DockerFile for app:

FROM docker.io/tomcat:10-jdk17-openjdk

WORKDIR /usr/local/tomcat/webapps

COPY target/scc2223-project-1.0.war ROOT.war

EXPOSE 8080

Kubernetes deployment files:

redis.yaml (Deployment do pod do redis e criação do seu serviço):

apiVersion: apps/v1

kind: Deployment

metadata:

  name: redis-back

spec:

  replicas: 1

  selector:

    matchLabels:

      app: redis-back

  template:

    metadata:

      labels:

        app: redis-back

    spec:

      nodeSelector:

        "kubernetes.io/os": linux

      containers:

      - name: redis-back

        image: redis

        env:

        - name: ALLOW\_EMPTY\_PASSWORD

          value: "yes"

        resources:

          requests:

            cpu: 100m

            memory: 128Mi

          limits:

            cpu: 250m

            memory: 256Mi

        ports:

        - containerPort: 6379

          name: redis

---

apiVersion: v1

kind: Service

metadata:

  name: redis-back

spec:

  ports:

  - port: 6379

  selector:

    app: redis-back

mongo-configmap.yaml (configmap com o url da MongoDB):

apiVersion: v1

kind: ConfigMap

metadata:

  name: mongodb-configmap

data:

  database\_url: mongodb-service

mongo.yaml (deployment do pod da mongodb e criação do seu serviço):

# MONGODB POD DEPLOYMENT

apiVersion: apps/v1

kind: Deployment

metadata:

  name: mongodb-deployment

  labels:

    app: mongodb

spec:

  replicas: 1

  selector:

    matchLabels:

      app: mongodb

  template:

    metadata:

      labels:

        app: mongodb

    spec:

      containers:

      - name: mongodb

        image: mongo

        ports:

        - containerPort: 27017

---

# MONGODB SERVICE

apiVersion: v1

kind: Service

metadata:

  name: mongodb-service

spec:

  ports:

  - port: 27017

    targetPort: 27017

  selector:

    app: mongodb

mongo-express.yaml (frontend para gestão facilitada da base de dados):

apiVersion: apps/v1

kind: Deployment

metadata:

  name: mongodb-express-deployment

  labels:

    app: mongodb-express

spec:

  replicas: 1

  selector:

    matchLabels:

      app: mongodb-express

  template:

    metadata:

      labels:

        app: mongodb-express

    spec:

      containers:

        - name: mongodb-express

          image: mongo-express

          ports:

            - containerPort: 8081

          env:

            - name: ME\_CONFIG\_MONGODB\_SERVER

              valueFrom:

                configMapKeyRef:

                  name: mongodb-configmap

                  key: database\_url

---

apiVersion: v1

kind: Service

metadata:

  name: mongodb-express-service

spec:

  selector:

    app: mongodb-express

  type: LoadBalancer # Uncomment for external service

  ports:

    - protocol: TCP

      port: 8081

      targetPort: 8081

      nodePort: 30000 # Uncomment for external service

volume-claim.yaml (deploy do persistente volume claim + persistente volume e criação do seu serviço):

##pvc.yaml

kind: PersistentVolumeClaim

apiVersion: v1

metadata:

  name: azure-managed-disk

  namespace: default

spec:

  accessModes:

    - ReadWriteOnce

  resources:

    requests:

      storage: 5Gi

  storageClassName: azurefile