

# 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:

```
adeel@Ubuntu-20: ~/Desktop/SIT737 - CNA/Week-9/data
adeel@Ubuntu-20:~/Desktop/SIT737 - CNA/Week-9/data$ sudo chown 1000:1000 mongo-*
adeel@Ubuntu-20:~/Desktop/SIT737 - CNA/Week-9/data$
```

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

```
! mongo-secret.yaml x
! mongo-secret.yaml
1  apiVersion: v1
2  kind: Secret
3  metadata:
4    name: mongo-root-secret
5    namespace: default
6  type: Opaque
7  stringData:
8    MONGO_INITDB_ROOT_USERNAME: "root"
9    MONGO_INITDB_ROOT_PASSWORD: "SecretPass123"
10
```

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

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

```
adeel@Ubuntu-20:~/Desktop/SIT737 - CNA/Week-9$ openssl rand -base64 756 > mongo-keyfile
adeel@Ubuntu-20:~/Desktop/SIT737 - CNA/Week-9$ chmod 400 mongo-keyfile
```

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 x
! mongo-storageclass.yaml
1  apiVersion: storage.k8s.io/v1
2  kind: StorageClass
3  metadata:
4    name: local-hostpath
5  provisioner: kubernetes.io/no-provisioner
6  volumeBindingMode: WaitForFirstConsumer
7

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 x
! mongo-pv.yaml
1  apiVersion: v1
2  kind: PersistentVolume
3  metadata:
4    name: mongo-pv-0
5  spec:
6    capacity:
7      storage: 3Gi
8    accessModes:
9      - ReadWriteOnce
10   storageClassName: "local-hostpath"
11   hostPath:
12     path: /data/mongo-repl/mongo-0
13     persistentVolumeReclaimPolicy: Retain
14   ---
15   apiVersion: v1
16   kind: PersistentVolume
17   metadata:
18     name: mongo-pv-1
19   spec:
20     capacity:
21       storage: 3Gi
22     accessModes:
23       - ReadWriteOnce
24     storageClassName: "local-hostpath"
25     hostPath:
26       path: /data/mongo-repl/mongo-1
27     persistentVolumeReclaimPolicy: Retain
28   ---
29   apiVersion: v1
30   kind: PersistentVolume
31   metadata:
32     name: mongo-pv-2
33   spec:
34     capacity:
35       storage: 3Gi
36     accessModes:
37       - ReadWriteOnce
38     storageClassName: "local-hostpath"
39     hostPath:
40       path: /data/mongo-repl/mongo-2
41     persistentVolumeReclaimPolicy: Retain
42

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 x
! mongo-headless-service.yaml
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: mongo
5    namespace: default
6    labels:
7      app: mongo
8  spec:
9    ports:
10     - port: 27017
11       targetPort: 27017
12    clusterIP: None
13    selector:
14      app: mongo
15
```

```
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 x  ! mongo-secret.yaml
! mongo-statefulset.yaml
1  apiVersion: apps/v1
2  kind: StatefulSet
3  metadata:
4    name: mongo
5    namespace: default
6  spec:
7    serviceName: mongo
8    replicas: 3
9    selector:
10     matchLabels:
11       app: mongo
12    template:
13     metadata:
14       labels:
15         app: mongo
16     spec:
17       # Mount the keyFile secret as /etc/mongo/keyfile (mode 0400, owner root)
18       volumes:
19         - name: keyfile-vol
20           secret:
21             secretName: mongo-keyfile-secret
22             defaultMode: 256 # decimal for 0400
23
24 containers:
25   - name: mongod
26     image: mongo:4.4 # Use MongoDB 4.4 image (compatible with older CPU)
27     ports:
28       - containerPort: 27017
29       name: mongo
30     env:
31       - name: MONGO_INITDB_ROOT_USERNAME
32         valueFrom:
33           secretKeyRef:
34             name: mongo-root-secret
35             key: MONGO_INITDB_ROOT_USERNAME
36       - name: MONGO_INITDB_ROOT_PASSWORD
37         valueFrom:
38           secretKeyRef:
39             name: mongo-root-secret
40             key: MONGO_INITDB_ROOT_PASSWORD
41     args:
42       - "--replSet=rs0" # --replSet=rs0 sets the replica set name to "rs0" for all members.
43       - "--bind_ip all" # --bind_ip all makes MongoDB listen on all network interfaces (required for intra-cluster connectivity).
44       - "--keyFile=/etc/mongo/keyfile"
45     volumeMounts:
46       - name: mongo-data
47         mountPath: /data/db # Mount the persistent volume to MongoDB's data directory
48       - name: keyfile-vol
49         mountPath: /etc/mongo
50         readOnly: true
51
52 volumeClaimTemplates:
53   - metadata:
54     name: mongo-data
55     spec:
56       accessModes: [ "ReadWriteOnce" ]
57       resources:
58         requests:
59           storage: 3Gi
60       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-d00f39fcd5b") }
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://docs.mongodb.org/manual/tutorial/convert-file-system-to-xfs/
  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
```

## New Connection

Manage your connection settings

URI  Edit Connection String 

`mongodb://root:SecretPass123@127.0.0.1:32001/?authSource=admin&authMechanism=DEFAULT&directConnection=true`

Name  Color

☐ **Favorite this connection**  
Favoriting a connection will pin it to the top of your list of connections

▼ **Advanced Connection Options**

General

**Authentication**

TLS/SSL

Proxy/SSH

In-Use Encryption

Advanced

**Authentication Method**

Username/Password

OIDC

X.509

Kerberos

LDAP

AWS IAM

Username  Optional

Password  Optional

### Compass

My Queries

**CONNECTIONS (3)**

- 127.0.0.1:32001
  - admin
  - config
  - local
- 127.0.0.1:32002
  - admin
  - config
  - local
- 127.0.0.1:32003
  - admin
  - config
  - local

**CONNECTIONS (3)**

- mongo-0 (Primary)
  - admin
  - config
  - local
- mongo-1
  - admin
  - config
  - local
- mongo-2
  - admin
  - config
  - local

## 9) Test Data Replication

### Create Database

Database Name

Collection Name

➤ **Additional preferences** (e.g. Custom collation, Clustered collections)

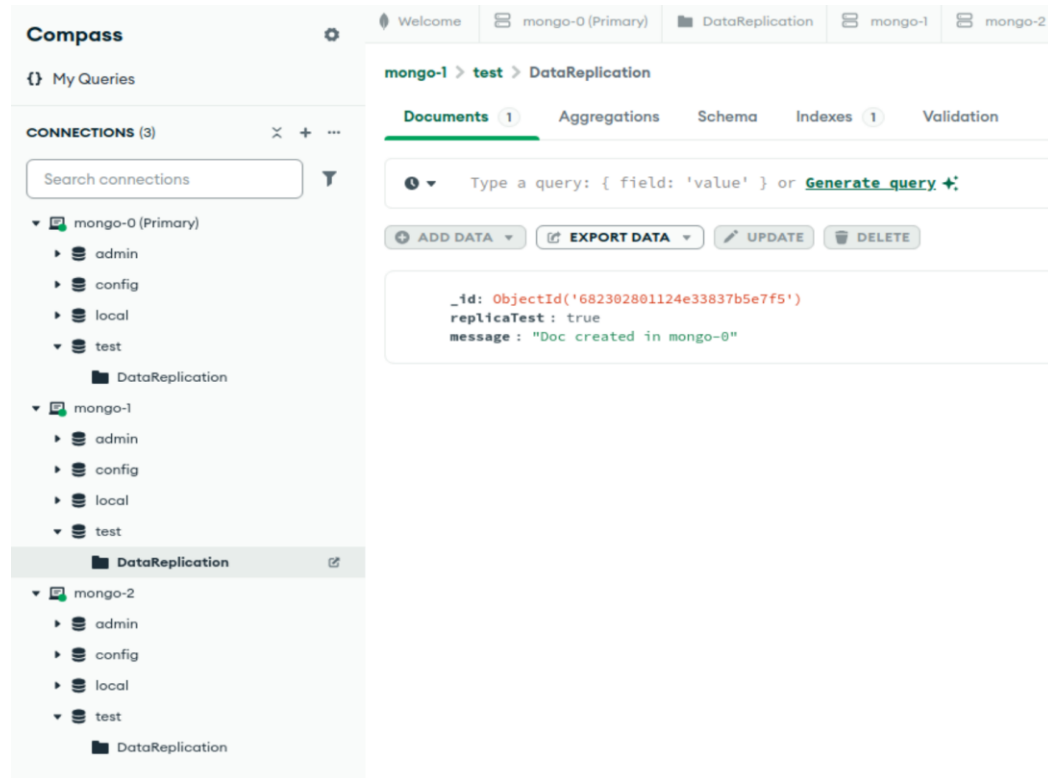
## Insert Document

To collection test.DataReplication

VIEW  

```
1 {
2   "replicaTest": true,
3   "message": "Doc created in mongo-0"
4 }
5
```

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



The screenshot shows the MongoDB Compass interface. On the left, the 'CONNECTIONS (3)' panel lists three nodes: mongo-0 (Primary), mongo-1, and mongo-2. Each node has a 'test' database, and under 'test', there is a 'DataReplication' collection. The 'DataReplication' collection is selected. The main panel shows the 'Documents' tab for the 'DataReplication' collection. It displays a single document with the following fields: `_id`: ObjectId('682302801124e33837b5e7f5'), `replicaTest`: true, and `message`: "Doc created in mongo-0". The document is shown in a light blue box.

## 10) Test Failover

Step Down the Primary

```
> rs.stepDown()
< {
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1747125089, i: 5 }),
    signature: {
      hash: Binary.createFromBase64('LXy5wvWzUd0p/5ZsbsUgdUUrCGH=', 0),
      keyId: Long('7503826792650637317')
    }
  },
  operationTime: Timestamp({ t: 1747125089, i: 5 })
}
rs0 [direct: secondary] test>
```

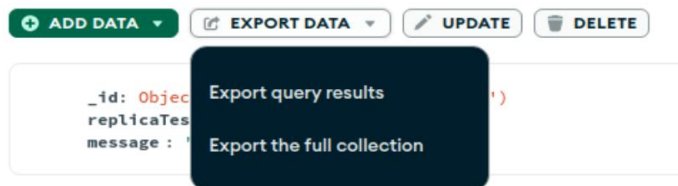


Mongo-0 is no longer the primary and Mongo-1 is the new primary

```
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://docs.mongodb.org/manual/tutorial/convert-file-system-to-xfs/
  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)



```
{ } test.DataReplication.json x ! mongo-statefulset.yaml ●
home > adeel > Downloads > { } test.DataReplication.json > ...
1  [ {
2    "_id": {
3      "$oid": "682302801124e33837b5e7f5"
4    },
5    "replicaTest": true,
6    "message": "Doc created in mongo-0"
7  } ]
```

View Metrics (Basic Monitoring) using performance tab in Compass

