Configuring and Managing Kubernetes Storage and Scheduling

CONFIGURING AND MANAGING STORAGE IN KUBERNETES

Anthony E. Nocentino
ENTERPRISE ARCHITECT @ CENTINO SYSTEMS

@nocentino www.centinosystems.com

Course Overview

Configuring and Managing Storage in Kubernetes

Configuration as Data - Environment Variables, Secrets, and ConfigMaps

Managing and Controlling the Kubernetes Scheduler

Overview

Persistent Storage in Containers

Kubernetes Storage Objects

Storage Lifecycle

Using Storage in Kubernetes

Persistent Storage and Containers

Containers are ephemeral

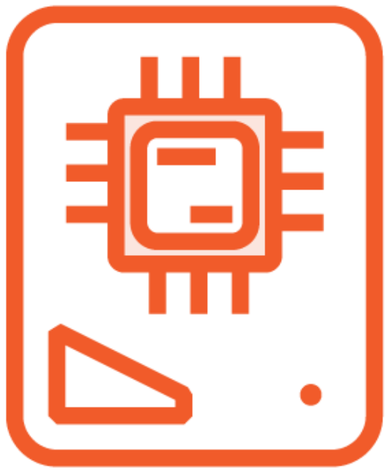
A container’s Writable Layer is deleted when the container is deleted

When a Pod is deleted, its container(s) is deleted from the Node

How can we persist data across a Pod’s lifecycle?

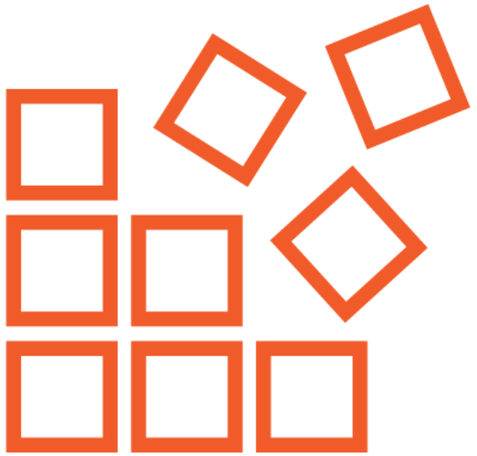
Storage API Objects in Kubernetes

Volume



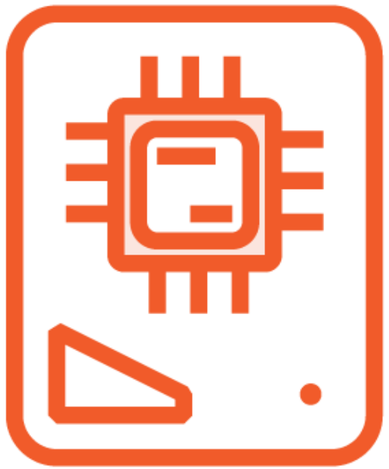
Persistent Volume

Persistent Volume Claim



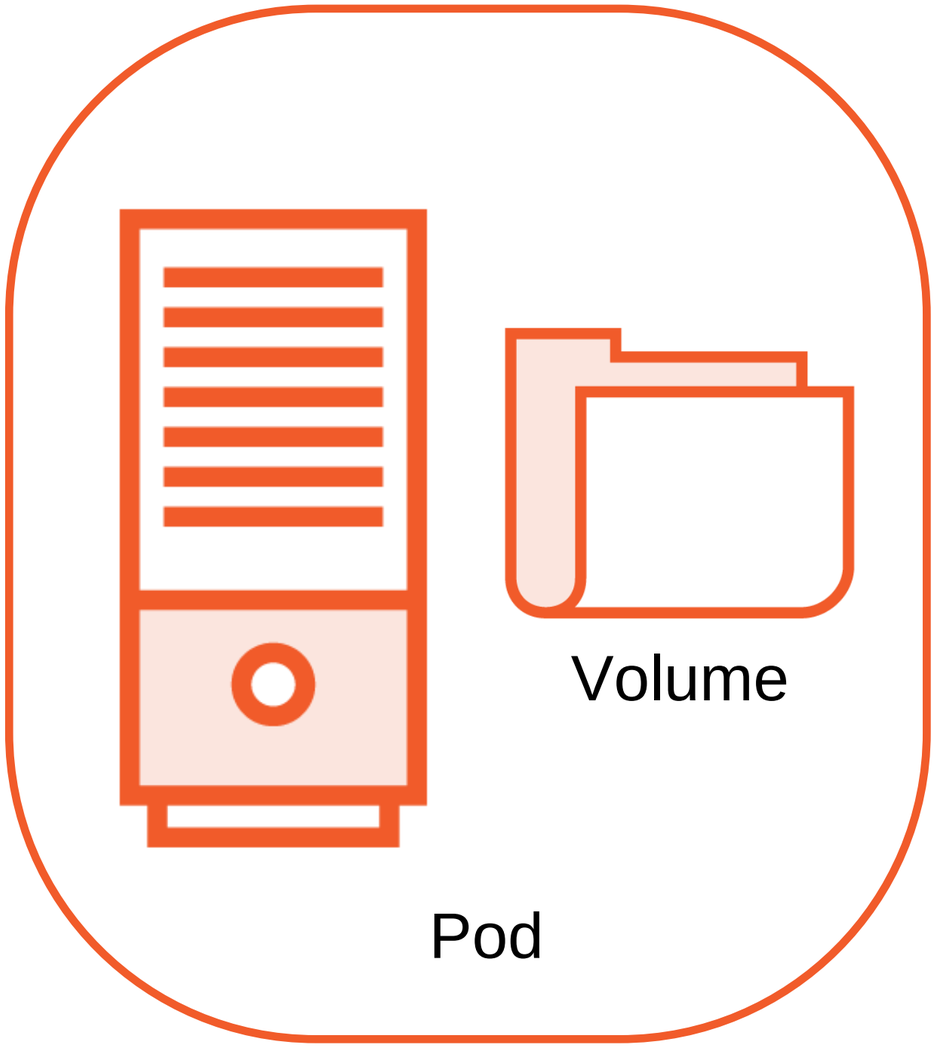
Storage Class

Storage in Kubernetes



Persistent Volume

Persistent Volume Claim



Volumes

Persistent storage deployed as part of the Pod spec

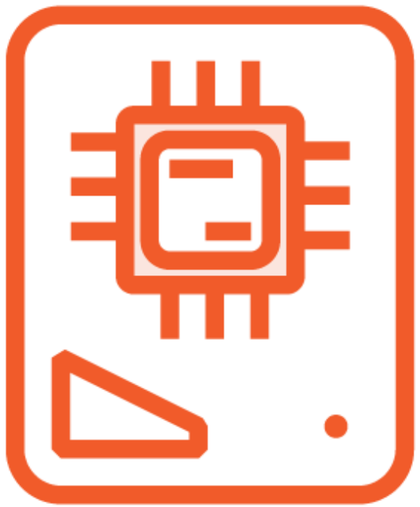
Implementation details for your storage

This can be challenging…

- Sharing code
- Same lifecycle as Pod

We can do better…

Persistent Volumes



Administrator defined storage in the Cluster

Implementation details for your storage

Lifecycle independent of the Pod

Managed by the Kubelet

- Maps the storage in the Node
- Exposes PV as a mount inside the container

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

Types of Persistent Volumes

Networked	Block	Cloud
NFS	Fibre Channel	awsElasticBlockStore
azureFile	iSCSI	azureDisk
		gcePersistentDisk

https://kubernetes.io/docs/concepts/storage/persistent-volumes/#types-of-persistent-volumes

Persistent Volumes Claims

A request for storage by a user

- Size
- Access Mode
- Storage Class

Enable portability of your application configurations

The Cluster will map a PVC to a PV

Access Modes

ReadWriteOnce (RWO)

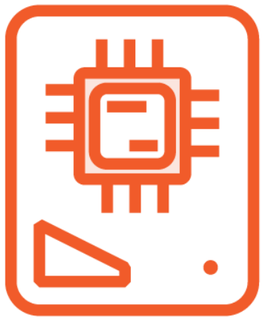
ReadWriteMany (RWX)

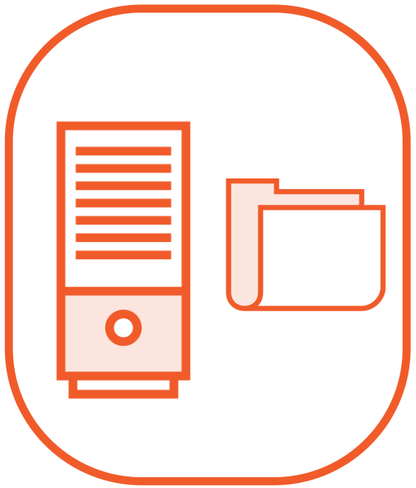
ReadOnlyMany (ROX)

Node level access, not Pod access

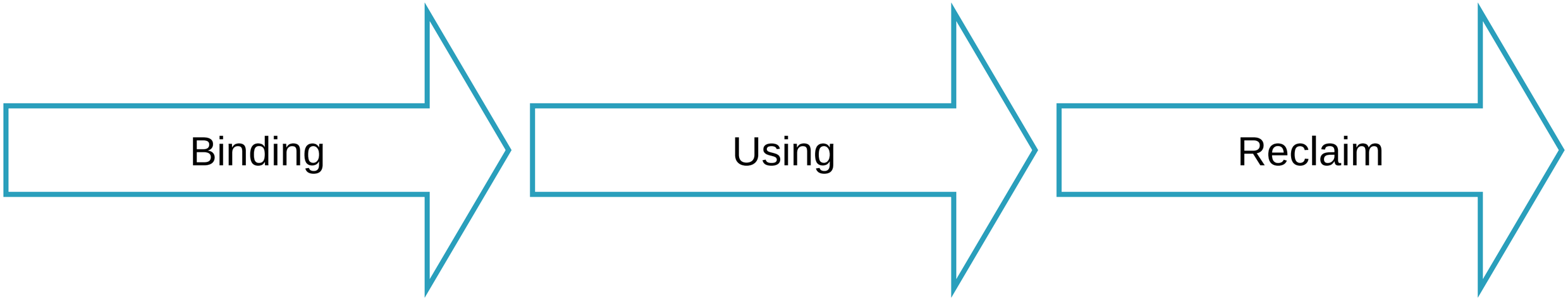
Static Provisioning Workflow

Create a `PersistentVolume`	Create a `PersistentVolumeClaim`	Define `Volume` in Pod Spec





Storage Lifecycle



PVC Created

Control Loop

Matches PVC->PV

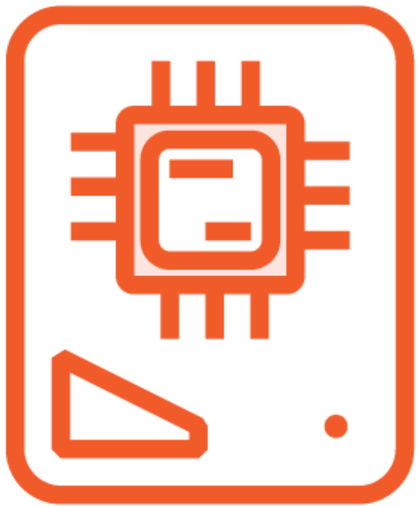
Pod's Lifetime

PVC Deleted

`Delete (default)`

`Retain`

Defining a Persistent Volume



type { nfs, fc, azureDisk, ... }

capacity

accessModes

persistentVolumeReclaimPolicy

Labels

Defining a Persistent Volume

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-nfs-data
spec:
  capacity:
    storage: 10Gi
  accessModes:
    - ReadWriteMany
 nfs:
    server: 172.16.94.5
    path: "/export/volumes/pod"
```

Defining a Persistent Volume Claim

`accessModes`

`resources`

`storageClassName`

`selector`

Defining a Persistent Volume Claim

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-nfs-data
spec:
  accessModes:
    - ReadWriteMany
 resources:
    requests:
      storage: 10Gi
```

Using Persistent Volumes in Pods

```
...
spec:
 volumes:
  - name: webcontent
    persistentVolumeClaim:
      claimName: pvc-nfs-data
 containers:
 - name: nginx
   ...
   volumeMounts:
   - name: webcontent
     mountPath: "/usr/share/nginx/html/web-app"
```

mountPath

volumeMounts

volumes

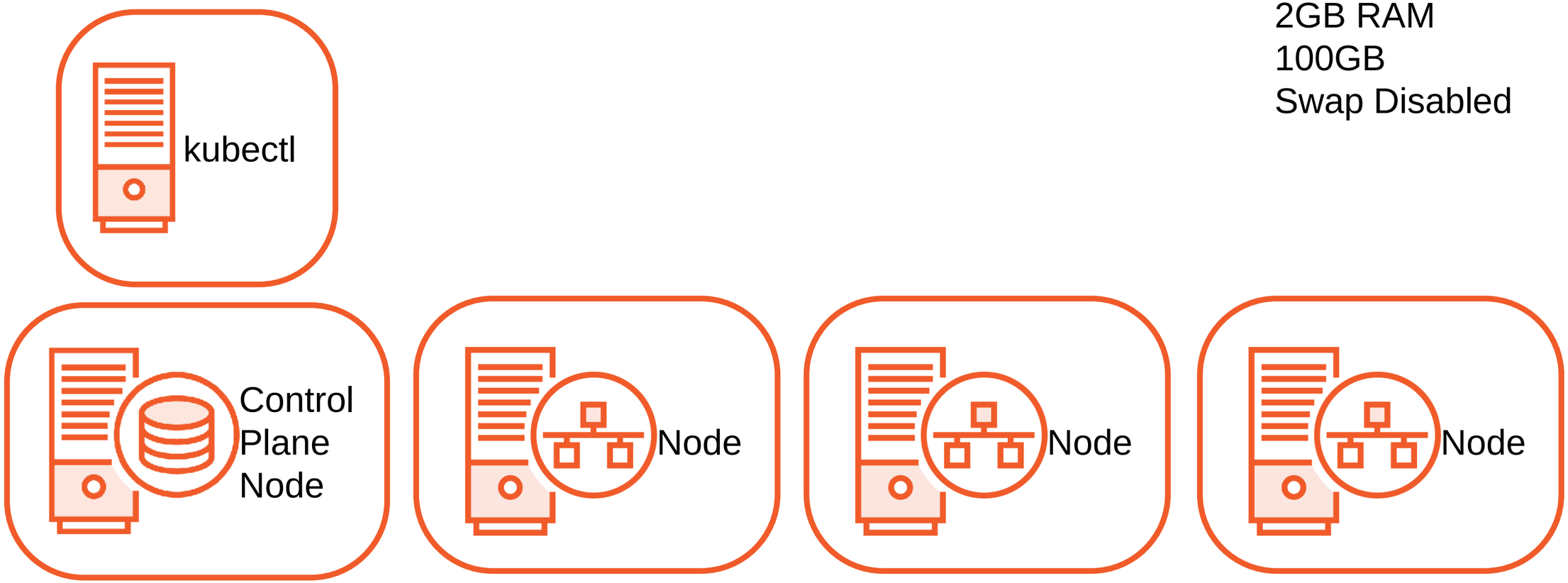
PersistentVolumeClaim

PersistentVolume

Hostnames set
Host file on each

Lab Environment

Ubuntu 18.0.4
VMware Fusion VMs
2vCPU
2GB RAM
100GB
Swap Disabled



c1-cp1
172.16.94.10

c1-node1
172.16.94.11

c1-node2
172.16.94.12

c1-node3
172.16.94.13

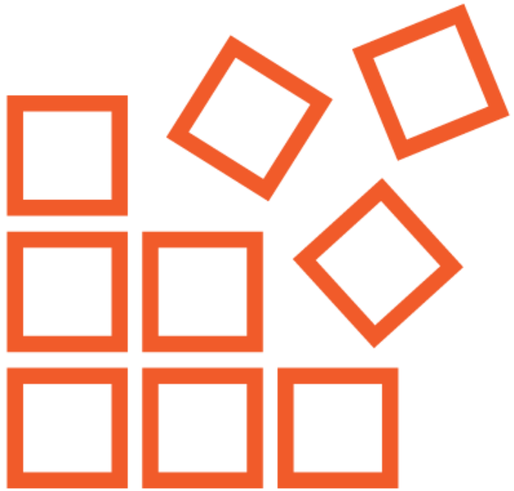
Demo

Storage Server Overview - NFS

Static Provisioning Persistent Volumes

Storage Lifecycle and Reclaim Policy

Storage Class



Define tiers/classes of storage

Enables Dynamic Provisioning

Define infrastructure specific parameters

Reclaim Policy

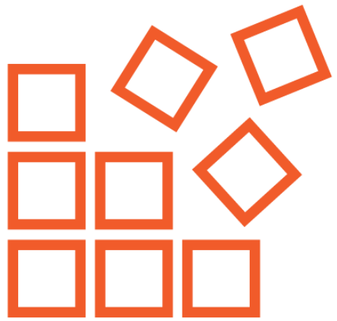
Dynamic Provisioning Workflow

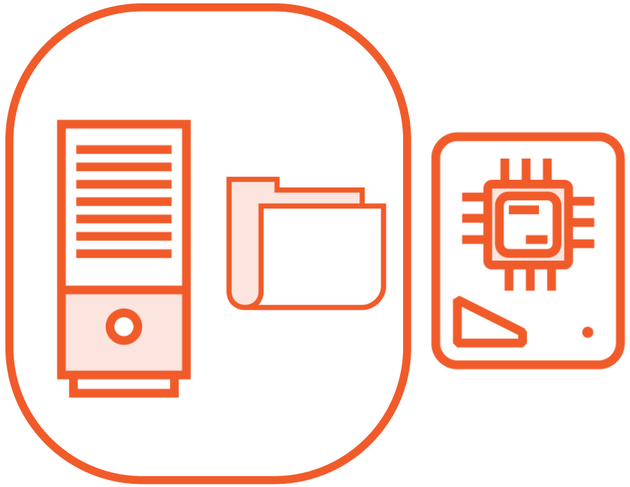
Create a `StorageClass`

Create a `PersistentVolumeClaim`

Define `Volume` in Pod Spec

Creates a `PersistentVolume`





Defining a StorageClass

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: managed-premium
parameters:
  kind: Managed
  storageaccounttype: Premium_LRS
provisioner: kubernetes.io/azure-disk
```

Dynamic Provisioning

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-azure-managed
spec:
  accessModes:
  - ReadWriteOnce
  storageClassName: managed-premium
  resources:
    requests:
      storage: 10Gi
```

Demo

Dynamic Provisioning in the Cloud

Defining a custom **`StorageClass`**

Review

Persistent Storage in Containers

Kubernetes Storage Objects

Storage Lifecycle

Using Storage in Kubernetes

What’s Next!

Configuration as Data - Environment Variables, Secrets, and ConfigMaps