

Create a PVC, asking Kubernetes for 1 GB of storage using the default storage class:

**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.**

1. Run the below commands on your AWS-Workstation.

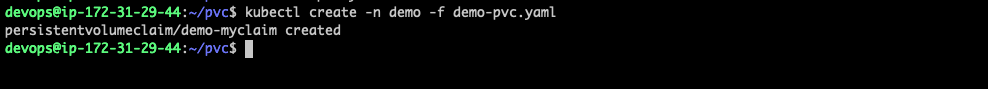
|  |
| --- |
| $ mkdir /home/devops/pvc $ cd /home/devops/pvc |

2. Create a Persistent Volume Claim by running the below command.

|  |
| --- |
| $ vim <your-name>-pvc.yaml |

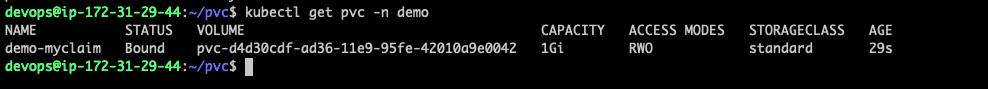
|  |
| --- |
| apiVersion: v1 kind: PersistentVolumeClaim metadata:  name: <your-name>-myclaim spec:  accessModes:  - ReadWriteOnce  resources:  requests:  storage: 1Gi |

|  |
| --- |
| $ kubectl create -n <your-name> -f <your-name>-pvc.yaml |



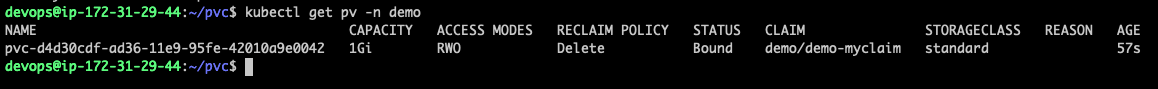
3. Check if the Persistent Volume Claim has been created.

|  |
| --- |
| $ kubectl get pvc -n <your-name> |



4. Check if the Persistent Volume has been created.

|  |
| --- |
| $ kubectl get pv -n <your-name> |

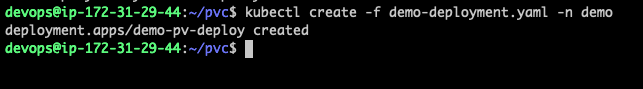


5. Now, create a deployment that uses above PVC to mount it as a volume into /tmp/persistent:

|  |
| --- |
| $ curl -k <https://pastebin.com/raw/MkPuxCXS> > <your-name>-deployment.yaml  $ vim <your-name>-deployment.yaml |

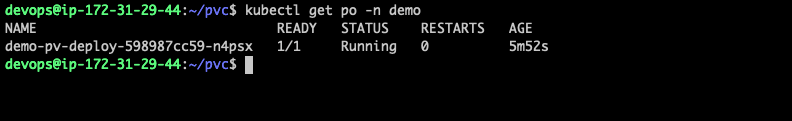
Update <your-name> with your name.

|  |
| --- |
| $ kubectl create -f <your-name>-deployment.yaml -n <your-name> |



6. Now we want to test if data in the volume actually persists. For this we find the pod managed by above deployment, exec into its main container and create a file called data in the /tmp/persistent directory (where we decided to mount the PV):

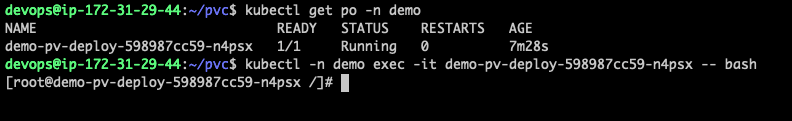
|  |
| --- |
| $ kubectl get po -n <your-name> |



7. Exec into the POD to check the data persistency.

|  |
| --- |
| $ kubectl -n <your-name> exec -it <POD NAME> -- bash |

Ex.



8. Run the below commands

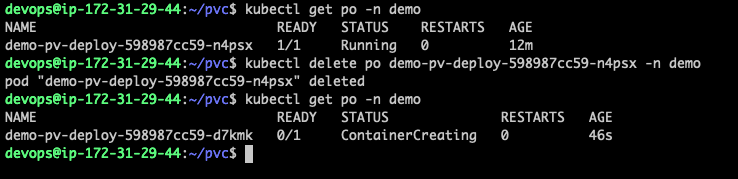
|  |
| --- |
| [root@demo-pv-deploy-598987cc59-n4psx /]# touch /tmp/persistent/<your-name> [root@demo-pv-deploy-598987cc59-n4psx /]# ls /tmp/persistent/ demo lost+found  [root@demo-pv-deploy-598987cc59-n4psx /]# exit |



9. It’s time to destroy the pod and let the deployment launch a new pod. The expectation is that the PV is available again in the new pod and the data in /tmp/persistent is still present. Let’s check that:

|  |
| --- |
| $ kubectl get po -n <your-name>  $ kubectl delete po <POD NAME> -n <your-name>  $ kubectl get po -n <your-name> |

Notice in the below example that the POD was deleted and recreated by the deployment. The POD Name has been updated and the PV has been attached to the new POD



10. Exec into the new POD and check if the DATA persists.

|  |
| --- |
| $ kubectl -n <your-name> exec -it <POD-NAME> -- bash  [root@demo-pv-deploy-598987cc59-d7kmk /]# ls /tmp/persistent/  demo lost+found  [root@demo-pv-deploy-598987cc59-d7kmk /]# exit |

The data file and its content is still where it is expected to be.

**CLEAN UP**

11. Run the below command to destroy the PVC

|  |
| --- |
| $ kubectl delete pvc/<your-name>-myclaim deploy/<your-name>-pv-deploy -n <your-name> |