Volumes in kubernetes

Ref - https://kubernetes.io/docs/concepts/storage/volumes/

Volumes

In Kubernetes, a Volume is a directory, possibly with some data in it, which is accessible to a Container as part of its filesystem. Kubernetes supports a variety of volume types, such as EmptyDir, PersistentVolumeClaim, Secret, ConfigMap, and others.
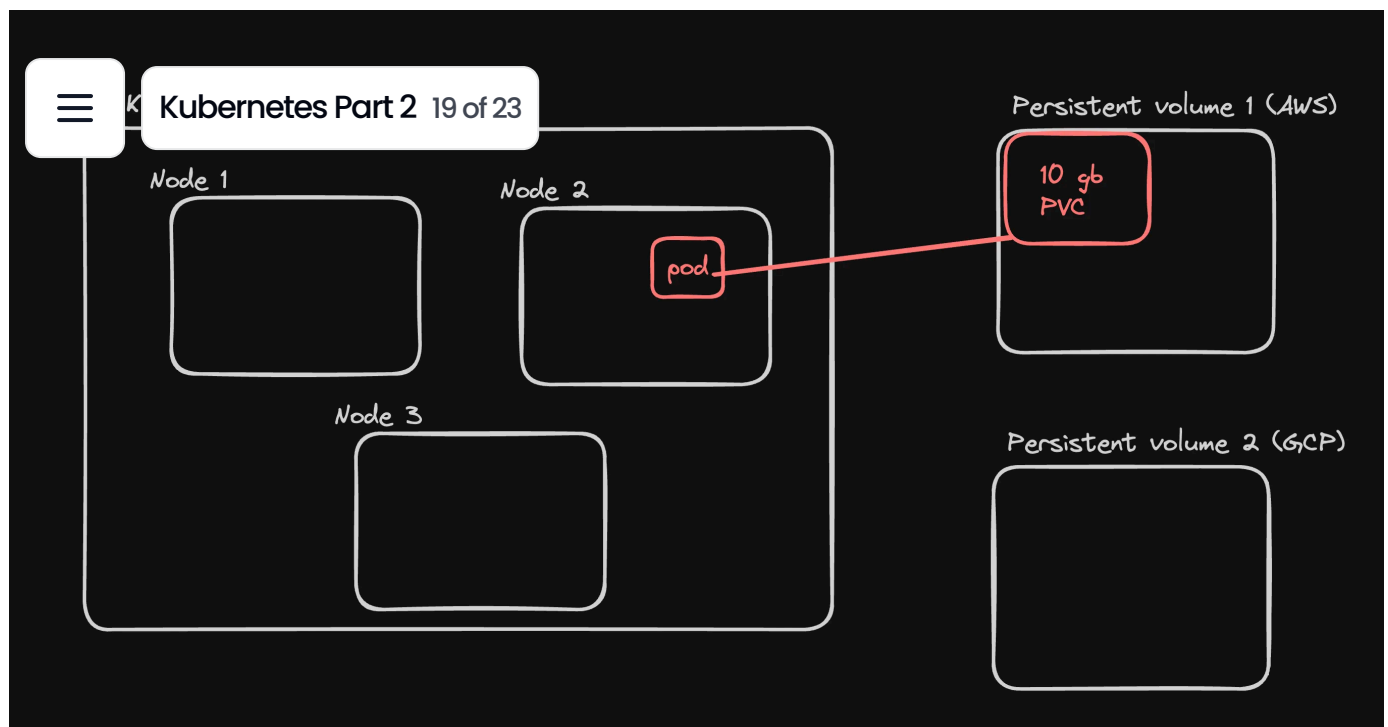
Why do you need volumes?

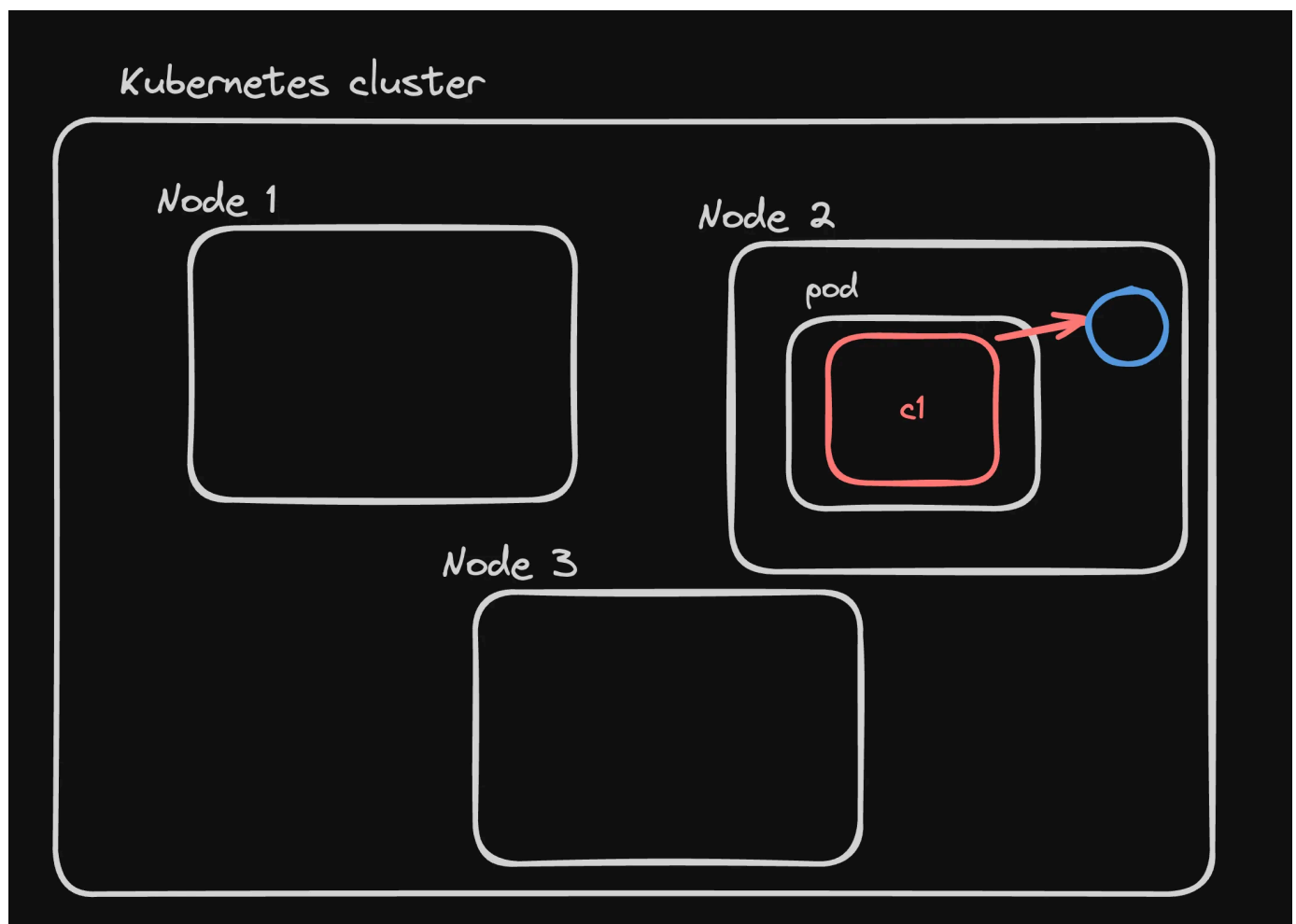- If two containers in the same `pod` want to share data/fs.
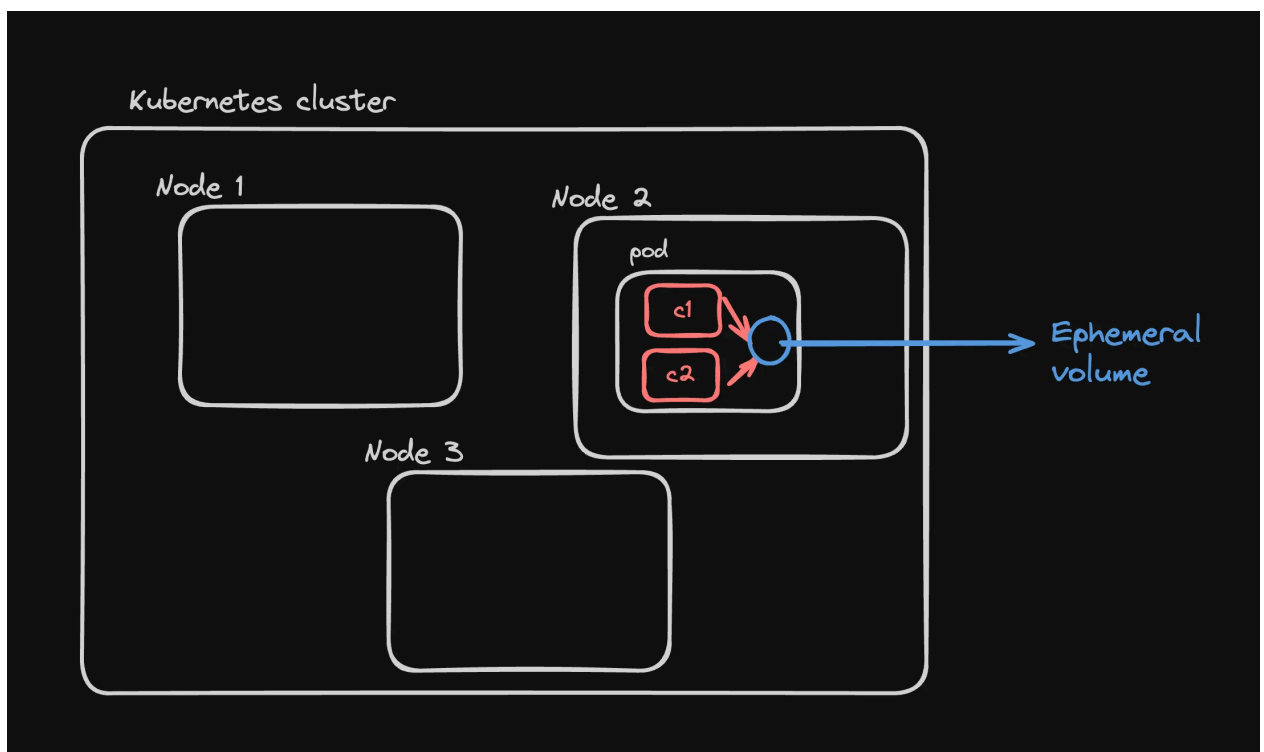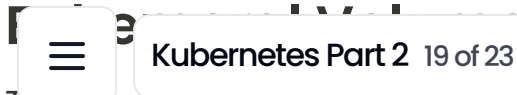

- If you want to create a database that persists data `even when` a container restarts (creating a DB)

- Your `pod` just needs extra space during execution (for caching lets say) but doesnt care if it persists or not.

Ephemeral Volume

Temporary volume that can be shared amongst various containers of a pod. When the pods dies, the volume dies with it.

For example -

1. ConfigMap

2. Secret

3. emptyDir

Persistent Volume

A Persistent Volume (PV) is a piece of storage in the cluster that has been provisioned by an administrator or dynamically provisioned using Storage Classes. It is a resource in the cluster just like a node is a cluster resource. PVs are volume plugins like Volumes but have a lifecycle independent of any individual Pod that uses the PV. This API object captures the details of the implementation of the storage, be that NFS, iSCSI, or a cloud-provider-specific storage system.

Persistent volume claim

A Persistent Volume Claim (PVC) is a request for storage by a user. It is similar to a Pod. Pods consume node resources and PVCs consume PV resources. Pods can request specific levels of resources (CPU and Memory). Claims can request specific size and access modes (e.g., can be mounted once read/write or many times read-only).