SmartLearn AI LLMOps System Architecture

A Fully Automated CI/CD and MLOps Pipeline for AI-driven Learning and Quiz Generation

Andrew Adel Labib

AI & NLP Engineer — Generative AI Engineer

andrewadellabib77@gmail.com

LinkedIn: andrew-adel-b865b1244

Date: August 2025 – September 2025

Overview

SmartLearn AI LLMOps System is a fully automated CI/CD and MLOps pipeline designed to streamline the deployment of AI-driven learning and quiz-generation applications. It integrates **GitHub**, **Jenkins**, **Docker Hub**, **ArgoCD**, **Groq AI**, **LangChain**, and **Kubernetes**, deployed on **AWS EC2 instances** using **NodePort** services.

This project demonstrates a complete AI/ML lifecycle — from **code commit → build → containerization → deployment → production inference** — powered by **Groq AI acceleration** and **LangChain orchestration**.

Features

- End-to-end CI/CD automation using Jenkins, ArgoCD, and Webhooks.
- AI-powered quiz generator based on user-provided topics.
- LangChain for orchestrating LLM logic and structured question generation.
- Groq AI LPUs for lightning-fast inference.
- Kubernetes orchestration on AWS EC2 (NodePort).
- Dockerized architecture for portability.
- GitHub Webhook integration for instant Jenkins triggers.
- ArgoCD GitOps synchronization for automated deployments.
- Real-time evaluation and results generation inside the app.

Project Idea: SmartLearn AI Quiz Generator

The core idea of SmartLearn AI is to provide an **interactive AI-powered quiz generator**. Users input a topic, question type, difficulty level, and number of questions, and the app automatically generates a custom quiz using **LLMs through LangChain**.

Supported Question Types

- Multiple Choice
- True/False
- Fill in the Blank
- Short Answer
- Descriptive

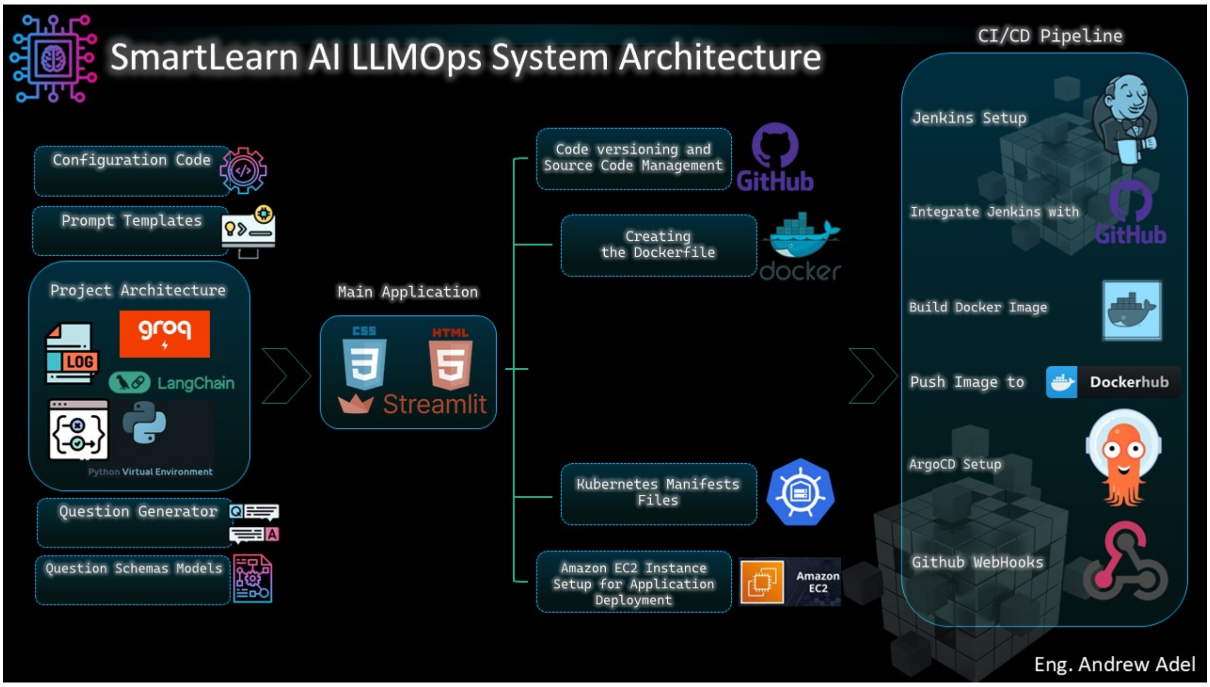
- Ordering
- Multi-Select
- Numerical

After generating the quiz:

- Users can answer interactively.
- The system evaluates responses and shows the score and feedback.
- Results can be saved and downloaded as CSV.

This AI quiz app is served via **Streamlit**, deployed through an automated CI/CD pipeline built with **Jenkins**, **Docker**, **ArgoCD**, and **Kubernetes**.

System Architecture



System Workflow

1. Code Push to GitHub → triggers Webhook → notifies Jenkins
2. Jenkins Pipeline → pulls code, builds Docker image, pushes to Docker Hub
3. Docker Hub → stores versioned images
4. ArgoCD → monitors GitHub manifests and deploys to Kubernetes
5. Kubernetes Cluster (AWS EC2) → runs the app using NodePort for public access

Setup and Installation

1 Clone Repository

Clone Repository

```
git clone https://github.com/andrew-adel-labib/SmartLearn-AI-LLMOps-AIOps-Cloud-System-using-Groq-LangChain-Docker-Jenkins-ArgoCD-Minikube-AWS-EC2.git

cd SmartLearn-AI-LLMOps-AIOps-Cloud-System-using-Groq-LangChain-Docker-Jenkins-ArgoCD-Minikube-AWS-EC2
```

2 Launch AWS EC2 Instance

Use **Ubuntu 22.04 (t2.medium or higher)** and open the following ports:

Port	Purpose
22	SSH
9595	Jenkins
31704	ArgoCD
9090	SmartLearn App
30000–32767	Kubernetes NodePort range

3 Install Dependencies

Install Dependencies

```
sudo apt update && sudo apt install -y docker.io kubectl minikube
sudo systemctl start docker
sudo systemctl enable docker
minikube start --driver=docker
```

Docker Build & Push Commands

Docker Commands

```
docker build -t <dockerhub-user>/smartlearn-ai:latest .
docker login
docker push <dockerhub-user>/smartlearn-ai:latest
docker run -d -p 8080:8080 <dockerhub-user>/smartlearn-ai:latest
```

Jenkins Integration (CI)

Run Jenkins on EC2

Run Jenkins Container

```
sudo docker pull jenkins/jenkins:lts
sudo mkdir -p /home/ubuntu/jenkins_home
sudo chown 1000:1000 /home/ubuntu/jenkins_home

sudo docker run -d \
  --name jenkins \
  -p 9595:8080 \
  -p 50000:50000 \
  -v /home/ubuntu/jenkins_home:/var/jenkins_home \
  jenkins/jenkins:lts
```

Access Jenkins: `http://<EC2-Public-IP>:9595`

GitHub Webhook Setup

- Go to GitHub → Settings → Webhooks → Add Webhook
- Payload URL: `http://<EC2-IP>:9595/github-webhook/`
- Content type: `application/json`
- Trigger: “Just the push event”
- Enable “GitHub hook trigger for GITScm polling”

Jenkinsfile Example

Jenkinsfile Example

```
pipeline {
  agent any
  stages {
    stage('Checkout') {
      steps {
    git branch: 'main', url: 'https://github.com/<user>/<repo>.git'
      }
    }
    stage('Build Docker Image') {
      steps {
        sh 'docker build -t <dockerhub-user>/smartlearn-ai:latest .'
      }
    }
    stage('Push to Docker Hub') {
      steps {
        withCredentials([string(credentialsId: 'dockerhub-pass',
        variable: 'DOCKER_PASS')]) {
          sh 'echo $DOCKER_PASS | docker login -u <dockerhub-user>
          --password-stdin'
          sh 'docker push <dockerhub-user>/smartlearn-ai:latest'
        }
      }
    }
  }
}
```

ArgoCD Deployment (CD)

Install ArgoCD

```
kubectl create namespace argocd
kubectl apply -n argocd -f \
https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/
install.yaml
```

Expose via NodePort:

Expose ArgoCD Server

```
kubectl patch svc argocd-server -n argocd \
  -p '{"spec": {"type": "NodePort", "ports": [{"port": 80,
  "targetPort": 8080, "nodePort": 31704}]}}'
```

Access ArgoCD: `http://<EC2-Public-IP>:31704`

Tech Stack

Tool / Service	Purpose	Description
LangChain	LLM Orchestration Framework	Connects large language models to backend logic.
Groq AI	AI Inference Accelerator	Provides ultra-fast inference using LPUs.
GitHub	Source Control	Hosts code and triggers Jenkins builds.
Jenkins	Continuous Integration	Automates build, test, and deployment.
Docker	Containerization	Packages app and dependencies.
ArgoCD	Continuous Deployment	Syncs manifests to Kubernetes.
AWS EC2	Cloud Infrastructure	Hosts Jenkins, ArgoCD, and Kubernetes.