## 零宕机知识—>健康检查

# 1.3 pod探针

StartupProbe：k8s1.16版本后新加的探测方式，用于判断容器内应用程序是否已经启动

如果配置了StartupProbe，会禁用其他Proe探针，直到他成功为止，成功后将不再进行探测（对于启动时间长的pod需要）。

LivenessProbe: 用于探测容器是否运行，如果探测失败，Kubelet会根据配置的重启策略进行相应处理。若是没有配置改探针，默认就是success。--决定pod会不会重启 eg: path: /health port:8080

ReadinessProbe: 一般用于探测容器内的程序是否健康，它的返回值如果为success，那就代表这个容器已经完成启动，并且程序已经是可以接受流量的状态。eg: path: /ready port:8081

# 1.4 pod探针的检测方式

ExecAction: 在容器内执行一个命令，如果返回值是0，则认为健康

TCPSocketAction: 通过TCP连接检查容器内的端口是否是通的，如果是通的就认为容器健康:类似 netstat-lntp telnet 127.0.0.1 2379（可能程序假死，telnet和curl正常，但是请求逻辑端口不工作，不可靠）

**HTTPGetAction: 通过应用程序暴露的API地址来检查程序是否是正常的，如果状态码为200-400之间，则认为容器健康（最常用）**

StartupProbe引入必要性：防止类似LivenessProbe 0但是StartupProbe非0，不能杀死，但是又进入不了流量

# 1.5 探针检查参数配置

initialDelaySeconds: 60 ##初始化时间—等60后才去对程序做监控检查

timeoutSeconds:2 #超时时间---建议1-2s，一般接口反应是毫秒级

periodSecond：5 #检查间隔

successThreshold：1 #检查成功为1次表示就绪

failureThreshold：2 #检测失败2次表示未就绪—1次可能由于网络波动等原因造成误判

#kubectl describe po nginx ##检查pod的状态

#kubectl get po –oyaml ##查看创建该pod的yaml文件

#kubectl –exec –it nginx –sh ##it打开控制台，nginx是pod名称

#kubectl get po –owide ##显示ip

验证：#curl 10.244.32.135（通过pod的ip可以访问到nginx的页面）

#kubectl replace –f pod.yaml##根据yaml文件替换相应的资源

# 1.6 pod退出

Pod退出时的检查

lifecycle:

prestep:###容器被杀死前执行的命令

执行PreStop命令，如果指令时间是90s，需修改配置terminationGracePeriodSeconds大于90s,否则指令不生效

pod退出流程

Endppint删除该pod的ip地址

用户执行删除操作

Pod变成terminating

#kubectl get event ###查看事件，可作为监控

# vim pod.yaml

terminationGracePeriodSeconds:30 #####作用于所有的containers，位置在其上

containers:

- name:

lifecycle:

prestop:

exec:

command:

- sh

- -c

- sleep 90

#time kubectl delete po nginx ###可以查看执行时间