



# kubernetes

---

*Kubernetes: Liveness (Health-Check)*

## *KUBERNETES : Basics of Kuebernetes*

---

- **Kubernetes liveness probes** is to detect and remedy **Application Malfunctions**.
- **Application Malfunctions:** Pods & Containers are running but Application might dead due to some reasons.
- To deduct and resolve Problem with Application, user need to put liveness(HealthCheck).
- Liveness will actually attempt to restart a container\* if it fails.

## *KUBERNETES : Basics of Kuebernetes*

---

- User can Execute **2 types** of LiveNess.
- Running Command in **Container Periodically**.
- Periodic Check on **HTTP Request(URL)**
- A Production Application always have HealthCheck to ensure the **availability and resiliency**.

# KUBERNETES : Basics of Kuebernetes

---

- Liveness Probes via Container Command in Manifest.

`livenessProbe:`

`exec:`

`command:`

- Some commands here-

`initialDelaySeconds: 5`

`periodSeconds: 5`

- **initialDelaySeconds:** How long to wait before sending a probe after a container starts.  
For liveness probes this should be safely longer than the time your app usually takes to start up. Without that, you could get stuck in a reboot loop.
- **periodSeconds:** How often a probe will be sent. The value you set here depends on finding a balance between sending too many probes to your service or going too long without detecting a failure.

## *KUBERNETES : Basics of Kuebernetes*

---

- Liveness Probes via HTTP Request in Manifest.

**livenessProbe:**

**httpGet:**

**path:**

**port:**

**initialDelaySeconds: 5**

**timeoutSeconds: 1**

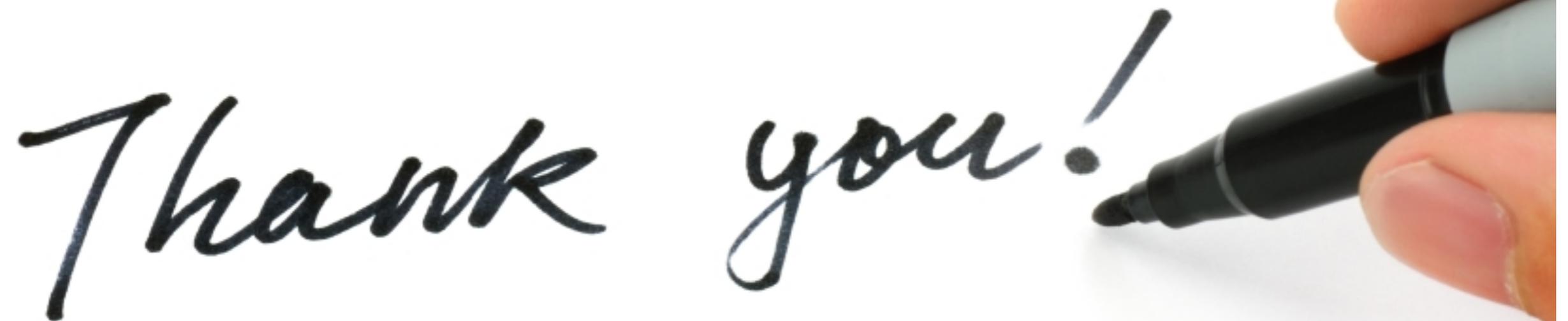
**periodSeconds: 5**

- **timeoutSeconds:** How long a request can take to respond before it's considered a failure.

*Will see you in Next Lecture...*

---

*Thank you!*

A close-up photograph of a hand holding a black marker, writing the words 'Thank you!' in a cursive script on a white surface. The hand is positioned on the right side of the frame, with the marker tip touching the paper. The background is plain white.

*See you in next lecture ...*