A Volume can be used to hold data and state for Pods and containers.

Pod State and Data

Pods live and die so their file system is short-lived (ephemeral)

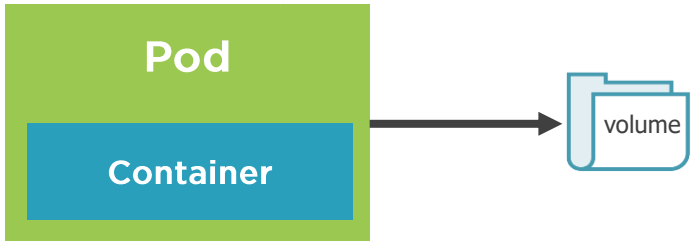
Volumes can be used to store state/data and use it in a Pod

A Pod can have multiple Volumes attached to it

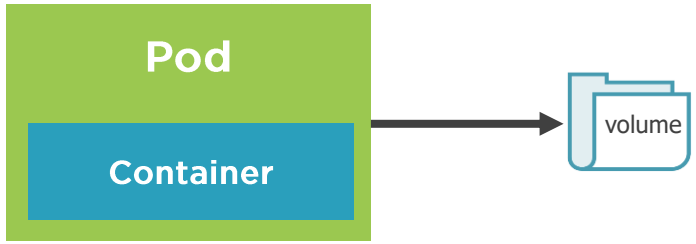
Containers rely on a mountPath to access a Volume

Kubernetes supports:

- Volumes
- PersistentVolumes
- PersistentVolumeClaims
- StorageClasses



Volumes and Volume Mounts



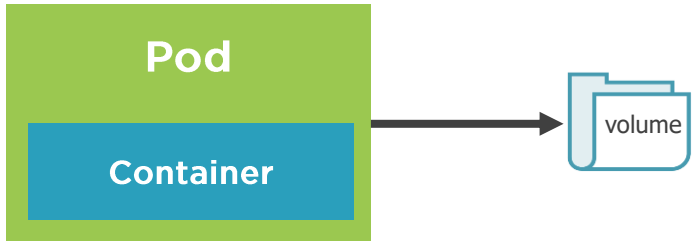
A Volume references a storage location

Must have a unique name

Attached to a Pod and may or may not be tied to the Pod's lifetime (depending on the Volume type)

A Volume Mount references a Volume by name and defines a mountPath

Volumes Type Examples



emptyDir – Empty directory for storing "transient" data (shares a Pod's lifetime) useful for sharing files between containers running in a Pod

hostPath – Pod mounts into the node's filesystem

nfs – An NFS (Network File System) share mounted into the Pod

configMap/secret – Special types of volumes that provide a Pod with access to Kubernetes resources

persistentVolumeClaim – Provides Pods with a more persistent storage option that is abstracted from the details

Cloud – Cluster-wide storage

Volume Types

awsElasticBlockStore	azureDisk	azureFile	cephfs	configMap
csi	downwardAPI	emptyDir	fc	flexVolume
flocker	gcePersistentDisk	glusterfs	hostPath	iscsi
local	nfs	persistentVolumeClaim	projected	portworxVolume
quobyte	rbd	scaleIO	secret	storageos
vsphereVolume				

Viewing a Pod's Volumes

Several different techniques can be used to view a Pod's Volumes

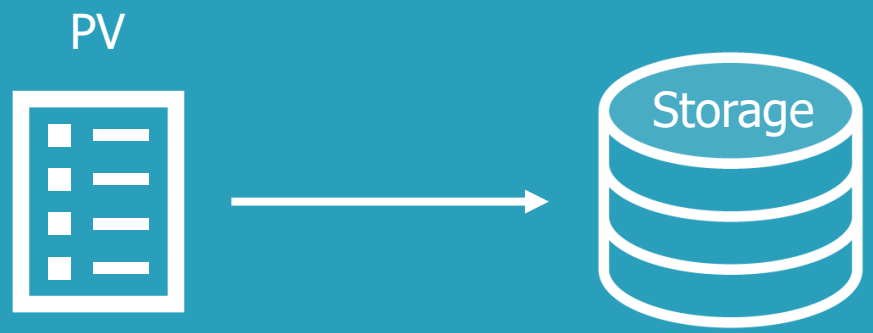
```
# Describe Pod
kubectl describe pod [pod-name]
```

```
Volumes:
  html:
    Type:    EmptyDir (a temporary directory that shares a pod's lifetime)
    Medium:
```

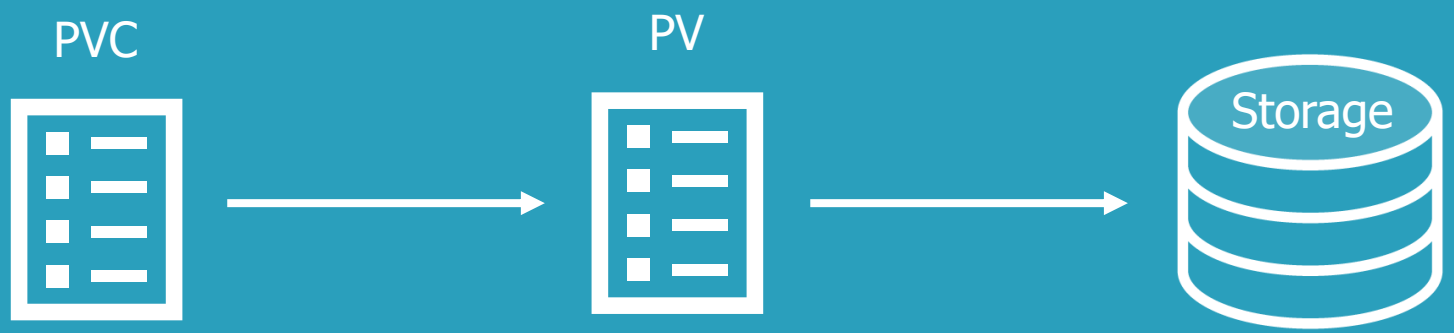
```
# Get Pod YAML
kubectl get pod [pod-name] -o yaml
```

```
volumeMounts:
- mountPath: /html
  name: html
```

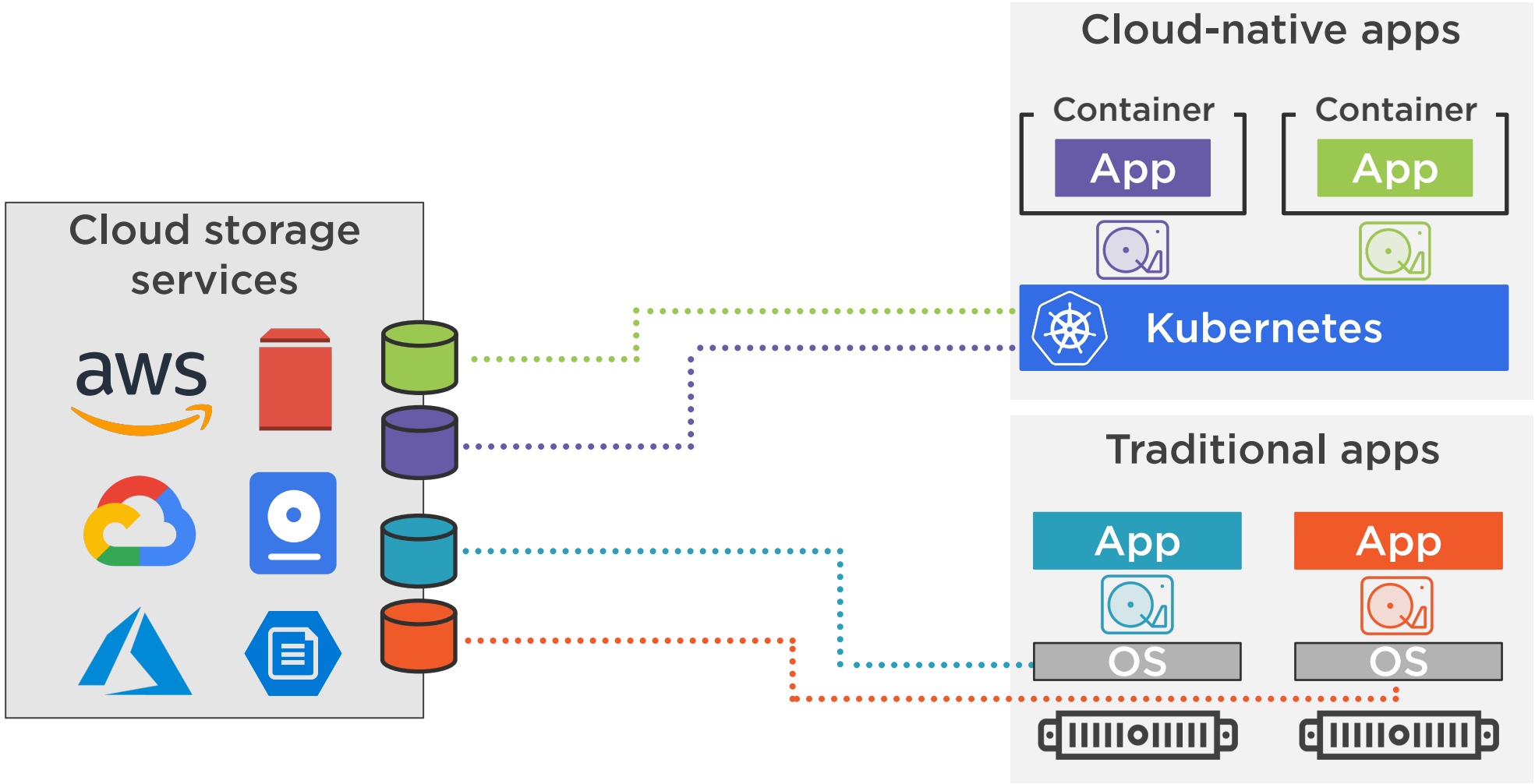
A PersistentVolume (PV) is a cluster-wide storage unit provisioned by an administrator with a lifecycle independent from a Pod.



A PersistentVolumeClaim (PVC) is a request for a storage unit (PV).

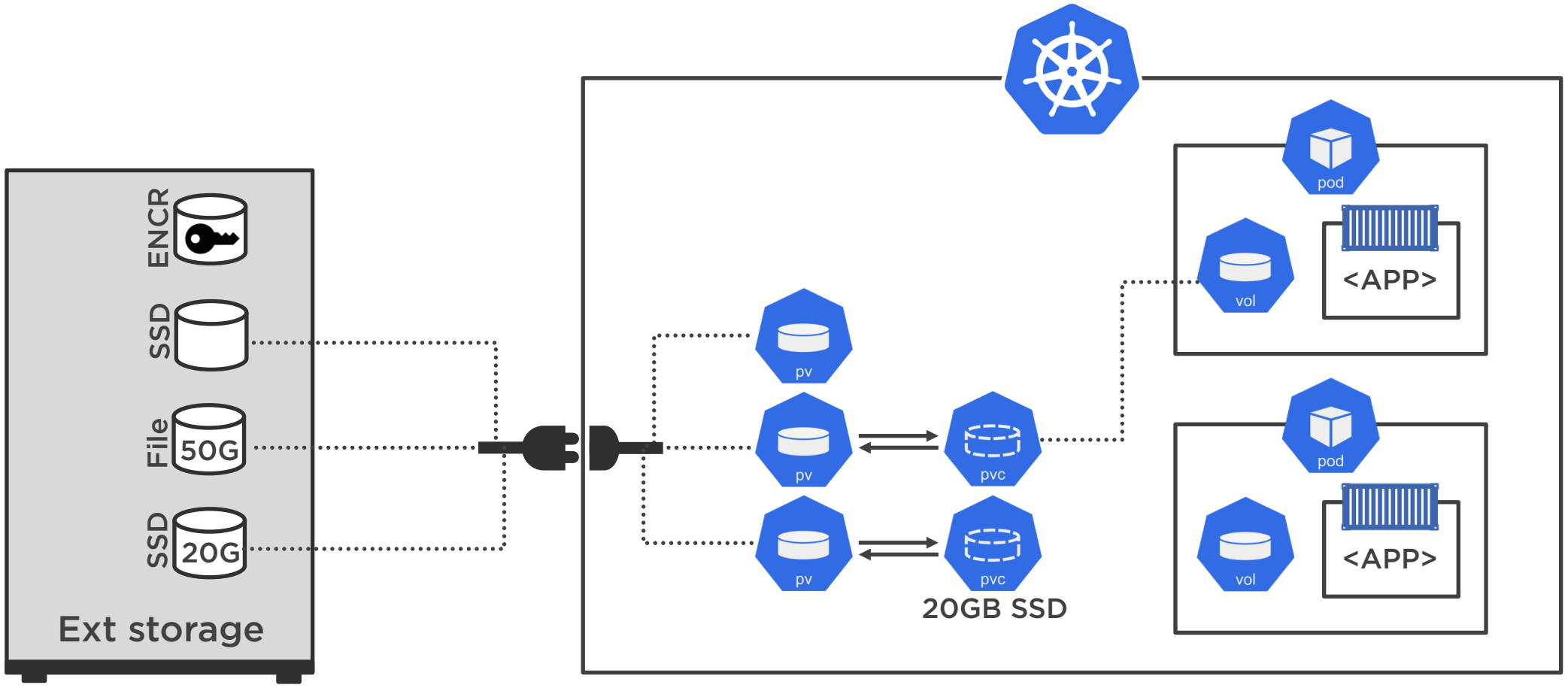


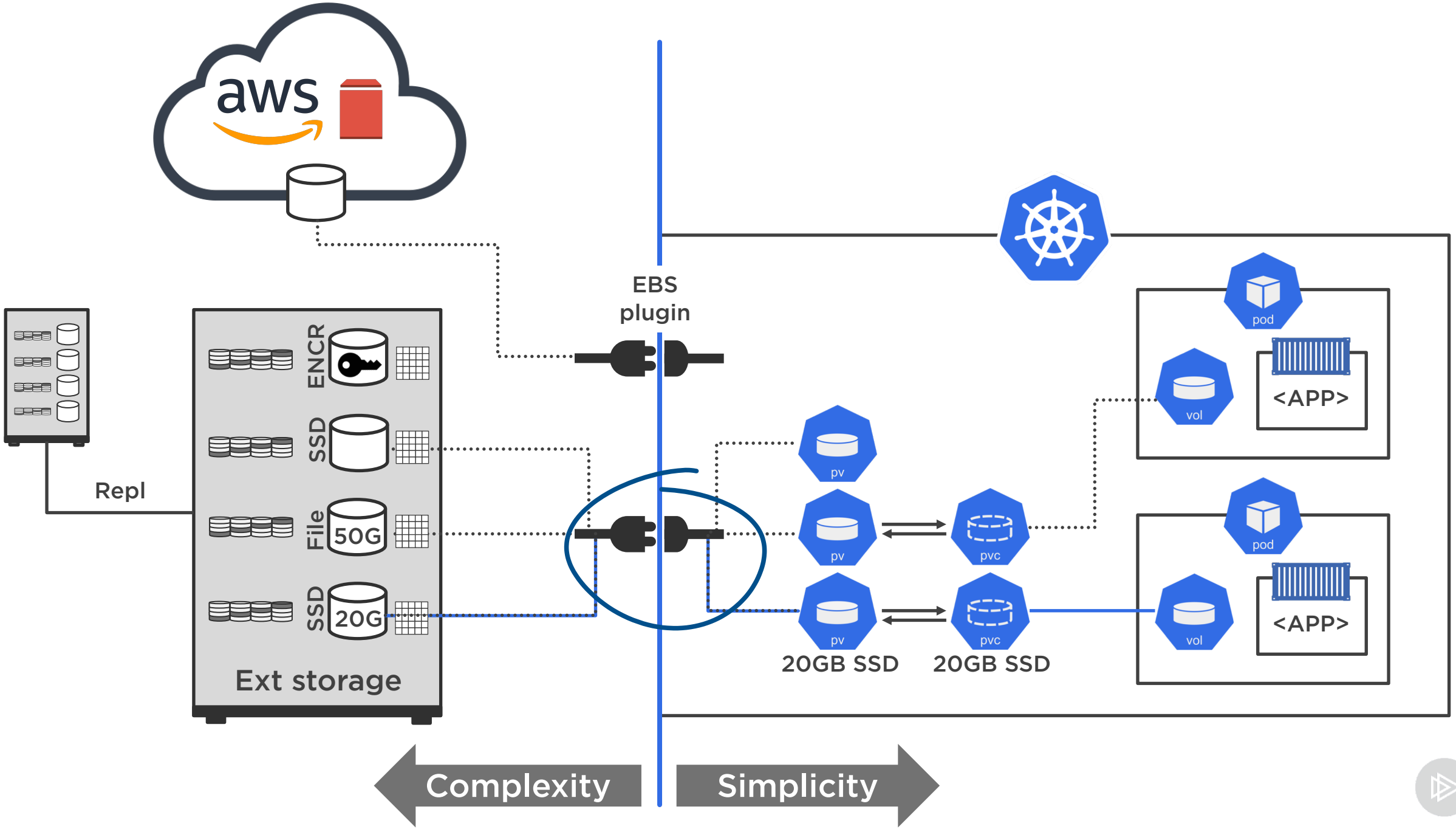
A StorageClass (SC) is a type of storage template that can be used to dynamically provision storage.

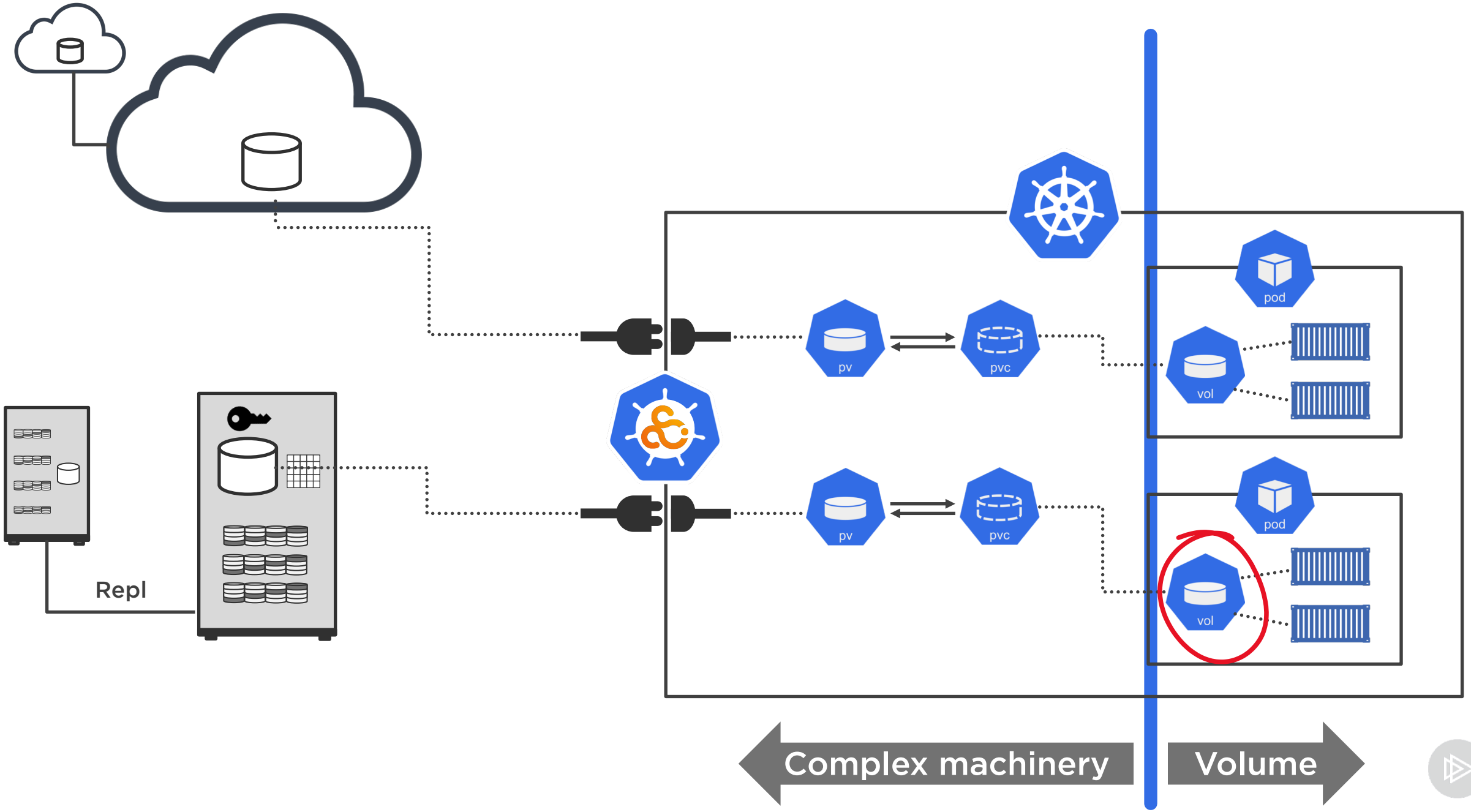

Cloud-native apps
Container
App
Container
App
Kubernetes
Cloud storage services
aws
Traditional apps
App
App
OS
OS
Public cloud

Stateful Application

An application that creates and saves data that needs to be kept.


ENCR
SSD
File
50G
SSD
20G
Ext storage
pv
pv
pv
pvc
pvc
20GB SSD
pod
vol
<APP>
pod
vol
<APP>


aws
EBS
plugin
Repl
ENCR
SSD
File
50G
SSD
20G
Ext storage
pv
pv
pv
pvc
pvc
20GB SSD
20GB SSD
pod
vol
<APP>
pod
vol
<APP>
Complexity
Simplicity


Repl
pv
pvc
pv
pvc
pod
vol
pod
vol
Complex machinery
Volume

YAML Files

Tying the pieces together

ps-pv.yml

```yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: ps-pv
spec:
  accessModes:
  - ReadWriteOnce
  storageClassName: ps-fast
  capacity:
    storage: 50Gi
  persistentVolumeReclaimPolicy: Retain
  gcePersistentDisk:
    pdName: ps-vol
```

ps-pvc.yml

```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: ps-pvc
spec:
  accessModes:
  - ReadWriteOnce
  storageClassName: ps-fast
  resources:
    requests:
      storage: 50Gi
```

ps-pod.yml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: first-pod
spec:
  volumes:
    - name: fast50g
      persistentVolumeClaim:
        claimName: ps-pvc
  containers:
  - image: ubuntu:latest
    name: ctr1
    command:
    - /bin/bash
    - "-c"
    - "sleep 60m"
    volumeMounts:
    - mountPath: /data
      name: fast50g
```

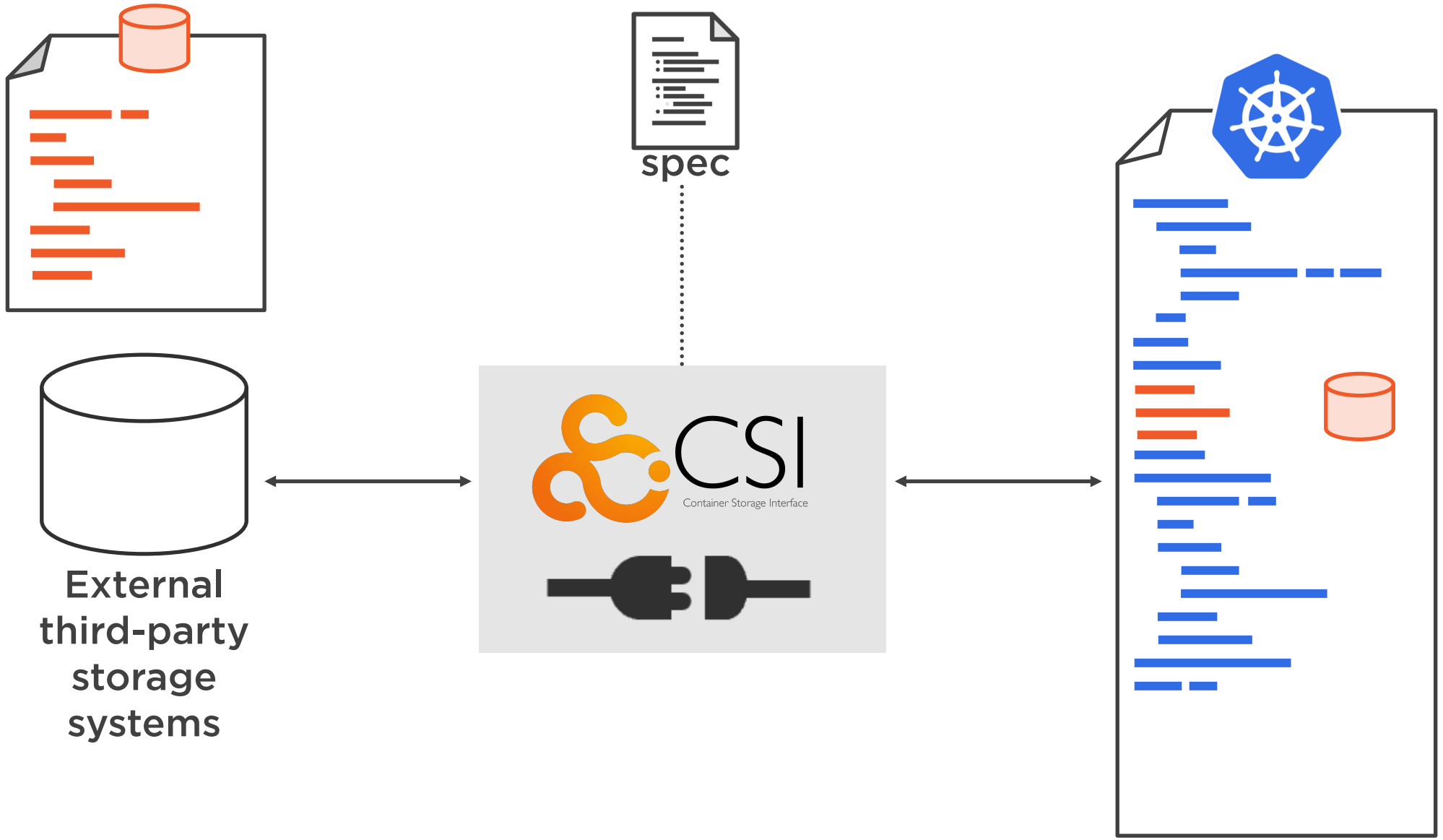
```yaml
apiVersion: v1
kind: Pod
metadata:
  name: first-pod
spec:
  volumes:
    - name: fast50g
      persistentVolumeClaim:
        claimName: ps-pvc
  containers:
  - image: ubuntu:latest
    name: ctr1
    command:
    - /bin/bash
    - "-c"
    - "sleep 60m"
    volumeMounts:
    - mountPath: /data
      name: fast50g
```

Volume

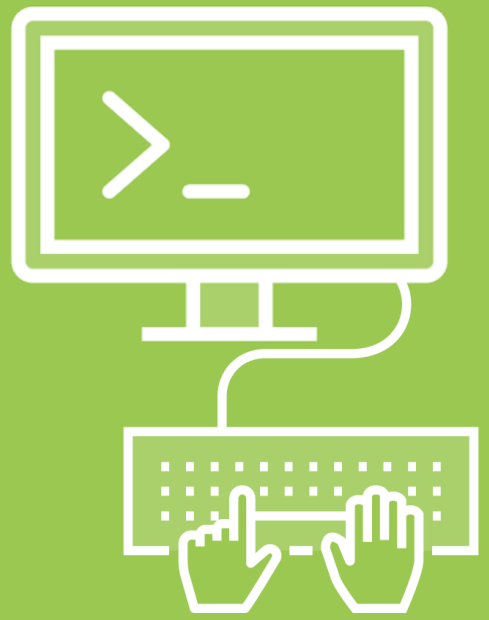
Container spec

- Volume name
- Creating the volume using the PVC
- Referencing the volume to be mounted

The Container Storage Interface (CSI)


spec
CSI
Container Storage Interface
External
third-party
storage
systems

Hands-on:
Static Provisioning



Hands-on

Examples are mainly on GKE, but should work on most other platforms. Kubernetes is Kubernetes!

- RWO: Read-write once
- RWM: Read-write many
- ROM: Read-only many

YAML Files

Tying the pieces together

ps-pv.yml

```yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: ps-pv
spec:
  accessModes:
  - ReadWriteOnce
  storageClassName: ps-fast
  capacity:
    storage: 50Gi
  persistentVolumeReclaimPolicy: Retain
  gcePersistentDisk:
    pdName: ps-vol
```

ps-pvc.yml

```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: ps-pvc
spec:
  accessModes:
  - ReadWriteOnce
  storageClassName: ps-fast
  resources:
    requests:
      storage: 50Gi
```

ps-pod.yml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: first-pod
spec:
  volumes:
    - name: fast50g
      persistentVolumeClaim:
        claimName: ps-pvc
  containers:
  - image: ubuntu:latest
    name: ctr1
    command:
    - /bin/bash
    - "-c"
    - "sleep 60m"
    volumeMounts:
    - mountPath: /data
      name: fast50g
```

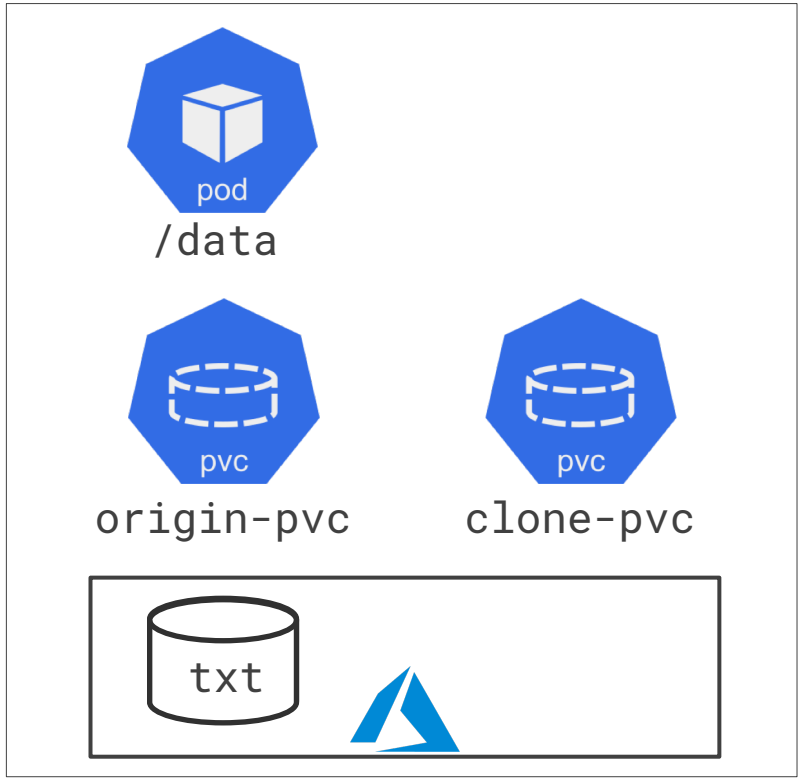
Hands-on:
Dynamic Provisioning

Dynamic volume creation

Classes/tiers

Advanced Volume Features

Clones


pod
/data
pvc
origin-pvc
pvc
clone-pvc
txt