

In this example show the practical concepts:

- Using a MySQL Client to connect to MySQL server
- See data persistence in action even when pods are deleted

command to deploy:

```
kubectl apply -f mysql-deployment.yaml
```

Once you deploy and checked your pvc you can see the storage being created on your GCP console.

Commands

```
kubectl describe deployment mysql
```

```
kubectl get pods
```

```
kubectl describe pvc mysql-pv-claim
```

Run my sql client to connect to the server

```
kubectl run -it --rm --image=mysql:5.6 --restart=Never mysql-client --  
mysql -h mysql -ppassword
```

This command creates a new Pod in the cluster running a MySQL client and connects it to the server through the Service. If it connects, you know your stateful MySQL database is up and running.

Before start looking around with mysql just open up another terminal and use

kubectl get pods and you can see the the mysql client is up and running.

If you delete the client it will not regenerate because it was not in the desired state document ie the yaml config file that we used to create the service and deployment.

Mysql commands

```
//create a new database
```

```
create database kube_db;
```

```
use kube_db;
```

```
//create table
```

```
create table names ( id int unsigned not null auto_increment, firstname varchar(20) not null,  
lastname varchar(20) not null, primary key(id) );
```

```
//insert values into the table
```

```
INSERT INTO names ( id, firstname, lastname )
```

```
VALUES ( null, 'Samuel', 'Lee' ),
```

```
(null, 'Daniel', 'Jackson'),
```

```
(null, 'Ann', 'Marie');
```

Doing some tests

1. Delete the mysql client pod and it should not regenerate. You can delete the pod by simply exiting the mysql terminal.
2. Delete the mysql server pod by using its label:

kubectl delete pod -l app=mysql

//check status, the name should change as well and you should see on terminating,

kubectl get pods

Connect back to mysql using the client and see if the database we created and the table created is still there, it should be still there, that is why its persisted volume and it is connected to this Pod and all containers within this pod.

Yep data is still there!

Now delete the deployment, service and pvc, delete by label and delete by name

kubectl delete deployment -l app=mysql

kubectl delete service mysql

kubectl delete pvc mysql-pv-claim

Or you could just delete the cluster to delete all resources. From GCP Console

//Check back with GCP console to make sure the PVC is deleted under storage.