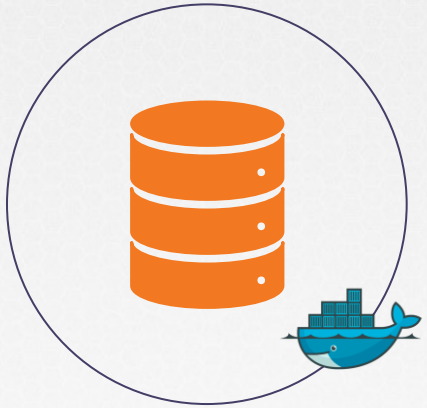
KodeKloud

DOCKER STORAGE



DOCKER STORAGE

STORAGE DRIVERS

VOLUME DRIVERS

DOCKER VOLUMES

STORAGE DRIVERS

AUFS | ZFS | BTRFS | DEVICE MAPPER | OVERLAY

VOLUME DRIVERS

Local | Azure File Storage | Convoy | DigitalOcean Block Storage | Flocker | gce-docker | GlusterFS | NetApp | RexRay | Portworx | VMware vSphere Storage

https://docs.docker.com/engine/extend/legacy_plugins

VOLUME DRIVERS

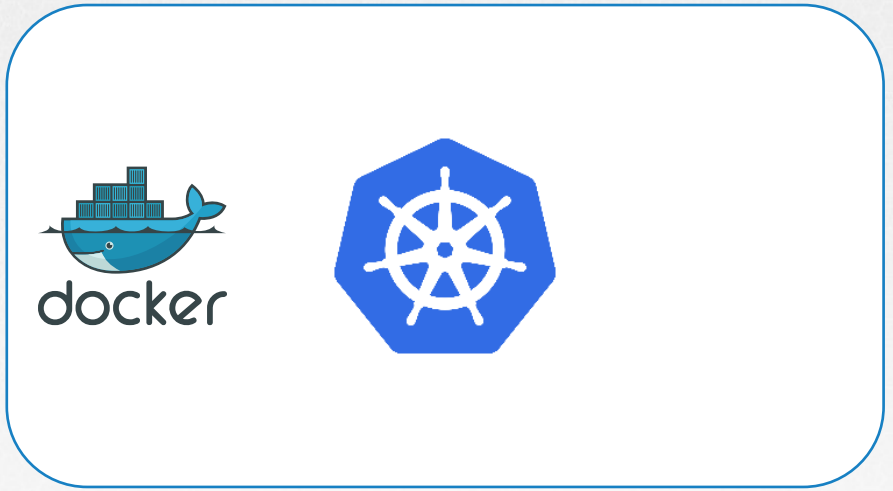
```
docker run -it \
    --name mysql
    --volume-driver rexray/ebs
    --mount src=ebs-vol,target=/var/lib/mysql
    mysql
```

https://github.com/rexray/rexray

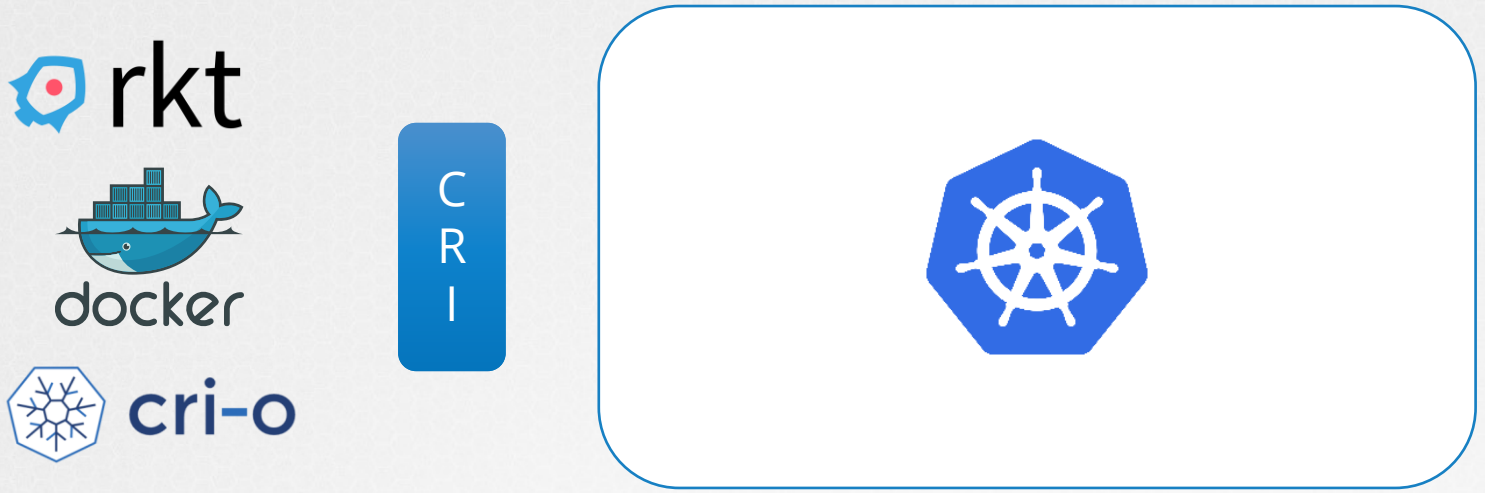
CONTAINER STORAGE INTERFACE


rkt

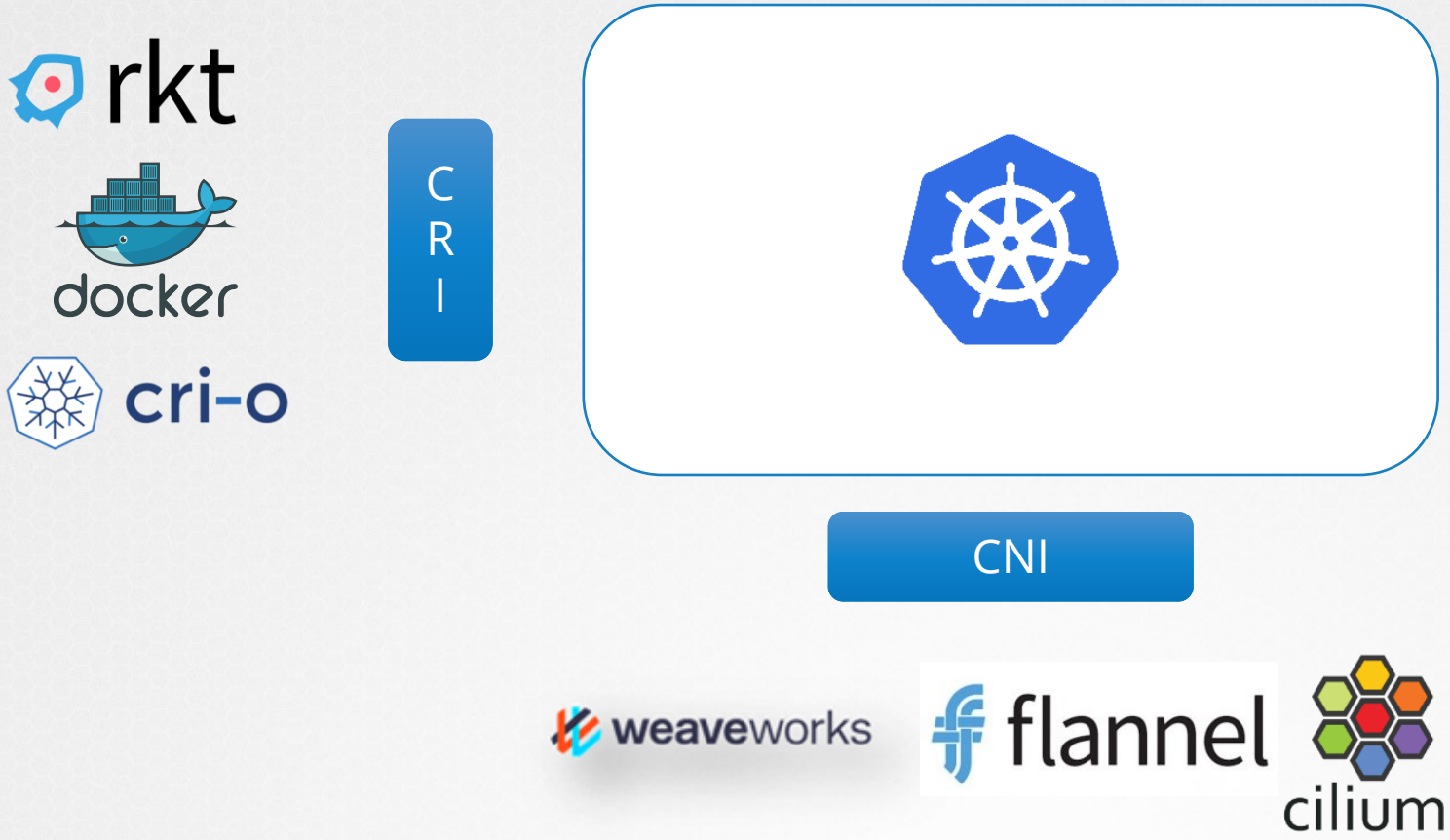

cri-o


docker

Container Runtime Interface

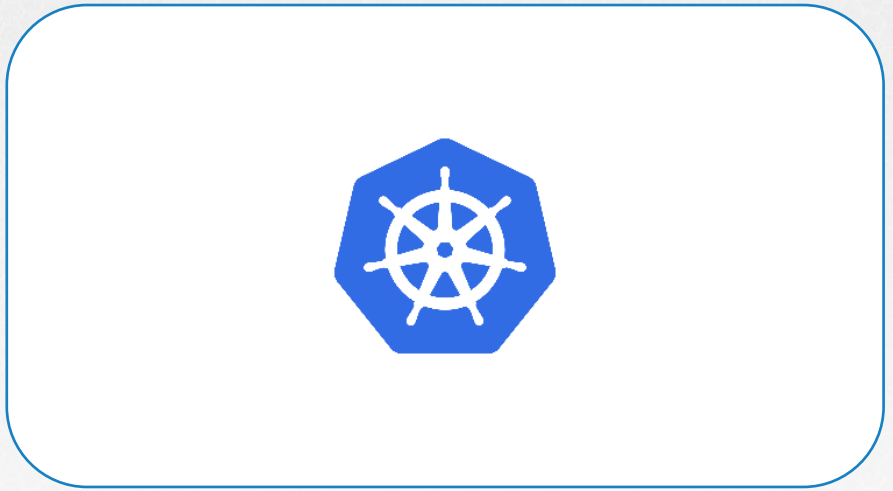


Container Network Interface



Container Storage Interface

CRI

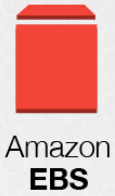


CSI

Amazon
EBS

CNI

CONTAINER
STORAGE
INTERFACE



RPC

CreateVolume

DeleteVolume

ControllerPublishVolume

- ✓ SHOULD call to provision a new volum
- ✓ SHOULD call to delete a volume
- ✓ SHOULD Call to place a workload that uses the volume onto a node.

- ✓ SHOULD provision a new volume on the storage
- ✓ SHOULD decommission a volume
- ✓ SHOULD make the volume available on a node

https://github.com/container-storage-interface/s

STORAGE CLASSES

PV and PVCs

pv-definition.yaml

```yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-vol1
spec:
  accessModes:
    - ReadWriteOnce
  capacity:
    storage: 500Mi
  gcePersistentDisk:
    pdName: pd-disk
    fsType: ext4
```

pvc-definition.yaml

```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: myclaim
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 500Mi
```

pod-definition.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: random-number-generator
spec:
  containers:
  - image: alpine
    name: alpine
    command: ["/bin/sh","-c"]
    args: ["shuf -i 0-100 -n 1 >>
/op
    volumeMounts:
    - mountPath: /opt
      name: data-volume
  volumes:
  - name: data-volume
    persistentVolumeClaim:
      claimName: myclaim
```

Static Provisioning

```
gcloud beta compute disks create \
    --size 1GB
    --region us-east1
    pd-disk
```

pv-definition.yaml

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-vol1
spec:
  accessModes:
    - ReadWriteOnce
  capacity:
    storage: 500Mi
  gcePersistentDisk:
    pdName: pd-disk
    fsType: ext4
```

PV

Dynamic Provisioning

pv-definition.yaml

```yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-vol1
spec:
  accessModes:
    - ReadWriteOnce
  capacity:
    storage: 500Mi
  gcePersistentDisk:
    pdName: pd-disk
    fsType: ext4
```

PV

sc-definition.yaml

```yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: google-storage
provisioner: kubernetes.io/gce-pd
```

SC

Dynamic Provisioning

pv-definition.yaml

```yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-vol1
spec:
  accessModes:
    - ReadWriteOnce
  capacity:
    storage: 500Mi
  gcePersistentDisk:
    pdName: pd-disk
    fsType: ext4
```

PV

pvc-definition.yaml

```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: myclaim
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 500Mi
```

pod-definition.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: random-number-generator
spec:
  containers:
  - image: alpine
    name: alpine
    command: ["/bin/sh","-c"]
    args: ["shuf -i 0-100 -n 1 >>
/op
    volumeMounts:
    - mountPath: /opt
      name: data-volume
  volumes:
  - name: data-volume
    persistentVolumeClaim:
      claimName: myclaim
```

sc-definition.yaml

```yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: google-storage
provisioner: kubernetes.io/gce-pd
```

SC

Dynamic Provisioning

sc-definition.yaml

```yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: google-storage
provisioner: kubernetes.io/gce-pd
```

pvc-definition.yaml

```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: myclaim
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: google-storage
  resources:
    requests:
      storage: 500Mi
```

pod-definition.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: random-number-generator
spec:
  containers:
  - image: alpine
    name: alpine
    command: ["/bin/sh","-c"]
    args: ["shuf -i 0-100 -n 1 >>
/op
    volumeMounts:
    - mountPath: /opt
      name: data-volume
  volumes:
  - name: data-volume
    persistentVolumeClaim:
      claimName: myclaim
```

SC

PV

PVC

Storage Class

sc-definition.yaml

```yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
   name: google-storage
provisioner: kubernetes.io/gce-pd
parameters:
  type: pd-standard [ pd-standard | pd-ssd ]
  replication-type: none [ none | regional-pd ]
```

Volume Plugin
AWSElasticBlockStore
AzureFile
AzureDisk
CephFS
Cinder
FC
FlexVolume
Flocker
GCEPersistentDisk
Glusterfs
iSCSI
Quobyte
NFS
RBD
VsphereVolume
PortworxVolume
ScaleIO
StorageOS
Local

Storage Class

sc-definition.yaml

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: silver
provisioner: kubernetes.io/gce-pd
parameters:
  type: pd-standard
  replication-type: none
```

sc-gold-definition.yaml

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: gold
provisioner: kubernetes.io/gce-pd
parameters:
  type: pd-ssd
  replication-type: none
```

sc-platinum-definition.y

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: platinum
provisioner: kubernetes.io/gce-p
parameters:
  type: pd-ssd
  replication-type: regional-pd
```

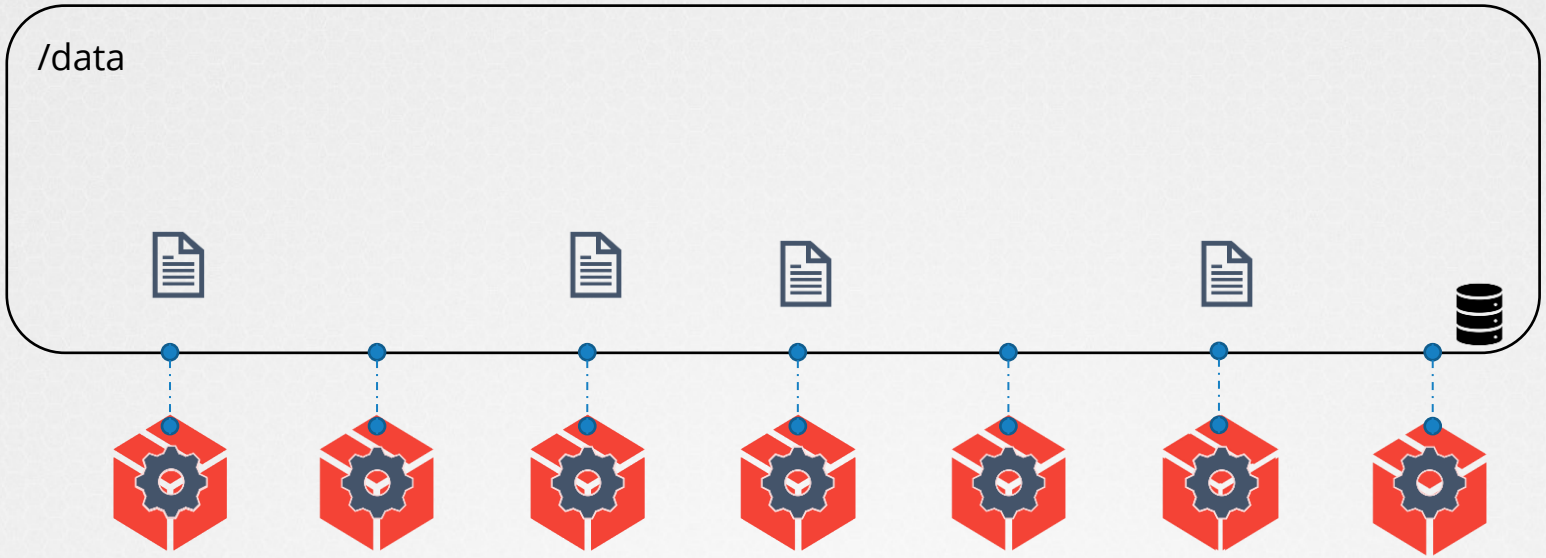
Silver
SC

Gold
SC

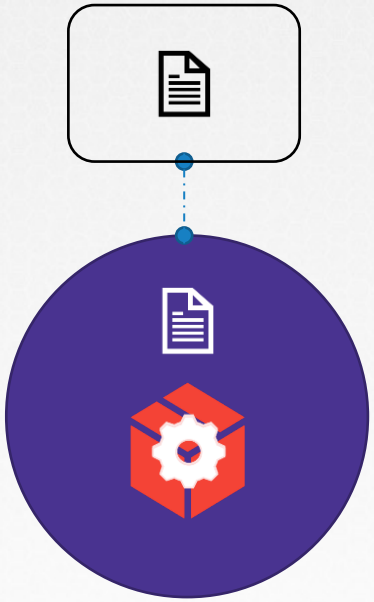
Platinu
m
SC

Volumes

Volume



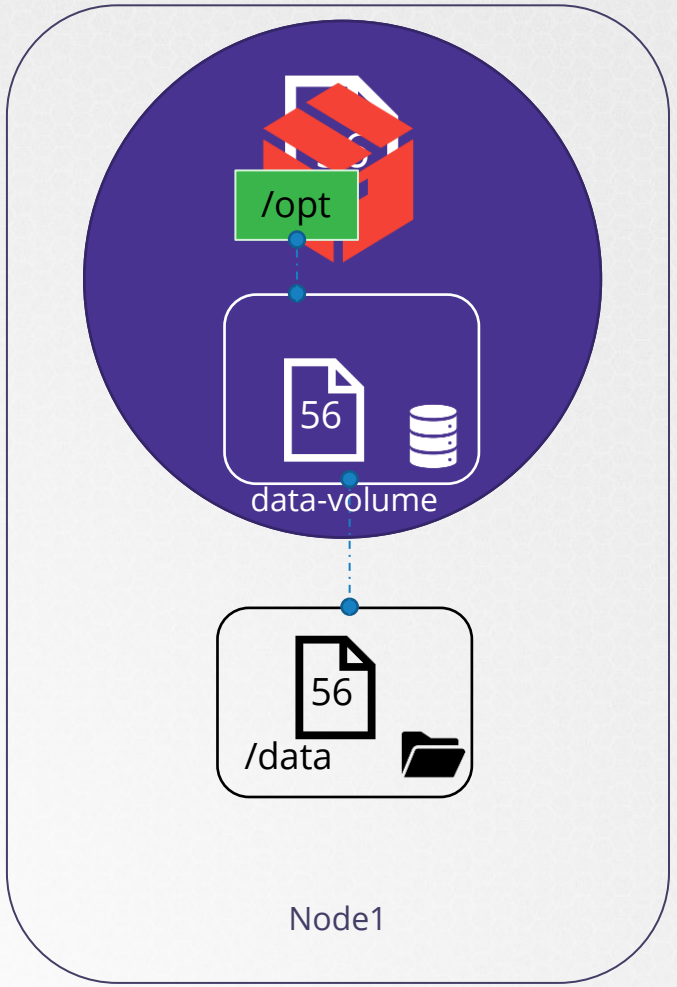
Volumes



Volumes & Mounts

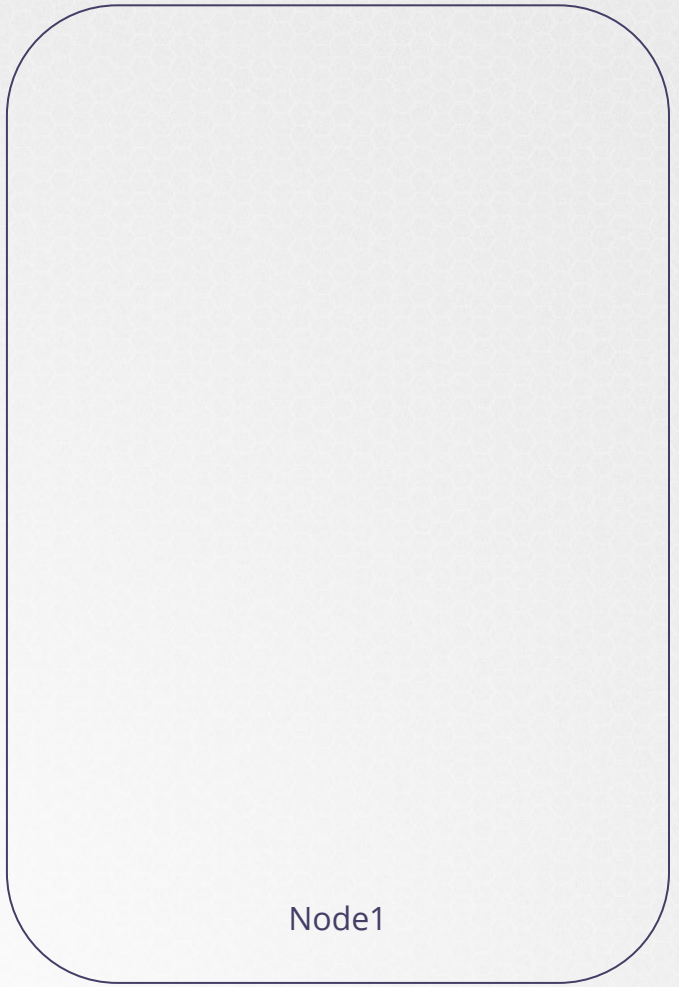
```yaml
apiVersion: v1
kind: Pod
metadata:
  name: random-number-generator
spec:
  containers:
  - image: alpine
    name: alpine
    command: ["/bin/sh","-c"]
    args: ["shuf -i 0-100 -n 1 >> /opt/number.out;"]
    volumeMounts:
    - mountPath: /opt
      name: data-volume

  volumes:
  - name: data-volume
    hostPath:
       path: /data
       type: Directory
```

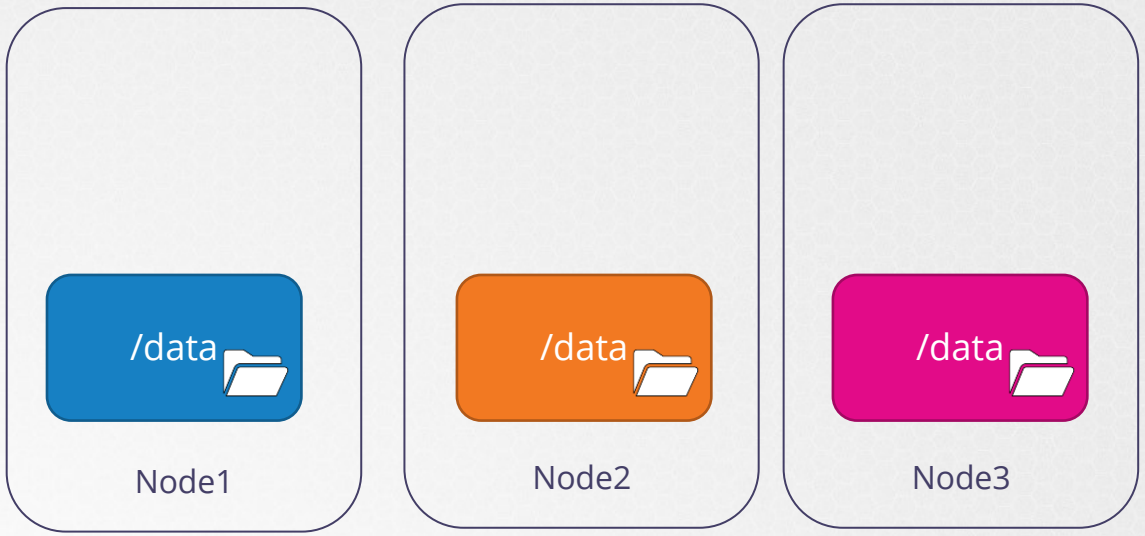


```yaml
volumes:
- name: data-volume
  hostPath:
    path: /data
    type: Directory
```

data-volume



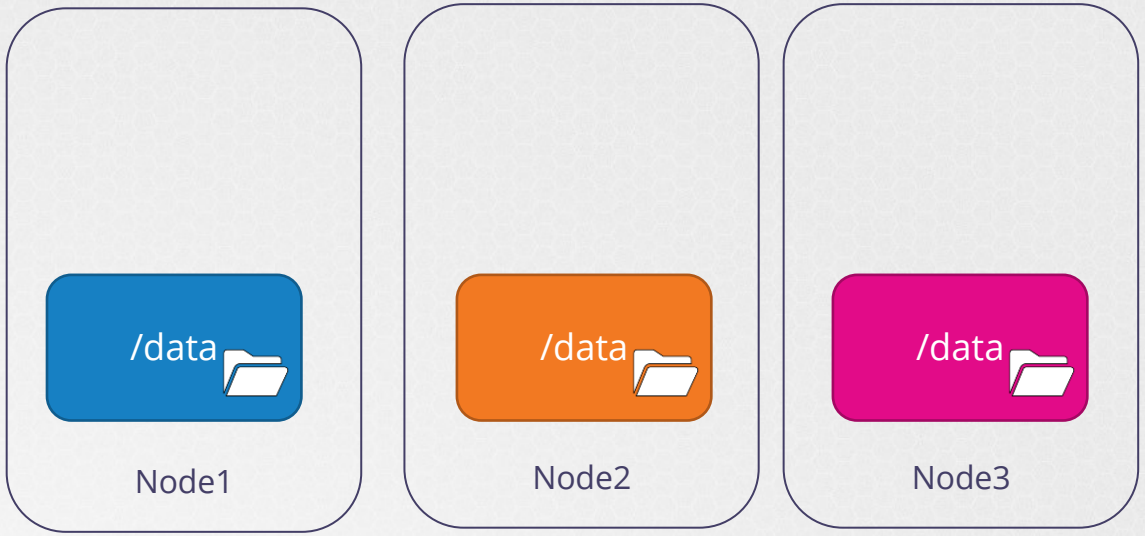
```yaml
volumes:
- name: data-volume
  hostPath:
    path: /data
    type: Directory
```



Volume Types

```
volumes:
- name: data-volume
  hostPath:
    path: /data
    type: Directory
```

data-volume



NFS

GlusterFS

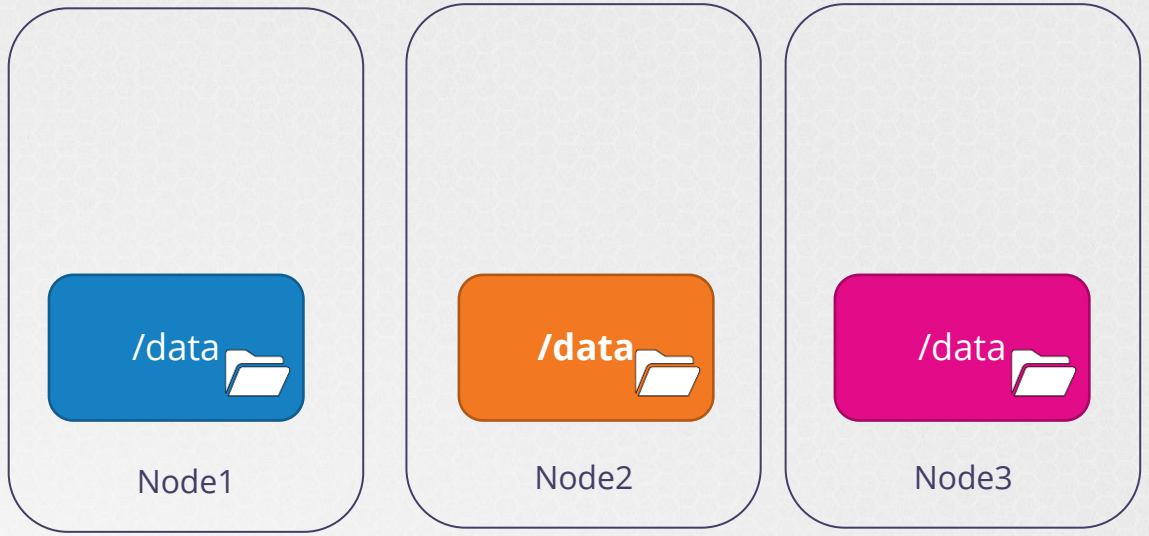
Flocker
by ClusterHQ

ceph

SCALEIO

amazon
web services

Volume Types



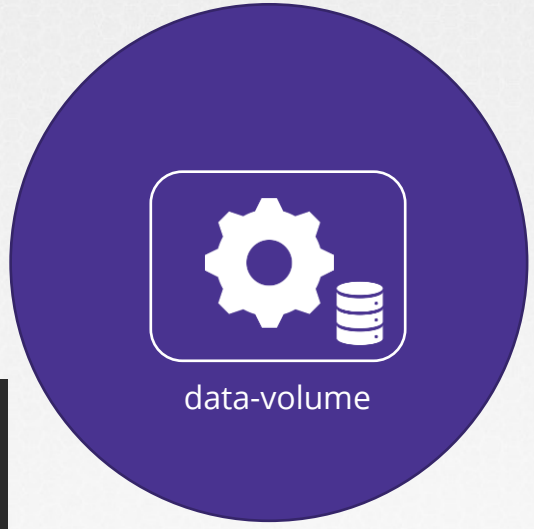

```
volumes:
- name: data-volume
  awsElasticBlockStore:
    volumeID: <volume-id>
    fsType: ext4
```

Course Objectives

- Core Concepts
- Configuration
- Multi-Container Pods
- Observability
- Pod Design
- Services & Networking
- State Persistence
  - Persistent Volumes
  - Persistent Volume Claims

Persistent Volumes

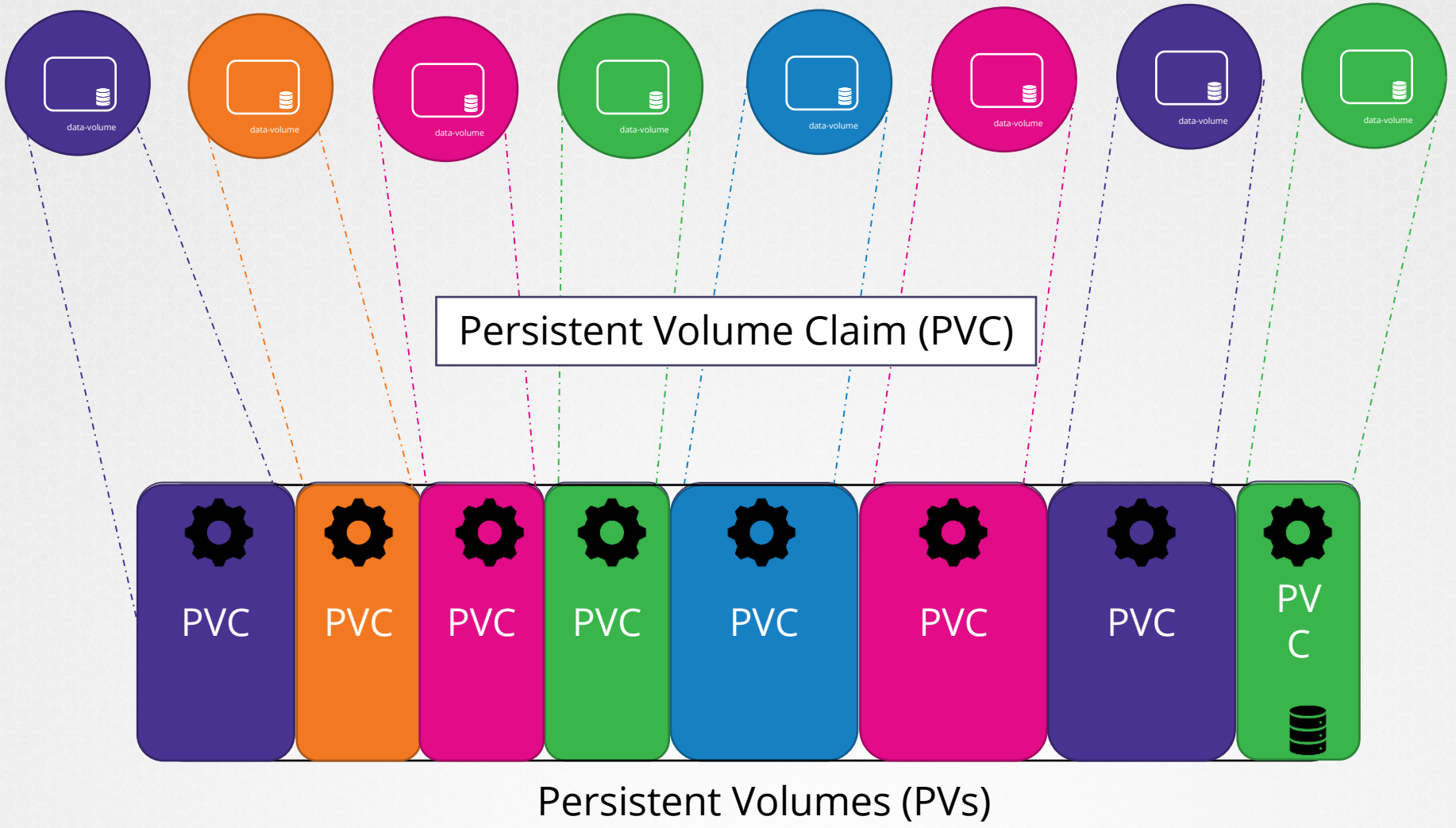
```
volumes:
- name: data-volume
  awsElasticBlockStore:
    volumeID: <volume-id>
    fsType: ext4
```




data-volume
data-volume
data-volume
data-volume
data-volume
data-volume
data-volume
data-volume


data-volume
data-volume
data-volume
data-volume
data-volume
data-volume
data-volume
data-volume

Persistent Volume



Persistent Volume

pv-definition.yaml

```yaml
apiVersion: v1
kind: PersistentVolume
metadata:
   name: pv-vol1
spec:
  accessModes:
      - ReadWriteOnce
  capacity:
      storage: 1Gi
  awsElasticBlockStore:
    volumeID: <volume-id>
    fsType: ext4
```

ReadOnlyMany
ReadWriteOnce
ReadWriteMany

Persistent Volume (PV)

```
kubectl create -f pv-definition.yaml
```

```
kubectl get persistentvolume
```

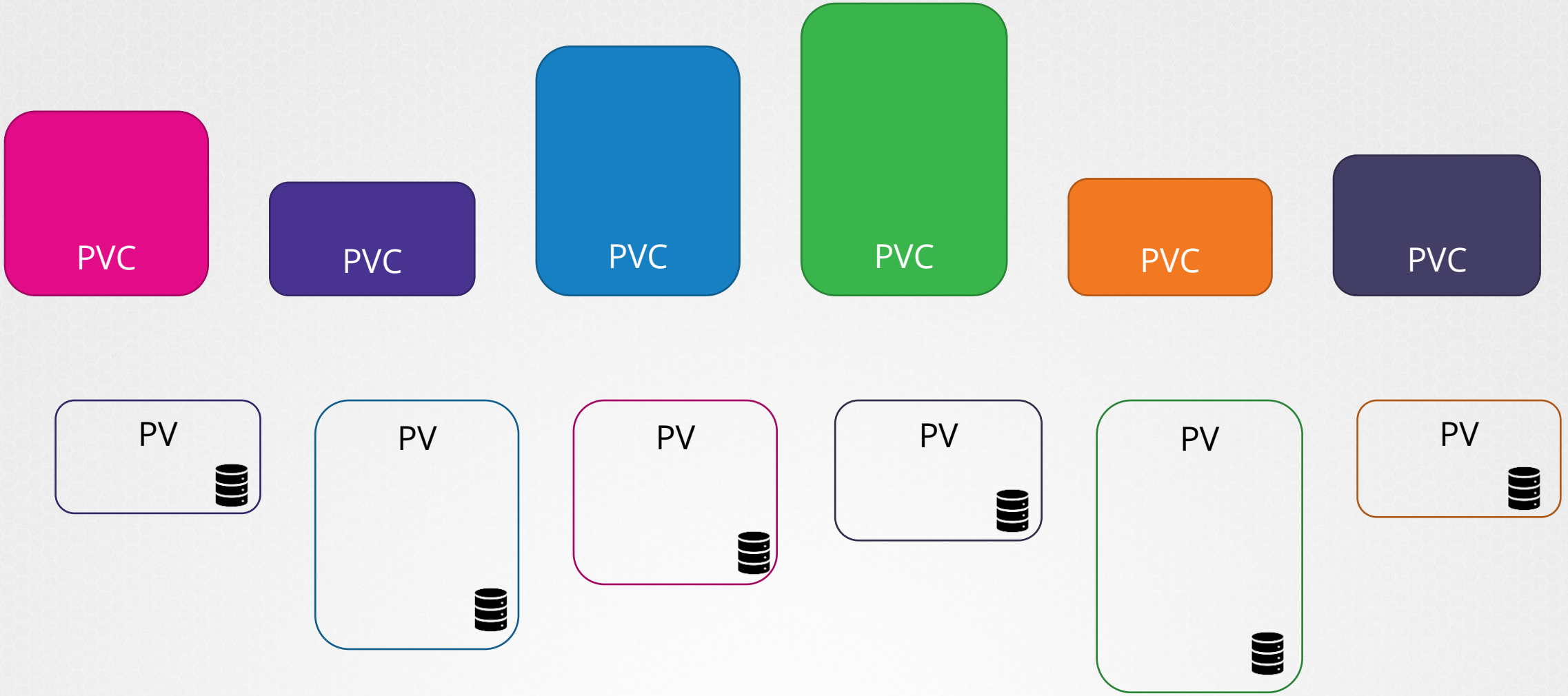
```
NAME      CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS      CLAIM     STORAGECLASS   REASON   AGE
pv-vol1   1Gi        RWO            Retain           Available                                     3m
```

Course Objectives

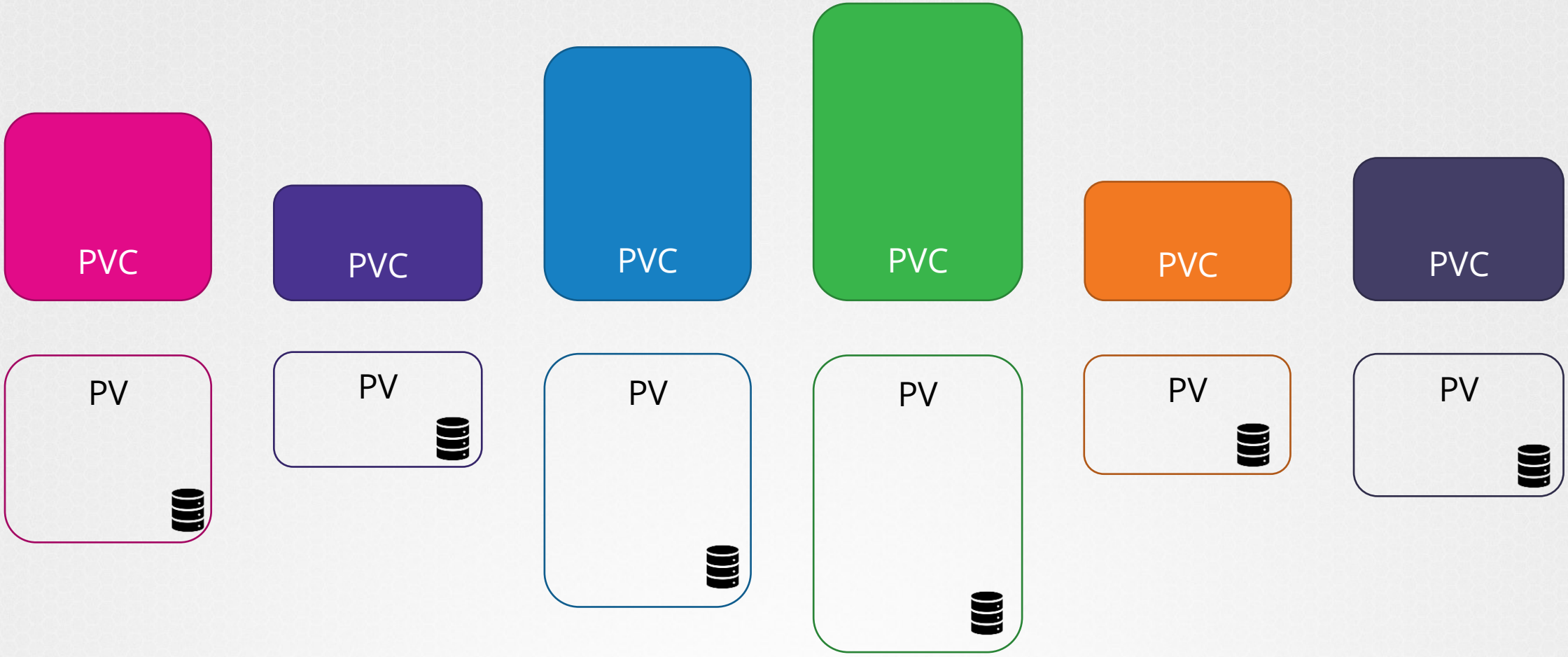
- Core Concepts
- Configuration
- Multi-Container Pods
- Observability
- Pod Design
- Services & Networking
- State Persistence
  - Persistent Volumes
  - Persistent Volume Claims

Persistent Volume Claims

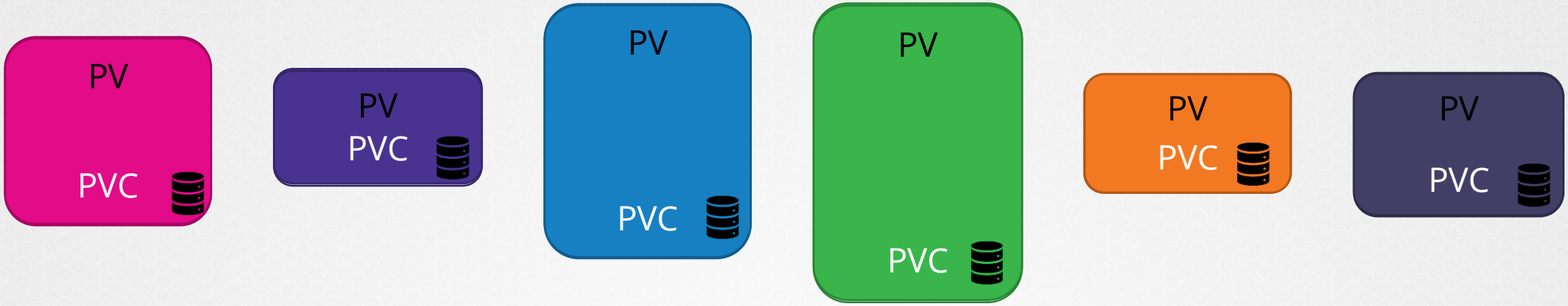
Persistent Volume Claim



Binding



Binding



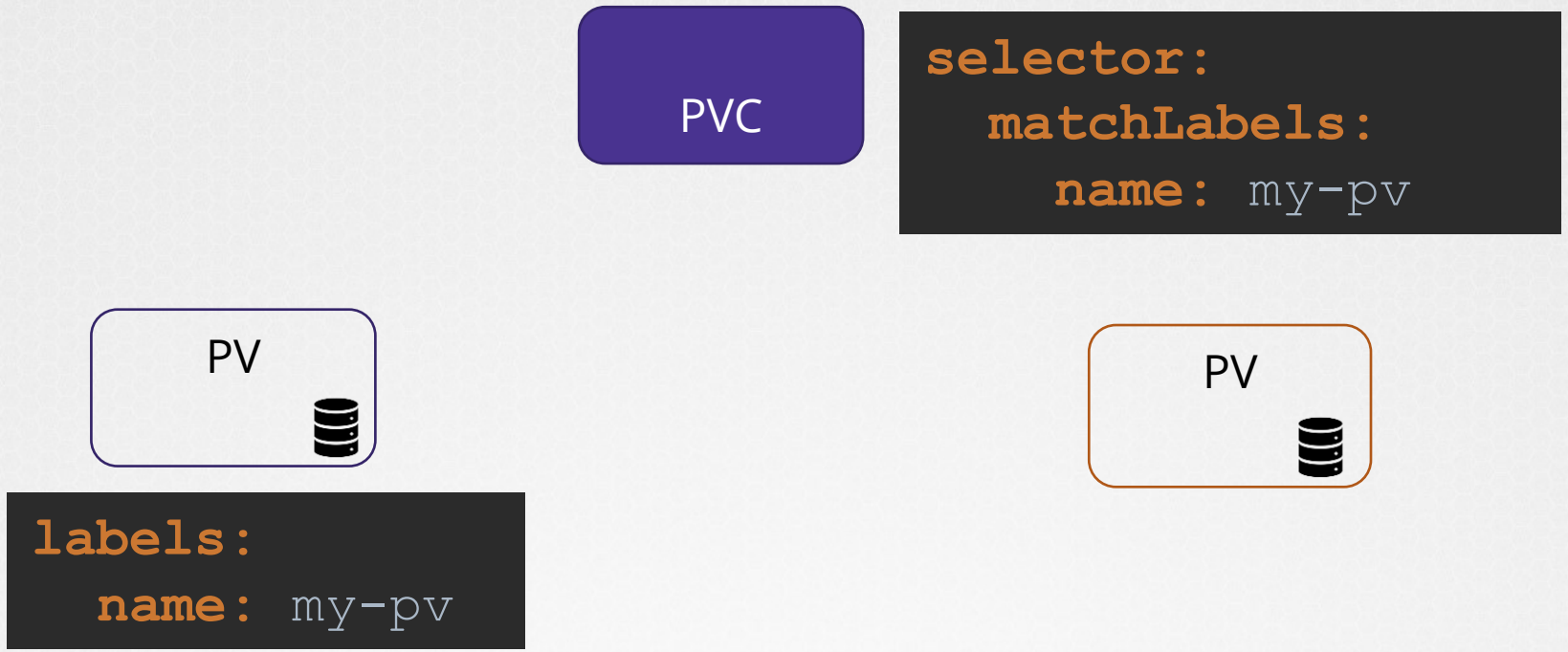
Sufficient Capacity

Access Modes

Volume Modes

Storage Class

Binding



Sufficient Capacity

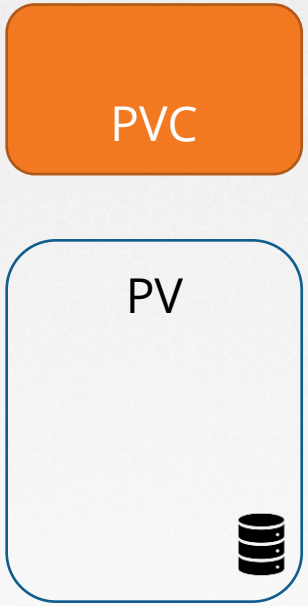
Access Modes

Volume Modes

Storage Class

Selecto r

Binding



Pending

Sufficient Capacity

Access Modes

Volume Modes

Storage Class

Selecto r

Persistent Volume Claim

pvc-definition.yaml

```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: myclaim
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 500Mi
```

```
kubectl create -f pvc-definition.yaml
```

```
kubectl get persistentvolumeclaim
```

```
NAME        STATUS    VOLUME    CAPACITY   ACCESS MODES
myclaim     Pending
```

Persistent Volume Claim

pvc-definition.yaml

```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: myclaim
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 500Mi
```

```
kubectl create -f pvc-definition.yaml
```

pv-definition.yaml

```yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-vol1
spec:
  accessModes:
      - ReadWriteOnce
  capacity:
      storage: 1Gi
  awsElasticBlockStore:
    volumeID: <volume-id>
    fsType: ext4
```

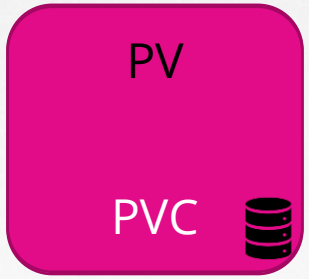
View PVCs

```
kubectl get persistentvolumeclaim
```

```
NAME        STATUS    VOLUME    CAPACITY   ACCESS MODES   STORAGECLASS   AGE
myclaim     Bound     pv-vol1   1Gi        RWO                           43m
```

Delete PVCs

```
kubectl delete persistentvolumeclaim myclaim
persistentvolumeclaim "myclaim" deleted
```




```
persistentVolumeReclaimPolicy:  Recycle
```