

Projet Étudiant : Déploiement d'une Plateforme E-commerce Scalable sur AWS

Titre du Projet : GreenLeaf - Votre E-commerce Éco-responsable sur le Cloud

Durée : 1 semaine

Équipes : 3 à 5 étudiants

1. Contexte de l'Entreprise : GreenLeaf SAS

GreenLeaf est une jeune startup française qui commercialise des produits du quotidien éco-responsables et durables (cosmétiques solides, accessoires zéro déchet, produits d'entretien écologiques). La mission de GreenLeaf est de rendre la consommation durable accessible à tous.

Après une première phase de vente réussie via les réseaux sociaux et des marketplaces, GreenLeaf souhaite lancer sa propre plateforme e-commerce pour maîtriser son image de marque, améliorer l'expérience client et gérer sa croissance.

La startup a des ambitions fortes mais des moyens financiers limités. Le choix de la solution technique et de l'infrastructure est donc crucial et doit allier **performance, scalabilité et maîtrise des coûts**.

2. Mission du Projet

En tant que cabinet de conseil Cloud/DevOps, votre équipe a été mandatée par GreenLeaf pour **concevoir, déployer et documenter une infrastructure cloud sur AWS capable d'héberger leur future plateforme e-commerce**.

Le projet doit non seulement aboutir à une solution technique fonctionnelle mais aussi intégrer une forte dimension de **chefferie de projet** et de **FinOps** pour garantir la viabilité économique de la solution.

La solution e-commerce open-source choisie pour ce projet est **Magento Open Source**, réputée pour sa richesse fonctionnelle mais aussi pour sa complexité et ses exigences en termes de ressources, ce qui en fait un excellent cas d'étude.

3. Exigences et Contraintes

3.1. Exigences Métier et Fonctionnelles

- **Haute Disponibilité** : La plateforme doit être accessible 24/7. Aucune interruption de service n'est acceptable en dehors des fenêtres de maintenance planifiées.
- **Scalabilité** : L'infrastructure doit pouvoir absorber des pics de trafic importants (lancement de produit, soldes, campagnes marketing) sans dégradation des performances.
- **Performance** : Le temps de chargement des pages doit être optimisé pour garantir une bonne expérience utilisateur et un bon référencement (SEO).
- **Sécurité** : Les données des clients et les transactions doivent être sécurisées selon les meilleures pratiques du web.

3.2. Contraintes Techniques et FinOps

- **Cloud Provider** : L'infrastructure doit être déployée **exclusivement sur AWS**.
- **Solution E-commerce** : **Magento Open Source** (dernière version stable).
- **Infrastructure as Code (IaC)** : L'ensemble de l'infrastructure AWS doit être provisionné via **Terraform**.
- **Gestion de Configuration** : La configuration des serveurs (installation de Magento, dépendances, etc.) doit être automatisée avec **Ansible**.
- **Budget Mensuel Cible** : L'architecture proposée ne doit pas dépasser un coût estimé de **500 \$/mois**. Chaque choix technique devra être justifié d'un point de vue coût/performance.

4. Rôles Suggérés au sein de l'Équipe

Pour mener à bien ce projet, il est recommandé de répartir les rôles suivants :

- **Chef de Projet / Architecte Cloud** : Responsable du planning, de la coordination, de la conception de l'architecture globale et garant de la vision d'ensemble.
- **Ingénieur DevOps / Sécurité** : En charge de l'automatisation (Terraform, Ansible), de la mise en place du CI/CD (optionnel) et de l'application des bonnes pratiques de sécurité.
- **Spécialiste FinOps & Monitoring** : Responsable du suivi des coûts, de l'optimisation du budget, de la mise en place de la surveillance (CloudWatch) et des stratégies de performance.

5. Livrables Attendus en Fin de Projet

1. **Document d'Architecture Technique (DAT)** : Un document détaillé présentant l'architecture AWS, les choix de services, les schémas réseau et les flux de données.

2. **Code Source de l'Infrastructure** : Les fichiers Terraform et les playbooks Ansible, versionnés sur un dépôt Git.
3. **Rapport d'Analyse FinOps** : Un document présentant l'estimation des coûts mensuels, les stratégies d'optimisation mises en place et un plan de suivi budgétaire.
4. **Guide de Déploiement et d'Exploitation** : Un manuel expliquant comment déployer l'infrastructure de zéro et les procédures de maintenance de base.
5. **Présentation Finale** : Un support de présentation (type PowerPoint) synthétisant le projet, l'architecture, les défis rencontrés et les résultats obtenus, à destination du comité de direction de GreenLeaf.

6. Critères de Réussite du Projet

Le projet sera considéré comme réussi si les critères suivants sont atteints :

6.1. Critères Techniques

- **Infrastructure Fonctionnelle** : La plateforme Magento est accessible via une URL publique (celle de l'ALB ou de CloudFront) et toutes les fonctionnalités de base sont opérationnelles (navigation, ajout au panier, création de compte).
- **Haute Disponibilité** : L'architecture est déployée sur au moins 2 Zones de Disponibilité. En cas de terminaison manuelle d'une instance EC2, l'application reste accessible grâce à l'Auto Scaling.
- **Scalabilité Démontrée** : Le groupe Auto Scaling est capable d'ajouter et de retirer des instances en fonction de la charge. Une simulation de montée en charge doit être documentée.
- **Infrastructure as Code** : L'intégralité de l'infrastructure est provisionnée via Terraform. Le code doit être propre, commenté et versionné sur Git.
- **Automatisation** : La configuration des serveurs est entièrement automatisée avec Ansible. Un nouveau serveur lancé par l'Auto Scaling doit être fonctionnel sans intervention manuelle.
- **Monitoring** : Au moins 3 alarmes CloudWatch pertinentes sont configurées et fonctionnelles.

6.2. Critères de Gestion de Projet

- **Respect du Planning** : Les jalons hebdomadaires sont globalement respectés. Des ajustements sont acceptables s'ils sont justifiés et documentés.

- **Collaboration** : Le dépôt Git montre une contribution équilibrée de tous les membres de l'équipe (commits réguliers).
- **Documentation** : Tous les livrables sont complets, bien rédigés et professionnels.

6.3. Critères FinOps

- **Respect du Budget** : Le coût mensuel estimé de l'architecture (hors crédits gratuits) ne dépasse pas 500 \$.
- **Justification des Choix** : Chaque service et chaque configuration sont justifiés d'un point de vue coût/performance dans le rapport FinOps.
- **Recommandations Concrètes** : Le rapport FinOps propose des stratégies d'optimisation chiffrées et réalistes pour réduire les coûts à moyen terme.

7. Ressources et Outils Recommandés

7.1. Outils de Gestion de Projet

- **Trello / Jira / GitHub Projects** : Pour gérer le backlog et suivre l'avancement des tâches.
- **Slack / Discord / Microsoft Teams** : Pour la communication quotidienne de l'équipe.

7.2. Outils Techniques

- **Terraform** : Pour le provisionnement de l'infrastructure.
- **Ansible** : Pour la gestion de configuration.
- **Git / GitHub / GitLab** : Pour le versioning du code.
- **AWS CLI** : Pour interagir avec AWS en ligne de commande.
- **Visual Studio Code** : Éditeur de code recommandé avec les extensions Terraform et Ansible.

7.3. Documentation AWS

- **Documentation officielle AWS** : <https://docs.aws.amazon.com/>
- **AWS Pricing Calculator** : <https://calculator.aws/>
- **AWS Well-Architected Framework** : <https://aws.amazon.com/architecture/well-architected/>

7.4. Ressources Magento

- **Documentation Magento Open Source** : <https://devdocs.magento.com/>

- **Tutoriels d'installation** : Recherchez des guides pour installer Magento sur Ubuntu/Amazon Linux avec Nginx et MySQL.

8. Conseils pour la Réussite

- **Commencez Simple** : Ne cherchez pas à tout faire parfaitement dès le début. Déployez d'abord une version minimale fonctionnelle (MVP) puis itérez.
 - **Testez Régulièrement** : Chaque fois que vous ajoutez un composant, testez-le immédiatement. Ne laissez pas les bugs s'accumuler.
 - **Documentez au Fur et à Mesure** : Ne laissez pas la documentation pour la fin. Prenez des notes à chaque étape, cela facilitera grandement la rédaction des livrables finaux.
 - **Communiquez** : Les problèmes techniques sont normaux. Parlez-en rapidement à votre équipe ou à votre encadrant.
 - **Surveillez les Coûts dès le Début** : Activez AWS Budgets dès la première semaine pour éviter les mauvaises surprises.
-