

Projet – Migration Cloud pour WebUnity

Contexte professionnel

Vous êtes développeur full stack dans l'entreprise WebUnity, une startup spécialisée dans la gestion de contenus web pour des TPE/PME.

Jusqu'à présent, WebUnity hébergeait toutes ses applications web sur ses propres serveurs internes. Suite à une croissance rapide, la direction a décidé de migrer progressivement toutes ses applications web vers le cloud Azure, pour :

- Gagner en scalabilité

La scalabilité (ou montée en charge) désigne la capacité d'une application ou d'une infrastructure à gérer une augmentation du trafic ou du volume de données, sans perte de performance.

Avec Azure :

- Azure App Service permet de passer d'un plan gratuit à un plan payant en 2 clics.
- On peut activer l'auto-scaling, c'est-à-dire :
 - Augmenter automatiquement le nombre d'instances de l'application quand le trafic augmente.
 - Réduire les ressources en période creuse, pour optimiser les coûts.

Exemple :

Un site e-commerce gère 200 visiteurs/jour habituellement. Le Black Friday, le trafic passe à 5000 visiteurs/jour.

Azure peut allouer automatiquement plus de serveurs en backend pour encaisser cette charge sans intervention humaine.

- Réduire la maintenance physique

Avant (On-premise) :

L'entreprise doit acheter et gérer ses propres serveurs.

Elle doit :

- Installer l'OS, les mises à jour de sécurité
- Gérer les pannes matérielles (disques, réseau...)
- Surveiller la température, la consommation, etc.
- Avoir des techniciens et des astreintes

Maintenant Avec Azure :

- Aucun matériel à entretenir : Microsoft gère l'infrastructure physique.
- Les mises à jour de sécurité système sont automatiques.
- Le développeur ne s'occupe que du code et des paramètres de l'app.

Avantage :

Cela recentre le travail des développeurs sur le métier (dev, tests, qualité) plutôt que sur des tâches d'infra lourdes et chronophages.

- Faciliter les déploiements CI/CD

CI : Continuous Integration (Intégration Continue)

Chaque push déclenche une vérification (build, tests...)

CD : Continuous Deployment (Déploiement Continu)

Le code validé est déployé automatiquement sur un environnement (test, prod...).

Avec Azure + GitHub :

- Azure App Service propose l'intégration directe avec GitHub Actions :
 - À chaque push sur main/master, le code est automatiquement déployé.
 - Plus besoin de FTP ou de déploiement manuel.
- On peut configurer plusieurs environnements (dev, staging, prod) en pipeline.

Exemple :

Un étudiant push son app sur GitHub. En 30 secondes, l'app est buildée, testée, et disponible en ligne via HTTPS sans aucune action manuelle.

- | |
|---|
| <ul style="list-style-type: none">• Améliorer la sécurité (HTTPS, monitoring intégré) |
|---|

HTTPS par défaut

- Azure App Service fournit un certificat SSL gratuit automatiquement :
 - Le site est sécurisé par HTTPS (chiffrement des données).
 - Cela protège les utilisateurs contre les attaques de type "man-in-the-middle".

Monitoring intégré

- Azure propose Application Insights :
 - Suivi en temps réel des erreurs (500, 404...)
 - Temps de réponse
 - Logs détaillés
 - Tableau de bord visuel
- Possibilité de recevoir des alertes automatiques (ex : CPU > 80%)

Exemple :

Si une route plante avec une erreur 500, Azure Insights peut détecter le problème, en montrer la cause (ex : variable manquante), et alerter l'équipe Dev.

Mission confiée

La direction vous confie la migration d'une première application de test.

Vous devez :

1. Développer une web app simple ou récupérer un projet existant.
2. La versionner sur GitHub
3. La déployer sur Azure App Service
4. L'exposer via HTTPS
5. Proposer un chiffrage estimatif du coût annuel de cette app

Livrables attendus

1. Procédure complète de la mise en place de ce projet sur Azure avec visuels et explications comprenant également :

- Lien vers le repo GitHub de l'application
- Tableau de chiffrage estimatif annuel en toute fin de rapport (Utilisation de Azure Pricing Calculator)