

Health Monitoring

Probes

oc set probe --help

Startup

Readiness

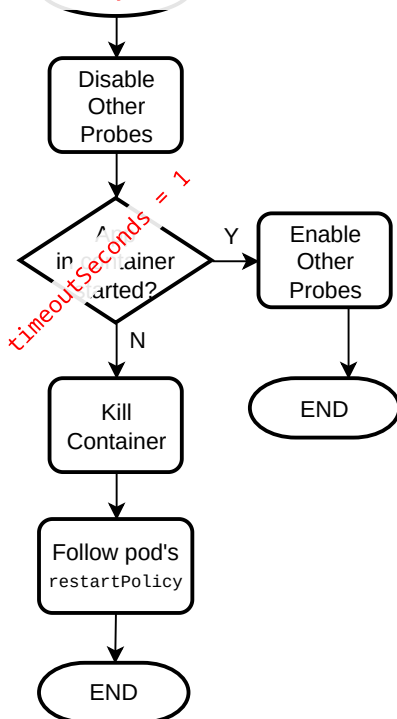
Liveness

Application in container started?
spec.containers.startupprobe

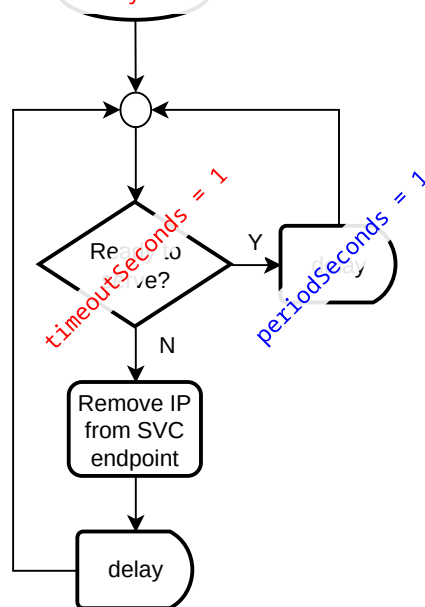
Container ready to serve requests?
spec.containers.readinessprobe

Container application healthy?
spec.containers.livenessprobe

initialDelaySeconds = 0

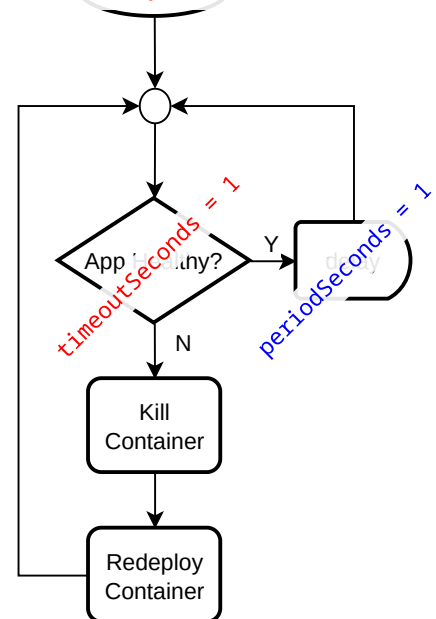


initialDelaySeconds = 0



successThreshold = 1
failureThreshold = 3

initialDelaySeconds = 0



successThreshold = 1
failureThreshold = 3

successThreshold = 1
failureThreshold = 3

Deployment/DC

```

spec:
  template:
    spec:
      containers:
        - image: ...
          ...Probe:
  
```

```

readinessProbe:
  httpGet:
    path: /ready
    port: 8080
    initialDelaySeconds: 15
    timeoutSeconds: 1
  
```

```

livenessProbe:
  exec:
    command:
      - cat
      - /tmp/health
    initialDelaySeconds: 15
    timeoutSeconds: 1
  
```

```

livenessProbe:
  tcpSocket:
    port: 3306
    periodSeconds: 15
    timeoutSeconds: 1
  
```

oc set probe deployment test-php \
--readiness \
--get-url=http://:8080/ready \
--initial-delay-seconds=15 \
--timeout-seconds=1

oc set probe deployment test-php \
--liveness \
-- cat /tmp/health \
--initial-delay-seconds=15 \
--timeout-seconds=1

oc set probe deployment test-php \
--liveness \
--open-tcp 3306 \
--period-seconds 15 \
--timeout-seconds 1

Mandatory

Optional

Blue-Green Deployment

Deploy production app, green

```
oc new-app registry.example.com/myapp:v1 --name green
oc expose deployment green      # expose dc/deployment to get svc
oc expose svc green            # expose svc to get route
oc get route                   # get hostname to access app
```

Deploy updated version of app, blue

```
oc new-app registry.example.com/myapp:v2 --name blue
oc expose deployment blue      # expose dc/deployment to get svc
```

```
oc patch route green -p '{"spec":{"to":{"name":"blue"}}}' # update route to use blue svc
```

A/B Deployment

Deploy production app, app-a

```
oc new-app registry.example.com/myapp:v1 --name app-a
oc expose deployment app-a      # expose dc/deployment to get svc
oc expose svc app-a --name myroute # expose svc to get route
oc get route                   # get hostname to access app
```

Deploy updated version of app, app-b

```
oc new-app registry.example.com/myapp:v2 --name app-b
oc expose deployment app-b      # expose dc/deployment to get svc
```

Note: router balances traffic according to weights(default=1).

Add service app-b to route and set weightage as 80% for app-a and 20% for app-b

```
oc set route-backends myroute app-a=80 app-b=20
oc set route-backends myroute # verify configuration
```

N-1 Compatibility & Graceful Termination - Refer to Notes

find out haproxy
weight pct vs conn