



kubernetes

Kubernetes: Readiness

KUBERNETES : Basics of Kuebernetes

- **Readiness** is used to detect if a container is ready to accept traffic.
- It indicates if Container is ready to serve the Traffic.
- If the Readiness check fails, the container will not be restarted, but the **Pod IP address will be removed** from the service(load Balancer).
- Readiness will make sure that at StartUp, the Pod will only receive traffic when the Test succeed.
- Readiness probe available means the pod will startup first before traffic starts flowing to it.

KUBERNETES : Basics of Kuebernetes

- Updating deployments without readiness probes can result in downtime as old pods are replaced by new pods. If the new pods are misconfigured or somehow broken, that downtime extends until you detect the problem and rollback.
- With readiness probes, Kubernetes will not send traffic to a pod until the probe is successful. When updating a deployment, it will also leave old replica(s) running until probes have been successful on new replica.

KUBERNETES : Basics of Kuebernetes

When to use Readiness and Liveness:

- **Liveness** and **Readiness** probes are both configured in the pod's YAML file. Each type has different use cases.
- **Liveness** probes are used to diagnose unhealthy containers.
- **Liveness** detect a problem in your service when it cannot progress, and will restart the problematic container according to its restart policy.
- **Readiness** probe on the other hand is useful in the case of an application being temporarily unable to serve traffic.
- For example, maybe the application needs to load a large dataset or some configuration files during the startup phase. Even though the service is running, it's not fully available, and in this scenario, Kubernetes may have trouble scaling it up and could fail. With a readiness probe, Kubernetes waits until the service is fully started before it sends traffic to the new copy.

KUBERNETES : Basics of Kuebernetes

- Readiness and liveness probes can be used in parallel for the same container. By using both types in the same container, you ensure that traffic does not reach a container that is not ready for it, and that containers are restarted when they fail.
- Example: One of our services is a FoodApp running on Kubernetes.

The time between when a pod starts running and the service is actually responding to requests can be significant (5–10 seconds). Without a readiness probe, we'd end up with at least that much downtime every time we updated that deployment.

When not to use Readiness and Liveness:

- If you have a process inside of your container that is able to crash on its own when it becomes unhealthy or encounters an error, it is not necessary to use a liveness probe.
- In case if you have a container that needs to load configuration files or other data before it starts up successfully.
- If you want to drain the requests after the Pod is deleted.

KUBERNETES : Basics of Kuebernetes

- Readiness probes are configured similarly to liveness probes. The only difference is that you use the **readinessProbe** field instead of the **livenessProbe** field.

readinessProbe:

 exec:

 command:

 - cat

 - /tmp/healthy

 initialDelaySeconds: 5

 periodSeconds: 5

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 fingers gripping the marker. The text is written in a dark, fluid cursive style. The background is plain white.

See you in next lecture ...