

# DevOps Final Assignment: A Full CI/CD and Successful Deployment of Application

May 29, 2025

## Abstract:

The DevOps Internship Final Assignment is a practical evaluation designed to test an intern's understanding of modern DevOps practices through the creation of a complete CI/CD pipeline. The goal is to develop, containerize, and deploy a simple web application using widely adopted tools and automation workflows. The assignment is divided into five structured tasks. It begins with the development of a basic web application using a framework such as Flask or Node.js. The app must display a message like "Hello from DevOps" and be hosted on GitHub with a clean project structure.

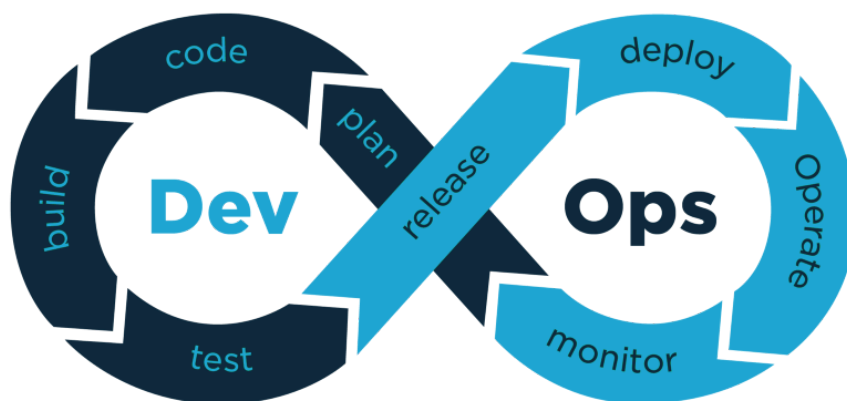
Next, the intern must containerize the application using Docker. This involves writing a functional Dockerfile, exposing the appropriate port, and optionally pushing the image to Docker Hub, demonstrating understanding of containerization and image management.

In the third task, a CI pipeline must be created using GitHub Actions. The workflow should trigger on pushes to the main branch, run basic tests, build the Docker image, and optionally push it to Docker Hub. This reinforces key DevOps principles like automation and continuous integration.

The fourth phase requires deploying the application to a local Kubernetes cluster. The intern must create Kubernetes manifests for deployment and service, ensuring the application is accessible and running as expected.

The final task focuses on documentation. Interns must write a comprehensive README.md that explains the project setup, tools used, and how the CI/CD workflow functions. Screenshots and an optional video walkthrough enhance clarity.

Evaluation is based on functionality, automation, adherence to best practices, and the quality of documentation. This assignment offers interns real-world experience with essential DevOps tools and workflows, helping bridge the gap between development and operations in a hands-on, structured manner.



CIRT & INFRA Team  
Business Automation Ltd.

Here's an assignment description with clear tasks for DevOps interns, structured as a set of requirements rather than solutions:

## **DevOps Internship Final Assignment** **End-to-End CI/CD Pipeline Implementation**

### **Objective**

Demonstrate your understanding of core DevOps principles by building, containerizing, and deploying a web application through an automated pipeline.

### **Assignment Tasks**

#### **1. Web Application Development**

- Create a simple web application (e.g., Flask, Node.js, or any framework of your choice).
- The app must return a "Hello from DevOps" message or similar.
- Host the source code in a GitHub repository .

Deliverables:

- ✓ Functional application code
- ✓ Proper project structure (e.g., `app.py`, `requirements.txt`)
- ✓ GitHub repository link

#### **2. Docker Containerization**

- Write a Dockerfile to containerize your application.
- Ensure the container runs correctly on port `5000` (or your chosen port).
- (Optional) Push the image to Docker Hub.

Deliverables:

- ✓ `Dockerfile` in the repository
- ✓ Commands used to build & run the container locally
- ✓ (Optional) Docker Hub image link

#### **3. CI Pipeline with GitHub Actions**

- Set up a GitHub Actions workflow that:
  - Triggers on `push` to the `main` branch
  - Runs a basic test (if applicable)
  - Builds the Docker image
  - (Optional) Pushes to Docker Hub on successful build

Deliverables:

- ✓ `.github/workflows/ci.yml` file
- ✓ Screenshot of a successful GitHub Actions run

## 4. CD Part and Kubernetes Deployment

- Deploy your application to a local Kubernetes cluster (k0s, Minikube, or Kind).
- Create:
  - A `deployment.yaml` for pod management
  - A `service.yaml` to expose the app (NodePort or LoadBalancer)
- Verify the app is accessible via `kubectl port-forward` or `minikube service`.

Deliverables:

- ✓ Kubernetes manifest files (`deployment.yaml`, `service.yaml`)
- ✓ Screenshot of `kubectl get pods` showing running containers
- ✓ Screenshot of the app running via Kubernetes

## 5. Documentation & Submission

- Write a `README.md` covering:
  - Project overview
  - Tools/languages used
  - How to run locally (with/without Docker)
  - Explanation of the CI/CD workflow
- Include 3–5 screenshots (app, CI pipeline, Kubernetes deployment).
- (Optional) Record a short video (2–3 min) explaining your implementation.

Deliverables:

- ✓ Well-structured `README.md`
- ✓ Screenshots embedded in documentation
- ✓ (Optional) Video walkthrough link



## Submission Guidelines

1. Share the GitHub repository URL with your mentor.
2. Ensure all deliverables are present (code, YAML files, documentation).
3. (Optional) Provide a summary (2–3 paragraphs) of:
  - What you learned
  - Challenges faced
  - Potential improvements

**Deadline: [Insert Deadline]**



## Evaluation Criteria

- ✓ Functionality (App works end-to-end)
- ✓ Automation (CI/CD pipeline executes correctly)
- ✓ Best Practices (Clean code, proper Docker/K8s configs)
- ✓ Documentation (Clear, concise, and complete)