Mid-Project: 3 Tier Application App DevOps Pipeline

A comprehensive project to build and deploy a microservices application using a complete DevOps pipeline, incorporating modern development practices, containerization, and automated deployment.

Project Overview

This project focuses on building and deploying a microservices application with separate backend and frontend components through a complete DevOps pipeline. The implementation will demonstrate modern software development practices with automated testing, deployment, and infrastructure management.

Application Components

The project consists of multiple integrated components working together to create a complete microservices application with a robust DevOps pipeline for continuous integration and deployment.

Technologies & Tools

Infrastructure & Automation

- Linux Ubuntu 24.04 LTS
- Git & GitLab
- Ansible 2.15
- Docker 24.0 & Docker Compose

Development

- Python 3.10 with Flask
- React 18
- PostgreSQL 15
- Swagger

Testing & Quality

- Jest and React Testing Library
- Pytest
- SonarQube

3 Tier Application

Backend

Flask REST API with Python 3.10

Frontend

React 18 application application

Database

PostgreSQL 15 for persistence

Documentation on

Swagger API docs with comprehensive testing

Infrastructure Setup

Server Environment

The infrastructure consists of two Ubuntu VMs:

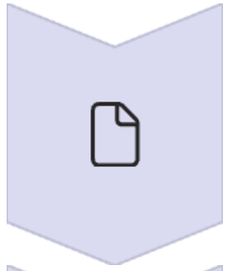
- Development/Staging server
- Production server

Ansible Automation

Ansible playbooks will handle:

- Base server configuration
- Docker installation
- PostgreSQL setup
- Application deployment

Docker Configuration

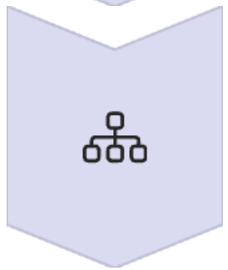


Multi-stage Dockerfile

Optimized build process with separate stages

Container Separation

Dedicated containers for Flask, React, and PostgreSQL



Docker Compose

Orchestration for local development environment

CI/CD Pipeline Stages

Build

Lint Python and JavaScript code, run unit tests, build Docker images

Docker images

Test

Execute comprehensive unit tests

Security Scanning

Integrate Trivy for vulnerability detection and reporting

Deploy

Deploy to staging, run smoke tests, promote to production

Notify

Slack integration for pipeline status notifications

Deliverables

1 **Documentation**

Comprehensive Git repository with detailed documentation

2 **Infrastructure Code**

Ansible playbooks, Dockerfiles, and Docker Compose Compose configurations

3 **Pipeline Configuration**

GitLab CI/CD pipeline with Slack integration

4 **Architecture**

Detailed architecture diagrams and documentation