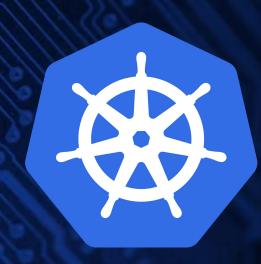
Monitor Containers in Kubernetes

Manage Container Health Checks





Container Monitoring

- Container Health
- → Liveness Probe
- → StartUp Probe
- → Readiness Probe
- → Hands On Demonstration

Container Health

- Kubernetes is feature Rich, and provide number of features to monitor the containers.
- Active Monitoring Helps K8s to decide the container state and Auto Restart in Case of Container Failure.

Liveness Probe

- Liveness probe helps to determine the Container State.
- By Default, K8s only consider container to be down, if container process stops.
- Liveness probe helps user to improve & customized this Container Monitoring mechanism.
- User can execute Two types of Liveness probes Run Command in Container, Periodic HTTP Health Check.

Liveness Probe

- Liveness via Container Command manifest.
- initialDelaySeconds: How long to wait before sending a probe after a container starts.
- periodSeconds: How often a probe will be sent.

```
livenessProbe:
exec:
command:
-some command here-
initialDelaySeconds: 5
periodSeconds: 5
```

Liveness Probe

- Liveness via HTTP Request manifest.
- timeoutSeconds: How long a request can take to respond before it's considered a failure.

```
livenessProbe:
httpGet:
path: /health.html
port: 8080
httpHeaders:
- name: Custom-Header
value: Awesome
initialDelaySeconds: 3
periodSeconds: 3
timeoutSeconds: 1
```

StartUp Probe

- Setting up Liveness probe is very Tricky with Application which have Long StartUp Time.
- > StartUp probe runs at container StartUp and stop running once container success.
- Once the startup probe has succeeded once, the liveness probe takes over to provide a fast response to container deadlocks.

StartUp Probe

- StartUp via HTTP Request manifest.
- failureThreshold: When a probe fails, Kubernetes will try failure Threshold times before giving up.
- Application will have a maximum of 5 minutes (30 * 10 = 300s) to finish its startup.

```
startupProbe:
httpGet:
path: /health.html
port: 8080
failureThreshold: 30
periodSeconds: 10
```

Readiness Probe

- Readiness is used to detect if a container is ready to accept traffic.
- Sometimes application might need to load large data or configuration files during startup, or depend on external services after startup.
- > NO Traffic will be sent to a pod until container pass the Readiness Probe.

Readiness Probe

- Readiness Probe manifest.
- Configuration for HTTP readiness probes also remains identical to liveness probes.
- Readiness and liveness probes can be used in parallel for the same container.

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

Hands On Demonstration

Thank You...

Don't be the Same! Be Better!!!