

**Intermediate Generative AI Engineering for LLMOps**

**Course Number:** AI-126WA
**Duration:** 2 days

**Overview**

This Intermediate Generative AI (GenAI) course is for DevOps and ITOps professionals who want to advance their GenAI skills with deployment strategies and best practices for building large language model (LLM) applications. Participants master popular tools and frameworks, including Docker, Kubernetes, and cloud platforms, in an LLM environment.

**Prerequisites**

* Practical Python programming and scripting for automation tasks (6+ months)
	+ API call access and event stream handling
	+ Exception handling, debugging, testing, and logging
* Experience with containerization technologies (e.g., Docker) and orchestration platforms (e.g., Kubernetes)
* Familiarity with CI/CD pipelines and tools, such as Jenkins, GitLab, or GitHub Actions
* Knowledge of cloud platforms (e.g., AWS, GCP, Azure) and their services
* Experience with monitoring and logging tools, such as Prometheus, Grafana, and ELK stack (Elasticsearch, Logstash, Kibana) is recommended but not required
* Machine Learning concepts recommended - classification, regression, clustering

**Materials**

All Generative AI training students receive comprehensive courseware.

**Software Needed on Each Student PC**

All attendees must have a modern web browser and an Internet connection.

**Objectives**

* Package LLM applications seamlessly into containers using Docker for efficient deployment and portability
* Orchestrate LLM containers with Kubernetes for scalable, resilient, and automated management
* Implement robust strategies to scale your LLM applications, ensuring optimal performance under varying workloads
* Monitor and troubleshoot LLM applications effectively, tracking key performance metrics and addressing potential issues proactively
* Implement automated testing and robust security measures to ensure that LLM applications are reliable, safe, and compliant while optimizing costs

**Outline**

* Introduction
* Containerization and Orchestration
	+ Containerizing LLM applications using Docker
	+ Orchestrating LLM containers using Kubernetes
	+ Deploying an LLM application using Docker and Kubernetes
* Scaling LLM Applications
	+ Strategies for horizontal and vertical scaling
	+ Load balancing and auto-scaling techniques
	+ Implementing auto-scaling for an LLM application
* Monitoring and Troubleshooting
	+ Key performance metrics for LLM applications
	+ Automated Testing for LLMOps
		- Differences of LLMOps testing and traditional software testing
		- Evaluation using CI/CD Tools
	+ Evaluating LLM problems like hallucinations, data drift, unethical/harmful outputs
	+ Monitoring tools and techniques (e.g., Weights and Biases, CircleCI)
		- Setting up monitoring for an LLM application
		- Creating dashboards and alerts for key metrics
* Security, Compliance, and Cost Optimization
	+ Securing LLM application infrastructure and data
	+ Ensuring compliance with relevant regulations and standards
	+ Strategies for optimizing resource usage and costs in cloud-based LLM deployments
* Conclusion