

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 —

05 | Pod：最小调度单元的使用进阶及实践

Pod 是 Kubernetes 中原子化的部署单元

可以包含一个或多个容器

而且容器之间可以共享网络、存储资源



-
- Pod 里的某一个容器异常退出了怎么办？
 - 有没有“健康检查”方便你知道业务的真实运行情况，比如容器运行正常，但是业务不工作了？
 - 容器在启动或删除前后，如果需要做一些特殊处理怎么办？比如做一些清理工作
 - 如果容器所在节点宕机，重启后会对你的容器产生影响吗？
 -

Pod 的运行状态

拉勾教育

— 互联网人实战大学 —

```
apiVersion: v1 #指定当前描述文件遵循v1版本的Kubernetes API
kind: Pod #我们在描述一个pod
metadata:
  name: twocontainers #指定pod的名称
  namespace: default #指定当前描述的pod所在的命名空间
  labels: #指定pod标签
    app: twocontainers
  annotations: #指定pod注释
    version: v1
    releasedBy: david
    purpose: demo
spec:
  containers:
    - name: sise #容器的名称
      image: quay.io/openshiftlabs/simple-service:0.5.0 #创建容器所使用的镜像
      ports:
        - containerPort: 9876 #应用监听的端口
        - name: shell #容器的名称
      image: centos:7 #创建容器所使用的镜像
      command: #容器启动命令
        - "bin/bash"
        - "-c"
        - "sleep 10000"
```

Pod 的运行状态

拉勾教育

— 互联网人实战大学 —

```
$ kubectl get pod twocontainers -o=jsonpath='{.status.phase}'
```

Pending

Pod 的运行状态

拉勾教育

— 互联网人实战大学 —

```
$ kubectl get pod twocontainers
```

| NAME | READY | STATUS | RESTARTS | AGE |
|---------------|-------|-------------------|----------|-----|
| twocontainers | 0/2 | ContainerCreating | 0 | 13s |

Pod 的运行状态

拉勾教育

— 互联网人实战大学 —

```
$ kubectl describe pod twocontainers
Name:      twocontainers
Namespace: default
...
Status:    Pending
IP:
IPs:       <none>
Containers:
  size:
    Container ID:
    Image:        quay.io/openshiftlabs/simple-service:0.5.0
    ...
    State:        Waiting
    Reason:       ContainerCreating
    Ready:        False
    Restart Count: 0
  ...
  shell:
    Container ID:
    Image:        centos:7
    Image ID:
    ...
    State:        Waiting
    Reason:       ContainerCreating
```


Pod 的运行状态

拉勾教育

— 互联网人实战大学 —

```
Image: quay.io/openshiftlabs/simple-service:0.5.0
...
State:      Waiting
Reason:     ContainerCreating
Ready:      False
Restart Count: 0
...
shell:
Container ID:
Image:      centos:7
Image ID:
...
State:      Waiting
Reason:     ContainerCreating
Ready:      False
...
...
Events:
Type Reason Age From Message
----
Normal Scheduled <unknown> default-scheduler Successfully assigned
default/twocontainers to node-1
Normal Pulling 3m57s kubelet, node-1 Pulling image
"quay.io/openshiftlabs/simple-service:0.5.0"
```

Pod 的运行状态

拉勾教育

— 互联网人实战大学 —

```
Image:      quay.io/openshiftlabs/simple-service:0.5.0
...
State:      Waiting
Reason:     ContainerCreating
Ready:      False
Restart Count: 0
...
shell:
Container ID:
Image:      centos:7
Image ID:
...
State:      Waiting
Reason:     ContainerCreating
Ready:      False
...
Events:
Type Reason Age From Message
-----
Normal Scheduled <unknown> default-scheduler Successfully assigned
default/twocontainers to node-1
Normal Pulling 3m57s kubelet, node-1 Pulling image
"quay.io/openshiftlabs/simple-service:0.5.0"
```

处于 **Pending** 状态的 Pod

不外乎以下 2 个原因：

1. Pod 还未被调度
2. Pod 内的容器镜像在待运行的节点上不存在

需要从镜像中心拉取

Pod 的运行状态

拉勾教育

互联网人实战大学

```
$ kubectl get pod twocontainers -o=jsonpath='{.status.phase}'
```

Running

```
$ kubectl describe pod twocontainers
```

Name: twocontainers

Namespace: default

...

Start Time: Wed, 26 Aug 2020 16:49:11 +0800

...

Status: Running

...

Containers:

size:

Container ID:

docker://4dc8244a19e8766b151b36d986b9b3661f3bf05260aedd2b76dd5f0fcd6e637f

Image: quay.io/openshiftlabs/simple-service:0.5.0

Image ID: docker-

pullable://quay.io/openshiftlabs/simple-service@sha256:72bfe1acc54829c306dd6683f

e28089d222cf50a2df9d10c4e9d32974a591673

Pod 的运行状态

拉勾教育

— 互联网人实战大学 —

e28089d222cf50a2df9d10c4e9d32974a591673

...

State: Running

Started: Wed, 26 Aug 2020 17:00:52 +0800

Ready: True

...

shell:

Container ID:

docker://1b6137b4cef60d0309412f5cd8ba7f0ff743ee03c1112112f6aadd78f9981bbaa

Image: centos:7

Image ID: docker-

pullable://centos@sha256:19a79828ca2e505eaaee0ff38c2f3fd9901f4826737295157cc5212b7a372cd2b

...

State: Running

Started: Wed, 26 Aug 2020 17:01:46 +0800

Ready: True

Pod 的运行状态

拉勾教育

— 互联网人实战大学 —

```
Container ID:
docker://1b6137b4cef60d0309412f5cdba7f0ff743ee03c1112112f6aadd78f9981bbaa
Image:      centos:7
Image ID:   docker-
pullable://centos@sha256:19a79828ca2e505eae0ff38c2f3fd9901f4826737295157cc5
212b7a372cd2b
...
State:      Running
Started:    Wed, 26 Aug 2020 17:01:46 +0800
Ready:      True

Conditions:
Type        Status
Initialized True
Ready       True
ContainersReady True
PodScheduled True
```

Pod 的运行状态

拉勾教育

— 互联网人实战大学 —

```
$ kubectl get pod twocontainers
```

| NAME | READY | STATUS | RESTARTS | AGE |
|---------------|-------|---------|----------|-----|
| twocontainers | 2/2 | Running | 0 | 2m |

Pod 的运行状态

拉勾教育

— 互联网人实战大学 —

Succeeded

表示 Pod 内的所有容器
均成功运行结束
即正常退出，退出码为 0

Failed

表示 Pod 内的所有容器均运行终止
且至少有一个容器终止失败
一般这种情况
都是由于容器运行异常退出
或者被系统终止掉了

Unknown

一般是由于 Node 失联导致的
Pod 状态无法获取到

Kubernetes 中定义了如下三种重启策略

可以通过 **spec.restartPolicy** 字段在 Pod 定义中进行设置

Pod 的重启策略

拉勾教育

— 互联网人实战大学 —

Kubernetes 中定义了如下三种重启策略

可以通过 **spec.restartPolicy** 字段在 Pod 定义中进行设置

Always

表示一直重启，也是默认的重启策略
Kubelet 会定期查询容器的状态
一旦某个容器处于退出状态
就对其执行重启操作

OnFailure

表示只有在容器异常退出
即退出码不为 0 时
才会对其进行重启操作

Never

表示从不重启

Pod 的重启策略

拉勾教育

— 互联网人实战大学 —

比如某些 Java 进程启动速度非常慢

在容器启动阶段其实是无法提供服务的，虽然这个时候该容器是处于运行状态

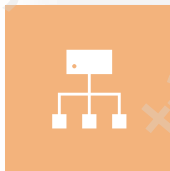
比如有些服务的进程发生阻塞

导致无法对外提供服务，这个时候容器对外还是显示为运行态

Kubernetes 中提供了一系列的健康检查

可以定制调用来帮助解决类似的问题

称之为 **Probe** (探针)



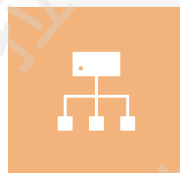
livenessProbe

可以用来探测容器是否真的在“运行”，即“探活”



readinessProbe

用于指示容器是否可以对外提供正常的服务请求，即“就绪”



startupProbe

可以用于判断容器是否已经启动好

Pod 中的健康检查

拉勾教育

— 互联网人实战大学 —

ExecAction

可以在容器内执行 shell 脚本

HTTPGetAction

方便对指定的端口和 IP 地址

执行 HTTP Get 请求

TCPSocketAction

可以对指定端口进行 TCP 检查

Pod 中的健康检查

拉勾教育

— 互联网人实战大学 —

```
apiVersion: v1
kind: Pod
metadata:
  name: probe-demo
  namespace: demo
spec:
  containers:
    - name: sise
      image: quay.io/openshiftlabs/simple-service:0.5.0
      ports:
        - containerPort: 9876
      readinessProbe:
        tcpSocket:
          port: 9876
        periodSeconds: 10
      livenessProbe:
        periodSeconds: 5
```

Pod 中的健康检查

拉勾教育

— 互联网人实战大学 —

```
ports:
- containerPort: 9876
readinessProbe:
  tcpSocket:
    port: 9876
  periodSeconds: 10
livenessProbe:
  periodSeconds: 5
  httpGet:
    path: /health
    port: 9876
startupProbe:
  httpGet:
    path: /health
    port: 9876
  failureThreshold: 3
  periodSeconds: 2
```

Pod 中的健康检查

拉勾教育

— 互联网人实战大学 —

```
ports:
- containerPort: 9876
readinessProbe:
  tcpSocket:
    port: 9876
  periodSeconds: 10
livenessProbe:
  periodSeconds: 5
  httpGet:
    path: /health
    port: 9876
  startupProbe:
    httpGet:
      path: /health
      port: 9876
    failureThreshold: 3
    periodSeconds: 2
```


容器生命周期内的 hook

拉勾教育

— 互联网人实战大学 —



PreStop 则在容器被终止之前被执行
是一种阻塞式的方式



PostStart 可以在容器启动之后就执行

容器生命周期内的 hook

拉勾教育

— 互联网人实战大学 —

```
apiVersion: v1
kind: Pod
metadata:
  name: lifecycle-demo
  namespace: demo
spec:
  containers:
  - name: lifecycle-demo-container
    image: nginx:1.19
    lifecycle:
      postStart:
        exec:
          command: ["/bin/sh", "-c", "echo Hello from the
postStart handler > /usr/share/message"]
      preStop:
        exec:
          command: ["/usr/sbin/nginx","-s","quit"]
```

通常用来做一些初始化工作

比如环境检测、OSS 文件下载、工具安装等等

- 总是运行到完成，可以理解为一次性的任务，不可以运行常驻型任务，因为会 block 应用容器的启动运行
- 顺序启动执行，下一个的 init 容器都必须在上一个运行成功后才可以启动
- 禁止使用 readiness/liveness 探针，可以使用 Pod 定义的 **activeDeadlineSeconds**

其中包含了 Init Container 的启动时间

- 禁止使用 lifecycle hook



```
apiVersion: v1
kind: Pod
metadata:
  name: myapp-pod
  namespace: demo
  labels:
    app: myapp
spec:
  containers:
    - name: myapp-container
      image: busybox:1.31
      command: [ 'sh', '-c', 'echo The app is running! && sleep 3600' ]
  initContainers:
    - name: init-myservice
      image: busybox:1.31
      command: [ 'sh', '-c', 'until nslookup myservice; do echo waiting for myservice; sleep 2; done;' ]
    - name: init-mydb
      image: busybox:1.31
      command: [ 'sh', '-c', 'until nslookup mydb; do echo waiting for mydb; sleep 2; done;' ]
```

Kubernetes 内部最核心的对象之一，**Pod** 承载了太多的功能

为了增加可扩展、可配置性

Kubernetes 增加了各种 Probe、Hook 等，以此方便使用者进行接入配置

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

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