**Step involved:**

1. Downloaded helm , MongoDB from (<https://helm.sh/>) and (<https://www.mongodb.com/>)
2. Create a MongoDB Deployment:

* Use a MongoDB Docker image to create a deployment in Kubernetes.

1. Use a MongoDB Operator or Helm Chart:

* For a replica set, it's recommended to use MongoDB Kubernetes Operator or a Helm chart which simplifies managing a MongoDB replica set.
* Configure the replica set parameters according to your requirements.

4.: Configure Persistent Storage

* Create a Persistent Volume (PV):
* Define a PV in your cluster that matches your storage requirements.

5. Create a Persistent Volume Claim (PVC):

* Create a PVC that requests storage from the PV.

6. Create MongoDB User

* Connect to MongoDB Pod:
* Use kubectl exec to access the MongoDB shell inside your pod.

7. Create Kubernetes Secret for MongoDB Credentials

* Create a Secret:
* Use kubectl create secret to store the MongoDB user credentials securely in the cluster.

8: Modify Kubernetes Deployment Manifest

* Add MongoDB Environment Variables:
* Modify your application’s deployment manifest to include MongoDB credentials as environment variables sourced from the secret.

9. Specify database connection parameters.

10. Configure Application Connection

11.Update Application Configuration:

* Include the MongoDB client library in your application.
* Configure the application to use the connection string defined in the Kubernetes deployment manifest to connect to MongoDB.

12. Test the Deployment

* Deploy Your Application:
* Use kubectl apply to deploy your application.

