# Share Volume In K8S

Jane Liu L

# Agenda

- Background knowledge share
- Question
- Overview process
- How to solve the question
- Why that design
- Answer

# Background knowledge share-卷

临时卷

不关心数据在Pod重启后是否可用
跟Pod的生命周期挂钩

emptyDir
configMap

```
apiVersion: v1
kind: Pod
metadata:
  name: volume-test
spec:
  containers:
  - image: busybox
    name: test-container
    volumeMounts:
    - mountPath: /data
      name: test-volume
  volumes:
  - name: test-volume
...
```

持久卷

持久化保存在卷内的数据
独立于Pod的生命周期

可以提前创建
也可以动态供应

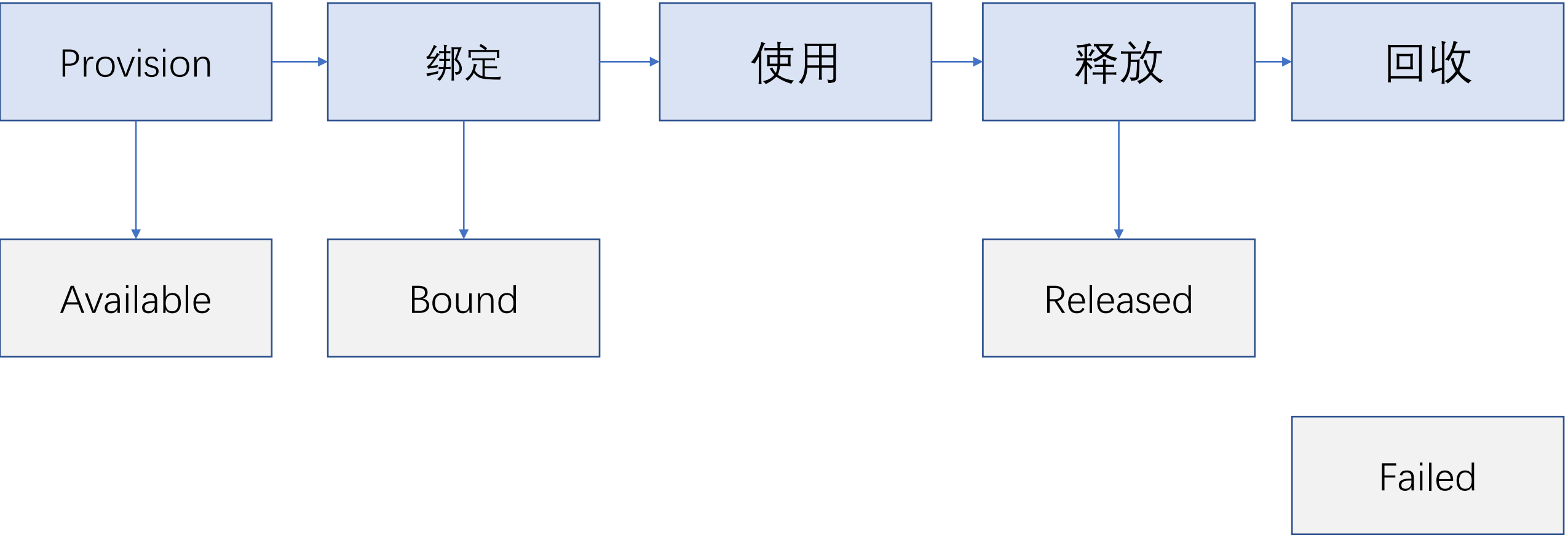
NFS
ISCSI
AWSEBS

# 持久卷

PersistentVolume
代表的是一块存储
ClusterScope

PersistentVolumeClaim
代表的是用户对存储的请求
NamespaceScope

# 持久卷生命周期



# 持久卷特性

容量:
Capacity

访问模式:
ReadWriteOnce
ReadOnlyMany
ReadWriteMany

挂载模式:
Filesystem
Block

回收策略:
Retain/Recycle/Delete

事先创建

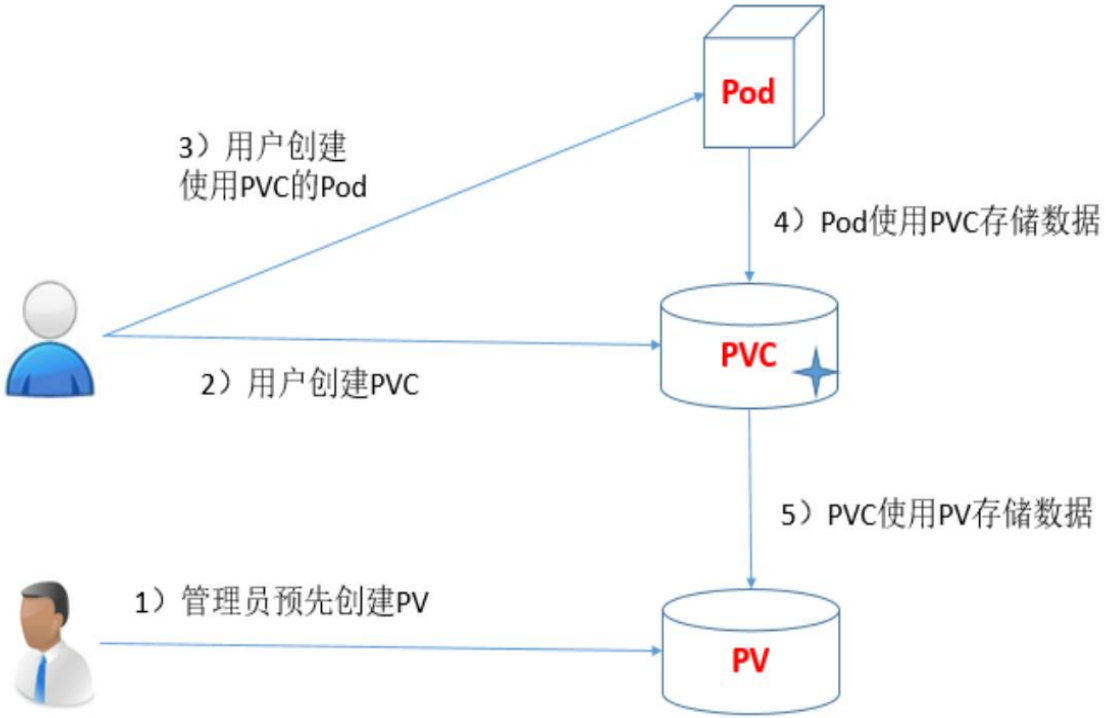
```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: xx-pv
spec:
  accessModes:
  - ReadWriteMany
  capacity:
    storage: 1Gi
  nfs:
    path: /xx
    server: sxxx
  persistentVolumeReclaimPolicy: Retain
  volumeMode: Filesystem
```

动态provision

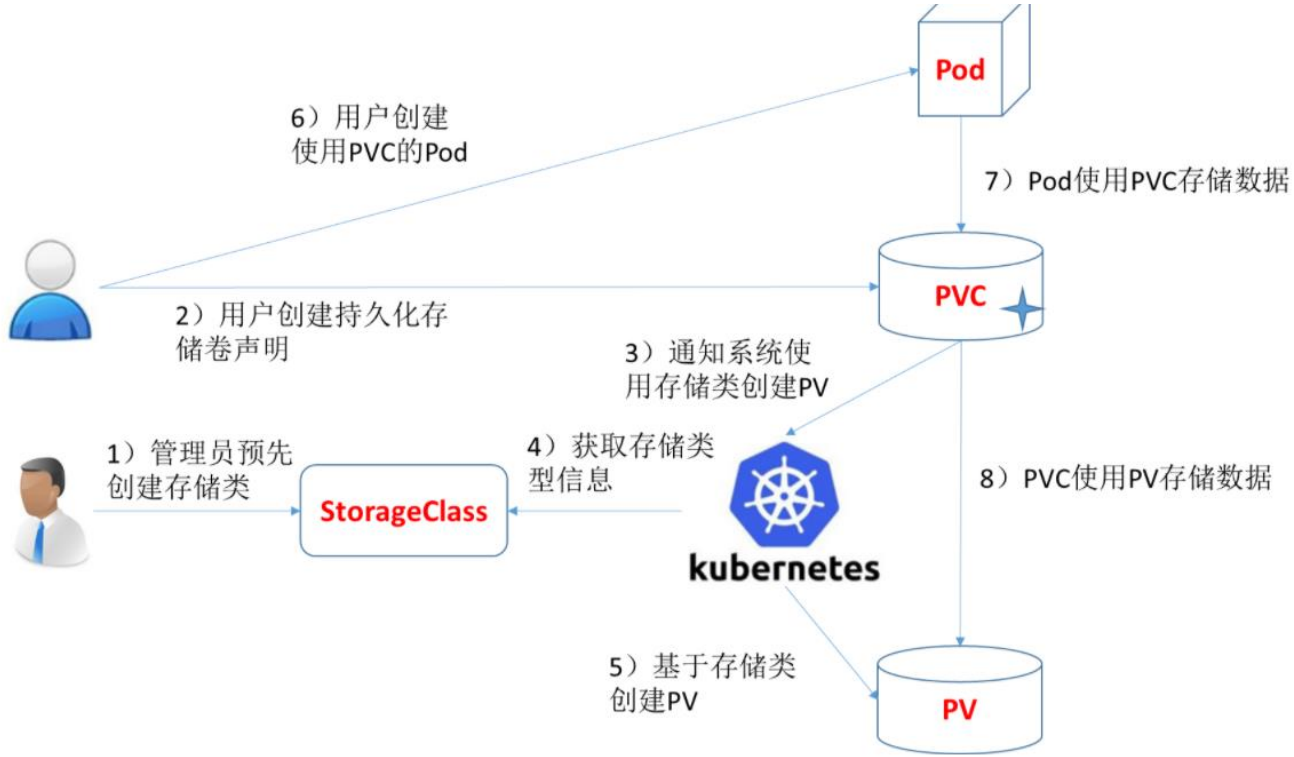
```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: claim1
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: awsebsx
  resources:
    requests:
      storage: 30Gi
```

# 持久卷Provision

事先创建




动态provision



# Agenda

- Background knowledge share
- Question
- Overview process
- How to solve the question
- Why that design
- Answer

# Question

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: pvc1
spec:
  storageClassName: azure-disk
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
```

```
apiVersion: v1
kind: Pod
metadata:
  name: busybox1
spec:
  volumes:
    - name: pvc1
      persistentVolumeClaim:
        claimName: pvc1
  containers:
    - name: busybox
      image: busybox
      command:
        - sleep
```

```
variable "default_node_pool_availability_zones" {
  type        = list(string)                                    ıta"
  description = "Availability zones for default node pool.
  default     = ["1", "2", "3"]
}
```

# Question

(我们刚刚观察到PV/PVC 里面没有zone的列)

(而我们在公有云里面，为了disaster recover会选择尽量把应用部署到不同的zone的机器上)

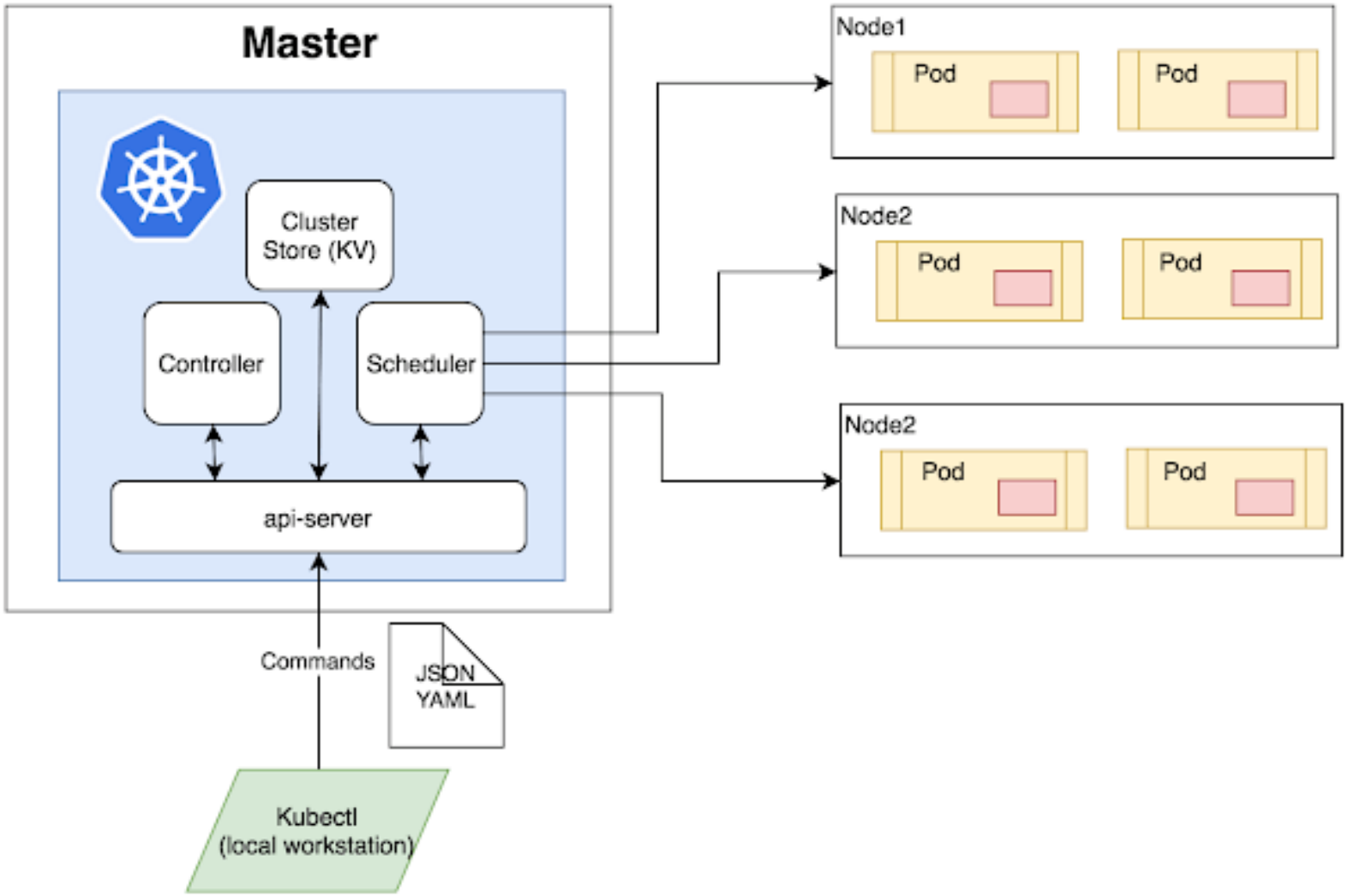
我们尝试使用PVC和Deployment的时候，当我们这个Deployment需要挂载PVC的时候，是否会出现了以下问题：

Deployment的Pod被调度到节点A, 而节点A是属于zone 1的，PVC的磁盘却是zone 2，导致无法绑定该PVC?

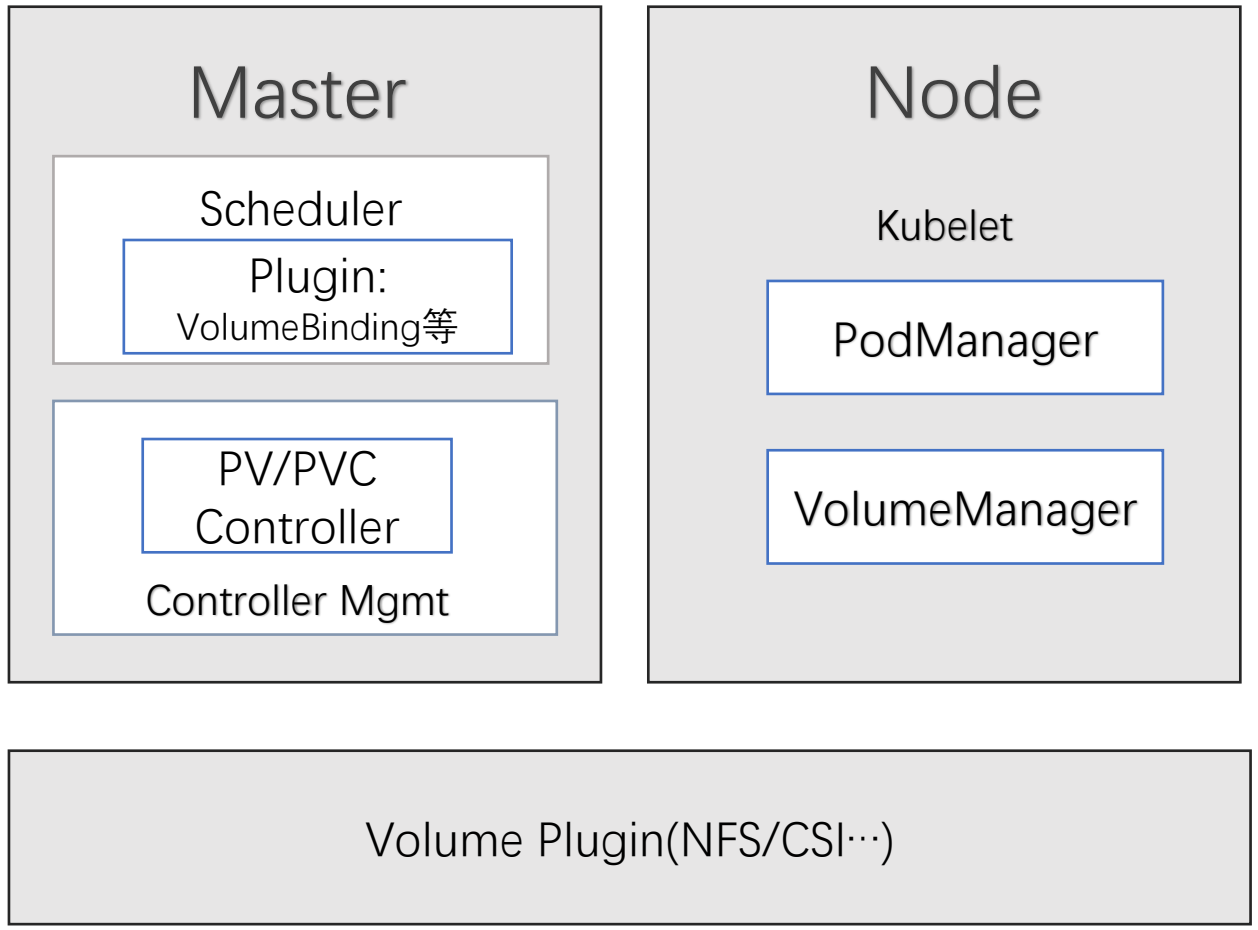
# Agenda

- Background knowledge share
- Question
- Overview process
- How to solve the question
- Why that design
- Answer

# Overview process



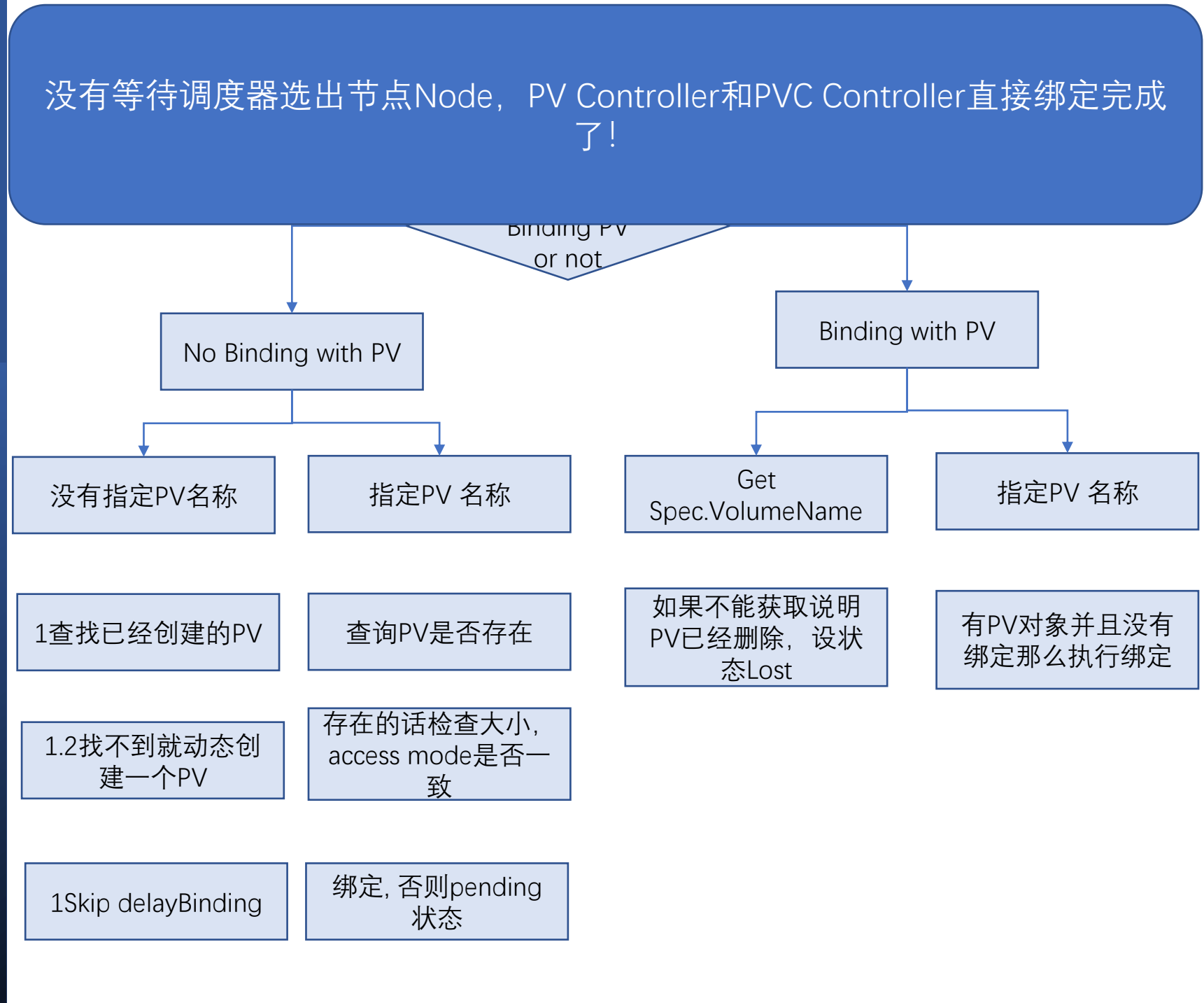
# Volume Related on components



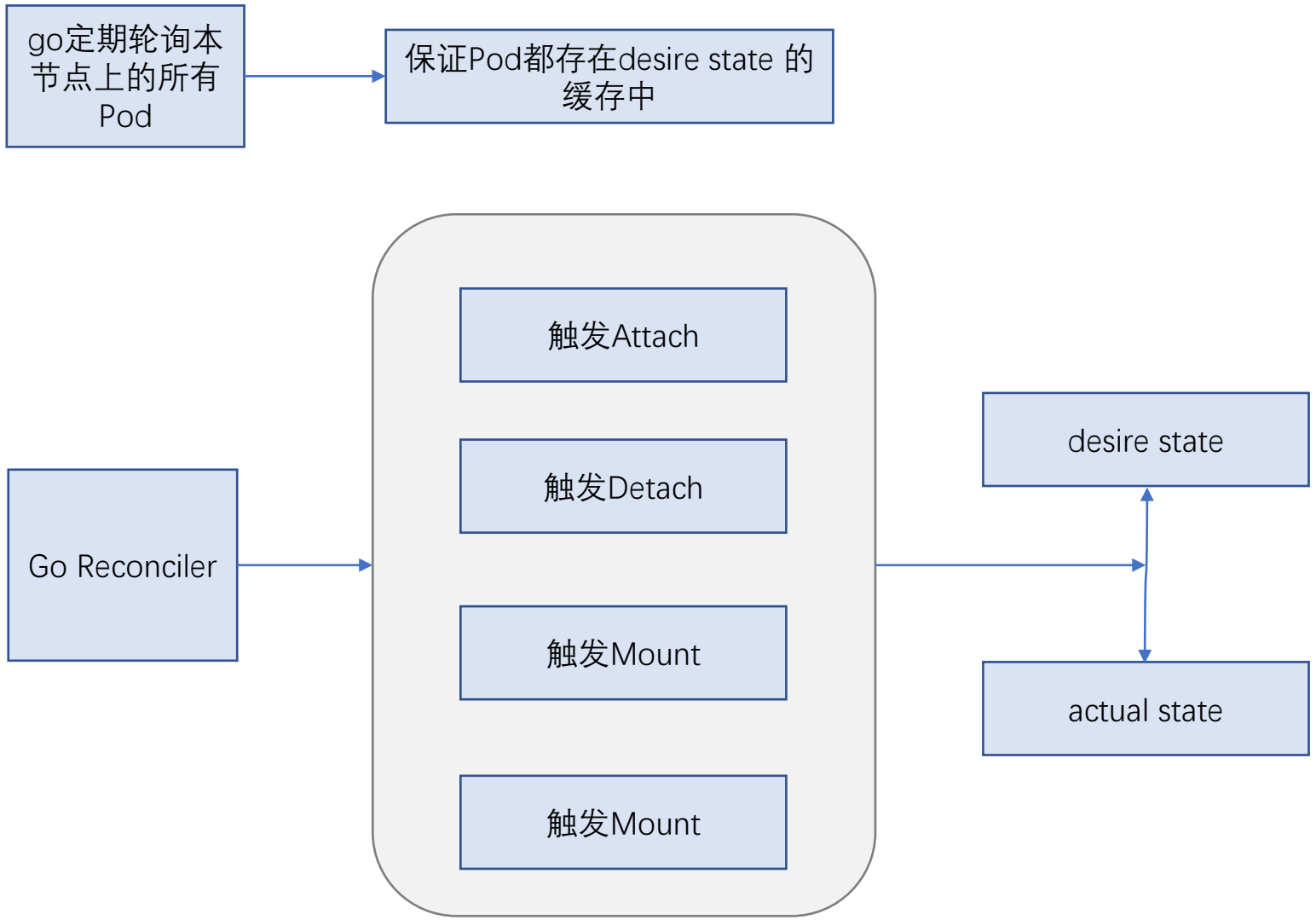
provisionClaim

provisionClaim

# Overview process – PVC Controller

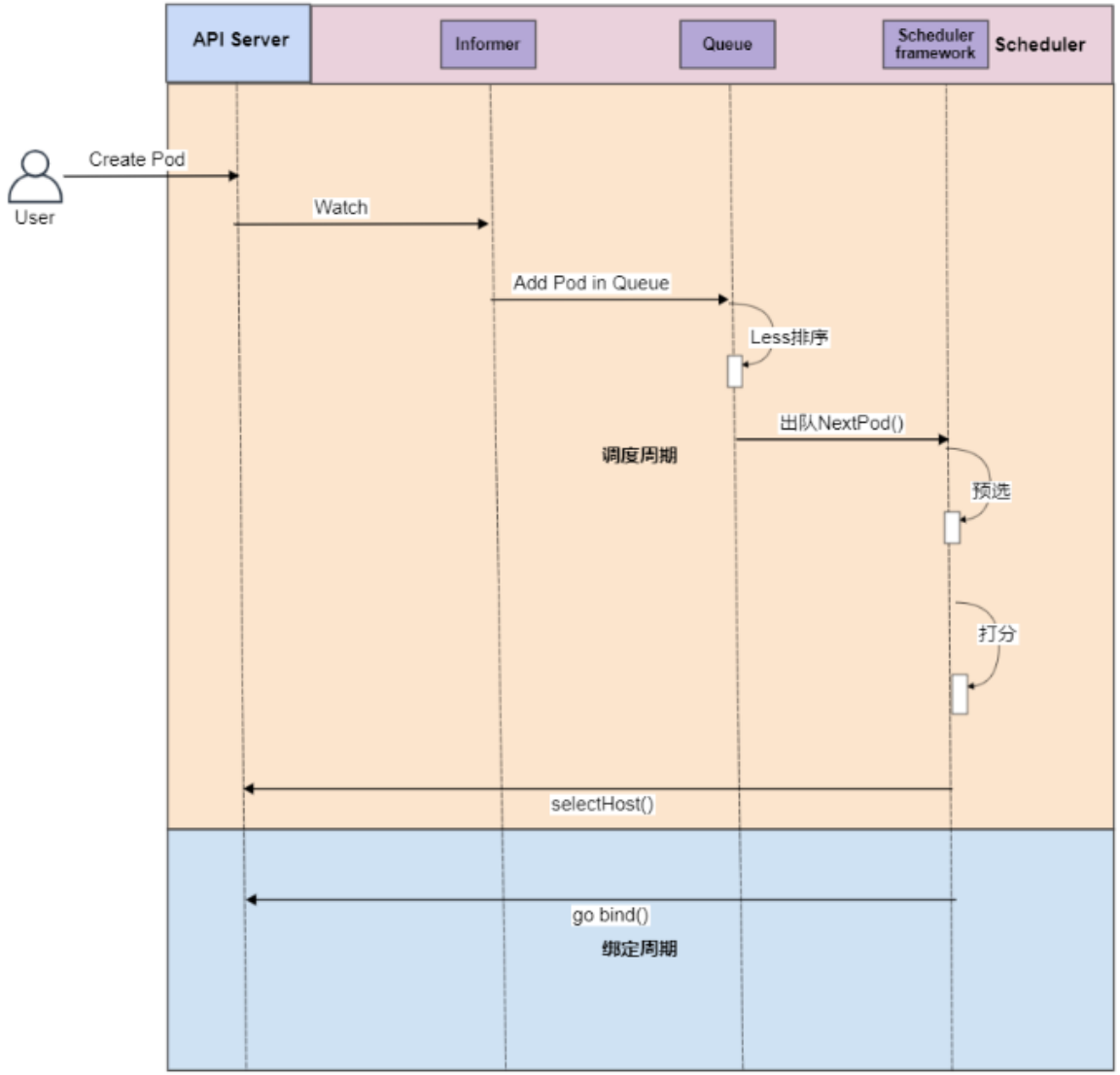


# Kubelet VolumeManager



# Overview process – Scheduler

- 入队
- 过滤
- 打分
- 异步绑定

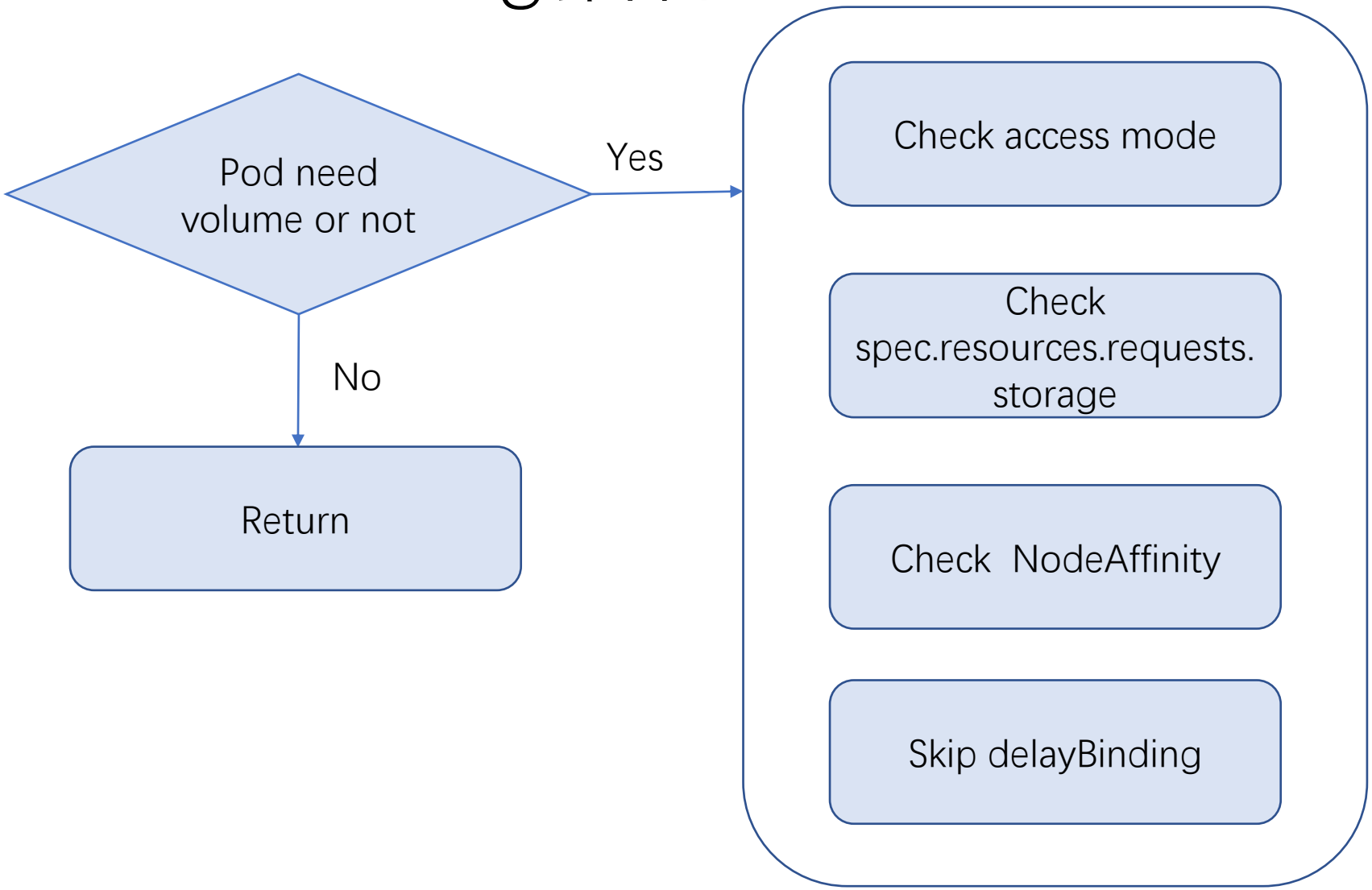


# Overview process – 调度过滤插件

涉及到Pod挂载卷的插件有以下两个调度插件:

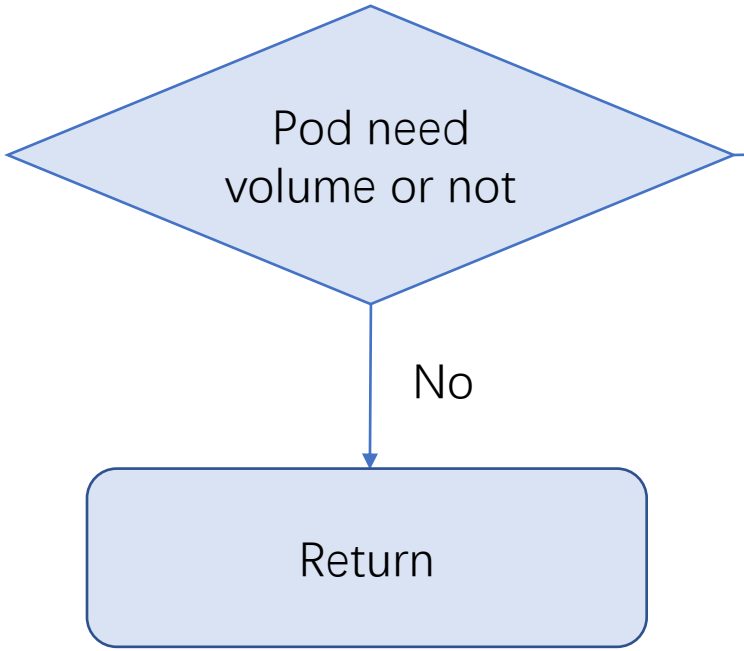
- Volumebinding插件
- Volumezone插件

# Volumebinding插件



# Volumezon

Once we use zone on K8S, all node will label with Zone




Check node label

failure-domain.beta.kubernetes.io/zone
failure-domain.beta.kubernetes.io/region

Skil delayBinding

PV Label

failure-domain.beta.kubernetes.io/zone
failure-domain.beta.kubernetes.io/region

Compare PV & node Label

# What is delayBinding

延时绑定策略

- 预分配使用本地卷的PV
- 通过NodeAffinity方式标记PV位置
- 创建StorageClass，通过StorageClass间接标记PVC的延时绑定
- 标记该PVC需要延后到Node选择出来之后再绑定

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: managed-csi
provisioner: disk.csi.azure.com
parameters:
  skuname: StandardSSD_LRS
reclaimPolicy: Delete
volumeBindingMode: WaitForFirstConsumer
```

# Agenda

- Background knowledge share
- Question
- Overview process
- How to solve the question
- Why that design
- Answer

# How to solve the question

在启动了multiple zone的情况下：
使用延迟调度
选择高版本》1.17以上，带有volumezone功能的K8S

# Agenda

- Background knowledge share
- Question
- Overview process
- How to solve the question
- Why that design
- Answer

# Why that design

Why not all bingding/attach by Kubelet, need to by PVC Controller binding & attach?

# Agenda

- Background knowledge share
- Question
- Overview process
- How to solve the question
- Why that design
- Answer

# Answer