Installation, configuration et utilisation d'Ollama

Table des matières / Sommaire

1. Cahier des charges – Expression des besoins

1.1 Descriptif de l’existant

Dans le cadre de l'évolution de la nouvelle infrastructure, l'équipe technique rédige régulièrement des scripts d'automatisation, des requêtes SQL et de la documentation technique. Actuellement, l'équipe utilise des moteurs de recherche classiques pour trouver de l'aide, ce qui peut s'avérer chronophage, ou bien s'appuie sur des IA hébergées dans le Cloud public, ce qui pose un risque de fuite de données (code source, configurations IP).

1.2 Besoin(s)

- Intégrer un assistant conversationnel (LLM - Large Language Model) directement sur l'environnement de travail.
- Fournir une aide à la rédaction de scripts (Bash, Python) et à la résolution de problèmes système.
- S'assurer que les prompts et les données générées restent strictement en interne.

1.3 Contrainte(s)

- **Confidentialité absolue :** L'outil doit fonctionner "Air-gapped" (capable de tourner sans aucune connexion Internet une fois installé).
- **Architecture matérielle :** Le logiciel doit être compatible avec les architectures ARM (Apple Silicon) pour exploiter l'accélération matérielle et la mémoire unifiée du poste de travail.
- **Format Open Source :** Utilisation de modèles libres (Llama 3, Mistral, etc.) sans coût de licence.

2. Ressources

2.1 Ressources mises à disposition

- **Matériel :** Un poste de travail sous environnement **macOS** doté d'une architecture ARM récente et d'au moins 16 Go de mémoire vive unifiée, permettant de charger des modèles d'IA lourds de manière fluide.
- **Réseau :** Une connexion Internet ponctuelle requise uniquement pour la phase de téléchargement (pull) des poids du modèle depuis le registre officiel.

2.2 Ressources nécessaires

- **Logiciel :** Le binaire d'installation de l'application **Ollama** pour macOS.
- **Administration :** L'application "Terminal" native de macOS pour l'exécution des requêtes et la gestion des modèles en ligne de commande.

2.3 Gestion des ressources

La gestion de l'espace disque est un point de vigilance majeur : chaque modèle de langage pèse entre 4 et 8 Go en moyenne. Les modèles obsolètes ou non utilisés devront être supprimés régulièrement via l'interface en ligne de commande (CLI) pour ne pas saturer le stockage SSD du poste de travail.

3. Analyse

3.1 Descriptifs des solutions

- **IA Cloud (Ex: ChatGPT, Claude) :** Solutions prêtes à l'emploi via navigateur web. Très performantes, mais nécessitent l'envoi des données sur des serveurs tiers et souvent un abonnement mensuel pour un usage professionnel.
- **IA Locale (Ex: Ollama, LM Studio) :** Solutions s'installant directement sur le système d'exploitation. Le traitement de la donnée se fait par le processeur local.

3.2 Comparaison des solutions

Critère	ChatGPT (Cloud)	Ollama (Local - Choisi)
Confidentialité	Partagée avec l'éditeur externe	**Totale (Zéro fuite de données)**
Coût	Abonnement (Version Pro)	**100% Gratuit (Open Source)**
Dépendance Internet	Indispensable en permanence	**Aucune (après le 1er téléchargement)**
Interface	Navigateur Web (SaaS)	**Terminal (Ligne de commande)**

3.3 Choix d'une solution – Argumentation

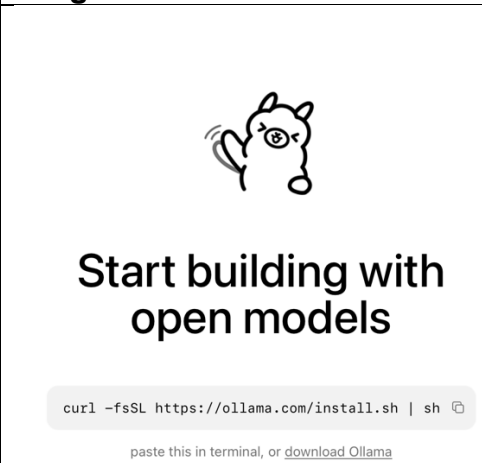
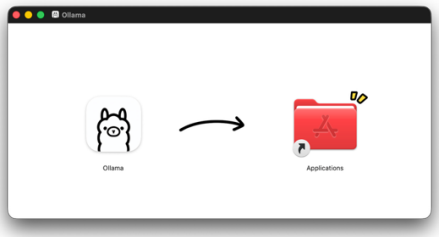
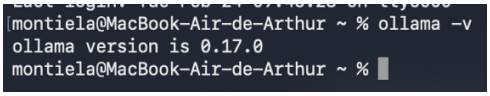
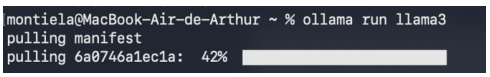
Le choix s'est porté sur **Ollama**. Il s'intègre parfaitement à l'écosystème macOS et permet de faire tourner des modèles très performants (comme Llama 3) de manière totalement locale. L'utilisation du terminal correspond parfaitement aux habitudes de travail d'un administrateur système. De plus, il répond strictement à l'exigence de souveraineté des données de l'entreprise.

3.4 Étude de l'impact sur le SI existant

- **Sécurité :** L'empreinte de sécurité est excellente. Aucun code source propriétaire ou adresse IP d'infrastructure ne quitte le réseau de l'entreprise.
- **Performance client :** Lors de la génération de texte, l'application sollicite intensément le processeur neural et la mémoire du Mac. Il faudra veiller à ne pas lancer de modèles trop lourds (ex: modèles de 70 milliards de paramètres) pour éviter de ralentir le poste.
- **Productivité :** Gain de temps majeur pour l'équipe technique dans la rédaction de procédures (GPA) et le débuggage de scripts.

4. Mise en place

4.1 Réalisation en suivant le phasage énoncé précédemment

Étape	Description	Images
1	**Téléchargement :** Récupération de l'exécutable pour macOS (Ollama.dmg) depuis le site officiel et décompression.	
2	**Installation système :** Glisser-déposer (Drag & Drop) de l'application Ollama vers le dossier "Applications", puis premier lancement pour valider l'installation de l'outil en ligne de commande.	
3	**Vérification :** Ouverture de l'application "Terminal" native de macOS et vérification de la bonne installation du service en arrière-plan. ollama -v	
4	**Téléchargement du modèle :** Récupération (pull) du modèle de langage Open Source (ex: Llama 3). ollama run llama3	
5	**Test final :** Premier échange en langage naturel avec l'IA directement dans le terminal pour tester la réactivité. >>> Rédige moi un script bash de désactivation du Wifi à 22H00	

4.3 Rapport de tests

Test de conformité	Action effectuée	Résultat attendu	Résultat obtenu
Exécution CLI	Saisir la commande ollama	Le menu d'aide s'affiche sans erreur	**OK**
Mode Hors-ligne	Couper le Wi-Fi et envoyer un prompt	L'IA génère une réponse (100% local)	**OK**
Performance	Demander la création d'un script Bash	Génération fluide et cohérente	**OK**

5. Bilan

5.1 Conclusion

Le déploiement de l'intelligence artificielle locale Ollama est validé. L'outil est pleinement opérationnel sur le poste de travail macOS et interagit de manière fluide via le terminal. L'entreprise dispose désormais d'un assistant technique intelligent garantissant une confidentialité totale, sans dépendance à un fournisseur Cloud.

5.2 Auto-critique / Auto-évaluation

- **Points forts :** La simplicité d'installation sur macOS est remarquable (aucun besoin de gérer les dépendances Python ou les pilotes graphiques manuellement). Les temps de réponse du modèle Llama 3 sont excellents grâce à l'architecture matérielle du poste.
- **Points de vigilance :** L'usage d'Ollama via le terminal brut peut être austère pour des utilisateurs non techniques. À l'avenir, le déploiement d'une interface graphique Web (comme *Open WebUI*) via Docker pourrait être envisagé pour rendre l'outil plus convivial pour d'autres collaborateurs de l'entreprise.
- **Surveillance :** Surveiller l'espace de stockage restant sur le disque SSD après le téléchargement de plusieurs modèles différents.

5.3 Compétence(s) SISR mobilisée(s)

- **Participer à l'évolution d'un site d'exploitation informatique** (Intégration de nouvelles technologies IA).
- **Mettre à disposition des utilisateurs un environnement de travail** (Déploiement d'un outil d'assistance au développement).