

拉勾教育

— 互联网人实战大学 —

# 《Kubernetes 原理剖析与实战应用》

正范

— 拉勾教育出品 —

# 06 | 无状态应用：剖析 Kubernetes 业务副本及水平扩展底层原理

每一个 Pod 都是应用的一个实例

通常不会直接在 Kubernetes 中创建和运行单个 Pod

因为 Pod 的生命周期是短暂的，即 **“用后即焚”**



单独地用一个 Pod 来承载业务

**无法保证高可用、可伸缩、负载均衡等要求，而且 Pod 也无法“自愈”**

这时就需要在 Pod 之上做一层抽象

通过多个副本（Replica）来保证可用 Pod 的数量，避免业务不可用

# 有状态服务 VS 无状态服务

拉勾教育

— 互联网人实战大学 —

无状态服务

如打网络游戏

每次请求都包含了需要的所有信息  
每次请求都和之前的没有任何关系

业务的服务类型

有状态服务

如浏览网页

其请求是状态化的  
服务器端需要保存请求的相关信息  
这样每个请求都可以默认地  
使用之前的请求上下文

Kubernetes 中各个对象的 metadata 字段

都有 **label (标签)** 和 **annotation (注解)** 两个对象

# Kubernetes 中的无状态工作负载

拉勾教育

— 互联网人实战大学 —



label (标签)

主要用来标识一些有意义且和对象密切相关的信息

用来支持 labelSelector (标签选择器)

以及一些查询操作，还有选择对象



annotation (注解)

主要用来记录一些非识别的信息

并不用于标识和选择对象



```
$ kubectl get pod -l label1=value1,label2=value2 -n my-namespace
```

查询出 my-namespace 这个命名空间下面，带有标签 label1=value1 和 label2=value2 的 pod

label 中的键值对在匹配的时候是“且”的关系

# Kubernetes 中的无状态工作负载

拉勾教育

— 互联网人实战大学 —

## ReplicationController

ReplicationController 通常缩写为 “rc”

```
$ kubectl get rc -n my-namespace
```

# Kubernetes 中的无状态工作负载

拉勾教育

— 互联网人实战大学 —

## ReplicaSet

**ReplicaSet (简写为 rs)** 用来替代 ReplicaController

目前支持三种操作符：**in**、**notin** 和 **exists**

# Kubernetes 中的无状态工作负载

拉勾教育

— 互联网人实战大学 —

## ReplicaSet

可以用 **environment in (production, qa)**

匹配 label 中带有 environment=production 或 environment=qa 的 Pod

可以用 **tier notin (frontend,backend)**

匹配 label 中不带有 tier=frontend 或 tier=backend 的 Pod

# Kubernetes 中的无状态工作负载

拉勾教育

— 互联网人实战大学 —

ReplicaSet

```
kubectl get pods -l environment=production,tier=frontend
```

# Kubernetes 中的无状态工作负载

拉勾教育

— 互联网人实战大学 —

## ReplicaSet

```
kubectl get pods -l environment=production,tier=frontend
```

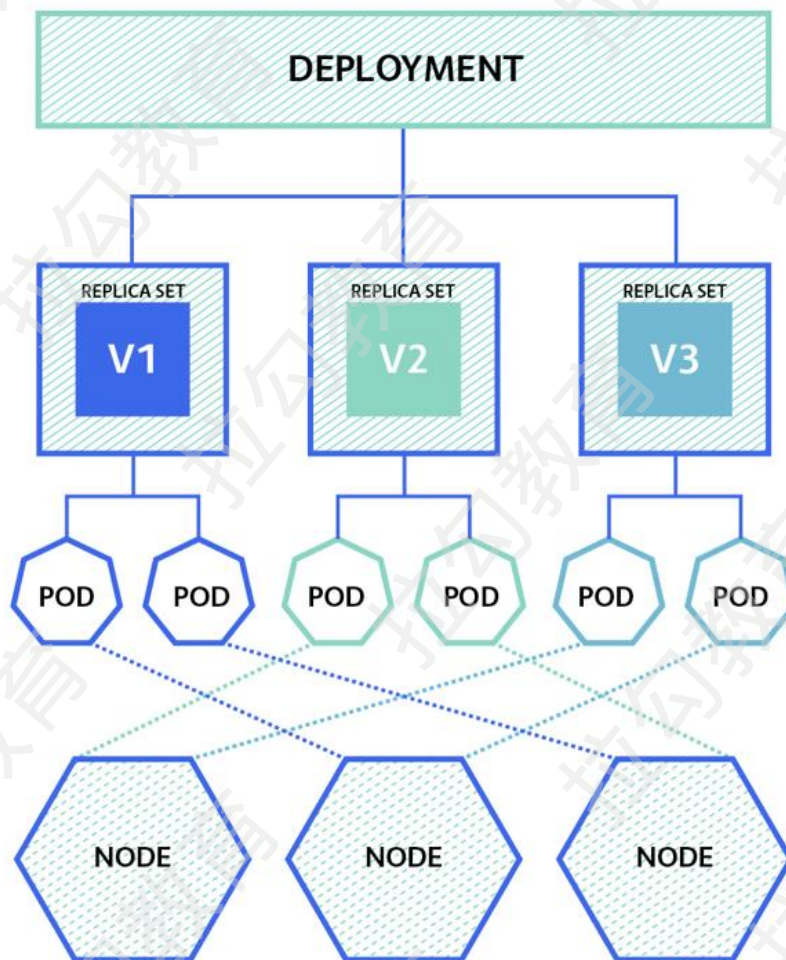
```
kubectl get pods -l 'environment in (production),tier in (frontend)'
```

# Kubernetes 中的无状态工作负载

拉勾教育

— 互联网人实战大学 —

## Deployment



# Kubernetes 中的无状态工作负载

拉勾教育

— 互联网人实战大学 —

## Deployment

```
apiVersion:
apps/v1
kind:
Deployment
metadata:
name:
nginx-deployment-demo
namespace: demo
labels:
app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
```



# Kubernetes 中的无状态工作负载

拉勾教育

— 互联网人实战大学 —

## Deployment

```
labels:  
  app: nginx  
spec:  
  replicas: 3  
  selector:  
    matchLabels:  
      app: nginx  
  template:  
    metadata:  
      labels:  
        app: nginx  
        version: v1  
    spec:  
      containers:  
        - name: nginx  
          image: nginx:1.14.2  
          ports:  
            - containerPort: 80
```

# Kubernetes 中的无状态工作负载

拉勾教育

— 互联网人实战大学 —

## Deployment

```
$ kubectl create ns demo
```

```
$ kubectl create -f deploy-demo.yaml
```

```
deployment.apps/nginx-deployment-demo created
```

# Kubernetes 中的无状态工作负载

拉勾教育

— 互联网人实战大学 —

## Deployment

```
$ kubectl get rs -n demo
```

NAME	DESIRED	CURRENT	READY	AGE
nginx-deployment-demo-5d65f98bd9	3	3	0	5s

# Kubernetes 中的无状态工作负载

拉勾教育

互联网人实战大学

## Deployment

```
$ kubectl get pod -n demo -l app=nginx,version=v1
```

NAME	READY	STATUS	RESTARTS	AGE
nginx-deployment-demo-5d65f98bd9-7w5gp	0/1	ContainerCreating	0	30s
nginx-deployment-demo-5d65f98bd9-d78fx	0/1	ContainerCreating	0	30s
nginx-deployment-demo-5d65f98bd9-ssstk	0/1	ContainerCreating	0	30s

```
$ kubectl get pod -n demo -l app=nginx,version=v1
```

NAME	READY	STATUS	RESTARTS	AGE
nginx-deployment-demo-5d65f98bd9-7w5gp	1/1	Running	0	63s
nginx-deployment-demo-5d65f98bd9-d78fx	1/1	Running	0	63s
nginx-deployment-demo-5d65f98bd9-ssstk	1/1	Running	0	63s

# Kubernetes 中的无状态工作负载

拉勾教育

— 互联网人实战大学 —

## Deployment

```
$ kubectl edit deploy nginx-deployment-demo -n demo
```

# Kubernetes 中的无状态工作负载

拉勾教育

— 互联网人实战大学 —

## Deployment

```
apiVersion:
  apps/v1
kind:
  Deployment
metadata:
  name:
    nginx-deployment-demo
  namespace: demo
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
```

# Kubernetes 中的无状态工作负载

拉勾教育

— 互联网人实战大学 —

## Deployment

```
labels:  
  app: nginx  
spec:  
  replicas: 3  
  selector:  
    matchLabels:  
      app: nginx  
  template:  
    metadata:  
      labels:  
        app: nginx  
        version: v2  
    spec:  
      containers:  
        - name: nginx  
          image: nginx:1.19.2  
          ports:  
            - containerPort: 80
```

# Kubernetes 中的无状态工作负载

拉勾教育

— 互联网人实战大学 —

## Deployment

```
$ kubectl apply -f deploy-demo.yaml
```

Warning:

kubectl apply should be used on resource created by either kubectl create  
--save-config or kubectl apply  
deployment.apps/nginx-deployment-demo configured



# Kubernetes 中的无状态工作负载

拉勾教育

— 互联网人实战大学 —

## Deployment

```
$ kubectl get rs -n demo
```

NAME	DESIRED	CURRENT	READY	AGE
nginx-deployment-demo-5d65f98bd9	3	3	3	4m10s
nginx-deployment-demo-7594578db7	1	1	0	3s

建议你使用默认的策略来保证可用性

# Kubernetes 中的无状态工作负载

拉勾教育

互联网人实战大学

## Deployment

```
$ kubectl get pod -n demo -l app=nginx,version=v1
```

NAME	READY	STATUS	RESTARTS	AGE
nginx-deployment-demo-5d65f98bd9-7w5gp	1/1	Running	0	4m15s
nginx-deployment-demo-5d65f98bd9-d78fx	1/1	Running	0	4m15s
nginx-deployment-demo-5d65f98bd9-ssstk	1/1	Running	0	4m15s

```
$ kubectl get pod -n demo -l app=nginx
```

NAME	READY	STATUS	RESTARTS	AGE
nginx-deployment-demo-5d65f98bd9-7w5gp	1/1	Running	0	4m22s
nginx-deployment-demo-5d65f98bd9-d78fx	1/1	Running	0	4m22s
nginx-deployment-demo-5d65f98bd9-ssstk	1/1	Running	0	4m22s
nginx-deployment-demo-7594578db7-zk8jq	0/1	ContainerCreating	0	15s

```
$ kubectl get pod -n demo -l app=nginx,version=v2
```

NAME	READY	STATUS	RESTARTS	AGE
------	-------	--------	----------	-----

# Kubernetes 中的无状态工作负载

拉勾教育

— 互联网人实战大学 —

## Deployment

RESTARTS AGE

Running 0 17s

Running 0 14s

Running 0 47s

AGE

0 5m5s

3 58s

app=nginx,version=v1

nginx-deployment-demo-7594578db7-4g4fk 1/1

nginx-deployment-demo-7594578db7-ftzmf 1/1

nginx-deployment-demo-7594578db7-zk8jq 1/1

\$ kubectl get rs -n demo

NAME	DESIRED	CURRENT	READY
------	---------	---------	-------

nginx-deployment-demo-5d65f98bd9	0	0	
----------------------------------	---	---	--

nginx-deployment-demo-7594578db7	3	3	
----------------------------------	---	---	--

\$ kubectl get pod -n demo -l

No resources found in demo namespace.

# Kubernetes 中的无状态工作负载

拉勾教育

— 互联网人实战大学 —

## Deployment

```
$ kubectl get pod -n demo -l app=nginx -w
```

有了 Deployment 这个高级对象

可以很方便地完成无状态服务的发布、更新升级，无须多余的人工参与

就能保证业务的高可用性

Next: 《07 | 有状态应用: Kubernetes 如何通过 StatefulSet 支持有状态应用? 》

# 拉勾教育

— 互联网人实战大学 —



关注拉勾「教育公众号」  
获取更多课程信息