Problem Statement

You are working as a Devops Administrator. You’ve been tasked to deploy a multitier application on Kubernetes Cluster. The application is a NodeJS application available on Docker Hub with the following name: devopsedu/employee This NodeJS application works with a mongo database. MongoDB image is available on DockerHub with the following name: mongo You are required to deploy this application on Kubernetes: • NodeJS is available on port 8888 in the container and will be reaching out to port 27017 for mongo database connection • MongoDB will be accepting connections on port 27017 You must deploy this application using the CLI. Once your application is up and running, ensure you can add an employee from the NodeJS application and verify by going to Get Employee page and retrieving your input. Hint: Name the Mongo DB Service and deployment, specifically as “mongo”.

**Tried to run docker container independently to ensure access:**

docker run -d --name emp1 -p 8888:8888 devopsedu/employee

docker run -d --name mongo -p 23017:27017 mongo

**Took reference from here:**

<https://yasharma06.medium.com/deploy-locally-nodejs-and-mongodb-application-locally-c1bad2308d11>

<https://learnk8s.io/deploying-nodejs-kubernetes>

**Solution from here:**

Create a file kemplyee.yaml

apiVersion: v1

kind: Service

metadata:

  name: kemployee

spec:

  selector:

    app: kemployee

  ports:

    - port: 8888

      targetPort: 8888

  type: LoadBalancer

---

apiVersion: apps/v1

kind: Deployment

metadata:

  name: kemployee

spec:

  replicas: 1

  selector:

    matchLabels:

      app: kemployee

  template:

    metadata:

      labels:

        app: kemployee

    spec:

      containers:

        - name: app

          image: devopsedu/employee

          ports:

            - containerPort: 8888

          env:

            - name: MONGO\_URL

              value: mongodb://mongo:27017/dev

          imagePullPolicy: Always

created another file mongo.yaml

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

  name: mongo-pvc

spec:

  accessModes:

    - ReadWriteOnce

  resources:

    requests:

      storage: 256Mi

---

apiVersion: v1

kind: Service

metadata:

  name: mongo

spec:

  selector:

    app: mongo

  ports:

    - port: 27017

      targetPort: 27017

---

apiVersion: apps/v1

kind: Deployment

metadata:

  name: mongo

spec:

  selector:

    matchLabels:

      app: mongo

  template:

    metadata:

      labels:

        app: mongo

    spec:

      containers:

        - name: mongo

          image: mongo:6.0.2-focal

          ports:

            - containerPort: 27017

          volumeMounts:

            - name: storage

              mountPath: /data/db

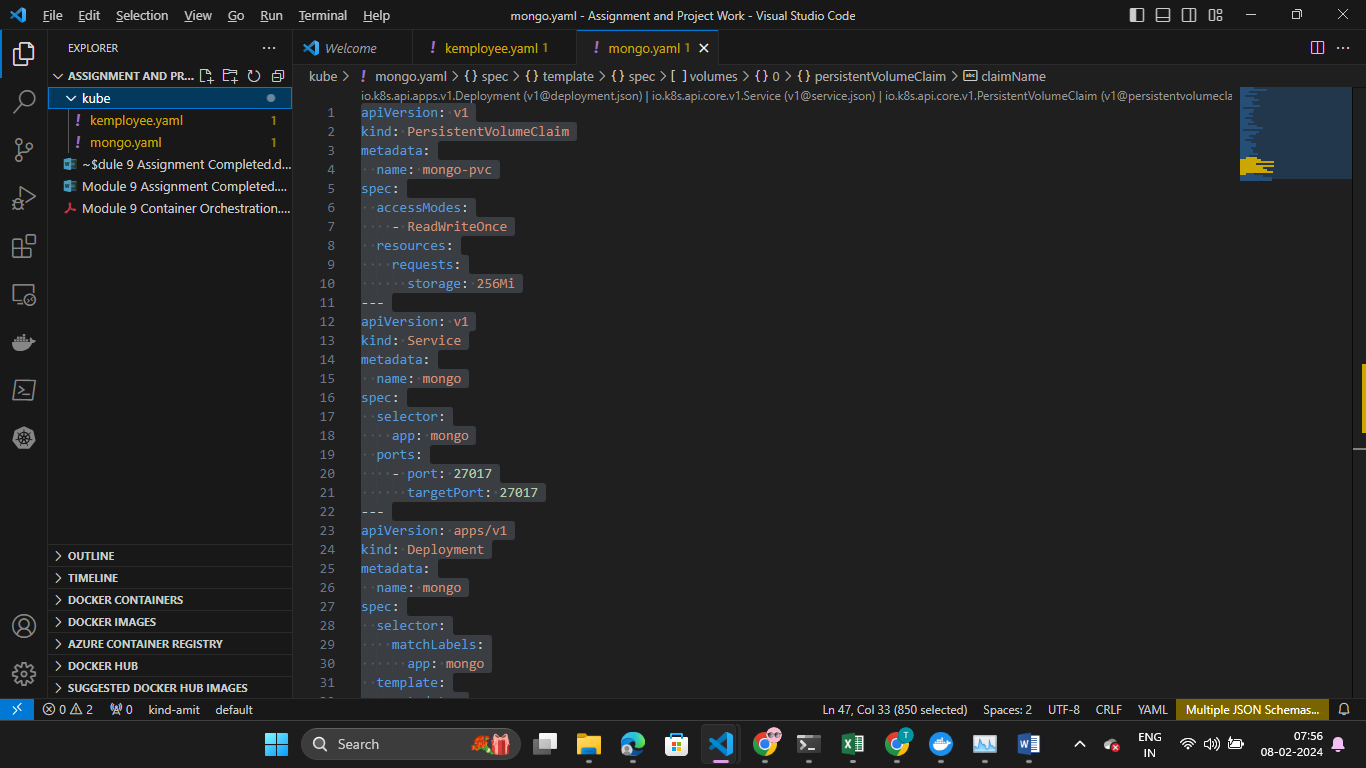
      volumes:

        - name: storage

          persistentVolumeClaim:

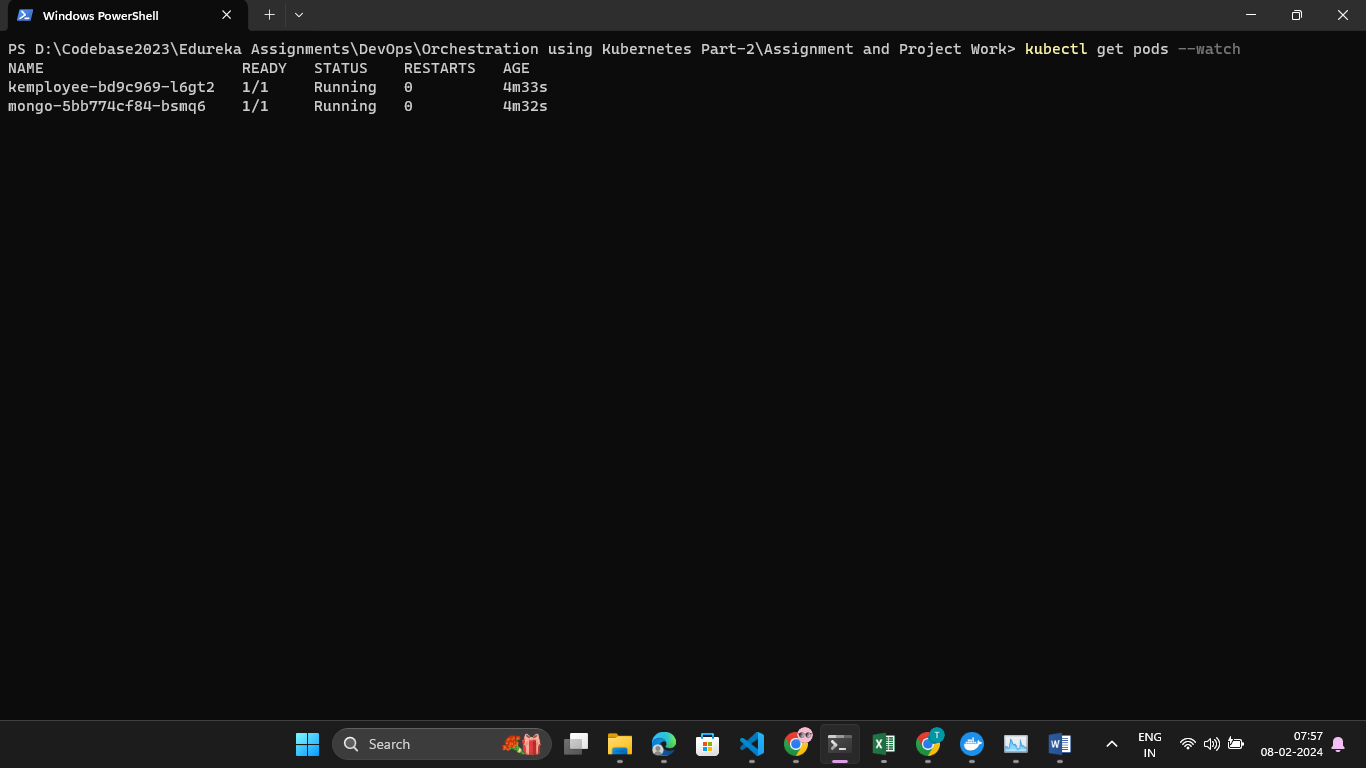
            claimName: mongo-pvc

both files are placed in a folder named kube:



Now running following command to create this configuration:

kubectl apply -f kube



Now services are running: