```
FROM openjdk:11-jre-slim

COPY target/my-application.jar /app.jar

CMD [“java”, “-jar”, “/app.jar”]
Docker build -t my-application .
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mongo
spec:
  selector:
    matchLabels:
      app: mongo
  replicas: 1
  template:
    metadata:
      labels:
        app: mongo
    spec:
      containers:
        -     Name: mongo
          Image: mongo
          Ports:
        -     containerPort: 27017
          volumeMounts:
        -     name: mongo-data
              mountPath: /data/db
      volumes:
```

```
    - name: mongo-data
      persistentVolumeClaim:
        claimName: mongo-pvc
```

```
apiVersion: v1
kind: Service
metadata:
  name: mongo
spec:
  selector:
    app: mongo
  ports:
    - name: mongo
      port: 27017
      targetPort: 27017
```

```
kubectl apply -f mongo-deployment.yaml
kubectl get pods
apiVersion: v1
kind: PersistentVolume
metadata:
  name: mongo-pv
spec:
  storageClassName: manual
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: “/mnt/data/mongo”
```

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mongo-pvc
spec:
  storageClassName: manual
  accessModes:
      -    ReadWriteOnce
  Resources:
    Requests:
      Storage: 1Gi
Kubectl apply -f mongo-pv.yaml
Kubectl apply -f mongo-pvc.yaml
Kubectl get pv,pvc
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-application
spec:
  selector:
    matchLabels:
      app: my-application
  replicas: 1
  template:
    metadata:
      labels:
        app: my-application
    spec:
      containers:
```

- Name: my-application

  Image: my-application

  Env:

- Name: MONGO_HOST

    Value: mongo

- Name: MONGO_PORT

    Value: “27017”

  Ports:

- containerPort: 808