Faysal Aziz

Designing and Implementing Azure AI Solutions

A 3-day intensive built around the real AI engineering work Microsoft expects today.

FA-0168 · 3-day

Agentic & Generative AI · DevOps, Cloud & Infrastructure · Data & Analytics

---

Introduction

Microsoft's AI-102 sits at an interesting point in Azure's certification path. It still maps to the working Azure AI engineer role and, as of the current Microsoft Learn study guide updated December 23, 2025, it covers six domains: planning and managing Azure AI solutions, generative AI, agentic solutions, computer vision, natural language processing, and knowledge mining with information extraction. At the same time, Microsoft has announced that the **Azure AI Engineer Associate certification (Exam AI-102)** is scheduled to retire on **June 30, 2026**, with **Azure AI App and Agent Developer Associate (Exam AI-103)** named as its replacement. Microsoft's Skills Hub announcement says AI-103 is aligned more directly with generative and agentic architectures, with training expected in March 2026, beta availability in April 2026, and general availability expected in June 2026.

This course therefore teaches AI-102 thoroughly, but it does so in a way that prepares learners for the direction Microsoft is clearly taking: Foundry-centric development, production-grade generative AI, and agent-based application design. It is delivered by an instructor with **over 30 years of industry experience**, using real, industry-demanded content rather than an academic treatment of AI.

Learning outcomes

By the end of this 3-day course, participants will be able to:

- Design and scope Azure AI solutions against the current AI-102 skills measured.
- Select appropriate Microsoft Foundry and Azure AI services for generative AI, vision, speech, language, search, and information extraction workloads.
- Provision, secure, monitor, and manage Azure AI resources, endpoints, authentication, keys, costs, and deployment options.

- Build generative AI applications using Microsoft Foundry, Azure OpenAI models, prompt flow, prompt templates, RAG patterns, and evaluation workflows.
- Create and operationalize agentic solutions using Microsoft Foundry Agent Service and multi-agent orchestration concepts.
- Implement computer vision, OCR, video analysis, language, translation, speech, and custom language capabilities in Azure AI solutions.
- Build knowledge mining and information extraction pipelines using Azure AI Search, Document Intelligence, and Content Understanding.
- Apply responsible AI controls including content moderation, content safety, filters, blocklists, prompt shields, and governance practices.
- Prepare effectively for AI-102 while understanding how its content is evolving toward AI-103.

Prerequisites

- Working knowledge of both **Python** and **C#**.
- Basic familiarity with **REST APIs** and SDK-based development.
- Foundational understanding of Azure concepts such as resources, authentication, and deployment. This is inferred from Microsoft's audience profile and course positioning for software engineers building and deploying AI solutions on Azure.
- Prior exposure to AI-900 or basic AI workloads such as NLP, vision, or search, since AI-102 assumes practical implementation rather than introductory theory. This is an instructional inference based on the exam audience and topic scope.

Training outline

1. Course orientation and certification context

1. The Azure AI engineer role
2. What AI-102 validates today
3. How AI-102 differs from fundamentals-level study
4. Microsoft's shift from Azure AI Engineer Associate to Azure AI App and Agent Developer Associate
5. What the AI-102 retirement on June 30, 2026 means for learners
6. How this 3-day course covers current objectives while anticipating AI-103 direction

2. Planning and managing an Azure AI solution

Azure AI portfolio overview

1. Microsoft Foundry services

2. Azure AI Vision
3. Azure AI Speech
4. Azure AI Language
5. Azure AI Search
6. Azure OpenAI
7. Azure AI Document Intelligence
8. Azure Content Understanding

Service selection strategy

1. Choosing services for generative AI solutions
2. Choosing services for computer vision solutions
3. Choosing services for natural language processing solutions
4. Choosing services for speech solutions
5. Choosing services for information extraction solutions
6. Choosing services for knowledge mining solutions

Planning, creating, and deploying Foundry services

1. Responsible AI planning
2. Creating Azure AI resources
3. Model selection strategy
4. Deployment options
5. SDK and API selection
6. Endpoint planning
7. CI/CD integration
8. Container deployment planning

Managing, monitoring, and securing services

1. Resource monitoring
2. Cost management
3. Key protection
4. Authentication models
5. Operational governance
6. Environment management for production workloads

Responsible AI implementation

1. Content moderation
2. Responsible AI insights
3. Content safety
4. Content filters and blocklists
5. Prompt shields

6. Harm detection
7. Governance framework design

3. Implementing generative AI solutions

Generative AI solution architecture in Microsoft Foundry

1. Planning a generative AI workload
2. Hub and project design
3. Resource dependencies
4. Model deployment strategy

Building generative applications

1. Prompt flow implementation
2. Prompt template design
3. Grounding models with enterprise data
4. Retrieval-augmented generation patterns
5. Evaluation of models and flows
6. SDK integration into applications

Azure OpenAI in Foundry Models

1. Resource provisioning
2. Model deployment and selection
3. Text and code generation
4. Image generation with DALL-E
5. Multimodal model usage
6. Application integration patterns

Optimization and operationalization

1. Parameter tuning for response behavior
2. Diagnostics and monitoring
3. Resource optimization and scalability
4. Foundational model update considerations
5. Tracing and feedback collection
6. Model reflection
7. Local and edge container deployment
8. Multi-model orchestration
9. Prompt engineering
10. Fine-tuning strategy

4. Implementing an agentic solution

Agent concepts and use cases

1. The role of an agent in modern AI applications
2. When to use agentic patterns instead of standard prompt-response apps

Agent platform setup

1. Resource configuration for agent development
2. Microsoft Foundry Agent Service fundamentals

Building custom agents

1. Agent creation workflows
2. Tool and capability configuration
3. State and task handling
4. Testing and iterative optimization

Advanced agent architecture

1. Microsoft Agent Framework concepts
2. Complex workflow design
3. Multi-agent orchestration
4. Multi-user interaction patterns
5. Autonomous capability considerations
6. Deployment and production readiness

5. Implementing computer vision solutions

Image analysis

1. Selecting visual features
2. Object detection
3. Tag generation
4. Request construction
5. Response interpretation
6. OCR from images
7. Handwritten text extraction

Custom vision models

1. Classification versus object detection
2. Image labeling
3. Model training workflows
4. Model evaluation metrics

5. Model publishing
6. Model consumption
7. Code-first implementation patterns

Video analysis

1. Azure AI Video Indexer
2. Insight extraction from recorded video
3. Insight extraction from live streams
4. Spatial analysis for people presence and movement

6. Implementing natural language processing solutions

Text analytics and translation

1. Key phrase extraction
2. Entity recognition
3. Sentiment analysis
4. Language detection
5. PII detection
6. Document and text translation

Speech solutions

1. Generative AI speaking capabilities
2. Speech-to-text
3. Text-to-speech
4. SSML enhancement
5. Custom speech solutions
6. Intent recognition
7. Keyword recognition
8. Speech translation workflows

Custom language models

1. Intents
2. Entities
3. Utterance design
4. Model training
5. Model evaluation
6. Deployment and testing
7. Optimization, backup, and recovery
8. Client application integration

Question answering solutions

1. Custom question answering projects
2. Source import and Q&A pair design
3. Knowledge base training and publishing
4. Multi-turn conversation design
5. Alternate phrasing and chit-chat
6. Export workflows
7. Multilingual question answering

Custom translation

1. Training custom models
2. Improving custom models
3. Publishing custom translation models

7. Implementing knowledge mining and information extraction solutions

Azure AI Search foundations

1. Resource provisioning
2. Index design
3. Skillset definition
4. Data source creation
5. Indexer creation and execution
6. Query syntax
7. Sorting, filtering, and wildcards
8. Knowledge Store projections
9. Semantic search
10. Vector store solutions

Azure Document Intelligence in Foundry Tools

1. Resource provisioning
2. Prebuilt model usage
3. Custom model implementation
4. Training, testing, and publishing
5. Composed models

Azure Content Understanding in Foundry Tools

1. OCR pipelines for images and documents
2. Summarization

3. Classification
4. Attribute detection
5. Entity extraction
6. Table extraction
7. Image extraction
8. Multimodal ingestion across documents, images, video, and audio

8. Cross-cutting engineering practices

1. SDK versus REST implementation choices
2. Architecture trade-offs across services
3. Security and authentication patterns
4. Cost-aware solution design
5. Monitoring and diagnostics
6. Production deployment considerations
7. Responsible AI by design
8. Exam-focused mapping of scenarios to services

9. Exam preparation and transition readiness

1. Interpreting the AI-102 weighting by skill area
2. High-weight domains to prioritize
3. Hands-on practice strategy
4. Reading Microsoft Learn study resources efficiently
5. Positioning AI-102 knowledge for the AI-103 transition
6. Recognizing which AI-102 topics are already pointing toward Foundry, generative AI, and agentic development

© 2026 Faysal Aziz · Kuala Lumpur