

# Create PersistentVolumes & Run a Stateful Application

## SSH to your AWS Workstation

ssh devops@<public-ip-addr> of your Workstation

Password is : Dev0p\$!!/

Replace <your-name> with your name throughout the lab.

Run the below commands on your AWS-Workstation.

### 1. Create your PersistentVolumes and PersistentVolumeClaims

```
$ sudo su
# mkdir stateless-application
# cd stateless-application/
# curl -f https://pastebin.com/raw/rVB5i6K0 > mysql-volumeclaim-<your-name>.yaml
```

### 2. Replace <your-name> with your name in the volume claim yaml file.

```
# vim mysql-volumeclaim-<your-name>.yaml
```

Save and exit.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: mysql-volumeclaim-albert
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 5Gi
~
```

### 3. Create a volume Claim

```
# kubectl apply -f mysql-volumeclaim-<your-name>.yaml
```

### 4. Check the Persistent Volume Claim created

```
# kubectl get pvc mysql-volumeclaim-<your-name>
```

```
root@ip-172-31-40-214:/home/devops/application/stateless-application# kubectl get pvc mysql-volumeclaim-albert
NAME                                STATUS    VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
mysql-volumeclaim-albert           Bound     pvc-72f27898-60ee-11e9-ab0c-42010aa000a8   5Gi         RWO             standard       29s
root@ip-172-31-40-214:/home/devops/application/stateless-application#
```

#### 5. Set up MySQL - Create a Secret to store password

```
# kubectl create secret generic mysql-<your-name> --from-  
literal=password=password
```

#### 6. Download the mysql deployment yaml

```
# curl -f https://pastebin.com/raw/eBY9xmTc > mysql-<your-name>.yaml
```

#### 7. Edit and replace <your-name> with your name in the mysql-<your-name>.yaml

```
# vim mysql-<your-name>.yaml
```

Save and exit.

#### 8. Create the mysql deployment on the Kubernetes Cluster.

```
# kubectl create -f mysql-<your-name>.yaml
```

#### 9. Check the mysql POD status.

```
# kubectl get pod -l app=mysql-<your-name>
```

#### 10. Create MySQL service

```
# curl -f https://pastebin.com/raw/x8jz3kf0 > mysql-service-<your-name>.yaml
```

#### 11. Edit and replace <your-name> with your name in the mysql-service-<your-name>.yaml

```
# vim mysql-service-<your-name>.yaml
```

#### 12. Create the mysql service on the Kubernetes Cluster

```
# kubectl create -f mysql-service-<your-name>.yaml
```

#### 13. Inspect the PersistentVolumeClaim:

```
# kubectl describe pvc mysql-volumeclaim-<your-name>
```

```

root@ip-172-31-40-214 /home/devops/application/stateless-application# kubectl describe pvc mysql-volumeclaim-albert
Name:          mysql-volumeclaim-albert
Namespace:    default
StorageClass:  standard
Status:       Bound
Volume:        pvc-72f27898-60ee-11e9-ab0c-42618aa000a8
Labels:        {}
Annotations:   kubernetes.io/last-applied-configuration:
                {"apiVersion": "v1", "kind": "PersistentVolumeClaim", "metadata": {"name": "mysql-volumeclaim-albert", "namespace": "default"}, "spec": {
                "storageClassName": "standard",
                "resources": {
                  "requests": {
                    "storage": "1Gi"
                  }
                }
              }
Finalizers:    [kubernetes.io/pvc-protection]
Access Modes:  RWO
Events:
  Type      Reason              Age   From                  Message
  ----      ------              -
  Normal    ProvisioningSucceeded  16s   persistentvolume-controller  Successfully provisioned volume pvc-72f27898-60ee-11e9-ab0c-42618aa000a8 using kubernetes.io/gce-pd
Mounted By:   mysql-albert-5c99674859-phl2z
root@ip-172-31-40-214 /home/devops/application/stateless-application#

```

14. Run a MySQL client to connect to the server. Replace <your-name> with your name.

```
# kubectl run -it --rm --image=mysql:5.6 --restart=Never mysql-client-
<your-name> -- mysql -h mysql-<your-name> -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.

```

root@ip-172-21-26-76 /home/devops/stateless-application# kubectl run -it --rm --image=mysql:5.6 --restart=Never mysql-client -- mysql -h mysql-demo -p
password
If you don't see a command prompt, try pressing enter.
mysql>

```

15. Get the databases running on mysql.

Run the below command on once you connect to the mysql server.

```
mysql> show databases;
```

```

+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
+-----+
3 rows in set (0.00 sec)

mysql> CREATE DATABASE demo;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| demo |
| mysql |
| performance_schema |
+-----+
4 rows in set (0.00 sec)

```

16. Create a sample database

```
mysql> CREATE DATABASE <dbname>;
```

Disconnect from the mysql server by pressing **CRTL+C**

17. Delete the Deployment and Services.

```
# kubectl delete deployment,svc mysql-<your-name>  
# kubectl delete pvc mysql-volumeclaim-<your-name>
```