**Assignment**

**Deploy a Sample Web Application**

**Objective**

Create a deployment for a sample NGINX-based web application that:

1. Reads configuration values from a ConfigMap.
2. Reads environment credentials from a Secret.
3. Uses a temporary shared volume with emptyDir.
4. Has defined CPU/Memory requests and limits.
5. Is governed by a ResourceQuota in its namespace.

**Assignment Structure**

**Directory Structure**

k8s-assignment/

├── 1-namespace.yaml

├── 2-quota.yaml

├── 3-configmap.yaml

├── 4-secret.yaml

├── 5-deployment.yaml

├── 6-service.yaml

├── README.md

**Step-by-Step Instructions & Solutions**

**🔹 Step 1: Create a Namespace**

**1-namespace.yaml**

apiVersion: v1

kind: Namespace

metadata:

name: webapp-ns

**Apply:**

kubectl apply -f 1-namespace.yaml

**🔹 Step 2: Set Resource Quota**

**2-quota.yaml**

apiVersion: v1

kind: ResourceQuota

metadata:

name: webapp-quota

namespace: webapp-ns

spec:

hard:

requests.cpu: "500m"

requests.memory: "512Mi"

limits.cpu: "1"

limits.memory: "1Gi"

pods: "2"

**Apply:**

kubectl apply -f 2-quota.yaml

**🔹 Step 3: Create ConfigMap for Web Configuration**

📄 **3-configmap.yaml**

apiVersion: v1

kind: ConfigMap

metadata:

name: webapp-config

namespace: webapp-ns

data:

APP\_THEME: "dark"

APP\_LANGUAGE: "en"

**Apply:**

kubectl apply -f 3-configmap.yaml

**🔹 Step 4: Create Secret for Credentials**

📄 **4-secret.yaml**

apiVersion: v1

kind: Secret

metadata:

name: webapp-secret

namespace: webapp-ns

type: Opaque

data:

DB\_USER: YWRtaW4= # admin

DB\_PASSWORD: c2VjcmV0MTIz # secret123

**Apply:**

kubectl apply -f 4-secret.yaml

You can encode your own values using:

echo -n "admin" | base64

echo -n "secret123" | base64

**🔹 Step 5: Create Deployment with:**

* Resource Requests & Limits
* ConfigMap & Secret envs
* emptyDir volume for shared temp storage

**5-deployment.yaml**

apiVersion: apps/v1

kind: Deployment

metadata:

name: webapp-deployment

namespace: webapp-ns

spec:

replicas: 1

selector:

matchLabels:

app: webapp

template:

metadata:

labels:

app: webapp

spec:

containers:

- name: nginx

image: nginx

resources:

requests:

memory: "128Mi"

cpu: "250m"

limits:

memory: "256Mi"

cpu: "500m"

envFrom:

- configMapRef:

name: webapp-config

- secretRef:

name: webapp-secret

volumeMounts:

- name: temp-storage

mountPath: /usr/share/nginx/html/temp

volumes:

- name: temp-storage

emptyDir: {}

🔧 **Apply:**

kubectl apply -f 5-deployment.yaml

**🔹 Step 6: Expose the Deployment with a Service**

**6-service.yaml**

apiVersion: v1

kind: Service

metadata:

name: webapp-service

namespace: webapp-ns

spec:

type: LoadBalancer

selector:

app: webapp

ports:

- port: 80

targetPort: 80

protocol: TCP

🔧 **Apply:**

kubectl apply -f 6-service.yaml

**🔍 Step 7: Validation Steps**

Run these commands to validate your setup:

# Check namespace and quota

kubectl get ns

kubectl describe quota webapp-quota -n webapp-ns

# Check configmap and secret

kubectl get configmap -n webapp-ns

kubectl get secret -n webapp-ns

# Check deployment and pod logs

kubectl get pods -n webapp-ns

kubectl describe pod <pod-name> -n webapp-ns

kubectl exec -it <pod-name> -n webapp-ns -- env | grep APP\_

kubectl exec -it <pod-name> -n webapp-ns -- env | grep DB\_