SIT737 - Task 9.1P

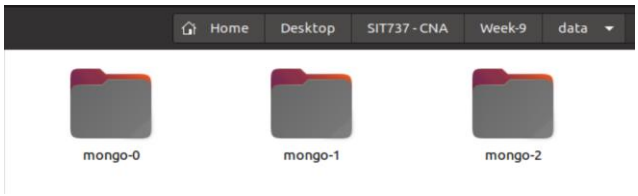
Adding a Database

Submitted By: Adeel Ahmed 224404186

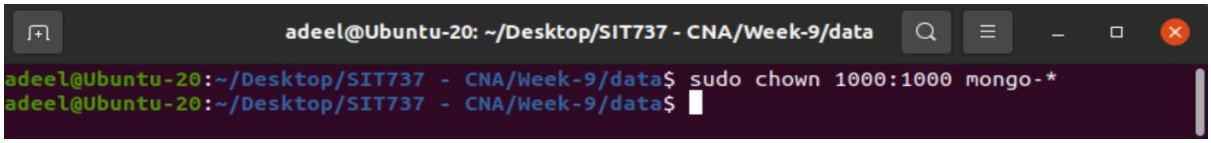
Github Link: https://github.com/AdeelAhmedIqbal/sit737-2025-prac7p-task9.1

1) Prepare Host Directories for MongoDB Data

Create directories for each MongoDB data volume:

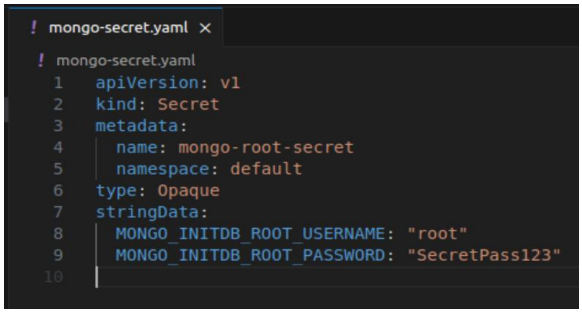


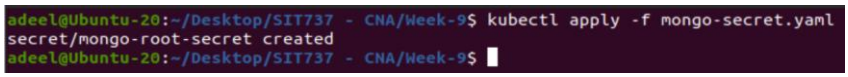
Ensure MongoDB (UID 1000) can write to them:



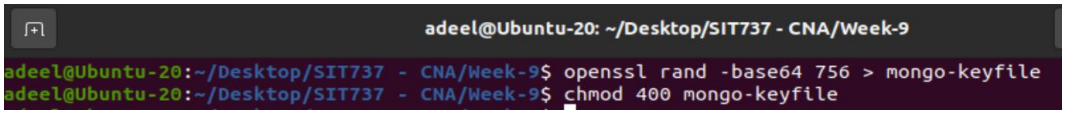
Note: Since the MongoDB container runs as a non-root user (UID 1000), you must set the hostPath directories to be owned by UID 1000 so the database can write data without permission errors.

2) Create a Kubernetes Secret to store the MongoDB root user credentials securely





Generate a key file because it is required when authorization is enabled with replica sets in MongoDB



Create a separate Secret for that key file

```
adeel@Ubuntu-20:~/Desktop/SIT737 - CNA/Week-9$ kubectl create secret generic mongo-keyfile-secret \
>    --from-file=keyfile=mongo-keyfile
secret/mongo-keyfile-secret created
adeel@Ubuntu-20:~/Desktop/SIT737 - CNA/Week-9$
```

3) Create a StorageClass

This step is optional, I do not really have to define a specific storageClassName as I am doing manual provisioning however consider this dummy StorageClass:

mongo-storageclass.yaml

```yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: local-hostpath
provisioner: kubernetes.io/no-provisioner
volumeBindingMode: WaitForFirstConsumer
```

```
adeel@Ubuntu-20:~/Desktop/SIT737 - CNA/Week-9$ kubectl apply -f mongo-storageclass.yaml
storageclass.storage.k8s.io/local-hostpath created
adeel@Ubuntu-20:~/Desktop/SIT737 - CNA/Week-9$
```

4) Define PersistentVolumes for each MongoDB Pod (hostPath)

mongo-pv.yaml

```yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: mongo-pv-0
spec:
  capacity:
    storage: 3Gi
  accessModes:
    - ReadWriteOnce
  storageClassName: "local-hostpath"
  hostPath:
    path: /data/mongo-repl/mongo-0
  persistentVolumeReclaimPolicy: Retain
---
apiVersion: v1
kind: PersistentVolume
metadata:
  name: mongo-pv-1
spec:
  capacity:
    storage: 3Gi
  accessModes:
    - ReadWriteOnce
  storageClassName: "local-hostpath"
  hostPath:
    path: /data/mongo-repl/mongo-1
  persistentVolumeReclaimPolicy: Retain
---
apiVersion: v1
kind: PersistentVolume
metadata:
  name: mongo-pv-2
spec:
  capacity:
    storage: 3Gi
  accessModes:
    - ReadWriteOnce
  storageClassName: "local-hostpath"
  hostPath:
    path: /data/mongo-repl/mongo-2
  persistentVolumeReclaimPolicy: Retain
```

```
adeel@Ubuntu-20:~/Desktop/SIT737 - CNA/Week-9$ kubectl apply -f mongo-pv.yaml
persistentvolume/mongo-pv-0 created
persistentvolume/mongo-pv-1 created
persistentvolume/mongo-pv-2 created
adeel@Ubuntu-20:~/Desktop/SIT737 - CNA/Week-9$
```

5) Create a Headless Service for MongoDB

A headless service gives each StatefulSet pod its own fixed DNS name, so the MongoDB replicas can find and connect to one another via these DNS records.

mongo-headless-service.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: mongo
  namespace: default
  labels:
    app: mongo
spec:
  ports:
    - port: 27017
      targetPort: 27017
  clusterIP: None
  selector:
    app: mongo
```

```
adeel@Ubuntu-20:~/Desktop/SIT737 - CNA/Week-9$ kubectl apply -f mongo-headless-service.yaml
service/mongo created
adeel@Ubuntu-20:~/Desktop/SIT737 - CNA/Week-9$
```

6) Deploy the MongoDB StatefulSet with 3 replicas

mongo-statefulset.yaml

```yaml
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: mongo
  namespace: default
spec:
  serviceName: mongo
  replicas: 3
  selector:
    matchLabels:
      app: mongo
  template:
    metadata:
      labels:
        app: mongo
    spec:
      # Mount the keyFile secret as /etc/mongo/keyfile (mode 0400, owner root)
      volumes:
        - name: keyfile-vol
          secret:
            secretName: mongo-keyfile-secret
            defaultMode: 256    # decimal for 0400
```

```yaml
  containers:
  - name: mongod
    image: mongo:4.4           # Use MongoDB 4.4 image (compatible with older CPU)
    ports:
    - containerPort: 27017
      name: mongo
    env:
    - name: MONGO_INITDB_ROOT_USERNAME
      valueFrom:
        secretKeyRef:
          name: mongo-root-secret
          key: MONGO_INITDB_ROOT_USERNAME
    - name: MONGO_INITDB_ROOT_PASSWORD
      valueFrom:
        secretKeyRef:
          name: mongo-root-secret
          key: MONGO_INITDB_ROOT_PASSWORD
    args:
      - "--replSet=rs0" # --replSet=rs0 sets the replica set name to "rs0" for all members.
      - "--bind_ip_all" # --bind_ip_all makes MongoDB listen on all network interfaces (required for intra-cluster connectivity).
      - "--keyFile=/etc/mongo/keyfile"
    volumeMounts:
    - name: mongo-data
      mountPath: /data/db      # Mount the persistent volume to MongoDB's data directory
    - name: keyfile-vol
      mountPath: /etc/mongo
      readOnly: true

volumeClaimTemplates:
- metadata:
    name: mongo-data
  spec:
    accessModes: [ "ReadWriteOnce" ]
    resources:
      requests:
        storage: 3Gi
    storageClassName: "local-hostpath"
```

```
adeel@Ubuntu-20:~/Desktop/SIT737 - CNA/Week-9$ kubectl apply -f mongo-statefulset.yaml
statefulset.apps/mongo created
adeel@Ubuntu-20:~/Desktop/SIT737 - CNA/Week-9$ k get pods
NAME      READY   STATUS              RESTARTS   AGE
mongo-0   1/1     Running             0          5s
mongo-1   1/1     Running             0          4s
mongo-2   0/1     ContainerCreating   0          1s
adeel@Ubuntu-20:~/Desktop/SIT737 - CNA/Week-9$ k get pods
NAME      READY   STATUS    RESTARTS   AGE
mongo-0   1/1     Running   0          9s
mongo-1   1/1     Running   0          8s
mongo-2   1/1     Running   0          5s
```

7) Initialize the MongoDB Replica Set (rs.initiate)

The three MongoDB instances are running but they are not yet configured as a replica set (they start in standalone mode waiting for initiation). We need to initiate the replica set and add members.

```
adeel@Ubuntu-20:~/Desktop/SIT737 - CNA/Week-9$ kubectl exec -it mongo-0 -- mongo -u root -p SecretPass123 --authenticationDatabase admin
MongoDB shell version v4.4.29
connecting to: mongodb://127.0.0.1:27017/?authSource=admin&compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("d66b0d6c-1a89-40e4-888e-d00f39fcdd5b") }
MongoDB server version: 4.4.29
---
The server generated these startup warnings when booting:
        2025-05-13T07:11:24.426+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://doc
em
        2025-05-13T07:11:25.157+00:00: You are running this process as the root user, which is not recommended
---
> rs.initiate({
...   _id: "rs0",
...   members: [
...     { _id: 0, host: "mongo-0.mongo.default.svc.cluster.local:27017" },
...     { _id: 1, host: "mongo-1.mongo.default.svc.cluster.local:27017" },
...     { _id: 2, host: "mongo-2.mongo.default.svc.cluster.local:27017" }
...   ]
... })
{ "ok" : 1 }
rs0:SECONDARY>
```

Verify the replica set status using rs.status()

```
rs0:PRIMARY> rs.status().members.forEach(function(m) {
...   print(m.name + " | " + m.stateStr + " | health: " + m.health);
... });
mongo-0.mongo.default.svc.cluster.local:27017 | PRIMARY | health: 1
mongo-1.mongo.default.svc.cluster.local:27017 | SECONDARY | health: 1
mongo-2.mongo.default.svc.cluster.local:27017 | SECONDARY | health: 1
rs0:PRIMARY>
```

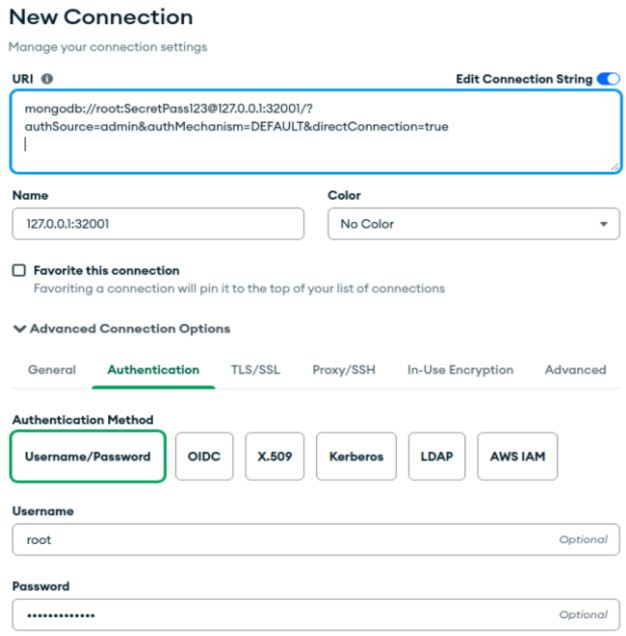
8) Launch Compass & Create a New Connection

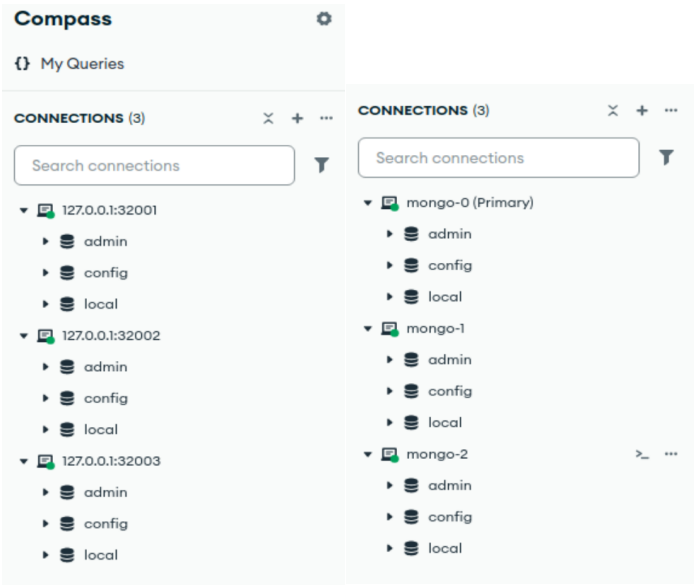
Port-Forward the MongoDB Service

```
adeel@Ubuntu-20:~/Desktop/SIT737 - CNA/Week-9$ kubectl port-forward pod/mongo-0 32001:27017
Forwarding from 127.0.0.1:32001 -> 27017
Forwarding from [::1]:32001 -> 27017
```

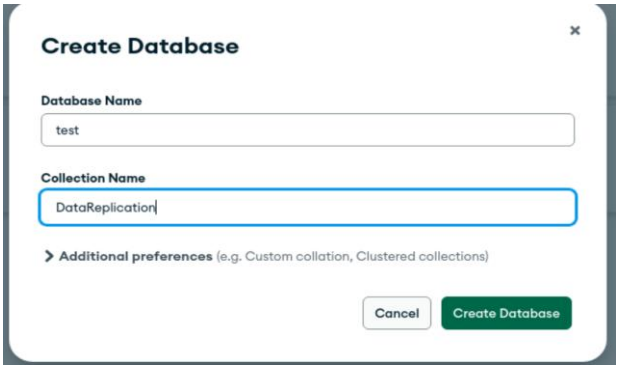
```
adeel@Ubuntu-20:~/Desktop/SIT737 - CNA/Week-9$ kubectl port-forward pod/mongo-1 32002:27017
Forwarding from 127.0.0.1:32002 -> 27017
Forwarding from [::1]:32002 -> 27017
```

```
adeel@Ubuntu-20:~/Desktop/SIT737 - CNA/Week-9$ kubectl port-forward pod/mongo-2 32003:27017
Forwarding from 127.0.0.1:32003 -> 27017
Forwarding from [::1]:32003 -> 27017
```

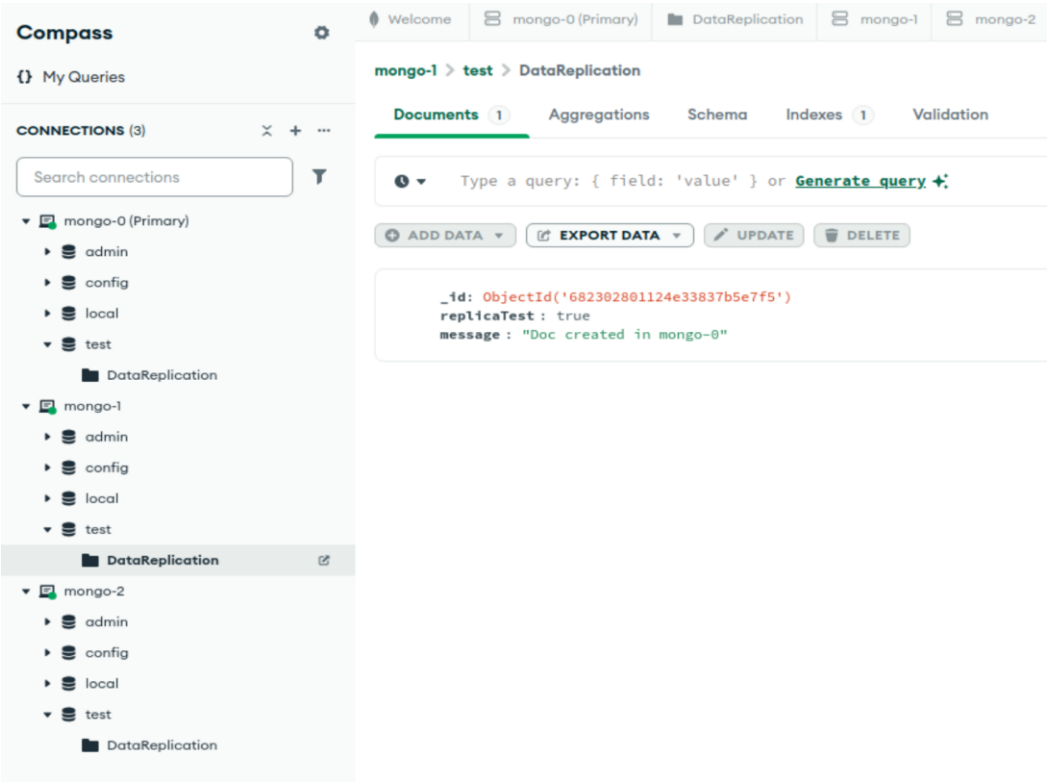




9) Test Data Replication



Hit Refresh on mongo-1 and mongo-2 to see the updates



10) Test Failover

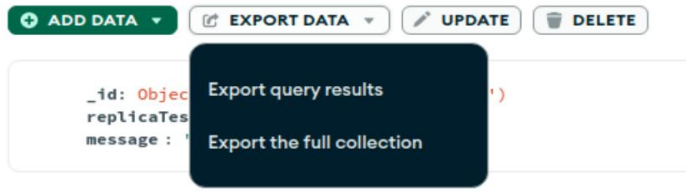
Step Down the Primary

Mongo-0 is no longer the primary and Mongo-1 is the new primary

```
adeel@Ubuntu-20:~/Desktop/SIT737 - CNA/Week-9$ kubectl exec -it mongo-0 -- mongo -u root -p SecretPass123 --authenticationDatabase admin
MongoDB shell version v4.4.29
connecting to: mongodb://127.0.0.1:27017/?authSource=admin&compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("c468a7c9-5ea8-49cc-9dc0-3b99e0e69232") }
MongoDB server version: 4.4.29
---
The server generated these startup warnings when booting:
        2025-05-13T07:11:24.426+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://do
em
        2025-05-13T07:11:25.157+00:00: You are running this process as the root user, which is not recommended
---
rs0:SECONDARY> rs.status().members.forEach(function(m) {   print(m.name + " | " + m.stateStr + " | health: " + m.health); });
mongo-0.mongo.default.svc.cluster.local:27017 | SECONDARY | health: 1
mongo-1.mongo.default.svc.cluster.local:27017 | PRIMARY | health: 1
mongo-2.mongo.default.svc.cluster.local:27017 | SECONDARY | health: 1
rs0:SECONDARY>
```

11) Backup & Basic Monitoring

Export a Collection (Backup)



View Metrics (Basic Monitoring) using performance tab in Compass