

Cahier des Charges — Projet: Déploiement Cloud et Pipeline DevOps pour une Application de Génération Automatique de Prompts d'Images

1. Contexte et objectifs

Concevoir et déployer un projet cloud complet démontrant une expertise DevOps et cloud engineering sur Azure. Le projet inclut une application Python (Prompt Generator) permettant de générer automatiquement des prompts descriptifs à partir d'images, déployée dans une infrastructure cloud moderne, sécurisée et automatisée.

2. Périmètre du projet

- API Python (Flask/FastAPI) pour générer un prompt à partir d'une image
 - Containerisation Docker optimisée
 - Gestion des images via Azure Container Registry (ACR)
 - Hébergement via Azure Container Apps
 - Pipeline CI/CD **multi-environnements** Staging puis Production, avec validations manuelles
 - Automatisation des notifications de déploiement et d'erreur via n8n ou Power Automate
 - Monitoring complet avec Azure Monitor et Log Analytics : logs applicatifs, métriques, alertes
 - Provisionnement complet de l'infrastructure via **Infrastructure as Code** (Bicep ou Terraform)
 - Gestion sécurisée des secrets et clés via Azure Key Vault
-

3. Fonctionnalités détaillées

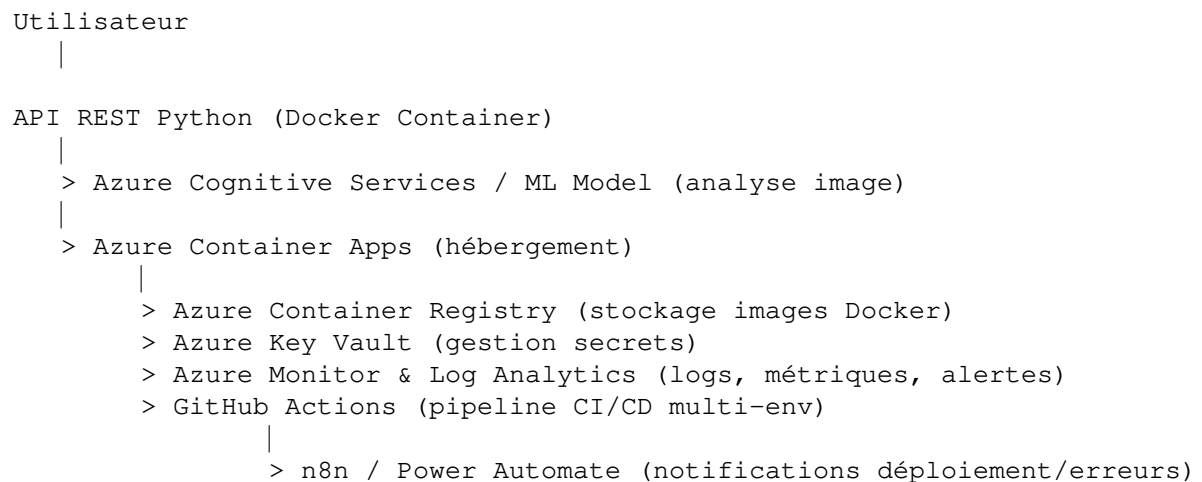
Fonctionnalité	Description	Priorité
Upload d'image via API	Endpoint REST acceptant une image (upload ou URL)	Haute
Génération de prompt IA	Intégration Azure Cognitive Services ou modèle ML pour produire une description textuelle précise	Haute
Conteneurisation Docker	Dockerfile avec build multi-étapes et optimisation	Haute
Registry Azure Container	Stockage sécurisé des images Docker	Haute
Déploiement Azure Container Apps	Hébergement scalable, exposé via un domaine public	Haute

Pipeline CI/CD GitHub Actions	Automatisation complète avec étapes build, test, push image, déploiement	Haute
Multi-environnements	Déploiement en Staging puis Production avec validation manuelle entre les deux environnements	Haute
Notifications automatisées	Envoi d'emails, messages Slack/Teams ou autre via workflows n8n ou Power Automate sur événements clés	Moyenne
Monitoring et alertes	Configuration d'Azure Monitor + Log Analytics pour collecter logs, métriques et déclencher alertes	Moyenne
Infrastructure as Code (IaC)	Déploiement infrastructure via scripts Bicep ou Terraform (ACR, Container Apps, Key Vault, Log Analytics)	Haute
Gestion des secrets	Stockage et injection sécurisée des clés API via Azure Key Vault	Haute

4. Contraintes techniques

- Langage : Python 3.x (Flask ou FastAPI)
 - Docker : multi-stage build, image légère
 - Azure Container Registry & Azure Container Apps
 - Pipeline CI/CD avec GitHub Actions
 - Provisionnement via Infrastructure as Code (Bicep ou Terraform)
 - Monitoring avec Azure Monitor et Log Analytics
 - Notifications via n8n (self-hosted ou SaaS) ou Power Automate
 - Gestion des secrets via Azure Key Vault
-

5. Architecture technique



6. Livrables attendus

- Code source complet (API + Dockerfile)

- Scripts IaC Bicep ou Terraform
 - Pipelines GitHub Actions pour multi-env staging/production
 - Documentation technique détaillée
 - Rapport de projet académique (présentation, architecture, démo)
 - Configuration du monitoring et notifications opérationnelles
-

7. Planning prévisionnel

Phase	Durée es- timée	Description
Analyse & concep- tion	4 jours	Définition des besoins, choix techniques
Développement API	4 jours	Implémentation API + intégration IA
Dockerisation & tests	2 jours	Création du Dockerfile et tests locaux
Infrastructure as Code	3 jours	Développement scripts Bicep/Terraform
Pipeline CI/CD multi-env	3 jours	Automatisation build, déploiement stag- ing/prod
Monitoring & notifi- cations	2 jours	Configuration Azure Monitor, alertes, n8n/Power Automate
Documentation & rapport	2 jours	Rédaction complète et préparation soutenance

8. Critères d'évaluation

- Respect des fonctionnalités décrites
 - Qualité et maintenabilité du code
 - Automatisation complète du déploiement multi-env
 - Fonctionnement opérationnel du monitoring et alertes
 - Mise en place efficace des notifications automatiques
 - Documentation claire et exhaustive
-