**❗ What Happens *Without* Graceful Shutdown?**

In a real-world production environment, if an app is shut down suddenly (e.g., pod deletion, deployment rollout, scale down), here's what can go wrong:

A screenshot of a computer

AI-generated content may be incorrect.

This leads to:

* Failed user actions
* Data loss
* Poor customer experience
* Hard-to-debug issues

**🚦 What is Graceful Shutdown?**

When Kubernetes wants to shut down a pod (e.g., during scaling, rolling updates, node failure), it follows this process:

When a pod is deleted (manually, scaled down, or during a rollout):

1. Kubernetes sends a SIGTERM signal to the main container process.
2. It **waits 30 seconds** (default) for the app to **clean up and exit**.
3. If the container is still running after 30 seconds, Kubernetes sends SIGKILL to **forcefully terminate** it.

**🧠 Why You Might Increase It**

If your application needs more time to:

* Finish long-running requests
* Flush logs
* Clean up DB connections
* Deregister from a service mesh or load balancer

Then set it to 40–120 seconds depending on the use case.

**No, you don’t *have* to use Java code for graceful shutdown in Kubernetes — but it depends on how deep your shutdown logic needs to go.**

**✅ What Kubernetes Handles *Without Java Code***

**If you simply want Kubernetes to stop sending traffic to your pod and wait before killing it:**

1. **Readiness probe will fail (you configure it).**
2. **preStop hook gives a delay before SIGTERM.**
3. **Kubernetes waits for:**
   * **terminationGracePeriodSeconds**
   * **preStop hook to finish**
   * **Then sends SIGTERM**

**This is enough if your app doesn’t need to release resources or save state.**

**✅ How Spring Boot Handles This**

Starting from **Spring Boot 2.3+**, graceful shutdown is **supported out-of-the-box** using the Spring lifecycle hooks.

**❗ But... If You Want Cleanup (e.g., flush queues, close DB, logs):**

That’s where Java code is **recommended**:

**🔍 What Happens When Pod is Terminated**

1. **readinessProbe fails**: Kubernetes stops routing traffic to the pod.
2. **preStop executes**: Gives app 10 seconds to finish work.
3. **terminationGracePeriodSeconds** (30s): Time to exit gracefully.
4. **Then**: SIGTERM → SIGKILL if app still alive after 30s.

# nginx-graceful.yaml

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-app

labels:

app: nginx-app

spec:

replicas: 3

selector:

matchLabels:

app: nginx-app

template:

metadata:

labels:

app: nginx-app

spec:

terminationGracePeriodSeconds: 30 # Wait time before force kill

containers:

- name: nginx

image: nginx:stable

ports:

- containerPort: 80

lifecycle:

preStop:

exec:

command: ["/bin/sh", "-c", "echo Graceful shutdown && sleep 15"]