Pod健康检查

前言

- 本章节主要介绍了Kubernetes中如何使用探针进行Pod健康检查。

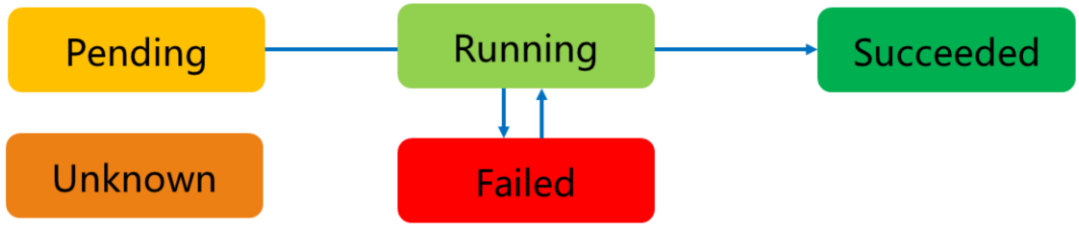
目标

- 学完本课程后，您将能够：
  - 掌握存活探针的概念和使用方式
  - 掌握就绪探针的概念和使用方式

目录

Pod状态

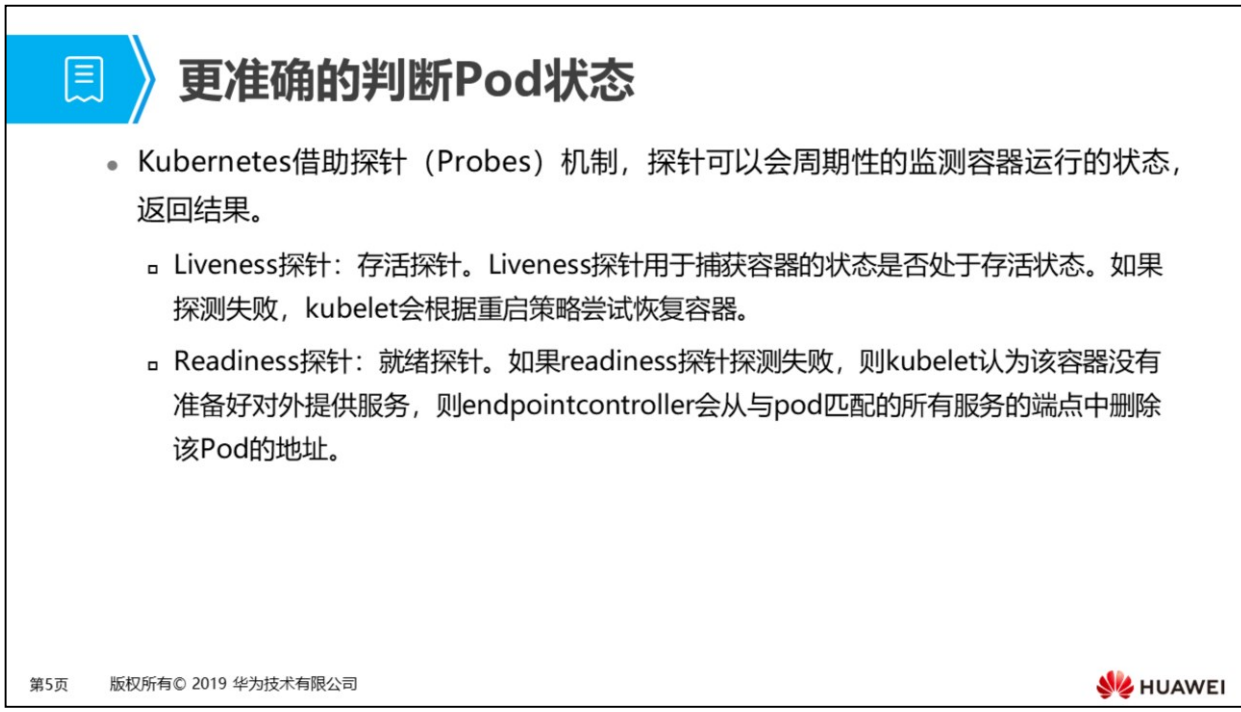
- Pod的状态信息在PodStatus中定义，其中有一个phase字段，就是我们熟悉的以下一些状态。



- 在何种状态下的Pod可以正常提供服务？

HUAWEI

- Running状态下的Pod是否表示该应用是正常的？
- 运行中（Running）：该Pod已经绑定到了一个节点上，Pod中所有的容器都已被创建。至少有一个容器正在运行，或者正处于启动或重启状态。

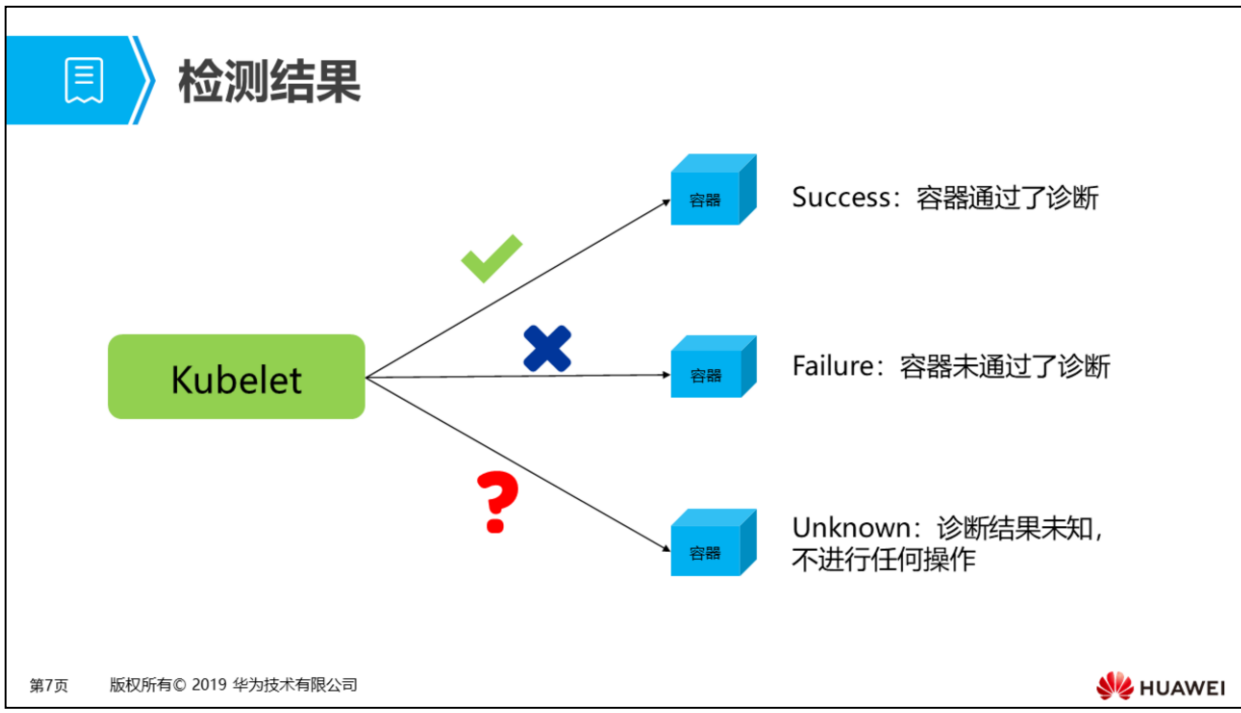


- 如果您希望在探测失败时杀死并重新启动Container，则指定LivenessProbe，并指定restartPolicy为Always或OnFailure。
- 如果您只想在探测成功时开始向Pod发送流量，请指定Readiness Probe。在这种情况下，Readiness Probe可能与Liveness probe相同，但规范中存在Readiness Probe意味着Pod将在不接收任何流量的情况下启动，并且仅在探测成功后才开始接收流量。
- 如果Container需要在启动期间处理大型数据，配置文件或迁移，请指定Readiness probe。
- 如果您希望Container能够自行维护，您可以指定一个Readiness probe，该探针检查特定于就绪状态的端点，该端点与Liveness probe不同。

容器探针

- Kubelet可以周期性的执行Container的诊断。为了执行诊断，kubelet调用Container实现的Handler，有三种Handler类型：
  - ExecAction：在容器内执行指定命令，如果命令退出时返回码为0（表示命令成功执行了），则认为诊断成功。
  - TCPSocketAction：对指定端口上的容器的IP地址进行TCP检查。如果端口打开，则认为诊断成功。
  - HTTPGetAction： 对指定端口和路径上的容器IP地址执行HTTP Get请求。如果响应的状态码≥200且<400，则认为诊断成功。

- 如果状态码返回了404，那就表示诊断不成功。



- 以就绪探针为例，如果未通过诊断，则根据重启策略处理。

目录

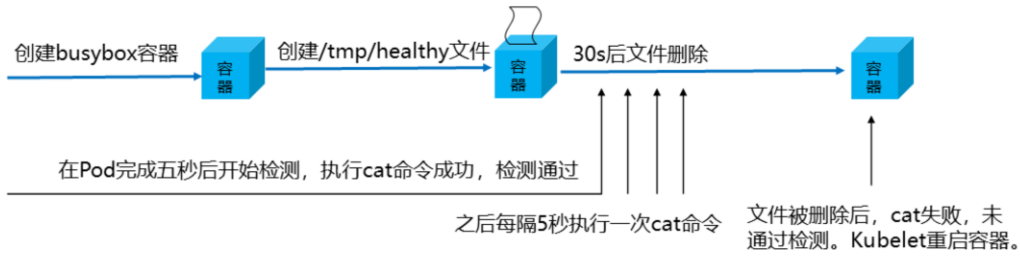
存活探针案例

- 本案例采用execaction模式的存活探针。
- livenessProbe字段详细定义了存活探针，包括：
  - Handler采用exec
  - 使用方式是运行cat /tmp/healthy命令
  - 探测延迟和探测周期是5秒钟。

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    test: liveness
  name: liveness-exec
spec:
  containers:
  - name: liveness
    args:
    - /bin/sh
    - -c
    - touch /tmp/healthy; sleep 30;
 rm -rf /tmp/healthy; sleep 600
    image: busybox
    livenessProbe:
      exec:
        command:
        - cat
        - /tmp/healthy
      initialDelaySeconds: 5
      periodSeconds: 5
```


- - touch行，指定容器在创建后往/tmp目录写入一个healthy文件。用于探针检测。

Liveness探针流程



HUAWEI

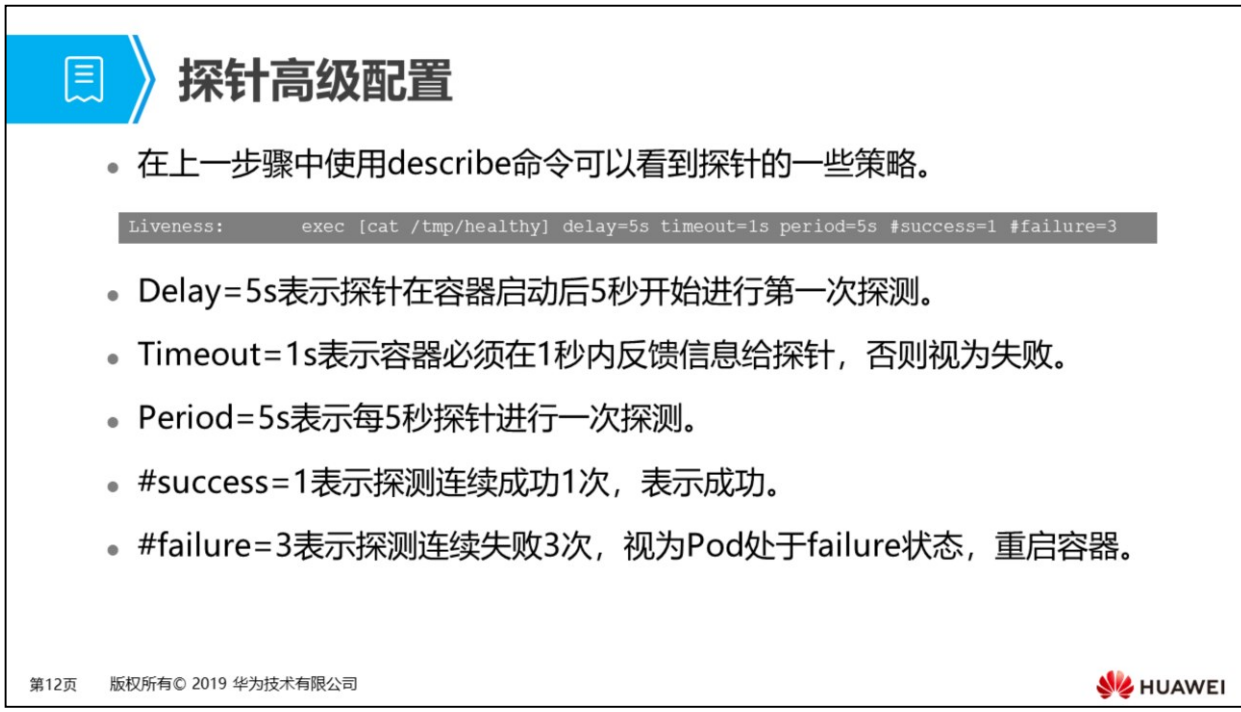
- 使用describe命令查看pod的详细信息

查看存活探针信息

- 使用describe命令查看pod信息

```
Events:
  Type     Reason     Age                From                Message
  ----     ------     ----               ----                -------
  Normal   Scheduled  92s                default-scheduler   Successfully assigned
default/liveness-exec to k8s-node2
  Normal   Killing    47s                kubelet, k8s-node2  Container liveness failed
liveness probe, will be restarted
  Warning  Unhealthy  47s (x3 over 57s)  kubelet, k8s-node2  Liveness probe failed: cat:
can't open '/tmp/healthy': No such file or directory
  Normal   Pulled     17s (x2 over 92s)  kubelet, k8s-node2  Successfully pulled image
"busybox"
  Normal   Created    17s (x2 over 92s)  kubelet, k8s-node2  Created container liveness
  Normal   Pulling    17s (x2 over 92s)  kubelet, k8s-node2  Pulling image "busybox"
  Normal   Started    16s (x2 over 91s)  kubelet, k8s-node2  Started container liveness
```


- 在默认情况下三次探测失败，则认为容器不存活，一次探测成功，则认为容器存活。这个默认次数可以改变。



- 这些参数可以在yaml文件中probe部分进行设置。

探针高级配置

- 高级配置参数可以在配置参数时指定，以下为配置样例。实现的功能与之前配置的探针一致。

```
livenessProbe:
      exec:
        command:
        - cat
        - /tmp/healthy
      initialDelaySeconds: 5
      periodSeconds: 5
      timeoutSeconds: 1
      successThreshold: 1
      successThreshold: 3
```

存活探针 - HTTP

- HTTP方式的存活探针，通过get方法定期向容器发送http请求。方法中定义了请求路径、端口、请求头等信息。
- 由于探针仅在返回码≥200，小于400的情况下返回正常，10秒后探针检测失败，kubelet会重启容器。

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    test: liveness
  name: liveness-http
spec:
  containers:
  - name: liveness
    image: mirrorgooglecontainers/liveness
    args:
    - /server
    livenessProbe:
      httpGet:
        path: /healthz
        port: 8080
        httpHeaders:
        - name: X-Custom-Header
          value: Awesome
      initialDelaySeconds: 3
      periodSeconds: 3
```


- Liveness容器会在最开始的10秒内处于存活状态，返回状态码200，之后将返回500的返回码。

存活探针 - TCP

- TCP探针检测能否建立连接。实验中部署一个telnet服务，探针探测23端口。
- TCP探针参数与HTTP探针相似。

```
apiVersion: v1
kind: Pod
metadata:
  name: ubuntu
  labels:
    app: ubuntu
spec:
  containers:
  - name: ubuntu
    image: ubuntu
    args:
    - /bin/sh
    - -c
    - apt-get update && apt-get -y install
openbsd-inetd telnetd &&
/etc/init.d/openbsd-inetd start; sleep 30000
    livenessProbe:
      tcpSocket:
        port: 23
      initialDelaySeconds: 60
      periodSeconds: 20
```


- 当探针检测到端口无法访问时，返回失败。

目录

HUAWEI

就绪探针

- Pod处于存活状态并不意味着可以提供服务，创建完成后通常需要进行诸如准备数据、安装和运行程序等步骤，才能对外提供服务。
- Liveness探针指示Pod是否处于存活状态，Readiness探针则可指示容器是否已经一切准备就绪，可以对外提供服务。

- 存活探针和就绪探针的实现方式类似，但处理方式不同。

存活探针和就绪探针对比

- 就绪探针与存活探针一致，可以使用ExecAction，TCPSocketAction，HTTPGetAction三种方法。
- 就绪探针用于检测和显示Pod是否已经准备好对外提供业务。在实际使用场景中，就绪探针需要和业务绑定。

	就绪探针	存活探针
当Pod未通过检测	等待	杀死Pod，重启一个新Pod
服务	如果检测失败，则从endpoint中移除pod	Endpoint自动更新新pod信息
作用	Pod是否准备好提供服务	Pod是否存活

- 两者主要区别是未通过检测时的处理方式。

创建HTTP服务

- 创建http的deployment和service，并在其中加入就绪探针，探测是否存在index.html文件。

```
apiVersion: v1
kind: Service
metadata:
  name: httpd-svc
spec:
  selector:
    app: httpd
  ports:
  - protocol: TCP
    port: 8080
    targetPort: 80
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: httpd-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: httpd
  template:
    metadata:
      labels:
        app: httpd
    spec:
      containers:
      - name: httpd
        image: httpd
        ports:
        - containerPort: 80
        readinessProbe:
          exec:
            command:
            - cat
            - /usr/local/apache2/htdocs/index.html
          initialDelaySeconds: 5
          periodSeconds: 5
```


- 本页和后两页为一个完整的探测实验。

查看Endpoint状态

- 查看服务状态，endpoints如下：

```
Endpoints:          10.244.1.15:80,10.244.2.22:80,10.244.2.23:80
```

- Pod状态如下：

```
NAME                                 READY   STATUS    RESTARTS   AGE
httpd-deployment-859778b7b6-7wcbt    1/1     Running   0          9m42s
httpd-deployment-859778b7b6-p62mw    1/1     Running   0          9m42s
httpd-deployment-859778b7b6-zvpsg    1/1     Running   0          9m42s
```

- 现在进入第一个容器，删除其中的index.html文件

```
[root@k8s-master probe]# kubectl exec -it httpd-deployment-859778b7b6-7wcbt /bin/sh
# rm /usr/local/apache2/htdocs/index.html
```


- 本页模拟了一个故障，使其不通过探针检测

查看故障后状态

- 查看服务状态，endpoints如下，其中一个Pod的端口信息已被移除endpoint。

```
Endpoints:          10.244.1.15:80,10.244.2.23:80
```

- Pod状态如下，可以看到第一个Pod处于notReady状态。

```
NAME                                READY   STATUS    RESTARTS   AGE
httpd-deployment-859778b7b6-7wcbt   0/1     notReady  0          15m
httpd-deployment-859778b7b6-p62mw   1/1     Running   0          15m
httpd-deployment-859778b7b6-zvpsg   1/1     Running   0          15m
```

- 通过查看该Pod的记录，可以看到没有通过就绪探针。

```
Warning  Unhealthy  102s (x24 over 3m37s)  kubelet, k8s-node1  Readiness probe
failed: cat: /usr/local/apache2/htdocs/index.html: No such file or directory
```

如何修复这个Pod呢?

- 重新恢复文件是一种修复思路，是否有更便捷的方式呢?

实验&实训任务

- 实验任务
  - 请按照实验手册2.8章节完成健康检查相关实验，包括：
    - 使用存活探针
    - 使用就绪探针
- 实训任务
  - 请灵活使用本章节课程及实验手册中学到的知识，按照实验手册2.8.3章节完成健康检查实训任务。

本章总结

- 本章介绍了两种探针：就绪探针和存活探针的使用方式和特性。


谢 谢
www.huawei.com