Faysal Aziz

OpenClaw - From Personal AI Agent to Production-Grade Agentic Systems

Build the gateway. Give it skills. Connect the world. Secure the autonomy. - 3 days

FA-0343 · 3 days

Agentic & Generative AI · Automation, RPA & Power Platform

Introduction

OpenClaw is more than another interface to a large language model. It is an open-source, self-hosted gateway for building AI agents that can operate through real communication channels, use tools, maintain sessions and memory, execute workflows, interact with connected devices, and coordinate specialized agents. Its current architecture brings together a central Gateway, channel plugins, agent sessions, tools, skills, nodes, automation, and multi-agent routing.

This three-day course is deliberately designed to move fast. Participants begin by understanding what OpenClaw actually is and getting a working environment running. From there, the course moves into skills, tools, models, memory, messaging channels and automation before descending into the deeper engineering layer: Gateway architecture, WebSocket control-plane concepts, nodes, multi-agent design, permissions, isolation and security.

The objective is not to turn three days into an academic survey of every configuration option. It is to build the mental model and technical foundation required to create useful OpenClaw systems in the real world. Particular attention is given to security because an agent capable of invoking tools, accessing credentials and executing commands creates a fundamentally different risk profile from an ordinary chatbot. OpenClaw itself recommends explicit trust boundaries, tool restrictions, security auditing and isolation where users do not share the same trust level.

The instructor brings **over 30 years of industry experience**, and the course will therefore emphasize **real industry-demanded architecture, operational practices and deployment decisions rather than academic-style treatment**. The final progression is from simply talking to an agent to understanding how to engineer an agentic platform that can safely act.

Learning outcomes

By the end of this course, participants will be able to:

- Explain the OpenClaw architecture and agent execution model.
- Install, configure and operate an OpenClaw Gateway.
- Connect AI model providers and communication channels.
- Work with agents, sessions, memory, tools and workspaces.
- Create and manage custom OpenClaw skills.
- Design tool-enabled agent workflows and automation.
- Understand plugins, hooks and capability extension.
- Configure nodes and distributed agent capabilities.
- Design multi-agent architectures and routing strategies.
- Understand the Gateway WebSocket control plane.
- Apply permissions, isolation, secret management and security controls.
- Use OpenClaw security auditing and operational diagnostics.
- Design a practical production-oriented OpenClaw deployment.

Prerequisites

- Basic Linux, macOS or Windows administration skills
- Basic command-line experience
- Familiarity with APIs and API keys
- Basic understanding of AI/LLM concepts
- Basic JSON/YAML configuration knowledge
- Node.js/npm familiarity is helpful but not mandatory
- Programming experience is useful for the advanced sections

Training outline

1. OpenClaw Foundations and Architecture

Understanding OpenClaw

1. Agentic AI versus conversational AI
2. Self-hosted agent architecture
3. OpenClaw use cases

4. OpenClaw ecosystem

Core Architecture

1. Gateway
2. Agents
3. Sessions and context
4. Workspaces
5. Channels
6. Tools and skills
7. Plugins
8. Nodes

Installation and Initial Configuration

1. Platform requirements
2. Node.js and npm environment
3. OpenClaw installation
4. Onboarding workflow
5. Gateway service
6. Control UI
7. CLI fundamentals
8. Configuration structure

2. Building a Working AI Agent

Model Configuration

1. Model providers
2. Authentication
3. Model selection
4. Model fallback strategies
5. Local versus hosted models

Agent Fundamentals

1. Agent configuration

2. Agent identity
3. Workspace organization
4. Sessions
5. Context
6. Memory

Communication Channels

1. Channel architecture
2. Channel plugins
3. Telegram, WhatsApp, Slack and Discord concepts
4. Pairing and authentication
5. Message routing
6. Access policies

3. Tools, Skills and Agent Capabilities

OpenClaw Tool Architecture

1. Built-in tools
2. Tool invocation
3. Tool policies
4. Allow and deny controls
5. Host execution

Skills Architecture

1. SKILL.md structure
2. Skill discovery and loading
3. Skill precedence
4. Skill invocation
5. Environment requirements
6. Skill verification

Custom Skill Development

1. Skill directory structure

2. YAML frontmatter
3. Agent instructions
4. Tool integration
5. Skill testing
6. Skill distribution

ClawHub and Community Skills

1. Skill installation
2. Skill updates
3. Verification
4. Trust and provenance
5. Third-party skill risks

4. Automation and Autonomous Workflows

Agent Automation

1. Scheduled operations
2. Cron capabilities
3. Webhooks
4. Event-driven workflows
5. Background agent tasks

Extending OpenClaw

1. Plugins
2. Hooks
3. Custom tools
4. External APIs
5. Capability composition

5. Advanced OpenClaw Engineering

Gateway Internals

1. Gateway runtime architecture

2. Control plane
3. WebSocket protocol
4. Operator and node roles
5. RPC concepts
6. Connection lifecycle

Nodes and Distributed Capabilities

1. Node architecture
2. Device pairing
3. Headless nodes
4. macOS and mobile nodes
5. Node-hosted skills
6. Remote command surfaces

Multi-Agent Architecture

1. Agent isolation
2. Multi-agent routing
3. Specialized agents
4. Workspace separation
5. Capability separation
6. Agent-to-agent workflows

6. Security, Hardening and Production Operations

OpenClaw Security Model

1. Operator trust boundaries
2. Agent permissions
3. Tool authority
4. Prompt injection exposure
5. Credential risks
6. Remote execution risks

Gateway Hardening

1. Authentication
2. Network exposure
3. File permissions
4. Secret management
5. Tool allowlists
6. Plugin trust
7. Skill trust

Security Validation

1. openclaw security audit
2. Deep security auditing
3. Security remediation
4. Configuration validation
5. Operational diagnostics

Production Architecture

1. Gateway placement
2. Remote access
3. Tenant isolation
4. Agent privilege separation
5. Logging and observability
6. Backup and recovery
7. Upgrade strategy

7. Engineering the Complete OpenClaw Platform

End-to-End Architecture

1. Channels to Gateway
2. Gateway to agents
3. Agents to tools
4. Skills and plugins
5. Nodes and external systems

6. Models and providers

Enterprise Design Considerations

1. Least-privilege agents
2. Segmented gateways
3. Controlled autonomy
4. Human approval boundaries
5. Governance and operational ownership
6. Scaling agentic systems

The outline reflects current OpenClaw concepts including its Gateway-centric architecture, skills system, connected nodes and current security model. OpenClaw's documentation describes the Gateway WebSocket protocol as the common control plane and node transport, while its current skills architecture supports workspace, project, personal, managed and bundled skill sources.

Disclaimer

This training outline is intended as a structured guideline for a three-day instructor-led programme. The trainer reserves the right to amend, reorder, expand, reduce, substitute or omit topics where reasonably necessary to accommodate participant experience, technical developments, available training time, platform changes or instructional requirements, without prior notice.

© 2026 Faysal Aziz · Kuala Lumpur