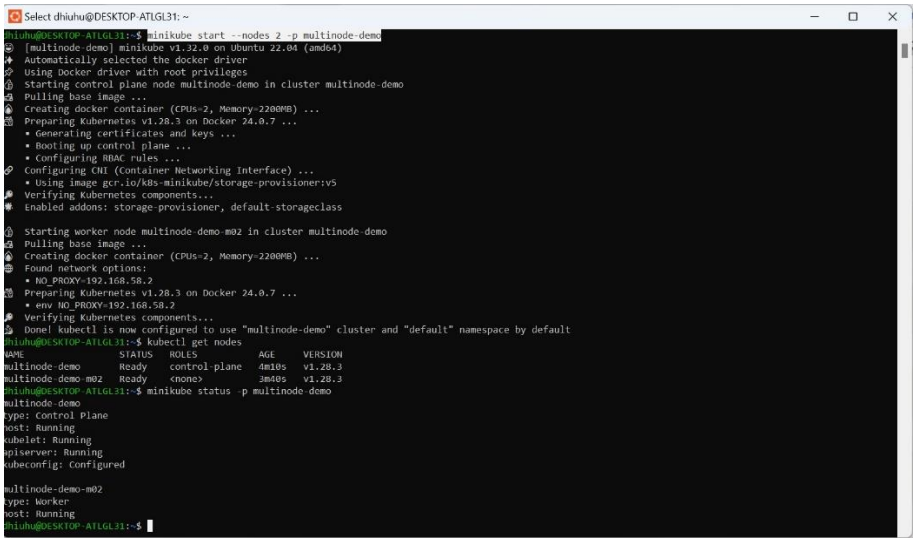
Week 2 Day 2 – Deployments

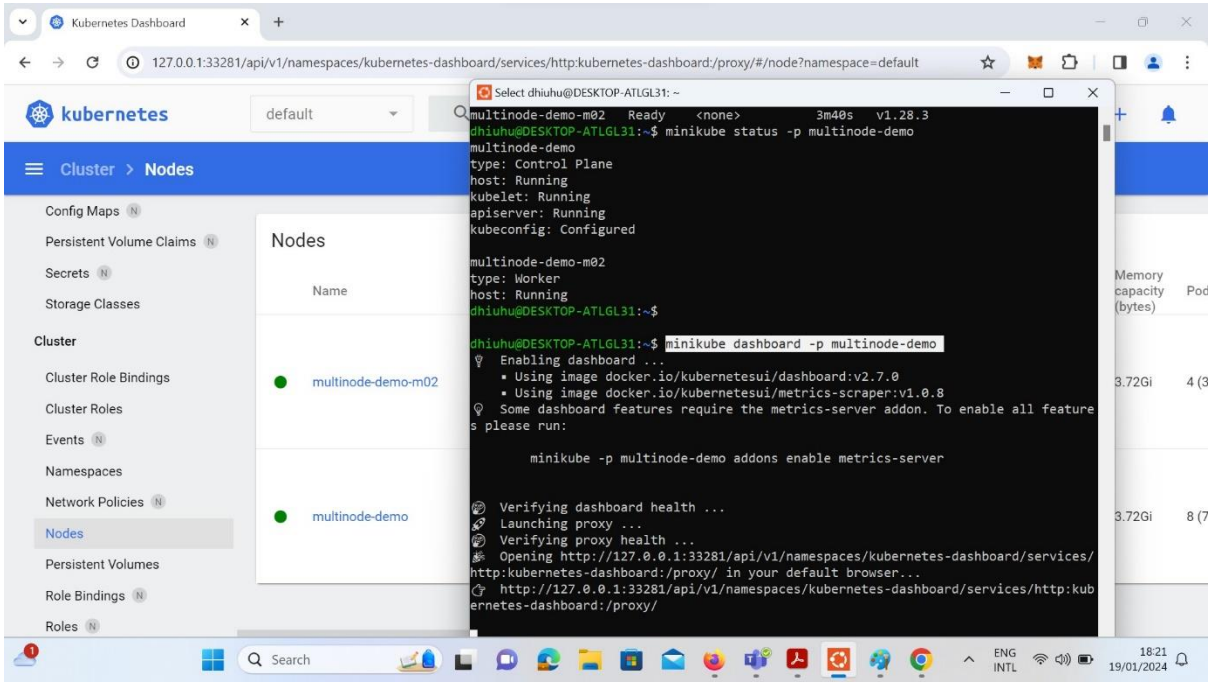
DAVID HIUHU

101509799

Deployment Demo

- **Minikube with two nodes**
    - **minikube start --nodes 2 -p multinode-demo**





```
Select dhiuhu@DESKTOP-ATLGL31: ~
  Some dashboard features require the metrics-server addon. To enable all features please run:

        minikube -p multinode-demo addons enable metrics-server


  Verifying dashboard health ...
  Launching proxy ...
  Verifying proxy health ...
  Opening http://127.0.0.1:33281/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in your default browser...
  http://127.0.0.1:33281/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/

^C
dhiuhu@DESKTOP-ATLGL31:~$ minikube stop -p multinode-demo
  Stopping node "multinode-demo"  ...
  Powering off "multinode-demo" via SSH ...
  Stopping node "multinode-demo-m02"  ...
  Powering off "multinode-demo-m02" via SSH ...
  2 nodes stopped.
dhiuhu@DESKTOP-ATLGL31:~$ minikube delete --all
  Deleting "minikube" in docker ...
  Removing /home/dhiuhu/.minikube/machines/minikube ...
  Removed all traces of the "minikube" cluster.
  Deleting "multinode-demo" in docker ...
  Removing /home/dhiuhu/.minikube/machines/multinode-demo ...
  Removing /home/dhiuhu/.minikube/machines/multinode-demo-m02 ...
  Removed all traces of the "multinode-demo" cluster.
  Successfully deleted all profiles
dhiuhu@DESKTOP-ATLGL31:~$ kubectl get nodes
E0119 18:24:06.588508   45632 memcache.go:265] couldn't get current server API group list: the server could not find the requested resource
E0119 18:24:06.591674   45632 memcache.go:265] couldn't get current server API group list: the server could not find the requested resource
E0119 18:24:06.595026   45632 memcache.go:265] couldn't get current server API group list: the server could not find the requested resource
E0119 18:24:06.597728   45632 memcache.go:265] couldn't get current server API group list: the server could not find the requested resource
E0119 18:24:06.599950   45632 memcache.go:265] couldn't get current server API group list: the server could not find the requested resource
Error from server (NotFound): the server could not find the requested resource
dhiuhu@DESKTOP-ATLGL31:~$
```

```
dhiuhu@DESKTOP-ATLGL31: ~
dhiuhu@DESKTOP-ATLGL31:~$ minikube delete --all
  Deleting "minikube" in docker ...
  Removing /home/dhiuhu/.minikube/machines/minikube ...
  Removed all traces of the "minikube" cluster.
  Deleting "multinode-demo" in docker ...
  Removing /home/dhiuhu/.minikube/machines/multinode-demo ...
  Removing /home/dhiuhu/.minikube/machines/multinode-demo-m02 ...
  Removed all traces of the "multinode-demo" cluster.
  Successfully deleted all profiles
dhiuhu@DESKTOP-ATLGL31:~$ kubectl get nodes
E0119 18:24:06.588508   45632 memcache.go:265] couldn't get current server API group list: the server could not find the requested resource
E0119 18:24:06.591674   45632 memcache.go:265] couldn't get current server API group list: the server could not find the requested resource
E0119 18:24:06.595026   45632 memcache.go:265] couldn't get current server API group list: the server could not find the requested resource
E0119 18:24:06.597728   45632 memcache.go:265] couldn't get current server API group list: the server could not find the requested resource
E0119 18:24:06.599950   45632 memcache.go:265] couldn't get current server API group list: the server could not find the requested resource
Error from server (NotFound): the server could not find the requested resource
dhiuhu@DESKTOP-ATLGL31:~$ clear
```

ReplicaSet Demo

- **Replica Set YAML.**

```
dhiuhu@DESKTOP-ATLGL31: ~
  GNU nano 6.2                                   replicaset.yaml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: nginx
  labels:
    app: nginx
    tier: lb
spec:
  replicas: 3
  selector:
    matchLabels:
      tier: lb
  template:
  metadata:
    labels:
      tier: lb
spec:
  containers:
    name: nginx-replicaset
      image: nginx

^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo       M-A Set Mark   M-] To Bracket
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line  M-E Redo       M-6 Copy       ^Q Where Was
```

- **Apply ReplicaSet YAML.**

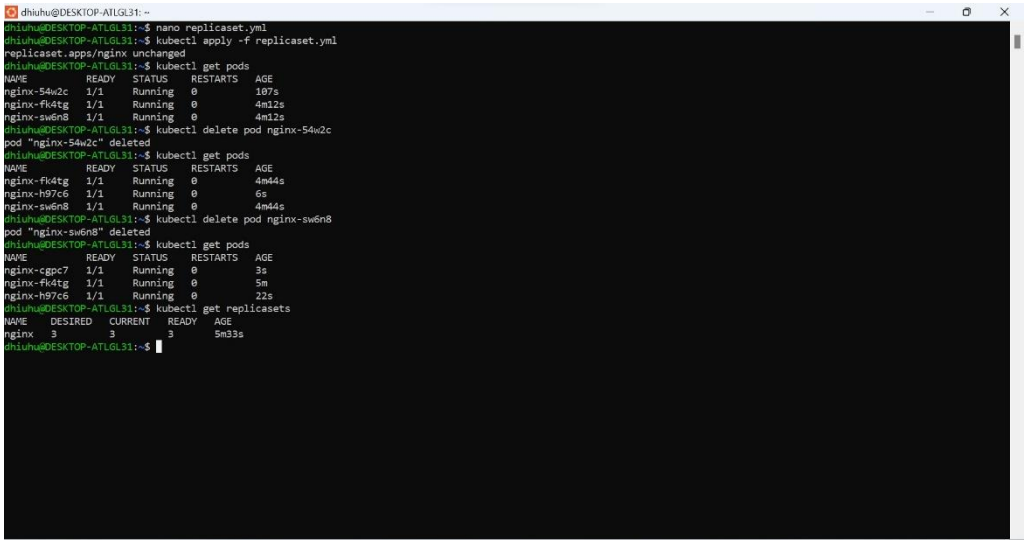
kubectl apply -f replicaset.yaml

Delete a pod in the ReplicaSet

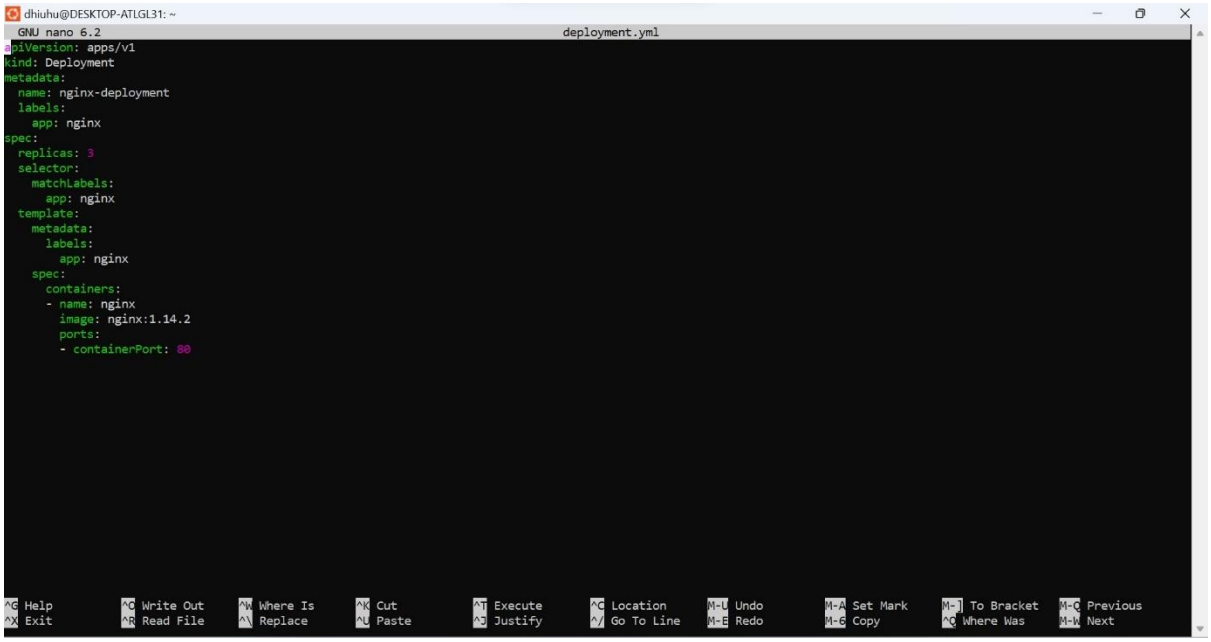
kubectl delete pod nginx-<code>

kubectl get pods

kubectl get replicasets



Deployment YAML with ReplicaSet



kubectl get deployments

kubectl rollout status deployment nginx

```
dhiuhu@DESKTOP-ATLGL31: ~
dhiuhu@DESKTOP-ATLGL31:~$ nano deployment.yml
dhiuhu@DESKTOP-ATLGL31:~$ kubectl apply -f deployment.yml
deployment.apps/nginx-deployment unchanged
dhiuhu@DESKTOP-ATLGL31:~$ kubectl get deployments
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   3/3     3            3           2m29s
dhiuhu@DESKTOP-ATLGL31:~$ kubectl rollout status deployment nginx-deployment
deployment "nginx-deployment" successfully rolled out
dhiuhu@DESKTOP-ATLGL31:~$
```

StatefulSet YAML

```
Select dhiuhu@DESKTOP-ATLGL31: ~
  GNU nano 6.2                                   StatefulSet.yml
apiVersion: v1
kind: Service
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  ports:
  - port: 80
  name: web
  clusterIP: None
  selector:
    app: nginx
---
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: web
spec:
  selector:
    matchLabels:
      app: nginx # has to match .spec.template.metadata.labels
  serviceName: "nginx"
  replicas: 3 # by default is 1
  template:
    metadata:
      labels:
        app: nginx # has to match .spec.selector.matchLabels
    spec:
      terminationGracePeriodSeconds: 10
      containers:
      - name: nginx
        image: k8s.gcr.io/nginx-slim:0.8
        ports:
        - containerPort: 80
          name: web
                                                 [ Wrote 48 lines ]
^G Help      ^O Write Out   ^W Where Is    ^K Cut       ^T Execute     ^C Location    M-U Undo      M-A Set Mark   M-] To Bracket  M-Q Previous
^X Exit      ^R Read File   ^\ Replace     ^U Paste     ^J Justify     ^/ Go To Line  M-E Redo      M-6 Copy       ^Q Where Was    M-W Next
```

```
dhiuhu@DESKTOP-ATLGL31: ~
  GNU nano 6.2                                StatefulSet.yml
apiVersion: v1
kind: Service
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  ports:
  - port: 80
    name: web
  clusterIP: None
  selector:
    app: nginx
---
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: web
spec:
  selector:
    matchLabels:
      app: nginx # has to match .spec.template.metadata.labels
  serviceName: "nginx"
  replicas: 3 # by default is 1
  template:
    metadata:
      labels:
        app: nginx # has to match .spec.selector.matchLabels
    spec:
      terminationGracePeriodSeconds: 10
      containers:
      - name: nginx
        image: k8s.gcr.io/nginx-slim:0.8
```

kubectl get statefulsets

```
dhiuhu@DESKTOP-ATLGL31: ~
dhiuhu@DESKTOP-ATLGL31:~$ nano StatefulSet.yml
dhiuhu@DESKTOP-ATLGL31:~$ kubectl apply -f StatefulSet.yml
service/nginx unchanged
statefulset.apps/web configured
dhiuhu@DESKTOP-ATLGL31:~$ kubectl get statefulsets
NAME    READY   AGE
web     0/3     14m
dhiuhu@DESKTOP-ATLGL31:~$
```

DaemonSet YAML.

```
dhiuhu@DESKTOP-ATLGL31: ~
  GNU nano 6.2                                          DaemonSet.yml
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: nginx
spec:
  selector:
    matchLabels:
      name: nginx-lb
  template:
    metadata:
      labels:
        name: nginx-lb
    spec:
      containers:
      - name: nginx
        image: nginx

                                                  [ Wrote 16 lines ]
^G Help      ^O Write Out   ^W Where Is    ^K Cut       ^T Execute     ^C Location    M-U Und
^X Exit      ^R Read File   ^\ Replace     ^U Paste     ^J Justify     ^/ Go To Line  M-E Red
```

Deployment example with resource limits.

```
dhiuhu@DESKTOP-ATLGL31: ~
  GNU nano 6.2                                          DeploymentHealthCheck.yml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:latest
        ports:
        - containerPort: 80
        livenessProbe:
          httpGet:
            path: / # The path to check for the liveness probe
            port: 80 # The port to check on
          initialDelaySeconds: 15 # Wait this many seconds before starting the probe
          periodSeconds: 10 # Check the probe every 10 seconds
        readinessProbe:
          httpGet:
            path: / # The path to check for the readiness probe
            port: 80 # The port to check on
          initialDelaySeconds: 5 # Wait this many seconds before starting the probe
          periodSeconds: 5 # Check the probe every 5 seconds

                                                  [ Wrote 31 lines ]
```

ENG INTL 21:19 19/01/2024

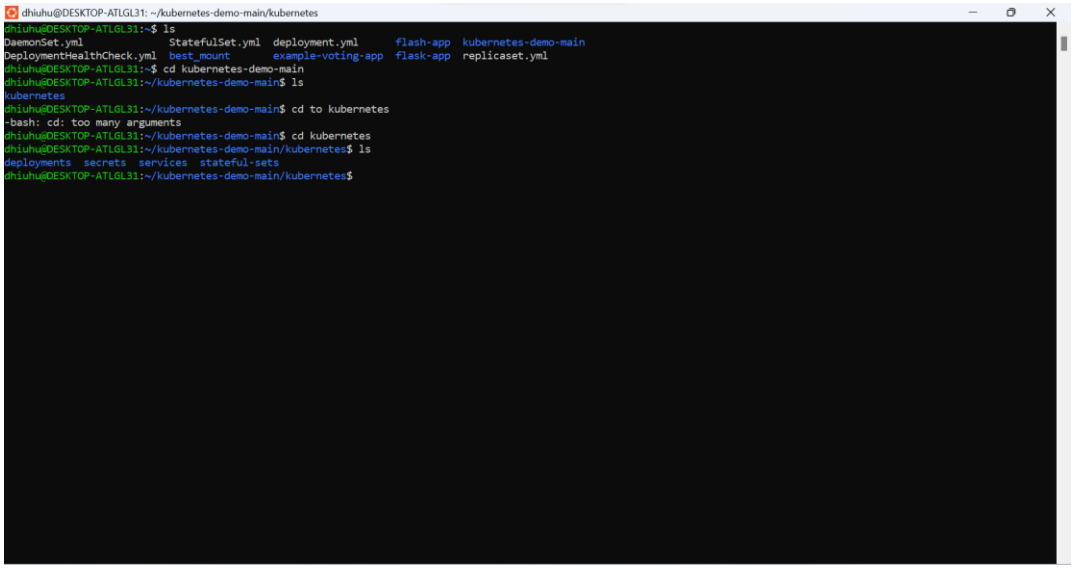
Deployment with health checks.

```
dhiuhu@DESKTOP-ATLGL31: ~
  GNU nano 6.2                                    DeploymentHealthCheck.yml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:latest
        ports:
        - containerPort: 80
        livenessProbe:
          httpGet:
            path: / # The path to check for the liveness probe
            port: 80 # The port to check on
          initialDelaySeconds: 15 # Wait this many seconds before starting the probe
          periodSeconds: 10 # Check the probe every 10 seconds
        readinessProbe:
          httpGet:
            path: / # The path to check for the readiness probe
            port: 80 # The port to check on
          initialDelaySeconds: 5 # Wait this many seconds before starting the probe
          periodSeconds: 5 # Check the probe every 5 seconds
```

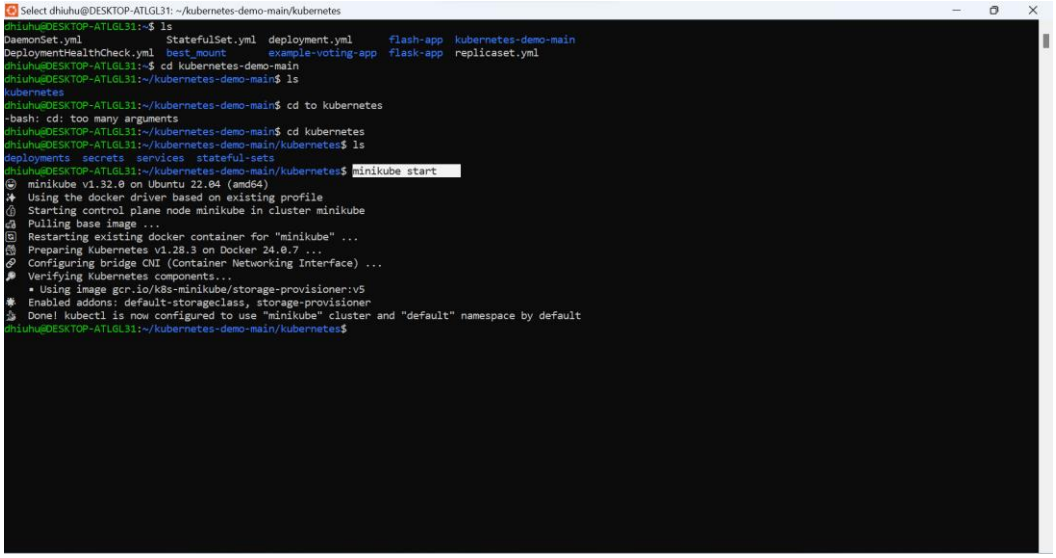
```
dhiuhu@DESKTOP-ATLGL31: ~
dhiuhu@DESKTOP-ATLGL31:~$ nano DeploymentHealthCheck.yml
dhiuhu@DESKTOP-ATLGL31:~$ kubectl apply -f  DeploymentHealthCheck.yml
deployment.apps/nginx-deployment configured
dhiuhu@DESKTOP-ATLGL31:~$ kubectl get deployments
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   3/3     3            3           46m
dhiuhu@DESKTOP-ATLGL31:~$
```

Deploy MERN App To K8s (Minikube)

1. **Clone repo on your local machine. Change directory to kubernetes**



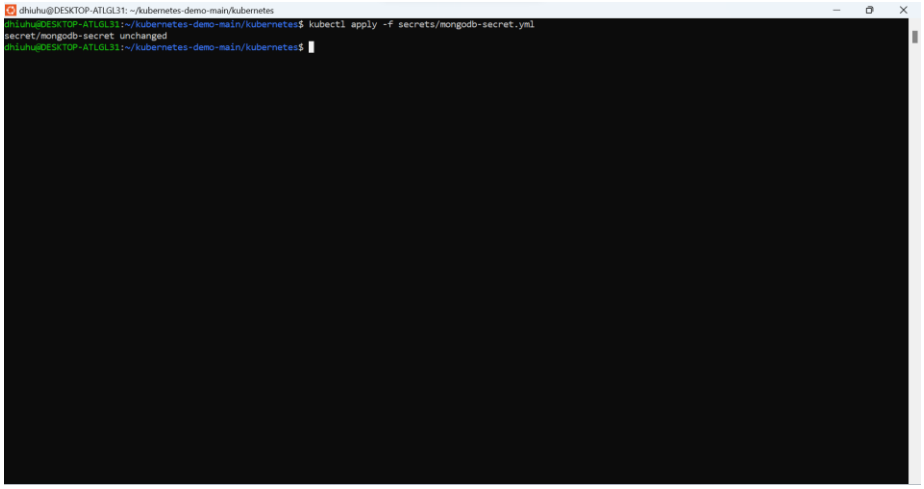
2. **Make sure you have Minikube installed and running properly.**
   a. **Start Minikube**



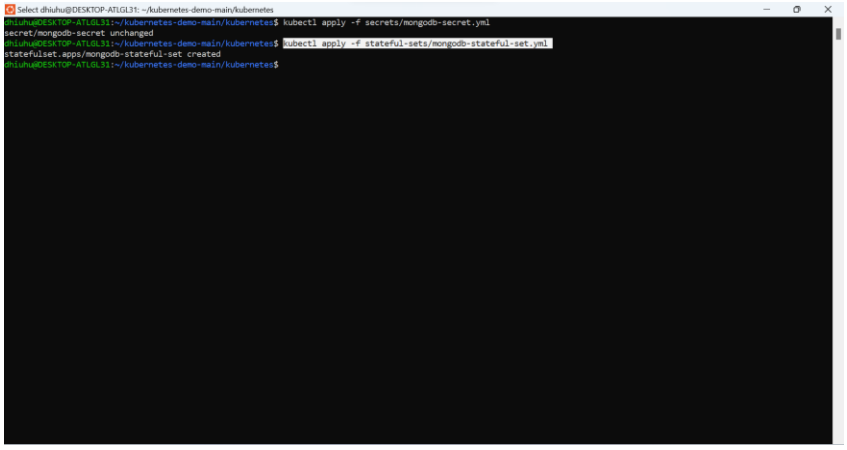
3. **You can use a Kubernetes IDE like "Lens" to connect to your Minikube cluster. Makes it easier for monitoring changes. GIT hub for LENs did not work** ttps://github.com/lensapp/lens/releases/download/v4.1.3/Lens-4.1.3.AppImage

4. **create the secret. (Holds mongodb admin username and password)**

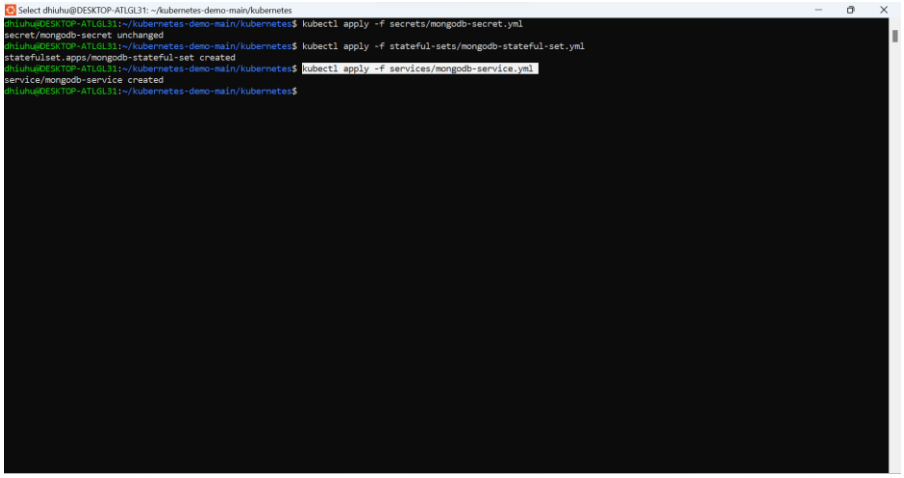
    a. **kubectl apply -f secrets/mongodb-secret.yml**



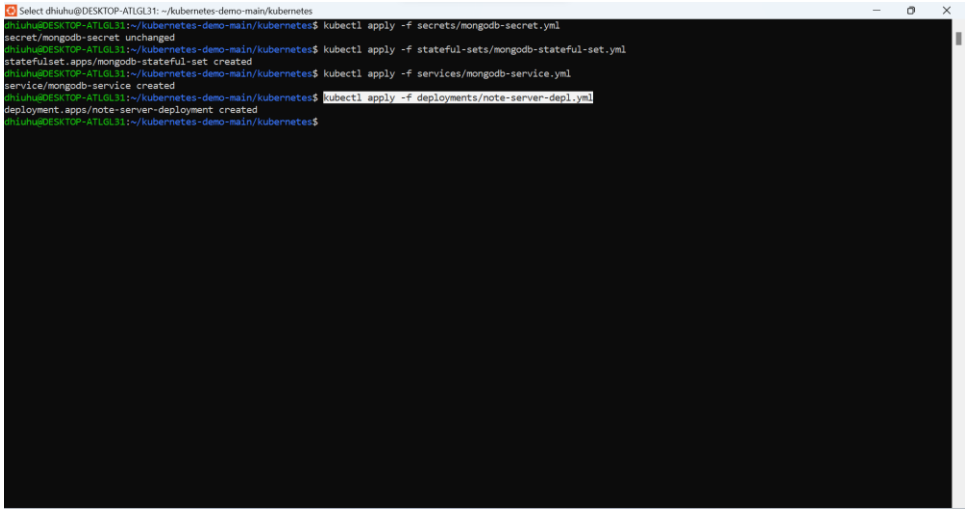
5. **create the mongodb stateful set. It will create 2 replicas.**
    a. **kubectl apply -f stateful-sets/mongodb-stateful-set.yml**



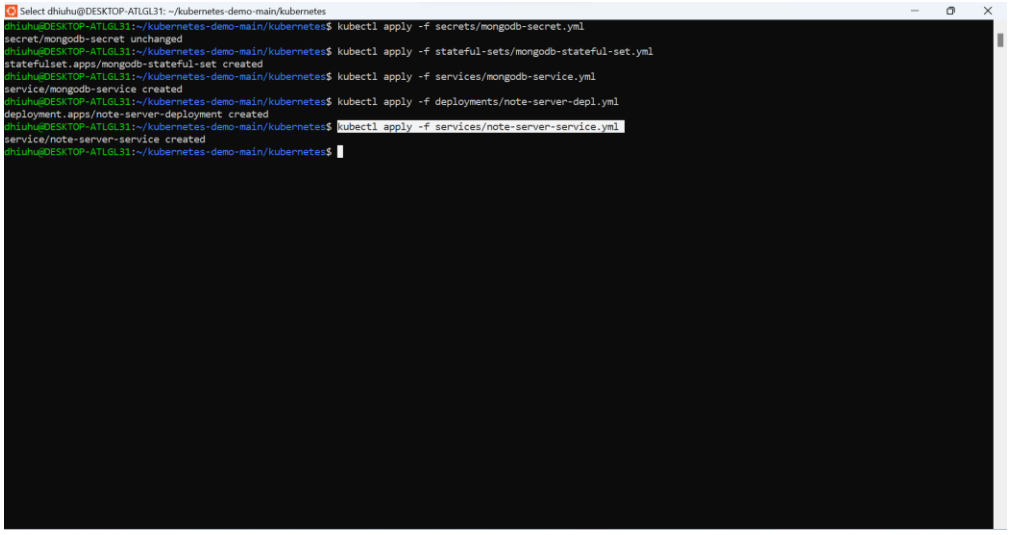
6. **create the mongodb internal service so that our server can later access the mongodb pods using it.**
    a. **kubectl apply -f services/mongodb-service.yml**



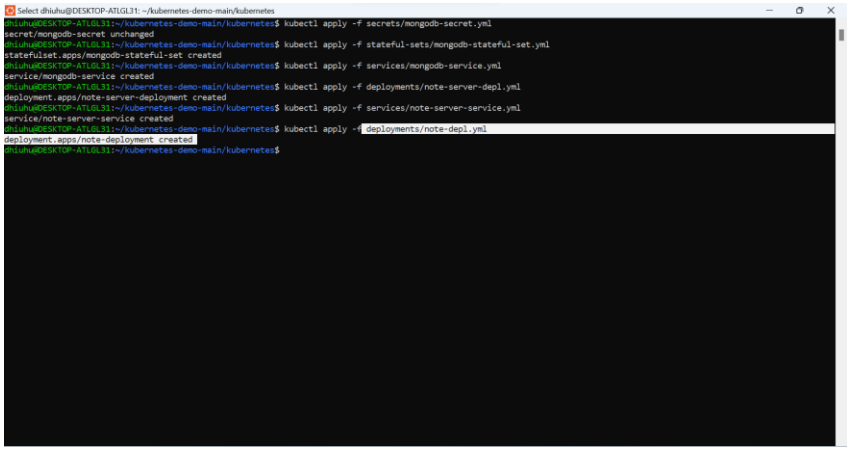
7. **deploy the server app by running:**
    a. **kubectl apply -f deployments/note-server-depl.yml**



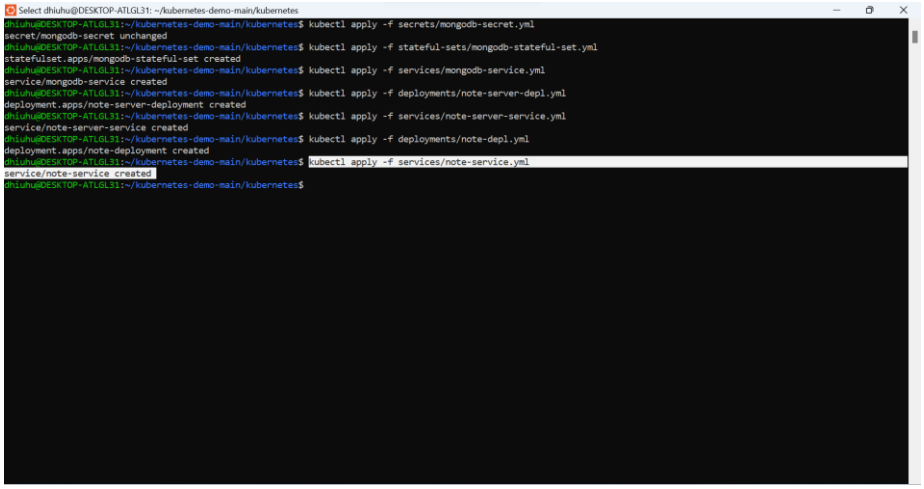
8. **create the server internal service for communication between the frontend and the backend.**
    a. **kubectl apply -f services/note-server-service.yml**



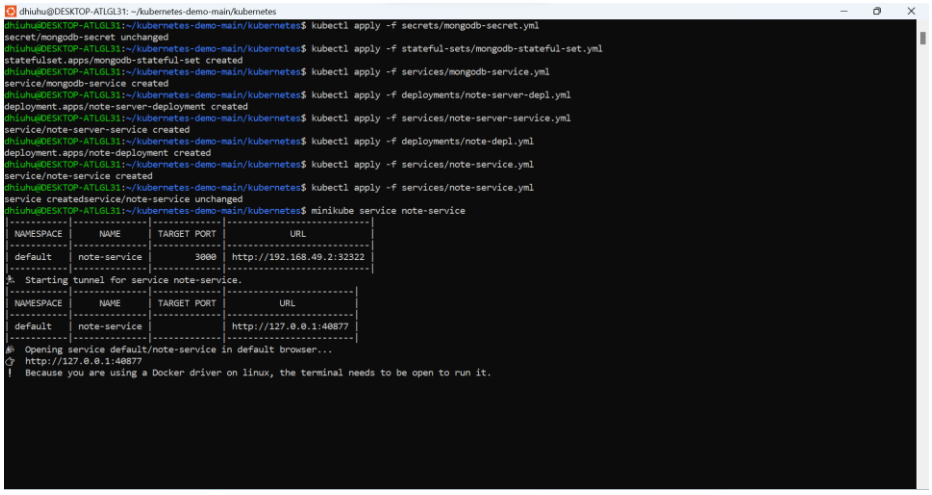
9. **Deploy frontend:**
    a. **kubectl apply -f deployments/note-depl.yml**



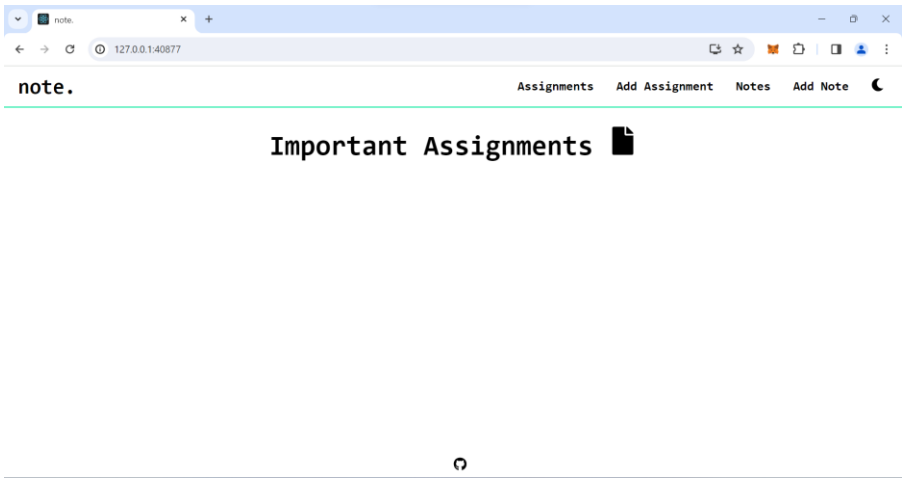
10. **Create external service for the frontend app (Load Balancer) by running:**
    a. **kubectl apply -f services/note-service.yml**



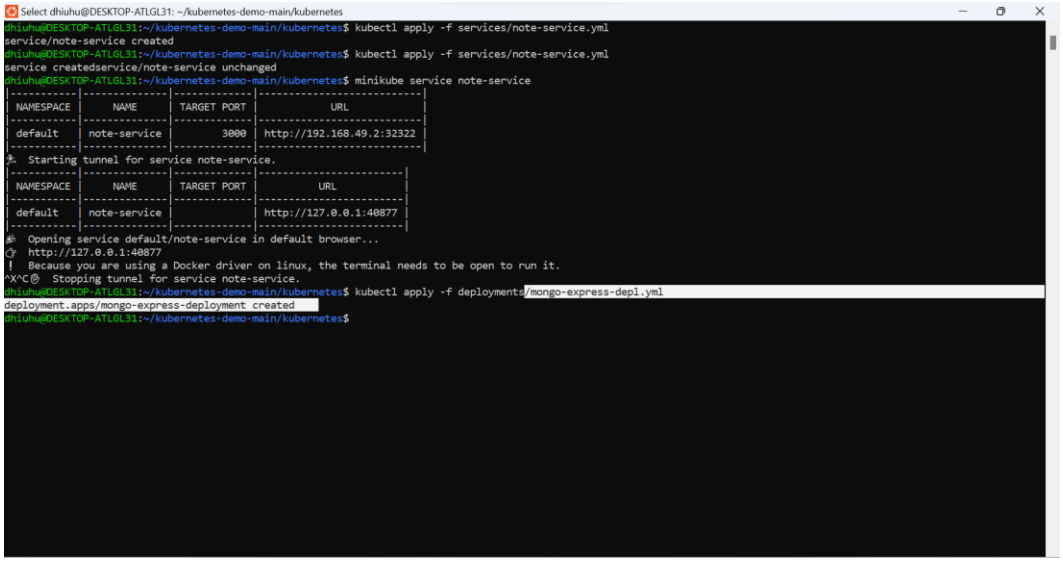
11. **To get an external IP assigned to the frontend deployment you can run:**
    a. **minikube service note-service**



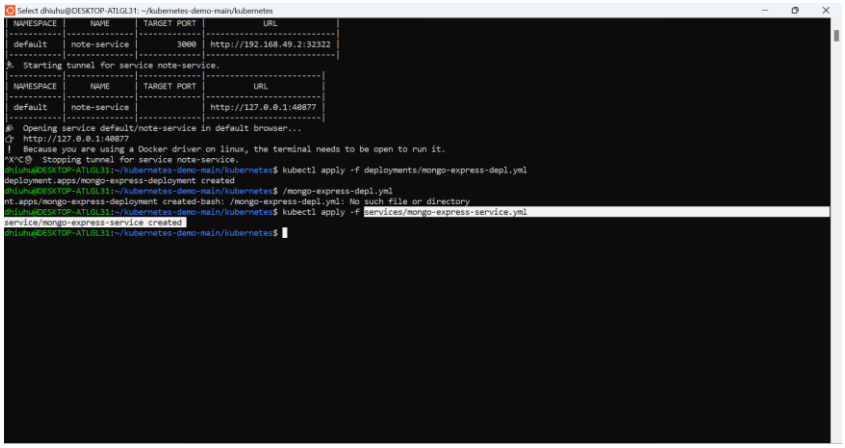
12. **Open the IP adddress in browser and you will be able to see the app running!**
    a. **default | note-service | | http://127.0.0.1:40877**



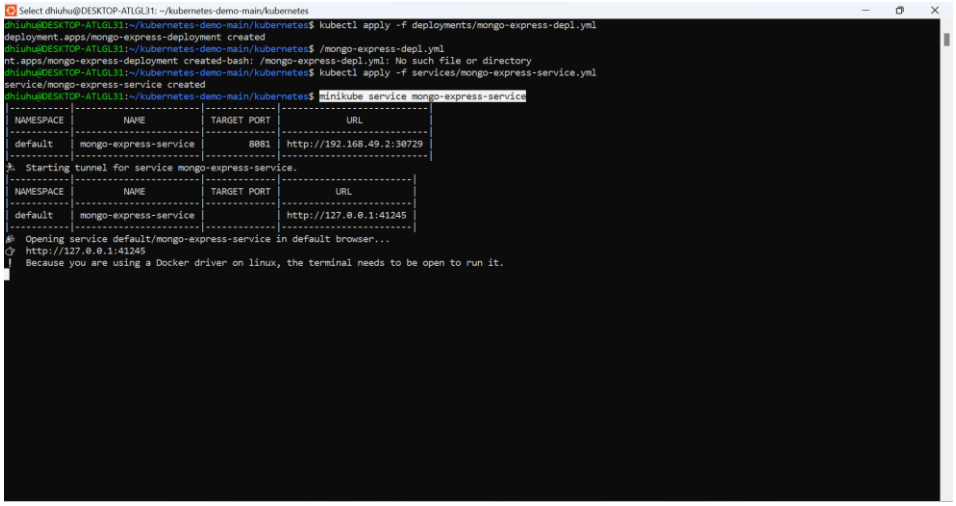
13. **If you want to run a database GUI like mongo-express, you can simply run:**
    a. **kubectl apply -f deployments/mongo-express-depl.yml**



14. **Later create mongo-express external service for acccessing the GUI in browser:**
    a. **kubectl apply -f services/mongo-express-service.yml**
    b. **minikube service mongo-express-service**



c. **Mongo Express Service**



d. **the IP adddress in browser and you will be able to see the app running!**

e. **default | mongo-express-service | | http://127.0.0.1:41245**