[1. Liveness Probe](https://note.youdao.com/ynoteshare1/iframe.html" \l "9130-1576327251487)

[1.1 liveness command](https://note.youdao.com/ynoteshare1/iframe.html#2255-1576327120608)

[1.2 liveness HTTP request](https://note.youdao.com/ynoteshare1/iframe.html#7528-1576327098009)

[1.3 TCP liveness probe](https://note.youdao.com/ynoteshare1/iframe.html#5174-1576327098012)

[2. readinessProbe](https://note.youdao.com/ynoteshare1/iframe.html#4495-1576327098013)

[2.1 readinessProbe + livenessProbe+Service](https://note.youdao.com/ynoteshare1/iframe.html#7019-1576327177794)

**1. Liveness Probe**

**1.1 liveness command**

<https://kubernetes.io/zh/docs/tasks/configure-pod-container/configure-liveness-readiness-startup-probes/>

docker search liveness

cat << EOF > pod-liveness-command.yaml

apiVersion: v1

kind: Pod

metadata:

labels:

test: liveness

name: liveness-exec

spec:

containers:

- name: liveness

image: radial/busyboxplus

imagePullPolicy: IfNotPresent

args:

- /bin/sh

- -c

- touch /tmp/healthy; sleep 60; rm -rf /tmp/healthy; sleep 600

livenessProbe:

exec:

command:

- cat

- /tmp/healthy

initialDelaySeconds: 5

periodSeconds: 5

EOF

[root@master01 damon]# kubectl apply -f pod-liveness-command.yaml

[root@master01 damon]# watch -n1 kubectl get pod //通过查看发现liveness-exec的RESTARTS 在30秒后由于检测到不健康一直在重启

NAME READY STATUS RESTARTS AGE

liveness-exec 1/1 Running 0 3s

nginx-deployment-676cc869-bnnvd 1/1 Running 0 30m

nginx-deployment-676cc869-ft8bw 1/1 Running 0 26m

[root@master01 damon]# kubectl get event // 查看Kubernetes事件

**1.2 liveness HTTP request**

cat << EOF > pod-liveness-http.yaml

apiVersion: v1

kind: Pod

metadata:

labels:

test: liveness

name: liveness-http

spec:

containers:

- name: liveness

image: seedoflife/liveness

imagePullPolicy: IfNotPresent

args:

- /server

livenessProbe:

httpGet:

path: /healthz

port: 8080

httpHeaders:

- name: X-Custom-Header

value: Awesome

initialDelaySeconds: 3

periodSeconds: 3

EOF

[root@master01 damon]# kubectl apply -f pod-liveness-http.yaml

[root@master01 damon]# curl -v 192.168.1.62:8080/healthz

\* About to connect() to 192.168.1.62 port 8080 (#0)

\* Trying 192.168.1.62...

\* Connected to 192.168.1.62 (192.168.1.62) port 8080 (#0)

> GET /healthz HTTP/1.1

> User-Agent: curl/7.29.0

> Host: 192.168.1.62:8080

> Accept: \*/\*

>

< HTTP/1.1 200 OK // OK显示正常

< Date: Tue, 20 Nov 2018 12:01:26 GMT

< Content-Length: 2

< Content-Type: text/plain; charset=utf-8

<

\* Connection #0 to host 192.168.1.62 left intact

ok[root@master01 damon]# curl -v 192.168.1.62:8080/healthz

\* About to connect() to 192.168.1.62 port 8080 (#0)

\* Trying 192.168.1.62...

\* Connected to 192.168.1.62 (192.168.1.62) port 8080 (#0)

> GET /healthz HTTP/1.1

> User-Agent: curl/7.29.0

> Host: 192.168.1.62:8080

> Accept: \*/\*

>

< HTTP/1.1 500 Internal Server Error // error检测异常

< Date: Tue, 20 Nov 2018 12:01:39 GMT

< Content-Length: 19

< Content-Type: text/plain; charset=utf-8

<

\* Connection #0 to host 192.168.1.62 left intact

**1.3 TCP liveness probe**

cat << EOF > pod-liveness-tcp.yaml

apiVersion: v1

kind: Pod

metadata:

name: liveness-tcp

labels:

app: liveness-tcp

spec:

containers:

- name: liveness-tcp

image: python:2.7

imagePullPolicy: IfNotPresent

command: ["bash", "-c", "echo test > index.html && sleep 30 && python -m SimpleHTTPServer 8080"]

ports:

- containerPort: 8080

readinessProbe:

tcpSocket:

port: 8080

initialDelaySeconds: 35

periodSeconds: 10

livenessProbe:

tcpSocket:

port: 8080

initialDelaySeconds: 15

periodSeconds: 20

EOF

[root@master01 ~]# kubectl apply -f pod-liveness-tcp.yaml

[root@master01 ~]# kubectl get pod --watch //参看30秒以内podr的R

NAME READY STATUS RESTARTS AGE

goproxy 1/1 Running 1 5h46m

liveness-tcp 0/1 Running 0 4s

**2. readinessProbe**

**2.1 readinessProbe + livenessProbe+Service**

cat << EOF > service-healthcheck.yaml

apiVersion: apps/v1

kind: Deployment

metadata:

name: service-health

spec:

replicas: 2

selector:

matchLabels:

app: service-health

template:

metadata:

labels:

app: service-health

spec:

containers:

- name: service-health

image: python:2.7

imagePullPolicy: IfNotPresent

command: ["/bin/bash","-c","echo \$(hostname) > index.html && sleep 30 && python -m SimpleHTTPServer 8080"]

ports:

- containerPort: 8080

readinessProbe:

tcpSocket:

port: 8080

initialDelaySeconds: 10

periodSeconds: 10

EOF

kubectl apply -f service-healthcheck.yaml

### 创建service

kubectl expose deployment service-health --port=8080

[root@master01 ~]# kubectl get service

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 6d

service-health ClusterIP 10.109.21.81 <none> 8080/TCP 4s

curl 10.109.21.81:8080