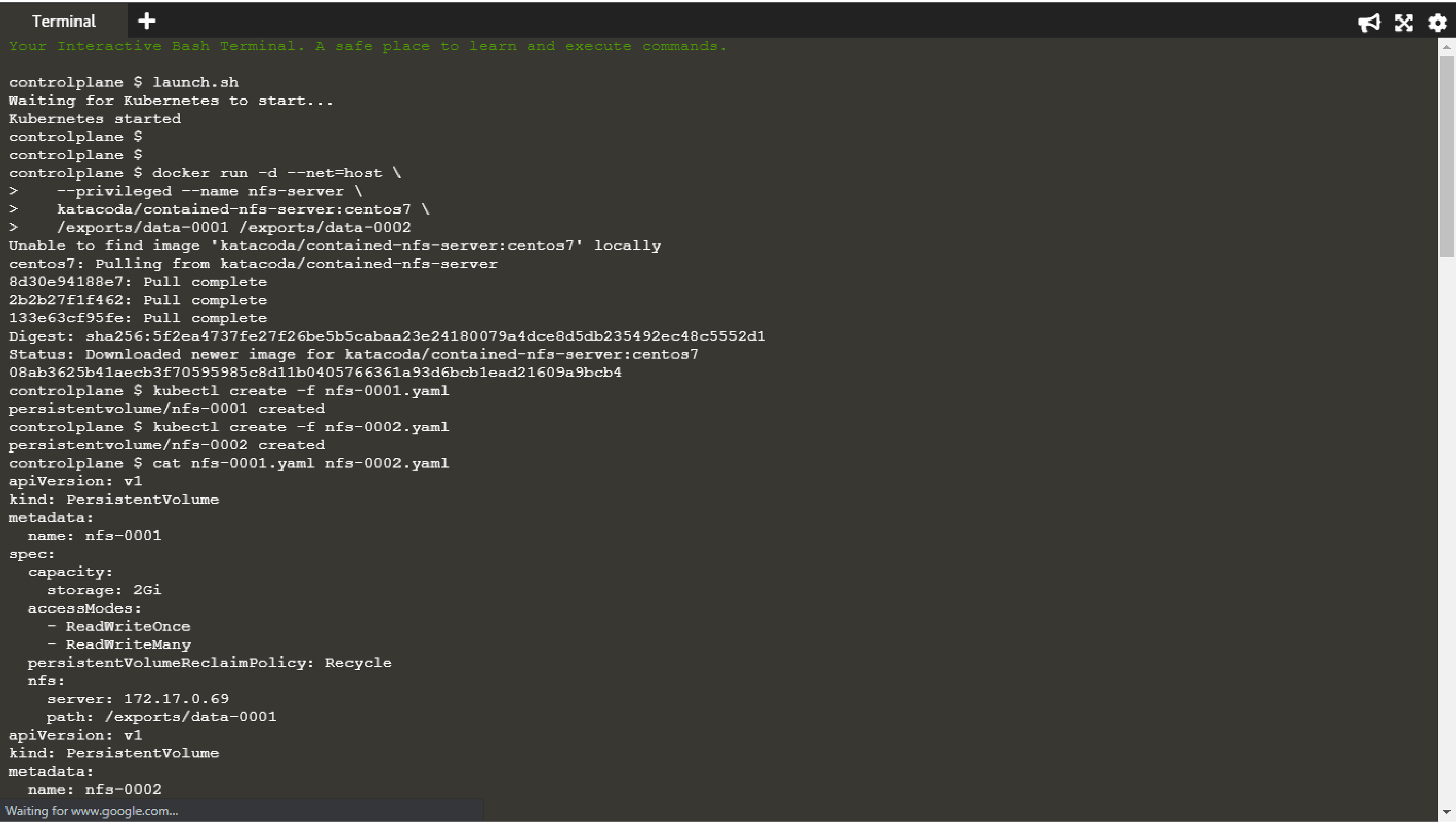
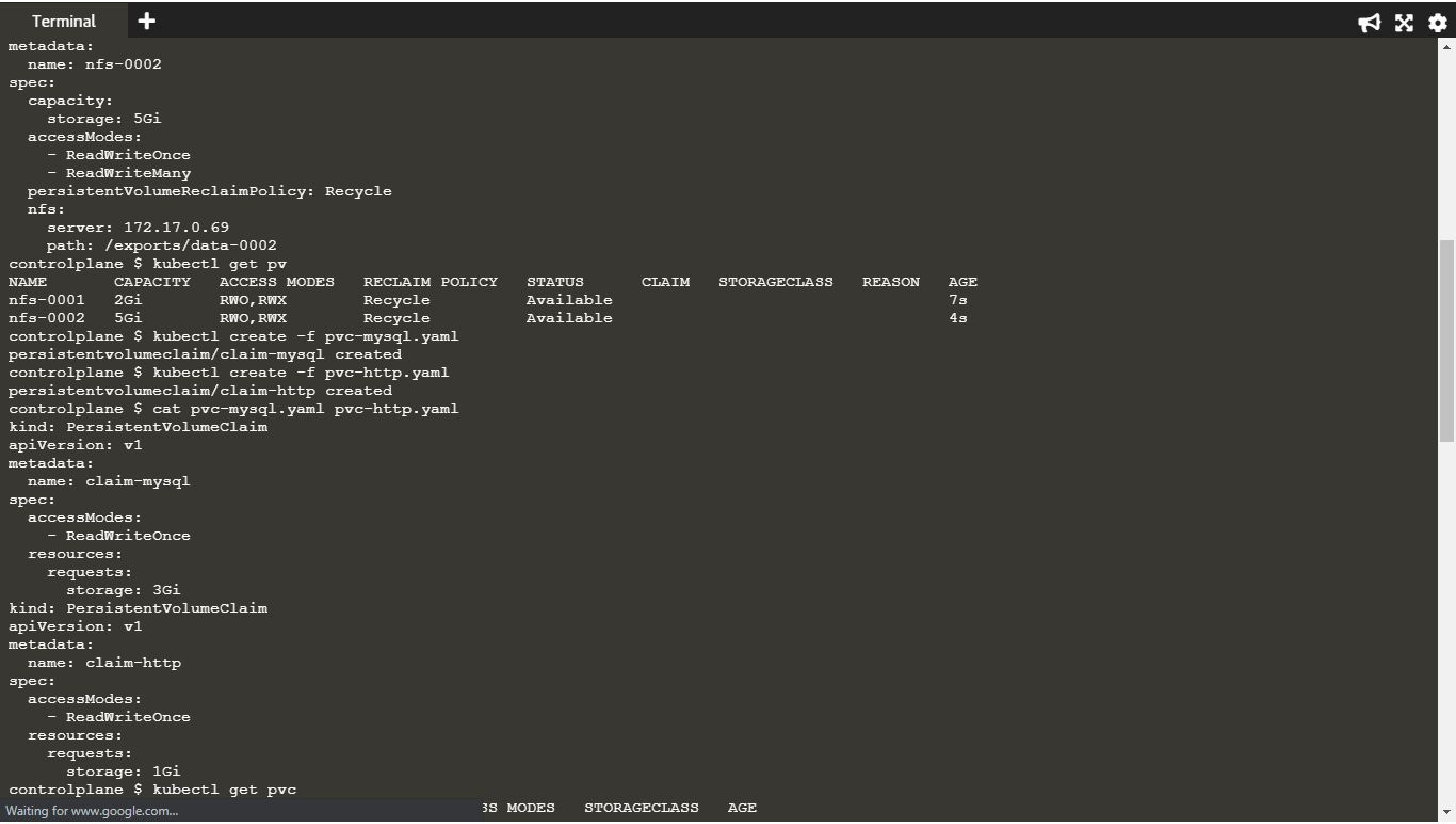
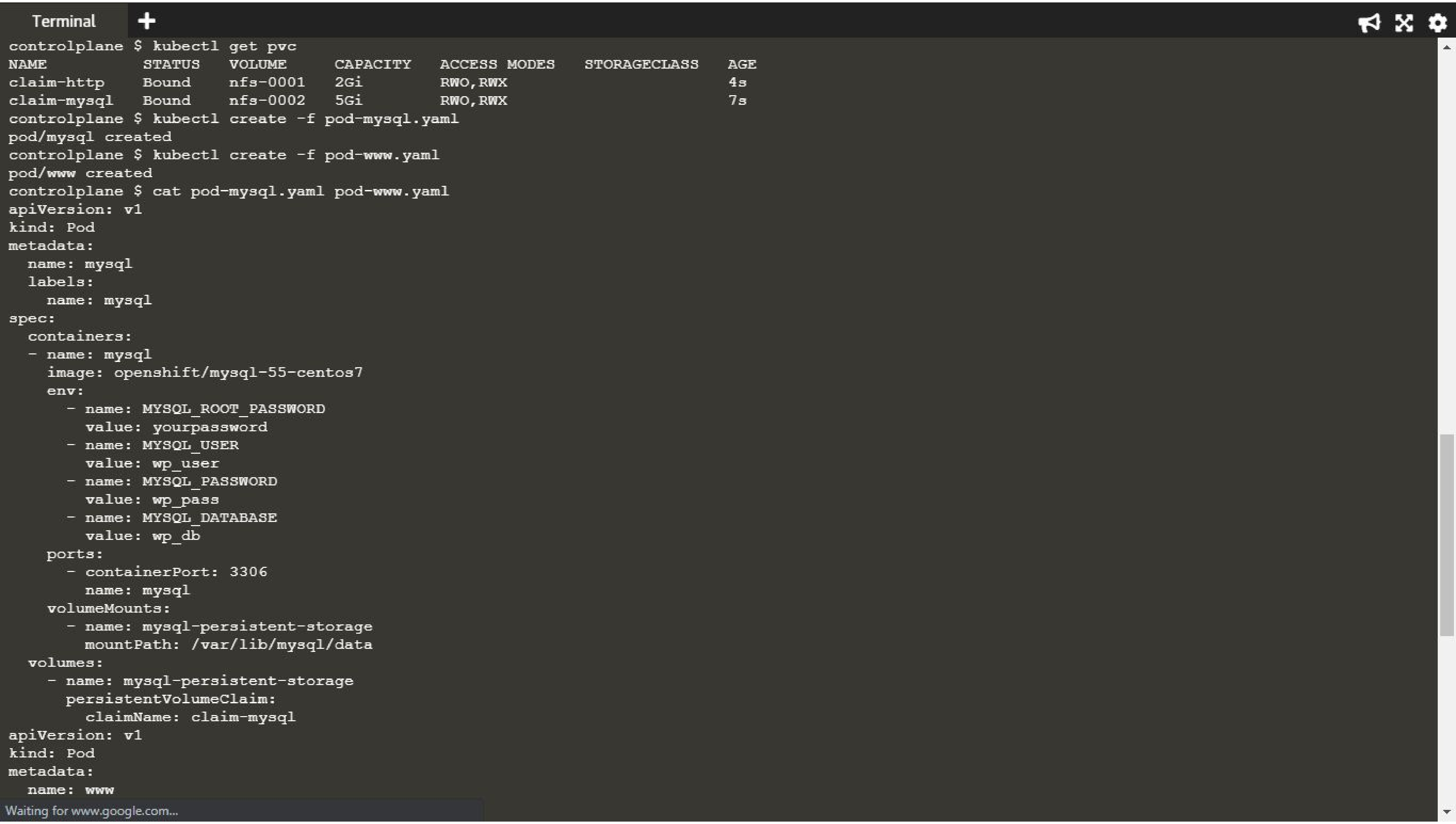


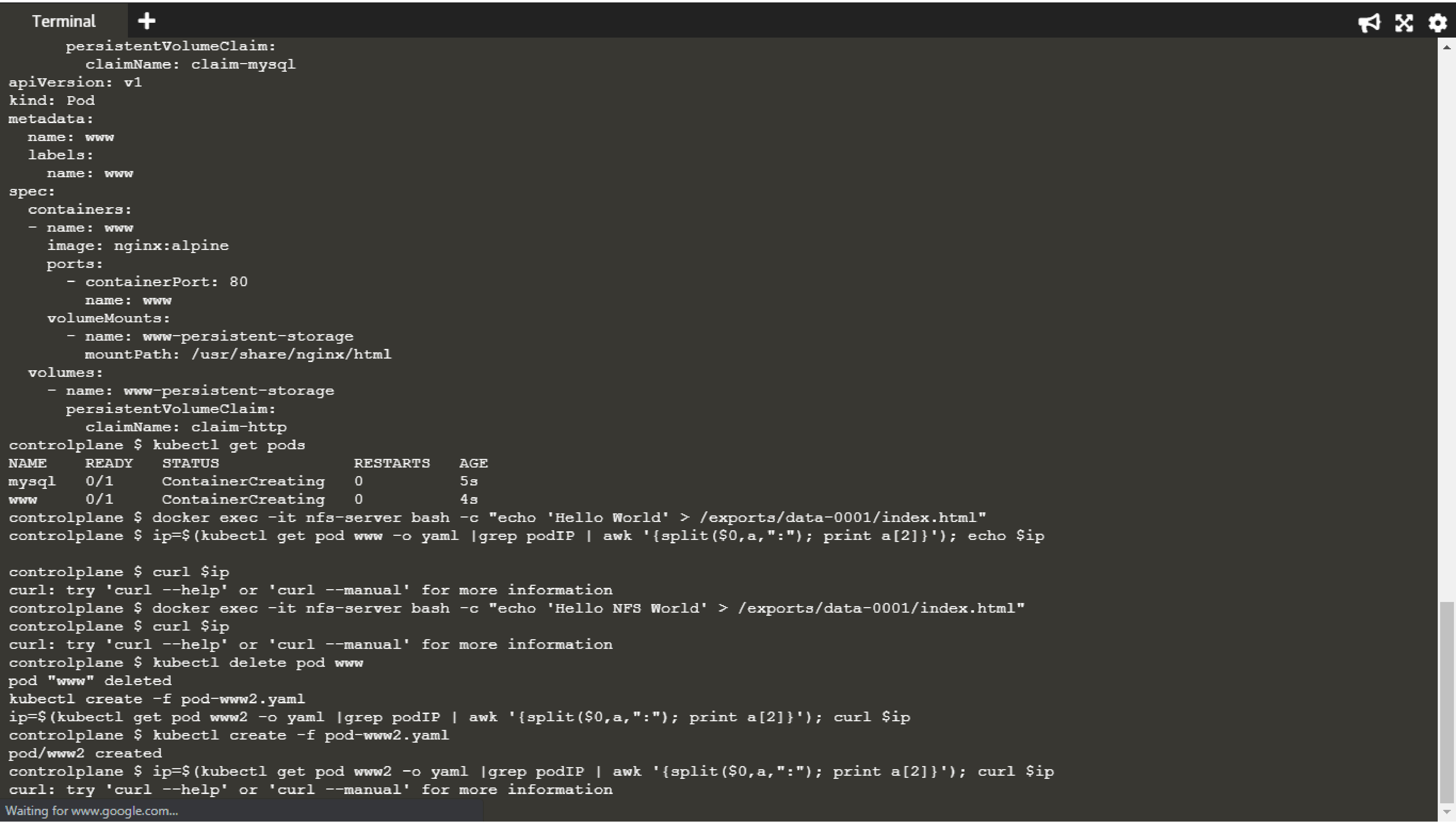
```
Your Interactive Bash Terminal. A safe place to learn and execute commands.

controlplane $ launch.sh
Waiting for Kubernetes to start...
Kubernetes started
controlplane $
controlplane $
controlplane $ docker run -d --net=host \
>    --privileged --name nfs-server \
>    katacoda/contained-nfs-server:centos7 \
>    /exports/data-0001 /exports/data-0002
Unable to find image 'katacoda/contained-nfs-server:centos7' locally
centos7: Pulling from katacoda/contained-nfs-server
8d30e94188e7: Pull complete
2b2b27f1f462: Pull complete
133e63cf95fe: Pull complete
Digest: sha256:5f2ea4737fe27f26be5b5cabaa23e24180079a4dce8d5db235492ec48c5552d1
Status: Downloaded newer image for katacoda/contained-nfs-server:centos7
08ab3625b41aecb3f70595985c8d11b0405766361a93d6bcb1ead21609a9bcb4
controlplane $ kubectl create -f nfs-0001.yaml
persistentvolume/nfs-0001 created
controlplane $ kubectl create -f nfs-0002.yaml
persistentvolume/nfs-0002 created
controlplane $ cat nfs-0001.yaml nfs-0002.yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: nfs-0001
spec:
  capacity:
    storage: 2Gi
  accessModes:
    - ReadWriteOnce
    - ReadWriteMany
  persistentVolumeReclaimPolicy: Recycle
  nfs:
    server: 172.17.0.69
    path: /exports/data-0001
apiVersion: v1
kind: PersistentVolume
metadata:
  name: nfs-0002
```



```
metadata:
  name: nfs-0002
spec:
  capacity:
    storage: 5Gi
  accessModes:
    - ReadWriteOnce
    - ReadWriteMany
  persistentVolumeReclaimPolicy: Recycle
  nfs:
    server: 172.17.0.69
    path: /exports/data-0002
controlplane $ kubectl get pv
NAME       CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS      CLAIM   STORAGECLASS   REASON   AGE
nfs-0001   2Gi        RWO,RWX        Recycle          Available                                   7s
nfs-0002   5Gi        RWO,RWX        Recycle          Available                                   4s
controlplane $ kubectl create -f pvc-mysql.yaml
persistentvolumeclaim/claim-mysql created
controlplane $ kubectl create -f pvc-http.yaml
persistentvolumeclaim/claim-http created
controlplane $ cat pvc-mysql.yaml pvc-http.yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: claim-mysql
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 3Gi
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: claim-http
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
controlplane $ kubectl get pvc
                                                  3S MODES   STORAGECLASS   AGE
```



```
controlplane $ kubectl get pvc
NAME          STATUS   VOLUME     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
claim-http    Bound    nfs-0001   2Gi        RWO,RWX                       4s
claim-mysql   Bound    nfs-0002   5Gi        RWO,RWX                       7s
controlplane $ kubectl create -f pod-mysql.yaml
pod/mysql created
controlplane $ kubectl create -f pod-www.yaml
pod/www created
controlplane $ cat pod-mysql.yaml pod-www.yaml
apiVersion: v1
kind: Pod
metadata:
  name: mysql
  labels:
    name: mysql
spec:
  containers:
  - name: mysql
    image: openshift/mysql-55-centos7
    env:
      - name: MYSQL_ROOT_PASSWORD
        value: yourpassword
      - name: MYSQL_USER
        value: wp_user
      - name: MYSQL_PASSWORD
        value: wp_pass
      - name: MYSQL_DATABASE
        value: wp_db
    ports:
      - containerPort: 3306
        name: mysql
    volumeMounts:
      - name: mysql-persistent-storage
        mountPath: /var/lib/mysql/data
  volumes:
    - name: mysql-persistent-storage
      persistentVolumeClaim:
        claimName: claim-mysql
apiVersion: v1
kind: Pod
metadata:
  name: www
```



```
      persistentVolumeClaim:
        claimName: claim-mysql
apiVersion: v1
kind: Pod
metadata:
  name: www
  labels:
    name: www
spec:
  containers:
  - name: www
    image: nginx:alpine
    ports:
      - containerPort: 80
        name: www
    volumeMounts:
      - name: www-persistent-storage
        mountPath: /usr/share/nginx/html
  volumes:
    - name: www-persistent-storage
      persistentVolumeClaim:
        claimName: claim-http
controlplane $ kubectl get pods
NAME    READY   STATUS              RESTARTS   AGE
mysql   0/1     ContainerCreating   0          5s
www     0/1     ContainerCreating   0          4s
controlplane $ docker exec -it nfs-server bash -c "echo 'Hello World' > /exports/data-0001/index.html"
controlplane $ ip=$(kubectl get pod www -o yaml |grep podIP | awk '{split($0,a,":"); print a[2]}'); echo $ip

controlplane $ curl $ip
curl: try 'curl --help' or 'curl --manual' for more information
controlplane $ docker exec -it nfs-server bash -c "echo 'Hello NFS World' > /exports/data-0001/index.html"
controlplane $ curl $ip
curl: try 'curl --help' or 'curl --manual' for more information
controlplane $ kubectl delete pod www
pod "www" deleted
kubectl create -f pod-www2.yaml
ip=$(kubectl get pod www2 -o yaml |grep podIP | awk '{split($0,a,":"); print a[2]}'); curl $ip
controlplane $ kubectl create -f pod-www2.yaml
pod/www2 created
controlplane $ ip=$(kubectl get pod www2 -o yaml |grep podIP | awk '{split($0,a,":"); print a[2]}'); curl $ip
curl: try 'curl --help' or 'curl --manual' for more information
```