Scalable Multi-Agent LLMOps System

Leveraging Groq LPU, Tavily Search, LangGraph, FastAPI, Streamlit, SonarQube, Jenkins, and AWS Cloud Deployment

Andrew Adel

AI & NLP Engineer — Generative AI Engineer

andrewadellabib77@gmail.com

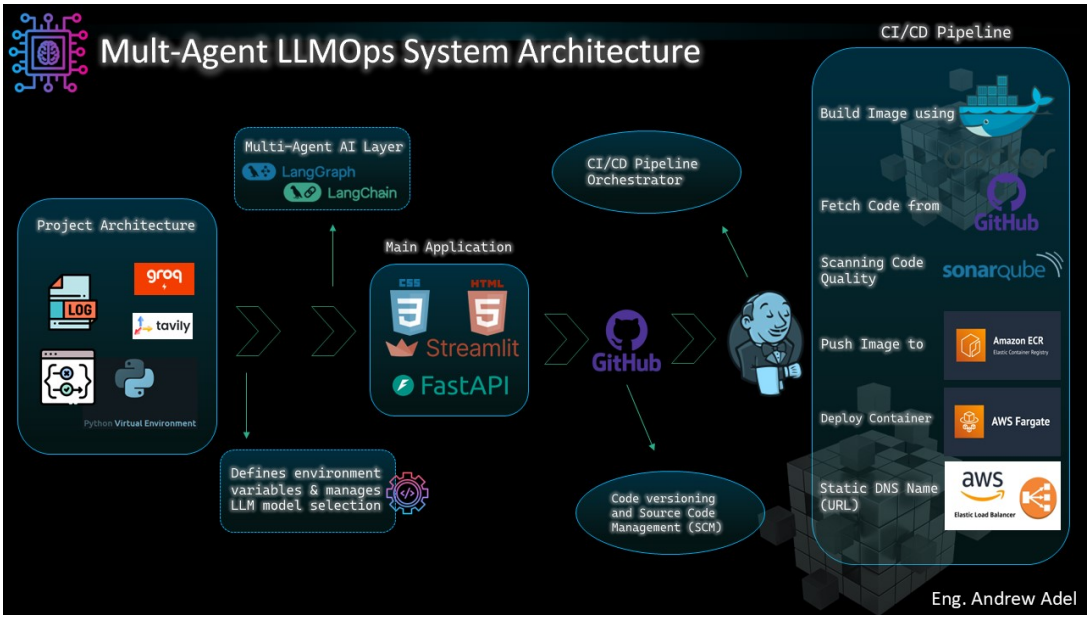
LinkedIn Profile

Date: September 2025 – October 2025

Overview

This project is a **production-ready Multi-Agent LLMOps System** designed for scalable, automated AI agent deployment and management. It integrates **Groq LPU** for high-speed inference, **Tavily Search** for real-time web reasoning, and **LangGraph** for multi-agent orchestration — all powered by **FastAPI**, **Streamlit**, and a complete **CI/CD pipeline** with **Jenkins**, **SonarQube**, and **AWS Fargate**.

System Architecture



Workflow Summary

1. Agent Development & Core Logic

- Multi-agent orchestration via **LangGraph**.
- Agents perform reasoning and data retrieval using **Tavily API**.
- **Groq LPU** delivers lightning-fast inference and model execution.

2. Service & Interface Layer

- **Backend:** FastAPI exposes RESTful and streaming endpoints.
- **Frontend:** Streamlit dashboard provides real-time interaction and monitoring.

3. CI/CD & Deployment

- Jenkins automates build and deploy pipelines.
- SonarQube ensures code quality with bug and vulnerability scanning.
- Docker containers are built and pushed to AWS ECR.
- AWS Fargate runs the containers serverlessly.
- AWS Load Balancer manages traffic and scalability.

Tech Stack

Category	Technology	Description
LLM Inference	Groq LPU	Low-latency model inference
Web Search	Tavily API	Real-time knowledge augmentation
Agent Framework	LangGraph, LangChain	Multi-agent orchestration
Backend	FastAPI	High-performance asynchronous API
Frontend	Streamlit	Interactive web dashboard
CI/CD	Jenkins	Continuous integration and delivery
Code Quality	SonarQube	Static code analysis
Containerization	Docker	Environment consistency
Cloud Deployment	AWS ECR, Fargate, Load Balancer	Scalable serverless deployment

Setup Instructions

1. Clone Repository

Git Clone Commands

```
git clone https://github.com/andrew-adel-labib/Scalable-Multi-Agent
-LLMOps-System-with-Groq-Tavily-LangGraph-FastAPI-Streamlit-CICD
-AWS-Deployment.git
```

Git Clone Commands

```
cd Scalable-Multi-Agent-LLMOps-System-with-Groq-Tavily-LangGraph
-FastAPI-Streamlit-CICD-AWS-Deployment
```

2. Install Dependencies

Install Requirements

```
pip install -r requirements.txt
```

3. Run Application

Run Backend and Frontend

```
python app/main.py
```

4. Build Docker Image

Docker Commands

```
docker build -t multi-agent-llmops .
docker run -p 8000:8000 multi-agent-llmops
```

5. Deploy via Jenkins → AWS

- Fetch latest code from GitHub
- Run SonarQube quality scan
- Build and tag Docker image
- Push image to AWS ECR
- Deploy on AWS Fargate
- Integrate with AWS Load Balancer

Monitoring & Reporting

SonarQube Dashboard: http://172.25.167.174:9000/dashboard?id=Multi-Agent-LLMOps

- Reports on bugs, code smells, vulnerabilities, and maintainability.

Example Use Cases

- Multi-agent web research and summarization
- Code generation and validation with Groq inference
- Knowledge-augmented reasoning using Tavily API
- Automated LLMOps pipelines with CI/CD and quality gates