

# K8S 21: Kubernetes. Manifests. Probes

## Описание:

### Liveness Probe

Проверки работоспособности приложения внутри контейнера очень важны. Это поможет Kubernetes своевременно рестартовать приложение. Liveness probe проверяет приложение на работоспособность. Есть три типа встроенных проверок:

- `liveness command` — команда выполняется внутри контейнера. Если результат выполнения отличен от нуля, то проверка считается проваленной.
- `liveness http request` — данная проверка выполняет HTTP GET на указанный адрес, ожидает ответ 200 в случае успеха.
- `tcp liveness probe` — проверка на установление TCP-соединения на указанный порт. Не подразумевает передачу данных, только проверка установления соединения.

Пример `liveness command`:

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    test: liveness
  name: liveness-exec
spec:
  containers:
  - name: liveness
    image: k8s.gcr.io/busybox
    args:
    - /bin/sh
    - -c
    - touch /tmp/healthy; sleep 30; rm -rf /tmp/healthy; sleep 600
  livenessProbe:
    exec:
      command:
      - cat
      - /tmp/healthy
    initialDelaySeconds: 3
    periodSeconds: 5
```

Команда проверяет наличие файла `/tmp/healthy`. Заметьте дополнительные параметры:

- `initialDelaySeconds` — параметр задержки первого запуска проверки. Например, если приложение долго запускается, а данный параметр не задан, то kubelet убьет

pod раньше, чем приложение будет готово. В данной конфигурации первая проверка будет запущена спустя 5 секунд после запуска контейнера.

- `periodSeconds` — период запуска проверки, в данном случае 5 секунд.

Пример liveness HTTP Request:

`livenessProbe:`

`httpGet:`

`path: /healthz`

`port: 8080`

`httpHeaders:`

`- name: X-Custom-Header`

`value: Awesome`

`initialDelaySeconds: 3`

`periodSeconds: 3`

В этой конфигурации будет выполнен HTTP GET на `/healthz`, порт 8080 с дополнительными заголовками. Пример TCP liveness probe:

`livenessProbe:`

`tcpSocket:`

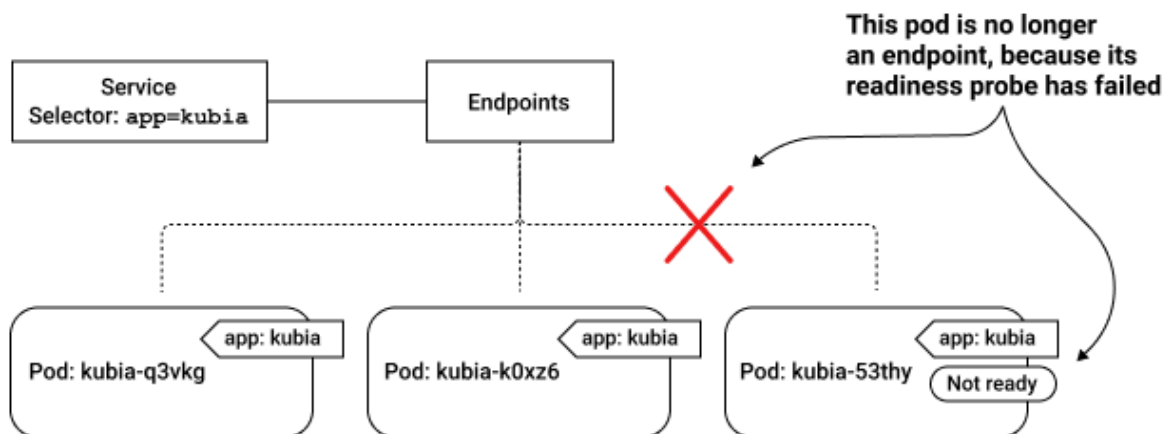
`port: 8080`

`initialDelaySeconds: 15`

`periodSeconds: 20`

В этом варианте проверяется соединение на порт 8080/TCP.

Readiness Probe



Это проверка готовности приложения к работе. Например, приложение долго запускалось, прошло проверку доступности, но еще не готово обрабатывать трафик пользователей, так как загружает кэши в `redis`. Данный тип проверок помогает kubelet понять, в какой момент времени можно начать направлять трафик на приложение.

Те же типы проверок, как в Liveness Probe, только описываются в `spec.containers.readinessProbe`, пример:

```
readinessProbe:
  exec:
    command:
      - cat
      - /tmp/healthy
  initialDelaySeconds: 5
  periodSeconds: 5
```

## Полезные ссылки:

- [Kubernetes Health Checks with Readiness and Liveness Probes](#)
- [Kubernetes Liveness and Readiness Probes](#)
- [Liveness probes в Kubernetes могут быть опасны \(habr\)](#)

## Задание:

1. Подготовьте Dockerfile и соберите приложение [gocalc](#) (в качестве registry использовать публичный docker.io).
2. Напишите deployment для указанного приложения с http Readiness/Liveness Probes.
3. Опишите, приемлемо ли в случае указанного приложения использовать иные виды Probes и почему?
4. Предоставьте написанный deployment в yaml-формате.