



Himanshu Shrivastava
MT2024059
Shivam Pandey
MT2024145

GitHub Repo Link: https://github.com/Himanshu-Shrivastava1729/DevOps_MajorProject.git

Automated MLOps Pipeline: A DevOps Framework for Loan Eligibility Prediction:

In modern software development, the need for rapid, reliable, and scalable deployment pipelines is critical. This project implements a fully automated DevOps framework that integrates Continuous Integration/Continuous Deployment (CI/CD), container orchestration, and real-time monitoring to streamline the Software Development Life Cycle (SDLC) for an **MLOps**-based application.

The solution leverages industry-standard tools, including:

- Git & GitHub (Version Control)
- Jenkins (CI/CD Automation)
- Docker & Docker Compose (Containerization)
- Ansible (Configuration Management)
- Kubernetes (K8s) (Orchestration & Scaling)
- ELK Stack (Monitoring & Logging)

Feature	Tools Used	Description
Version Control	Git, GitHub	Codebase managed with Git; changes trigger CI/CD pipelines.
CI/CD Automation	Jenkins, GitHub Webhooks	Jenkins pipelines build, test, and deploy on Git push.
Containerization	Docker, Docker Compose	Backend (ML model) and Frontend containerized.
Configuration Mgmt.	Ansible (Roles & Playbooks)	Automated deployment using modular Ansible roles.
Orchestration	Kubernetes (k8s)	Deployed with k8s manifests (Deployments, Services, HPA).
Monitoring & Logging	ELK Stack (Elasticsearch, Logstash, Kibana)	Logs visualized in Kibana dashboard.

Advanced Features Implemented:

To enhance security, scalability, and reliability, the following advanced functionalities were incorporated:

- **Ansible Vault** – Secured sensitive credentials (Docker Hub password) using encrypted storage.
- **Modular Ansible Roles** – Improved reusability and maintainability of deployment scripts.
- **Kubernetes Horizontal Pod Autoscaling (HPA)** – Automated scaling of backend pods based on CPU/memory usage.
- **Live Logging with ELK Stack** – Enabled real-time log analysis via Kibana dashboards/Discovery.
- **MLOps Integration** – Deployed a machine learning model as a scalable microservice, demonstrating domain-specific DevOps adaptation.

Innovation: Liveness & Readiness Probes:

- **Liveness Probe:** Ensures the application restarts if it becomes unresponsive.
- **Readiness Probe:** Ensures traffic is sent only to healthy pods.
- **Implementation:** Added in Kubernetes deployment manifests (backend-deployment.yaml).

Workflow:

- Code Commit → GitHub Webhook → Jenkins Pipeline
- Jenkins fetches code, runs tests, builds Docker images, and pushes to Docker Hub.
- Ansible Playbook Execution
- Deploys the application on Kubernetes using deploy.yml.
- Kubernetes Orchestration
- Pods are managed via Deployments, Services, and HPA.
- ELK Stack Integration
- Logstash collects logs → Elasticsearch stores them → Kibana visualizes.

Domain-Specific Implementation (MLOps):

Loan Eligibility Prediction Engine:

The backend of this project is a Python-based ML service (ml_project.py) that predicts whether a user is eligible for a bank loan based on:

Input Features:

- Age
- Income
- Credit score
- Existing loan status
- Homeownership status (owns house/rents)
- Employment
- Output: Binary classification (Eligible or Not Eligible).
- Etc

Machine Learning Models:

Three models were trained:

- Gaussian Naive Bayes – For probabilistic classification.
- Decision Tree – Rule-based eligibility assessment.
- Logistic Regression – Baseline model for binary outcomes.

MLOps Integration:

Dockerized Microservice:

- Packaged as a Flask REST API.
- Accepts JSON input (user data) and returns eligibility predictions.

Kubernetes Deployment:

- Scalable via backend-deployment.yaml (replicas managed by HPA).
- Served behind a backend-service.yaml (ClusterIP).

CI/CD Pipeline:

- Jenkins rebuilds the ML Docker image on code updates.

Monitoring

- ELK Stack: Logs prediction requests/responses for auditability.
- Kibana Dashboard: Tracks model performance metrics.

Key Workflow:

- User submits data → Flask API process input → Model predicts eligibility.
- Kubernetes scales pods if request volume increases (HPA).
- Logstash streams logs to Elasticsearch for real-time analysis.

Conclusion:

This project represents a comprehensive implementation of DevOps methodologies applied to an MLOps solution for financial decision-making. By establishing an automated pipeline that integrates continuous integration, container orchestration, and infrastructure-as-code practices, we have demonstrated how modern software engineering principles can be effectively employed in machine learning applications.

The system's ability to process loan eligibility requests through multiple machine learning models - Gaussian Naive Bayes, Decision Tree, and Logistic Regression - while maintaining scalability and reliability through Kubernetes, illustrates the practical convergence of DevOps

and data science. The inclusion of production-grade features such as health probes, horizontal pod autoscaling, and secure credential management reflects industry best practices for operationalizing machine learning systems.