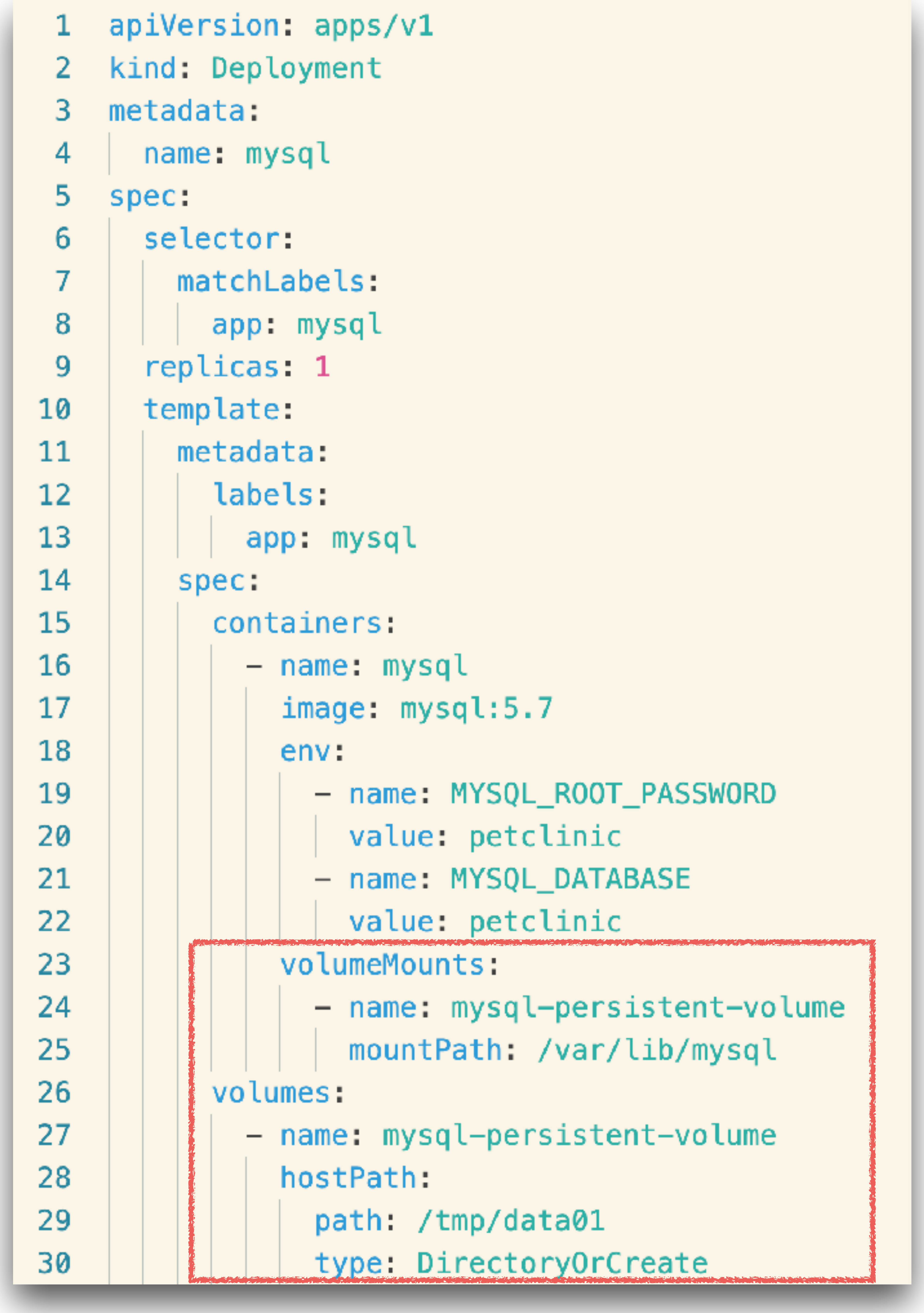
第 2 节

K8s存储解耦PVC和PV

本课内容

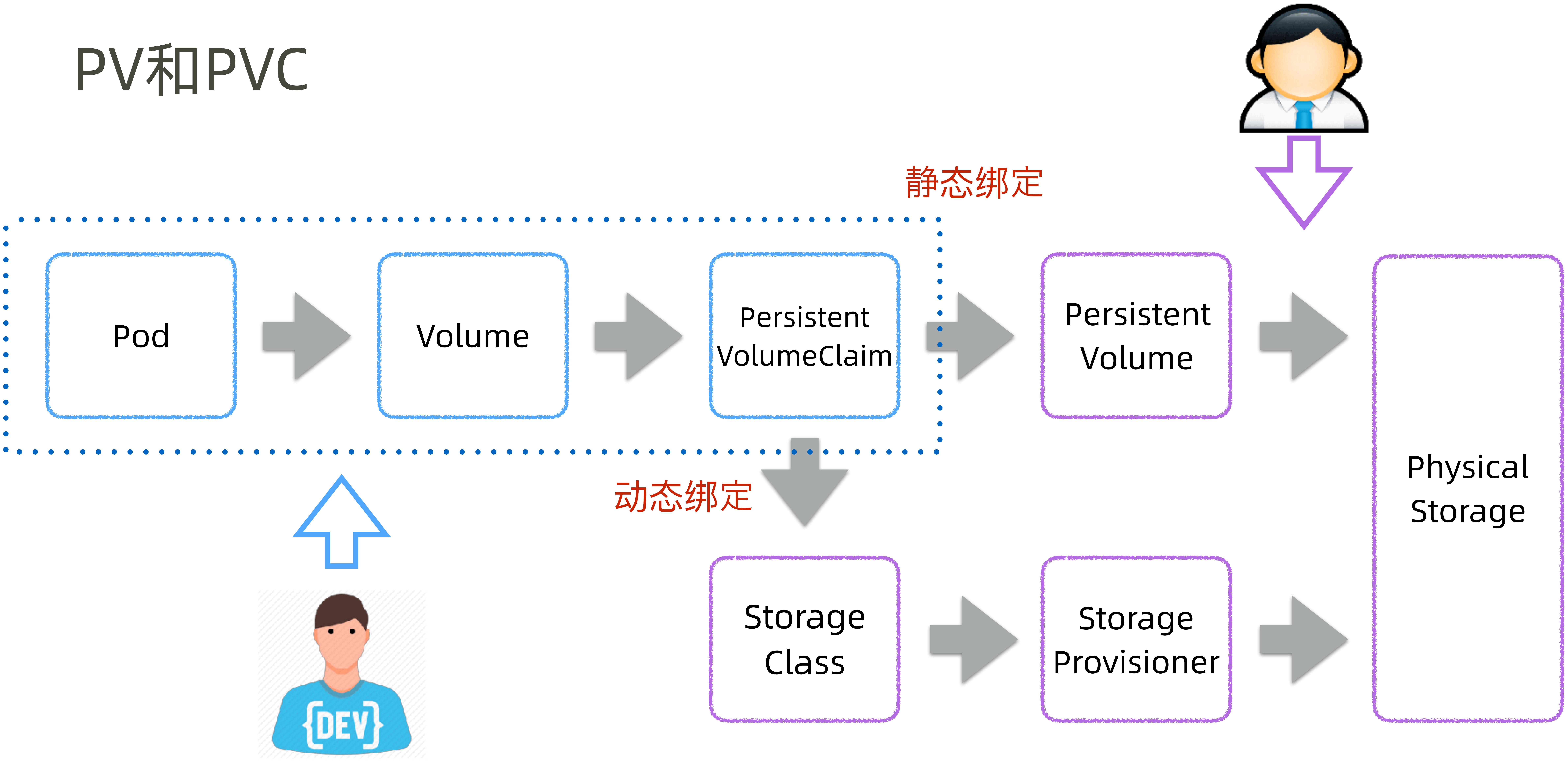
- K8s存储解耦机制
  - PersistentVolumeClaim(PVC)
  - PersistentVolume(PV)
- PVC和PV演示

存储卷Volume回顾



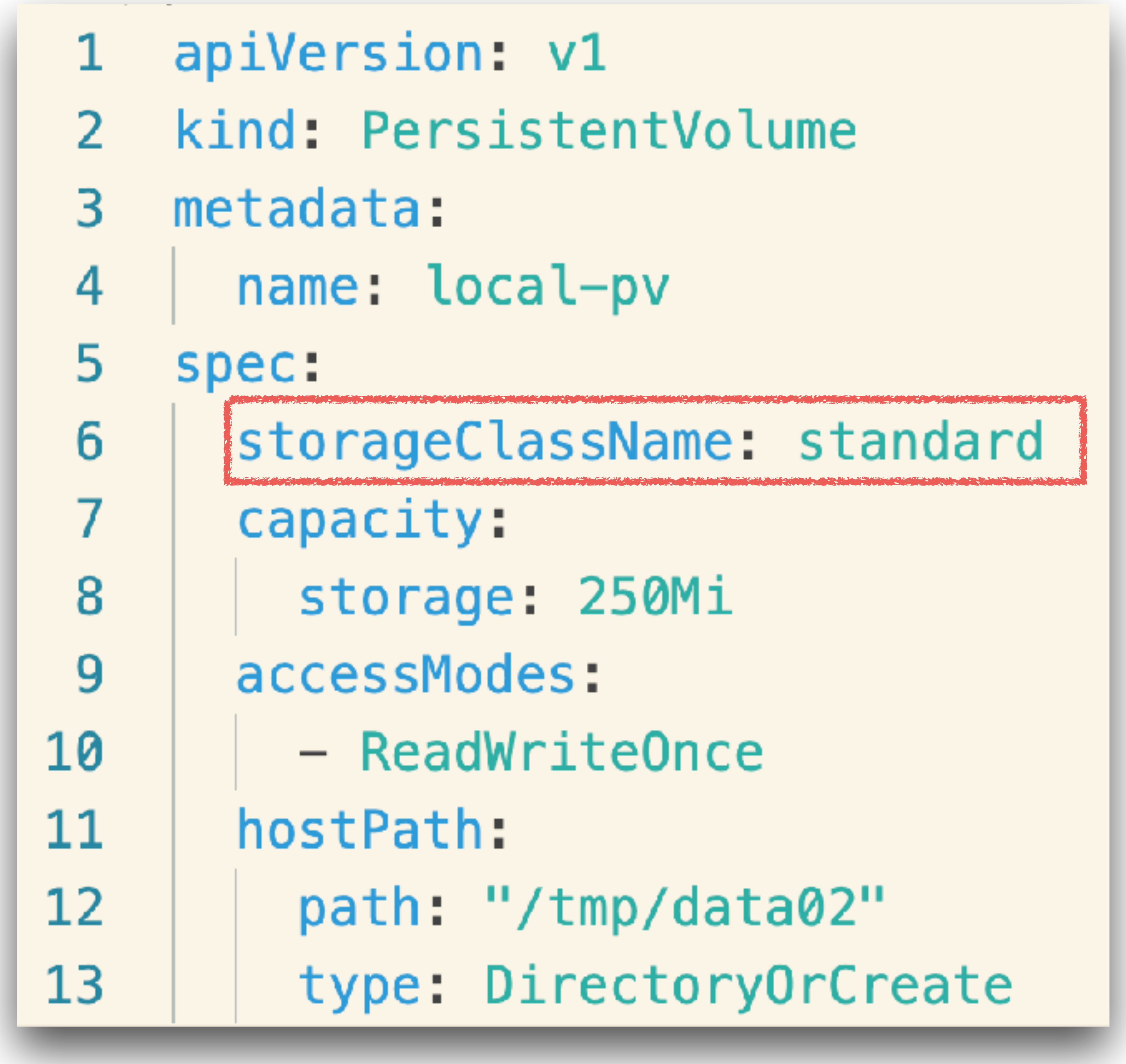
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mysql
spec:
  selector:
    matchLabels:
      app: mysql
  replicas: 1
  template:
    metadata:
      labels:
        app: mysql
    spec:
      containers:
      - name: mysql
        image: mysql:5.7
        env:
        - name: MYSQL_ROOT_PASSWORD
          value: petclinic
        - name: MYSQL_DATABASE
          value: petclinic
        volumeMounts:
        - name: mysql-persistent-volume
          mountPath: /var/lib/mysql
      volumes:
      - name: mysql-persistent-volume
        hostPath:
          path: /tmp/data01
          type: DirectoryOrCreate
```

PV和PVC

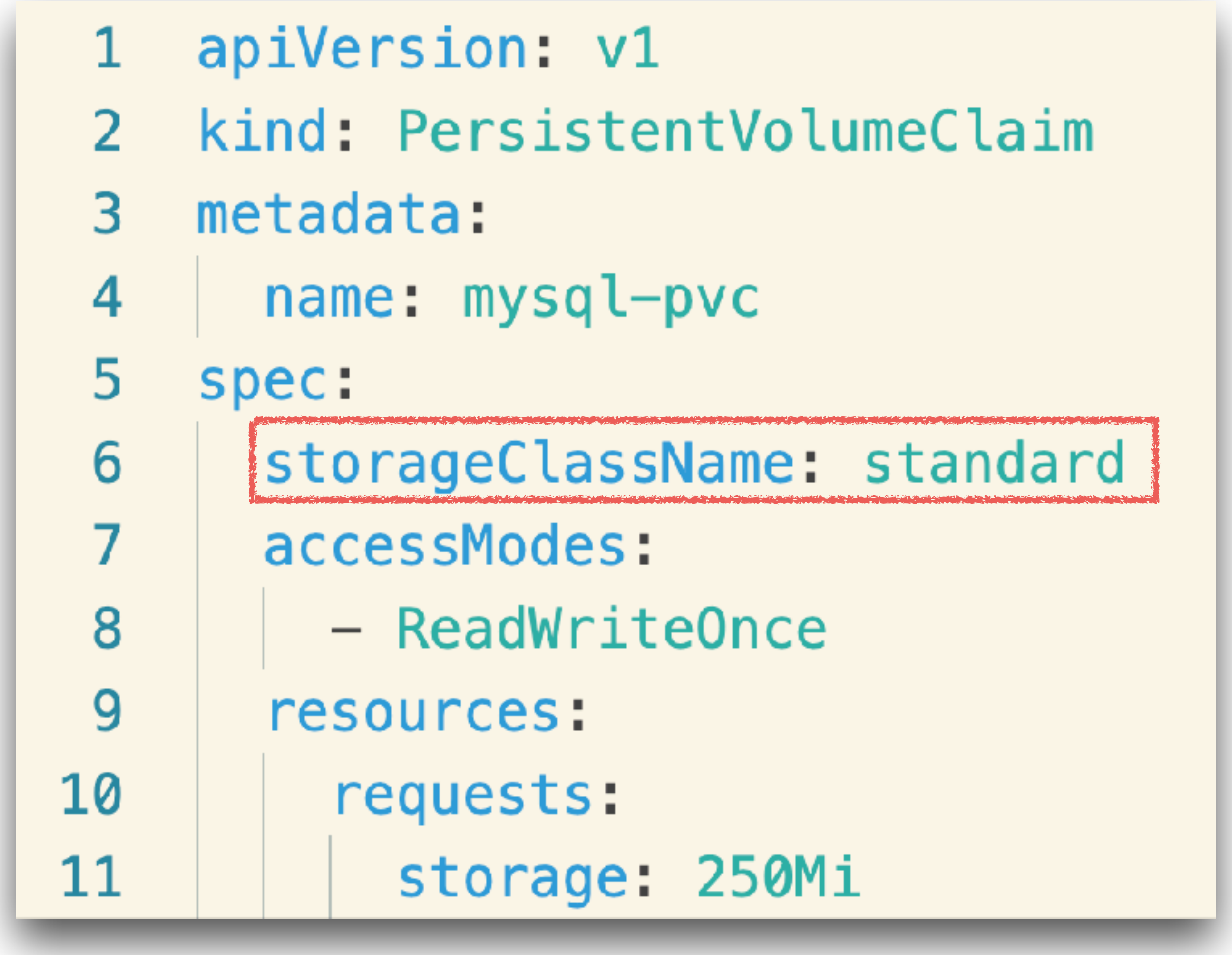


https://kubernetes.io/docs/concepts/storage/persistent-volumes/

local-pv.yml和mysql-pvc.yml

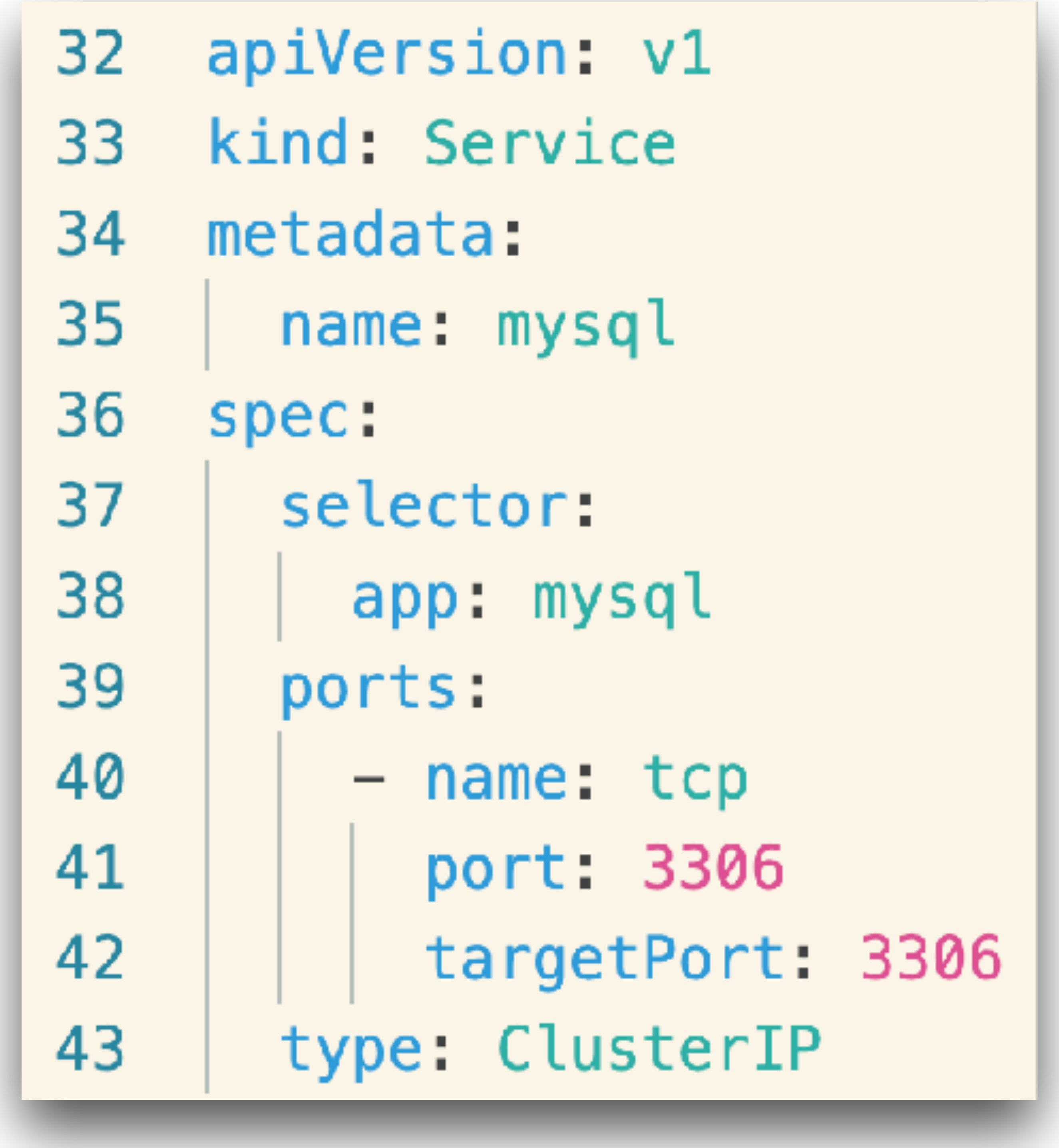


```yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: local-pv
spec:
  storageClassName: standard
  capacity:
    storage: 250Mi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: "/tmp/data02"
    type: DirectoryOrCreate
```

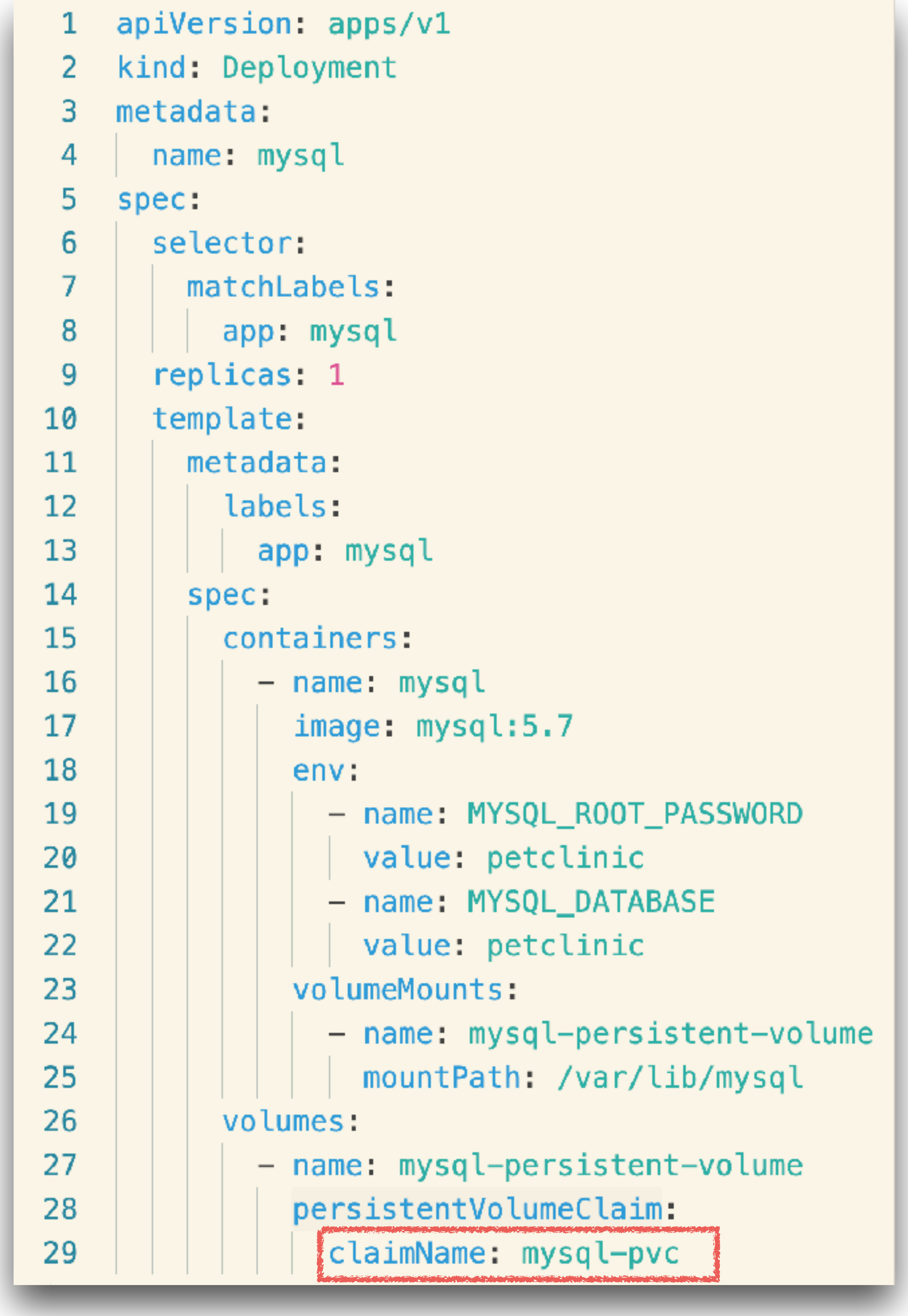


```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mysql-pvc
spec:
  storageClassName: standard
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 250Mi
```

mysql-svc.yml

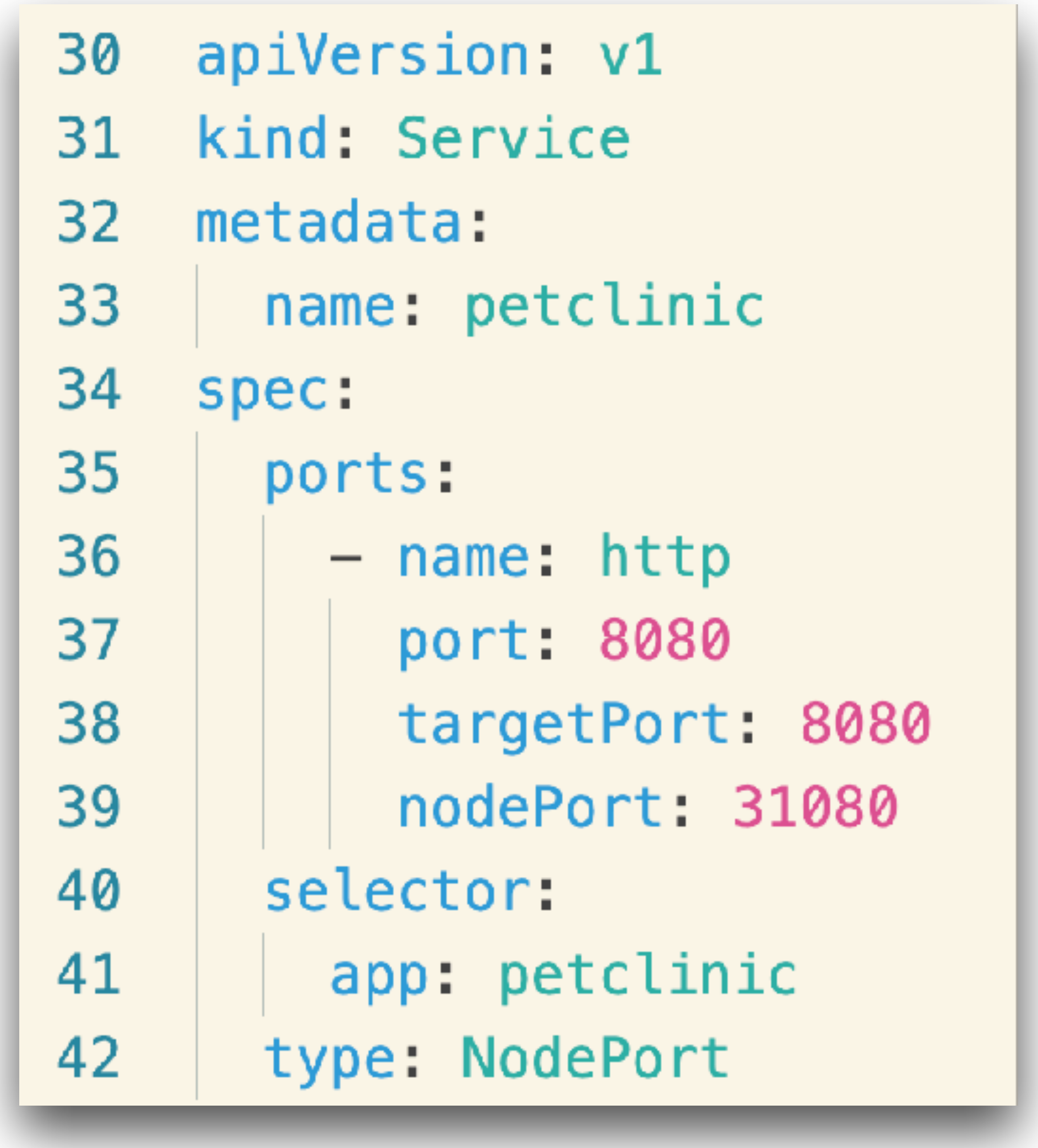


```yaml
apiVersion: v1
kind: Service
metadata:
  name: mysql
spec:
  selector:
    app: mysql
  ports:
    - name: tcp
      port: 3306
      targetPort: 3306
  type: ClusterIP
```

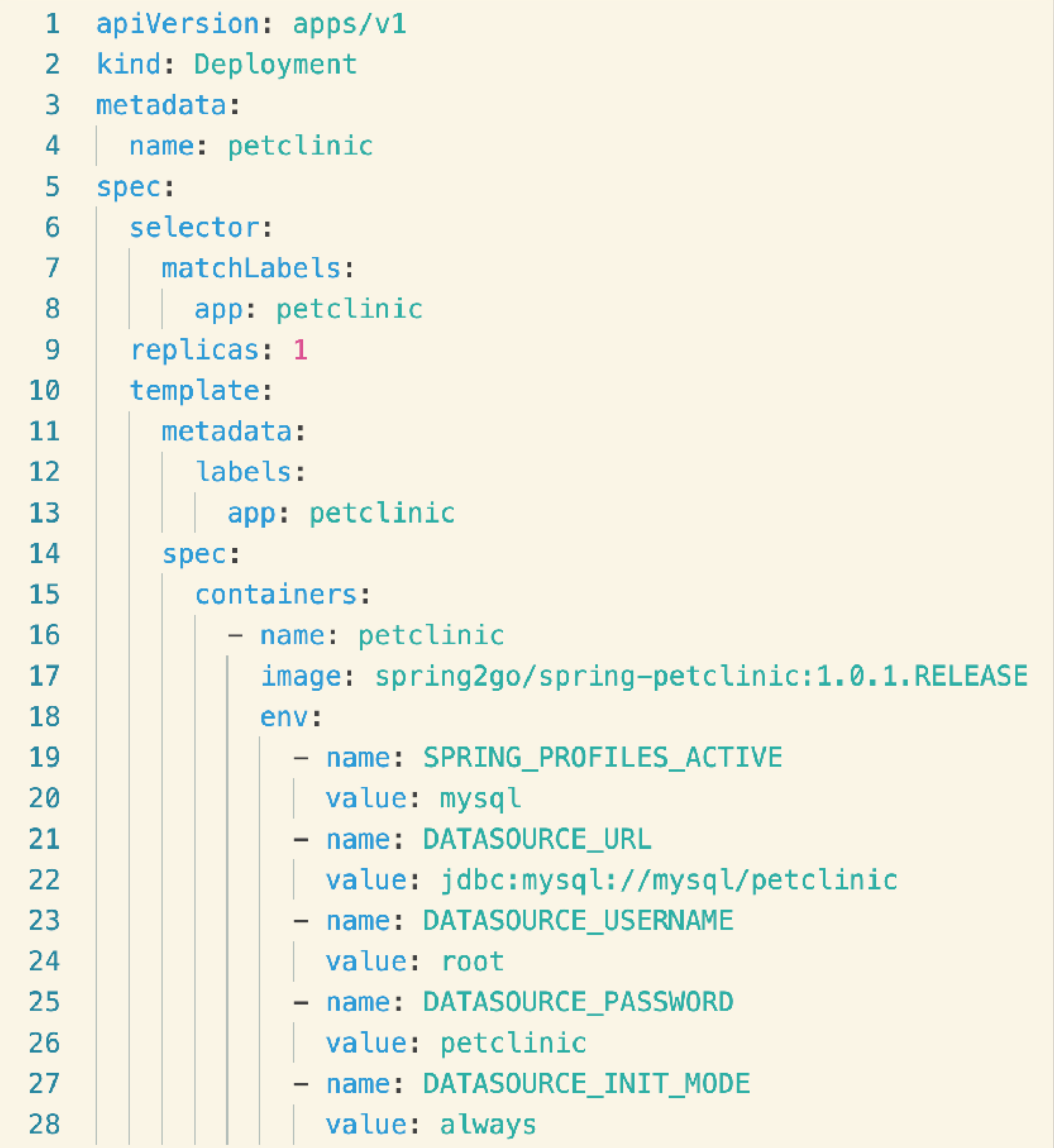


```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mysql
spec:
  selector:
    matchLabels:
      app: mysql
  replicas: 1
  template:
    metadata:
      labels:
        app: mysql
    spec:
      containers:
        - name: mysql
          image: mysql:5.7
          env:
            - name: MYSQL_ROOT_PASSWORD
              value: petclinic
            - name: MYSQL_DATABASE
              value: petclinic
          volumeMounts:
            - name: mysql-persistent-volume
              mountPath: /var/lib/mysql
      volumes:
        - name: mysql-persistent-volume
          persistentVolumeClaim:
            claimName: mysql-pvc
```

petclinic-svc.yml

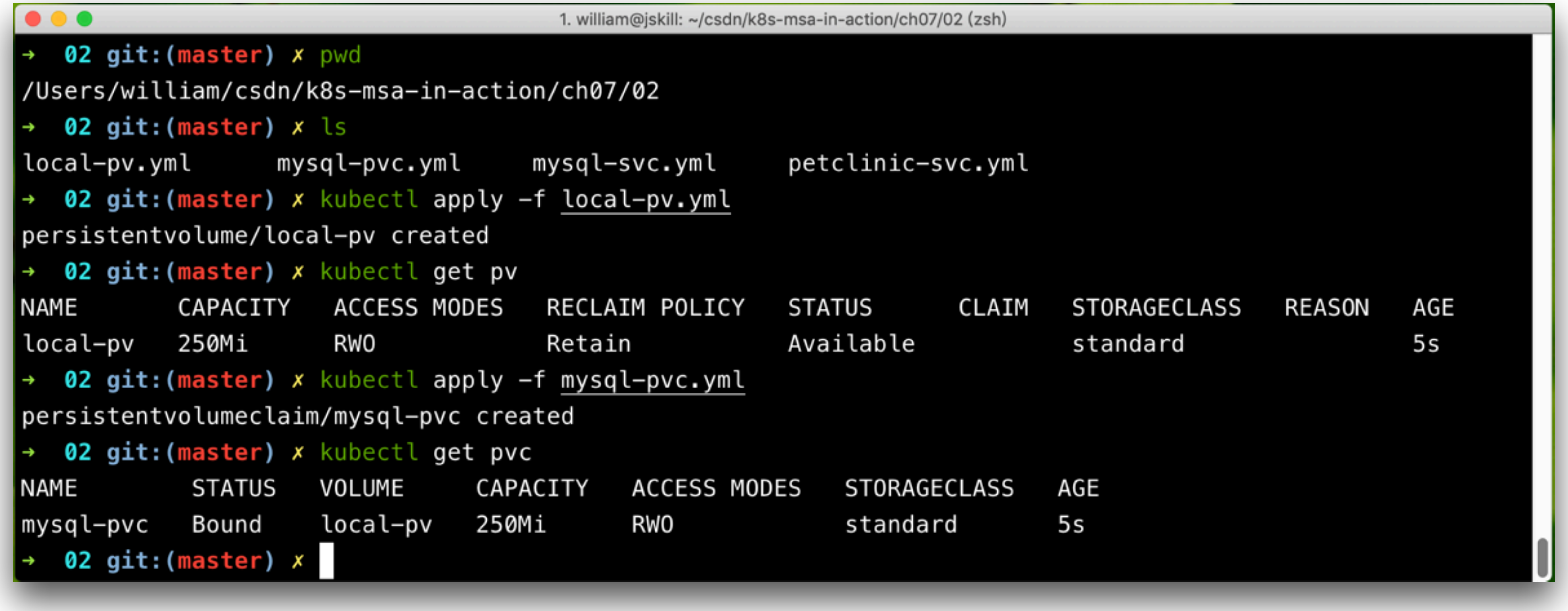


```yaml
apiVersion: v1
kind: Service
metadata:
  name: petclinic
spec:
  ports:
    - name: http
      port: 8080
      targetPort: 8080
      nodePort: 31080
  selector:
    app: petclinic
  type: NodePort
```



```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: petclinic
spec:
  selector:
    matchLabels:
      app: petclinic
  replicas: 1
  template:
    metadata:
      labels:
        app: petclinic
    spec:
      containers:
        - name: petclinic
          image: spring2go/spring-petclinic:1.0.1.RELEASE
          env:
            - name: SPRING_PROFILES_ACTIVE
              value: mysql
            - name: DATASOURCE_URL
              value: jdbc:mysql://mysql/petclinic
            - name: DATASOURCE_USERNAME
              value: root
            - name: DATASOURCE_PASSWORD
              value: petclinic
            - name: DATASOURCE_INIT_MODE
              value: always
```

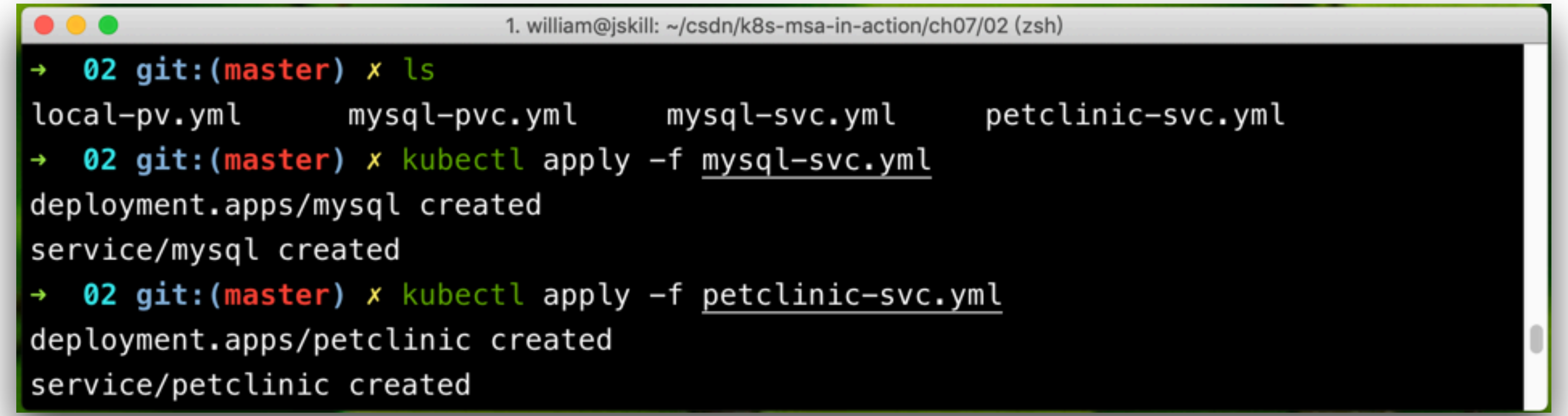
发布local-pv.yml和mysql-pvc.yml



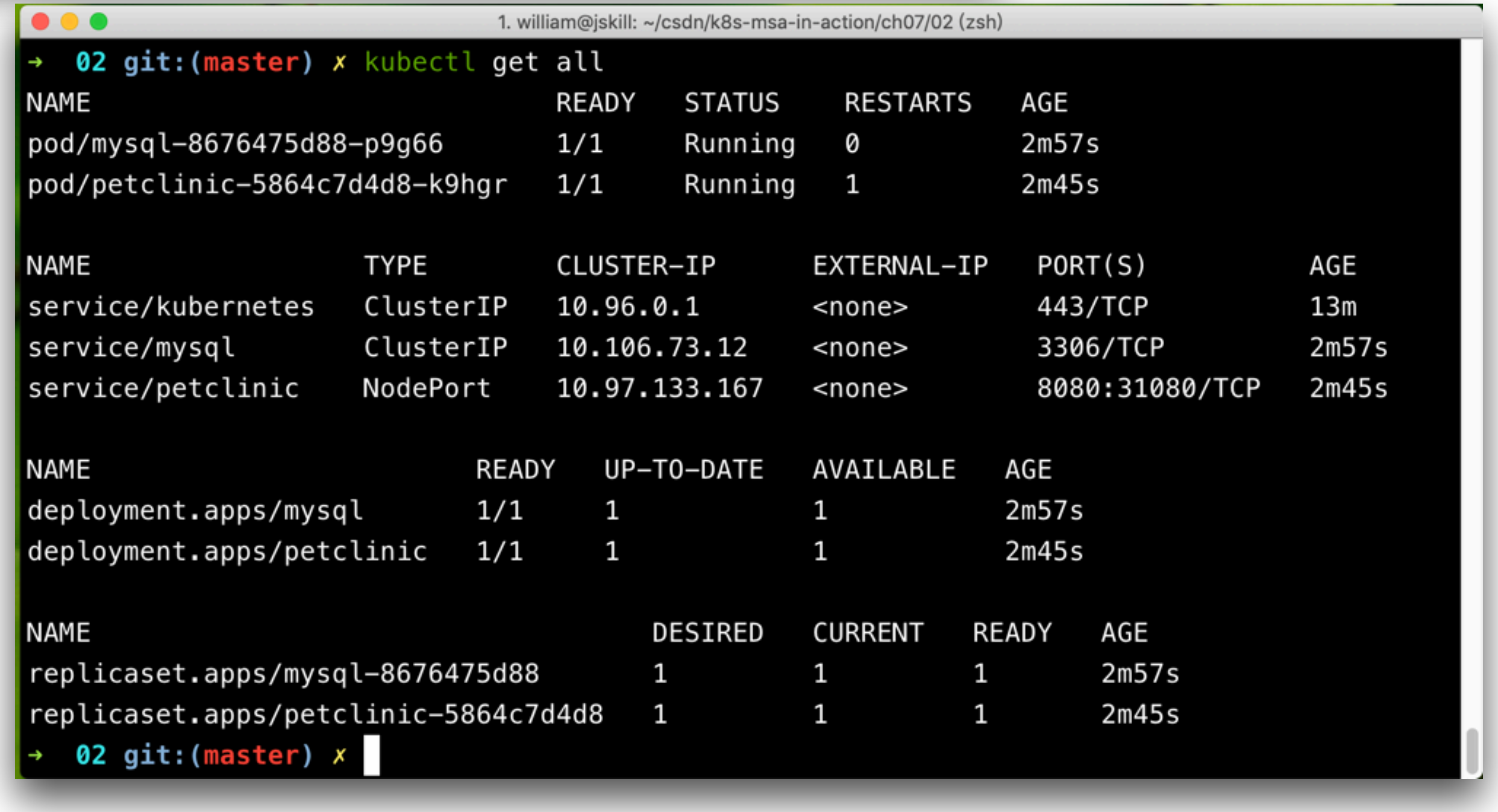
1. william@jskill: ~/csdn/k8s-msa-in-action/ch07/02 (zsh)

```
➜  02 git:(master) ✗ pwd
/Users/william/csdn/k8s-msa-in-action/ch07/02
➜  02 git:(master) ✗ ls
local-pv.yml      mysql-pvc.yml     mysql-svc.yml     petclinic-svc.yml
➜  02 git:(master) ✗ kubectl apply -f local-pv.yml
persistentvolume/local-pv created
➜  02 git:(master) ✗ kubectl get pv
NAME       CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS      CLAIM   STORAGECLASS   REASON   AGE
local-pv   250Mi      RWO            Retain           Available           standard                5s
➜  02 git:(master) ✗ kubectl apply -f mysql-pvc.yml
persistentvolumeclaim/mysql-pvc created
➜  02 git:(master) ✗ kubectl get pvc
NAME        STATUS   VOLUME     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
mysql-pvc   Bound    local-pv   250Mi      RWO            standard       5s
➜  02 git:(master) ✗
```

发布mysql-svc.yml和petclinic-svc.yml

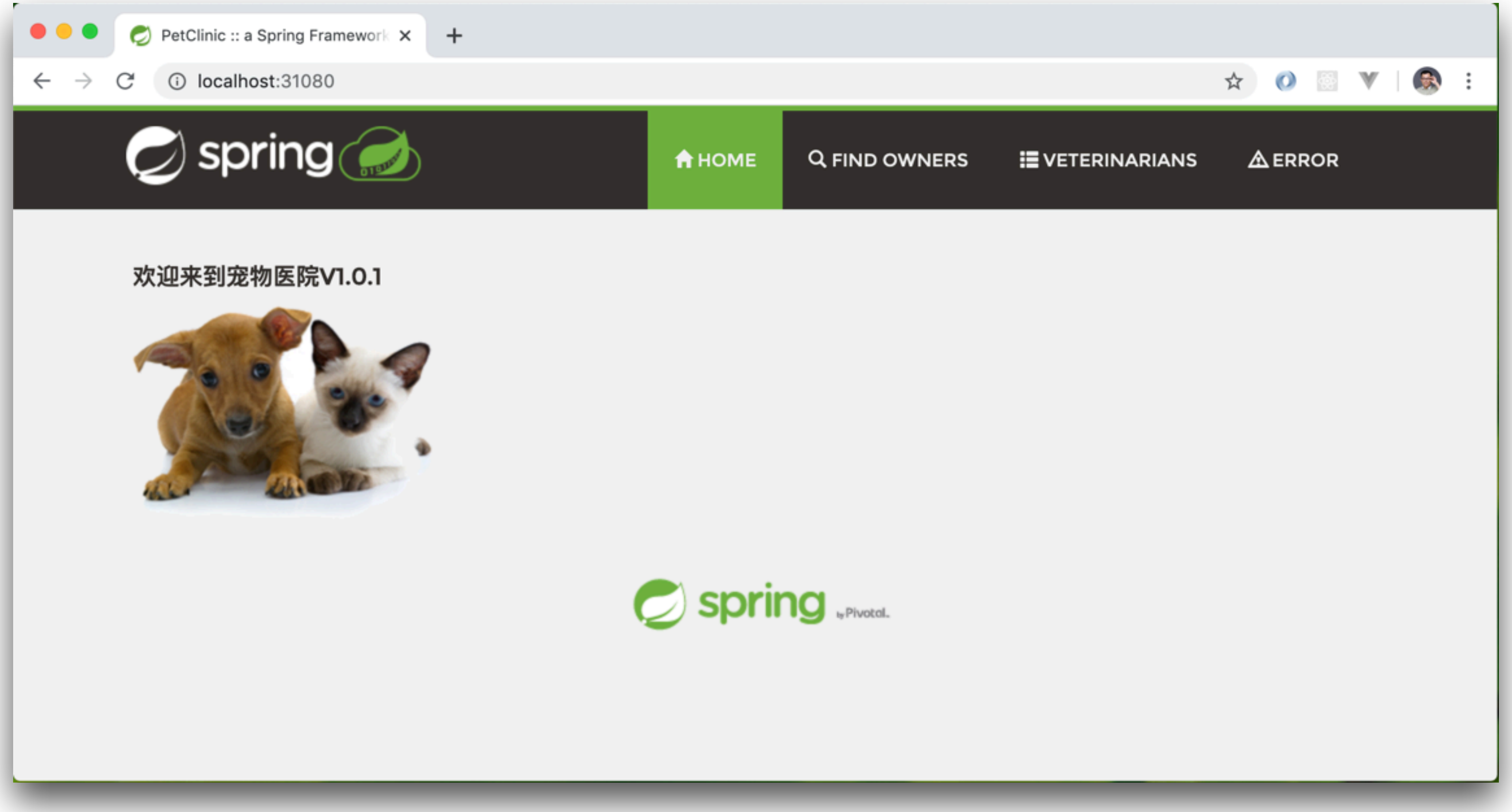


```
→  02 git:(master) ✗ ls
local-pv.yml        mysql-pvc.yml       mysql-svc.yml       petclinic-svc.yml
→  02 git:(master) ✗ kubectl apply -f mysql-svc.yml
deployment.apps/mysql created
service/mysql created
→  02 git:(master) ✗ kubectl apply -f petclinic-svc.yml
deployment.apps/petclinic created
service/petclinic created
```



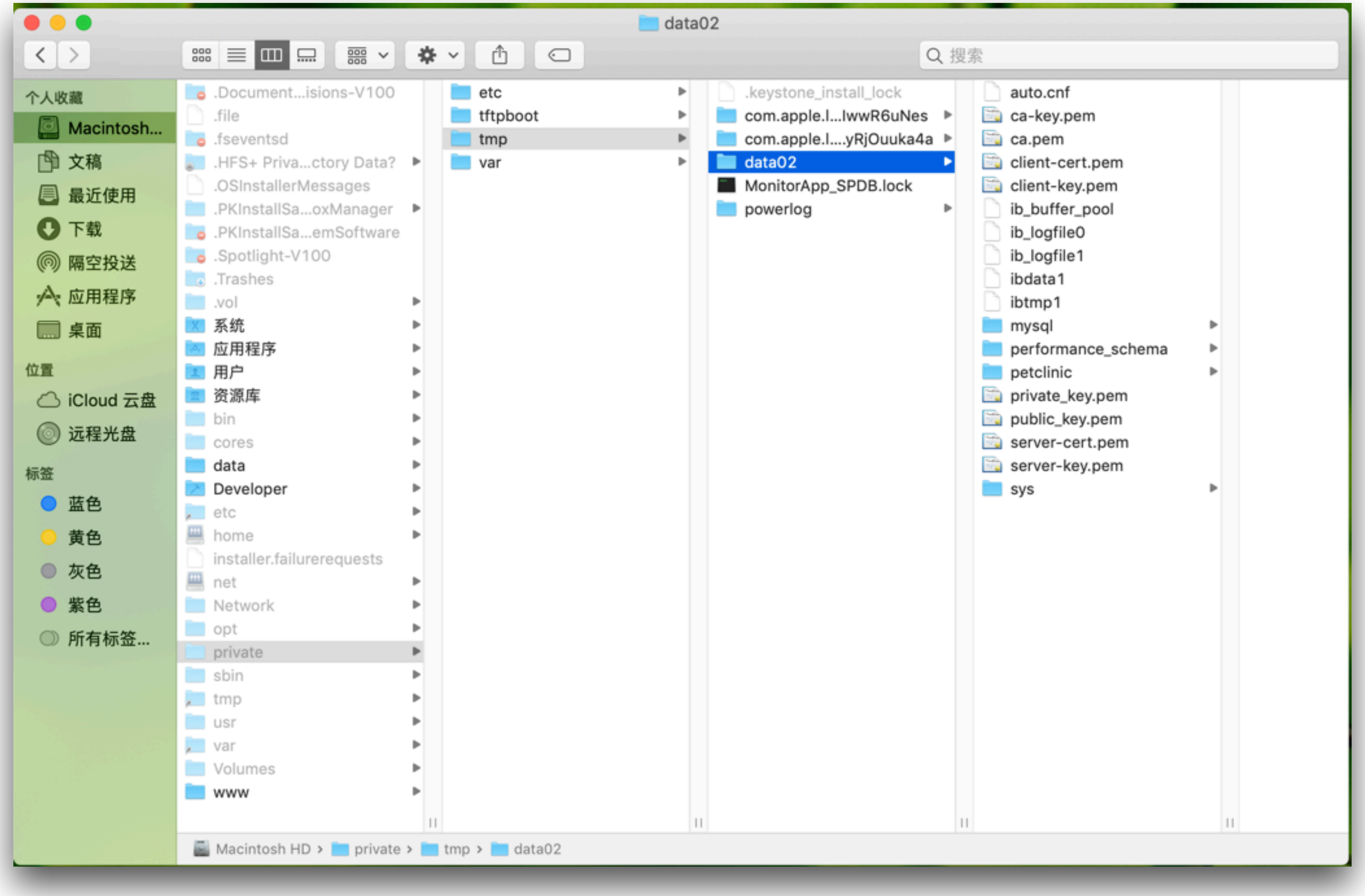
```
→  02 git:(master) ✗ kubectl get all
NAME                                 READY   STATUS    RESTARTS   AGE
pod/mysql-8676475d88-p9g66           1/1     Running   0          2m57s
pod/petclinic-5864c7d4d8-k9hgr       1/1     Running   1          2m45s

NAME                    TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
service/kubernetes      ClusterIP   10.96.0.1       <none>        443/TCP          13m
service/mysql           ClusterIP   10.106.73.12    <none>        3306/TCP         2m57s
service/petclinic       NodePort    10.97.133.167   <none>        8080:31080/TCP   2m45s

NAME                        READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/mysql       1/1     1            1           2m57s
deployment.apps/petclinic   1/1     1            1           2m45s

NAME                                   DESIRED   CURRENT   READY   AGE
replicaset.apps/mysql-8676475d88       1         1         1       2m57s
replicaset.apps/petclinic-5864c7d4d8   1         1         1       2m45s
→  02 git:(master) ✗
```

浏览器校验成功



欢迎来到宠物医院V1.0.1

tmp/data02目录下生成mysql和petclinic数据文件



```
.Document...isions-V100
.file
.fseventsd
.HFS+ Priva...ctory Data?
.OSInstallerMessages
.PKInstallSa...oxManager
.PKInstallSa...emSoftware
.Spotlight-V100
.Trashes
.vol
系统
应用程序
用户
资源库
bin
cores
data
Developer
etc
home
installer.failurerequests
net
Network
opt
private
sbin
tmp
usr
var
Volumes
www
```

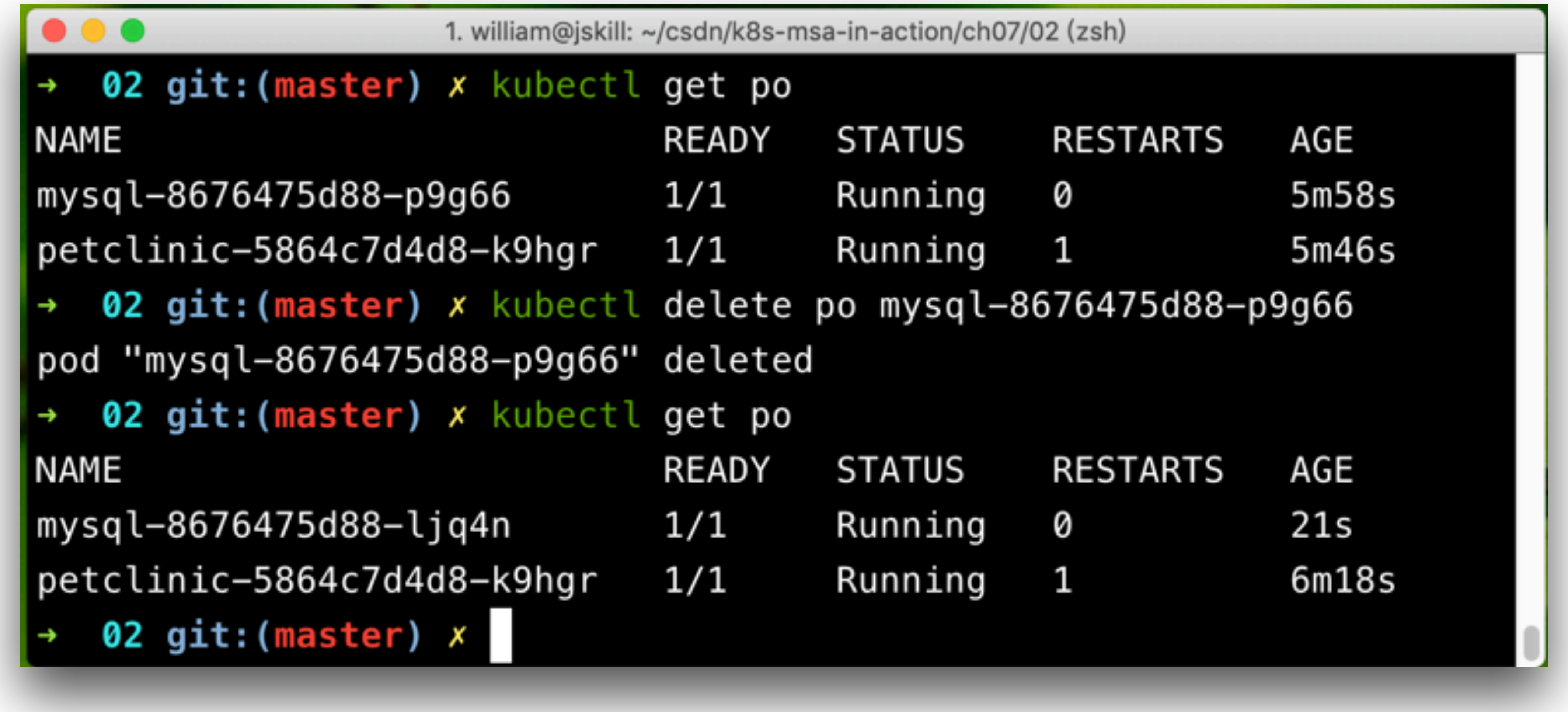
```
etc
tftpboot
tmp
var
```

```
.keystone_install_lock
com.apple.l...lwwR6uNes
com.apple.l....yRjOuuka4a
data02
MonitorApp_SPDB.lock
powerlog
```

```
auto.cnf
ca-key.pem
ca.pem
client-cert.pem
client-key.pem
ib_buffer_pool
ib_logfile0
ib_logfile1
ibdata1
ibtmp1
mysql
performance_schema
petclinic
private_key.pem
public_key.pem
server-cert.pem
server-key.pem
sys
```

Macintosh HD › private › tmp › data02

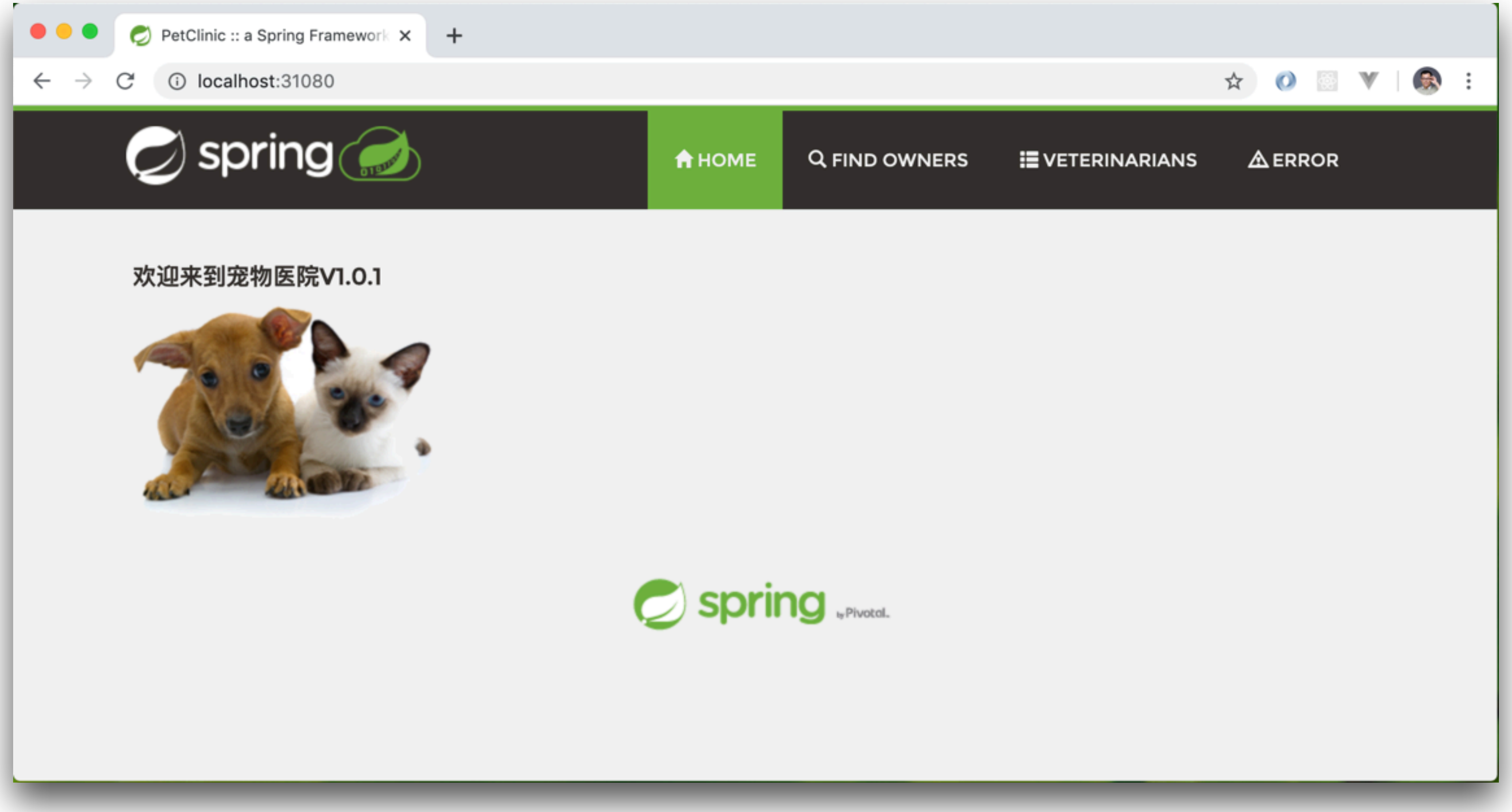
删除并重启mysql pod



1. william@jskill: ~/csdn/k8s-msa-in-action/ch07/02 (zsh)

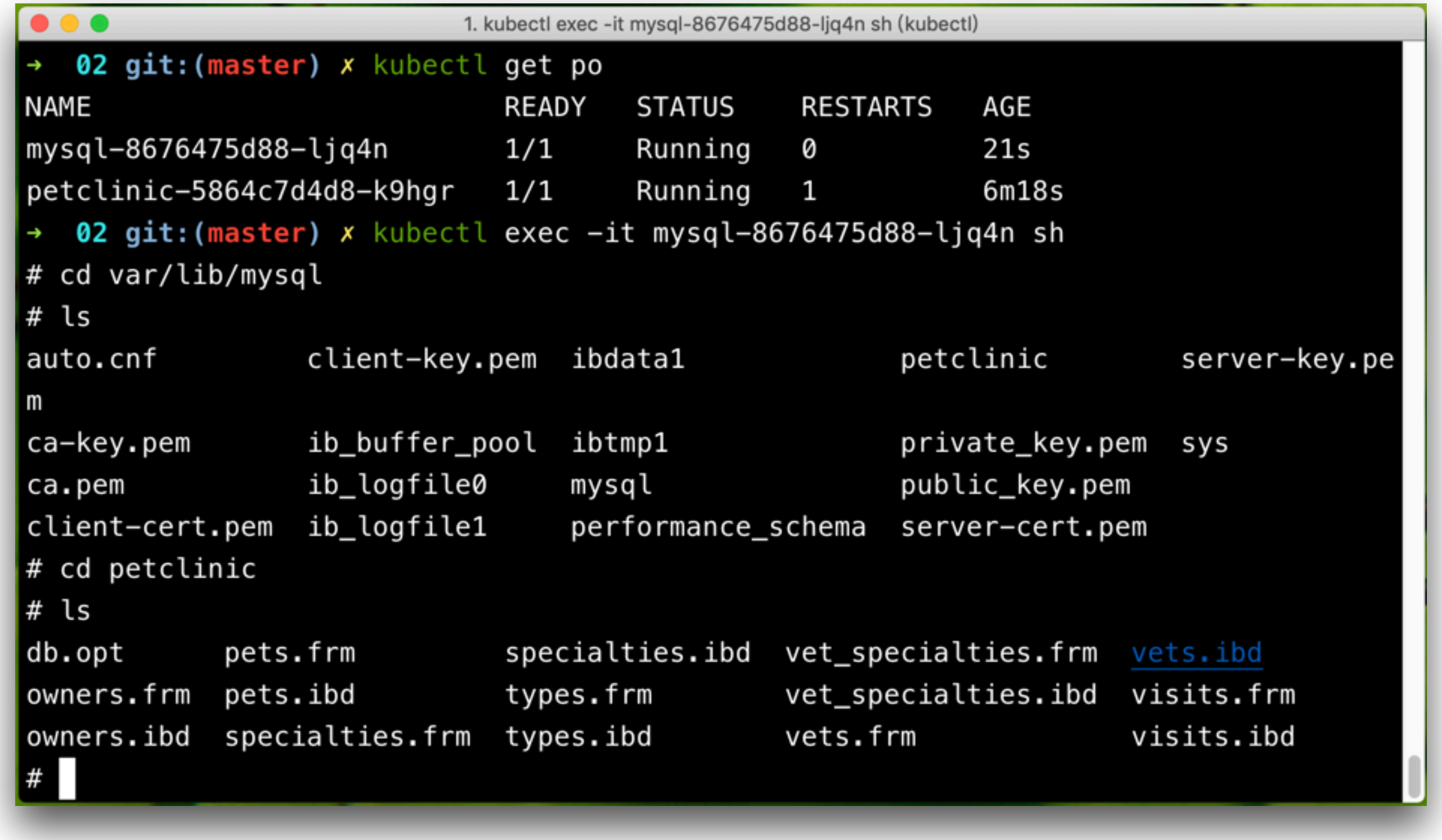
```
➜  02 git:(master) ✗ kubectl get po
NAME                          READY   STATUS    RESTARTS   AGE
mysql-8676475d88-p9g66        1/1     Running   0          5m58s
petclinic-5864c7d4d8-k9hgr    1/1     Running   1          5m46s
➜  02 git:(master) ✗ kubectl delete po mysql-8676475d88-p9g66
pod "mysql-8676475d88-p9g66" deleted
➜  02 git:(master) ✗ kubectl get po
NAME                          READY   STATUS    RESTARTS   AGE
mysql-8676475d88-ljq4n        1/1     Running   0          21s
petclinic-5864c7d4d8-k9hgr    1/1     Running   1          6m18s
➜  02 git:(master) ✗
```

浏览器校验仍然成功



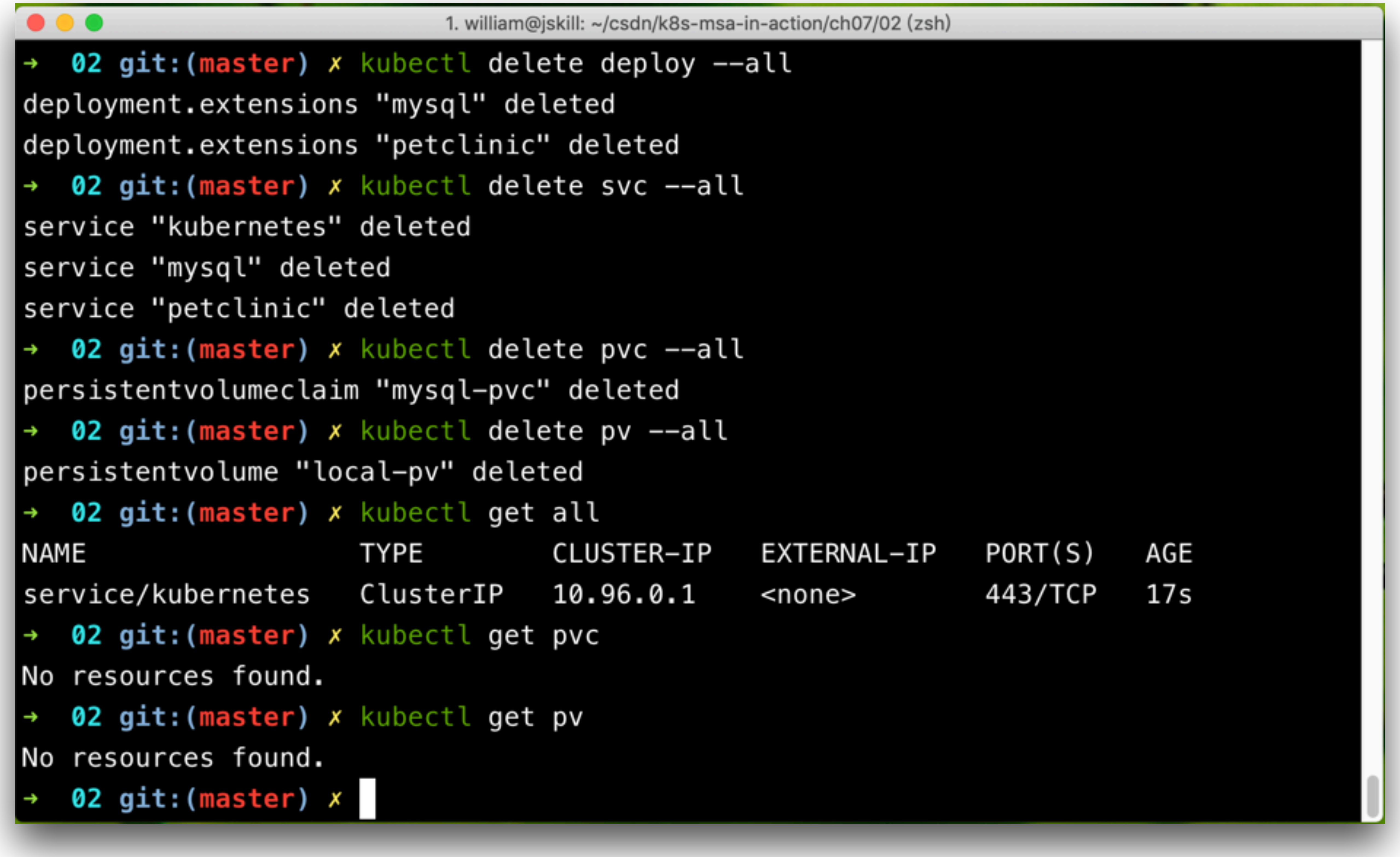
欢迎来到宠物医院V1.0.1

var/lib/mysql表和数据都存在



```
➜  02 git:(master) ✗ kubectl get po
NAME                             READY     STATUS    RESTARTS   AGE
mysql-8676475d88-ljq4n           1/1       Running   0          21s
petclinic-5864c7d4d8-k9hgr       1/1       Running   1          6m18s
➜  02 git:(master) ✗ kubectl exec -it mysql-8676475d88-ljq4n sh
# cd var/lib/mysql
# ls
auto.cnf          client-key.pem  ibdata1             petclinic         server-key.pe
m
ca-key.pem        ib_buffer_pool  ibtmp1              private_key.pem   sys
ca.pem            ib_logfile0     mysql               public_key.pem
client-cert.pem   ib_logfile1     performance_schema  server-cert.pem
# cd petclinic
# ls
db.opt      pets.frm         specialties.ibd  vet_specialties.frm  vets.ibd
owners.frm  pets.ibd         types.frm        vet_specialties.ibd  visits.frm
owners.ibd  specialties.frm  types.ibd        vets.frm             visits.ibd
# 
```

环境清理



```
➜  02 git:(master) ✗ kubectl delete deploy --all
deployment.extensions "mysql" deleted
deployment.extensions "petclinic" deleted
➜  02 git:(master) ✗ kubectl delete svc --all
service "kubernetes" deleted
service "mysql" deleted
service "petclinic" deleted
➜  02 git:(master) ✗ kubectl delete pvc --all
persistentvolumeclaim "mysql-pvc" deleted
➜  02 git:(master) ✗ kubectl delete pv --all
persistentvolume "local-pv" deleted
➜  02 git:(master) ✗ kubectl get all
NAME                 TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   17s
➜  02 git:(master) ✗ kubectl get pvc
No resources found.
➜  02 git:(master) ✗ kubectl get pv
No resources found.
➜  02 git:(master) ✗
```

本课小结

- PersistentVolumeClaim(PVC)是K8s支持的一种存储解耦机制，PVC在Volume和PV之间引入了一层间接。

- PersistentVolume(PV)是K8s支持的持久化存储抽象，底层可以对接各种物理存储机制。

- Pod -> Volume -> PVC -> PV -> 物理存储