Deploy Postgres Client in cluster (psql)

We will first create a new namespace as **client-postgress**

$ kubectl create namespace client-postgress

```
lostinserver@lostinserver:~/Documents/K8s/postgres$ kubectl create namespace cli
ent-postgress
namespace/client-postgress created
lostinserver@lostinserver:~/Documents/K8s/postgres$
```

Deploy using the new namespace as,

$ kubectl apply -f postgres-config.yaml -n client-postgress

$ kubectl apply -f postgres-deployment.yaml -n client-postgress

$ kubectl apply -f postgres-volume.yaml -n client-postgress

```
lostinserver@lostinserver:~/Documents/K8s/postgres$ kubectl apply -f postgres-co
nfig.yaml -n client-postgress
configmap/postgres-config unchanged
lostinserver@lostinserver:~/Documents/K8s/postgres$ kubectl apply -f postgres-de
ployment.yaml -n client-postgress
deployment.apps/postgres created
lostinserver@lostinserver:~/Documents/K8s/postgres$ kubectl apply -f postgres-vo
lume.yaml -n client-postgress
persistentvolume/postgrespv-volume unchanged
persistentvolumeclaim/postgres-volume-claim created
lostinserver@lostinserver:~/Documents/K8s/postgres$
```

Now to view the deployed clients,

$ kubectl get all -n client-postgres

```
lostinserver@lostinserver:~/Documents/K8s/postgres$ kubectl get all -n client-po
stgress
NAME                            READY   STATUS    RESTARTS   AGE
pod/postgres-5d9c946c6f-99rxn   0/1     Pending   0          2m32s

NAME                       READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/postgres   0/1     1            0           2m32s

NAME                                  DESIRED   CURRENT   READY   AGE
replicaset.apps/postgres-5d9c946c6f   1         1         0       2m32s
lostinserver@lostinserver:~/Documents/K8s/postgres$
```

Here, we can see the running pod of client.