

Administration de solutions d'Intelligence Artificielle (IA) en production

3 jours (21 heures)

Expertise

Classe à distance

à partir de

2 580 € HT par personne (TVA en vigueur)

Contenu mis à jour le 25/09/2026. Document téléchargé le 07/10/2026.

Objectifs

À l'issue de cette formation, vous serez capable de :

- Installer, configurer et maintenir des serveurs d'inférence ML et LLM
- Superviser les métriques spécifiques à l'Intelligence Artificielle (IA) (dérive, performance modèle, coûts tokens)
- Gérer le cycle de vie des modèles et des prompts en production
- Administrer les bases vectorielles et les pipelines RAG
- Sécuriser et auditer les déploiements IA
- Diagnostiquer et résoudre les incidents propres aux solutions IA.

Les compétences visées par cette formation font l'objet d'une évaluation.

À qui s'adresse cette formation ?

Administrateurs systèmes, ingénieurs DevOps / MLOps, ingénieurs d'exploitation en charge de solutions d'Intelligence Artificielle (IA) en production (ML classique et LLM).

Prérequis

Avoir une expérience en administration Linux et des notions de conteneurisation (Docker).

Afin de valider vos prérequis avant la formation, un questionnaire est disponible ; n'hésitez pas à nous contacter.

Programme

1 Pendant votre formation

Administration de solutions d'Intelligence Artificielle (IA) en production

Paysage opérationnel d'une solution d'Intelligence Artificielle (IA)

- Ce que l'admin IA gère que les autres admins ne gèrent pas : modèles, prompts, embeddings, GPU, dérive
- Anatomie d'une solution IA en production : modèle(s), serveur d'inférence, gateway, base vectorielle, guardrails, observabilité – cartographie des composants à administrer
- Deux mondes à opérer : pipeline ML classique (entraînement > registre > serving) vs stack LLM (modèle pré-entraîné > adaptateurs LoRA > serving > RAG > agents)
- Environnements types : Cloud managé (Bedrock, Vertex AI), auto-hébergé (vLLM, TGI), hybride, edge (Jetson / Ollama)
- Les interlocuteurs de l'admin IA : data scientists, développeurs, architectes – qui fait quoi, qui décide quoi

Exemple de travaux pratiques (à titre indicatif)

- *Cartographie d'une solution IA existante : inventaire des composants, identification des responsabilités, repérage des points de fragilité opérationnelle*

Administration des serveurs d'inférence

- Serveurs d'inférence ML classique : TensorFlow Serving, Triton, ONNX Runtime – installation, configuration, gestion multi-modèles
- Serveurs d'inférence LLM : vLLM (PagedAttention, continuous batching), TGI (Hugging Face), Ollama (poste / edge) – installation, paramétrage, différences opérationnelles
- Configuration des paramètres d'inférence : température, top-p, max tokens, contexte – comprendre ce qu'on configure et pourquoi
- Gestion GPU : allocation VRAM, KV-cache sizing, tensor parallelism multi-GPU – comprendre nvidia-smi et ses métriques
- Serving multi-LoRA : charger un modèle de base et plusieurs adaptateurs, routage par requête – configuration vLLM
- Mise à jour d'un modèle sans interruption : téléchargement des poids, validation, bascule, rollback
- Gestion du stockage des modèles : poids volumineux (dizaines de Go), cache Hugging Face, nettoyage, distribution sur plusieurs nœuds

Exemple de travaux pratiques (à titre indicatif)

- Installation et configuration de vLLM et Ollama : chargement d'un modèle 7B, configuration multi-LoRA sur vLLM, benchmark de performance (tokens / s, TTFT, VRAM), procédure de mise à jour de modèle

Administration des pipelines RAG et bases vectorielles

- Composants d'un RAG en production : pipeline d'ingestion, service d'embedding, base vectorielle, reranker – ce que l'admin doit surveiller
- Administration des bases vectorielles : Chroma, pgvector, Weaviate – installation, configuration, sauvegarde et restauration des index
- Ingestion de documents : monitoring du pipeline, gestion des erreurs, réindexation partielle ou complète
- Dimensionnement : nombre de vecteurs, dimensions, latence de recherche, mémoire – quand ça ne tient plus
- Maintenance des embeddings : changement de modèle d'embedding > réindexation complète, procédure et planification
- Qualité du RAG en production : monitoring du taux de retrieval vide, des scores de similarité, détection de la dégradation

Exemple de travaux pratiques (à titre indicatif)

- Administration d'un pipeline RAG : installation pgvector, ingestion d'un corpus, sauvegarde / restauration des index, simulation de réindexation après changement de modèle d'embedding, monitoring de la qualité de recherche

Supervision et observabilité IA

- Ce qu'il faut surveiller et que Prometheus / Grafana ne couvrent pas nativement
- Métriques ML classique : précision, F1, latence d'inférence, dérive des données (data drift), dérive du modèle (concept drift) – détection automatique, seuils adaptatifs
- Métriques LLM : tokens / s, latence TTFT (time to first token), latence de génération complète, consommation VRAM, taille du KV-cache, file d'attente d'inférence, coût par requête
- Outils de traçabilité LLM : Langfuse, LangSmith, Phoenix – instrumentation des chaînes d'appels, traces de raisonnement, détection d'anomalies dans les réponses
- Métriques métier : taux de résolution (pour un assistant), taux de réponses vides, taux d'escalade, satisfaction utilisateur
- Construction de tableaux de bord : Grafana pour l'infra GPU et Langfuse pour les traces LLM – vue unifiée
- Alerting : quand alerter (dégradation de latence, dérive, saturation GPU, explosion des coûts), comment escalader

Exemple de travaux pratiques (à titre indicatif)

- Déploiement d'une stack d'observabilité IA : Prometheus / Grafana pour les métriques infra et GPU, instrumentation Langfuse sur une chaîne RAG, construction d'un tableau de bord unifié, configuration d'alertes sur dérive et saturation

Cycle de vie des modèles et des prompts

- Gestion des modèles en production : registre de modèles (MLflow, Hugging Face Hub privé), étiquetage des versions, traçabilité de la provenance
- Procédures de mise à jour de modèles ML : validation sur jeu de test, déploiement canari, critères de rollback automatique

- Migration de modèles LLM : passage d'une version à une autre (ex : Llama 3.1 > 3.2), tests de régression automatisés (comparaison des réponses sur un jeu de tests métier)
- Gestion des prompts en production : versionning des system prompts, stockage structuré, promotion dev > staging > prod, historique des modifications, revue avant déploiement
- A / B testing de prompts : deux versions en parallèle, collecte de métriques, décision de promotion
- Gestion des adaptateurs LoRA : registre, association modèle de base / adaptateur, procédure d'ajout et de retrait
- Documentation opérationnelle : ce qu'il faut documenter (versions en production, dépendances, paramètres d'inférence, prompts actifs)

Exemple de travaux pratiques (à titre indicatif)

- *Cycle de vie complet – mise à jour d'un modèle LLM avec tests de régression automatisés, versionning et promotion d'un prompt système, déploiement canari, rollback sur échec*

Sécurité et guardrails en production

- Menaces spécifiques : prompt injection (directe, indirecte), exfiltration de données via function calling, jailbreak, abus (génération de contenu interdit)
- Guardrails : installation et configuration de filtres d'entrée / sortie. Gestion des règles, mise à jour, traitement des faux positifs
- Rate limiting et quotas : par utilisateur, par équipe, par clé API – protéger l'infra et maîtriser les coûts
- Journalisation des interactions : que logger (prompt, réponse, outils appelés, latence, coût), combien de temps conserver, anonymisation si nécessaire
- Audit et conformité : traçabilité des décisions automatisées, RGPD dans un contexte IA (droit à l'oubli dans un vector store, suppression de données d'entraînement)
- Isolation des données : multi-tenancy dans un RAG (un index par client vs filtrage par métadonnées), isolation des adaptateurs LoRA

Exemple de travaux pratiques (à titre indicatif)

- *Sécurisation d'un déploiement LLM : mise en place de guardrails, configuration du rate limiting, journalisation complète, test de prompt injection et vérification du blocage, procédure de suppression de données dans un vector store*

Diagnostic et résolution d'incidents IA

- Les incidents qu'un admin IA rencontre et que les admins classiques ne connaissent pas
- Dégradation de la qualité des réponses : dérive des données, prompt inadapté, modèle obsolète, base vectorielle corrompue – arbre de diagnostic
- Problèmes de performance : saturation VRAM, KV-cache overflow, file d'attente qui explose, modèle trop gros pour le GPU – diagnostic et remédiation
- Erreurs de serving : OOM GPU, timeout d'inférence, erreurs de tokenization, modèle corrompu – procédures de récupération
- Incidents de sécurité : prompt injection réussie, fuite de données, comportement anormal d'un agent – détection, confinement, investigation
- Incidents RAG : retrieval vide, réponses hors sujet, index corrompu, embedding model mismatch – diagnostic et correction
- Runbooks IA : structure, contenu, exemples – rédaction de procédures réutilisables

Exemple de travaux pratiques (à titre indicatif)

- *Diagnostic sur incidents simulés : 4 scénarios d'incidents (dégradation qualité, OOM GPU, prompt injection, RAG défaillant), investigation avec les outils vus aux chapitres précédents, résolution, rédaction d'un runbook pour chaque cas*

Coûts, maintenance planifiée et projet final

- FinOps IA : suivi des coûts tokens, coûts GPU, coûts stockage vectoriel – tableaux de bord de coûts, alertes budgétaires, leviers d'optimisation (caching, modèle plus petit, quantification)
- Maintenance planifiée : calendrier de réindexation RAG, rotation des modèles, revue des prompts, nettoyage des caches et des logs, tests de restauration
- Automatisation : scripts de health check, rotation de modèle automatisée, pipeline de tests de régression en CI

Exemple de travaux pratiques (à titre indicatif)

- *Projet final : prise en charge complète d'une solution IA en production (serveur d'inférence LLM, pipeline RAG, guardrails) – installation, configuration du monitoring, sécurisation, simulation d'incidents et résolution, rédaction de la documentation opérationnelle et des runbooks*

Financement

Plusieurs dispositifs de financement sont accessibles via les OPCO (Opérateurs de Compétences), organismes agréés par le ministère du Travail dont le rôle est d'accompagner, collecter et gérer les contributions des entreprises au titre du financement de la formation professionnelle.

Pour plus d'information, une équipe de gestionnaires SKOLAE Formation spécialisée vous accompagne dans le choix de vos formations et la gestion administrative.

[Nous contacter pour en savoir plus →](#)

Modalités pédagogiques

Formation délivrée en présentiel ou distanciel* (blended-learning, e-learning, classe virtuelle, présentiel à distance).

Le formateur alterne entre méthode** démonstrative, interrogative et active (via des travaux pratiques et/ou des mises en situation).

Variables suivant les formations, les moyens pédagogiques mis en oeuvre sont :

- Ordinateurs Mac ou PC (sauf pour certains cours de l'offre Management), connexion internet fibre, tableau blanc ou paperboard, vidéoprojecteur ou écran tactile interactif (pour le distanciel)
- Environnements de formation installés sur les postes de travail ou en ligne
- Supports de cours et exercices

En cas de formation intra sur site externe à M2i, le client s'assure et s'engage également à avoir toutes les ressources matérielles pédagogiques nécessaires (équipements informatiques...) au bon déroulement de l'action de formation visée conformément aux prérequis indiqués dans le programme de formation communiqué.

* nous consulter pour la faisabilité en distanciel

** ratio variable selon le cours suivi

Accessibilité aux personnes en situation de handicap

Détails de la session

Du 16 novembre 2026 au 18 novembre 2026 (3 jours / 21 heures)

À distance

Du 16 novembre 2026 au 18 novembre 2026

À distance

- 16 novembre 2026 9h00 - 16h00
- 17 novembre 2026 9h00 - 16h00
- 18 novembre 2026 9h00 - 16h00