



What the heck is serverless computing? Find out in our latest episode of the [Weekly Update](https://www.youtube.com/watch?v=KITuH9w0AoO) (https://www.youtube.com/watch?v=KITuH9w0AoO)!

Creating Persistent Storage for Pods in Kubernetes

28 Min. Remaining Intermediate (/search?type=Hands-On Lab-Live Environment Learning Activity&difficulty=Intermediate&categories=Containers)

Cancel Lab Complete Lab

How was this lab?

Credentials

Usage Help (https://support.linuxacademy.com/hc/en-us/articles/360028198971)

Cloud Server

Kube Master

Username

cloud_user

Password

KZHZtaYLje

Kube Master Public IP

54.224.66.212 (http://guac.linuxacademy.com/?a=3269ac86a149a545db85&b=1e7260326dd2e5710c16)

How do I connect? (https://support.linuxacademy.com/hc/en-us/articles/360028198971-Connecting-to-Hands-On-Labs)

Additional Information and Resources

You have been given access to a two-node cluster. You must first create a PersistentVolume object in Kubernetes. Once the PersistentVolume has been created, you must create a PersistentVolumeClaim in order for you to claim that volume for the pod. Once you have your PersistentVolume and PersistentVolumeClaim, you are now ready to create the pod.

Create the pod with the image mongodb and include the volume, mounted to the /data/db directory. Then, delete the pod and create a new pod that will mount that same volume. Perform the following tasks in order to complete this hands-on lab:

- Create a PersistentVolume.
- Create a PersistentVolumeClaim.
- Create the pod with the volume mounted to it.
- Delete the pod.
- Recreate the pod with the same YAML.
- Verify the data still resides on the volume.

Learning Objectives

✓ Create a PersistentVolume.

1. Use the following YAML spec for the PersistentVolume named `mongodb-pv.yaml` :

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: mongodb-pv
spec:
  storageClassName: local-storage
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: "/mnt/data"
```

✓ Create a PersistentVolumeClaim.

2. Then, create the PersistentVolume:

1. Use the following YAML spec for the PersistentVolumeClaim named `mongodb-pvc.yaml` :

```
kubectl apply -f mongodb-pv.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mongodb-pvc
spec:
  storageClassName: local-storage
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
```

2. Then, create the PersistentVolumeClaim:

```
kubectl apply -f mongodb-pvc.yaml
```

✓ Create a pod from the mongodb image, with a mounted volume to mount path `/data/db`.

1. Use the following YAML spec for the pod named `mongodb-pod.yaml` :

```
apiVersion: v1
kind: Pod
metadata:
  name: mongodb
spec:
  containers:
    - image: mongo
      name: mongodb
      volumeMounts:
        - name: mongodb-data
          mountPath: /data/db
      ports:
        - containerPort: 27017
          protocol: TCP
  volumes:
    - name: mongodb-data
      persistentVolumeClaim:
        claimName: mongodb-pvc
```

2. Then, create the pod:

```
kubectl apply -f mongodb-pod.yaml
```

3. Verify the pod was created:

```
kubectl get pods
```

✓ Access the node and view the data within the volume. ^

1. Connect to the node:

```
ssh <node_hostname>
```

2. Switch to the `/mnt/data` directory:

```
cd /mnt/data
```

3. List the contents of the directory:

```
ls
```

✓ Delete the pod and create a new pod with the same YAML spec. ^

1. Delete the pod:

```
kubectl delete pod mongodb
```

2. Create a new pod:

```
kubectl apply -f mongodb-pod.yaml
```

✓ Verify the data still resides on the volume. ^

1. Log in to the node:

```
ssh <node_hostname>
```

2. Switch to the `/mnt/data` directory:

```
cd /mnt/data
```

3. List the contents of the directory:


```
ls
```

Tools

Instant Terminal

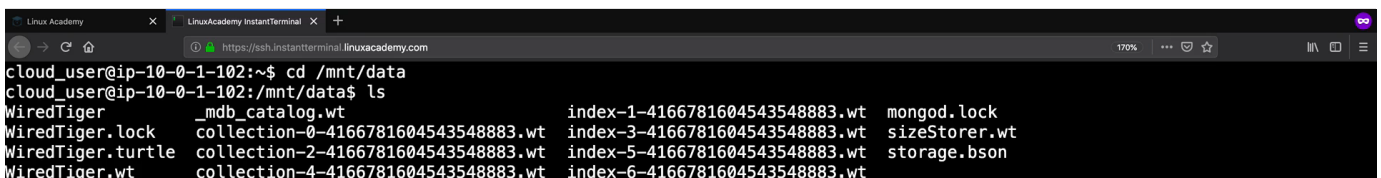
(<https://ssh.instantterminal.linuxacademy.com>)

Diagram

 (<https://support.linuxacademy.com/hc/en-us/articles/360028193131>)

Video

Guide



```
cloud_user@ip-10-0-1-102:~$ cd /mnt/data
cloud_user@ip-10-0-1-102:/mnt/data$ ls
WiredTiger      _mdb_catalog.wt      index-1-4166781604543548883.wt  mongod.lock
WiredTiger.lock collection-0-4166781604543548883.wt index-3-4166781604543548883.wt  sizeStorer.wt
WiredTiger.turtle collection-2-4166781604543548883.wt index-5-4166781604543548883.wt  storage.bson
WiredTiger.wt   collection-4-4166781604543548883.wt index-6-4166781604543548883.wt
```