

GANDAKI COLLEGE OF ENGINEERING AND SCIENCE

Lamachaur, Pokhara



LAB REPORT OF Agile Software Development LAB – 3

SUBMITTED BY:

Baradan Parajuli

Roll No: 16

6th Semester

BE Software

SUBMITTED TO:

Er. Rajendra Bdr. Thapa

LAB 3: Deployment Tools

Objective

To understand the role of deployment tools in software development and learn how to automate application deployment using standard tools and practices

Theory

Deployment tools are used to automate the process of delivering software from development to production environments. They ensure that applications are released efficiently, consistently, and with minimal human error. In agile development, continuous deployment and integration tools are essential to support fast iterations and frequent releases.

Some popular deployment tools include:

- Docker – Containerizes applications for consistent deployment across environments.
- Kubernetes – Manages and orchestrates containers at scale.
- Jenkins – Automates build, test, and deployment pipelines.
- GitHub Actions – Integrates with GitHub to automate workflows like testing and deployment.
- Heroku / Netlify / Vercel – Simplified cloud platforms for hosting and deploying applications.

Tools Used

- Docker (for containerization)
- GitHub and GitHub Actions (for version control and deployment automation)
- Heroku (for web app deployment)
- VS Code (as the development environment)

Methodology

- Application Setup – Created a basic web application (e.g., Node.js or HTML/CSS project).
- Version Control – Pushed the code to a GitHub repository.
- Dockerization – Wrote a Dockerfile to containerize the application.
- CI/CD Configuration – Used GitHub Actions to automate build and deployment steps.
- Deployment – Deployed the application to Heroku (or another platform).
- Testing – Accessed the deployed app online to verify successful deployment.

Observations

Strengths:

- Deployment became faster and repeatable using automated tools.
- Docker ensured the same environment across development and production.
- GitHub Actions eliminated the need for manual deployments.
- Deployment logs helped in debugging any issues during release.

Challenges:

- Initial setup of Docker and CI/CD pipelines required attention to detail.
- Managing environment variables securely was important but tricky.
- Platform-specific limitations (like free tier limits on Heroku) affected deployment options

Conclusion

This lab introduced the key concepts and tools used for modern software deployment. By practicing containerization and CI/CD automation, we learned how to streamline the deployment process and reduce the risk of manual errors. Deployment tools are essential in agile development for fast, reliable, and scalable software delivery.