DevOps Intern Case Study: CI/CD with Containerization and Basic Infrastructure Automation

---

Scenario:

Your team is developing a Python Flask application that serves a REST API. The application is currently hosted on a virtual machine (VM) and manually deployed after every code change.

The team now wants to:

- Containerize the application for portability.
- Automate the deployment process using a CI/CD pipeline.
- Use Infrastructure as Code (IaC) to manage resources efficiently.

Task 1: Containerize the Application

Goal:

- Create a `Dockerfile` to containerize the Flask application.
- Ensure the container runs properly and exposes the application on port `5000`.

Task 2: Setup a CI/CD Pipeline

Goal:

- Create a GitHub Actions pipeline to:
    - Build the Docker image.
    - Push the image to a container registry (DockerHub or any private registry).
    - Deploy the container on a VM using a simple script.

Task 3: Basic Infrastructure as Code (IaC) with Terraform

Goal:

- Write a basic Terraform script to:

    - Launch a virtual machine (VM).
    - Open security group ports to allow HTTP and SSH traffic.
    - Configure the VM to pull and run the Docker container.

Task 4: Basic Application Monitoring and Alerts

Goal:

- Set up monitoring to track application logs and system performance.
- Create a basic alert to notify the team if CPU usage exceeds 70%.

Deliverables:

- A GitHub repository with:
    - Dockerfile.
    - GitHub Actions workflow file.
    - Terraform scripts.
    - Basic documentation with setup and deployment instructions.

Time Allocation:

- **Estimated Time:** 4–6 hours.
- **Submission Format:** GitHub repository with detailed setup and deployment instructions.

Evaluation Criteria:

- Understanding of Docker and containerization.
- Ability to set up a CI/CD pipeline.
- Familiarity with basic Terraform commands and IaC concepts.
- Basic knowledge of monitoring and system performance.
- Problem-solving and communication skills.

Optional Bonus Task:

- Add a rollback mechanism in the CI/CD pipeline.

- Implement auto-scaling for containers if resource usage spikes.