1. **Create a MySQL pod with Stateful Set with 1 replica**

apiVersion: apps/v1

kind: StatefulSet

metadata:

  name: mysql-st

  labels:

    env: uat

spec:

  minReadySeconds: 10

  replicas: 2

  selector:

    matchLabels:

      env: uat

  template:

    metadata:

      name: mysql-st

      labels:

        env: uat

    spec:

      containers:

        - name: mysql

          image: mysql:8.0

          volumeMounts:

            - mountPath: /var/lib/mysql

              name: nop-sql

          ports:

            - name: mysql-st

              containerPort: 3306

              protocol: TCP

          env:

            - name: MYSQL\_ROOT\_PASSWORD

              value: rootroot

            - name: MYSQL\_USER

              value: Su

            - name: MYSQL\_PASSWORD

              value: rootroot

            - name: MYSQL\_DATABASE

              value: Persons

            # - MYSQL\_ROOT\_PASSWORD: rootroot

            #   MYSQL\_DATABASE: Persons

            #   MYSQL\_USER: Su

            #   MYSQL\_PASSWORD: rootroot

  volumeClaimTemplates:

    - metadata:

        name: htmlhome

      spec:

        accessModes:

          - ReadWriteOnce

        resources:

          requests:

            storage: 1Gi

        storageClassName: managed

---

apiVersion: v1

kind: Service

metadata:

  name: mysqlsvc

  labels:

    env: uat

spec:

  clusterIP: None

  type: ClusterIP

  ports:

    - name: mysqlsvc

      port: 8081

      protocol: TCP

      targetPort: 3306

  selector:

    env: uat

**2.Create a nopCommerce deployment with 1 replica**

apiVersion: apps/v1

kind: Deployment

metadata:

  name: newjenkinsdeploy

  labels:

    env: testing

spec:

  minReadySeconds: 2

  replicas: 2

  selector:

    matchLabels:

      env: testing

  strategy:

    rollingUpdate:

      maxSurge: 25%

      maxUnavailable: 25%

    type: RollingUpdate

  template:

    metadata:

      name: newjenkinsdeploy

      labels:

        env: testing

    spec:

      containers:

        - name: newjenkinsdeploy

          image: sridhar231/jenkins:1.0

          ports:

            - name: tcpport

              protocol: TCP

              containerPort: 8080

---

apiVersion: v1

kind: Service

metadata:

  name: jenkins-svc

  labels:

    env: testing

spec:

  selector:

    env: testing

  ports:

    - name: jenkins-svc

      port: 8081

      protocol: TCP

      targetPort: 8080

  type: LoadBalancer

**3.Create a Headless Service to interact with nopCommerce with MySQL**

---

apiVersion: v1

kind: Service

metadata:

  name: nopCommerce-svc

spec:

  selector:

    app: nopCommerce

  type: ClusterIP

  clusterIP: None

  ports:

    - name: nopCommerce-svc

      port: 5000

      targetPort: 5000

      protocol: TCP

 ---

apiVersion: v1

kind: Pod

metadata:

  name: mysql

spec:

  containers:

        - name: mysql

          image: mysql:8.0

          ports:

            - name: mysql-st

              containerPort: 3306

              protocol: TCP

          env:

            - name: MYSQL\_ROOT\_PASSWORD

              value: rootroot

            - name: MYSQL\_USER

              value: Su

            - name: MYSQL\_PASSWORD

              value: rootroot

            - name: MYSQL\_DATABASE

              value: Persons

1. **Create a Load Balancer to expose the nopCommerce to External World**

---

apiVersion: v1

kind: Service

metadata:

  name: nop-svc

  labels:

    env: testing

spec:

  selector:

    env: testing

  ports:

    - name: nop-svc

      port: 5000

      targetPort: 5000

  type: LoadBalancer

---

apiVersion: v1

kind: Pod

metadata:

  name: nopcommerce

spec:

  containers:

    - name: nopcommerce

      image: nopcommerce:latest

      ports:

        containerPort: 5000