root@ss: ~/k8s/ps

```
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# vi postgres-configmap.yaml
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# kubectl apply -f postgres-configmap.yaml
configmap/postgres-secret created
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# kubectl get configmap
NAME                DATA   AGE
kube-root-ca.crt    1      4d18h
postgres-secret     3      29s
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# vi postgres-configmap.yaml
root@ss:~/k8s/ps# kubectl apply -f postgres-configmap.yaml
configmap/postgres-secret configured
root@ss:~/k8s/ps# kubectl get configmap
NAME                DATA   AGE
kube-root-ca.crt    1      4d18h
postgres-secret     4      4m3s
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# cat postgres-configmap.yaml
# Create ConfigMap postgres-secret for the postgres app
# Define default database name, user, and password
apiVersion: v1
kind: ConfigMap
metadata:
  name: postgres-secret
  labels:
    app: postgres
data:
  POSTGRES_DB: testdatabase
  POSTGRES_USER: testuser
  POSTGRES_PASSWORD: testpassword
```

```
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# vi postgres-configmap.yaml
root@ss:~/k8s/ps# kubectl apply -f postgres-configmap.yaml
configmap/postgres-secret configured
root@ss:~/k8s/ps# kubectl get configmap
NAME                 DATA   AGE
kube-root-ca.crt     1      4d18h
postgres-secret      4      4m3s
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# cat postgres-configmap.yaml
# Create ConfigMap postgres-secret for the postgres app
# Define default database name, user, and password
apiVersion: v1
kind: ConfigMap
metadata:
  name: postgres-secret
  labels:
    app: postgres
data:
  POSTGRES_DB: testdatabase
  POSTGRES_USER: testuser
  POSTGRES_PASSWORD: testpassword
  PGDATA: /var/lib/postgresql/data/pgdata

root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# vi postgres-volume.yaml
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# cat postgres-volume.yaml
apiVersion: v1
kind: PersistentVolume # Create PV
metadata:
  name: postgres-volume # Sets PV name
  labels:
```

```
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# vi postgres-volume.yaml
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# cat postgres-volume.yaml
apiVersion: v1
kind: PersistentVolume # Create PV
metadata:
  name: postgres-volume # Sets PV name
  labels:
    type: local # Sets PV's type
    app: postgres
spec:
  storageClassName: manual
  capacity:
    storage: 10Gi # Sets PV's size
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: "/var/lib/postgresql/data" # Sets PV's host path

root@ss:~/k8s/ps# cat postgres-volume.yaml
apiVersion: v1
kind: PersistentVolume # Create PV
metadata:
  name: postgres-volume # Sets PV name
  labels:
    type: local # Sets PV's type
    app: postgres
spec:
  storageClassName: manual
  capacity:
    storage: 10Gi # Sets PV's size
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: "/var/lib/postgresql/data" # Sets PV's host path

root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# kubectl apply -f postgres-volume.yaml -n ps
persistentvolume/postgres-volume created
```

```
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# kubectl apply -f postgres-volume.yaml -n ps
persistentvolume/postgres-volume created
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# kubectl apply -f postgres-configmap.yaml -n ps
configmap/postgres-secret created
root@ss:~/k8s/ps# kubectl get configmap -n ps
NAME               DATA   AGE
kube-root-ca.crt   1      3d2h
postgres-secret    4      37s
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# vi postgres-pvc.yaml
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# cat postgres-pvc.yaml
apiVersion: v1
kind: PersistentVolumeClaim # Create PVC
metadata:
  name: postgres-volume-claim # Sets PVC's name
  labels:
    app: postgres # Defines app to create PVC for
spec:
  storageClassName: manual
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi # Sets PVC's size

root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# kubectl apply -f postgres-pvc.yaml -n ps
persistentvolumeclaim/postgres-volume-claim created
```

```
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# vi postgres-deployment.yaml
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# cat postgres-deployment.yaml
apiVersion: apps/v1
kind: Deployment # Create a deployment
metadata:
  name: postgres # Set the name of the deployment
spec:
  replicas: 1 # Set 1 deployment replicas
  strategy:
    type: Recreate
  selector:
    matchLabels:
      app: postgres
  template:
    metadata:
      labels:
        app: postgres
    spec:
      containers:
        - name: postgres
          image: postgres:10.13 # Docker image
          imagePullPolicy: "IfNotPresent"
          ports:
            - containerPort: 5432 # Exposing the container port 5432 for Postgre
SQL client connections.
          envFrom:
            - configMapRef:
                name: postgres-secret # Using the ConfigMap postgres-secret
          volumeMounts:
            - mountPath: /var/lib/postgresql/data
              name: postgresdata
      volumes:
        - name: postgresdata
          persistentVolumeClaim:
            claimName: postgres-volume-claim

root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
```

```
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# kubectl taint nodes minikube dedicated-
node/minikube untainted
root@ss:~/k8s/ps# kubectl get pods -n ps -w
NAME                        READY   STATUS    RESTARTS   AGE
postgres-5b9685c75c-ctx89   0/1     Pending   0          9m45s
^Croot@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# kubectl taint nodes minikube demo-taint-
node/minikube untainted
root@ss:~/k8s/ps# kubectl get pods -n ps -w
NAME                        READY   STATUS    RESTARTS   AGE
postgres-5b9685c75c-ctx89   1/1     Running   0          10m
^Croot@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# kubectl get deployments -n ps
NAME       READY   UP-TO-DATE   AVAILABLE   AGE
postgres   1/1     1            1           11m
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# vi postgres-service.yaml
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# cat postgres-service.yaml
apiVersion: v1
kind: Service # Create service
metadata:
  name: postgres # Sets the service name
  labels:
    app: postgres # Defines app to create service for
spec:
  type: NodePort # Sets the service type
```

```
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# vi postgres-service.yaml
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# cat postgres-service.yaml
apiVersion: v1
kind: Service # Create service
metadata:
  name: postgres # Sets the service name
  labels:
    app: postgres # Defines app to create service for
spec:
  type: NodePort # Sets the service type
  ports:
    - port: 5432 # Sets the port to run the postgres application
  selector:
    app: postgres

root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# kubectl apply -f postgres-service.yaml -n ps
service/postgres created
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# kubectl get svc
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   4d19h
root@ss:~/k8s/ps# kubectl get svc -n ps
NAME       TYPE       CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
postgres   NodePort   10.105.215.33   <none>        5432:31897/TCP   119s
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# kubectl exec -it  -- psql -h localhost -U appuser --password -
```

```
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# kubectl get pod -o wide
No resources found in default namespace.
root@ss:~/k8s/ps# kubectl get pod -o wide -n ps
NAME                        READY   STATUS    RESTARTS   AGE   IP           NODE
       NOMINATED NODE   READINESS GATES
postgres-5b9685c75c-ctx89   1/1     Running   0          34m   172.17.0.4   mini
kube   <none>           <none>
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# kubectl run -t -i --rm --image postgres:10.13 test bash
If you don't see a command prompt, try pressing enter.
root@test:/#
root@test:/#
root@test:/#
root@test:/#
root@test:/#
root@test:/# psql -h 172.17.0.4 -U testuser testdatabase
Password for user testuser:
psql (10.13 (Debian 10.13-1.pgdg90+1))
Type "help" for help.

testdatabase=#
testdatabase=#
testdatabase=#
testdatabase=#
testdatabase=#
testdatabase=# CREATE TABLE testtable (testcolumn VARCHAR (50) );
CREATE TABLE
testdatabase=#
testdatabase=#
testdatabase=#
testdatabase=# \dt
          List of relations
 Schema |   Name    | Type  |  Owner
--------+-----------+-------+----------
 public | testtable | table | testuser
(1 row)

testdatabase=#
```

```
testdatabase=#
testdatabase=#
testdatabase=#
testdatabase=#
testdatabase=# exit
testdatabase-# q
testdatabase-# \q
root@test:/# \q
bash: q: command not found
root@test:/# exit
exit
Session ended, resume using 'kubectl attach test -c test -i -t' command when the
 pod is running
pod "test" deleted
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# kubectl delete po postgres-5b9685c75c-ctx89 -n ps
pod "postgres-5b9685c75c-ctx89" deleted
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# kubectl get po -o wide -n ps
NAME                        READY   STATUS    RESTARTS   AGE   IP           NODE
       NOMINATED NODE   READINESS GATES
postgres-5b9685c75c-h9spq   1/1     Running   0          56s   172.17.0.5   mini
kube   <none>           <none>
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# kubectl run -t -i --rm --image postgres:10.13 test bash
If you don't see a command prompt, try pressing enter.
root@test:/#
root@test:/#
root@test:/#
root@test:/#
root@test:/# psql -h 172.17.0.5 -U testuser testdatabase
Password for user testuser:
psql (10.13 (Debian 10.13-1.pgdg90+1))
Type "help" for help.

testdatabase=#
```

```
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# kubectl delete po postgres-5b9685c75c-ctx89 -n ps
pod "postgres-5b9685c75c-ctx89" deleted
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# kubectl get po -o wide -n ps
NAME                        READY   STATUS    RESTARTS   AGE   IP           NODE
       NOMINATED NODE   READINESS GATES
postgres-5b9685c75c-h9spq   1/1     Running   0          56s   172.17.0.5   mini
kube   <none>           <none>
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps#
root@ss:~/k8s/ps# kubectl run -t -i --rm --image postgres:10.13 test bash
If you don't see a command prompt, try pressing enter.
root@test:/#
root@test:/#
root@test:/#
root@test:/#
root@test:/# psql -h 172.17.0.5 -U testuser testdatabase
Password for user testuser:
psql (10.13 (Debian 10.13-1.pgdg90+1))
Type "help" for help.

testdatabase=#
testdatabase=#
testdatabase=#
testdatabase=#
testdatabase=# \dt
          List of relations
 Schema |   Name    | Type  |  Owner
--------+-----------+-------+----------
 public | testtable | table | testuser
(1 row)

testdatabase=#
testdatabase=#
testdatabase=#
testdatabase=#
testdatabase=#
testdatabase=#
```