

Updated on 02/18/2026

Register

AWS Generative AI Developer Professional Certification Training

4 days (28 hours)

Overview

AWS Certified Generative AI Developer – Professional is an advanced certification dedicated to developing generative artificial intelligence applications on AWS.

It covers the design of solutions based on Foundation Models, application integration, and best practices for security, governance, and optimization.

Our training will enable you to master the implementation of generative AI on AWS: using Amazon Bedrock, prompt engineering, RAG architectures, serverless integration, observability, and industrialization.

You will learn how to design, secure, deploy, and optimize production-ready GenAI applications, controlling latency and costs while applying cloud architecture best practices.

At the end of the training, you will be able to develop a complete GenAI solution on AWS, implement RAG pipelines, put safeguards in place against risks, and effectively prepare for the AIP-C01 exam through practical case studies and a mock exam.

Like all our training courses, this one is based on the latest AWS [AIP-C01 certification](#) framework and focuses on a decidedly practical and operational approach.

Objectives

- Understand the AWS ecosystem for generative AI and Foundation Models.
- Use Amazon Bedrock to integrate models and drive inference.
- Implement advanced prompt engineering strategies.
- Design a RAG architecture.

- Secure, monitor, and industrialize a GenAI application.
- Prepare for the AIP-C01 certification.

Target audience

- Experienced developers
- Cloud/DevOps engineers on AWS
- Data/ML/AI engineers wishing to industrialize GenAI solutions
- Technical architects involved in GenAI projects

Prerequisites

- Good knowledge of AWS fundamentals
- Experience in application development
- Basic knowledge of AI/machine learning recommended

Our training program AWS Generative AI Developer – Professional Certification (AIP-C01)

[Day 1 - Morning]

AWS Generative AI ecosystem and certification overview

- Positioning generative AI in the AWS ecosystem: services, responsibilities, and scope
- Understanding the role of Foundation Models and the differences with custom-trained models
- Mapping use cases: chatbots, copilots, RAG, agents, automation
- "Exam" reading: expected skills and types of questions AIP-C01
- Best practices for getting started: accounts, regions, quotas, environments
- Hands-on workshop: Verify access to GenAI services, configure the AWS SDK, and validate the environment.

[Day 1 - Afternoon]

Amazon Bedrock: models, inference, and parameters

- Discover Amazon Bedrock: model access logic and application integration
- Compare models (quality, latency, cost, context, formats) according to your needs
- Master inference parameters: temperature, top-p, max tokens, stop sequences
- Manage limits: quotas, throttling, timeouts, context management
- Best practices for structuring prompts and messages (system/user/assistant)
- Hands-on workshop: Call Bedrock via SDK, compare two models, and measure cost/latency.

Security and governance: IAM, isolation, and compliance

- Applying the Shared Responsibility Model to generative AI on AWS
- Implement access control: IAM, policies, roles, separation of duties
- Securing data: encryption, storage, KMS, secret management
- Traceability: logs, audit, governance principles, and compliance requirements
- Define an isolation strategy (environments, accounts, VPC endpoints if applicable)
- Hands-on workshop: Create a minimalist IAM role for Bedrock and validate "least privilege" access.

[Day 2 - Morning]

Advanced prompt engineering and response quality

- Structure robust prompts: objectives, constraints, style, output format
- Techniques: zero-shot, few-shot, controlled reasoning chains, decomposition
- Context management: token limits, summaries, memory, dynamic context
- Error reduction: hallucinations, ambiguities, format non-compliance, drifts
- Evaluation: quality criteria, reproducibility, prompt set testing
- Hands-on workshop: Improving a "bad" prompt into a "prod" prompt with quality metrics.

[Day 2 - Afternoon]

RAG: embeddings, indexing, and semantic search

- Understanding the RAG pattern: why, when, limitations, and alternatives
- Generating embeddings and choosing a chunking strategy (size, overlap)
- Storing and querying a vector index (e.g., OpenSearch, other AWS options depending on context)
- Building the pipeline: ingestion, cleaning, enrichment, indexing, and querying
- Improving relevance: reranking, filters, metadata, citations, and grounding
- Hands-on workshop: Building a mini-RAG.

GenAI applications: architecture patterns and streaming

- Application patterns: chat, assistant, extraction, structured generation (JSON)
- Session management: conversational context, short/long memory, idempotence
- Response streaming: UX, perceived latency, interruption management
- Error management: retries, backoff, timeouts, circuit breaker, fallback model
- Best practices for front/back integration (API, formats, validation)
- Hands-on workshop: Exposing a chat API with streaming and output format validation.

[Day 3 - Morning]

Serverless GenAI: Lambda, API Gateway, and orchestration

- Designing a serverless GenAI architecture: components and responsibilities
- Integrating Lambda and API Gateway: authentication, throttling, quotas

- Orchestration: workflows, events, integration with AWS services
- Optimization strategies: cold starts, execution time, caching
- Reliability: DLQ, idempotence, peak load management
- Hands-on workshop: Deploying a prod-ready serverless API calling Bedrock.

[Day 3 - Afternoon]

Cost and performance optimization: multi-model strategy

- Understanding cost factors: tokens, context, models, call frequency
- Reducing costs: prompt compression, summaries, caching, context limitation
- Optimizing latency: model choice, streaming, parallelization, batching
- Multi-model approach: routing, fallback, "cheap-first," escalation based on complexity
- Measurement: cost/latency dashboards, budgets, alerts
- Hands-on workshop: Benchmark 2 models.

CI/CD and Infrastructure as Code for GenAI applications

- Automating deployment: CI/CD pipelines, artifacts, environments
- Infrastructure as Code: Terraform (or equivalent), modularization, and secrets
- Versioning prompts and parameters: dev promotion? Staging? Prod
- Tests: unit, contract, prompt non-regression, load testing
- Deployment strategies: blue/green, canary, rollback
- Hands-on workshop: CI/CD pipeline deploying a GenAI API + automated testing.

[Day 4 - Morning]

GenAI application security: risks and protections

- GenAI threats: prompt injection, data exfiltration, jailbreak, overconfidence
- Protection techniques: sanitation, constraints, context/instruction separation
- Validation and control: schemas, safeguards, filtering, and content policy
- Secret and sensitive data management: masking, redaction, conditional access
- Secure by design approach: reviews, testing, security runbooks
- Hands-on workshop: Simulate prompt injection and implement countermeasures.

[Day 4 - Afternoon]

Observability and operation: logs, metrics, alerting

- Defining indicators: latency, errors, cost, quality, fallback rate
- Centralize logs and traces: query correlation, auditing, debugging
- Alerting: thresholds, anomalies, budgets, cost alerts
- Run in production: SLO/SLI, incident management, post-mortem
- Continuous improvement: feedback loops, A/B prompts, operational tuning

- Hands-on workshop: Building an observability dashboard.

Preparation for the AIP-C01 exam

- Review of assessed areas and consolidation of key points AIP-C01
- Problem-solving method: analysis, elimination, pitfalls, and distractors
- Case studies: architecture choices, security, costs, RAG, serverless
- Checklists: architecture, security, observability, optimization, and best practices
- Revision plan: priorities, resources, pace, and practice
- Practical workshop: Mock exam + correction.

Companies involved

This training is aimed at both individuals and companies, large or small, wishing to train their teams in new advanced IT technology or to acquire specific professional knowledge or modern methods.

Placement at the start of training

The placement test at the start of the training course complies with Qualiopi quality criteria. Once they have finalized their registration, learners receive a self-assessment questionnaire that allows us to assess their estimated level of proficiency in different types of technologies, as well as their expectations and personal objectives for the upcoming training course, within the limits imposed by the selected format. This questionnaire also allows us to anticipate certain connection or internal security issues within the company (intra-company or virtual classroom) that could be problematic for the monitoring and smooth running of the training session.

Teaching methods

Practical training: 60% practical, 40% theory. Training materials distributed in digital format to all participants.

Organization

The course alternates between theoretical input from the trainer, supported by examples and reflection sessions, and group work.

Validation

At the end of the session, a multiple-choice questionnaire is used to verify that the skills have been correctly acquired.

Certification

A certificate will be issued to each trainee who has completed the entire training course.