**Why did liveness prob failed? Or crash loop backoff ? or restart of pods many times?**

The logs you've provided show that your Spring Boot application has started successfully and is running on port **8030**. Additionally, it specifies that two endpoints are exposed beneath the base path **/actuator**, which means the Actuator functionality should be available.

Given this information, the liveness probe should be configured to hit the Actuator health endpoint, which is typically **/actuator/health**. Ensure your **liveness Probe** in the deployment YAML is set up as follows:

yamlCopy code

livenessProbe:

httpGet: path: /actuator/health

port: 8030

initialDelaySeconds: 30

timeoutSeconds: 5

successThreshold: 1

failureThreshold: 3

This configuration tells Kubernetes to wait for 30 seconds after the container starts before performing the first health check. It will attempt the health check every default period (usually 10 seconds), and it will wait for 5 seconds for the httpGet request to **path: /actuator/health** at **port: 8030** before considering it a failure. If the health check fails more than three consecutive attempts (as specified by **failure Threshold**), Kubernetes will restart the container.

Summary: Essentially, this configuration means that after the container starts, Kubernetes will wait for 30 seconds and then start executing an HTTP GET request every default probe interval (usually 10 seconds) to the **/actuator/health** endpoint on port **8030**. If this request does not succeed within 5 seconds or the HTTP response is anything other than a 200-399 status code, the failure will be counted. If the probe fails 3 consecutive times, the container will be restarted.

Commands:

Kubectl log podname ( it will provide the pod logs)

Kubectl describe pod podname (this provide status when container is created , started , running etc..

Solution: the health check path I mentioned was wrong and I did increase initial timeout second.