Instructions
1. Deploy MongoDB using a Kubernetes configuration

Create a file named mongodb.yaml and add the following content to it:

apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2

Then apply the configuration file –

$ kubectl apply -f mongodb.yaml

```
└─$ kubectl apply -f mongodb.yaml
deployment.apps/mongodb created
```

2. Create a Persistent Volume and Persistent Volume Claim

Then apply the configuration file –

$ kubectl apply -f pv.yaml

```
└─$ kubectl apply -f pv.yaml
persistentvolume/mongodb-pv-volume created
```

2. Create a file named pvc.yaml and add the following content to it:

Then apply the configuration file –

$ kubectl apply -f pvc.yaml

```
└─$ kubectl apply -f pvc.yaml
persistentvolumeclaim/mongodb-pv-claim created
```

3. Create a Secret for the MongoDB User Credentials

Create a file named secret.yaml and add the following content to it:

```
apiVersion: v1
kind: Secret
metadata:
  name: mongodb-secret
type: Opaque
stringData:
  MONGO_USERNAME: <username>        # set the MongoDB username
  MONGO_PASSWORD: <password>        # set the MongoDB password
```

Then apply the configuration file –

$ kubectl apply -f secret.yaml

4.
```
└─$ kubectl apply -f secret.yaml
secret/mongodb-secret created
```

Modify the Deployment Manifest

Update the deployment manifest to include the secret and the MongoDB database parameters.

Then apply the configuration file –

$ kubectl apply -f deployment.yaml

```
└─$ kubectl apply -f deployment.yaml
deployment.apps/mongo-operator created
```

5. Configure Application to Connect to MongoDB

Update the application code to use the MongoDB credentials and connection string from the deployment manifest.

Launch the Kubernetes dashboard.

```
└─$ minikube dashboard
🤔  Verifying dashboard health ...
🚀  Launching proxy ...
🤔  Verifying proxy health ...
```

6. Test the Deployment

Test the deployment to ensure that the application can connect to the MongoDB database and perform basic CRUD (Create, Read, Update, Delete) operations.

```
└─$ kubectl get pods
NAME                                READY   STATUS             RESTARTS      AGE
hello-minikube-77b6f68484-d4shz     1/1     Running            1 (17m ago)   176m
mongo-operator-57fd747d4d-jwsxq     0/1     ImagePullBackOff   0             57m
mongodb-6464b585f5-r7g69            1/1     Running            1 (17m ago)   158m
```

7. Set Up Database Backups and Disaster Recovery

Utilise tools like mongodump or mongo-backup-utils to create a backup of your MongoDB database, and set up automated backups for the duration of your choosing. Additionally, to make sure that your data is safe in case of an emergency, you can set up disaster recovery procedures like replication or replica sets.

8. Monitor Performance

To make sure the MongoDB database is operating smoothly and effectively, track the performance of both the application and the database. To keep an eye on your MongoDB instance's performance, you can utilise monitoring tools like Atlas from MongoDB or MongoDB Ops Manager. In addition, you can track the performance of your application using application performance monitoring (APM) tools like New Relic or AppDynamic.