

# Persistent Volumes (PVs) in Kubernetes

Persistent Volumes (PVs) are a key component in Kubernetes for managing storage resources. They provide a way to abstract and manage storage independently of pods, ensuring data persistence even when pods are deleted or rescheduled.

## Persistent Volume (PV)

A cluster-wide storage resource provisioned by an administrator.

Represents a piece of storage (e.g., NFS, AWS EBS, GCP Persistent Disk, local storage).

Has a lifecycle independent of pods.

## Persistent Volume Claim (PVC)

A request for storage by a user or application.

Binds to a PV that matches the requested size and access mode.

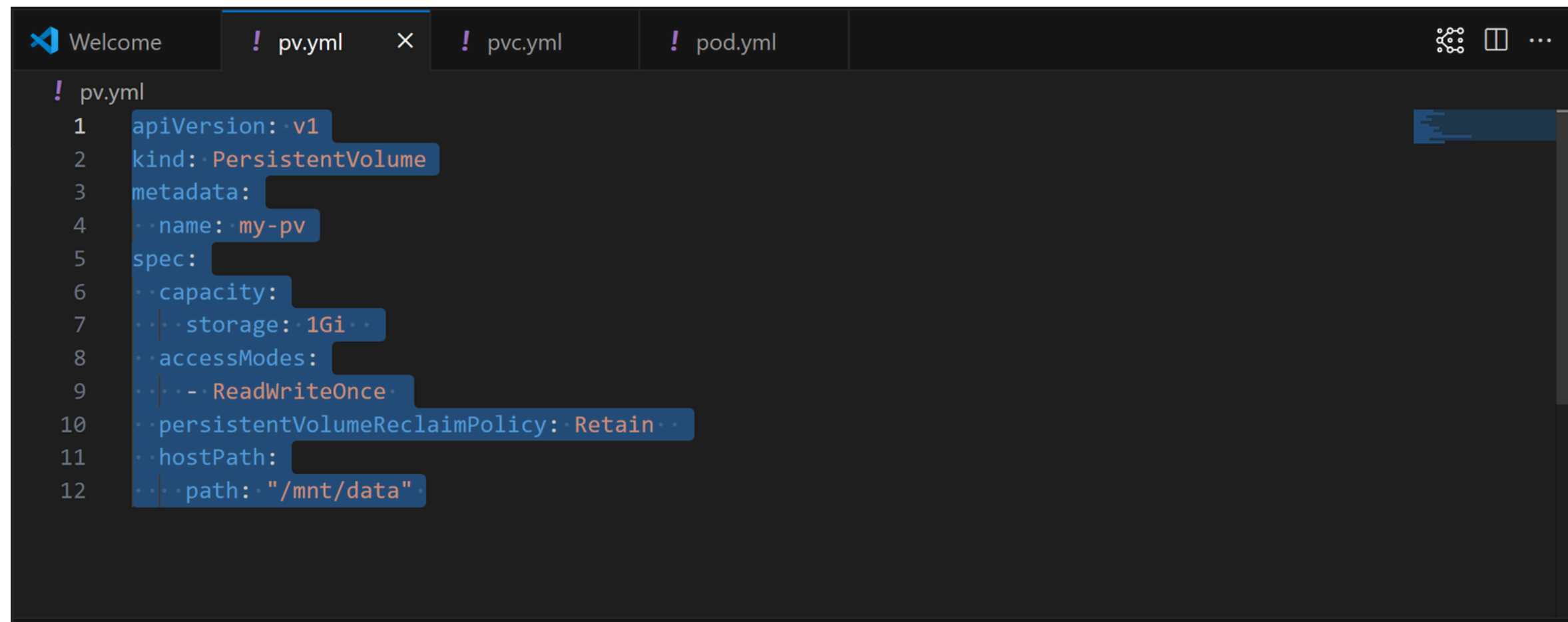
Acts as a "ticket" for pods to access the storage.

## Access Modes

ReadWriteOnce (RWO): Read-write by a single node.

ReadOnlyMany (ROX): Read-only by multiple nodes.

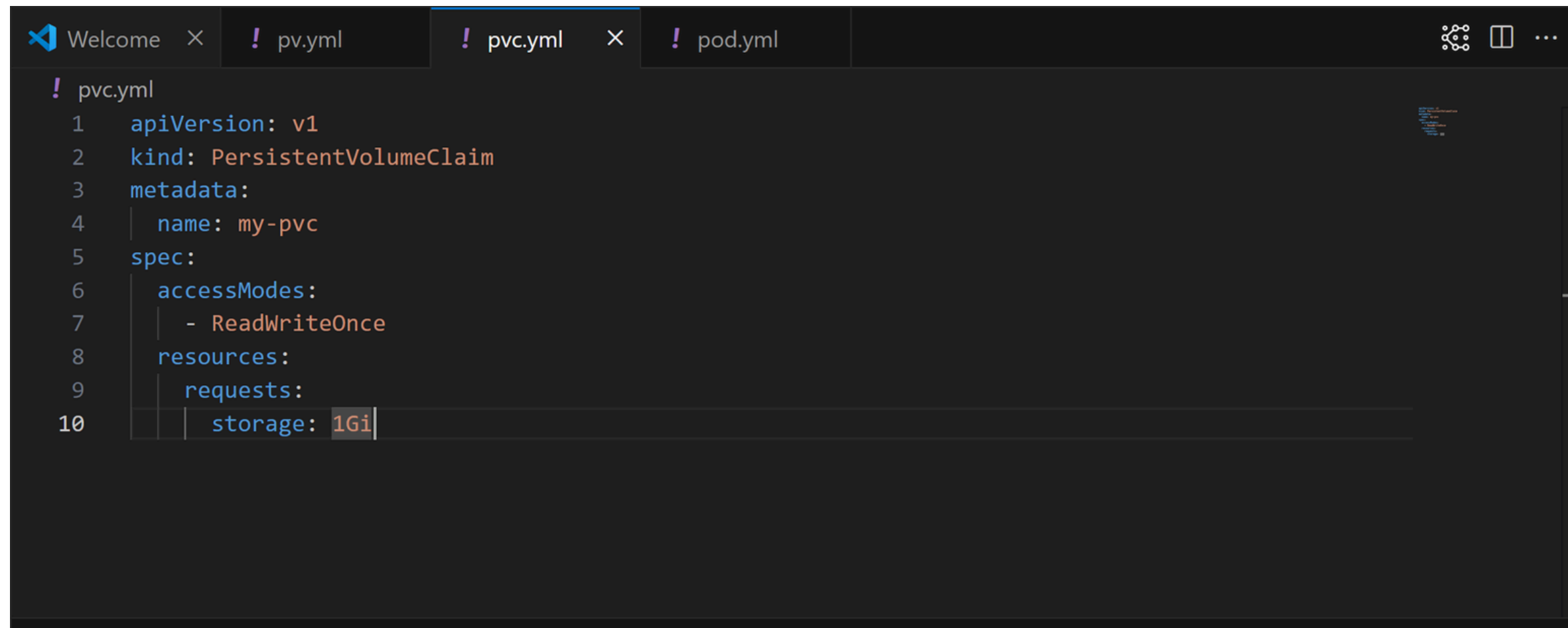
ReadWriteMany (RWX): Read-write by multiple nodes

A screenshot of a code editor window with a dark theme. The editor has several tabs at the top: 'Welcome', 'pv.yml' (active), 'pvc.yml', and 'pod.yml'. The 'pv.yml' tab is selected, and the file content is displayed. The code is a YAML configuration for a PersistentVolume. It includes fields for apiVersion, kind, metadata (name), spec (capacity, storage, accessModes, persistentVolumeReclaimPolicy, hostPath, and path). Line numbers 1 through 12 are visible on the left side of the editor.

```
! pv.yml
1  apiVersion: v1
2  kind: PersistentVolume
3  metadata:
4    name: my-pv
5  spec:
6    capacity:
7      storage: 1Gi
8    accessModes:
9      - ReadWriteOnce
10   persistentVolumeReclaimPolicy: Retain
11   hostPath:
12     path: "/mnt/data"
```

## Created pv.yml file

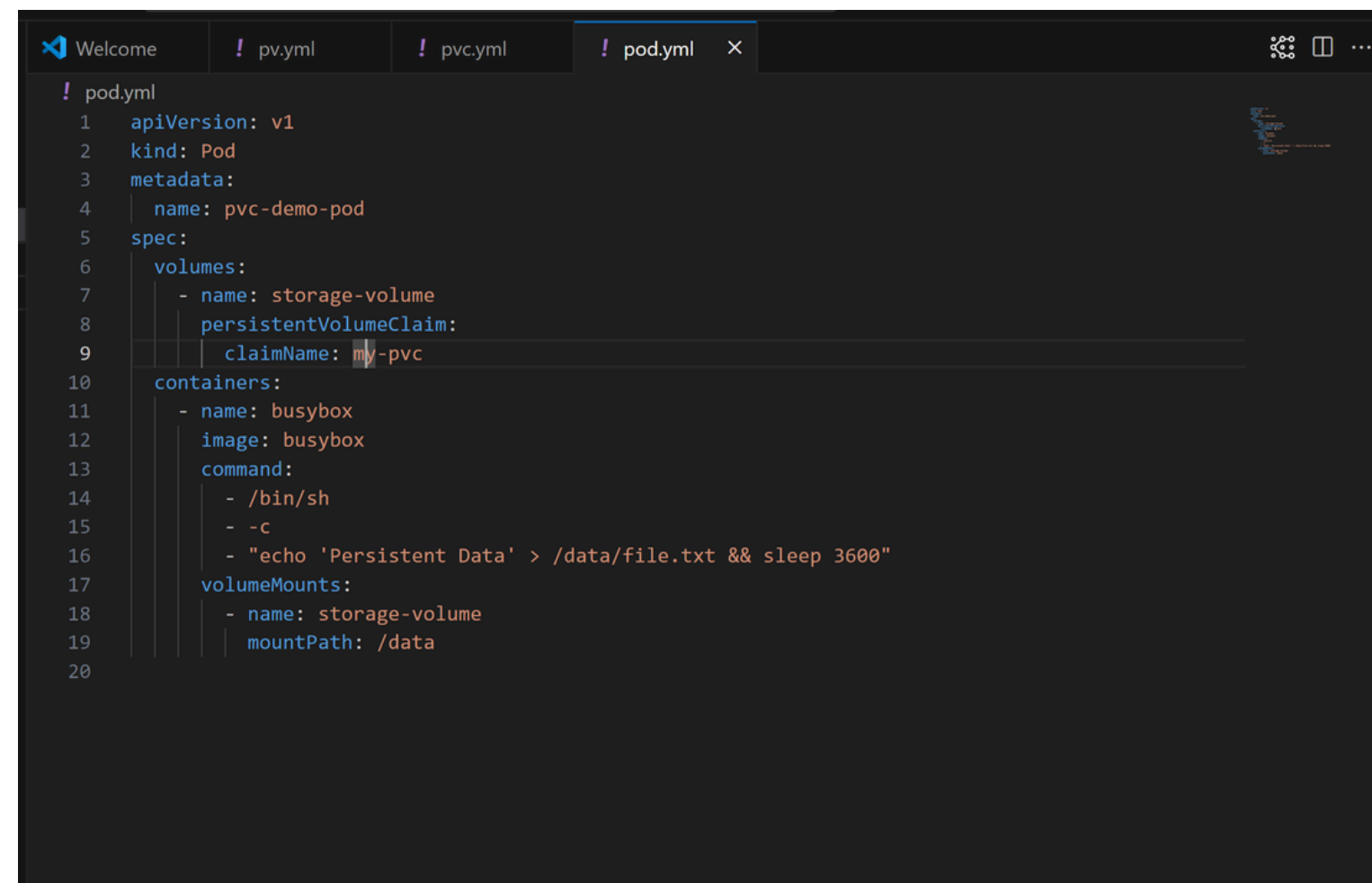
- First create pv.yml file
- Then set a name
- Then set a spec
- storage for 1gb
- set access mode RWO
- Then set Retain policy
- Then set host path
- Normaly pv file created by admin

A screenshot of a code editor interface with a dark theme. The top bar shows three open files: 'pv.yml', 'pvc.yml', and 'pod.yml', each with a warning icon. The 'pvc.yml' file is active and displays the following YAML content:

```
! pvc.yml
1  apiVersion: v1
2  kind: PersistentVolumeClaim
3  metadata:
4    name: my-pvc
5  spec:
6    accessModes:
7      - ReadWriteOnce
8    resources:
9      requests:
10     storage: 1Gi
```

Created pvc.yml file

- Second Create PVC.YML FILE
- Then set a kind of persistentvolumeclaim
- Then specify the name wich created in pv.yml
- Then spec set rwo mode
- Then storage set it for 1gb



```
! pod.yml
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: pvc-demo-pod
5  spec:
6    volumes:
7      - name: storage-volume
8        persistentVolumeClaim:
9          claimName: my-pvc
10   containers:
11     - name: busybox
12       image: busybox
13       command:
14         - /bin/sh
15         - -c
16         - "echo 'Persistent Data' > /data/file.txt && sleep 3600"
17       volumeMounts:
18         - name: storage-volume
19           mountPath: /data
20
```

## Finally Create POD.YML file

- Finally Create Pod.yml
- Then Set a Pod name
- Then spec create for volume
- Then Set from we need to claim set pv file name
- Then create container as busybox
- This means the container will write the "Persistent Data" to the file /data/file.txt, which is actually stored in the PersistentVolume linked to the my-pvc PersistentVolumeClaim.
- then mount on data path

```
PS C:\Users\Niree\volumekubernets> kubectl apply -f pv.yml
persistentvolume/my-pv created
PS C:\Users\Niree\volumekubernets> kubectl apply -f pvc.yml
PS C:\Users\Niree\volumekubernets> kubectl apply -f pod.yml
pod/pvc-demo-pod created
PS C:\Users\Niree\volumekubernets> kubectl get pods
```

Then Apply those three files and check pod is running or not

```
PS C:\Users\Niree\volumekubernets> kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
nginx-configmap-demo	1/1	Running	1 (39m ago)	10h
pvc-demo-pod	0/1	ContainerCreating	0	5s
nginx-configmap-demo	1/1	Running	1 (39m ago)	10h
pvc-demo-pod	0/1	ContainerCreating	0	5s
pvc-demo-pod	0/1	ContainerCreating	0	5s

- After That With Help this command **kubectl exec -it pvc-demo-pod -- cat /data/file.txt**
- Check persistent data file created or not
- Then it created means delete that pod and recreate that time also that file will be available