Kubernetes Volumes

Students, write your response!

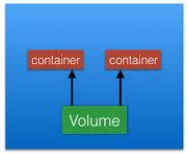
Pear Deck Interactive Slide
Do not remove this bar

CLARUSWAY©
WAY TO REINVENT YOURSELF

Table of Contents

1

Volumes

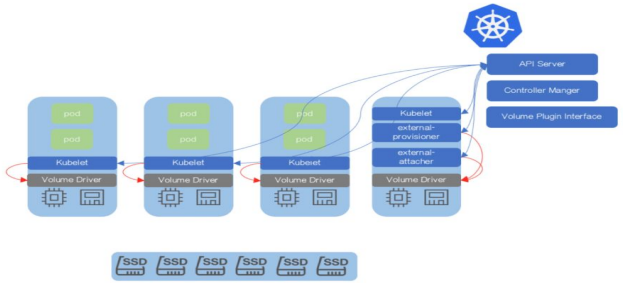


Pod

Volumes

- on-disk files in a Container are ephemeral. All data stored inside a container is deleted if the container crashes. When a Container crashes, kubelet will restart it, but the files will be lost which means that it will not have any of the old data.
- To overcome this problem, Kubernetes uses **Volumes**. A Volume is essentially a directory backed by a storage medium. The storage medium, content and access mode are determined by the Volume Type.

Volumes



A **volume** can be thought of as a directory which is accessible to the containers in a pod.

2

Volume Types

Volume Types

Kubernetes supports several types of Volumes.

- **emptyDir:** An emptyDir volume is first created when a Pod is assigned to a Node and exists as long as that Pod is running on that node. As the name says, it is initially empty. When a Pod is removed from a node for any reason, the data in the emptyDir is deleted forever.
- **hostPath:** A hostPath volume mounts a file or directory from the host node's filesystem into your Pod. If the Pod is terminated, the content of the Volume is still available on the host.

Volume Types

- **awsElasticBlockStore:** An awsElasticBlockStore volume mounts an Amazon Web Services (AWS) EBS Volume into your Pod.
- **azureDisk:** An azureDisk is used to mount a Microsoft Azure Data Disk into a Pod.
- **Secret:** A secret volume is used to pass sensitive information, such as passwords, to Pods.

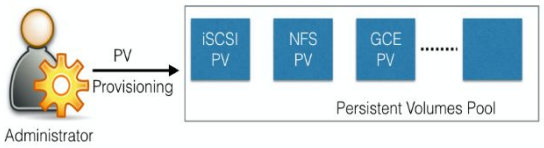
Volume Types

- **configMap:** The configMap resource provides a way to inject configuration data, or shell commands and arguments into a Pod.
- **persistentVolumeClaim:** A persistentVolumeClaim volume is used to mount a PersistentVolume into a Pod.

3 PersistentVolumes

PersistentVolumes

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



4 PersistentVolumeClaims

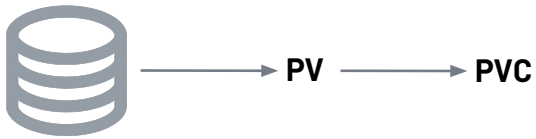
PersistentVolumeClaims

A **PersistentVolumeClaim (PVC)** is a request for storage by a user. Users request for PersistentVolume resources based on type, access mode, and size. There are three access modes:

- ReadWriteOnce (read-write by a single node)
- ReadOnlyMany (read-only by many nodes)
- ReadWriteMany (read-write by many nodes).
- Once a suitable PersistentVolume is found, it is bound to a PersistentVolumeClaim.

5 The interaction between PVs and PVCs

The interaction between PVs and PVCs



The interaction between PVs and PVCs

Provisioning

There are two ways PVs may be provisioned: **statically** or **dynamically**.

Static

A cluster administrator creates a number of PVs. They carry the details of the real storage, which is available for use by cluster users. They exist in the Kubernetes API and are available for consumption.

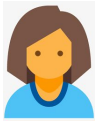
Dynamic

When none of the static PVs the administrator created match a user's PersistentVolumeClaim, the cluster may try to dynamically provision a volume specially for the PVC. This provisioning is based on StorageClasses.

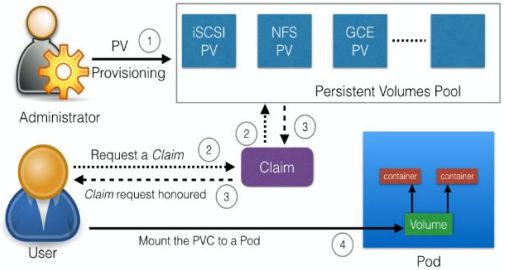
The interaction between PVs and PVCs

Static

Dynamic

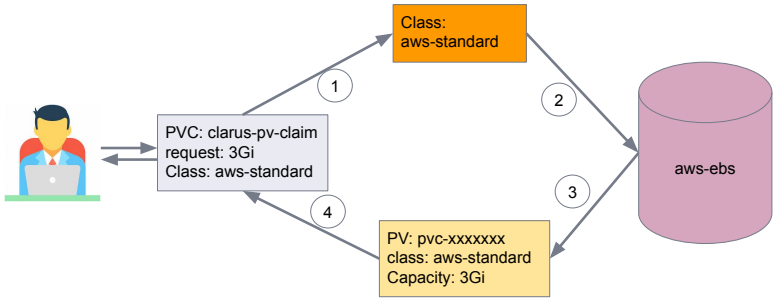


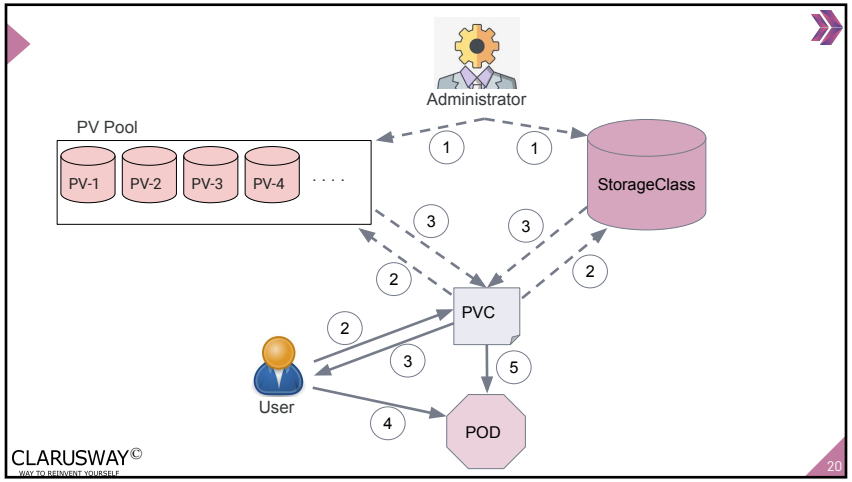
Static PV Provisioning



Dynamic PV Provisioning

A **StorageClass** provides a way for administrators to describe the "classes" of storage they offer.





THANKS!

Any questions?

You can find me at:

- james@clarusway.com

Students, write your response!