Project Title

Empowerment Hub: A Flask-Based Motivational Web Application Featuring DevOps Practices

Project Objective

The objective of this project is to **develop, containerize, and deploy a multi-page motivational web application** using Flask. The focus is on implementing key DevOps practices such as version control, environment management, CI/CD pipelines, containerization with Docker, and deployment to a cloud platform. This project provides hands-on experience in integrating development with operations, ensuring automated, scalable, and maintainable application delivery.

Technologies Involved

- **Web Framework: Flask (Python)**
- **Programming Languages: HTML5, CSS3, JavaScript**
- **Styling and User Interface: Customized CSS**
- **Development Tools: Python 3.x, Git, Visual Studio Code**
- **Deployment: AWS EC2**

Phase 1: Application Setup & Cloud Deployment (Initial Submission)

Week 1: Introduction & Local Environment Setup

Tasks:

- Review the app structure and page flow: Home, About, Quotes, Blog, Contact.
- Set up essential tools on the local system:
  - Python, Flask, Git, Docker, AWS CLI, Terraform.
- Acquire cloud credentials or sandbox environment for deployment.
- Design basic infrastructure using Infrastructure as Code (IaC):
  - Use **Terraform** to provision cloud resources such as:
    - **EC2 instance or S3 static hosting** for Flask deployment.
    - **Security Groups, IAM roles, and VPC** configurations if required.
- Maintain all infrastructure and setup scripts in a version-controlled GitHub repository.

Week 2: Dockerization of the Flask Application

Tasks:

- Create a **Dockerfile** to containerize the Motivation Flask App.
- Define all dependencies within requirements.txt and ensure they are installed inside the image.
- Build and test the Docker image locally to validate functionality and UI rendering.

Push the finalized image to a container registry such as **AWS ECR** or **Docker Hub**.

📌 **Expected Deliverables:**

- A fully functioning local web application.
- Source code uploaded to a GitHub repository including a `README.md.`
- Screenshots of each implemented route/page.

Week 3: Deployment Using Kubernetes

Tasks:

- Write Kubernetes manifest files:
  - **Deployment**, **Service**, **Ingress**, and optionally **ConfigMaps** for environment settings.
- Deploy the app to **AWS EKS** or simulate deployment using **Minikube** for local orchestration.
- Validate routing and UI across all pages: index, about, quotes, blog, contact.
- Implement **readiness/liveness probes**, **auto-scaling**, and secure service exposure.

📌 **Expected Deliverables:**

- Fully functional **Terraform IaC scripts**, **Dockerfile**, and **Kubernetes YAML manifests**.
- Hosted version of the Motivation App accessible via a public IP/domain.
- GitHub repository containing:
  - Full source code of the app.
  - Infrastructure and deployment scripts.
  - Documentation for setup and deployment.

Deadline: 10/06/2025

Phase 2: Implementing CI/CD, Observability, & Final Submission Process (Final Submission)

Week 4: Establishing CI/CD Workflows

Tasks:

- Automate the deployment process utilizing GitHub Actions or Jenkins.
- Create workflows that will:

    - Trigger deployments upon commits to the main branch.
    - Build, test, and push the Docker image.
    - Deploy the image to S3/EC2/EKS through scripts or Infrastructure as Code (IaC).
- Ensure that AWS credentials are stored securely using GitHub secrets.

Week 5: Application Monitoring & Log Management

Tasks:

- Implement AWS CloudWatch for monitoring logs, errors, and overall performance.
- If using Kubernetes:
    - Integrate monitoring tools like Prometheus and Grafana for visual metrics.
    - Set up alerts for issues such as downtime or CPU/memory capacity thresholds.

Week 6: Final Review & Presentation Workflow

Tasks:

- Finalize documentation for inclusion in the GitHub repository.
- Conduct a live demonstration covering:
    - Workflow of the deployment process.
    - Monitoring tools utilized.
    - A walkthrough of the application, explaining its features.
- Reflect on challenges faced during development and the resolutions found.

📌 **Final Submission Includes:**

- GitHub repository containing:
    - Infrastructure scripts, CI/CD workflows, and Docker/K8s files.
    - Comprehensive deployment and monitoring instructions.
    - Screenshots of the monitoring dashboard and sample logs.
    - Final presentation slides.

Deadline: 10/07/2025

Submission & Collaboration Guidelines

Documentation Requirements

Ensure that the `README.md` file includes:

- An overview of the project.
- Descriptions of system architecture and utilized cloud services.
- A detailed guide for both local setup and cloud deployment.
- Documentation regarding CI/CD and monitoring processes.

Version Control Best Practices

Use a branching strategy like:

- Utilize structured branches for each component, e.g., `infra/terraform, ci-cd/github-actions, monitoring/setup`, etc.
- Submit pull requests for all contributions with descriptive summaries.
- Complete reviews and updates for pull requests within a 48-hour timeframe.

Project Reviews

- Code and infrastructure reviews at the end of each phase.
- Evaluation will be based on:
  - Code quality.
  - Proper use of DevOps tools.
  - Deployment success.
  - Documentation and presentation.