






Integrate MongoDB & Mongo Express on Kubernetes

This project deploys **MongoDB** and **Mongo Express** on a Kubernetes cluster using a **StatefulSet** and **Secrets** for authentication.

Features

- **MongoDB**  - NoSQL Database deployed as a **StatefulSet**
 - **Mongo Express**  - Web-based MongoDB admin interface
 - **Kubernetes Secrets**  - Securely store database credentials
 - **Persistent Storage**  - Data persists across pod restarts
 - **Port Forwarding**  - Access Mongo Express locally
-

Project Structure

```
└─ Lab-2
   ├── mongodb-namespace.yaml      # Namespace for MongoDB
   ├── mongodb-deployment.yaml    # MongoDB StatefulSet deployment
   ├── mongodb-service.yaml       # Service to expose MongoDB
   ├── mongo-express-deployment.yaml # Mongo Express deployment
   └── mongo-express-service.yaml  # Service to expose Mongo Express
```

Setup & Deployment

1 Create a Namespace (Optional)

```
# mongodb-namespace.yaml
apiVersion: v1
kind: Namespace
metadata:
  name: mongodb
```

```
kubectl apply -f mongodb-namespace.yaml
```

2 Create Secrets for MongoDB Credentials

```
kubectl create secret generic mongodb-secrets \  
  --namespace=mongodb \  
  --from-literal=root-username=admin \  
  --from-literal=root-password=SecurePass123! \  
  --from-literal=database=mydb
```

3 Deploy MongoDB with Authentication

```
# mongodb-deployment.yaml  
apiVersion: apps/v1  
kind: StatefulSet  
metadata:  
  name: mongodb  
  namespace: mongodb  
spec:  
  serviceName: mongodb  
  replicas: 1  
  selector:  
    matchLabels:  
      app: mongodb  
  template:  
    metadata:  
      labels:  
        app: mongodb  
    spec:  
      containers:  
        - name: mongodb  
          image: mongo:6.0  
          env:  
            - name: MONGO_INITDB_ROOT_USERNAME  
              valueFrom:  
                secretKeyRef:  
                  name: mongodb-secrets  
                  key: root-username  
            - name: MONGO_INITDB_ROOT_PASSWORD  
              valueFrom:  
                secretKeyRef:  
                  name: mongodb-secrets  
                  key: root-password  
          ports:  
            - containerPort: 27017  
          volumeMounts:  
            - name: mongodb-pvc  
              mountPath: /data/db  
      volumeClaimTemplates:  
        - metadata:  
            name: mongodb-pvc  
          spec:  
            accessModes: [ "ReadWriteOnce" ]  
            resources:
```

```
requests:
  storage: 2Gi
```

```
# mongodb-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: mongodb
  namespace: mongodb
spec:
  selector:
    app: mongodb
  ports:
    - protocol: TCP
      port: 27017
      targetPort: 27017
```

```
kubectl apply -f mongodb-deployment.yaml
kubectl apply -f mongodb-service.yaml
```

4 Deploy MongoDB-Express

```
# mongo-express-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mongo-express
  namespace: mongodb
spec:
  replicas: 1
  selector:
    matchLabels:
      app: mongo-express
  template:
    metadata:
      labels:
        app: mongo-express
    spec:
      containers:
        - name: mongo-express
          image: mongo-express:1.0
          env:
            - name: ME_CONFIG_MONGODB_ADMINUSERNAME
              valueFrom:
                secretKeyRef:
                  name: mongodb-secrets
                  key: root-username
```

```
- name: ME_CONFIG_MONGODB_ADMINPASSWORD
  valueFrom:
    secretKeyRef:
      name: mongodb-secrets
      key: root-password
- name: ME_CONFIG_MONGODB_URL
  value:
"mongodb://admin:SecurePass123!@mongodb.mongodb.svc.cluster.local:27017/"
```

```
# mongo-express-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: mongo-express
  namespace: mongodb
spec:
  type: NodePort
  selector:
    app: mongo-express
  ports:
    - protocol: TCP
      port: 8081
      targetPort: 8081
```

```
kubectl apply -f mongo-express-deployment.yaml
kubectl apply -f mongo-express-service.yaml
```

Accessing Mongo Express

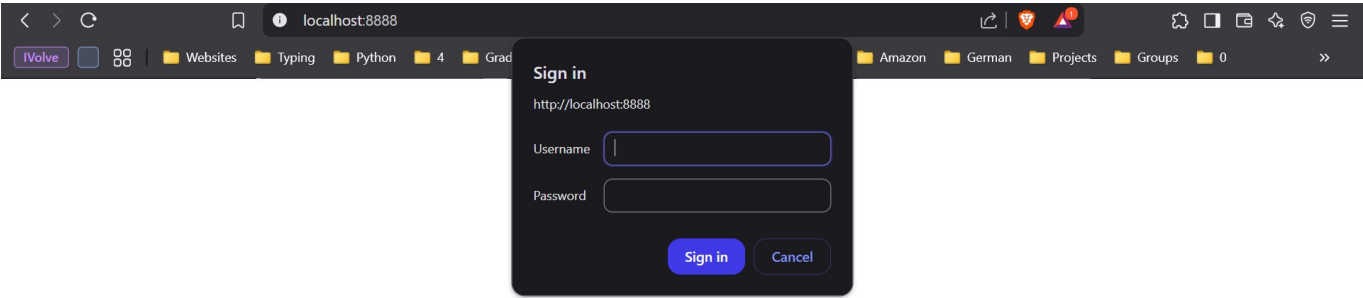
Option 1: Using Port Forwarding

```
kubectl port-forward -n mongodb svc/mongo-express 8888:8081
```

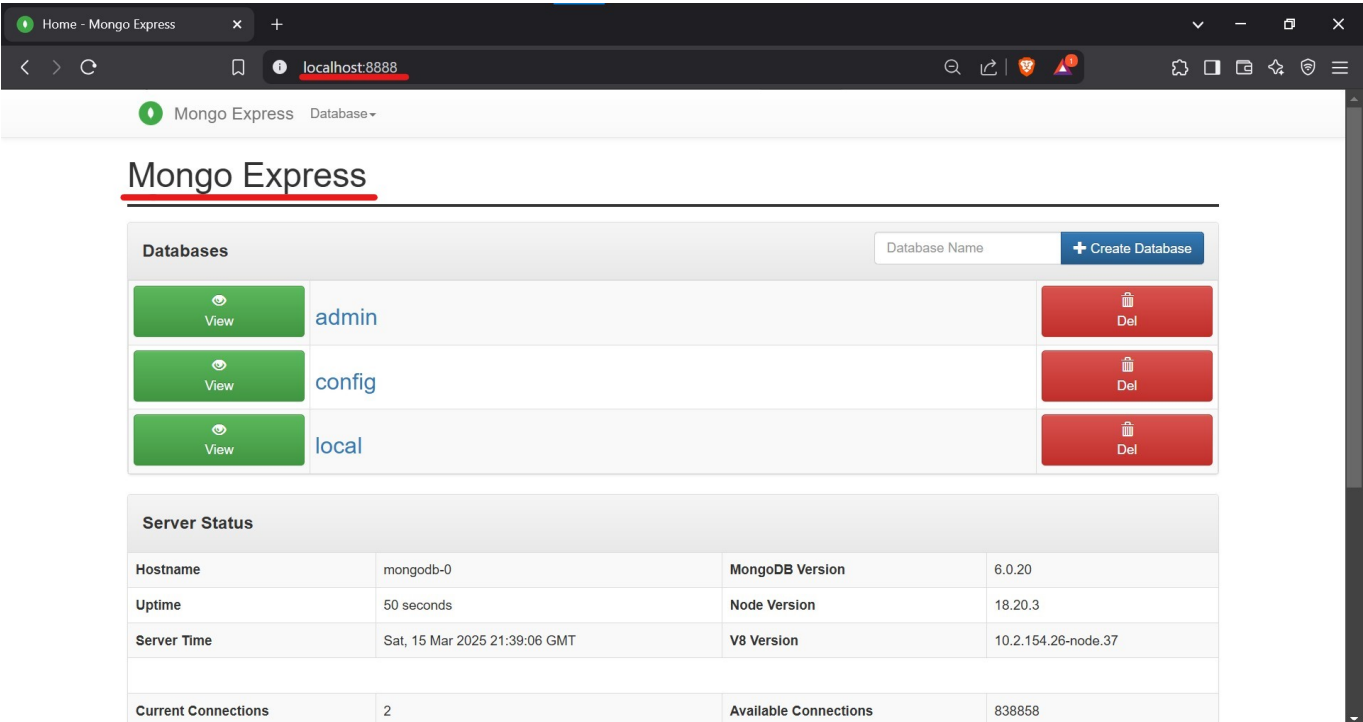
```
abdelhamed@DESKTOP-RT16ELH:~$ kubectl port-forward -n mongodb svc/mongo-express 8888:8081
Forwarding from 127.0.0.1:8888 -> 8081
Forwarding from [::1]:8888 -> 8081
Handling connection for 8888
Handling connection for 8888
Handling connection for 8888
```

- Open your browser and visit: **http://localhost:8888**
- Login using credentials: **admin / SecurePass123!**

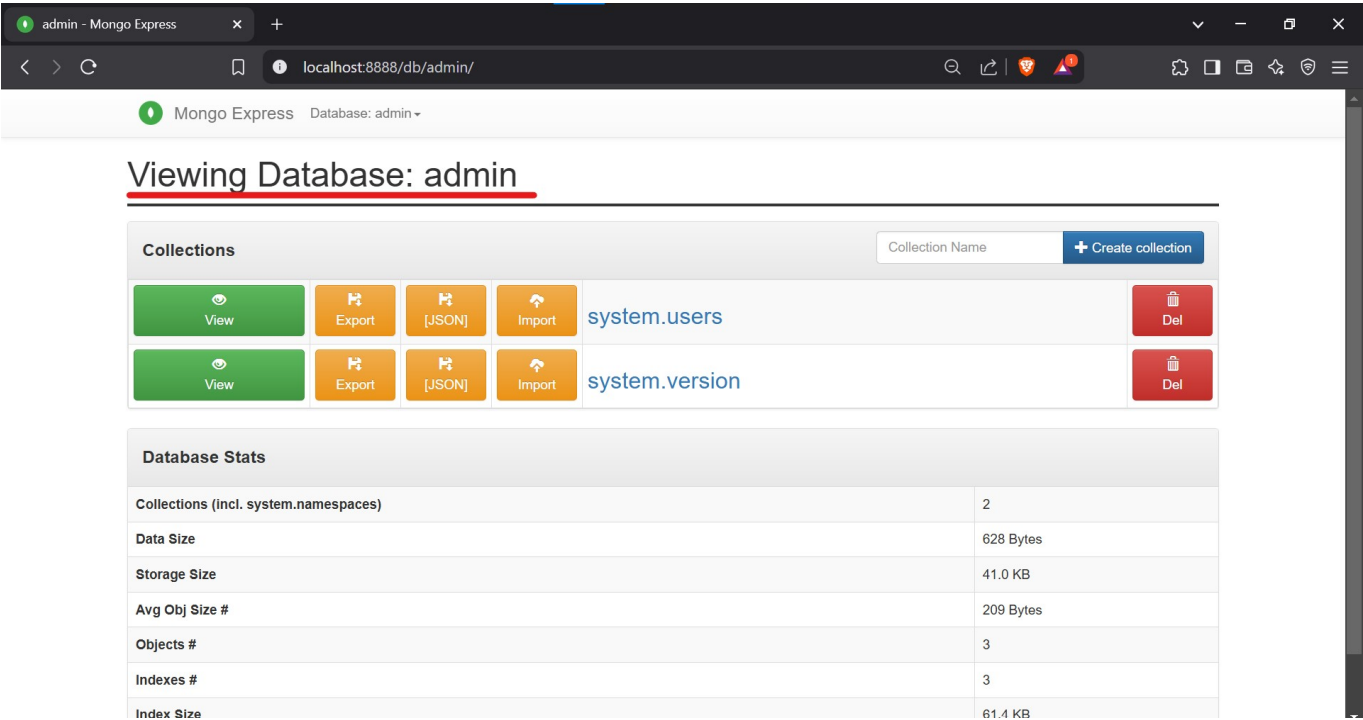
Login



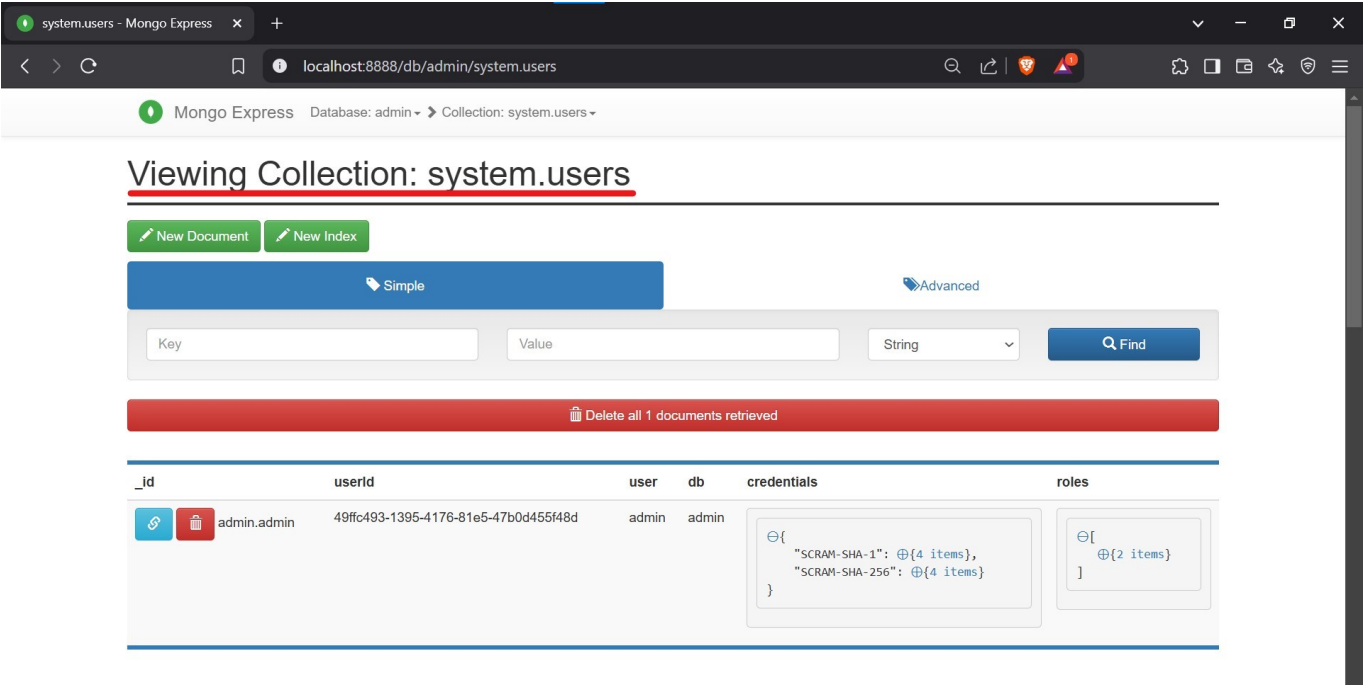
Dashboard



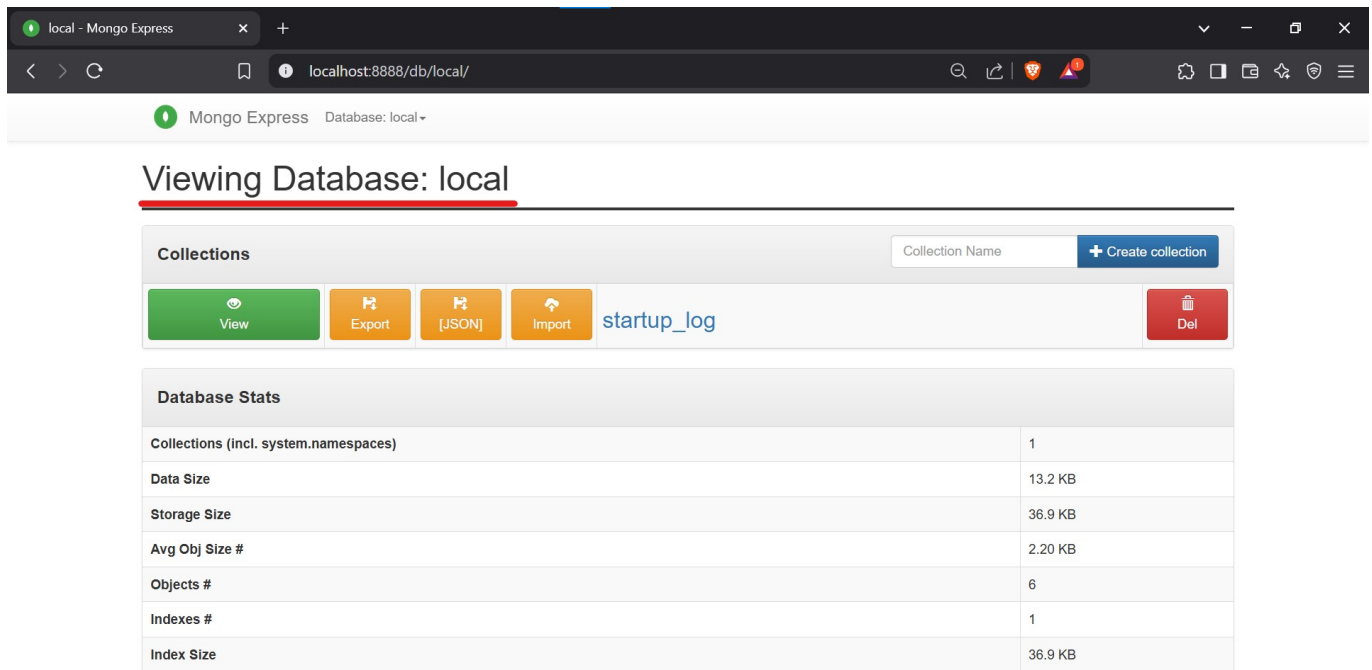
Viewing Admin Database



Viewing System Users



Viewing Local Database



Viewing Database: local

Collections

Collection Name [+ Create collection](#)

[View](#) [Export](#) [\[JSON\]](#) [Import](#) [startup_log](#) [Del](#)

Database Stats

Collections (incl. system.namespaces)	1
Data Size	13.2 KB
Storage Size	36.9 KB
Avg Obj Size #	2.20 KB
Objects #	6
Indexes #	1
Index Size	36.9 KB

🌐 Option 2: Test connectivity from the mongo-express pod

```
kubectl exec -n mongodb -it mongodb-0 -- mongosh -u admin -p SecurePass123!
```

```
abdelhamed@DESKTOP-RT16ELH:~$ kubectl exec -n mongodb -it mongodb-0 -- mongosh -u admin -p SecurePass123!
Current Mongosh Log ID: 67d5f65aa20eadf65e544ca6
Connecting to:      mongodb://<credentials>@127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&app
Name=mongosh+2.3.8
Using MongoDB:      6.0.20
Using Mongosh:      2.3.8

For mongosh info see: https://www.mongodb.com/docs/mongosh-shell/

-----
The server generated these startup warnings when booting
2025-03-15T21:38:16.337+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. S
ee http://dochub.mongodb.org/core/prodnotes-filesystem
2025-03-15T21:38:18.085+00:00: /sys/kernel/mm/transparent_hugepage/enabled is 'always'. We suggest setting it to 'nev
er' in this binary version
2025-03-15T21:38:18.085+00:00: vm.max_map_count is too low
-----

test> show dbs
admin 100.00 KiB
config 108.00 KiB
local 72.00 KiB
test>
```

🔍 Debugging

📄 Check Pod Status

```
kubectl get pods -n mongodb
```

📄 View Mongo Express Logs

```
kubectl logs -n mongodb -l app=mongo-express
```

Restart Mongo Express

```
kubectl delete pod -n mongodb -l app=mongo-express
```

Key Takeaways

✓ Uses **StatefulSet** for MongoDB persistence ✓ Stores sensitive data in **Kubernetes Secrets** ✓ Provides **Mongo Express** for easy database management ✓ Uses **Services** to enable internal and external communication ✓ Implements **authentication** for secure access
