



Virtual Office Hours: K8s

April 1st, 2021 | 10:00-11:00 AM Pacific



Liran Cohen

CTO, Co-Founder
Sosivio

Liveness, Readiness, and Startup Probes

We will begin shortly...

www.sosiv.io

Download Presentation
Here:



Probe Types

- **Liveness**

Make sure the application is started and running and not stuck in a deadlock or similar state - **If failed - the container will be restarted**

- **Readiness**

Make sure that the application is ready to accept connections - **if failed the pod will not receive traffic through the service**

- **Startup**

Take care of slow starting applications to prevent crash loops- **if failed neither liveness nor readiness probes will start**

Why Does K8s Need Probes?

- Much like an SRE, K8s requires **as precise as possible feedback** to take decisions
- Probes are a way to “explain” to K8s how your application starts, runs and when should it be considered failing.
- **Avoid and or be aware bad requests or timeout situations**
due to incomplete provisioning/bad configuration and other situations.
- **Assist in scale out situations, rolling deployments and replacing a failed pods**
with minimal outages.

Probe Checks

- **Tcp probe**

Check if the process inside the container is allowing to perform a tcp connect in the designated port

- **HTTP probe**

- Without http headers - will indicate success if HTTP code returned is higher than 200 and lower than 400
- With headers - expects an HTTP header in the response (must be specified in the deployment)

- **Command**

Run a command and examine the exit status, any status higher than 0 will fail the probe.

Probe Configuration

- **initialDelaySeconds**

Number of seconds after the container has started before liveness or readiness probes are initiated. Defaults to 0 seconds. Minimum value is 0.

- **periodSeconds**

How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.

- **timeoutSeconds:**

Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1.

Before Kubernetes 1.20, the field timeoutSeconds was not respected for exec probes: probes continued running indefinitely, even past their configured deadline, until a result was returned.

- **successThreshold:**

Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup Probes. Minimum value is 1.

- **failureThreshold:**

When a probe fails, Kubernetes will try failureThreshold times before giving up. Giving up in case of liveness probe means restarting the container. In case of readiness probe the Pod will be marked Unready. Defaults to 3. Minimum value is 1.

When Probes are Not Configured

- **A pod will be deemed “Running” as soon as the container starts**
 - If the process is trying to perform an operation but times out or fails without failing the process, the pod will stay at running state although the process(es) inside it are stuck in a deadlock state.
 - If a main process creates child process(es) and they exit abnormally but the parent process is still running the pod will remain in “Running” state and will never recover
- **Traffic will be directed to the pod by the service as soon as the container starts**
 - A pod will immediately receive traffic as soon as all of its containers reach “Ready” state, this may lead to a situation where the application is still in warmup/provisioning stage and is not ready to receive traffic.
 - If a pod is depended on an external resource to perform its operations (such as a database) and the database is unavailable, the pod will receive traffic but will not be able to perform its designated operation.

Probe and Application Design

- Design applications with probes in mind
- Externalize different application/process states
- fail fast