17:26	ubuntu	FINISHED	41 s
app-20161009131720-0002	Inverted Index	36	1024.0 MB	2016/10/09 13:17:20	ubuntu	FINISHED	0.4 s

Web gui when our app is running  
It shows all the instances and the apps running.

Application: Inverted Index - Google Chrome

Spark 2.0.0 Application: Inverted Index

ID: app-20161009134936-0011  
 Name: Inverted Index  
 User: ubuntu  
 Cores: Unlimited (36 granted)  
 Executor Memory: 1024.0 MB  
 Submit Date: Sun Oct 09 13:49:36 IST 2016  
 State: RUNNING  
[Application Detail UI](#)

**Executor Summary**

ExecutorID	Worker	Cores	Memory	State	Logs
2	worker-20161008125048-172.50.89.94-39272	4	1024	EXITED	<a href="#">stdout stderr</a>
5	worker-20161009131026-172.50.89.95-36852	4	1024	RUNNING	<a href="#">stdout stderr</a>
3	worker-20161008130900-172.50.89.94-43137	4	1024	EXITED	<a href="#">stdout stderr</a>
9	worker-20161008125048-172.50.89.94-39272	4	1024	RUNNING	<a href="#">stdout stderr</a>
4	worker-20161008130903-172.50.89.94-33069	4	1024	EXITED	<a href="#">stdout stderr</a>
11	worker-20161008130900-172.50.89.94-43137	4	1024	RUNNING	<a href="#">stdout stderr</a>
0	worker-20161008130834-172.50.89.97-44621	8	1024	EXITED	<a href="#">stdout stderr</a>
10	worker-20161008130903-172.50.89.94-33069	4	1024	RUNNING	<a href="#">stdout stderr</a>
8	worker-20161008130834-172.50.89.97-44621	8	1024	RUNNING	<a href="#">stdout stderr</a>
1	worker-20161008130832-172.50.89.97-34692	8	1024	EXITED	<a href="#">stdout stderr</a>
7	worker-20161008130832-172.50.89.97-34692	8	1024	RUNNING	<a href="#">stdout stderr</a>
6	worker-20161009131024-172.50.89.95-42626	4	1024	RUNNING	<a href="#">stdout stderr</a>

Provides info about our running app like which instance is using how much memory, its status etc

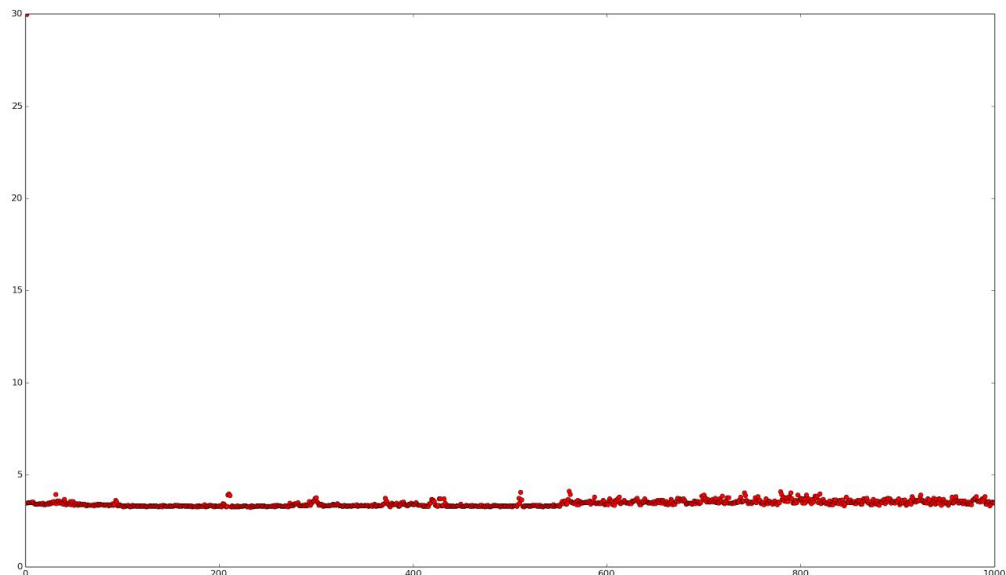
Spark 2.0.0 Spark Worker at 172.50.89.94:43137

ID: worker-20161008130900-172.50.89.94-43137  
 Master URL: spark://172.50.89.95:7077  
 Cores: 4 (4 Used)  
 Memory: 14.6 GB (1024.0 MB Used)  
[Back to Master](#)

**Running Executors (1)**

ExecutorID	Cores	State	Memory	Job Details	Logs
101	4	RUNNING	1024.0 MB	ID: app-20161009134936-0011 Name: Inverted Index User: ubuntu	<a href="#">stdout stderr</a>

Tells specifically about each instances running on each node. We can see logs also from here.



Graph generated by running 1000 queries

In the directory along with install.sh, spark.py, the query file and output got by running it for corpus of three docs is also included.

\*\*\*\*\*THANK YOU\*\*\*\*\*