**KPLABS Course**

**Certified Kubernetes Application Developer**

**State Persistence**

## **ISSUED BY**

Zeal Vora

## **REPRESENTATIVE**

[instructors@kplabs.in](mailto:instructors@kplabs.in)

# 

## **Module 1: Docker Volumes**

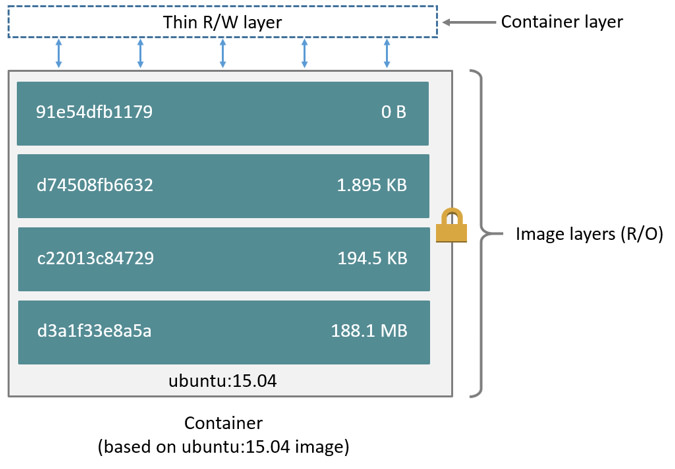
3.1 Challenges with files in Container Writable Layer

By default, all files created inside a container are stored on a writable container layer. This means that:

The data doesn’t persist when that container no longer exists, and it can be difficult to get the data out of the container if another process needs it.

Writing into a container’s writable layer requires a storage driver to manage the filesystem. The storage driver provides a union filesystem, using the Linux kernel.

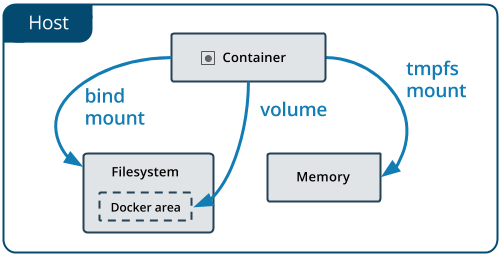
This extra abstraction reduces performance as compared to using data volumes, which write directly to the host filesystem.



3.2 Ideal Approach for Persistent Data

Docker has two options for containers to store files in the host machine, so that the files are persisted even after the container stops: volumes, and bind mounts.

If you’re running Docker on Linux you can also use a tmpfs mount.



3.3 Important Pointers to Remember:

A given volume can be mounted into multiple containers simultaneously.

When no running container is using a volume, the volume is still available to Docker and is not removed automatically.

When you mount a volume, it may be named or anonymous. Anonymous volumes are not given an explicit name when they are first mounted into a container, so Docker gives them a random name that is guaranteed to be unique within a given Docker host.

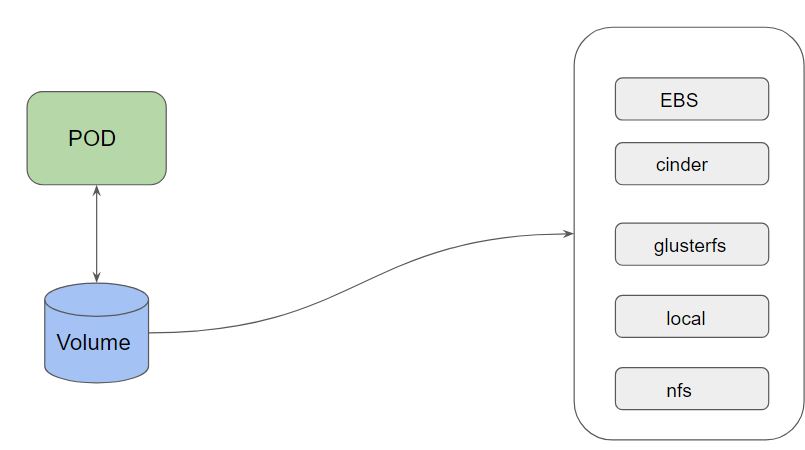
## 

## **Module 2: Volumes in Kubernetes**

On-disk files in a Container are ephemeral.

When there are multiple containers who want to share the same data, it becomes a challenge.

One of the benefits of Kubernetes is that it supports multiple types of volumes.



## **Module 3: PersistentVolume and PersistentVolumeClaim**

3.1 PersistentVolume (PV)

A PersistentVolume (PV) is a piece of storage in the cluster that has been provisioned by an administrator or dynamically provisioned using Storage Classes

Every Volume which is created can be of different type.

This can be taken care by the Storage Administrator / Ops Team



3.2 PersistentVolumeClaim (PVC)

A PersistentVolumeClaim is a request for the storage by a user.

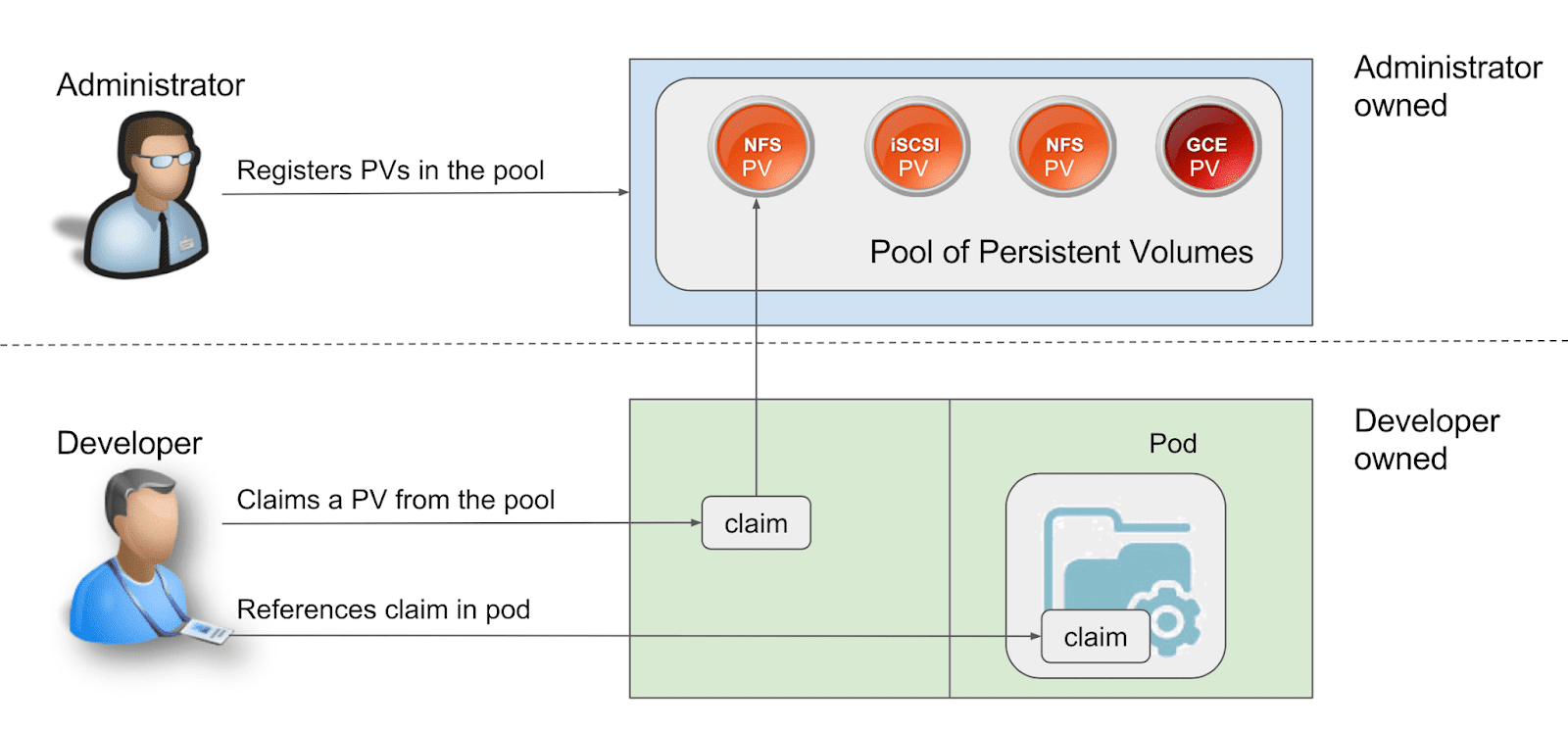
Within the claim, user need to specify the size of the volume along with access mode.

Developer:

I want a volume of size 10 GB which has speed of Fast for my pod.

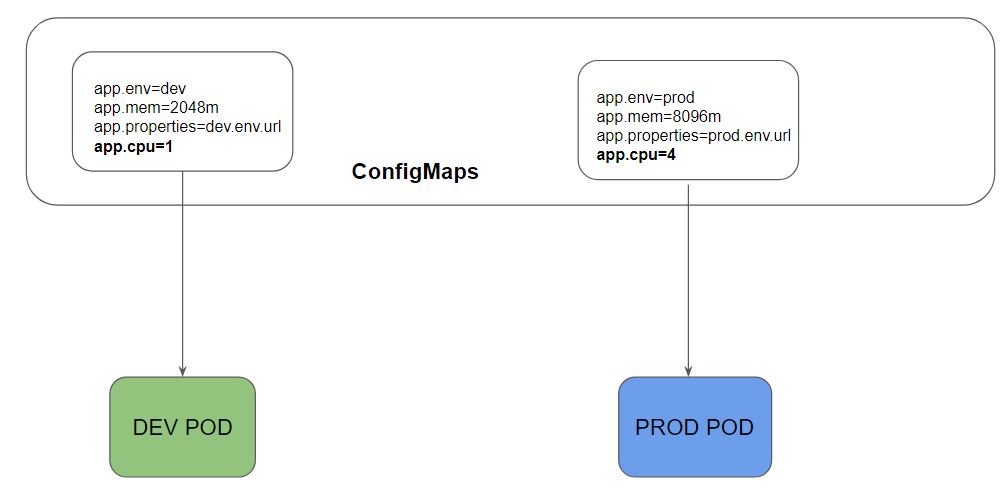
3.3 High-Level Working Steps:

* Storage Administrator takes care of creating PV.
* Developer can raise a “Claim” (I want a specific type of PV).
* Reference that claim within the PodSpec file.



## **Module 4: ConfigMaps**

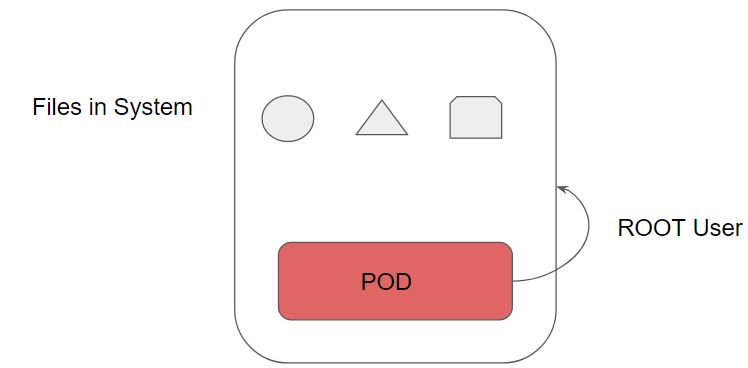
ConfigMaps allow you to decouple configuration artifacts from image content to keep containerized applications portable.



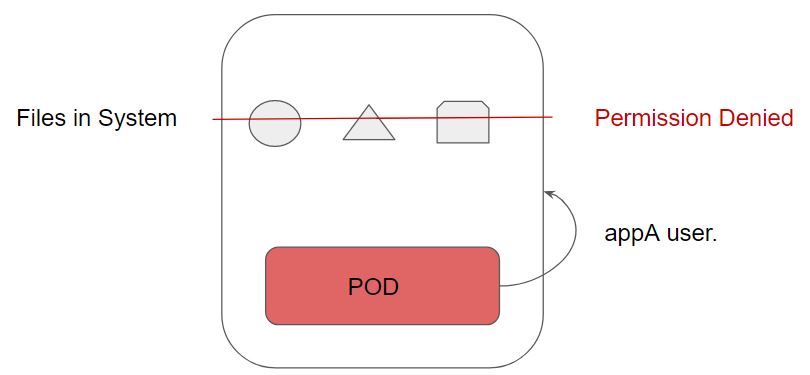
## **Module 5: Security Contexts**

When you run a container, it runs with the UID 0 (Administrative Privilege)

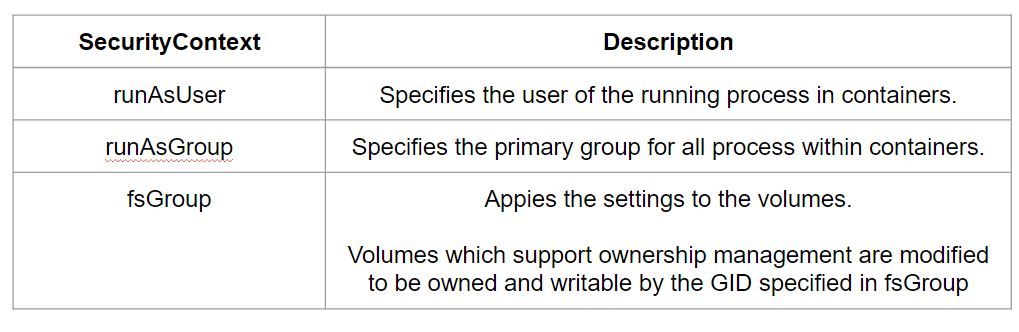
In-case of container breakouts, an attacker can get root privileges to your entire system.



We can run POD and container with limited privilege user instead of the ROOT user.



The following are the three important permissions:



## 

## 

## **Join Our Discord Community**

We invite you to join our Discord community, where you can interact with our support team for any course-based technical queries and connect with other students who are doing the same course.

Joining URL:

<http://kplabs.in/chat>

