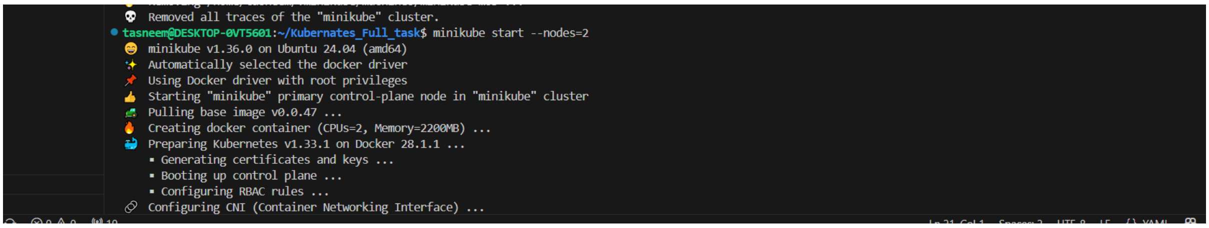
Kubernates Full task:

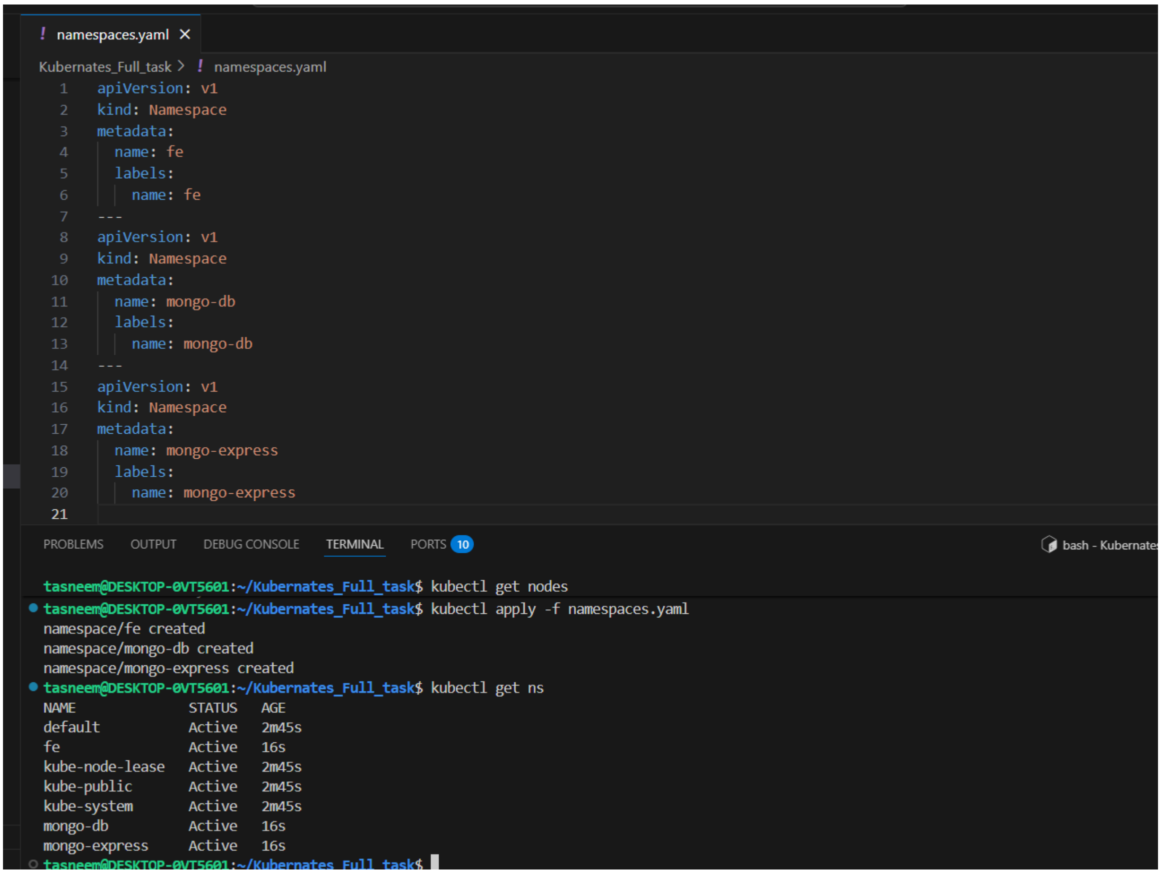
1- start your minikube with 2 nodes



```
💀 Removed all traces of the "minikube" cluster.
● tasneem@DESKTOP-0VT5601:~/Kubernates_Full_task$ minikube start --nodes=2
😄 minikube v1.36.0 on Ubuntu 24.04 (amd64)
✨ Automatically selected the docker driver
📌 Using Docker driver with root privileges
👍 Starting "minikube" primary control-plane node in "minikube" cluster
🚜 Pulling base image v0.0.47 ...
🔥 Creating docker container (CPUs=2, Memory=2200MB) ...
🐳 Preparing Kubernetes v1.33.1 on Docker 28.1.1 ...
    ▪ Generating certificates and keys ...
    ▪ Booting up control plane ...
    ▪ Configuring RBAC rules ...
🔗 Configuring CNI (Container Networking Interface) ...
```

2- create 3 namespaces

- FE
- mongo-db
- mongo-express

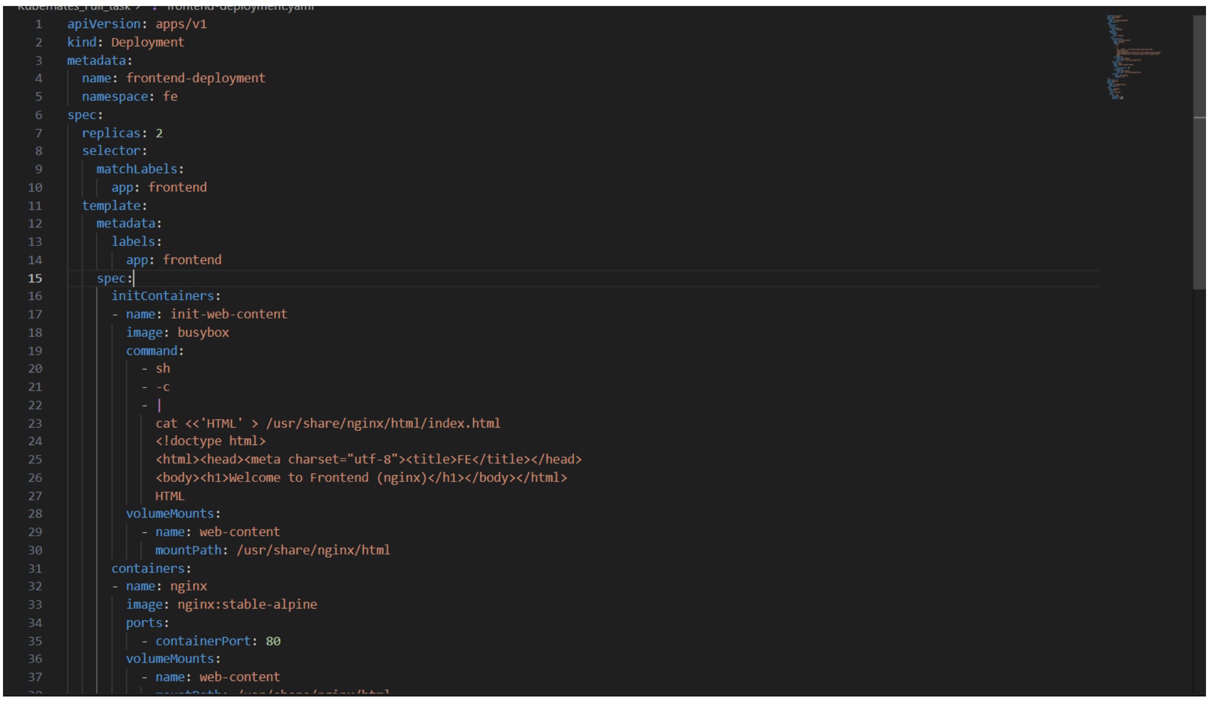


```yaml
apiVersion: v1
kind: Namespace
metadata:
  name: fe
  labels:
    name: fe
---
apiVersion: v1
kind: Namespace
metadata:
  name: mongo-db
  labels:
    name: mongo-db
---
apiVersion: v1
kind: Namespace
metadata:
  name: mongo-express
  labels:
    name: mongo-express

```

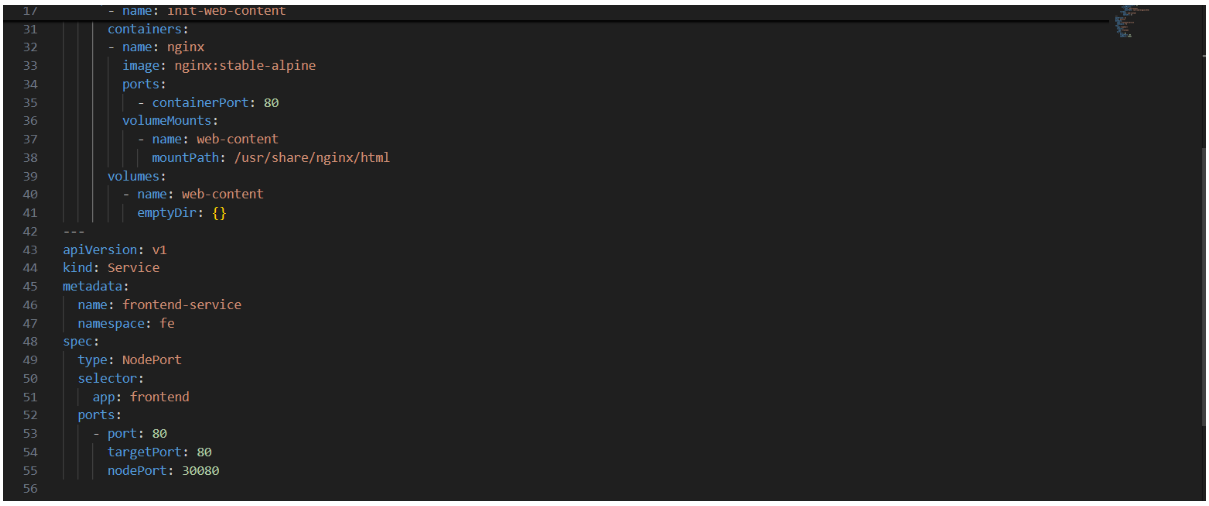
```
tasneem@DESKTOP-0VT5601:~/Kubernates_Full_task$ kubectl get nodes
● tasneem@DESKTOP-0VT5601:~/Kubernates_Full_task$ kubectl apply -f namespaces.yaml
namespace/fe created
namespace/mongo-db created
namespace/mongo-express created
● tasneem@DESKTOP-0VT5601:~/Kubernates_Full_task$ kubectl get ns
NAME              STATUS   AGE
default           Active   2m45s
fe                Active   16s
kube-node-lease   Active   2m45s
kube-public       Active   2m45s
kube-system       Active   2m45s
mongo-db          Active   16s
mongo-express     Active   16s
○ tasneem@DESKTOP-0VT5601:~/Kubernates_Full_task$
```

3- Deployments&services: A- simple web frontend application in the FE-namespace namespace with 2 riplca - and Use an emptyDir Volume to store the web content and mount it to /usr/share/nginx/html in the POD - Create a NodePort Service named frontend-service to expose the Nginx application externally on port 80.



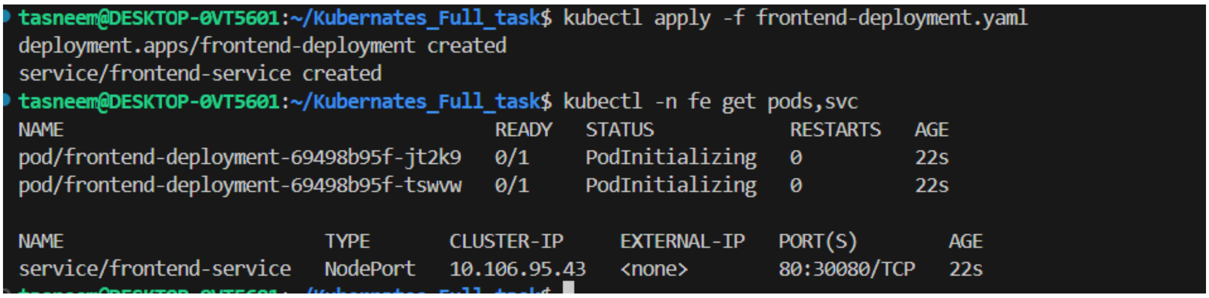
kubernates_full_task > ! frontend-deployment.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: frontend-deployment
  namespace: fe
spec:
  replicas: 2
  selector:
    matchLabels:
      app: frontend
  template:
    metadata:
      labels:
        app: frontend
    spec:
      initContainers:
      - name: init-web-content
        image: busybox
        command:
          - sh
          - -c
          - |
            cat <<'HTML' > /usr/share/nginx/html/index.html
            <!doctype html>
            <html><head><meta charset="utf-8"><title>FE</title></head>
            <body><h1>Welcome to Frontend (nginx)</h1></body></html>
            HTML
        volumeMounts:
          - name: web-content
            mountPath: /usr/share/nginx/html
      containers:
      - name: nginx
        image: nginx:stable-alpine
        ports:
          - containerPort: 80
        volumeMounts:
          - name: web-content
```



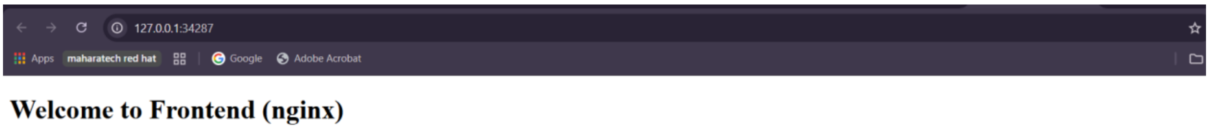
```yaml
      - name: init-web-content
      containers:
      - name: nginx
        image: nginx:stable-alpine
        ports:
          - containerPort: 80
        volumeMounts:
          - name: web-content
            mountPath: /usr/share/nginx/html
      volumes:
        - name: web-content
          emptyDir: {}
---
apiVersion: v1
kind: Service
metadata:
  name: frontend-service
  namespace: fe
spec:
  type: NodePort
  selector:
    app: frontend
  ports:
    - port: 80
      targetPort: 80
      nodePort: 30080

```



```
tasneem@DESKTOP-0VT5601:~/Kubernates_Full_task$ kubectl apply -f frontend-deployment.yaml
deployment.apps/frontend-deployment created
service/frontend-service created
tasneem@DESKTOP-0VT5601:~/Kubernates_Full_task$ kubectl -n fe get pods,svc
NAME                                   READY   STATUS            RESTARTS   AGE
pod/frontend-deployment-69498b95f-jt2k9   0/1     PodInitializing   0          22s
pod/frontend-deployment-69498b95f-tswvw   0/1     PodInitializing   0          22s

NAME                       TYPE       CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
service/frontend-service   NodePort   10.106.95.43   <none>        80:30080/TCP   22s
tasneem@DESKTOP-0VT5601:~/Kubernates_Full_task$
```



Welcome to Frontend (nginx)

b - Deploy a MongoDB database in the mongo-db namespace.

Use a Deployment named mongodb-deployment with:

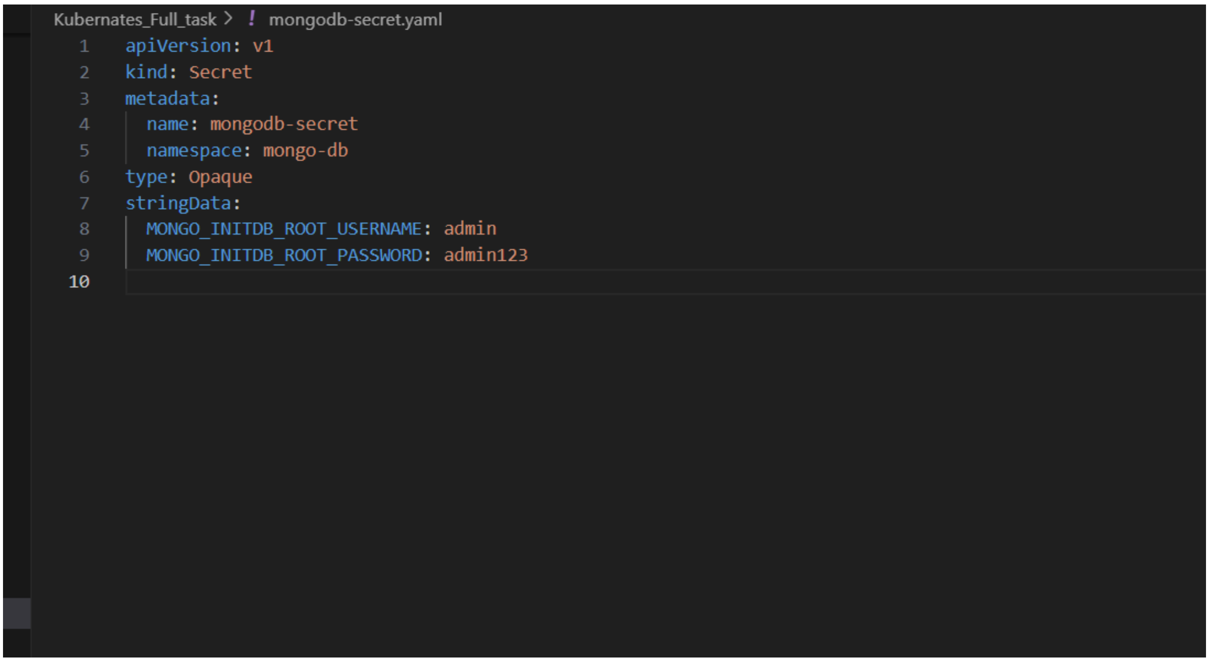
1 replica.

The image: mongo:latest

Create a Secret named mongodb-secret in the mongo-db namespace to store the MongoDB root username (MONGO_INITDB_ROOT_USERNAME) and password (MONGO_INITDB_ROOT_PASSWORD).

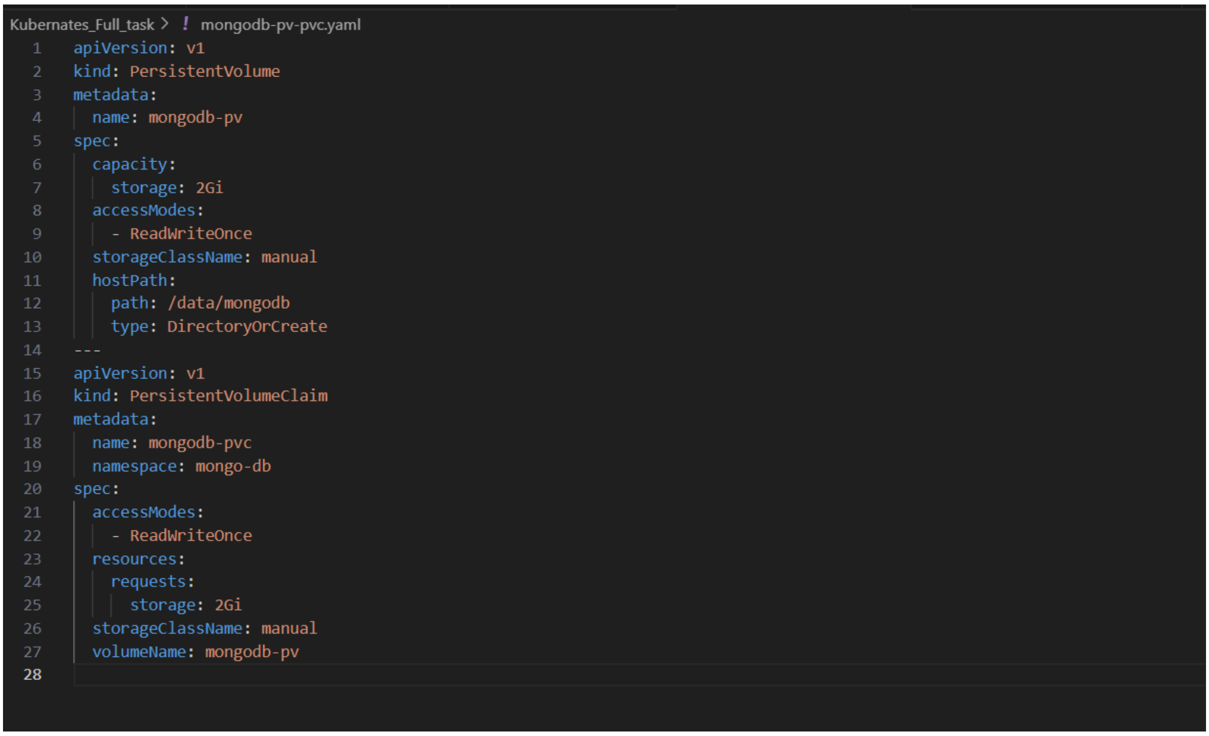
Root username: admin

Root password: admin123

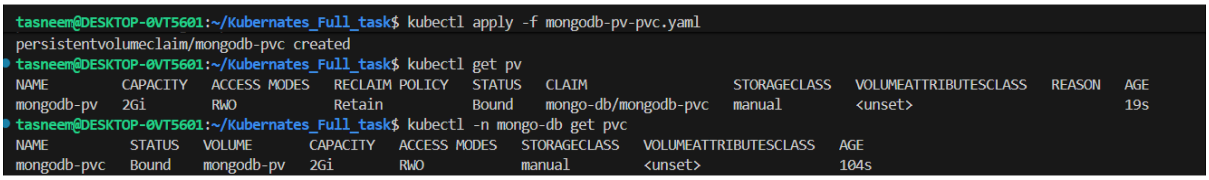


```yaml
apiVersion: v1
kind: Secret
metadata:
  name: mongodb-secret
  namespace: mongo-db
type: Opaque
stringData:
  MONGO_INITDB_ROOT_USERNAME: admin
  MONGO_INITDB_ROOT_PASSWORD: admin123
```

Use a Persistent Volume and Persistent Volume Claim (PVC) named mongodb-pvc to store MongoDB data at /data/db.

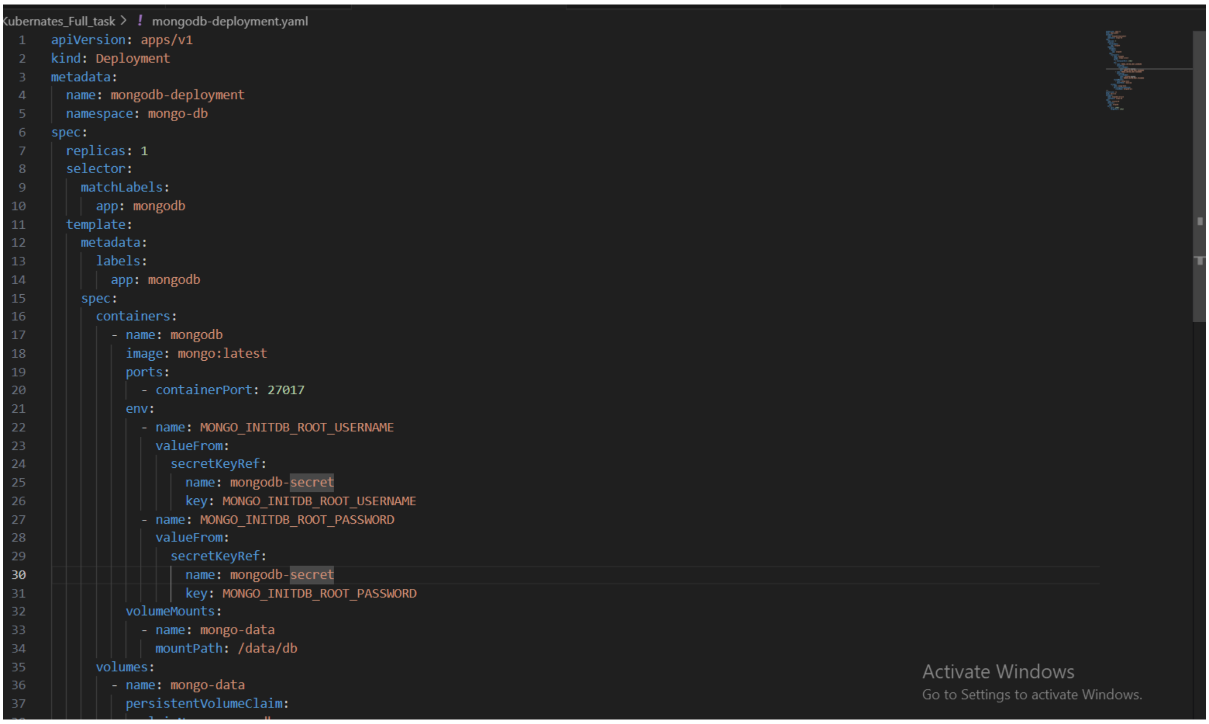


```yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: mongodb-pv
spec:
  capacity:
    storage: 2Gi
  accessModes:
    - ReadWriteOnce
  storageClassName: manual
  hostPath:
    path: /data/mongodb
    type: DirectoryOrCreate
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mongodb-pvc
  namespace: mongo-db
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 2Gi
  storageClassName: manual
  volumeName: mongodb-pv
```



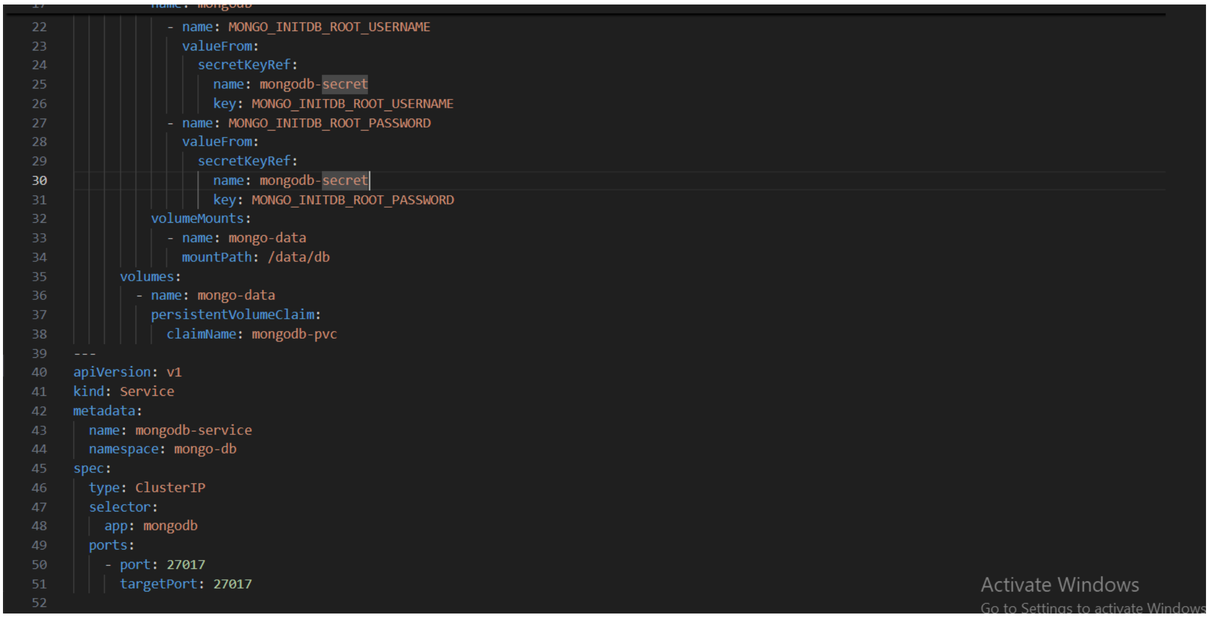
```
tasneem@DESKTOP-0VT5601:~/Kubernates_Full_task$ kubectl apply -f mongodb-pv-pvc.yaml
persistentvolumeclaim/mongodb-pvc created
tasneem@DESKTOP-0VT5601:~/Kubernates_Full_task$ kubectl get pv
NAME         CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM                    STORAGECLASS   VOLUMEATTRIBUTESCLASS   REASON   AGE
mongodb-pv   2Gi        RWO            Retain           Bound    mongo-db/mongodb-pvc     manual         <unset>                          19s
tasneem@DESKTOP-0VT5601:~/Kubernates_Full_task$ kubectl -n mongo-db get pvc
NAME          STATUS   VOLUME       CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
mongodb-pvc   Bound    mongodb-pv   2Gi        RWO            manual         <unset>                 104s
```

Create a ClusterIP Service named mongodb-service to expose the MongoDB database internally within the cluster on port 27017.



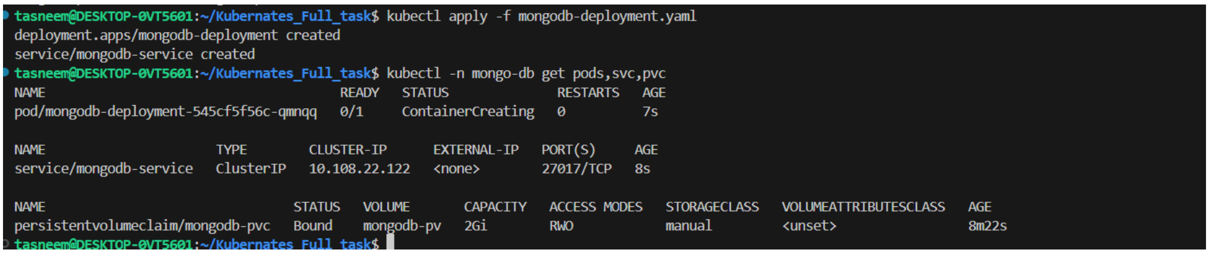
Kubernates_Full_task > ! mongodb-deployment.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mongodb-deployment
  namespace: mongo-db
spec:
  replicas: 1
  selector:
    matchLabels:
      app: mongodb
  template:
    metadata:
      labels:
        app: mongodb
    spec:
      containers:
        - name: mongodb
          image: mongo:latest
          ports:
            - containerPort: 27017
          env:
            - name: MONGO_INITDB_ROOT_USERNAME
              valueFrom:
                secretKeyRef:
                  name: mongodb-secret
                  key: MONGO_INITDB_ROOT_USERNAME
            - name: MONGO_INITDB_ROOT_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: mongodb-secret
                  key: MONGO_INITDB_ROOT_PASSWORD
          volumeMounts:
            - name: mongo-data
              mountPath: /data/db
      volumes:
        - name: mongo-data
          persistentVolumeClaim:
```



```yaml
            - name: MONGO_INITDB_ROOT_USERNAME
              valueFrom:
                secretKeyRef:
                  name: mongodb-secret
                  key: MONGO_INITDB_ROOT_USERNAME
            - name: MONGO_INITDB_ROOT_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: mongodb-secret
                  key: MONGO_INITDB_ROOT_PASSWORD
          volumeMounts:
            - name: mongo-data
              mountPath: /data/db
      volumes:
        - name: mongo-data
          persistentVolumeClaim:
            claimName: mongodb-pvc
---
apiVersion: v1
kind: Service
metadata:
  name: mongodb-service
  namespace: mongo-db
spec:
  type: ClusterIP
  selector:
    app: mongodb
  ports:
    - port: 27017
      targetPort: 27017

```



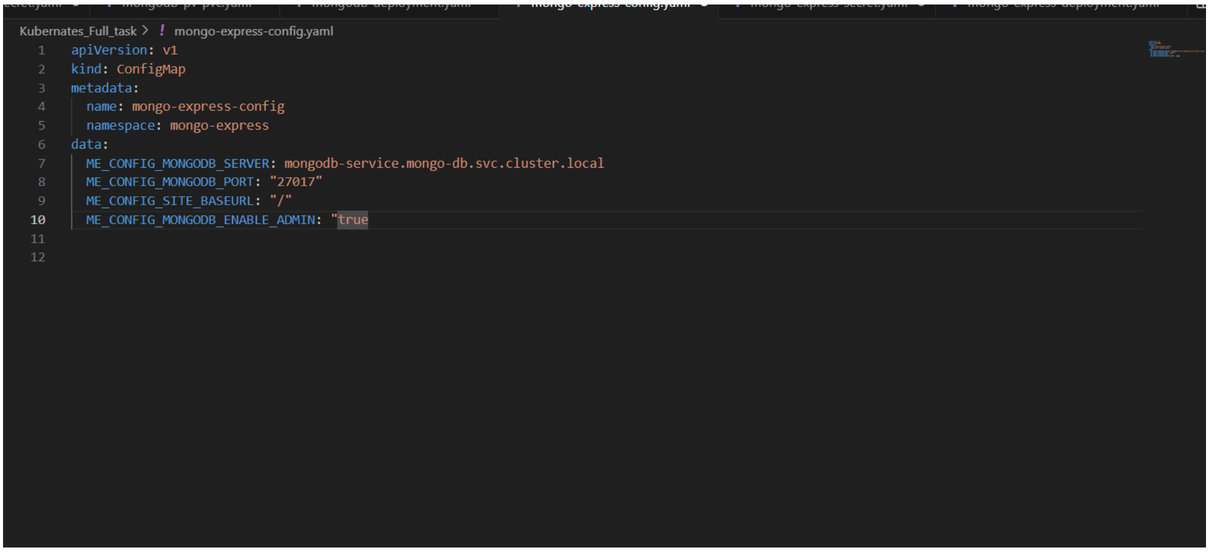
```
tasneem@DESKTOP-0VT5601:~/Kubernates_Full_task$ kubectl apply -f mongodb-deployment.yaml
deployment.apps/mongodb-deployment created
service/mongodb-service created
tasneem@DESKTOP-0VT5601:~/Kubernates_Full_task$ kubectl -n mongo-db get pods,svc,pvc
NAME                                     READY   STATUS              RESTARTS   AGE
pod/mongodb-deployment-545cf5f56c-qmnqq  0/1     ContainerCreating   0          7s

NAME                      TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)     AGE
service/mongodb-service   ClusterIP   10.108.22.122   <none>        27017/TCP   8s

NAME                                      STATUS   VOLUME       CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
persistentvolumeclaim/mongodb-pvc         Bound    mongodb-pv   2Gi        RWO            manual         <unset>                 8m22s
tasneem@DESKTOP-0VT5601:~/Kubernates_Full_task$
```

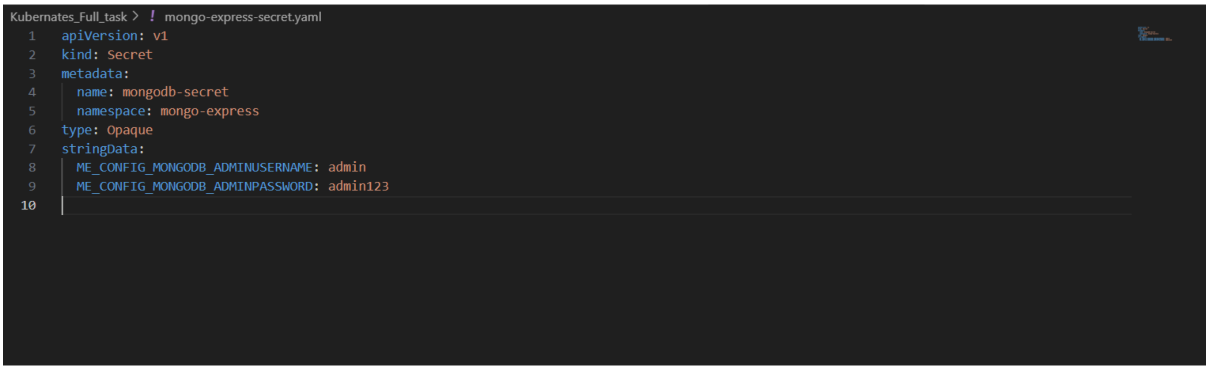
c- Deploy Mongo Express in the mongo-express namespace. Use a Deployment named mongo-express-deployment with: 1 replica. The image: mongo-express:latest. Create a

ConfigMap named mongo-express-config in the mongo-express namespace to store environment variables for the Mongo Express application. Hint (you should accces the MongoDB by mongo-express deployment) Create a NodePort Service named mongo-express-service to expose the Mongo Express interface externally on port 8081.



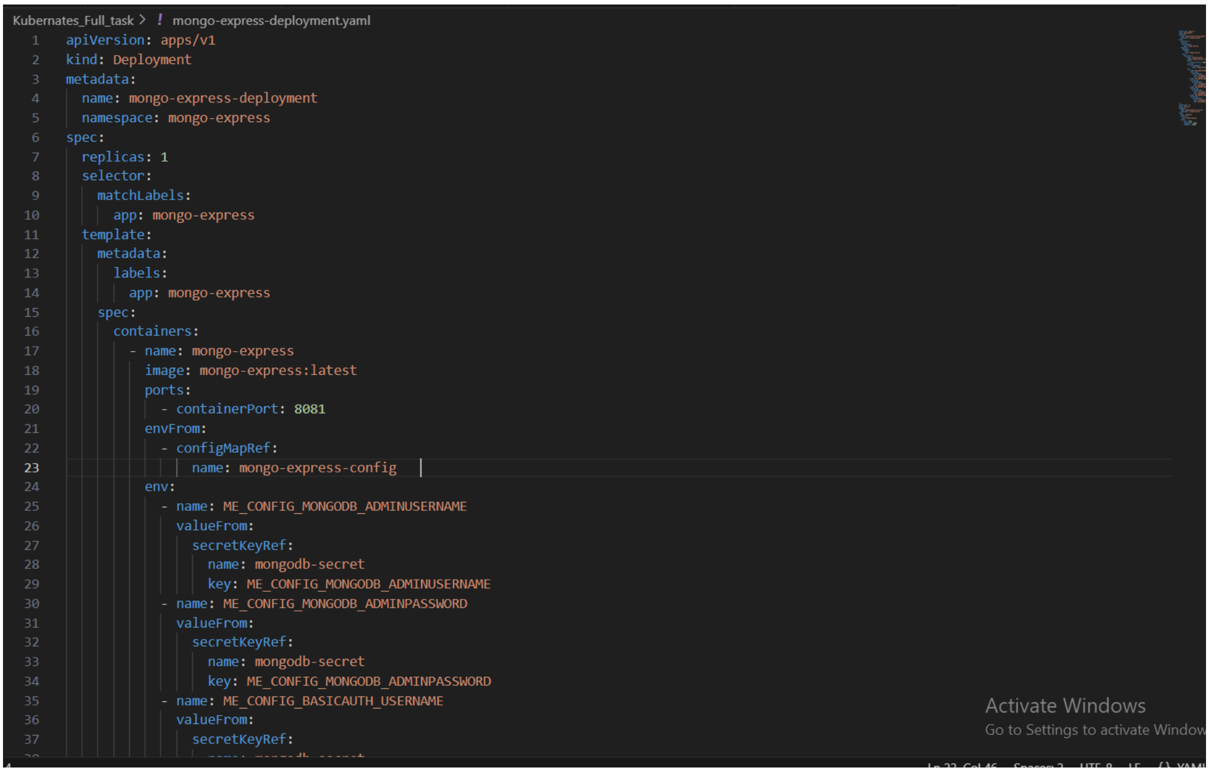
Kubernates_Full_task > ! mongo-express-config.yaml

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: mongo-express-config
  namespace: mongo-express
data:
  ME_CONFIG_MONGODB_SERVER: mongodb-service.mongo-db.svc.cluster.local
  ME_CONFIG_MONGODB_PORT: "27017"
  ME_CONFIG_SITE_BASEURL: "/"
  ME_CONFIG_MONGODB_ENABLE_ADMIN: "true
```



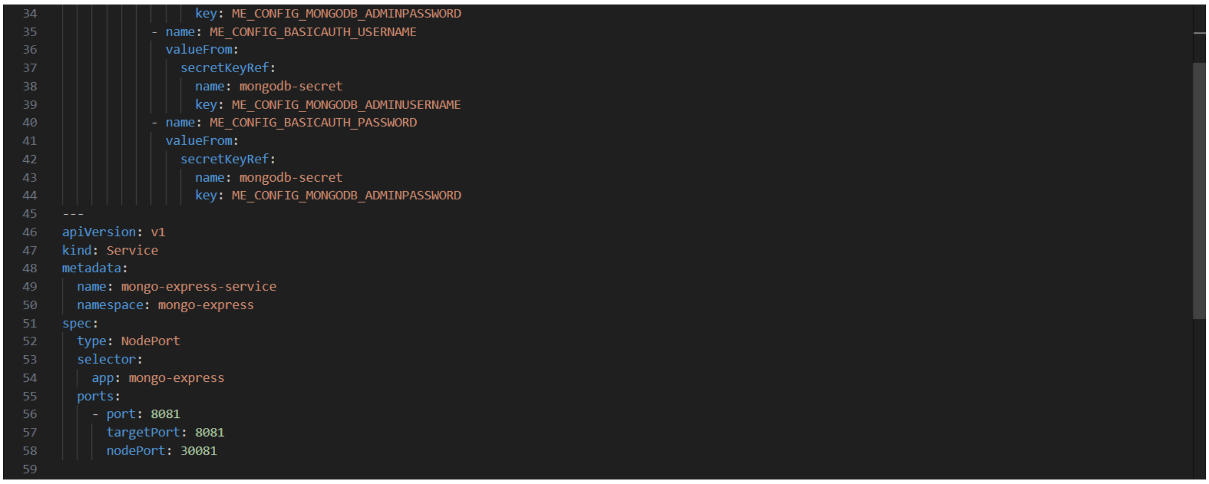
Kubernates_Full_task > ! mongo-express-secret.yaml

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: mongodb-secret
  namespace: mongo-express
type: Opaque
stringData:
  ME_CONFIG_MONGODB_ADMINUSERNAME: admin
  ME_CONFIG_MONGODB_ADMINPASSWORD: admin123
```

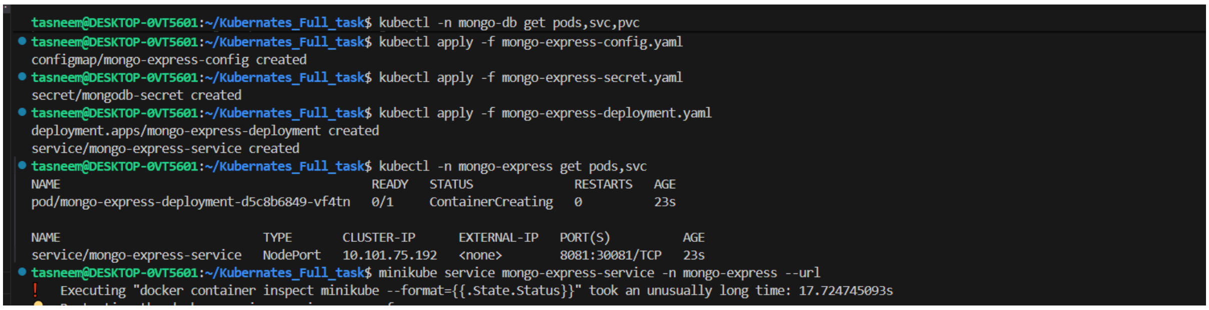


Kubernates_Full_task > ! mongo-express-deployment.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mongo-express-deployment
  namespace: mongo-express
spec:
  replicas: 1
  selector:
    matchLabels:
      app: mongo-express
  template:
    metadata:
      labels:
        app: mongo-express
    spec:
      containers:
        - name: mongo-express
          image: mongo-express:latest
          ports:
            - containerPort: 8081
          envFrom:
            - configMapRef:
                name: mongo-express-config
          env:
            - name: ME_CONFIG_MONGODB_ADMINUSERNAME
              valueFrom:
                secretKeyRef:
                  name: mongodb-secret
                  key: ME_CONFIG_MONGODB_ADMINUSERNAME
            - name: ME_CONFIG_MONGODB_ADMINPASSWORD
              valueFrom:
                secretKeyRef:
                  name: mongodb-secret
                  key: ME_CONFIG_MONGODB_ADMINPASSWORD
            - name: ME_CONFIG_BASICAUTH_USERNAME
              valueFrom:
                secretKeyRef:
```

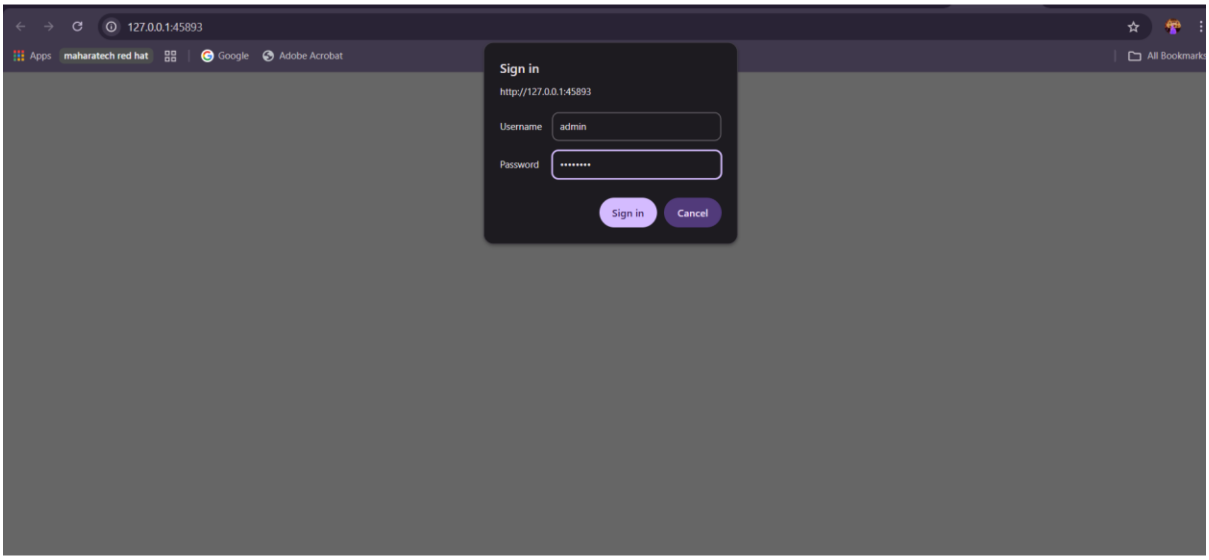


```
                  key: ME_CONFIG_MONGODB_ADMINPASSWORD
            - name: ME_CONFIG_BASICAUTH_USERNAME
              valueFrom:
                secretKeyRef:
                  name: mongodb-secret
                  key: ME_CONFIG_MONGODB_ADMINUSERNAME
            - name: ME_CONFIG_BASICAUTH_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: mongodb-secret
                  key: ME_CONFIG_MONGODB_ADMINPASSWORD
---
apiVersion: v1
kind: Service
metadata:
  name: mongo-express-service
  namespace: mongo-express
spec:
  type: NodePort
  selector:
    app: mongo-express
  ports:
    - port: 8081
      targetPort: 8081
      nodePort: 30081
```



```
tasneem@DESKTOP-0VT5601:~/Kubernates_Full_task$ kubectl -n mongo-db get pods,svc,pvc
tasneem@DESKTOP-0VT5601:~/Kubernates_Full_task$ kubectl apply -f mongo-express-config.yaml
configmap/mongo-express-config created
tasneem@DESKTOP-0VT5601:~/Kubernates_Full_task$ kubectl apply -f mongo-express-secret.yaml
secret/mongodb-secret created
tasneem@DESKTOP-0VT5601:~/Kubernates_Full_task$ kubectl apply -f mongo-express-deployment.yaml
deployment.apps/mongo-express-deployment created
service/mongo-express-service created
tasneem@DESKTOP-0VT5601:~/Kubernates_Full_task$ kubectl -n mongo-express get pods,svc
NAME                                          READY   STATUS              RESTARTS   AGE
pod/mongo-express-deployment-d5c8b6849-vf4tn   0/1     ContainerCreating   0          23s

NAME                            TYPE       CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
service/mongo-express-service   NodePort   10.101.75.192   <none>        8081:30081/TCP   23s
tasneem@DESKTOP-0VT5601:~/Kubernates_Full_task$ minikube service mongo-express-service -n mongo-express --url
!   Executing "docker container inspect minikube --format={{.State.Status}}" took an unusually long time: 17.724745093s
```

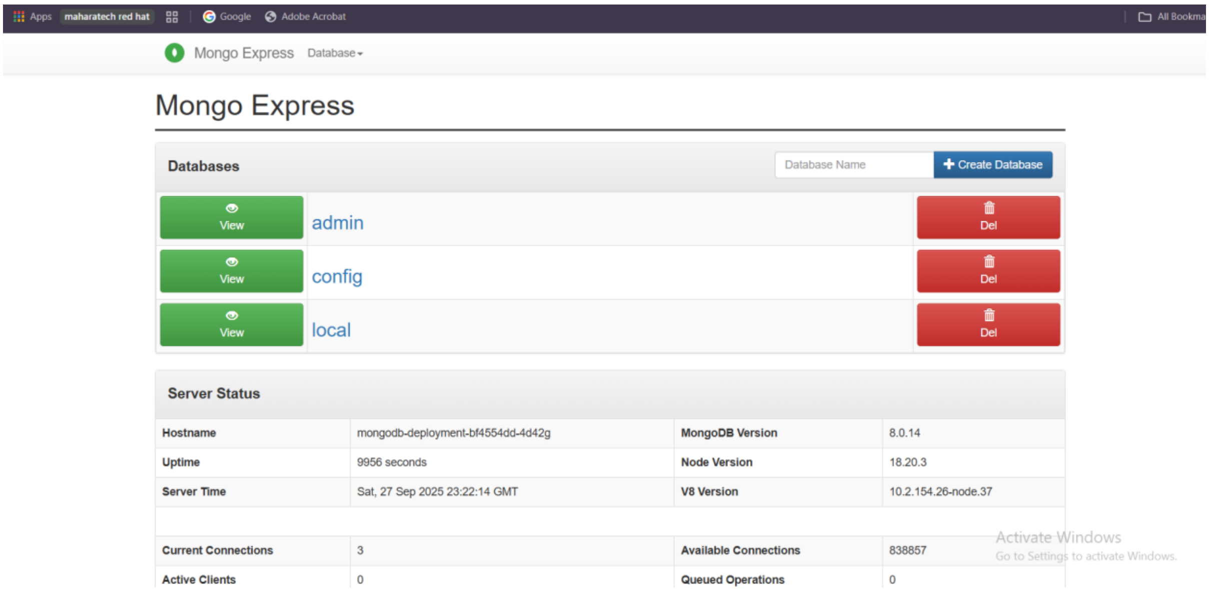


Sign in

http://127.0.0.1:45893

Username admin

Password ••••••••



Mongo Express

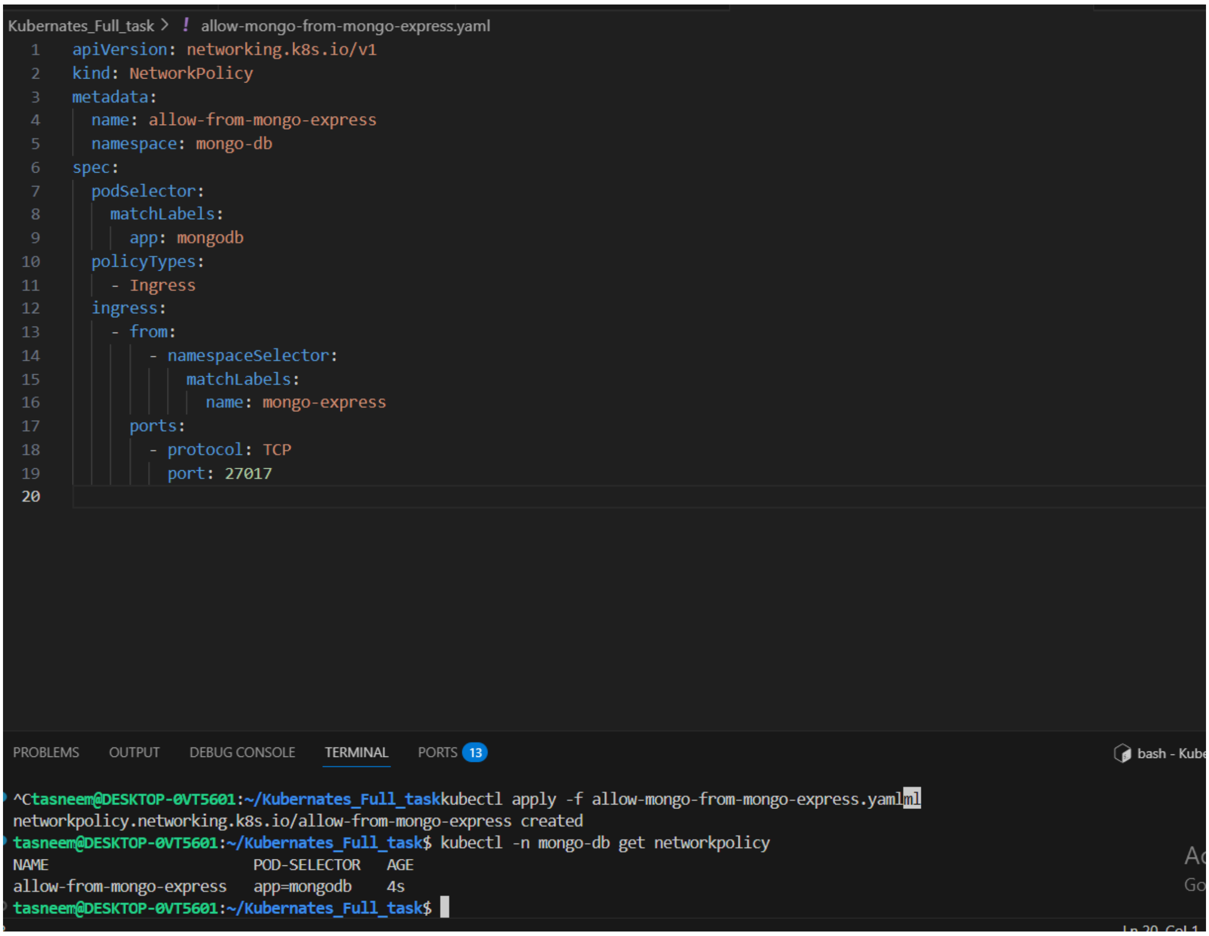
Databases

View	admin	Del
View	config	Del
View	local	Del

Server Status

Hostname	mongodb-deployment-bf4554dd-4d42g	MongoDB Version	8.0.14
Uptime	9956 seconds	Node Version	18.20.3
Server Time	Sat, 27 Sep 2025 23:22:14 GMT	V8 Version	10.2.154.26-node.37
Current Connections	3	Available Connections	838857
Active Clients	0	Queued Operations	0

d- apply network policy in the namespace mongo-db to be accssiable from only mongo-express namespace



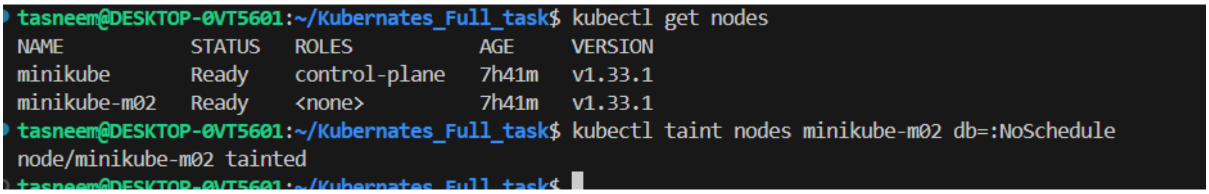
```yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: allow-from-mongo-express
  namespace: mongo-db
spec:
  podSelector:
    matchLabels:
      app: mongodb
  policyTypes:
    - Ingress
  ingress:
    - from:
        - namespaceSelector:
            matchLabels:
              name: mongo-express
      ports:
        - protocol: TCP
          port: 27017

```

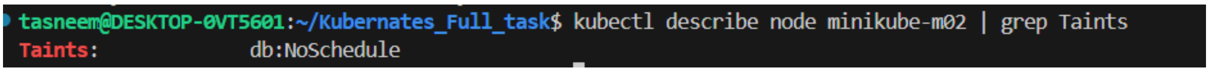
```
^Ctasneem@DESKTOP-0VT5601:~/Kubernates_Full_taskkubectl apply -f allow-mongo-from-mongo-express.yamlml
networkpolicy.networking.k8s.io/allow-from-mongo-express created
tasneem@DESKTOP-0VT5601:~/Kubernates_Full_task$ kubectl -n mongo-db get networkpolicy
NAME                       POD-SELECTOR   AGE
allow-from-mongo-express   app=mongodb    4s
tasneem@DESKTOP-0VT5601:~/Kubernates_Full_task$ 
```

4- Use Taints and Tolerations:

a- Taint one of the Minikube nodes with key=db:NoSchedule.

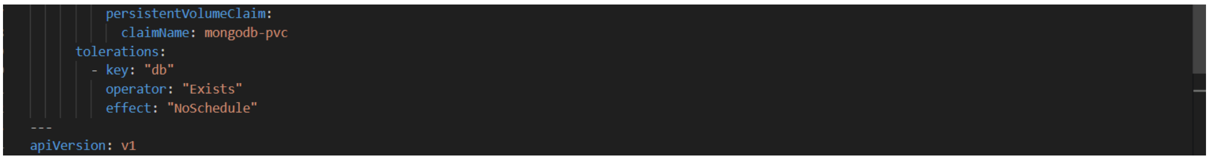


```
tasneem@DESKTOP-0VT5601:~/Kubernates_Full_task$ kubectl get nodes
NAME           STATUS   ROLES           AGE    VERSION
minikube       Ready    control-plane   7h41m  v1.33.1
minikube-m02   Ready    <none>          7h41m  v1.33.1
tasneem@DESKTOP-0VT5601:~/Kubernates_Full_task$ kubectl taint nodes minikube-m02 db=:NoSchedule
node/minikube-m02 tainted
tasneem@DESKTOP-0VT5601:~/Kubernates_Full_task$
```

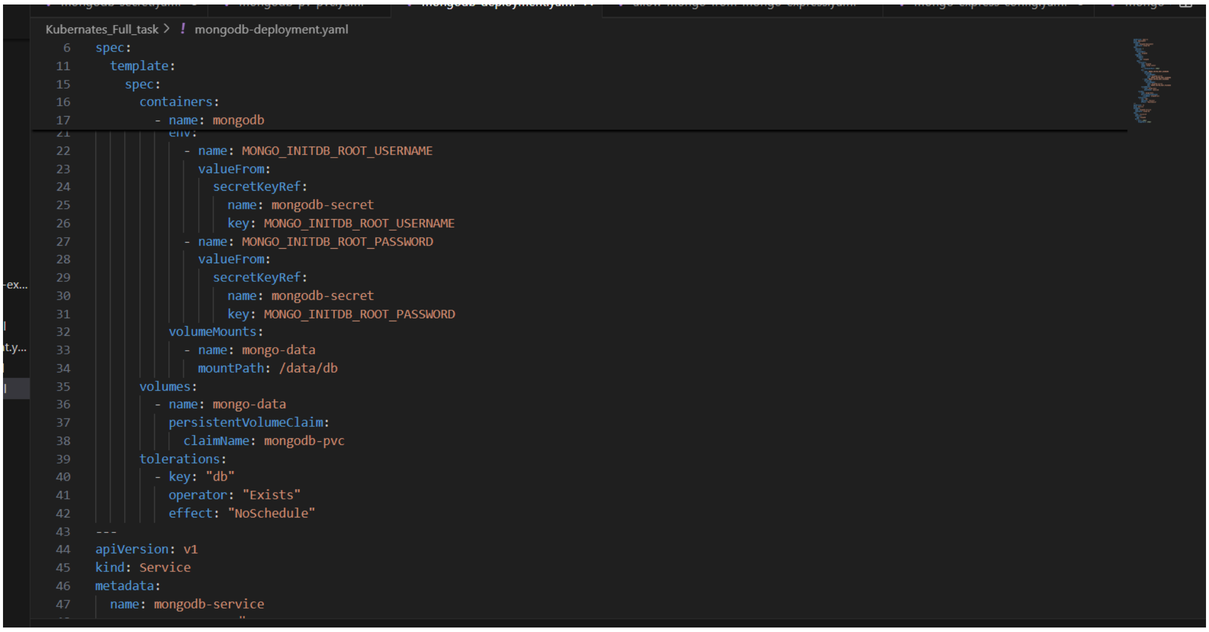


```
tasneem@DESKTOP-0VT5601:~/Kubernates_Full_task$ kubectl describe node minikube-m02 | grep Taints
Taints:             db:NoSchedule
```

b- Apply a Toleration in the mongodb-deployment to allow the MongoDB pod to be scheduled on the tainted node



```yaml
          persistentVolumeClaim:
            claimName: mongodb-pvc
      tolerations:
        - key: "db"
          operator: "Exists"
          effect: "NoSchedule"
---
apiVersion: v1
```



Kubernates_Full_task > ! mongodb-deployment.yaml

```yaml
spec:
  template:
    spec:
      containers:
        - name: mongodb
          env:
            - name: MONGO_INITDB_ROOT_USERNAME
              valueFrom:
                secretKeyRef:
                  name: mongodb-secret
                  key: MONGO_INITDB_ROOT_USERNAME
            - name: MONGO_INITDB_ROOT_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: mongodb-secret
                  key: MONGO_INITDB_ROOT_PASSWORD
          volumeMounts:
            - name: mongo-data
              mountPath: /data/db
      volumes:
        - name: mongo-data
          persistentVolumeClaim:
            claimName: mongodb-pvc
      tolerations:
        - key: "db"
          operator: "Exists"
          effect: "NoSchedule"
---
apiVersion: v1
kind: Service
metadata:
  name: mongodb-service
```