Faysal Aziz

Agent Systems Engineering

From autonomous agent loops to production release gates in 1 intensive day or 2 days

FA-0044 · 2 days

Agentic & Generative AI · DevOps, Cloud & Infrastructure

Introduction

Software teams are no longer choosing a single AI assistant; they are deciding how much agency each tool should receive, what systems it may access and where human approval must remain mandatory. This advanced course develops that judgement while covering the engineering foundations behind autonomous agents, graph-based orchestration, model customisation, security response and pre-deployment scanning.

Current tooling is reflected throughout the course. ChatGPT now separates **Chat, Work and Codex**: Work handles longer multi-step deliverables, reusable Skills, project context and scheduled workflows, while Codex remains focused on repositories, coding and technical execution. AWS Bedrock AgentCore provides managed agent loops, runtime isolation, gateways, memory, policy controls, Guardrails, observability and evaluations. Google currently positions Gemini 3.5 Flash for agentic workflows and Nano Banana 2, Nano Banana 2 Lite and Nano Banana Pro for image generation and editing. Claude Code supports isolated worktrees, parallel sessions and coordinated agent teams.

The instructor has over 30 years of industry experience and will use current, industry-demanded engineering practices rather than an academic treatment of the subject. Approximately six to seven instructional hours can provide a controlled end-to-end implementation overview, but not production mastery of every platform. Because full model-customisation jobs may require several hours, training will concentrate on data preparation, configuration, job submission, monitoring and evaluation using pre-staged resources.

Target roles: Software development managers, front-end managers, senior and junior software developers, cloud engineers and QA automation engineers.

Learning outcomes

Participants will be able to:

- Select ChatGPT Chat, Work, Codex, Claude Code, Gemini and AWS Bedrock capabilities for appropriate engineering tasks.
- Design a bounded autonomous agent with tools, state, memory, validation and termination controls.
- Structure agentic workflows using loops, state graphs, branching, checkpoints and human approval nodes.
- Prepare and evaluate a controlled open-weight model-customisation workflow.
- Identify major agentic AI threats and design an orchestrated containment and response process.
- Establish mandatory pre-deployment security and quality gates for AI-generated code.

Mandatory Prerequisites

All prerequisites are compulsory. Participants who fail the technical-readiness check should not attend this advanced course.

- **Linux or WSL2 is mandatory. Native Windows development environments, PowerShell-only setups and WSL1 are not supported.**
- Ability to work independently in the Linux shell, including files, permissions, processes, environment variables and package management.
- Professional proficiency in Python or TypeScript/Node.js.
- Previous delivery of at least one working API-based application.
- Working knowledge of asynchronous programming, REST APIs, JSON schemas, authentication and error handling.
- Practical Git and GitHub experience, including branches, pull requests, rebasing and resolving conflicts.
- Working knowledge of Docker, CI/CD pipelines, automated testing and cloud IAM concepts.
- Basic experience calling at least one commercial language-model API.
- Authorised access to an AWS account with Amazon Bedrock and AgentCore permissions.
- Authorised access to ChatGPT Work and Codex, Claude Code and the Gemini API.
- Access to a GitHub repository with permission to configure Actions, required checks and security scanning.
- Git, Docker, AWS CLI, Python or Node.js, VS Code and all trainer-specified dependencies installed before arrival.
- Successful completion of the trainer-provided repository build, test, API-access and AWS-permission checks.
- A non-confidential training repository; production secrets and proprietary customer data are prohibited.

Training outline

1. Advanced AI Tool Selection and Operating Modes

ChatGPT Operating Options

1. Chat for conversational assistance
2. Work for multi-step deliverables
3. Projects and persistent context
4. Skills and reusable workflows
5. Scheduled and trigger-based work
6. Connected applications and permission controls
7. Codex for repository and terminal operations

Claude Code Development Workflows

1. Repository-level context
2. Local and cloud execution
3. Worktrees and isolated changes
4. Parallel development sessions
5. Agent teams and delegated tasks
6. Tool permission boundaries

Gemini and AWS Bedrock Capabilities

1. Gemini 3.5 Flash agentic workflows
2. Nano Banana 2 image generation
3. Nano Banana Pro visual workflows
4. Bedrock model selection
5. AgentCore service selection
6. Task, risk, latency and cost alignment

2. Autonomous Agent and Graph Engineering

Bounded Agentic Loops

1. Goal and constraint definition

2. Observe, plan, act and verify lifecycle
3. Tool schemas and structured outputs
4. Context and memory boundaries
5. Iteration budgets and stop conditions
6. Failure and fallback paths

Graph-Based Orchestration

1. Shared state schemas
2. Nodes, edges and conditional routing
3. Deterministic and model-driven nodes
4. Parallel branches and subagents
5. Checkpointing and durable execution
6. Retries and recovery paths
7. Human-in-the-loop interrupts

AWS Bedrock AgentCore Architecture

1. Harness and managed agent loops
2. Runtime and session isolation
3. Gateway and tool connectivity
4. MCP and A2A integration
5. Short-term and long-term memory
6. Identity and least-privilege access
7. Tracing, evaluation and optimisation

3. Open-Weight Model Customisation

Customisation Decision Framework

1. Prompt engineering
2. Retrieval-augmented generation
3. Tool augmentation
4. Supervised fine-tuning
5. Reinforcement fine-tuning
6. Open-weight licensing considerations

Training Data Preparation

1. Dataset objectives
2. JSONL data structures
3. Training and validation separation
4. Data quality and deduplication
5. Sensitive-data removal
6. Data and model provenance

Fine-Tuning and Evaluation Workflow

1. Base-model selection
2. Parameter-efficient tuning concepts
3. Hyperparameter configuration
4. Reward and grading functions
5. Training jobs and checkpoints
6. Baseline and regression evaluation
7. Model versioning and rollback
8. Bedrock custom model import

4. Agentic AI Security and Response Orchestration

Agent Threat Modelling

1. Direct and indirect prompt injection
2. Tool, Skill and MCP poisoning
3. Sensitive-information disclosure
4. Excessive agency
5. Improper output handling
6. Data and model poisoning
7. Agent supply-chain risks

Preventive Security Controls

1. Agent trust boundaries
2. Runtime sandboxing
3. Scoped credentials and tool allowlists

4. Network and egress restrictions
5. Input and output validation
6. Bedrock Guardrails and gateway policies
7. Approval gates for consequential actions

Detection and Response Workflow

1. Agent trace and tool-call monitoring
2. Threat classification and severity
3. Session isolation
4. Credential revocation
5. Tool and connector suspension
6. Evidence and audit-log preservation
7. Human escalation and recovery approval

5. Pre-Deployment Code Scanning and Release Control

AI-Generated Code Review

1. Claude Code and Codex review boundaries
2. Independent writer and reviewer agents
3. Diff and test-result inspection
4. Human ownership of accepted changes

Automated Security Gates

1. Static application security testing
2. CodeQL pull-request scanning
3. Dependency and licence review
4. Secret scanning and push protection
5. Infrastructure-as-Code scanning
6. Container and package scanning

Protected Release Workflow

1. Required pull-request checks
2. Severity-based merge thresholds

3. Test and coverage gates
4. Alert triage and false-positive handling
5. Prompt, Skill and agent-configuration review
6. Deployment approval and rollback readiness

LangGraph currently provides graph state, nodes, conditional routing, persistence, durable execution and human interrupts; OWASP's current agentic guidance highlights risks including prompt injection, excessive agency and compromised agent components. GitHub supports pull-request code scanning, dependency review and secret scanning as release controls.

Disclaimer

This outline is provided solely as an indicative training framework and should not be interpreted as a fixed, exhaustive or contractually binding syllabus. The trainer reserves the right to amend, reorder, substitute, reduce or omit any content where reasonably necessary to address participant readiness, platform availability, security requirements, technical developments or operational constraints, without prior notice.

© 2026 Faysal Aziz · Kuala Lumpur