Faysal Aziz

Agentic AI Development

Build, Deploy, and Scale Autonomous Python Agents with LangChain - 3 days

FA-0045 · 3 days

Agentic & Generative AI · DevOps, Cloud & Infrastructure

Introduction

The automation landscape is undergoing a fundamental shift as agentic AI emerges as the next frontier in intelligent software development. Unlike traditional automation tools, these autonomous agents can interact with systems in real-time, adapt their behavior through continuous learning, and seamlessly integrate with existing APIs and databases to create truly dynamic solutions.

This intensive course is designed for experienced Python developers who are ready to master the cutting-edge technologies driving this revolution. You'll gain hands-on experience building sophisticated agentic systems using LangChain, local Large Language Models, the Model Context Protocol (MCP), and Retrieval-Augmented Generation (RAG) – all within familiar Linux deployment environments.

What sets this course apart is its relentless focus on practical implementation. Led by an instructor with over 30 years of industry experience, every lesson is grounded in real-world, market-driven use cases rather than academic theory. You'll work with scenarios that mirror actual enterprise challenges, ensuring that the autonomous systems you build are production-ready from day one.

Learning outcomes

By completing this course, participants will:

- Understand core concepts and emerging trends in agentic AI.
- Build and deploy autonomous agents with LangChain.
- Integrate MCP for effective tool and API integration.
- Implement and optimize RAG pipelines for knowledge retrieval.

- Retrain and augment LLMs with custom knowledge sources.
- Deploy and manage agentic applications in Linux-based production environments.
- Execute and safely manage real-time execution of code generated by LLMs.

Prerequisites

- Expert-level Python proficiency.
- Proficient Linux system administration and deployment skills.
- Familiarity with LLM frameworks (OpenAI, Ollama, Llama, HuggingFace).
- Basic experience with LangChain concepts (chains, agents, retrievers).
- Working knowledge of vector databases, RESTful APIs, and containerization tools (Docker).

Training outline

1. 1. Deep Dive into Agentic AI

- Defining agentic AI: autonomous agents vs. generative models.
- Use-cases across industries: automation, personalization, real-time decision-making.
- Ethical, security, and governance considerations (TRiSM framework).

2. 2. Advanced LangChain Fundamentals

- Refresher: chains, agents (Zero-shot, ReAct, Tool-enabled), retrievers.
- Advanced agent patterns: Agent-to-agent communication, multi-step workflows.
- Lab: Building a complex LangChain workflow for automation.

3. 3. Retrieval-Augmented Generation (RAG)

- Core concepts: Combining retrieval systems with LLM-generated responses.
- Implementing effective retrievers and embedding pipelines.
- Lab: Build a robust RAG pipeline for agents using LangChain.

4. 4. Model Context Protocol (MCP) in Depth

- Understanding MCP and its role in agentic AI.
- Integrating MCP with LangChain for seamless agent-to-tool communication.
- Lab: Connecting APIs and data sources via MCP adapters.

5. 5. Multi-Agent Systems and Real-Time Coordination

- Designing multi-agent workflows and communication strategies.
- MCP-enabled inter-agent communication patterns.
- Real-time decision-making and task orchestration among agents.
- Lab: Multi-agent simulation with agent orchestration using LangChain.

6. 6. Deploying Local LLMs and Agentic Systems on Linux

- Overview of local models (Llama, Ollama, HuggingFace).
- Deploying local LLMs and agents in Linux environments.
- Best practices for Linux deployments: monitoring, logging, containerization (Docker).
- Lab: Deploying a local agentic solution on a Linux server.

7. 7. Customizing and Retraining LLMs

- Methods for retraining or fine-tuning local LLMs (PEFT, LoRA, QLoRA).
- Adding domain-specific knowledge through fine-tuning or prompting techniques.
- Lab: Fine-tuning a local LLM using custom datasets for targeted agentic responses.

8. 8. Executing Real-Time Code Generated by LLMs

- Safe and effective execution of dynamic code generated by agents.
- Sandboxing, validation, and execution strategies for generated scripts.
- Security considerations when executing LLM-generated code.
- Lab: Building a dynamic, agent-driven code execution environment on Linux.

9. 9. Production Deployment and Scalability

- Transitioning from development to production: considerations and strategies.
- Using container orchestration tools (Docker, Kubernetes) for scaling agentic applications.
- Monitoring and governance of deployed agentic solutions (security, logging, alerting).

- Continuous improvement and model retraining strategies.

10. 10. Capstone Project: Autonomous Agentic System

- Project: Build and deploy a comprehensive, autonomous agentic application.
- Scenario: Automated data analysis agent executing real-time Python scripts.
- Integrate RAG and MCP, deploying on a scalable Linux infrastructure.
- Demonstrate retraining capability for continuous learning.

11. 12. Metrics and Performance evaluation (Overview)

- Holistic approach
- Quantitative method
- Qualitative Metrics
- Evaluation Techniques
- Agent-Specific Considerations

© 2026 Faysal Aziz · Kuala Lumpur