Part 1 - Kubernetes Workloads Challenge

1. Backend Deployment (Immutable)

```
apiVersion: apps/v1
kind: Deployment
metadata:
  namespace: project-plato
  name: backend
  labels:
    app: backend
spec:
  replicas: 1
  selector:
    matchLabels:
      app: backend
  template:
    metadata:
      labels:
        app: backend
    spec:
      containers:
      - name: busybox
        image: busybox:1.32.0
        command: ['sh', '-c', 'tail -f /dev/null']
        imagePullPolicy: IfNotPresent
        securityContext:
          readOnlyRootFilesystem: true
          runAsNonRoot: true
          allowPrivilegeEscalation: false
        volumeMounts:
        - name: tmp-volume
          mountPath: /tmp
      restartPolicy: Always
      volumes:
      - name: tmp-volume
        emptyDir: {}
```

2. DB1 and DB2 Deployments

```
apiVersion: apps/v1
kind: Deployment
metadata:
  namespace: project-plato
  name: db1
  labels:
    app: db1
spec:
  replicas: 1
  selector:
    matchLabels:
```

```
      app: db1
  template:
    metadata:
      labels:
        app: db1
    spec:
      containers:
      - name: nginx
        image: nginx:1.16.1-alpine
        ports:
        - containerPort: 6379
---
apiVersion: apps/v1
kind: Deployment
metadata:
  namespace: project-plato
  name: db2
  labels:
    app: db2
spec:
  replicas: 1
  selector:
    matchLabels:
      app: db2
  template:
    metadata:
      labels:
        app: db2
    spec:
      containers:
      - name: nginx
        image: nginx:1.16.1-alpine
        ports:
        - containerPort: 5432
```

3. Services for DB1 and DB2

```
apiVersion: v1
kind: Service
metadata:
  namespace: project-plato
  name: db1
spec:
  selector:
    app: db1
  ports:
  - port: 6379
    targetPort: 6379
---
apiVersion: v1
kind: Service
metadata:
  namespace: project-plato
  name: db2
```

```
spec:
  selector:
    app: db2
  ports:
  - port: 5432
    targetPort: 5432
```

5. Liveness and Readiness Probes for Backend

```
livenessProbe:
  exec:
    command: ["true"]
  initialDelaySeconds: 3
  periodSeconds: 10

readinessProbe:
  tcpSocket:
    port: 6379
  initialDelaySeconds: 5
  periodSeconds: 10
```

6. NetworkPolicy for Backend

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  namespace: project-plato
  name: np-backend
spec:
  podSelector:
    matchLabels:
      app: backend
  policyTypes:
  - Ingress
  - Egress
  egress:
  - to:
    - podSelector:
        matchLabels:
          app: db1
    ports:
    - protocol: TCP
      port: 6379
  - to:
    - podSelector:
        matchLabels:
          app: db2
    ports:
    - protocol: TCP
      port: 5432
```

7. Secret for DB2

```
apiVersion: v1
```

```
kind: Secret
metadata:
  name: db2-secret
  namespace: project-plato
type: Opaque
data:
  username: palash
  password: 123456@Rooya
```

Bonus Tasks:

PostgreSQL and Prometheus Helm Deployments

```
#!/bin/bash


# Add Helm repositories

helm repo add bitnami https://charts.bitnami.com/bitnami

helm repo add prometheus-community https://prometheus-community.github.io/helm-charts


# Update Helm repositories

helm repo update


# Deploy PostgreSQL with a password

helm install postgres bitnami/postgresql -- set postgresqlPassword=mysecretpassword


# Deploy Prometheus Stack

helm install kube-prometheus-stack prometheus-community/kube-prometheus-stack


echo "PostgreSQL and Prometheus deployments are complete!"
```