+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Mis à jour le 06/02/2026

S'inscrire

Formation Certification AWS Generative AI Developer Professional

4 jours (28 heures)

Présentation

AWS Certified Generative AI Developer – Professional est une certification avancée dédiée au développement d'applications d'intelligence artificielle générative sur AWS.

Elle couvre la conception de solutions basées sur des Foundation Models, l'intégration applicative et les bonnes pratiques de sécurité, gouvernance et optimisation.

Notre formation vous permettra de maîtriser la mise en œuvre de l'IA générative sur AWS : utilisation d'Amazon Bedrock, prompt engineering, architectures RAG, intégration serverless, observabilité et industrialisation.

Vous apprendrez à concevoir, sécuriser, déployer et optimiser des applications GenAI prêtes pour la production, en contrôlant la latence et les coûts, tout en appliquant les meilleures pratiques d'architecture cloud.

À l'issue de la formation, vous serez en mesure de développer une solution GenAI complète sur AWS, d'implémenter des pipelines RAG, de mettre en place des garde-fous contre les risques et de préparer efficacement l'examen AIP-C01 grâce à des cas pratiques et un examen blanc.

Comme toutes nos formations, celle-ci s'appuie sur la dernier référentiel de la certification AIP-C01 d'AWS et privilégie une approche résolument pratique et opérationnelle.

Objectifs

- Comprendre l'écosystème AWS pour l'IA générative et les Foundation Models.
- Utiliser Amazon Bedrock pour intégrer des modèles et piloter l'inférence.
- Mettre en œuvre des stratégies de prompt engineering avancées
- Concevoir une architecture RAG.

- Sécuriser, superviser et industrialiser une application GenAI.
- Se préparer à la certification AIP-C01.

Public visé

- Développeurs confirmés
- Ingénieurs Cloud / DevOps sur AWS
- Ingénieurs Data / ML / IA souhaitant industrialiser des solutions GenAI
- Architectes techniques impliqués dans des projets GenAI

Pré-requis

- Bonne connaissance des fondamentaux AWS
- Expérience en développement applicatif
- Notions de base en IA / Machine Learning recommandées

Programme de formation Certification AWS Generative AI Developer – Professional (AIP-C01)

[Jour 1 - Matin]

Écosystème AWS Generative AI et cadrage de la certification

- Positionner l'IA générative dans l'écosystème AWS : services, responsabilités et périmètre
- Comprendre le rôle des Foundation Models et les différences avec modèles entraînés sur mesure
- Cartographie des cas d'usage : chatbots, copilots, RAG, agents, automatisation
- Lecture "examen" : compétences attendues et types de questions AIP-C01
- Bonnes pratiques de mise en route : comptes, régions, quotas, environnements
- Atelier pratique : Vérifier l'accès aux services GenAI, configurer le SDK AWS et valider l'environnement.

[Jour 1 - Après-midi]

Amazon Bedrock : modèles, inférence et paramètres

- Découvrir Amazon Bedrock : logique d'accès aux modèles et intégration applicative
- Comparer les modèles (qualité, latence, coût, contexte, formats) selon les besoins
- Maîtriser les paramètres d'inférence : temperature, top-p, max tokens, stop sequences
- Gérer les limites : quotas, throttling, timeouts, gestion du contexte
- Bonnes pratiques de structuration des prompts et messages (system/user/assistant)
- Atelier pratique : Appeler Bedrock via SDK, comparer 2 modèles et mesurer coût/latence.

Sécurité et gouvernance : IAM, isolation et conformité

- Appliquer le Shared Responsibility Model à l’IA générative sur AWS
- Mettre en place le contrôle d’accès : IAM, policies, rôles, séparation des responsabilités
- Sécuriser les données : chiffrement, stockage, KMS, gestion des secrets
- Traçabilité : logs, audit, principes de gouvernance et exigences de conformité
- Définir une stratégie d’isolation (environnements, comptes, VPC endpoints si applicable)
- Atelier pratique : Créer un rôle IAM minimaliste pour Bedrock et valider l’accès “least privilege”.

[Jour 2 - Matin]

Prompt engineering avancé et qualité des réponses

- Structurer des prompts robustes : objectifs, contraintes, style, format de sortie
- Techniques : zero-shot, few-shot, chaînes de raisonnement contrôlées, décomposition
- Gestion du contexte : limites de tokens, résumés, mémoire, contexte dynamique
- Réduction des erreurs : hallucinations, ambiguïtés, non-respect du format, dérives
- Évaluation : critères de qualité, reproductibilité, tests sur jeux de prompts
- Atelier pratique : Améliorer un prompt “mauvais” en prompt “prod” avec métriques de qualité.

[Jour 2 - Après-midi]

RAG : embeddings, indexation et recherche sémantique

- Comprendre le pattern RAG : pourquoi, quand, limites et alternatives
- Générer des embeddings et choisir une stratégie de chunking (taille, overlap)
- Stocker et interroger un index vectoriel (ex : OpenSearch, autres options AWS selon contexte)
- Construire le pipeline : ingestion, nettoyage, enrichissement, indexation et requêtage
- Améliorer la pertinence : reranking, filtres, métadonnées, citations et grounding
- Atelier pratique : Construire un mini-RAG.

Applications GenAI : patterns d’architecture et streaming

- Patterns applicatifs : chat, assistant, extraction, génération structurée (JSON)
- Gestion des sessions : contexte conversationnel, mémoire courte/longue, idempotence
- Streaming des réponses : UX, latence perçue, gestion des interruptions
- Gestion des erreurs : retries, backoff, timeouts, circuit-breaker, fallback modèle
- Bonnes pratiques d’intégration front/back (API, formats, validation)
- Atelier pratique : Exposer une API chat avec streaming et validation du format de sortie.

[Jour 3 - Matin]

Serverless GenAI : Lambda, API Gateway et orchestration

- Concevoir une architecture serverless GenAI : composants et responsabilités
- Intégrer Lambda et API Gateway : authentification, throttling, quotas

- Orchestration : workflows, événements, intégration avec services AWS
- Stratégies d’optimisation : cold starts, temps d’exécution, mise en cache
- Fiabilité : DLQ, idempotence, gestion des pics de charge
- Atelier pratique : Déployer une API serverless “prod-ready” appelant Bedrock.

[Jour 3 - Après-midi]

Optimisation coûts et performances : stratégie multi-modèles

- Comprendre les facteurs de coût : tokens, contexte, modèles, fréquence d’appels
- Réduire le coût : compression du prompt, résumés, caching, limitation de contexte
- Optimiser la latence : choix modèle, streaming, parallélisation, batching
- Approche multi-modèles : routage, fallback, “cheap-first”, escalade selon complexité
- Mesurer : tableaux de bord coûts/latence, budgets, alertes
- Atelier pratique : Benchmark 2 modèles.

CI/CD et Infrastructure as Code pour applications GenAI

- Automatiser le déploiement : pipelines CI/CD, artefacts, environnements
- Infrastructure as Code : Terraform (ou équivalent), modularisation et secrets
- Versionner prompts et paramètres : promotion dev ? staging ? prod
- Tests : unitaires, contract, non-régression de prompts, tests de charge
- Stratégies de déploiement : blue/green, canary, rollback
- Atelier pratique : Pipeline CI/CD déployant une API GenAI + tests automatisés.

[Jour 4 - Matin]

Sécurité applicative GenAI : risques et protections

- Menaces GenAI : prompt injection, data exfiltration, jailbreak, sur-confiance
- Techniques de protection : sanitation, contraintes, séparation contexte/instructions
- Validation et contrôle : schémas, garde-fous, filtrage et policy de contenu
- Gestion des secrets et des données sensibles : masquage, redaction, accès conditionnel
- Approche “secure by design” : revues, tests, runbooks sécurité
- Atelier pratique : Simuler une prompt injection et mettre en place des contre-mesures.

[Jour 4 - Après-midi]

Observabilité et exploitation : logs, métriques, alerting

- Définir les indicateurs : latence, erreurs, coût, qualité, taux de fallback
- Centraliser logs et traces : corrélation requêtes, audit, debugging
- Alerting : seuils, anomalies, budgets, alertes coûts
- Run en production : SLO/SLI, gestion d’incidents, post-mortem
- Amélioration continue : feedback loops, A/B prompts, tuning opérationnel

- Atelier pratique : Construire un dashboard d’observabilité.

Préparation à l'examen AIP-C01

- Revue des domaines évalués et consolidation des points clés AIP-C01
- Méthode de résolution : analyse, élimination, pièges et “distractors”
- Études de cas : choix d’architecture, sécurité, coûts, RAG, serverless
- Checklists : architecture, sécurité, observabilité, optimisation et bonnes pratiques
- Plan de révision : priorités, ressources, rythme et entraînement
- Atelier pratique : Passage de l'examen blanc + correction.

Sociétés concernées

Cette formation s’adresse à la fois aux particuliers ainsi qu’aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

Positionnement à l'entrée en formation

Le positionnement à l'entrée en formation respecte les critères qualité Qualiopi. Dès son inscription définitive, l'apprenant reçoit un questionnaire d’auto-évaluation nous permettant d’apprécier son niveau estimé sur différents types de technologies, ses attentes et objectifs personnels quant à la formation à venir, dans les limites imposées par le format sélectionné. Ce questionnaire nous permet également d’anticiper certaines difficultés de connexion ou de sécurité interne en entreprise (intraentreprise ou classe virtuelle) qui pourraient être problématiques pour le suivi et le bon déroulement de la session de formation.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l’acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.