Partie 2: Questions Théoriques

1.

Plan de déploiement étape par étape

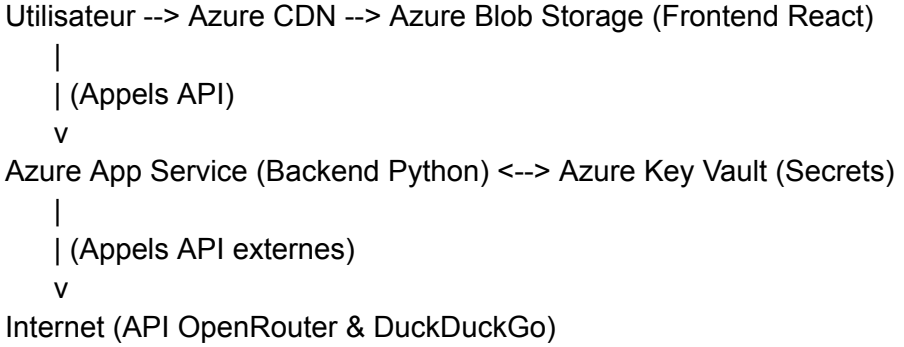
1. Prérequis:

   Organisationnels : Un abonnement Azure avec des droits de Propriétaire ou Contributeur ; un budget alloué.

   Techniques : Docker et Azure CLI installés localement ; un compte OpenRouter pour la clé API ; un repo GitHub pour le code source.

2. **Containerisation du Backend :**
   - Créer un `Dockerfile` pour l'application Python/Flask.
   - Construire l'image Docker et la pousser vers un registre privé **Azure Container Registry (ACR)**.
3. **Déploiement du Backend :**
   - Créer un **Azure App Service Plan**
   - Déployer une **Azure App Service for Containers** en utilisant l'image stockée dans ACR.
4. **Déploiement du Frontend :**
   - Compiler l'application React en fichiers statiques (`npm run build`).
   - Activer la fonctionnalité de site web statique sur un compte **Azure Blob Storage**.
   - Uploader les fichiers statiques dans le conteneur `$web` du Blob Storage.
5. **Mise en réseau et Sécurité :**
   - Placer un **Azure CDN** devant le Blob Storage pour une distribution rapide et la gestion du HTTPS.
   - Configurer les règles CORS sur l'App Service pour n'autoriser que le domaine du frontend.
   - Stocker la clé API OpenRouter dans **Azure Key Vault**. L'App Service y accédera de manière sécurisée via une **Managed Identity**, évitant ainsi de stocker des secrets dans le code.

Architecture Scalable



Services Azure et Justifications

- **Azure App Service :** PaaS entièrement géré, idéal pour les applications web. Fournit une mise à l'échelle automatique, un déploiement intégré et la sécurité.
- **Azure Blob Storage (Static Website) :** Solution la plus rentable et performante pour héberger un site statique.
- **Azure CDN :** Réduit la latence mondiale et décharge le trafic du stockage.
- **Azure Key Vault :** La solution standard et sécurisée pour la gestion des secrets en production.
- **Azure Monitor / Log Analytics :** Essentiel pour la surveillance, les alertes et le diagnostic en production.

Estimation des Coûts Mensuels

- **App Service Plan (B1) :** ~55€ / mois
- **Blob Storage (LRS, 10 Go) :** ~0.20€ / mois
- **CDN (Standard, 100 Go de transfert) :** ~8€ / mois
- **Key Vault & Monitor :** Coût négligeable pour ce volume.
- **Total Estimé : ~65-75€ / mois.**

Considérations de Sécurité

- **Gestion des Secrets :** Aucune clé en clair dans le code ; tout passe par Azure Key Vault.
- **Permissions :** Appliquer le principe du moindre privilège avec les Managed Identities. L'App Service a uniquement un accès `get` aux secrets nécessaires dans le Key Vault.
- **Accès Réseau :** Restreindre les adresses IP pouvant accéder au backend si nécessaire. Activer l'option "HTTPS Only".

2.

Stratégie de Mise en Production

- Pipeline CI/CD:
  - Utiliser **GitHub Actions**.
  - **CI :** À chaque `push` sur la branche `main`, le workflow exécute les tests, build l'image Docker du backend et les fichiers statiques du frontend.
  - **CD :** Si la CI réussit, le workflow pousse l'image vers ACR, déploie les fichiers sur Blob Storage, et met à jour l'App Service.
- Monitoring et Logs:
  - Configurer **Azure Monitor** pour collecter les métriques (CPU, mémoire) de l'App Service.
  - Activer les logs applicatifs et les streamer vers **Log Analytics** pour pouvoir requêter et créer des alertes sur les erreurs.
- Stratégie de Backup:
  - **Code :** Le repository Git est la sauvegarde principale.

- **Infrastructure :** Définir toute l'infrastructure via du code (**Terraform** ou **Bicep**) pour pouvoir la recréer à l'identique rapidement.
- **Configuration :** Les configurations et secrets dans Key Vault sont versionnés et peuvent être sauvegardés.