7 Common Kubernetes Pitfalls

01. Incorrect Labels and Selectors

02. Ignoring Health Checks

03. Using the Default Namespace for all Objects

04. Using the ‘Latest’ Tag

05. Lack of Monitoring and Logging

06. Wrong Container Port Mapped to a Service

07. Crashloopbackoff error

1. Incorrect Labels and Selectors

A non-matching selector puts the deployment resource into an unsupported state, and you might see an error related to an incorrect label and selector.

Make sure you use correct labels and selectors in your YAML files and carefully check for typos.

2. Ignoring Health Checks

When deploying your services to Kubernetes, health checks play a crucial role in maintaining your services.

Through health checks, you keep an eye on the health of the pods and their containers.

3. Using the Default Namespace for all Objects

In the development environment, using the default namespace might not be an issue, but it can cause production problems if you **execute the command without mentioning the namespace.**

4. Using the ‘Latest’ Tag

The “latest” tag doesn’t always deploy the version you think is the most recent one. Using the “latest” command for deployment, **you will not be able to roll back to an earlier version.**

5. Lack of Monitoring and Logging

You should set up a **log aggregation server and monitoring system** to keep an eye on your application.

That will help you not only see various bottlenecks in your system but also how to measure and optimize the performance of your Kubernetes clusters.

6. Wrong Container Port Mapped to a Service

If you are facing the error “**connection refused**” or no reply from containers, then it might be an issue of an incorrect container port mapped to the service.

This is because the two parameters in the service are similar to each other. **One is “Targetport” while the other is “port”.**

7. Crashloopbackoff error

Another frequent error in Kubernetes is the crashloopbackoff error. It occurs when a pod is running, but one of its containers keeps restarting due to termination.

You need to check the pod description and pod logs to troubleshoot and fix the root cause.