

# DevOps Project

## Overview:

In this project, you will work **individually** to design, build, and deploy a **small** backend service or a REST API (under **150 lines** of code). The goal is to practice DevOps concepts end-to-end from coding and containerization to automation, deployment, observability, and security.

## Objectives:

- Use GitHub [Issues](#) or [Projects](#) to break your project into smaller tasks.
- Write a small backend service or REST API
- Use Git/GitHub for version control.
- Open **dedicated Pull Requests** for each task and perform at least one **peer review exchange** with a classmate.
- Set up a **CI/CD** pipeline (e.g., GitHub Actions) to build and test your service automatically.
- Implement **observability basics**:
  - **Metrics**: expose simple metrics (e.g., request count, response time).
  - **Logs**: ensure structured logs for incoming requests and errors.
  - **Tracing**: add basic request tracing.
- Add **security checks**:
  - **SAST (Static Application Security Testing)**: run a static scan of your code
  - **DAST (Dynamic Application Security Testing)**: run a simple runtime scan of the deployed API.
- Containerize your application using **Docker**.
- Deploy and run the service in **Kubernetes** (minikube or kind)(bonus: use cloud service).
- Create proper documentation: a clear **README.md** with setup instructions, local running guide, Docker usage, and API examples.
- Deliver a **final report (1–2 pages)** describing your architecture, tools, observability, security, Kubernetes setup, and lessons learned

## Deliverables:

1. GitHub repository containing your source code, Dockerfile, and Kubernetes manifests.
2. A functioning CI/CD pipeline that builds, tests, scans, and deploys your project.
3. A published Docker image (Docker Hub or similar).
4. The service is deployed and accessible locally (bonus: use the cloud).
5. Evidence of observability: metrics endpoint, dashboard, sample logs, and basic tracing.
6. Evidence of security scans (SAST + DAST results).
7. Final report and presentation with Q&A (~10 min).

### **Evaluation Criteria (100%):**

- Backend functionality (under 150 lines) **10%**
- GitHub workflow (issues, PRs, reviews) **10%**
  - For peer reviews, you will be graded on **the feedback you provide**.
  - Make sure your comments are constructive and respectful.
- CI/CD pipeline (build, test, deploy) **15%**
- Containerization (Dockerfile, Docker image, Docker compose) **10%**
- Observability (metrics, logs, tracing) **15%**
- Security (SAST + DAST scans) **10%**
- Kubernetes deployment **10%**
- Final report, presentation & answering questions **20%**

**“Enjoy the journey, not just the destination.”  
Good luck. Have fun!**