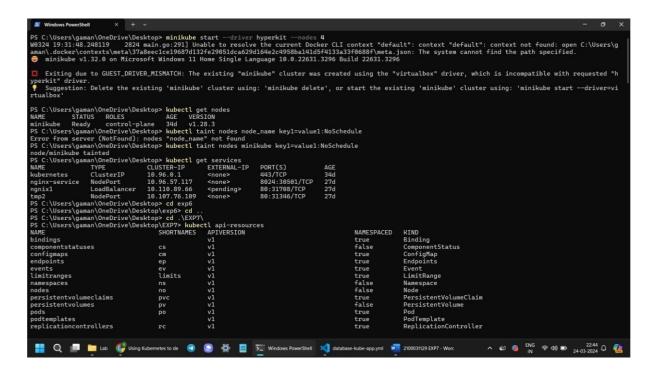
# minikube start kubectl get nodes kubectl taint nodes minikube key1=value1:NoSchedule kubectl get services kubectl api-resources



## frontend-kube-app:

```
apiVersion:
apps/v1
                kind:
Deployment metadata:
 name: web-httpd-deployment
spec:
  selector:
   matchLabels:
     app: web-httpd
 replicas: 10
 template:
    metadata:
     labels:
       app: web-httpd
    spec:
      containers:
         - name: web-httpd
            image:
            httpd:2.4.55
            ports:
         - containerPort: 80
```

## backend-kube-app.yml:

```
apiVersion:
             kind:
apps/v1
Deployment
metadata:
  name: nodejs-app-deployment
spec:
  selector:
   matchLabels:
     app: nodejs-app
 replicas: 4
 template:
  metadata:
     labels:
       app: nodejs-app
    spec:
      containers:
        - name: nodejs-app image:
         node:19-alpine3.16 command:
         ["sleep", "100000"]
```

#### database-kube-app.yml:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: postgres-db-deployment # Sets Deployment name
spec:
  replicas: 1
  selector:
    matchLabels:
      app: postgres-db
  template:
    metadata:
      labels:
        app: postgres-db
    spec:
      containers:
        - name: postgres-db
          image: postgres:10.1 # Sets Image
          imagePullPolicy: "IfNotPresent"
          ports:
            - containerPort: 5432 # Exposes container port
          envFrom:
            - configMapRef: # Maps env variable from ConfigMap
                name: postgres-config
          volumeMounts:
            - mountPath: /var/lib/postgresql/data
              name: postgres-volume
      volumes:
        - name: postgres-volume
          persistentVolumeClaim:
            claimName: postgres-pv-claim # Maps claim from
PersistantVolumeClaim
```

#### postgres-config.yml:

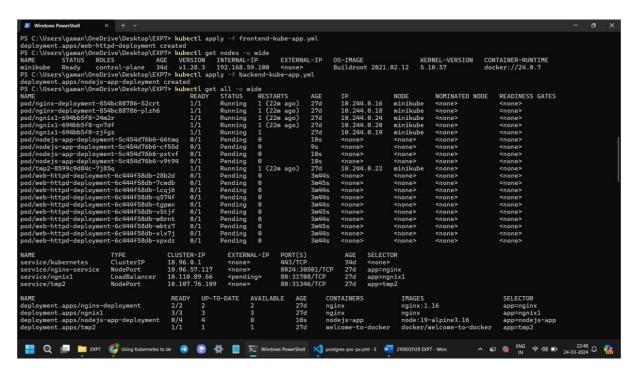
```
kind: ConfigMap
metadata: name:
postgres-config
labels:
    app: postgres-db
data:
    POSTGRES_DB: postgresdb
    POSTGRES_USER: admin
    POSTGRES_PASSWORD: mypass
```

#### postgres-pvc-pv.yml:

```
kind:
      PersistentVolume apiVersion:
                                       v1 metadata:
                                                       name:
postgres-pv-volume # Sets PV's name labels:
   type: local # Sets PV's type to local app: postgres-db
spec:
  storageClassName: manual capacity:
    storage: 5Gi # Sets PV Volume
  accessModes:
    - ReadWriteMany hostPath:
   path: "/mnt/data"
--- kind: PersistentVolumeClaim apiVersion: v1 metadata: name:
postgres-pv-claim # Sets name of PVC labels:
   app: postgres-db
spec:
  storageClassName: manual
```

```
accessModes:
   - ReadWriteMany # Sets read and write access resources:
   requests:
    storage: 5Gi # Sets volume size
```

kubectl apply -f frontend-kubeapp.yml kubectl get nodes -o wide kubectl apply -f backendkube-app.yml kubectl get all -o wide



kubectl apply -f postgresconfig.yml kubectl apply -f postgres-pvc-pv.yml kubectl

## apply -f postgres-pvc-pv.yml

# kubectl get all -o wide

