In Kubernetes, Liveness and Readiness Probes are mechanisms used to determine the health and operational status of a pod.

**Liveness Probe**

A Liveness Probe checks if an application inside a pod is still running. If the liveness probe fails, Kubernetes will kill the pod and restart it according to its restart policy.

**Readiness Probe**

A Readiness Probe checks if an application inside a pod is ready to serve traffic. If the readiness probe fails, the pod will be removed from the service load balancers and will not receive any traffic until it passes the readiness check.

path: / - This endpoint should return a status indicating the app is ready to serve traffic.

port: 5000 - The port on which the application is listening.

initialDelaySeconds: 1 - The probe will start after 1 second.

periodSeconds: 10 - The probe will run every 10 seconds.

**Liveness Probe Failures**

1. Application Crash:

Scenario: The application inside the container crashes or exits unexpectedly.

Result: The liveness probe fails because the application is no longer running, and Kubernetes restarts the pod.

2. Endpoint Not Responding:

Scenario: The / endpoint is not responding due to the application being in a deadlock or an unresponsive state.

Result: The liveness probe fails, and Kubernetes restarts the pod to recover from the unresponsive state.

3. Configuration Errors:

Scenario: The specified liveness probe path (/) is incorrect or does not exist in the application.

Result: The probe fails repeatedly because the path returns a non-200 status code, causing Kubernetes to restart the pod.

4. Port Misconfiguration:

Scenario: The liveness probe is configured to check a port that the application is not listening on.

Result: The probe fails because it cannot establish a connection, leading to pod restarts.

**Readiness Probe Failures**

1. Application Not Ready:

Scenario: The application is not ready to handle requests, perhaps due to initial loading, ongoing startup processes, or dependency issues.

Result: The readiness probe fails, and the pod is not marked as ready, thus it does not receive any traffic.

2. Endpoint Not Responding:

Scenario: The / endpoint is not responding or is slow to respond, indicating the application is not ready to handle traffic.

Result: The readiness probe fails, removing the pod from the service endpoints until it responds correctly.

3. Configuration Errors:

Scenario: The readiness probe path (/) is incorrect or does not exist in the application.

Result: The probe fails because the path returns a non-200 status code, causing the pod to be marked as not ready.

4. Port Misconfiguration:

Scenario: The readiness probe is configured to check a port that the application is not listening on.

Result: The probe fails because it cannot establish a connection, leading to the pod being marked as not ready.