

Project Title

Gym Pro Professional Static Web App Deployment Using DevOps Practices

Project Objective

The goal of this project is to build and deploy a responsive and professional static website for a gym or fitness center. Interns will apply DevOps principles by containerizing the Flask-based web app, provisioning cloud infrastructure, automating deployments through CI/CD pipelines, and integrating monitoring tools to ensure performance and uptime.

Technologies Involved

- **Frontend:** HTML, CSS, Bootstrap
- **Backend:** Flask
- **Cloud Platform:** AWS (S3, EC2, ECR, EKS, CloudWatch)
- **DevOps Tools:** GitHub Actions, Docker, Terraform, Kubernetes, Prometheus, Grafana

Phase 1: Infrastructure Provisioning & Initial Hosting (First Submission)

Week 1: Project Setup & Cloud Infrastructure Design

Tasks:

1. Understand the Gym Pro application structure (HTML, CSS, Flask).
2. Install: Python, Flask, Git, AWS CLI, Terraform.
3. Set up AWS IAM credentials and CLI config.
4. Write Terraform scripts to:
 - Create EC2 instance or S3 static hosting for app.
 - Set up networking (VPC, subnets, security groups).
 - Define IAM roles/policies.

Week 2: Application Containerization Using Docker

Tasks:

1. Create a Dockerfile for the Flask application, including app dependencies and server configurations.
2. Test the container locally using Docker CLI.

3. Push the Docker image to AWS Elastic Container Registry (ECR).

Week 3: Kubernetes Deployment

Tasks:

1. Create Kubernetes manifests including:
 - Deployments, Services, Ingress, and optional ConfigMaps.
2. Deploy the containerized application to AWS EKS or a local Minikube cluster.
3. Confirm that all website routes (home, about, contact) are operational.
4. Implement readiness and liveness probes for health checks.

★ Expected Submission:

- Terraform files, Dockerfile, Kubernetes manifests.
- Hosting URL or IP (EC2/S3/K8s).
- GitHub repository containing the full project structure.

Deadline: 10/06/2025

Phase 2: CI/CD Automation & Observability (Second Submission)

Week 4: CI/CD Pipeline Integration via GitHub Actions

Tasks:

1. Create a .github/workflows pipeline to:
 - Build the application and Docker image.
 - Automatically deploy updated builds to AWS infrastructure.
 - Securely set environment variables using GitHub Secrets.
 - Validate the workflow through pushing updates and ensuring deployment success.

Week 5: Monitoring & Logging Implementation

Tasks:

1. Enable logging for the EC2 instance or CloudFront (if hosted on S3).
2. Integrate AWS CloudWatch for performance monitoring and log analysis.
3. (Optional) Utilize Prometheus and Grafana dashboards if deployed via Kubernetes.
4. Establish basic alerting policies (e.g., for downtime or high CPU usage).

Week 6: Final Submission & Presentation

Tasks:

1. Finalize the GitHub repository, ensuring it includes:
 - All infrastructure code, deployment steps, and Docker/Kubernetes files.
2. Prepare a professional presentation covering:
 - Project summary.
 - Live demonstration of the deployment.
 - Overviews of DevOps implementation.
 - Lessons learned throughout the project.

✦ Expected Submission:

- GitHub repository containing documentation and all configuration/scripts.
- Screenshots or logs of the monitoring dashboard.
- Slide deck (either in .pptx format or Google Slides).

Deadline: 10/07/2025

Submission & Collaboration Guidelines

Documentation Requirements

Maintain a well-structured README.md that includes:

- Project overview and objectives.
- Overview of the tech stack and cloud architecture.
- Step-by-step deployment guide.
- Configuration details for CI/CD and monitoring.

GitHub Workflow

- Employ a branching strategy that includes:
 - infra/terraform, feature/docker, ci/cd-workflow, etc.
- Submit pull requests for every deliverable.
- Complete code reviews and make revisions within 48 hours.

Project Reviews

- Conduct code and infrastructure reviews at the completion of each phase.
- Evaluation will be based on:
 - Quality of code.
 - Appropriate use of DevOps tools.
 - Success of the deployment.
 - Clarity and comprehensiveness of documentation and presentation.