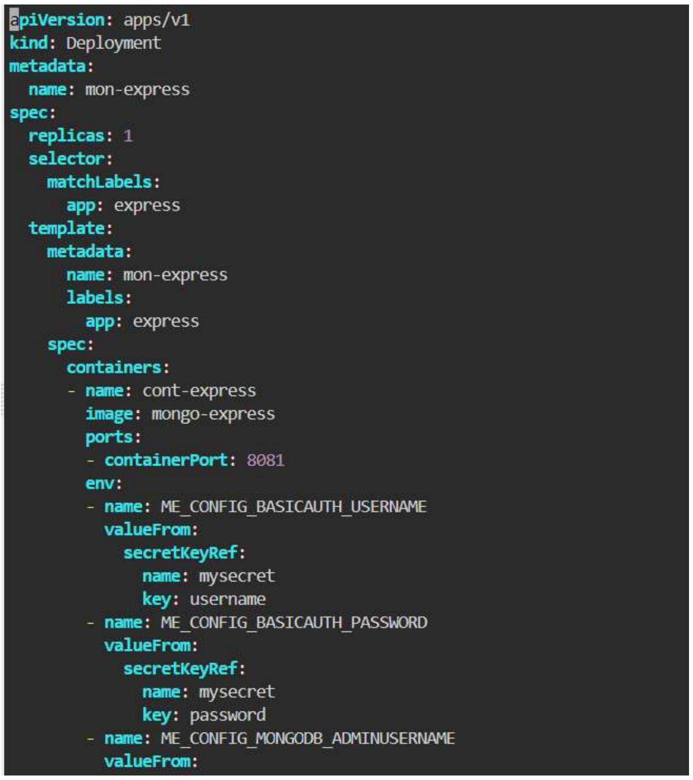
## Connecting application

### Frontend application (Mongo express)

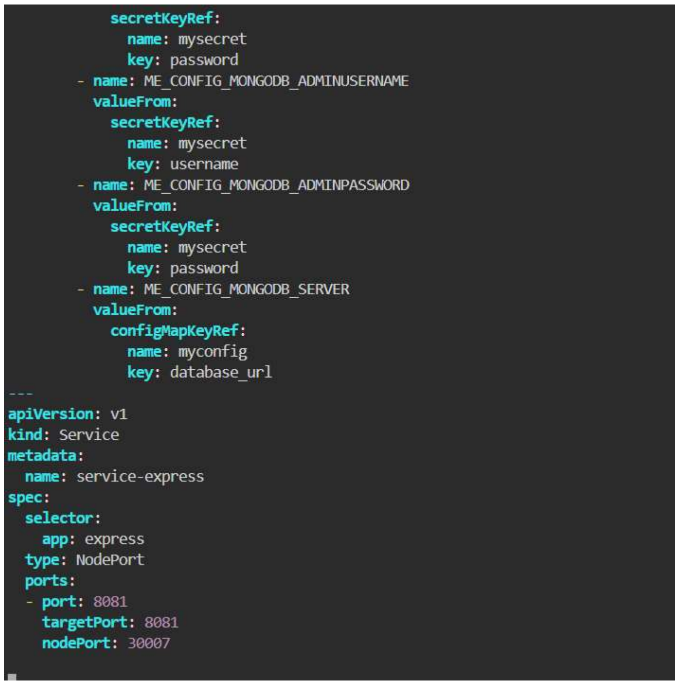
- ❖ Kubernetes (K8s), the frontend with Mongo Express involves running Mongo Express as a pod within the cluster.
- ❖ Mongo Express provides a web-based interface for managing MongoDB databases, allowing users to interact with and visualize their data.
- ❖ Kubernetes handles the deployment, scaling, and maintenance of the Mongo Express pod.

### 1. Create the yaml file (Vi A1.yaml)

#### Vi A1.yaml



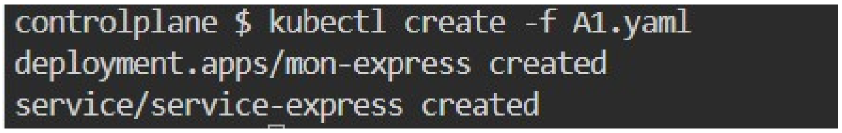
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mon-express
spec:
  replicas: 1
  selector:
    matchLabels:
      app: express
  template:
    metadata:
      name: mon-express
      labels:
        app: express
    spec:
      containers:
      - name: cont-express
        image: mongo-express
        ports:
        - containerPort: 8081
        env:
        - name: ME_CONFIG_BASICAUTH_USERNAME
          valueFrom:
            secretKeyRef:
              name: mysecret
              key: username
        - name: ME_CONFIG_BASICAUTH_PASSWORD
          valueFrom:
            secretKeyRef:
              name: mysecret
              key: password
        - name: ME_CONFIG_MONGODB_ADMINUSERNAME
          valueFrom:
```



```
            secretKeyRef:
              name: mysecret
              key: password
        - name: ME_CONFIG_MONGODB_ADMINUSERNAME
          valueFrom:
            secretKeyRef:
              name: mysecret
              key: username
        - name: ME_CONFIG_MONGODB_ADMINPASSWORD
          valueFrom:
            secretKeyRef:
              name: mysecret
              key: password
        - name: ME_CONFIG_MONGODB_SERVER
          valueFrom:
            configMapKeyRef:
              name: myconfig
              key: database_url
---
apiVersion: v1
kind: Service
metadata:
  name: service-express
spec:
  selector:
    app: express
  type: NodePort
  ports:
  - port: 8081
    targetPort: 8081
    nodePort: 30007
```

## 2. Create the Deployment and service on mon-express:

**Kubectl create –f A1.yaml**



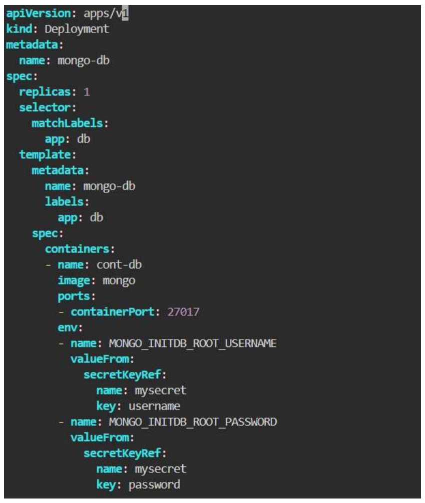
```
controlplane $ kubectl create -f A1.yaml
deployment.apps/mon-express created
service/service-express created
```

## Backend application (Mongo db)

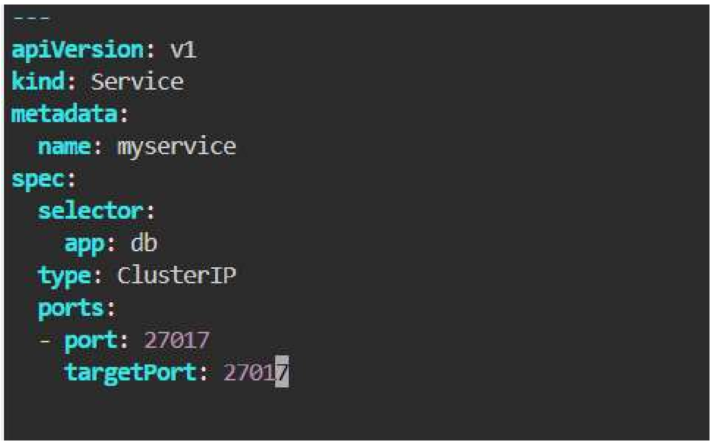
- ❖ In Kubernetes (K8s), the backend with MongoDB involves running MongoDB instances as pods within the cluster.
- ❖ These pods handle data storage and retrieval, providing database services to other application components.
- ❖ Kubernetes manages the deployment, scaling, and maintenance of these MongoDB pods.

**1. Create the Yaml file (vi B1.yaml)**

**Vi B1.yaml**



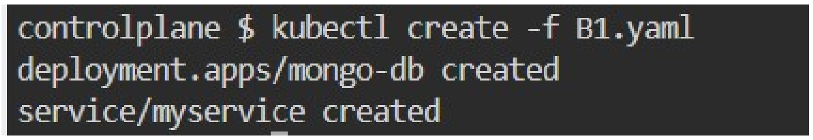
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mongo-db
spec:
  replicas: 1
  selector:
    matchLabels:
      app: db
  template:
    metadata:
      name: mongo-db
      labels:
        app: db
    spec:
      containers:
      - name: cont-db
        image: mongo
        ports:
        - containerPort: 27017
        env:
        - name: MONGO_INITDB_ROOT_USERNAME
          valueFrom:
            secretKeyRef:
              name: mysecret
              key: username
        - name: MONGO_INITDB_ROOT_PASSWORD
          valueFrom:
            secretKeyRef:
              name: mysecret
              key: password
```



```yaml
---
apiVersion: v1
kind: Service
metadata:
  name: myservice
spec:
  selector:
    app: db
  type: ClusterIP
  ports:
  - port: 27017
    targetPort: 27017
```

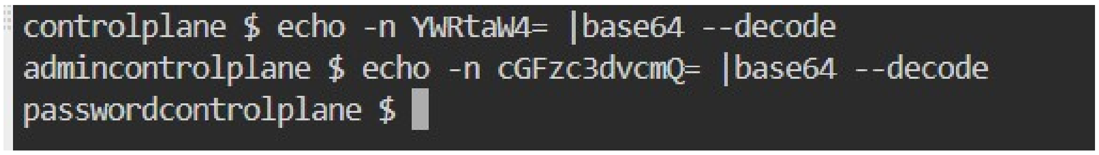
**2. Create the Deployment and service on Mongo db:**

**Kubectl create –f B1.yaml**



```
controlplane $ kubectl create -f B1.yaml
deployment.apps/mongo-db created
service/myservice created
```

## 3. You can check the username and password:



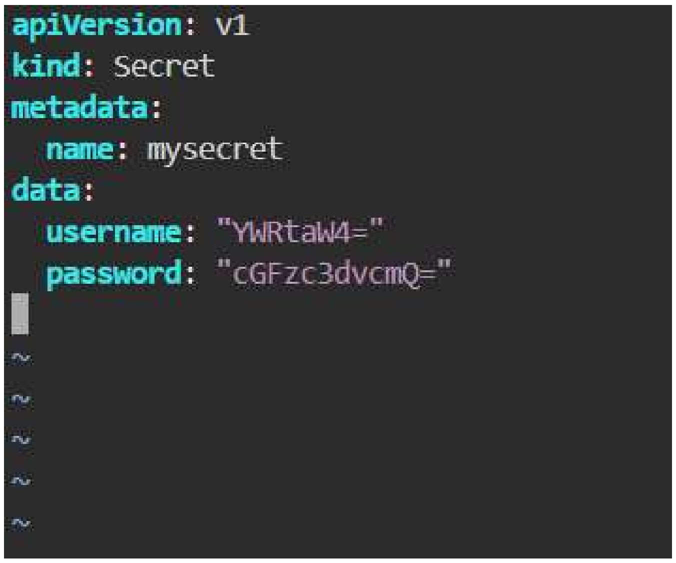
```
controlplane $ echo -n YWRtaW4= |base64 --decode
admincontrolplane $ echo -n cGFzc3dvcmQ= |base64 --decode
passwordcontrolplane $
```

## Secrets

- ❖ In Kubernetes (K8s), a Secret is an object that stores sensitive data, such as passwords, OAuth tokens, and SSH keys.
- ❖ Secrets allow you to securely manage and access confidential information in your applications.
- ❖ Kubernetes ensures that Secrets are only accessible to authorized pods and users.

1. **Create the Yaml file (Vi C1.yaml)**

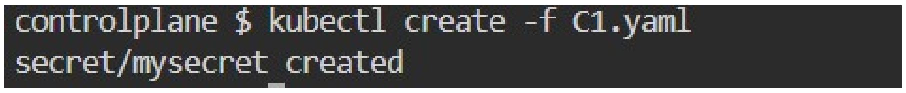
**Vi C1.yaml**



```
apiVersion: v1
kind: Secret
metadata:
  name: mysecret
data:
  username: "YWRtaW4="
  password: "cGFzc3dvcmQ="

~
~
~
~
~
```

2. **Create the secret:**

**Kubectl create –f C1.yaml**



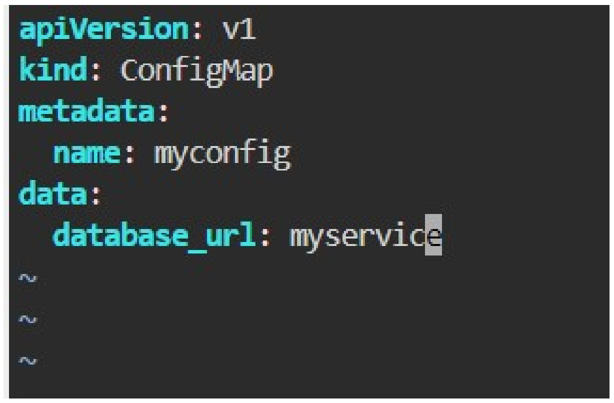
```
controlplane $ kubectl create -f C1.yaml
secret/mysecret created
```

**ConfingMap**

- ❖ In Kubernetes (K8s), a ConfigMap is an object used to store non-confidential configuration data in key-value pairs.
- ❖ ConfigMaps allow you to decouple configuration artifacts from image content to keep containerized applications portable.
- ❖ Kubernetes uses ConfigMaps to inject configuration data into pods and containers at runtime.

### 1. Create the Yaml file (vi D1.yaml)

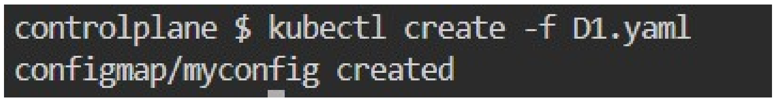
**Vi D1.yaml**



```
apiVersion: v1
kind: ConfigMap
metadata:
  name: myconfig
data:
  database_url: myservice
~
~
~
```

## 2. Create the Configmap:

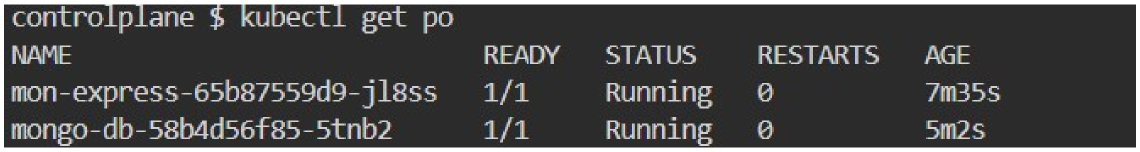
## Kubectl create –f D1.yaml



```
controlplane $ kubectl create -f D1.yaml
configmap/myconfig created
```

### 3. To verify the Pods:

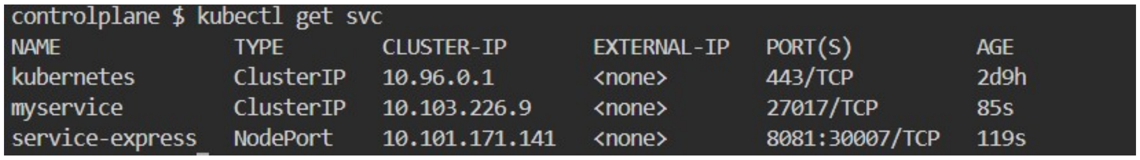
**Kubectl get po**



```
controlplane $ kubectl get po
NAME                            READY   STATUS    RESTARTS   AGE
mon-express-65b87559d9-jl8ss    1/1     Running   0          7m35s
mongo-db-58b4d56f85-5tnb2       1/1     Running   0          5m2s
```

**4. To verify the service:**

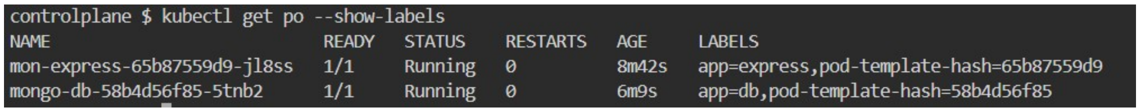
**Kubectl get svc**



```
controlplane $ kubectl get svc
NAME              TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
kubernetes        ClusterIP   10.96.0.1        <none>        443/TCP          2d9h
myservice         ClusterIP   10.103.226.9     <none>        27017/TCP        85s
service-express   NodePort    10.101.171.141   <none>        8081:30007/TCP   119s
```

**5. To using the command given below:**

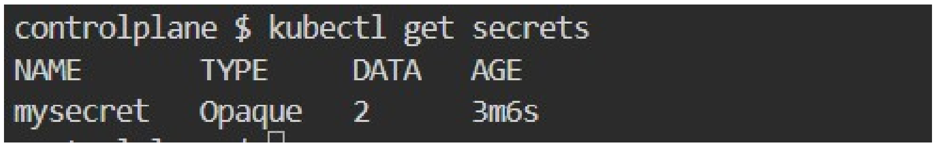
**Kubectl get po –show-labels**



```
controlplane $ kubectl get po --show-labels
NAME                           READY   STATUS    RESTARTS   AGE     LABELS
mon-express-65b87559d9-jl8ss   1/1     Running   0          8m42s   app=express,pod-template-hash=65b87559d9
mongo-db-58b4d56f85-5tnb2      1/1     Running   0          6m9s    app=db,pod-template-hash=58b4d56f85
```

**6. To verify the secrets:**

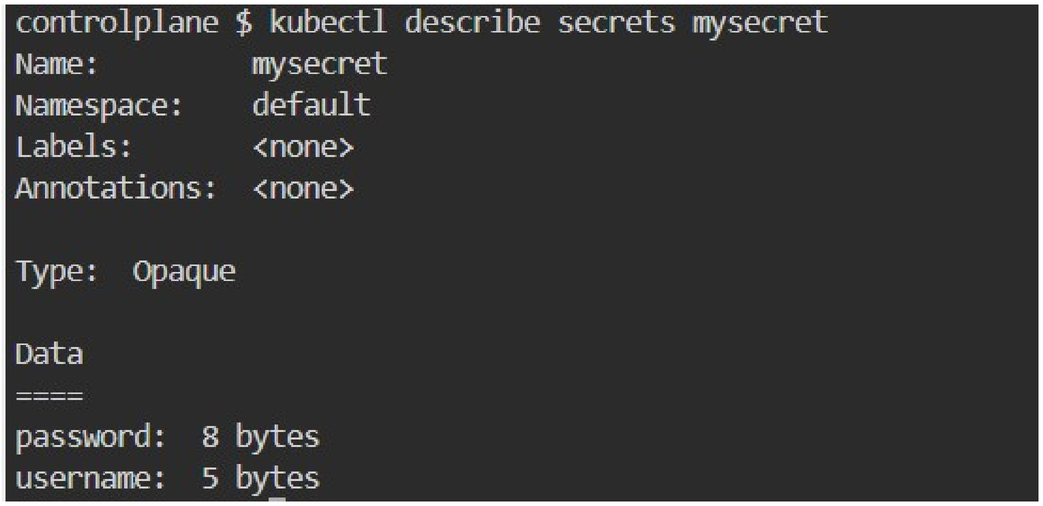
**Kubectl get secrets**



```
controlplane $ kubectl get secrets
NAME       TYPE     DATA   AGE
mysecret   Opaque   2      3m6s
```

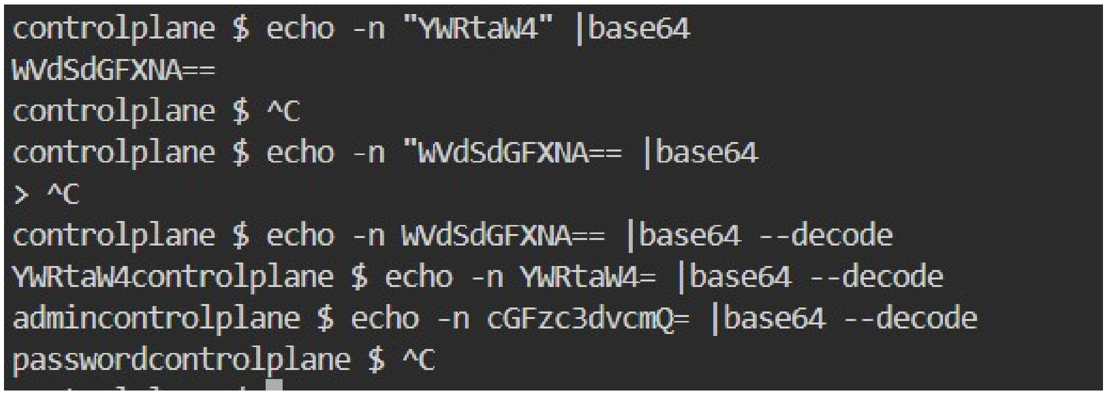
**7. To using describe details:**

**Kubectl describe sercets mysecret**



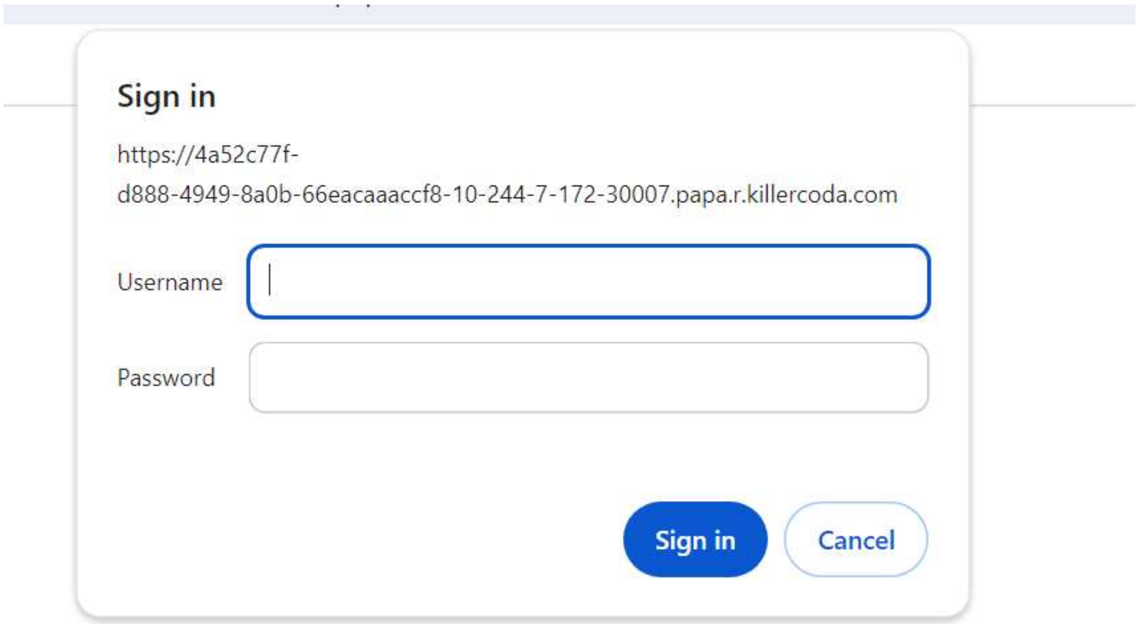
```
controlplane $ kubectl describe secrets mysecret
Name:         mysecret
Namespace:    default
Labels:       <none>
Annotations:  <none>

Type:  Opaque

Data
====
password:  8 bytes
username:  5 bytes
```

**8. To using the incept user and password details:**



```
controlplane $ echo -n "YWRtaW4" |base64
WVdSdGFXNA==
controlplane $ ^C
controlplane $ echo -n "WVdSdGFXNA== |base64
> ^C
controlplane $ echo -n WVdSdGFXNA== |base64 --decode
YWRtaW4controlplane $ echo -n YWRtaW4= |base64 --decode
admincontrolplane $ echo -n cGFzc3dvcmQ= |base64 --decode
passwordcontrolplane $ ^C
```

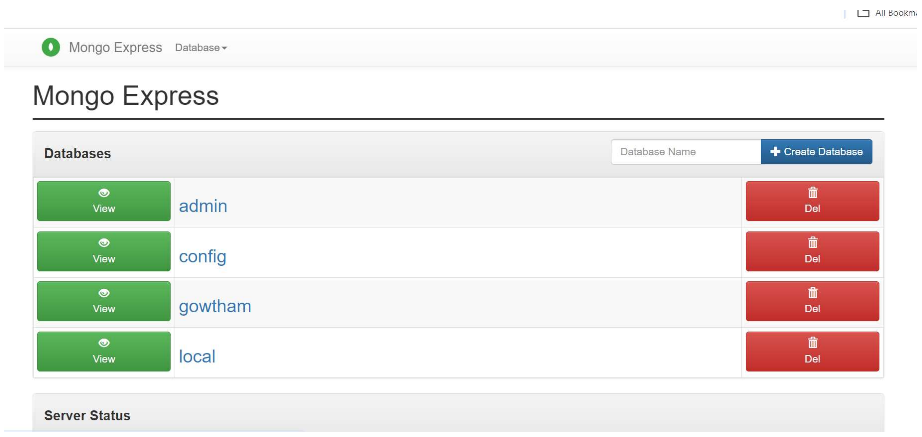
**9. Going to outside on webpage and enter the port no to access it and enter the username and then password:**



Sign in

https://4a52c77f-d888-4949-8a0b-66eacaaaccf8-10-244-7-172-30007.papa.r.killercoda.com

## 10. Finally to verify the web pages details:



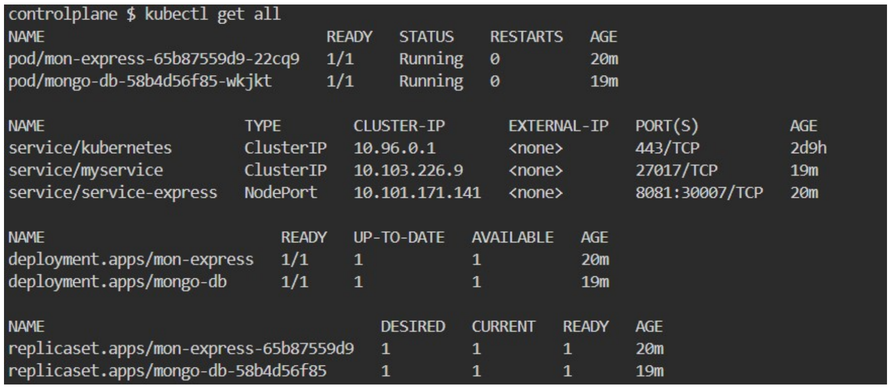
Mongo Express

Databases

| | | |
|---|---|---|
| View | admin | Del |
| View | config | Del |
| View | gowtham | Del |
| View | local | Del |

## 11. To using one command to over list the pod and service:

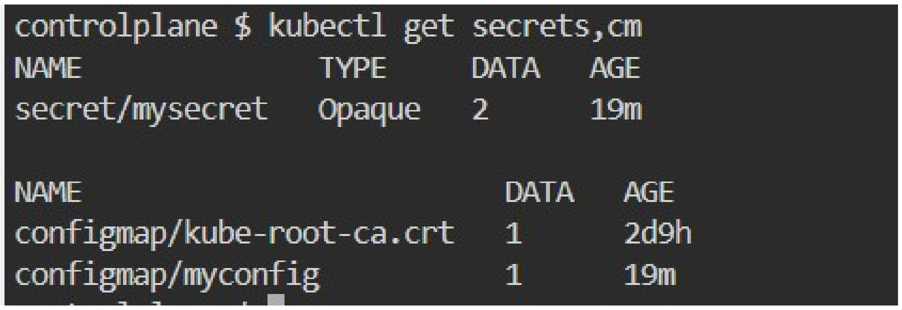
**Kubectl get all**



```
controlplane $ kubectl get all
NAME                                READY   STATUS    RESTARTS   AGE
pod/mon-express-65b87559d9-22cq9    1/1     Running   0          20m
pod/mongo-db-58b4d56f85-wkjkt       1/1     Running   0          19m

NAME                       TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
service/kubernetes         ClusterIP   10.96.0.1        <none>        443/TCP          2d9h
service/myservice          ClusterIP   10.103.226.9     <none>        27017/TCP        19m
service/service-express    NodePort    10.101.171.141   <none>        8081:30007/TCP   20m

NAME                          READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/mon-express   1/1     1            1           20m
deployment.apps/mongo-db      1/1     1            1           19m

NAME                                     DESIRED   CURRENT   READY   AGE
replicaset.apps/mon-express-65b87559d9   1         1         1       20m
replicaset.apps/mongo-db-58b4d56f85      1         1         1       19m
```

## 12. To using one command to list the secret and configmap:

**Kubectl get secrets,cm**



```
controlplane $ kubectl get secrets,cm
NAME              TYPE     DATA   AGE
secret/mysecret   Opaque   2      19m

NAME                           DATA   AGE
configmap/kube-root-ca.crt    1      2d9h
configmap/myconfig            1      19m
```