PROJECT: CI/CD Pipeline with Jenkins

PROBLEM: The development team was struggling with manual deployment processes, leading to errors, delays, and inconsistent releases. The goal was to create a streamlined CI/CD pipeline to automate the entire process.

TOOLS USED:

- Jenkins
- SonarQube
- Nexus
- Trivy
- Git
- AWS EC2
- AWS EKS
- AWS

SOLUTION:

1. Source Control: Git was used as the source control system, with GitHub as the repository.
2. Build and Package: Jenkins was used to create a build pipeline, compiling and packaging the application code into a Docker image.
3. Automated Testing: Selenium and JUnit were used for automated unit testing and integration testing.
4. Deployment: Kubernetes was used to deploy the application to a cloud environment (AWS EKS).
5. Monitoring and Logging: Prometheus, Grafana, and ELK Stack were used for monitoring and logging.
6. CI/CD Tooling: Jenkinsfile was used to define the CI/CD pipeline, with automated triggers and workflows.

DevOps Practices:

1. Continuous Integration: Code changes triggered automated builds and tests.
2. Continuous Deployment: Automated deployment to production after successful testing.
3. Continuous Monitoring: Real-time monitoring and logging for issue detection and resolution.

4. Infrastructure as Code: Terraform was used to manage infrastructure as code.

Results:

1. Faster Time-to-Market: Automated pipeline reduced deployment time by 90%.
2. Improved Reliability: Automated testing and deployment reduced errors by 75%.
3. Increased Efficiency: Automated pipeline saved 20 hours of manual effort per week.
4. Enhanced Visibility: Real-time monitoring and logging enabled proactive issue detection.

Lessons Learned:

1. Automation: Automating the CI/CD pipeline reduced manual errors and increased efficiency.
2. Consistency: Consistent deployment processes ensured reliable releases.
3. Visibility: Real-time monitoring and logging enabled proactive issue detection and resolution.
4. Collaboration: Cross-functional collaboration between development, QA, and operations teams improved communication and reduced silos.