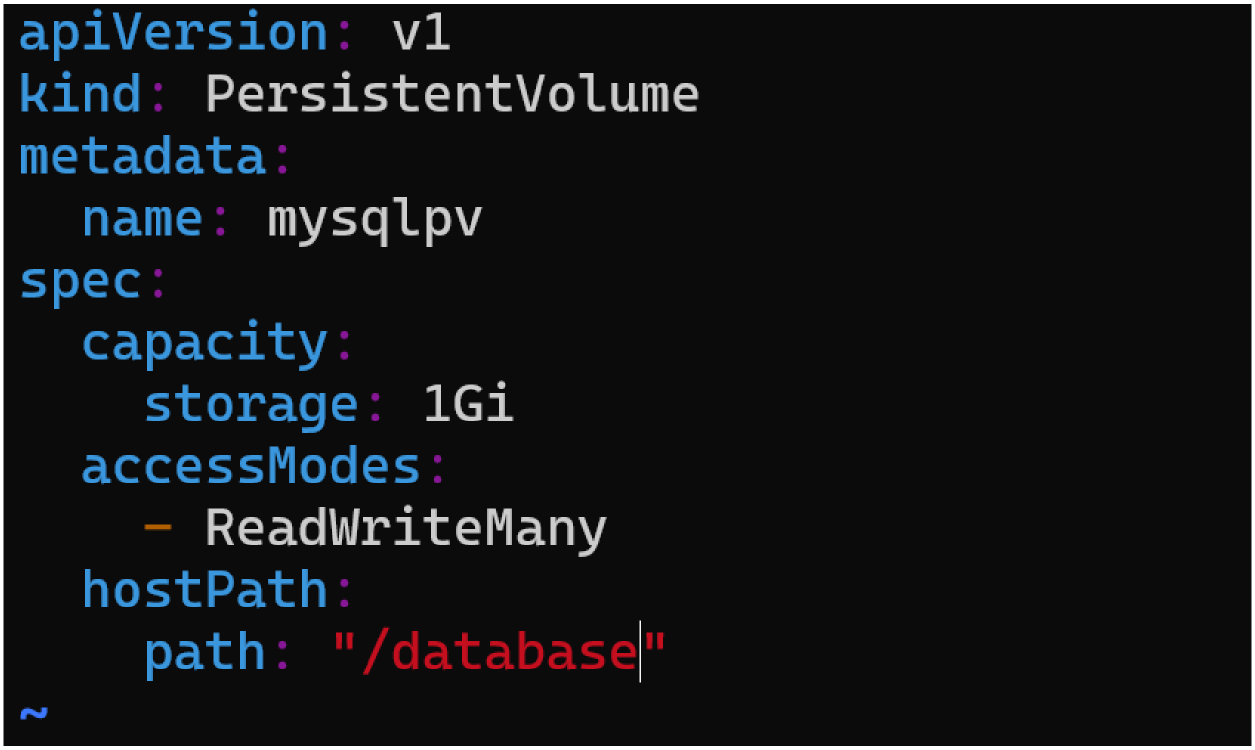
PV-PVC & Ingress

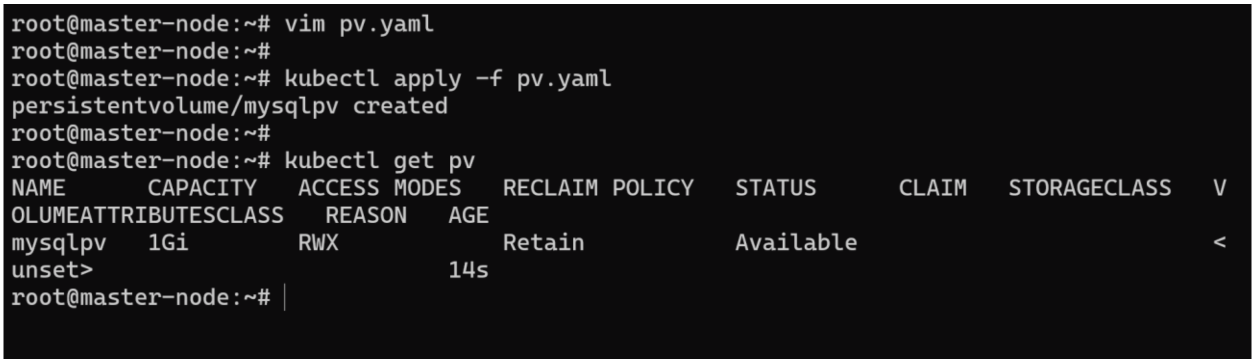
PV - PVC

Configure a Pod to Use a PersistentVolume for Storage

vim pv.yaml



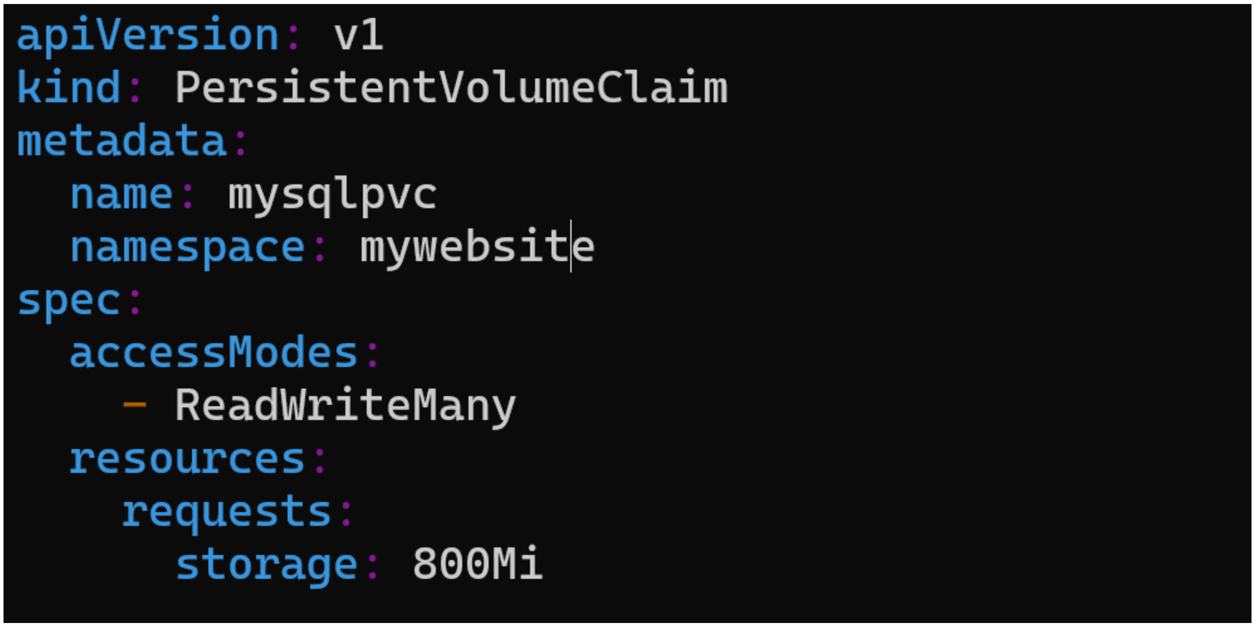
```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: mysqlpv
spec:
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteMany
  hostPath:
    path: "/database"
~
```



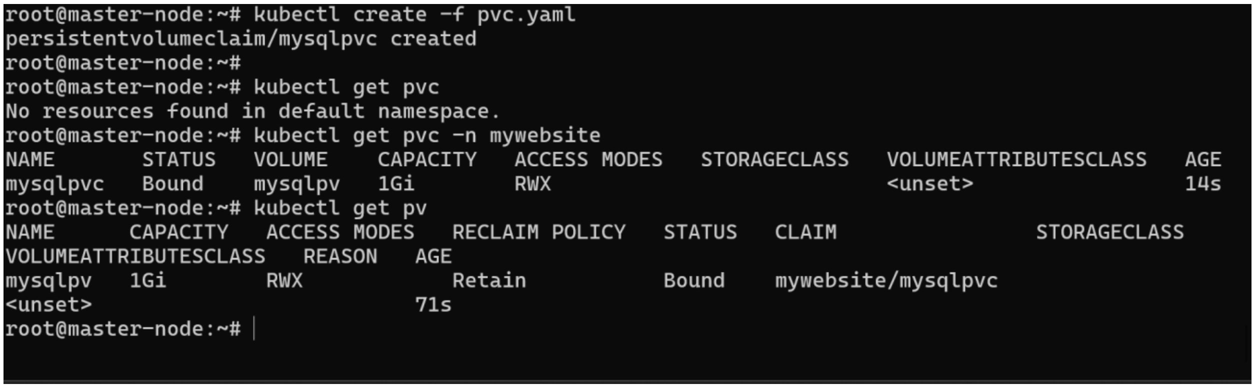
```
root@master-node:~# vim pv.yaml
root@master-node:~#
root@master-node:~# kubectl apply -f pv.yaml
persistentvolume/mysqlpv created
root@master-node:~#
root@master-node:~# kubectl get pv
NAME      CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS      CLAIM    STORAGECLASS   V
OLUMEATTRIBUTESCLASS   REASON   AGE
mysqlpv   1Gi        RWX            Retain           Available                           <
unset>                          14s
root@master-node:~#
```

Now let’s create a pvc

vim pvc.yaml



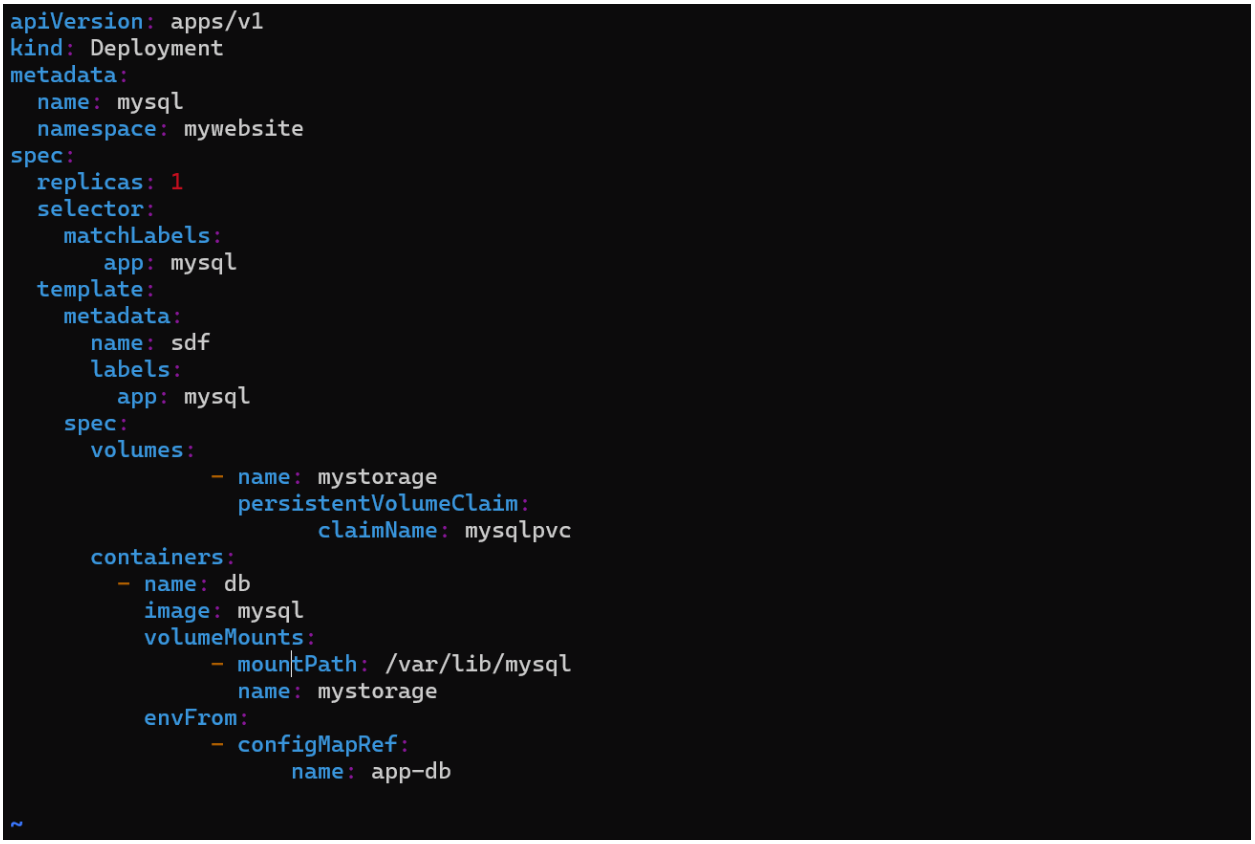
```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mysqlpvc
  namespace: mywebsite
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 800Mi
```



```
root@master-node:~# kubectl create -f pvc.yaml
persistentvolumeclaim/mysqlpvc created
root@master-node:~#
root@master-node:~# kubectl get pvc
No resources found in default namespace.
root@master-node:~# kubectl get pvc -n mywebsite
NAME       STATUS   VOLUME    CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
mysqlpvc   Bound    mysqlpv   1Gi        RWX                           <unset>                 14s
root@master-node:~# kubectl get pv
NAME      CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM                  STORAGECLASS
VOLUMEATTRIBUTESCLASS   REASON   AGE
mysqlpv   1Gi        RWX            Retain           Bound    mywebsite/mysqlpvc
<unset>                          71s
root@master-node:~#
```

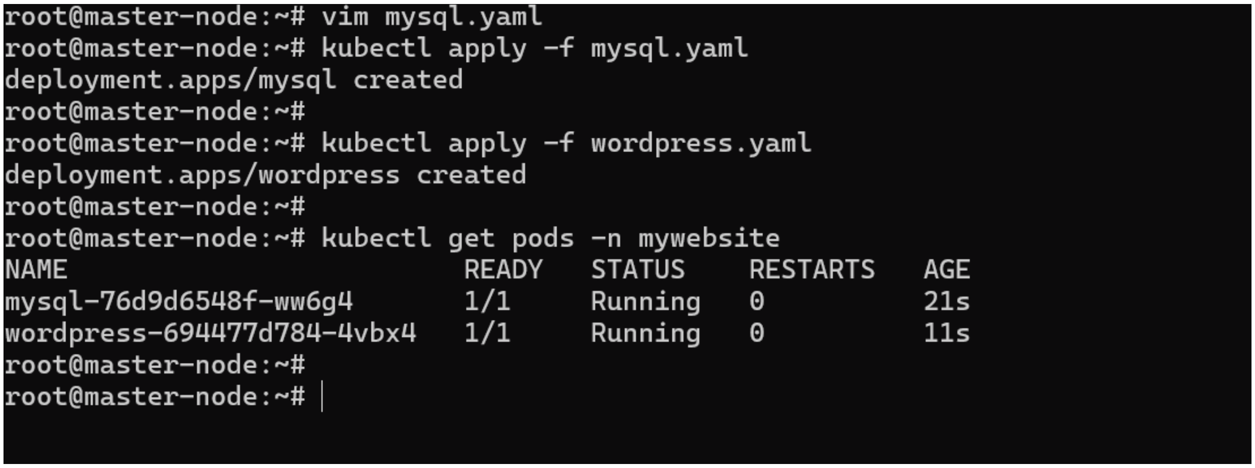
Now use this pvc for wordpress

vim mysql.yaml

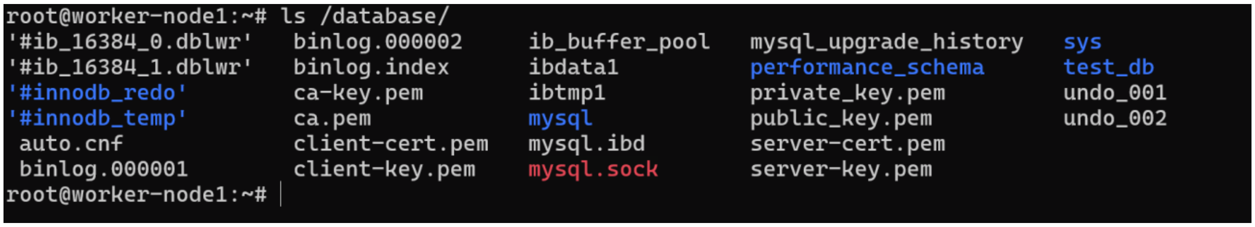


```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mysql
  namespace: mywebsite
spec:
  replicas: 1
  selector:
    matchLabels:
       app: mysql
  template:
    metadata:
      name: sdf
      labels:
        app: mysql
    spec:
      volumes:
               - name: mystorage
                 persistentVolumeClaim:
                       claimName: mysqlpvc
      containers:
        - name: db
          image: mysql
          volumeMounts:
               - mountPath: /var/lib/mysql
                 name: mystorage
          envFrom:
               - configMapRef:
                     name: app-db

~
```



```
root@master-node:~# vim mysql.yaml
root@master-node:~# kubectl apply -f mysql.yaml
deployment.apps/mysql created
root@master-node:~#
root@master-node:~# kubectl apply -f wordpress.yaml
deployment.apps/wordpress created
root@master-node:~#
root@master-node:~# kubectl get pods -n mywebsite
NAME                         READY   STATUS    RESTARTS   AGE
mysql-76d9d6548f-ww6g4       1/1     Running   0          21s
wordpress-694477d784-4vbx4   1/1     Running   0          11s
root@master-node:~#
root@master-node:~#
```



```
root@worker-node1:~# ls /database/
'#ib_16384_0.dblwr'   binlog.000002     ib_buffer_pool   mysql_upgrade_history   sys
'#ib_16384_1.dblwr'   binlog.index      ibdata1          performance_schema      test_db
'#innodb_redo'        ca-key.pem        ibtmp1           private_key.pem         undo_001
'#innodb_temp'        ca.pem            mysql            public_key.pem          undo_002
 auto.cnf             client-cert.pem   mysql.ibd        server-cert.pem
 binlog.000001        client-key.pem    mysql.sock       server-key.pem
root@worker-node1:~#
```

As we can see on worker node mysql data stored into /database directory.

Ingress Controllers