

Exercise 7c

Creating a Cassandra Cluster in EC2

Prior Knowledge

Unix Command Line Shell
Amazon AWS / EC2 Console
Previous Cassandra exercises

Learning Objectives

Demonstrate Cassandra replication and scaling

Software Requirements

- AWS, Cassandra AMI

Overall plan

We are going to start a new (medium size) instance and deploy DataStax OpsCenter on it. From there we are going to create a cluster on EC2 (OpsCenter will do all the hard work for us). Finally we will install the Cassandra stress tools on the opscenter machine and test the cluster. We can view the cluster status on the dashboard.

Please note that DataStax OpsCenter is not open source. As of the latest versions it does not support using the Community edition of Cassandra either. We are using an older version that still supports that. There are plenty of other ways of setting up a Cassandra cluster in the cloud. However, they probably would take too long for this course.

1. Start up the AWS Console and login.
2. Make sure you are using the eu-west (Ireland) region.
3. Check that you can see the cassandra-security-group
4. Start up a new Launch Wizard and choose the same Ubuntu 14.04 VM we've used previously.
5. Choose **t2.medium** as the instance size.
6. Click through Next repeatedly until you get to Tag instance and call it: oxcloXX-opscenter
7. **Next: Configure Security Group**
8. Choose Existing Group and then **cassandra-security-group**

9. Continue to Launch the instances and choose your own keypair (so that you can access the instances).
10. Find the IP address of the new instance and once it has started up, then SSH to the instance.
11. In the SSH session type cut and paste the following shell command from:
<http://freo.me/ox-clo-cass>

```
echo "deb http://debian.datastax.com/community stable main" | sudo tee -a /etc/apt/sources.list.d/datastax.community.list && \
curl -L http://debian.datastax.com/debian/repo_key | sudo apt-key add - && \
sudo add-apt-repository -y ppa:openjdk-r/ppa && \
sudo apt-get update && \
sudo apt-get install -y openjdk-8-jdk python-support && \
sudo apt-get install -y cassandra=2.1.11 cassandra-tools=2.1.11 && \
sudo apt-get install -y opscenter
```

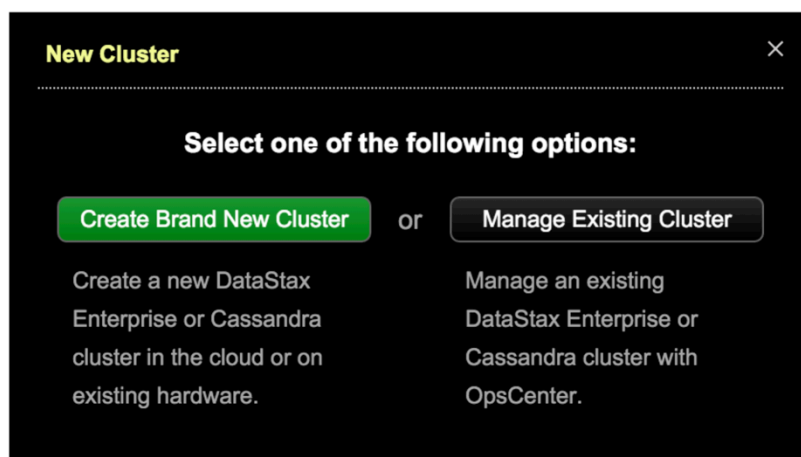
It should go away and install DataStax OpsCenter and Cassandra

12. By default there is no security on the opscenter. Ideally we would enable full SSL, certificates, etc. But that is overkill for this lab. On the other hand, leaving it wide open on an internet facing host is a bit mad, so we'll turn on authentication (even if its only over HTTP and not HTTPS).

```
sudo nano /etc/opscenter/opscenterd.conf
```

13. Now change the line under [authentication] from enabled=False to enabled=True
14. Start the opscenter:

```
sudo service opscenterd start
```
15. Now from your browser browse ip:8888 (where ip is the ipaddress of your instance).
16. Login with admin/admin
You should see:



17. Click on Create Brand New Cluster

18. Now fill in the following:

Cluster name:	oxcloXX-cassandra
Provisioning type:	Cloud
Package:	Datastax Community 2.1.11
Access Key/Secret Access Key:	your keys from exercise 1

Create Cluster

Cluster Name:

Provisioning Type:

Package:

Endpoint Snitch:

Amazon EC2 Credentials:

[View Advanced Options](#)

19. Now click Add Datacenter

20. On the next page fill in:

Number of Nodes: 3
Leave alone the next few
Size: m3.large
Untick **Use OpsCenter specific security group**
since we already have a Cassandra security group
Security Group: cassandra-security-group
Untick **Use OpsCenter specific keypair**
Keypair: your keypair

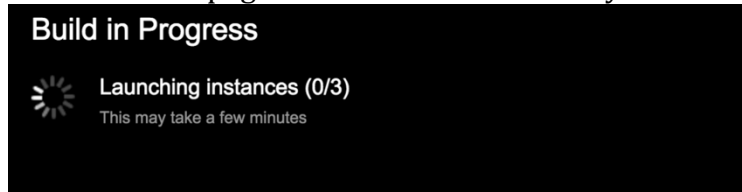
Now copy and paste the contents of your keypair .PEM file into the box
(it will look a bit like this but longer):

```
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAjk5z0qN0jayxkP//c6nSxD17CfLPYPwgh0iJ0dy+up4pYkrW/30YqVDP1WE
...
t6TYiViupkIXqzwdkrkvWkgUqyVqfxzBRhsrCsGH0yWzyZFSS45+eYh8mx00eLB4dhw=
-----END RSA PRIVATE KEY-----
```

It should look like:

21. Click Add

22. On the review page click **Build Cluster** and you will see:



23. Go check on the EC2 Instances page and you should see three new instances being launched.

24. Now that we have completed the wizard, opscenter will let us change our password from admin. Best do that right away.

25. Now find the IP address of one of the newly created nodes. You can see the IP address from the EC2 dashboard or from the opscenter webpage. Check that you can connect to the cluster using cqlsh from your SSH session:

```
cqlsh ip-address
```

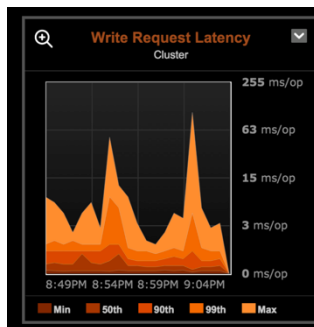
You should see something like:

```
[ubuntu@ip-172-31-27-235:~$ cqlsh 172.31.10.202
Connected to oxclo20-cassandra at 172.31.10.202:9042.
[cqlsh 5.0.1 | Cassandra 2.1.11 | CQL spec 3.2.1 | Native protocol v3]
Use HELP for help.
cqlsh> █
```

26. Now let's try cassandra-stress again:

```
cassandra-stress write n=10000 -node ip-address
```

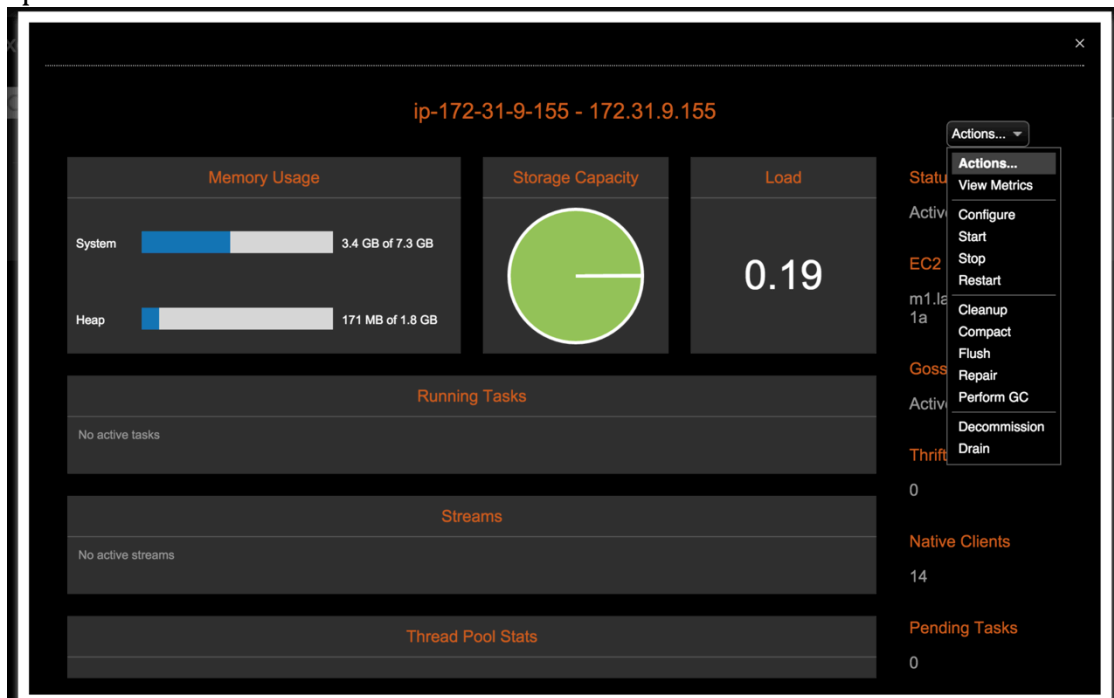
27. You will see some log showing that the stress tool is now connected to three different servers. You can check on the cluster from the opscenter. Have a good browse around. You can see nice graphs of the status like this. You can also add new graphs to the dashboard.



28. Once the write test is finished, do a mixed test:
`cassandra-stress mixed n=10000 -node ip-address`

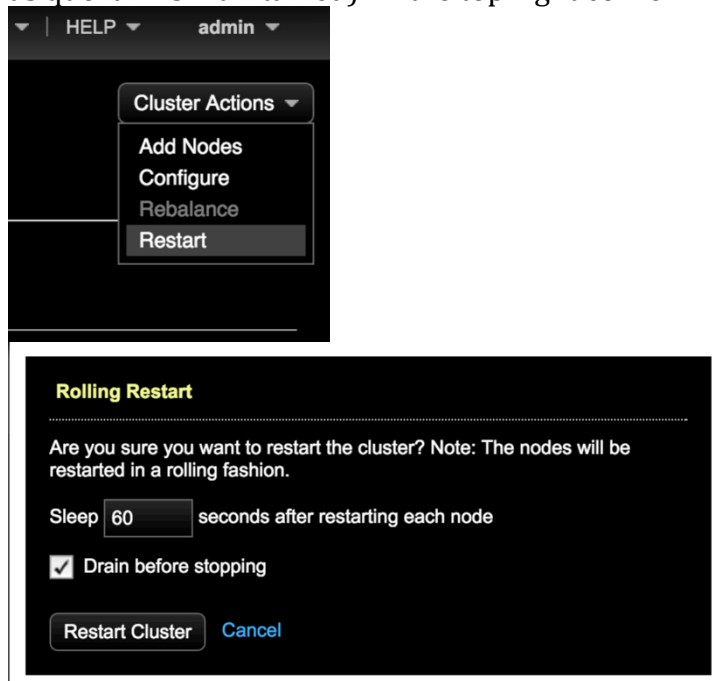
Wait a while and try to understand some of the numbers. What do you think of them? Leave it running until it gets to at least 32 threads before the next step.

29. While the mixed test is running, you can go to the opscenter. Click on the nodes icon and then click on part of the ring. You will see a panel pop up specific to that node.



30. Try stopping the node.
31. Wait a bit and restart it.

32. Ideally the stress test will continue, but just a bit slower.
33. You can even restart the whole cluster while the test is running (as long as quorum is maintained). In the top right corner:



34. Recap: we have set up a cluster of Cassandra servers in the EC2 cloud and performance tested them, as well as doing various administration aspects to them.
35. Before you finish the lab, you must **terminate** all 4 running instances that this lab has created. These m3.large servers are quite expensive to leave running!
36. Congratulations.