# Lab: Exploring Probes

# Lab 1: Liveness Probe Demonstration

## 🎯 Objective:

Explore the Kubernetes liveness probe.

## 🛠️ Steps

**File:** `liveness-pod.yaml`

1. Deploy:

```
kubectl apply -f liveness-pod.yaml
```

2. Observe automatic container restart using:

```
kubectl describe pod liveness-demo
```

2. (continued)

`Events`

```
Type     Reason     Age                From               Message
----     ------     ----               ----               -------
Warning  Unhealthy  5s (x3 over 15s)   kubelet            Liveness probe failed: HTTP probe failed with statuscode: 404
Normal   Killing    5s                 kubelet            Container liveness-container failed liveness probe, will be restarted
```

**This means:**

- The liveness probe started after 10 seconds
- It made a request to http://<pod_ip>:8080/healthz
- NGINX didn’t respond (wrong port) or returned 404 (wrong path)
- Kubernetes restarted the container

3. Modify the livenessProbe path and the initialDelaySeconds of the yaml to the following. This uses a path that NGINX can serve (e.g., /) and modifies the initial delay.

```
livenessProbe:
httpGet:
   path: /
   port: 80
initialDelaySeconds: 5
periodSeconds: 5
```

After editing the file use these commands:

```
kubectl delete -f liveness-pod.yaml
kubectl apply -f liveness-pod.yaml
```

**Successfully running pod:**

`Events`

```
Type    Reason     Age   From               Message
----    ------     ----  ----               -------
Normal  Scheduled  14s   default-scheduler  Successfully assigned default/liveness-demo to minikube-m03
Normal  Pulling    14s   kubelet            Pulling image "nginx"
Normal  Pulled     13s   kubelet            Successfully pulled image "nginx" in 1.039s (1.039s including waiting). Image size: 197741282 bytes.
Normal  Created    13s   kubelet            Created container: liveness-container
Normal  Started    13s   kubelet            Started container liveness-container
```

## 4. Cleanup

```
kubectl delete -f liveness-pod.yaml
```

# Lab 2: Readiness Probe Demonstration

## 🎯 Objective:

Explore the Kubernetes readiness probe.

Unlike liveness probes (which restarts a Pod if it's unhealthy), readiness probes control traffic flow. When the readiness check fails:

- The Pod stays running
- But it is removed from the Service endpoints
- So it does not receive any traffic

## 🛠️ Steps

**File:** `readiness-pod.yaml`

1. Deploy:

```
kubectl apply -f readiness-pod.yaml
```

This pod has defined a readinessProbe path of `/ready`. NGINX does not have a `/ready` path, so this readiness probe will fail with a 404.

2. Check pod by running command:

```
kubectl get pods
```

```
NAME                                READY   STATUS    RESTARTS   AGE
readiness-demo                      0/1     Running   0          18s
```

Notice the pod status is `Running` and the `Ready` column is showing `0/1`, meaning not ready.

Lets view the pod events by using describe:

```
kubectl describe pod readiness-demo
```

`Events`

```
  Type     Reason     Age                  From               Message
----     ------     ----                 ----               -------
Normal   Scheduled  2m54s                default-scheduler  Successfully assigned default/readiness-demo to minikube-m03
Normal   Pulling    2m55s                kubelet            Pulling image "nginx"
Normal   Pulled     2m53s                kubelet            Successfully pulled image "nginx" in 1.196s (1.196s including waiting). Image size: 197741282 bytes.
Normal   Created    2m53s                kubelet            Created container: readiness-container
Normal   Started    2m53s                kubelet            Started container readiness-container
Warning  Unhealthy  97s (x25 over 2m48s)  kubelet            Readiness probe failed: HTTP probe failed with statuscode: 404
```

3. Modify the readiness-pod.yaml file by making the following change;

   `path: /ready` to `path: /`

4. Delete the old pod and deploy the changed pod (you must delete the old pod before creating the new pod ):

```
kubectl delete -f readiness-pod.yaml
kubectl apply - readiness-pod.yaml
```

5. Check pod by running the command quickly, multiple times, after deploying the pod:

```
kubectl get pods
```

```
NAME                          READY   STATUS    RESTARTS   AGE
readiness-demo                0/1     Running   0          3s

NAME                          READY   STATUS    RESTARTS   AGE
readiness-demo                0/1     Running   0          6s

NAME                          READY   STATUS    RESTARTS   AGE
readiness-demo                1/1     Running   0          7s
```

Note the AGE column is showing how many seconds the pod has bee running.

# Lab 3: startup Probe Demonstration

## 🎯 Objective:

Explore the Kubernetes readiness probe.

## 🛠️ Startup Probe Lab

**File:** `startup-pod.yaml`

1. Deploy:

```
kubectl apply -f startup-pod.yaml
```

2. Observe how liveness probe waits until startup completes:

```
kubectl describe pod startup-demo
```

`Events`

```
Type     Reason     Age              From               Message
----     ------     ----             ----               -------
Normal   Scheduled  26s              default-scheduler  Successfully assigned default/startup-demo to minikube-m03
Normal   Pulling    26s              kubelet            Pulling image "nginx"
Normal   Pulled     25s              kubelet            Successfully pulled image "nginx" in 916ms (916ms including waiting). Image size: 197741282 bytes.
Normal   Created    25s              kubelet            Created container: nginx
Normal   Started    25s              kubelet            Started container nginx
Warning  Unhealthy  6s (x2 over 16s)  kubelet            Startup probe failed: Get "http://10.244.2.68:80/": dial tcp 10.244.2.68:80: connect: connection refused
```

2. (continued)

You're seeing:

```
Startup probe failed: Get "http://10.244.2.68:80/": dial tcp 10.244.2.68:80: connect: connection refused
```

**This means:**

- The container was started
- The probe tried to connect to port 80 inside the container
- But nothing was listening on that port at that moment

**This happened because of:**

```
command: ["sh", "-c", "sleep 60 && nginx -g 'daemon off;'"]
```

**Explanation:**

- You told the container to sleep for 60 seconds before starting NGINX
- During that 60-second sleep, there is no NGINX process running yet
- But your startupProbe begins sending HTTP requests before NGINX is running
- So the connection is refused, not just returning a 404 — it's literally not accepting TCP connections

3. Stop the non-working pod:

```
kubectl delete -f startup-pod.yaml
```

4. Deploy the fixed pod.

```
kubectl deploy startup-pod-fixed.yaml
```

These changes have been made in the new yaml file.

```
startupProbe:
httpGet:
   path: /
   port: 80
initialDelaySeconds: 30    # ← Added this
failureThreshold: 10
periodSeconds: 10
```

You could also shorten the sleep;

```
command: ["sh", "-c", "sleep 20 && nginx -g 'daemon off;'"]
```

5. Cleanup:

```
kubectl delete -f startup-pod-fixed.yaml
```