

阿里云 × CLOUD NATIVE
COMPUTING FOUNDATION

云原生技术公开课

第 02 讲

应用编排与管理：Job & DaemonSet

陈显鹭 阿里巴巴技术专家



扫码关注“阿里巴巴云原生”公众号
获取第一手技术资料



需求来源



用例解读



操作演示



架构设计



课后习题

Job 背景问题

我们可以通过Pod 来直接运行任务进程吗？

如果这样做， 以下的问题有什么方式来解决？

1. 如何保证Pod内进程正确的结束？
2. 如果进程运行失败， 如何重试？
3. 如何管理多个任务且任务之间有互相依赖关系？
4. 如何并行运行任务并管理他们的队列大小？

Job: 管理任务的控制器

Job 能帮助我们做什么事情?

1. 创建一个或多个Pod确保指定数量的Pod可以成功地运行终止
2. 跟踪Pod状态, 根据配置及时重试失败的 Pod
3. 确定依赖关系, 保证上一个任务运行完毕后再运行下一个任务
4. 控制任务并行度, 并根据配置确保Pod 队列大小



Job 语法

新知识点:

restartPolicy: 重启策略

backoffLimit: 重试次数限制

```
controllers/job.yaml
```

```
apiVersion: batch/v1
```

```
kind: Job
```

Job 类型以及元信息

```
metadata:
```

```
  name: pi
```

```
spec:
```

```
  template:
```

```
    spec:
```

```
      containers:
```

```
      - name: pi
```

```
        image: perl
```

```
        command: ["perl", "-Mbignum=bpi", "-wle", "print bpi(2000)"]
```

```
restartPolicy: Never
```

重启策略

```
backoffLimit: 4
```

重试次数

查看Job状态

```
$ kubectl create -f job1.yaml
```

```
$ kubectl get jobs
```

NAME	COMPLETIONS	DURATION	AGE
pi	1/1	71s	4m6s

COMPLETIONS: 完成pod数量

DURATION: Job实际业务运行时长

AGE: deployment创建的时长

查看Pod

```
$ kubectl get pod  
NAME  
pi-4cfrp
```

Pod名字格式:
`${job-name}-${random-suffix}`

```
$ kubectl get pods pi-4cfrp -o yaml  
apiVersion: v1  
kind: Pod  
metadata:  
  labels:  
    controller-uid: d2912fc6-6599-11e9-9fea-00163e00cf4e  
    job-name: pi  
  name: pi-4cfrp  
  namespace: default  
  ownerReferences:  
  - apiVersion: batch/v1  
    blockOwnerDeletion: true  
    controller: true  
    kind: Job  
    name: pi  
    uid: d2912fc6-6599-11e9-9fea-00163e00cf4e
```

labels均为自动匹配与添加

并行运行Job

新知识点:

1. completions: 代表本pod队列执行次数，这里8代表这个任务将被执行8次
2. parallelism: 代表并行执行个数，这里的2代表并行执行的pod数量，也就是说会有2个 pod 同时运行

```
apiVersion: batch/v1
kind: Job
metadata:
  name: paral-1
spec:
  completions: 8
  parallelism: 2
  template:
    spec:
      containers:
      - name: param
        image: ubuntu
        command: ["/bin/sh"]
        args: ["-c", "sleep 30; date"]
      restartPolicy: OnFailure
```

查看并行Job运行

kubectl get jobs			
NAME	COMPLETIONS	DURATION	AGE
paral-1	8/8	2m23s	2m34s

\$ kubectl get pods				
NAME	READY	STATUS	RESTARTS	AGE
paral-1-4w6k9	0/1	Completed	0	2m26s
paral-1-jckpp	0/1	Completed	0	2m26s
paral-1-6r4l4	0/1	Completed	0	110s
paral-1-thfcv	0/1	Completed	0	110s
paral-1-8dzw8	0/1	Completed	0	40s
paral-1-s56kg	0/1	Completed	0	40s
paral-1-lzldc	0/1	Completed	0	73s
paral-1-x4w7v	0/1	Completed	0	73s

- 1. pod age 表明每次job同时创建了两个pod
- 2. 本job一共运行了四个批次，一共运行8个pod

CronJob 语法

新知识点:

schedule: crontab 时间格式相同

startingDeadlineSeconds: Job最长启动时间

concurrencyPolicy: 是否允许并行运行

successfulJobsHistoryLimit: 允许留存历史job个数

```
apiVersion: batch/v1beta1
kind: CronJob
metadata:
  name: hello
spec:
  schedule: "*/1 * * * *" (every minute)
  jobTemplate:
    spec:
      template:
        spec:
          containers:
            - name: hello
              image: busybox
              args:
                - /bin/sh
                - -c
                - date; echo Hello from the
Kubernetes cluster

          restartPolicy: OnFailure
        startingDeadlineSeconds: 10
        concurrencyPolicy: Allow
        successfulJobsHistoryLimit: 3
```



需求来源



用例解读



操作演示



架构设计



课后习题

1

需求来源

.....

2

用例解读

.....

3

操作演示

.....

4

架构设计

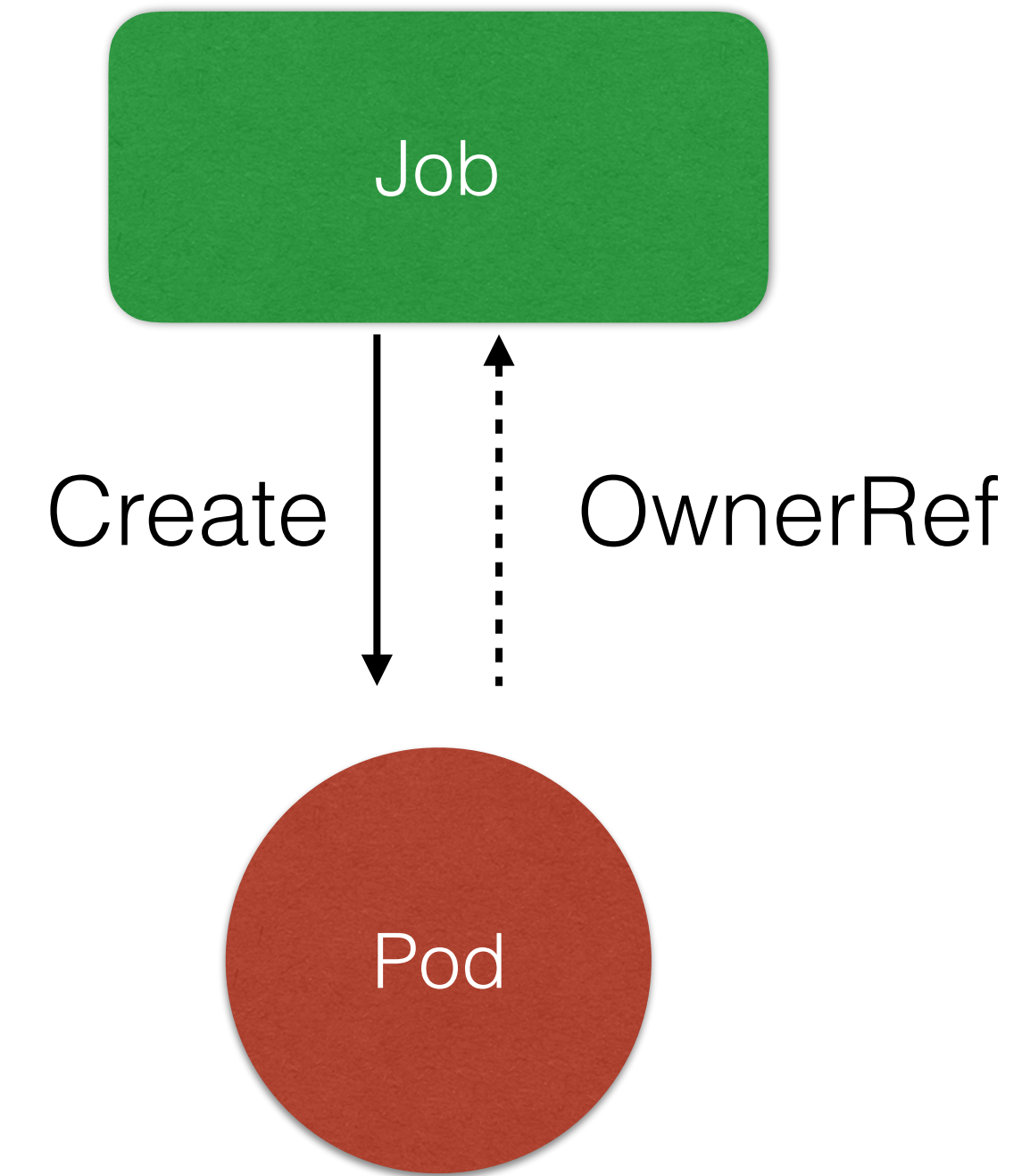
.....

5

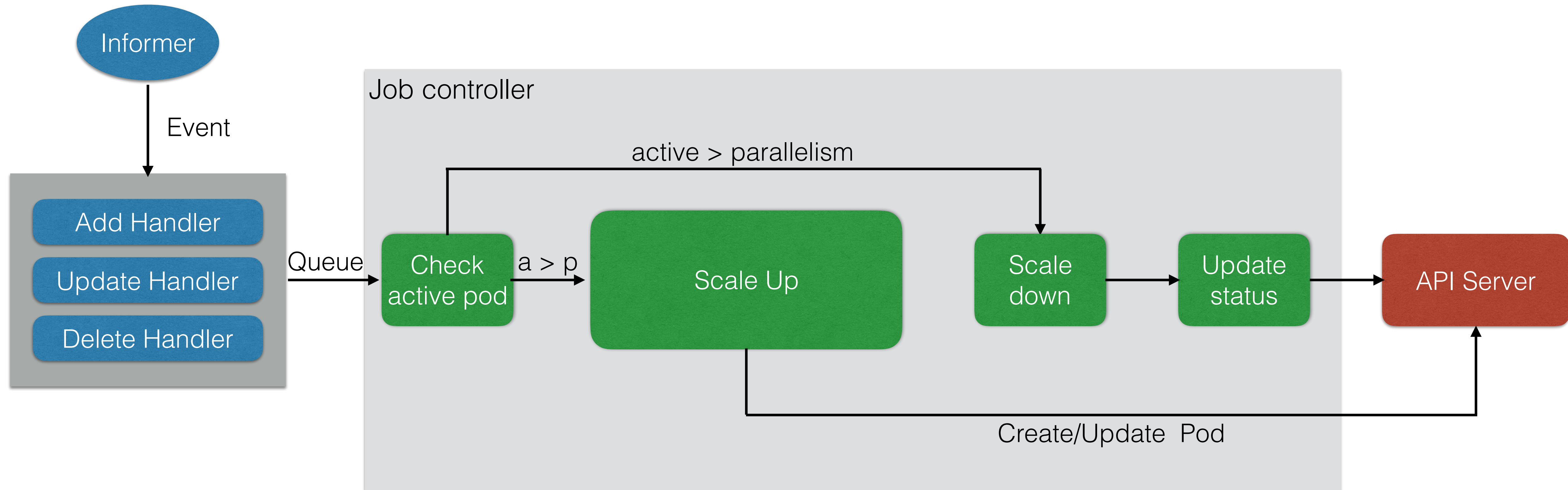
课后习题

管理模式

1. Job Controller 负责根据配置创建Pod
2. Job Controller 跟踪Job 状态，根据配置及时重试Pod或者继续创建
3. Job Controller 会自动添加label 来跟踪对应的 pod,并根据配置并行或者串行创建 Pod



Job 控制器





需求来源



用例解读



操作演示



架构设计



课后习题

DaemonSet 背景问题

我们可以让每个集群内的节点都运行一个相同的Pod吗？

如果这样做， 以下的问题有什么方式来解决？

1. 如何保证每个节点都运行一个pod？
2. 如果新节点加入集群， 如何感知并部署对应的Pod？
3. 如果有节点退出， 如何删除对应的Pod？
4. 如果Pod状态异常， 如何监控并恢复Pod的状态？

DaemonSet: 守护进程控制器

DaemonSet 能帮助我们做什么事情?

1. 保证集群内每一个(或者一些)节点都运行一组相同的Pod
2. 跟踪集群节点状态, 保证新加入的节点自动创建对应的Pod
3. 跟踪集群节点状态, 保证移除的节点删除对应的Pod
4. 跟踪Pod状态, 保证每个节点Pod 处于运行状态



DaemonSet 语法

适用场景:

1. 集群存储进程: glusterd, ceph
2. 日志收集进程: fluentd, logstash
3. 需要在每个节点运行的监控收集器

```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: fluentd-elasticsearch
  namespace: kube-system
  labels:
    k8s-app: fluentd-logging
spec:
  selector:
    matchLabels:
      name: fluentd-elasticsearch
  template:
    metadata:
      labels:
        name: fluentd-elasticsearch
    spec:
      containers:
        - name: fluentd-elasticsearch
          image: fluent/fluentd:v1.4-1
```

查看DaemonSet 状态

```
# kubectl get ds
NAME                DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
fluentd-elasticsearch 4           4         4         4             4           <none>          4s

# kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
fluentd-elasticsearch-bdq2q 1/1     Running   0          9s
fluentd-elasticsearch-gxv68 1/1     Running   0          9s
fluentd-elasticsearch-nwczf 1/1     Running   0          9s
fluentd-elasticsearch-xmv8k 1/1     Running   0          9s
```

状态描述:(本集群公有四个节点)

DESIRED: 需要的pod个数

CURRENT: 当前已存在的pod个数

READY: 就绪的个数

UP-TO-DATE: 最新创建的个数

AVAILABLE: 可用pod个数

NODE SELECTOR: 节点选择标签

更新DaemonSet

更新策略:

1. RollingUpdate: DaemonSet默认更新策略, 当更新DaemonSet 模板后, 老的Pod会被先删除, 然后再去创建新的Pod, 可以配合健康检查做滚动更新
2. OnDelete: 当DaemonSet模板更新后, 只有手动的删除某一个对应的 Pod, 此节点 Pod 才会被更新。

```
kubectl set image ds/fluentd-elasticsearch fluentd-elasticsearch=fluent/fluentd:v1.4
```

```
# kubectl rollout status ds/fluentd-elasticsearch
```

```
Waiting for daemon set "fluentd-elasticsearch" rollout to finish: 2 out of 4 new pods have been updated...
```

```
Waiting for daemon set "fluentd-elasticsearch" rollout to finish: 2 out of 4 new pods have been updated...
```

```
Waiting for daemon set "fluentd-elasticsearch" rollout to finish: 3 of 4 updated pods are available...
```

```
daemon set "fluentd-elasticsearch" successfully rolled out
```

1

需求来源

.....

2

用例解读

.....

3

操作演示

.....

4

架构设计

.....

5

课后习题

1

需求来源

.....

2

用例解读

.....

3

操作演示

.....

4

架构设计

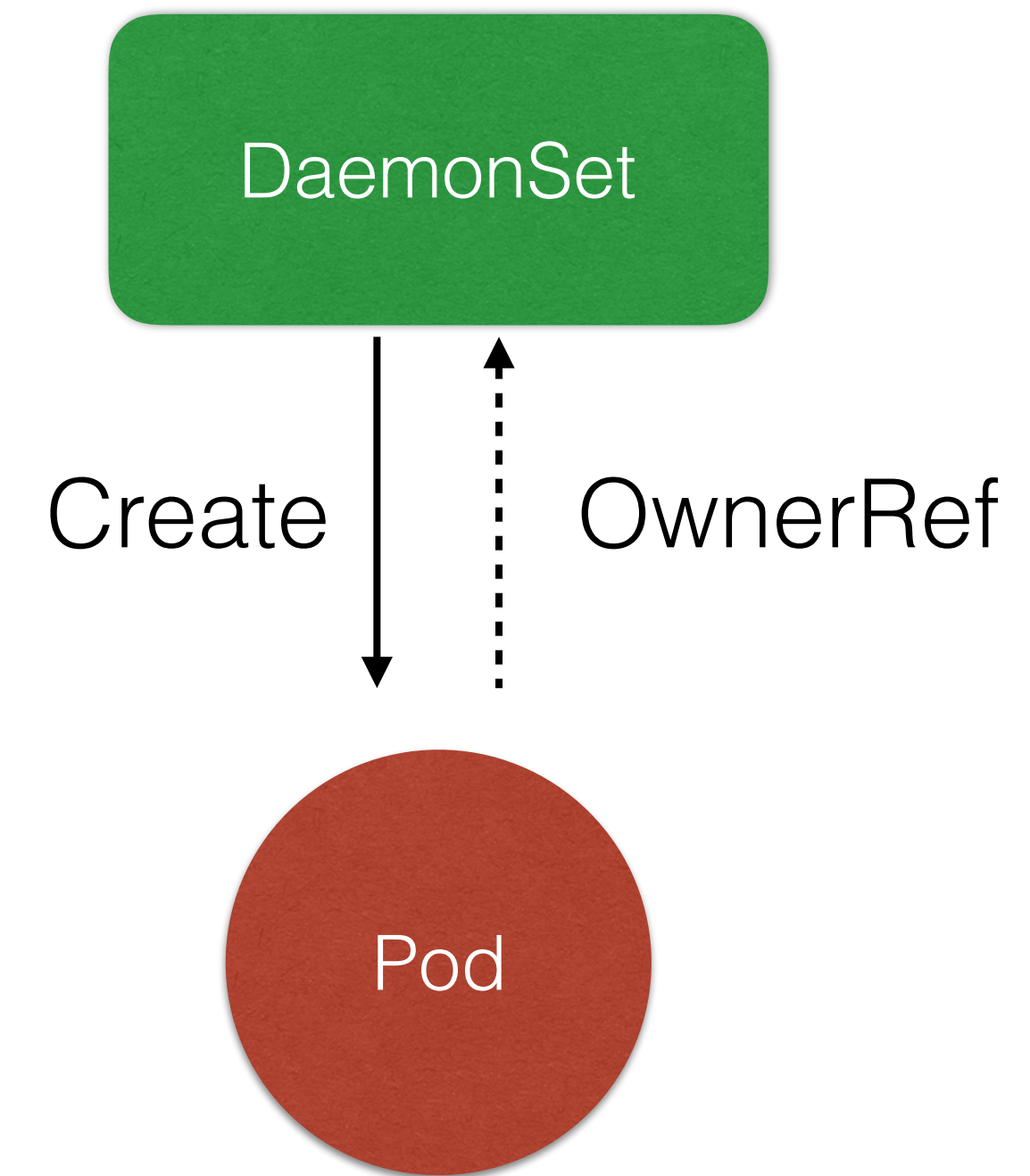
.....

5

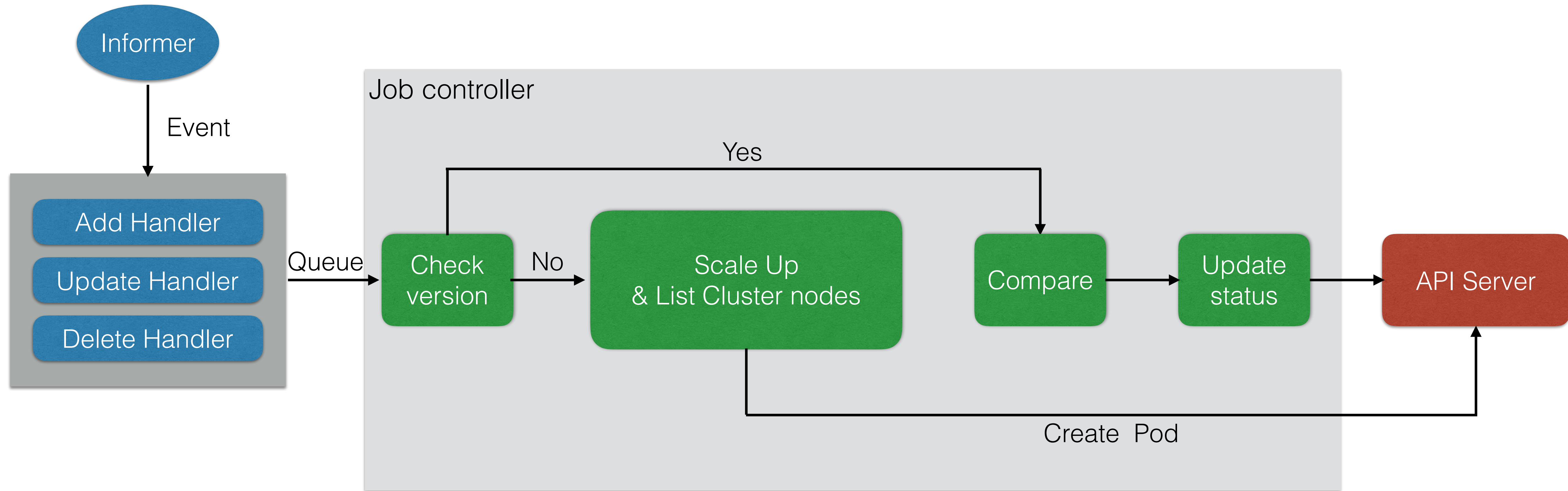
课后习题

管理模式

1. DaemonSet Controller 负责根据配置创建Pod
2. DaemonSet Controller 跟踪Job 状态，根据配置及时重试Pod或者继续创建
3. DaemonSet Controller 会自动添加affinity&label来跟踪对应的pod,并根据配置在每个节点或者适合的部分节点创建 Pod



Job 控制器





谢谢观看
THANK YOU

