DevOps Java Application Deployment - Day-by-Day Documentation

Introduction

This documentation explains step-by-step what was done each day during the DevOps Java Application Deployment project. It describes each task in simple words, so it's easier to understand what was done and why.

Day 1: Java Build and GitHub Setup

What we did:

- Created a simple Java application using Maven, a tool that helps build Java projects.
- Uploaded (pushed) the code to GitHub so it's stored safely and easy to share.

Why:

- Using Maven makes it easy to manage and build Java code.
- GitHub is used to keep track of our project, share code, and use in automation later.

Day 2: Dockerizing the Java App

What we did:

- Created a Dockerfile which tells Docker how to package our app.
- Built a Docker image of the Java app and tested it to see if it works.

Why:

- Docker allows us to run our app in any environment without issues.
- It makes our app portable and easy to deploy.

Day 3: Setting Up Jenkins for CI

What we did:

- Installed Jenkins, a tool that helps us automate tasks like building code.
- Created a Jenkins pipeline to build our Java app and Docker image whenever we push new code to GitHub.

DevOps Java Application Deployment - Day-by-Day Documentation

Why:

- Jenkins helps automate the build process so we don't have to do it manually every time.
- It saves time and avoids mistakes.

Day 4: Pushing Docker Image to Docker Hub

What we did:

- Configured Jenkins to push the Docker image to Docker Hub after it is built.
- Checked if our image is publicly available on Docker Hub.

Why:

- Docker Hub is like GitHub, but for Docker images.
- It lets us store and share images so they can be used anywhere.

Day 5: Deploying App on Kubernetes Using Minikube

What we did:

- Installed and started Minikube, a tool that runs Kubernetes locally.
- Wrote Kubernetes YAML files to tell Kubernetes how to run our app.
- Used kubectl to deploy the app on the Kubernetes cluster.
- Checked that the app was running and accessible.

Why:

- Kubernetes manages applications that run in containers.
- It allows us to scale apps easily and keep them running smoothly.

Day 6: Planning Jenkins to Deploy to Kubernetes

What we did:

- Planned how to make Jenkins deploy the app to Kubernetes automatically.
- This step needs extra setup to connect Jenkins with the Kubernetes cluster.

DevOps Java Application Deployment - Day-by-Day Documentation

Why:

- This makes the process fully automatic (CI/CD).
- Once setup, Jenkins will build, test, and deploy without us doing it manually.

Conclusion

This project taught us how to take a simple Java app and deploy it using modern DevOps tools like Docker, Jenkins, and Kubernetes. We now understand the complete pipeline from writing code to deploying it automatically in a real-world setup.