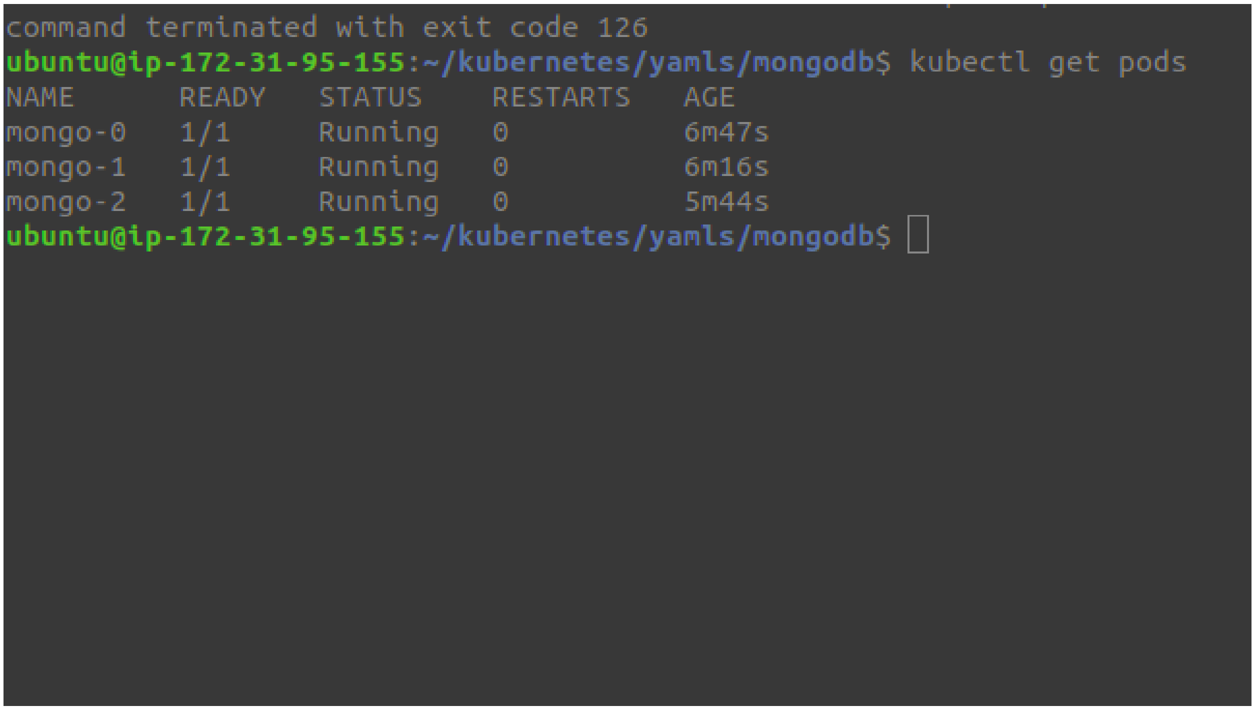
HOW TO SETUP MONGODB ON KUBERNETES:-

1) Clone the repository https://github.com/justmeandopensource/kubernetes
2) Go to cd kubernetes/yamls/mongodb
3) Kubectl create -f .
4) Now you will get a service and stateful set of mongo deployed and along with it 3 pods for mongodb
5) To check the pods execute:-

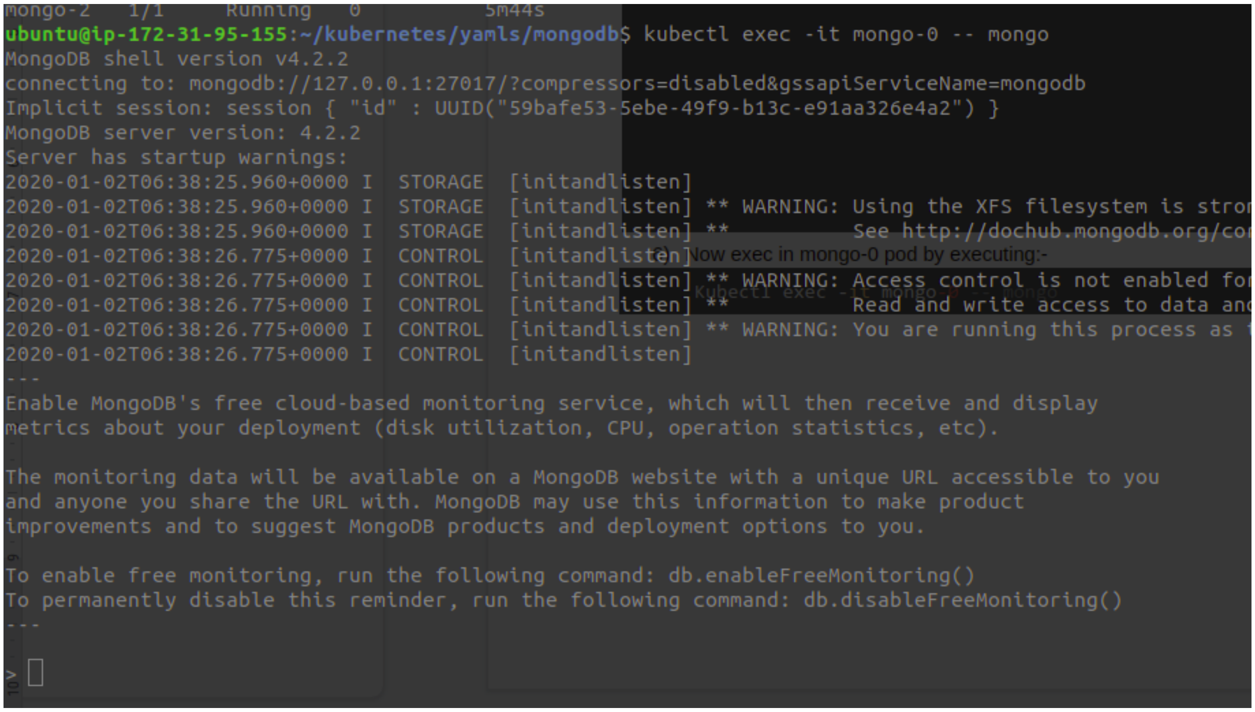
```
Kubectl get pods
```



```
command terminated with exit code 126
ubuntu@ip-172-31-95-155:~/kubernetes/yamls/mongodb$ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
mongo-0   1/1     Running   0          6m47s
mongo-1   1/1     Running   0          6m16s
mongo-2   1/1     Running   0          5m44s
ubuntu@ip-172-31-95-155:~/kubernetes/yamls/mongodb$
```

6) Now exec in mongo-0 pod by executing:-

```
      Kubectl exec -it mongo-0 -- mongo
```



```
mongo-2   1/1     Running   0          5m44s
ubuntu@ip-172-31-95-155:~/kubernetes/yamls/mongodb$ kubectl exec -it mongo-0 -- mongo
MongoDB shell version v4.2.2
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("59bafe53-5ebe-49f9-b13c-e91aa326e4a2") }
MongoDB server version: 4.2.2
Server has startup warnings:
2020-01-02T06:38:25.960+0000 I  STORAGE  [initandlisten]
2020-01-02T06:38:25.960+0000 I  STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is stron
2020-01-02T06:38:25.960+0000 I  STORAGE  [initandlisten] **          See http://dochub.mongodb.org/cor
2020-01-02T06:38:26.775+0000 I  CONTROL  [initandlisten]
2020-01-02T06:38:26.775+0000 I  CONTROL  [initandlisten] ** WARNING: Access control is not enabled for
2020-01-02T06:38:26.775+0000 I  CONTROL  [initandlisten] **          Read and write access to data and
2020-01-02T06:38:26.775+0000 I  CONTROL  [initandlisten] ** WARNING: You are running this process as t
2020-01-02T06:38:26.775+0000 I  CONTROL  [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---

> 
```

7) Follow the steps to start connecting to your replica sets

```
rs.initiate()
```

It will show you result like this

```
> rs.initiate()
{
	"info2" : "no configuration specified. Using a default configuration for the set",
	"me" : "mongo-0:27017",
	"ok" : 1,
	"$clusterTime" : {
		"clusterTime" : Timestamp(1577948138, 1),
		"signature" : {
			"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
			"keyId" : NumberLong(0)
		}
	},
	"operationTime" : Timestamp(1577948138, 1)
}
```

8) Now declare a variable and configure the host along with port number i.e. mongo-0.mongo:27017

```
rs0:OTHERS> var cfg = rs.conf()
rs0:PRIMARY> cfg.members[0].host="mongo-0.mongo:27017"
mongo-0.mongo:27017
rs0:PRIMARY> rs.reconfig(cfg)
{
	"ok" : 1,
	"$clusterTime" : {
		"clusterTime" : Timestamp(1577948225, 1),
		"signature" : {
			"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
			"keyId" : NumberLong(0)
		}
	},
	"operationTime" : Timestamp(1577948225, 1)
}
```

9) Now check the status by executing `rs.status()` and it should give you a response like this

```
members" : [
		{
			"_id" : 0,
			"name" : "mongo-0.mongo:27017",
			"ip" : "100.96.1.4",
			"health" : 1,
			"state" : 1,
			"stateStr" : "PRIMARY",
			"uptime" : 1126,
			"optime" : {
				"ts" : Timestamp(1577948225, 1),
				"t" : NumberLong(1)
			},
			"optimeDate" : ISODate("2020-01-02T06:57:05Z"),
			"syncingTo" : "",
			"syncSourceHost" : "",
                  "syncSourceId" : -1,
                  "infoMessage" : "could not find member to sync from",
                  "electionTime" : Timestamp(1577948138, 2),
                  "electionDate" : ISODate("2020-01-02T06:55:38Z"),
                  "configVersion" : 2,
                  "self" : true,
                  "lastHeartbeatMessage" : ""
            }
      ],
```

10) Now we will add another two hosts by

```
rs.add("mongo-1.mongo:27017")
rs.add("mongo-2.mongo:27017")
```

Now again check the status by **rs.status()**, in the response all the 3 replica sets should be the members for your mongodb

11) To check weather all of them are working fine run a mongo container and try to connect to your replica sets by executing:-

```
kubectl run  mongo --rm -it --image mongo -- sh
      mongo mongodb://mongo-0.mongo,mongo-1.mongo,mongo-2.mongo --eval
'rs.status()' | grep name
                  "name" : "mongo-0.mongo:27017",
                  "name" : "mongo-1.mongo:27017",
                  "name" : "mongo-2.mongo:27017",
```

When you will run the mongo command then all your replica sets should be shown in the response

12) To connect to you mongo update your service along with a cluster ip, update your service yaml that you have pulled fro git hub by commenting the cluster ip line and update your service by **kubectl apply -f <service-yaml-name>**, if you are not seeing the cluster ip with your service then try to delete the old service and create new one.

13) For checking your connectivity with mongo, deploy an ubuntu shell in kubernetes by

```
kubectl apply -f https://k8s.io/examples/application/shell-demo.yaml
```

14) Now exec into the pod and telnet to your mongo service ip along with the port number, if it is not able to connect then there is something wrong in your configuration