Faysal Aziz

Agentic AI for Technical Professionals

Building Practical AI Agent Systems for professional deplyment - 2 days

FA-0047 · 2 days

Agentic & Generative AI · Software Development · DevOps, Cloud & Infrastructure

---

Introduction

Agentic AI is becoming one of the most important areas in modern software engineering, infrastructure automation, enterprise platforms, and intelligent operations. Organizations are rapidly moving beyond simple chatbot implementations toward autonomous and semi-autonomous systems capable of planning, tool usage, workflow orchestration, reasoning, event processing, and multi-agent collaboration. Technical professionals are increasingly expected to understand how these systems are designed, deployed, governed, and integrated into real production environments.

This course focuses on the practical implementation of Agentic AI systems using Python on Linux environments. Participants will work with modern architectural approaches including simple agent pipelines, multi-agent orchestration, Kafka-based distributed workflows, cloud-hosted LLM APIs, on-premise SLM and LLM deployments, quantization techniques, Human-in-the-Loop workflows, and MCP-based interoperability patterns. The delivery emphasizes real engineering workflows and operational practices rather than purely theoretical or academic discussions.

The course is delivered by an instructor with over 30 years of industry experience using practical, industry-driven implementation approaches aligned with current enterprise adoption patterns and operational requirements.

Learning outcomes

- Understand Agentic AI architecture and workflows
- Build simple agentic pipelines using Python
- Design multi-agent orchestration systems
- Implement Kafka-based distributed agent workflows

- Deploy and operate on-premise SLMs and LLMs
- Understand and apply quantization techniques
- Integrate cloud-hosted LLM APIs into agent systems
- Implement Human-in-the-Loop workflows
- Understand MCP architecture and interoperability
- Build Linux-based AI development environments
- Apply operational, security, and governance best practices
- Design scalable enterprise agentic systems

Prerequisites

- Basic Python programming knowledge
- Familiarity with Linux command-line operations
- Basic API knowledge
- Basic networking concepts
- General software development experience
- Familiarity with JSON and YAML

Training outline

1. Introduction to Agentic AI

1. Agentic AI concepts
2. Autonomous workflows
3. Agents versus traditional AI systems
4. Agent lifecycle
5. Planning and reasoning
6. Tool usage
7. Context management
8. Agent memory concepts
9. Enterprise use cases
10. Current industry trends

2. Professional Environment Preparation

1. Python installation

2. Virtual environments
3. Package management
4. GPU considerations
5. CUDA overview
6. Linux AI tooling
7. IDE and editor setup
8. Environment configuration
9. Logging setup
10. Project structures

3. Python Foundations for Agentic AI

1. API integration
2. REST workflows
3. JSON handling
4. Async processing
5. Event-driven programming
6. Error handling
7. Retry logic
8. Configuration management
9. Secrets management

4. Creating a Simple Agentic Pipeline

1. Agent loop creation
2. Prompt handling
3. Tool registration
4. Tool invocation
5. Function calling
6. Context management
7. State management
8. Response handling
9. Structured outputs
10. Error handling
11. Retry workflows
12. Logging and tracing

5. Working with Cloud LLM APIs

1. Cloud LLM providers
2. API authentication
3. API request handling
4. Streaming responses
5. Function calling APIs
6. Context window management
7. Multi-model workflows
8. Rate limiting
9. Token management
10. Cost considerations
11. Security considerations

6. Human-in-the-Loop (HITL)

1. HITL concepts
2. Approval workflows
3. Manual intervention models
4. Escalation workflows
5. Human review pipelines
6. Workflow interruption handling
7. Confidence scoring
8. Governance considerations
9. Audit logging
10. Operational controls

7. Multi-Agent Systems

1. Multi-agent architecture
2. Agent orchestration
3. Planner-executor patterns
4. Hierarchical agents
5. Shared memory approaches
6. Agent communication
7. Event-driven workflows
8. Distributed orchestration

9. Workflow coordination
10. Failure handling

8. Creating Multi-Agent Pipelines with Microservices and Kafka

1. Microservices architecture
2. Service decomposition
3. Stateless services
4. API gateway concepts
5. Kafka fundamentals
6. Topics and partitions
7. Producers and consumers
8. Event streaming
9. Message serialization
10. Python Kafka integration
11. Agent-to-agent messaging
12. Distributed task execution
13. Event choreography
14. Retry patterns
15. Dead-letter queues
16. Workflow resiliency
17. Observability and tracing

9. On-Premise SLM and LLM Deployment

1. SLM concepts
2. LLM concepts
3. Model selection
4. CPU versus GPU deployment
5. Local inference
6. Model serving
7. Linux deployment considerations
8. Containerized deployment
9. Resource management
10. Performance considerations

11. Security considerations
12. Air-gapped deployments

10. Quantization

1. Quantization fundamentals
2. FP16
3. INT8
4. INT4
5. GPTQ
6. AWQ
7. GGUF
8. Dynamic quantization
9. Static quantization
10. Memory optimization
11. CPU optimization
12. GPU optimization
13. Accuracy trade-offs
14. Benchmarking
15. Deployment considerations

11. Model Context Protocol (MCP)

1. MCP concepts
2. MCP architecture
3. MCP clients
4. MCP servers
5. Tool interoperability
6. Context exchange
7. Session handling
8. Tool registration
9. Python integration
10. MCP service design
11. Security considerations
12. Governance considerations

12. Security and Governance

1. Prompt injection risks
2. Unauthorized tool execution
3. Context poisoning
4. Credential management
5. Sandboxed execution
6. Access controls
7. Policy enforcement
8. Secure API integration
9. Auditability
10. Observability
11. Compliance considerations
12. Operational governance

13. Monitoring and Operational Reliability

1. Logging
2. Distributed tracing
3. Agent telemetry
4. Workflow monitoring
5. Failure handling
6. Retry workflows
7. Recovery strategies
8. Performance monitoring
9. Scalability considerations
10. Operational best practices

14. Capstone Architecture Walkthrough

1. Single-agent architecture
2. Multi-agent architecture
3. Kafka orchestration workflows
4. HITL integration
5. MCP interoperability
6. Hybrid cloud and on-prem models
7. Quantized inference deployment

8. Operational architecture patterns

Disclaimer

This training outline is intended solely as a general guideline for course delivery. The trainer reserves the right to modify, reorganize, expand, reduce, or amend the content, technologies, tooling, sequence of topics, demonstrations, and technical depth based on participant requirements, operational considerations, software updates, infrastructure availability, and evolving industry practices without prior notice.

© 2026 Faysal Aziz · Kuala Lumpur