**Scalable multi-node Cassandra deployment on Kubernetes Cluster**

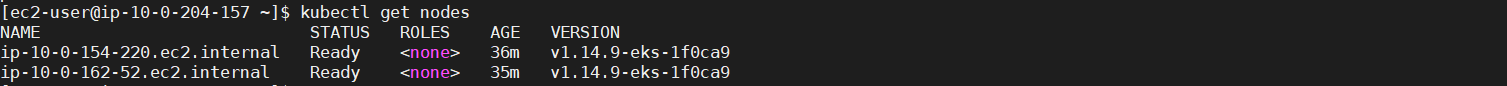
**Pre-requisites:**

* Install Git
* EKS cluster

**EKS Cluster:**

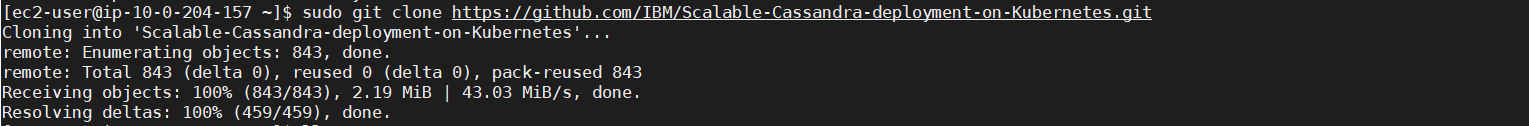
Here I taken **two node** EKS cluster. Each having **t2.xlarge** Instance type

kubectl get nodes



**Clone code from Github:**

sudo git clone <https://github.com/Naresh240/Scalable-Cassandra-deployment-on-Kubernetes.git>



cd Scalable-Cassandra-deployment-on-Kubernetes/

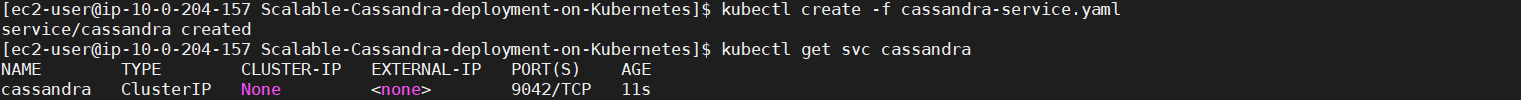
1. **Create a Cassandra Headless Service**

To allow us to do simple discovery of the cassandra seed node (which we will deploy shortly) we can create a "headless" service. We do this by specifying none for the clusterIP in the cassandra-service.yaml. This headless service allows us to use KubeDNS for the Pods to discover the IP address of the Cassandra seed.

Create the headless service using the cassandra-service.yaml file:

kubectl create -f cassandra-service.yaml

kubectl get svc cassandra



Most applications deployed to Kubernetes should be cloud native and rely on external resources for their data (or state). However since Cassandra is a database we can use Stateful sets and Persistent Volumes to ensure resiliency in our database.

1. **Create Local Volumes**

To create persistent Cassandra nodes, we need to provision Persistent Volumes. There are two ways to provision PV's: dynamically and statically.

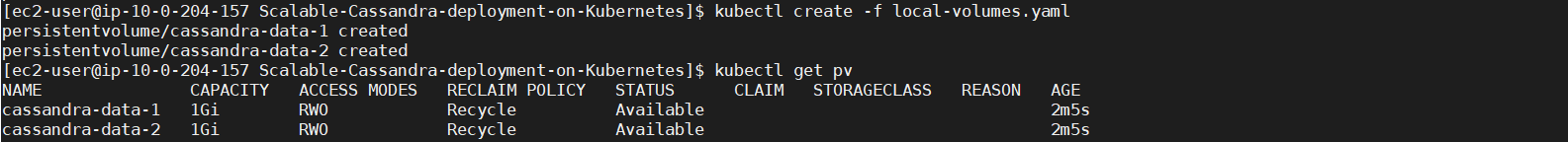
For the sake of simplicity and compatibility we will use Static provisioning where we will create volumes manually using the provided yaml files.

Note: You'll need to have the same number of Persistent Volumes as the number of your Cassandra nodes. If you are expecting to have 2 Cassandra nodes, you'll need to create 2 Persistent Volumes.

The provided local-volumes.yaml file already has 2 Persistent Volumes defined. Update the file to add more if you expect to have greater than 2 Cassandra nodes. Create the volumes:

kubectl create -f local-volumes.yaml

kubectl get pv



1. **Create a StatefulSet**

The StatefulSet is responsible for creating the Pods. It provides ordered deployment, ordered termination and unique network names. Run the following command to start a single Cassandra server:

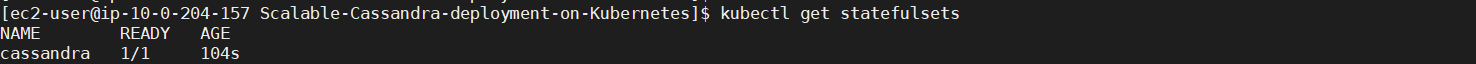
kubectl create -f cassandra-statefulset.yaml



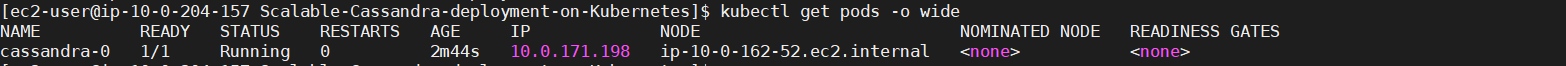
1. **Validate the StatefulSet**

You can check if your StatefulSet has deployed using the command below.

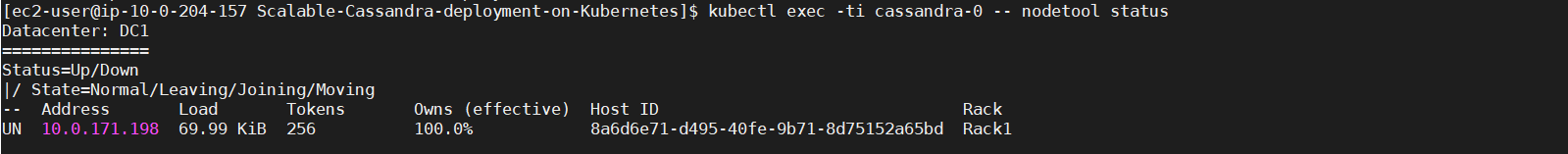
kubectl get statefulsets



If you view the list of the Pods, you should see 1 Pod running. Your Pod name should be cassandra-0 and the next pods would follow the ordinal number (cassandra-1, cassandra-2,..) Use this command to view the Pods created by the StatefulSet:



To check if the Cassandra node is up, perform a nodetool status:



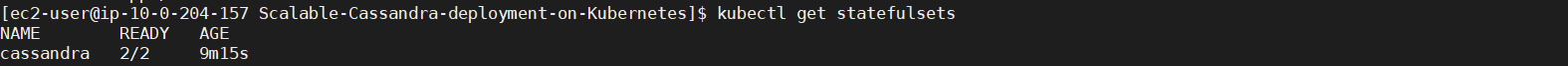
1. **Scale the StatefulSet**

To increase or decrease the size of your StatefulSet you can use the scale command:

kubectl scale --replicas=2 statefulset/cassandra



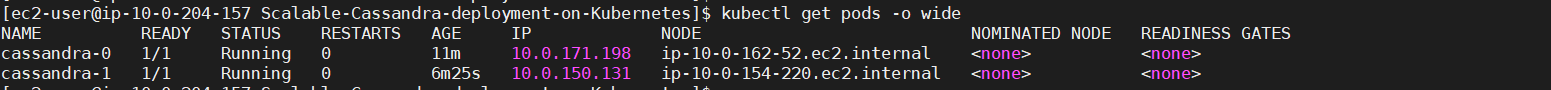
Wait a minute or two and check if it worked:



If you watch the Cassandra pods deploy, they should be created sequentially.

You can view the list of the Pods again to confirm that your Pods are up and running.

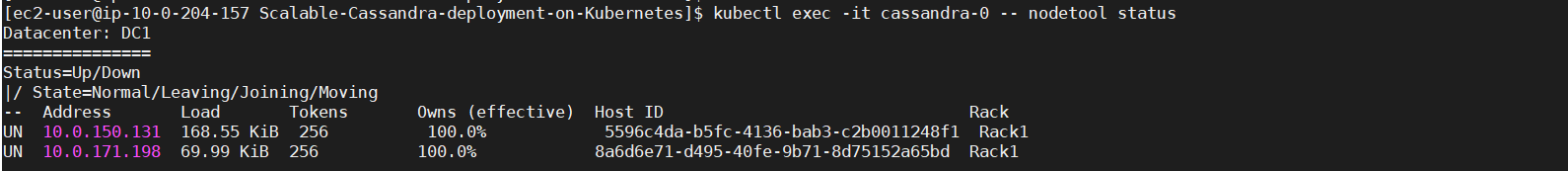
kubectl get pods -o wide



You can perform a nodetool status to check if the other cassandra nodes have joined and formed a Cassandra cluster.

Note: It can take around 5 minutes for the Cassandra database to finish its setup.

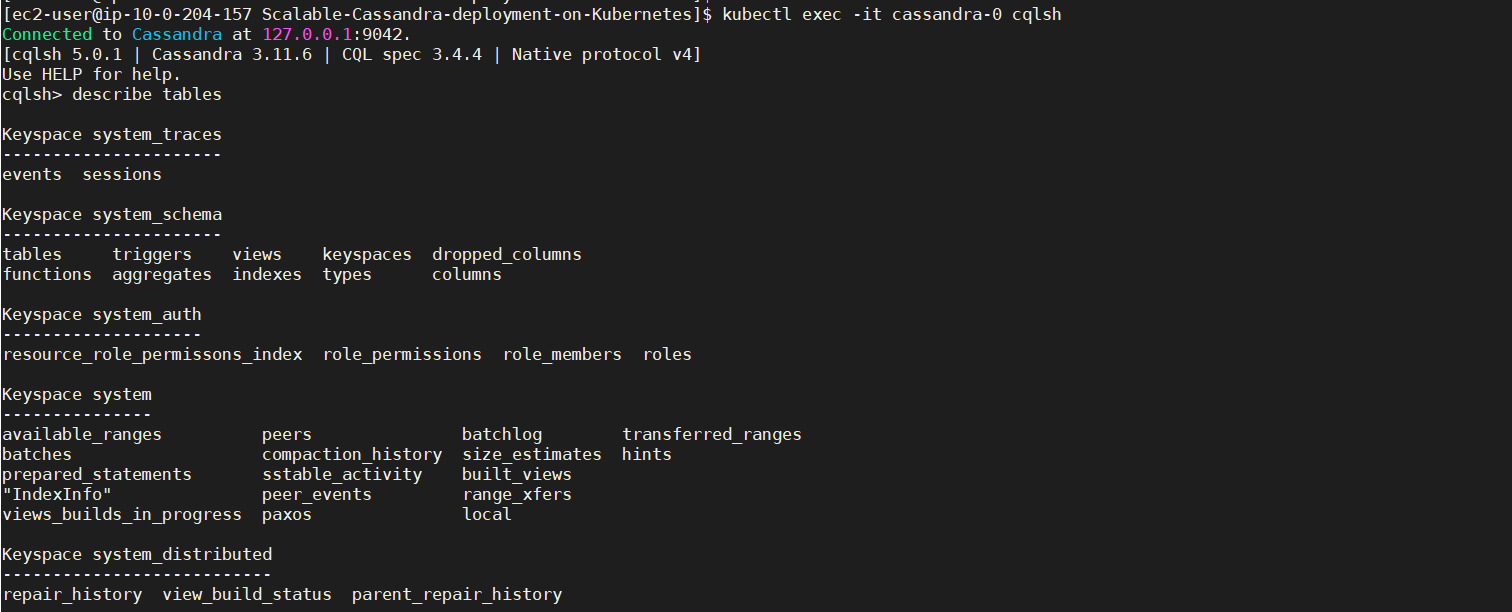
kubectl exec -it cassandra-0 -- nodetool status



You will need to wait for the status of the nodes to be Up and Normal (UN) to execute the commands in the next steps.

1. **Using CQL**

kubectl exec -it cassandra-0 cqlsh



**Troubleshooting**

If your Cassandra instance is not running properly, you may check the logs using

kubectl logs <your-pod-name>

To clean/delete your data on your Persistent Volumes, delete your PVCs using

kubectl delete pvc -l app=cassandra

If your Cassandra nodes are not joining, delete your controller/statefulset then delete your Cassandra service.

kubectl delete statefulset cassandra if you created the Cassandra StatefulSet

kubectl delete svc cassandra

To delete everything:

kubectl delete statefulset,pvc,pv,svc -l app=cassandra

