

Kubernetes 系列（三十二）Pod 健康检查 -Liveness and Readiness 探测

在这个博客中，我们将探索如何使用 **Liveness** 和 **Readiness** 检查 Pod 的健康

准备工作：

1. 功能齐备的 kubernetes 集群
2. kubectl 命令行工具，可以运行指令部署应用到 kubernetes 中

大体过程：

1. 介绍 Probes/Health Check
2. 配置 Pod Liveness Probe 并且没有 Restart Policy
3. 创建一个 Pod 具备 Liveness 探测以及 Restart 策略
4. 创建一个 Pod 具备 Readiness 探测

Probes/Health 检查

- 可以将其配置为检查在 Pod 中运行的容器的状态。
- 用于确定容器是否正在运行或准备好接收请求

一个 probe/health 检查可以返回如下结果:

Success : 容器通过健康检查.

Failure : 容器未通过健康检查.

Unknown : 未知原因健康检查失败.

探测类型

1. Liveness 探测

- 用于确定特定容器是否正在运行.
- 如果一个容器的 Liveness 探测失败，控制器将根据为该容器配置的重启策略尝试在同一节点上重启该容器.

重启策略

我们可以在pod中定义 restart Policy 来指示控制器重启pod所需的条件:

默认值是 `always`

`Restart Policy` 有如下值:

- **Always** : 当 Pod 终止时，总是重新启动它.
- **OnFailure** : 只有当 Pod 因失败而终止时才重新启动.
- **Never** : 从不在 Pod 终止后重新启动它.

Pod 配置 Liveness 探测且未设置Restart Policy

在这个yaml文件中，我们将定义liveness probe 且不设置 restart policy。

如果我们不指定重启策略，那么缺省情况下它是 `always`

```
kind: Pod
apiVersion: v1
metadata:
  name: liveness-probe
spec:
  containers:
    - name: ubuntu-container
      image: ubuntu
      command:
        - /bin/bash
        - -ec
        - touch /tmp/live; sleep 30; rm /tmp/live; sleep 600
      livenessProbe:
        exec:
          command:
            - cat
            - /tmp/live
          initialDelaySeconds: 5
          periodSeconds: 5
```

Pod 配置

- 使用 ubuntu 镜像创建一个容器
- 当容器启动时，它将创建文件 `/tmp/live` 并且睡眠30 秒钟，最后删除文件 `/tmp/live`
- 这意味着该文件只在30秒内可用，之后在容器中就不再可用
- 在 liveness Probe 配置中 — 它将尝试每隔 5秒钟找一下文件 `/tmp/live`，第一次尝试延迟为5秒

`initialDelaySeconds` : 控制器在启动探测前等待的秒数

`periodSeconds` : 周期重复探测的秒数

创建 pod

```
kubectl create -f liveness-probe.yaml
kubectl describe pod liveness-probe
```

你可以看到 `liveness-probe` 成功，因为 command 执行是成功的

```
node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type    Reason            Age   From                  Message
  ----    -
  Normal  Scheduled         11s   default-scheduler     Successfully assigned kube-public/liveness-probe to gke-my-k8s-cluster-default-pool-5938de9e-hlbd
  Normal  Pulling           10s   kubelet, gke-my-k8s-cluster-default-pool-5938de9e-hlbd   Pulling image "ubuntu"
  Normal  Pulled            8s    kubelet, gke-my-k8s-cluster-default-pool-5938de9e-hlbd   Successfully pulled image "ubuntu"
  Normal  Created           7s    kubelet, gke-my-k8s-cluster-default-pool-5938de9e-hlbd   Created container ubuntu-container
  Normal  Started           7s    kubelet, gke-my-k8s-cluster-default-pool-5938de9e-hlbd   Started container ubuntu-container
```

现在等待30秒，然后运行以下命令

```
kubectl describe pod liveness-probe
```

```
node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type    Reason            Age   From                  Message
  ----    -
  Normal  Scheduled         44s   default-scheduler     Successfully assigned kube-public/liveness-probe to gke-my-k8s-cluster-default-pool-9d273e9f-82hc
  Normal  Pulling           43s   kubelet, gke-my-k8s-cluster-default-pool-9d273e9f-82hc   Pulling image "ubuntu"
  Normal  Pulled            43s   kubelet, gke-my-k8s-cluster-default-pool-9d273e9f-82hc   Successfully pulled image "ubuntu"
  Normal  Created           43s   kubelet, gke-my-k8s-cluster-default-pool-9d273e9f-82hc   Created container ubuntu-container
  Normal  Started           43s   kubelet, gke-my-k8s-cluster-default-pool-9d273e9f-82hc   Started container ubuntu-container
  Warning Unhealthy         4s (x2 over 9s) kubelet, gke-my-k8s-cluster-default-pool-9d273e9f-82hc   Liveness probe failed: cat: /exp/110: No such file or directory
```

现在您可以看到，由于默认的Restart策略，容器正在一次又一次地重新启动

```
C:\gitcode\kubernetes-sample-deployment\pods>kubectl get pod liveness-probe
NAME          READY   STATUS    RESTARTS   AGE
liveness-probe 1/1     Running   1          2m14s
```

```
C:\gitcode\kubernetes-sample-deployment\pods>kubectl get pod liveness-probe
NAME          READY   STATUS    RESTARTS   AGE
liveness-probe 1/1     Running   6          9m10s
```

创建一个 Pod 配置Liveness Probe 且设置 Restart Policy

现在我们将使用相同的pod配置，但Restart策略为 `Never`

```

kind: Pod
apiVersion: v1
metadata:
  name: liveness-probe-never-restart
spec:
  restartPolicy: Never
  containers:
  - name: ubuntu-container
    image: ubuntu
    command:
    - /bin/bash
    - -ec
    - touch /tmp/live; sleep 30; rm /tmp/live; sleep 100
    livenessProbe:
      exec:
        command:
        - cat
        - /tmp/live
      initialDelaySeconds: 5
      periodSeconds: 5

```

创建pod

```
kubectl create -f liveness-probe-with-restart-policy.yaml
```

等待一分钟，然后运行以下命令

```
kubectl describe pod liveness-probe-never-restart
```

```

Normal      Scheduled      53s      default-scheduler      Successfully assigned kube-public/liveness-probe-never-restart to gke-my-k8s-cluster-default-pool-9d273e9f-g4cj
Normal      Pulling        53s      kubelet, gke-my-k8s-cluster-default-pool-9d273e9f-g4cj      Pulling image "ubuntu"
Normal      Pulled         50s      kubelet, gke-my-k8s-cluster-default-pool-9d273e9f-g4cj      Successfully pulled image "ubuntu"
Normal      Created        49s      kubelet, gke-my-k8s-cluster-default-pool-9d273e9f-g4cj      Created container ubuntu-container
Normal      Started        49s      kubelet, gke-my-k8s-cluster-default-pool-9d273e9f-g4cj      Started container ubuntu-container
Warning     Unhealthy      5s (x3 over 15s) kubelet, gke-my-k8s-cluster-default-pool-9d273e9f-g4cj      Liveness probe failed: Get http://localhost:8080: dial tcp 127.0.0.1:8080: connect: connection refused
Normal      Killing        5s      kubelet, gke-my-k8s-cluster-default-pool-9d273e9f-g4cj      Stopping container ubuntu-container

```

```

C:\gitcode\kubernetes-sample-deployment\pods>kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
command-pod                         1/1     Running   0           3h19m
liveness-probe                       0/1     CrashLoopBackOff   7           3h19m
liveness-probe-never-restart         0/1     Error     0           84s

```

从输出中你可以看到控制器杀死了容器，并且从未尝试重新启动 Pod。

2. Readiness 探测

- 用于确定特定容器是否准备好接收请求。
- 例如一个为web应用服务的容器，就绪意味着容器已经加载了所有的静态资源，数据库连接，启动了web服务器，并打开了一个端口开始服务请求

如果容器的 readiness 探测失败会发生什么？

Kubernetes 控制器将确保 pod 不接收任何请求。

Readiness 探测状态

Failure — 是默认状态，直到readiness 探测成功。

Success- 只有在准备探测返回Success状态后，容器才会开始接收请求。

如果没有配置 readiness 探测，容器将在启动时立即开始接收请求，换句话说 service 对象接受的请求不会转发到该 pod。

创建一个 Pod 配置 Readiness 探测

在这个yaml文件中，我们将定义Readiness Probe

```
kind: Pod
apiVersion: v1
metadata:
  name: readiness-probe
spec:
  containers:
    - name: ubuntu-container
      image: ubuntu
      command:
        - /bin/bash
        - -ec
        - sleep 30; touch /tmp/ready; sleep 600
      readinessProbe:
        exec:
          command:
            - cat
            - /tmp/ready
          initialDelaySeconds: 10
          periodSeconds: 5
```

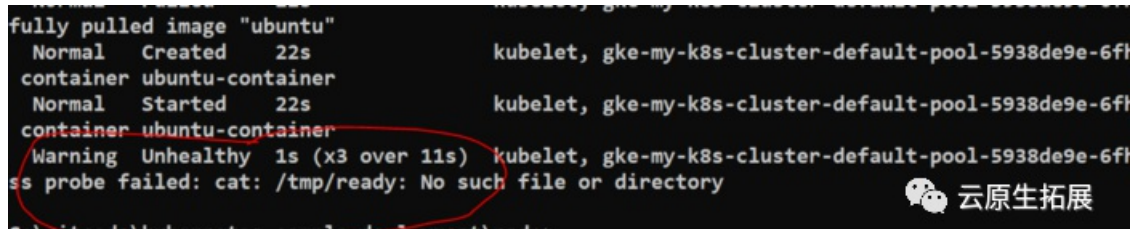
Pod 配置

- 使用 ubuntu 镜像创建容器
- 当容器启动后，将会睡眠 30 秒钟，然后创建文件 `/tmp/ready`
- 这意味着文件在30秒后才可用
- readiness 配置 — 它将尝试每5秒查找一次文件，初始延迟为10秒

创建 pod

```
kubectl create -f readiness-probe.yaml
kubectl get pod readiness-probe
kubectl describe pod readiness-probe
```

```
fully pulled image "ubuntu"
Normal   Created    22s                kubelet, gke-my-k8s-cluster-default-pool-5938de9e-6ft
container ubuntu-container
Normal   Started    22s                kubelet, gke-my-k8s-cluster-default-pool-5938de9e-6ft
container ubuntu-container
Warning  Unhealthy  1s (x3 over 11s)   kubelet, gke-my-k8s-cluster-default-pool-5938de9e-6ft
ss probe failed: cat: /tmp/ready: No such file or directory
```



云原生拓展

等待一分钟，然后运行以下命令，您应该能够看到成功消息

```
kubectl describe pod readiness-probe
```

使用 Probes 时的最佳实践

Liveness Probes

`initialDelaySeconds` : 应该尽量大于应用程序启动时间，这样容器就不会被卡在重启循环中

Readiness Probes

`initialDelaySeconds` : 可以是很小的，因为我们想在容器准备好后就能让请求进入 Pod

欢迎关注我的公众号“云原生拓展”，原创技术文章第一时间推送。