

CSC 555: Mining Big Data

Project, Phase 1 (due Thursday, February 15th)

In this part of the project, you will 1) Set up a 3-node cluster and 2) perform data warehousing and transformation queries using Hive, Pig and Hadoop streaming. The modified Hive-style schema is at: http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/SSBM1/SSBM_schema_hive.sql
It is based on SSBM benchmark (derived from industry standard TPCB benchmark). I modified it from SQL to HiveQL. This is Scale1, or the smallest unit – lineorder is the largest table at about 0.6GB. You can use wget to download the following links. Keep in mind that data is | -separated (not csv).

<http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/SSBM1/dwdate.tbl>

<http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/SSBM1/lineorder.tbl>

<http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/SSBM1/part.tbl>

<http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/SSBM1/supplier.tbl>

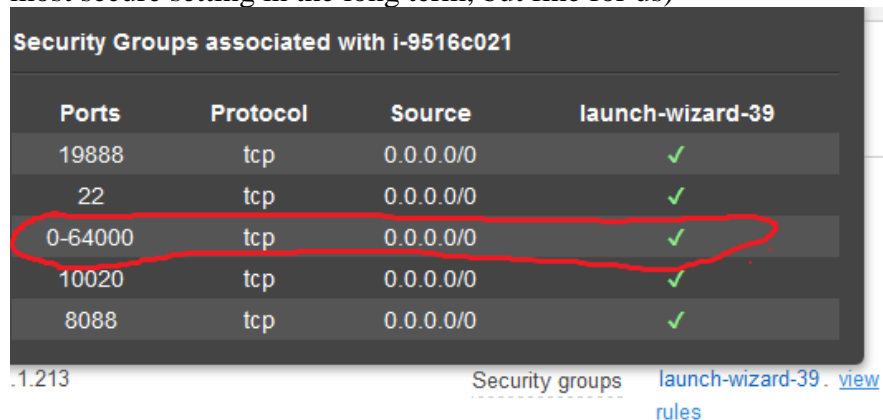
<http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/SSBM1/customer.tbl>

Please be sure to submit all code (pig, python and SQL).

Part 1: Multi-node cluster

1) Your first step is to setup a multi-node cluster and re-run a simple wordcount. For this part, you will create a 3-node cluster (with a total of 1 master + 2 worker nodes). Include your master node in the “slaves” file, to make sure all 3 nodes are working.
You need to perform the following steps:

1. Create a new node of a medium size. It is possible, but I do not recommend trying to reconfigure your existing Hadoop into this new cluster (it is much easier to make 3 new nodes for a total of 4 in your AWS account).
 - a. When creating a node I recommend changing the default 8G hard drive to 30G so that you do not run out of space easily.
 - b. Change your security group setting to open firewall access. Rather than figure out all individual port, you can set 0-64000 range opening up all ports (not the most secure setting in the long term, but fine for us)



Security Groups associated with i-9516c021

Ports	Protocol	Source	launch-wizard-39
19888	tcp	0.0.0.0/0	✓
22	tcp	0.0.0.0/0	✓
0-64000	tcp	0.0.0.0/0	✓
10020	tcp	0.0.0.0/0	✓
8088	tcp	0.0.0.0/0	✓

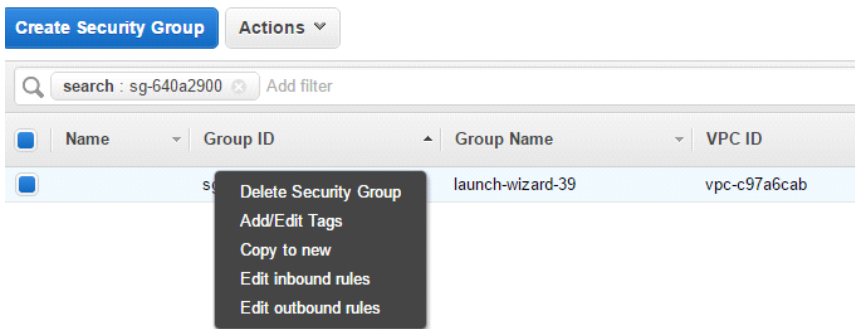
.1.213 Security groups launch-wizard-39. [view rules](#)

- c. Step by step instructions on how to make the change to open up the ports:

Click on security group (launch-wizard-x)

Elastic IPs	
Availability zone	us-west-1b
Security groups	launch-wizard-39. view rules
Scheduled events	-

Right click on the security group and choose Edit inbound rules



Add a new rule and put in the ports 0-64000 and “Anywhere” and click save.

Custom TCP Rule	TCP	0-64000	Anywhere	0.0.0.0/0	X
Add Rule				Cancel	Save

This will open the firewall completely for all ports.

- d. Finally, right click on the Master node and choose “create more like this” to create 2 more nodes with same settings. If you configure the network settings on master first, security group information will be copied.
NOTE: Hard drive size will not be copied and default to 8G unless you change it.
2. Connect to the master and set up Hadoop similarly to what you did previously. Do not attempt to repeat these steps on workers yet – you will only need to set up Hadoop once.
 - a. Configure core-site.xml, adding the **PrivateIP** (do not use public IP) of the master.

```
limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>
<name>fs.defaultFS</name>
<value>hdfs://172.31.7.201/</value>
</property>

</configuration>
[ec2-user@ip-172-31-7-201 ~]$ cat hadoop-2.6.4/etc/hadoop/core-site.xml
```

- b. Configure hdfs-site and set replication factor to 2.

```
<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>
<name>dfs.replication</name>
<value>2</value>
</property>

</configuration>
[ec2-user@ip-172-31-9-105 ~]$
```

- c. cp hadoop-2.6.4/etc/hadoop/mapred-site.xml.template hadoop-2.6.4/etc/hadoop/mapred-site.xml and then configure mapred-site.xml

```
<!-- Put site-specific property overrides in this file. -->

<configuration>

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

</configuration>
[ec2-user@ip-172-31-9-105 ~]$ cat hadoop-2.6.4/etc/hadoop/mapred-site.xml
```

- d. Configure yarn-site.xml (once again, use PrivateIP of the master)

```
<!-- Site specific YARN configuration properties -->

<property>
<name>yarn.resourcemanager.hostname</name>
<value>172.31.7.201</value>
</property>

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

</configuration>
[ec2-user@ip-172-31-7-201 ~]$ cat hadoop-2.6.4/etc/hadoop/yarn-site.xml
```

Finally, edit the slaves file and list your 3 nodes (master and 2 workers) using Private IPs

```
[ec2-user@ip-172-31-7-201 ~]$ cat hadoop-2.6.4/etc/hadoop/slaves
172.31.7.201
172.31.5.246
172.31.11.50
```

Make sure that you use private IP (private DNS is also ok) for your configuration files (such as conf/masters and conf/slaves or the other 3 config files). The advantage of the Private IP is that it does not change after your instance is stopped (if you use the Public IP, the cluster would need to be reconfigured every time it is stopped). The downside of the Private IP is that it is only

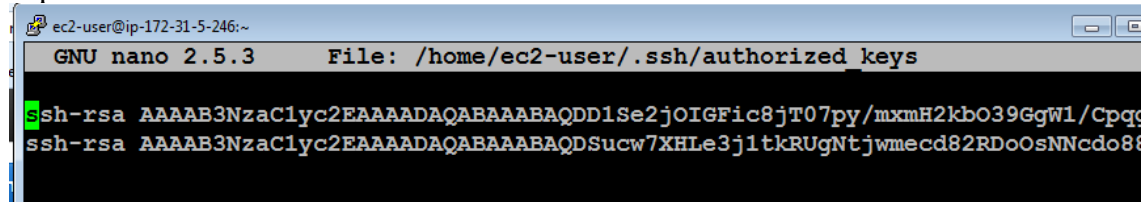
meaningful within the Amazon EC2 network. So all nodes in EC2 can talk to each other using Private IP, but you cannot connect to your instance from the outside (e.g., from your laptop) because Private IP has no meaning for your laptop (since your laptop is not part of the Amazon EC2 network).

Now, we will pack up and move Hadoop to the workers. All you need to do is to generate and then copy the public key to the worker nodes to achieve passwordless access across your cluster.

1. Run `ssh-keygen -t rsa` (and enter empty values for the passphrase) on the master node. That will generate `.ssh/id_rsa` and `.ssh/id_rsa.pub` (private and public key). You now need to manually copy the `.ssh/id_rsa.pub` and append it to `~/.ssh/authorized_keys` **on each worker.**

Keep in mind that this is a single-line public key and accidentally introducing a line break would break it. For example:

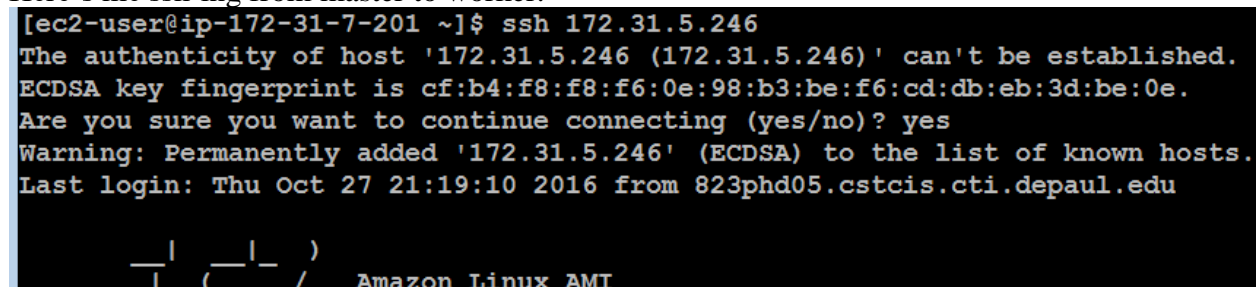
Note that this is NOT the master, but one of the workers (ip-172-31-5-246). The first public key is the .pem Amazon half and the 2nd public key is the master's public key copied in as one line.



```
ec2-user@ip-172-31-5-246:~  
GNU nano 2.5.3 File: /home/ec2-user/.ssh/authorized_keys  
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDD1Se2jOIGFic8jT07py/mxmH2kbo39GgW1/Cpq  
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDSucw7XHLe3j1tkRUgNtjwmecd82RDoOsNNcdo88
```

You can add the public key of the master to the master by running this command:
`cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys`

Make sure that you can ssh to all of the nodes from the master node (by running `ssh 54.186.221.92`, where the IP address is your worker node) from the master and ensuring that you were able to login. You can exit after successful ssh connection by typing `exit` (the command prompt will tell you which machine you are connected to, e.g., `ec2-user@ip-172-31-37-113`). Here's me ssh-ing from master to worker.



```
[ec2-user@ip-172-31-7-201 ~]$ ssh 172.31.5.246  
The authenticity of host '172.31.5.246 (172.31.5.246)' can't be established.  
ECDSA key fingerprint is cf:b4:f8:f8:f6:0e:98:b3:be:f6:cd:db:eb:3d:be:0e.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added '172.31.5.246' (ECDSA) to the list of known hosts.  
Last login: Thu Oct 27 21:19:10 2016 from 823phd05.cstcis.cti.depaul.edu  
  
_ | _ | )  
_ | ( _ | / Amazon Linux AMI
```

Once you have verified that you can ssh from the master node to every cluster member including the master itself (ssh localhost), you are going to return to the master node (**exit** until your prompt shows the IP address of the master node) and pack the contents of the hadoop directory there. Make sure your Hadoop installation is configured correctly (because from now on, you will have 3 copies of the Hadoop directory and all changes need to be applied in 3 places).

cd (go to root home directory, i.e. /home/ec2-user/)

(pack up the entire Hadoop directory into a single file for transfer. You can optionally compress the file with gzip)

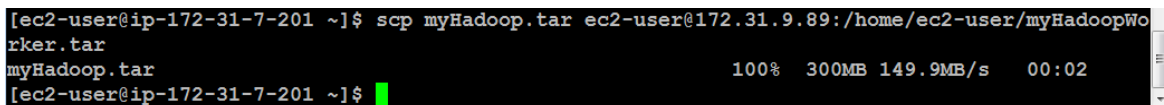
tar cvf myHadoop.tar hadoop-2.6.4

ls -al myHadoop.tar (to verify that the .tar file had been created)

Now, you need to copy the myHadoop.tar file to every non-master node in the cluster. If you had successfully setup public-private key access in the previous step, this command (for each worker node) will do that:

(copies the myHadoop.tar file from the current node to a remote node into a file called myHadoopWorker.tar. Don't forget to replace the IP address with that your worker nodes. By the way, since you are on the Amazon EC2 network, either Public or Private IP will work just fine.)

scp myHadoop.tar ec2-user@54.187.63.189:/home/ec2-user/myHadoopWorker.tar



```
[ec2-user@ip-172-31-7-201 ~]$ scp myHadoop.tar ec2-user@172.31.9.89:/home/ec2-user/myHadoopWorker.tar
myHadoop.tar                                100% 300MB 149.9MB/s   00:02
[ec2-user@ip-172-31-7-201 ~]$
```

Once the tar file containing your Hadoop installation from master node has been copied to each worker node, you need to login to each non-master node and unpack the .tar file.

Run the following command (on each worker node, not on the master) to untar the hadoop file. We are purposely using a different tar archive name (i.e., **myHadoopWorker.tar**), so if you get “file not found” error, that means you are running this command on the master node or have not successfully copied myHadoopWorker.tar file.

tar xvf myHadoopWorker.tar

Once you are done, run this on the master (nothing needs to be done on the workers to format the cluster unless you are re-formatting, in which case you'll need to delete the dfs directory).

hadoop namenode -format

Once you have successfully completed the previous steps, you should be able to start and use your new cluster by going to the master node and running the start-dfs.sh and start-yarn.sh scripts (you do not need to explicitly start anything on worker nodes – the master will do that for you).

You should verify that the cluster is running by pointing your browser to the link below.

[http://\[insert-the-public-ip-of-master\]:50070/](http://[insert-the-public-ip-of-master]:50070/)

Make sure that the cluster is operational (you can see the 3 nodes under Datanodes tab).

Submit a screenshot of your cluster status view.

Repeat the steps for wordcount using bioproject.xml from Assignment 1 and submit screenshots of running it.

Part 2: Hive

Run the following three (1.2, 1.3 and 2.1) queries in Hive and record the time they take to execute:

http://rasinsrv07.cstcis.cti.depaul.edu/CSC555/SSBM1/SSBM_queries.sql

Perform the following transform operation on the dwdate.tbl

For the 7th column (e.g., Jan1992), transform it into two columns (e.g., Jan and 1992, repeating the year). This will change dwdate from a 17-column table into an 18-column table, replacing 7th column with two columns instead.

Part 3: Pig

Convert and load the data into Pig, implementing only queries 0.1, 0.2, 0.3.

Check disk storage, if your disk usage is over 90% Pig may hang without an error or a warning.

One easy way to time Pig is as follows: put your sequence of pig commands into a text file and then run, from command line in pig directory (e.g., [ec2-user@ip-172-31-6-39 pig-0.15.0]\$), **bin/pig -f pig_script.pig** (which will inform you how long the pig script took to run).

Part 4: Hadoop Streaming

Implement query **0.3** (only that query) using Hadoop streaming with python.

NOTE: You may implement this part in Java if you prefer.

Submit a single document containing your written answers. Be sure that this document contains your name and “CSC 555 Project Phase 1” at the top.