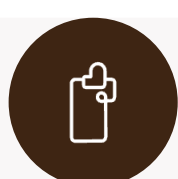


# Documentation de l'Infrastructure PrestaShop sur AWS

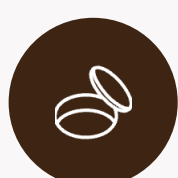
Réalisé par: Fanta DOUMBIA –  
Ndiambe GUEYE – Hela ABDELKAFI

*Ce document technique s'adresse aux Ingénieurs DevOps et administrateurs cloud. Il détaille l'architecture, le déploiement et la maintenance d'une plateforme e-commerce PrestaShop automatisée et résiliente.*



## Public Cible

*Ingénieurs DevOps et administrateurs cloud cherchant une solution de plateforme e-commerce automatisée.*



## Technologie Clé

*Entièrement gérée par Terraform sur Amazon Web Services (AWS), assurant une gestion complète par Infrastructure as Code.*




## Objectif Principal

*Fournir une solution hautement disponible et scalable pour l'automatisation complète du cycle de vie de l'application et de son infrastructure associée.*




# Introduction et Composants Clés

*Le projet déploie un environnement PrestaShop entièrement automatisé sur AWS, en s'appuyant sur une infrastructure cloud native. La résilience et la scalabilité sont assurées par l'intégration de services gérés clés d'AWS.*




### ALB (Application Load Balancer)

*Assure la répartition de la charge du trafic web et gère les certificats SSL/TLS.*




### ASG (Auto Scaling Group)

*Gère la flotte d'instances EC2 exécutant PrestaShop dans des conteneurs Docker, garantissant élasticité et haute disponibilité.*




### RDS MySQL

*Base de données relationnelle gérée, offrant durabilité et performance pour les données de l'e-commerce.*



### EFS (Elastic File System)

*Stockage persistant et partagé pour le répertoire `/var/www/html`, critique pour les images, thèmes et modules.*



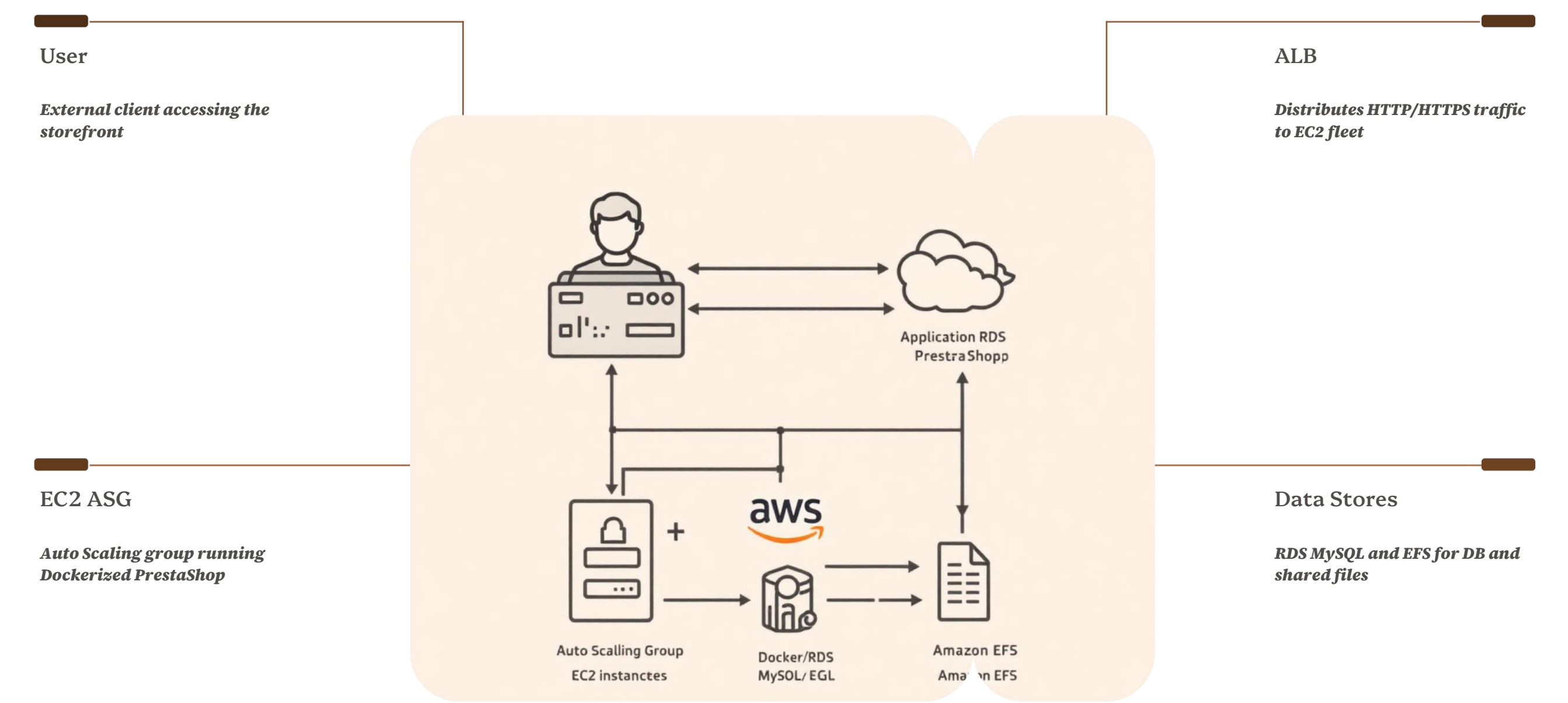
### SSM Parameter Store

*Utilisé pour stocker les informations sensibles (mots de passe, secrets de la base de données) en toute sécurité et les injecter dynamiquement.*

*L'intégralité du déploiement est pilotée par Terraform, garantissant l'immutabilité et la reproductibilité de l'infrastructure.*

# Architecture de la Solution PrestaShop sur AWS

*L'architecture est conçue pour maximiser la disponibilité, la sécurité et la performance, en séparant les couches de présentation, d'application et de données.*



## Description des Flux

- **ALB :** Point d'entrée unique, il distribue les requêtes HTTP/HTTPS vers le groupe cible des instances EC2.
- **EC2/Docker :** Chaque instance exécute le conteneur PrestaShop. Elles montent EFS sur un point de montage partagé pour la persistance des données du site.
- **RDS :** La base de données est isolée dans des sous-réseaux privés, accessible uniquement par les instances EC2, assurant une sécurité accrue.
- **EFS :** Garantit que toutes les instances EC2 partagent le même état de fichier pour les assets de PrestaShop (images, modules).



# Exigences Techniques et Prérequis au Déploiement

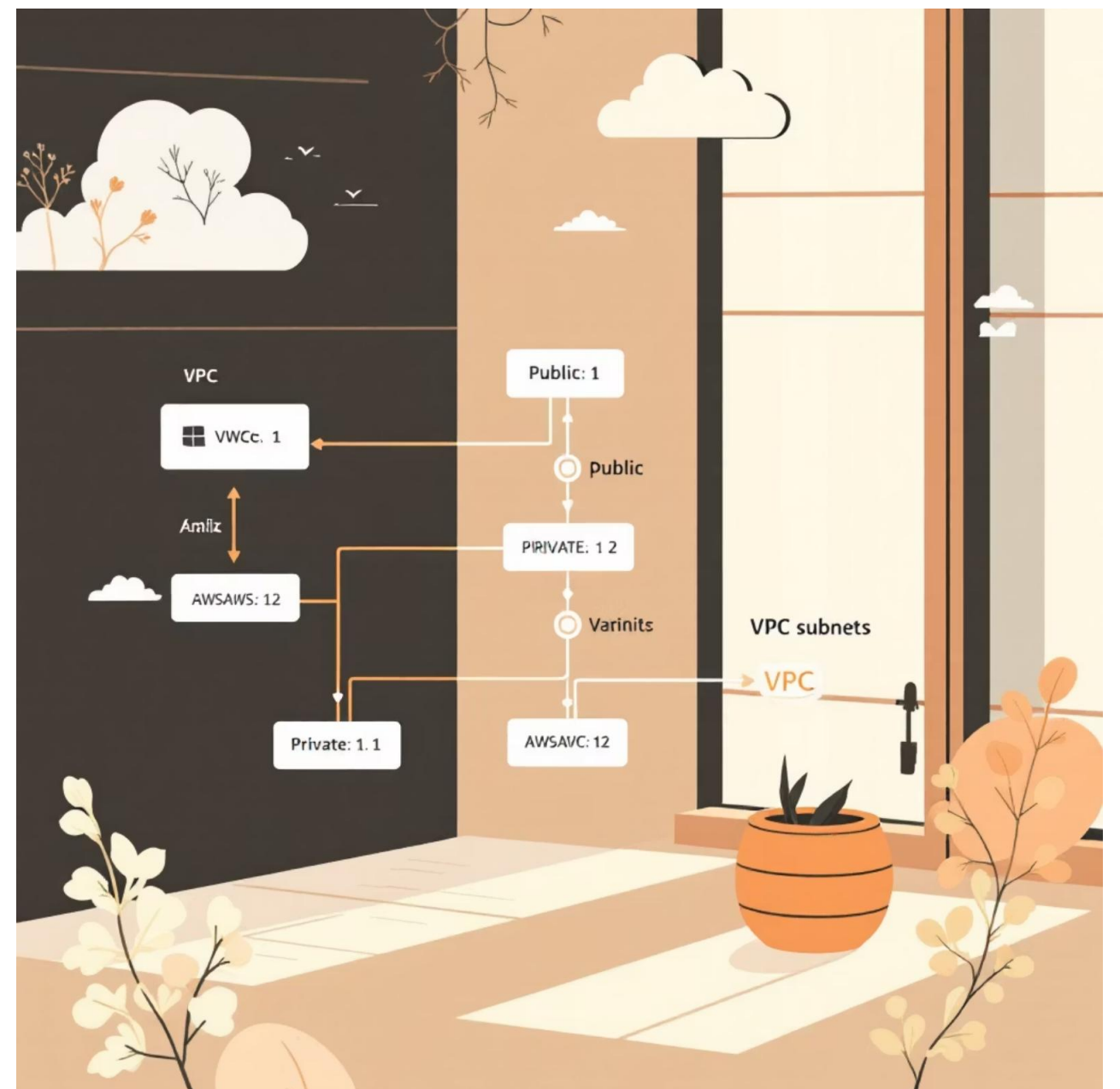
*Pour un déploiement réussi de l'infrastructure PrestaShop via Terraform, l'environnement de travail doit respecter les configurations suivantes. Ces éléments sont fondamentaux pour interagir correctement avec AWS et garantir la stabilité du déploiement.*

## Outils et Versions

- **AWS CLI** : Configurée et authentifiée avec des droits IAM suffisants pour créer l'ensemble des ressources (EC2, ALB, RDS, EFS, SSM).
- **Terraform** : Version 1.5 ou supérieure requise pour l'utilisation optimale des fonctionnalités du fournisseur AWS.
- **Docker** : L'image de base des instances EC2 doit être compatible avec l'installation de Docker et l'exécution de conteneurs (ex: Amazon Linux 2).

## Configuration AWS Nécessaire

- **VPC** : Une Virtual Private Cloud préexistante ou créée via Terraform, segmentée en sous-réseaux publics (pour ALB) et privés (pour EC2 et RDS).
- **Subnets Privés** : Essentiels pour isoler la couche applicative (EC2/RDS) du trafic internet direct.
- **Accès Réseau** : Des Groupes de Sécurité correctement configurés sont nécessaires pour autoriser le trafic entre ALB et EC2 (HTTP/S), et entre EC2 et RDS (MySQL).



# Procédure de Déploiement via Terraform

*Le déploiement de l'infrastructure est standardisé grâce à Infrastructure as Code (IaC) en utilisant Terraform. Les étapes suivantes assurent un cycle de vie d'infrastructure maîtrisé, allant de l'initialisation à l'application des changements.*



## Initialisation du Répertoire

*Exécutez `terraform init`. Cette commande télécharge les plugins et initialise le répertoire de travail pour le fournisseur AWS.*



## Vérification du Plan d'Exécution

*Générez et examinez le plan d'exécution avec `terraform plan -out=tfplan`. Assurez-vous que les ressources créées correspondent aux attentes, notamment en termes de coûts et de dépendances.*



## Application des Modifications

*Lancez l'application avec `terraform apply tfplan`. Confirmez l'opération (en tapant "yes") pour démarrer la création des ressources AWS.*



## Récupération des Sorties Clés

*Une fois l'application terminée, les outputs fournissent des informations vitales pour la configuration externe et le debugging. Les plus importants sont :*

- `alb_dns_name`: *DNS public de l'Application Load Balancer (point d'accès du site).*
- `tg_arn`: *Amazon Resource Name du Target Group (utile pour le suivi des cibles EC2).*

*URL pour accéder au site:*

<http://taylor-dev-alb-1036715090.eu-west-3.elb.amazonaws.com/>

# Détails de l'Initialisation du Conteneur PrestaShop

*Le script `user_data.sh` exécuté sur chaque instance EC2 est la clé de l'automatisation. Il garantit que chaque nouvelle instance de l'ASG est prête à servir l'application immédiatement après son démarrage.*

*Le script exécute les tâches suivantes en séquence :*

- *Mise à jour du système d'exploitation et installation du moteur Docker.*
- *Configuration et montage du volume EFS sur le chemin local `/mnt/efs/psdata`.*
- *Récupération sécurisée des secrets (comme le mot de passe MySQL) depuis SSM Parameter Store.*
- *Lancement du conteneur PrestaShop en utilisant l'image Docker officielle, en s'assurant que les variables d'environnement nécessaires à l'installation sont passées :*

## **Variables d'environnement critiques pour l'auto-installation de PrestaShop :**

- `PS_DOMAIN` : **Défini sur le DNS de l'ALB pour les liens internes.**
- `PS_INSTALL_AUTO=1` : **Déclenche l'installation automatique du CMS.**
- `PS_HANDLE_DYNAMIC_DOMAIN=1` : **Permet à PrestaShop de gérer les changements d'hôte dynamique, essentiel dans un environnement ASG.**



# Maintenance et Gestion du Cycle de Vie

***Une maintenance efficace implique la mise à jour cohérente de l'application (PrestaShop) et de son infrastructure (Terraform), ainsi que des opérations courantes de gestion de cache et d'assets.***

## Mise à Jour de l'Application (PrestaShop)

***Les mises à jour se font par renouvellement des conteneurs après un pull de la nouvelle image Docker :***

```
docker pull
prestashop/prestashop:latest
docker rm
-f prestashop# Relancer avec la même
commande user_data
```

***Ce processus, combiné à l'ASG, peut être orchestré pour un déploiement "rolling" sans interruption de service.***

## Opérations de Cache et Assets

***Il est crucial de purger le cache et, si nécessaire, de reconstruire les miniatures d'images après une mise à jour ou un changement de thème :***

```
# Purge du cache et redémarrage
d'Apachedocker exec -it prestashop
bash -lc 'rm -rf
/var/www/html/var/cache/*; apachectl
-k graceful'# Reconstruction des
images produits
docker exec -it
prestashop bash -lc 'php -d
memory_limit=-1
/tmp/rebuild_thumbs.php'
```

***La mise à jour de l'infrastructure (taille d'instance, version RDS) se fait en ré-appliquant simplement les changements via terraform plan puis terraform apply.***

# Résolution des Problèmes Courants

Face aux défis courants d'une infrastructure conteneurisée et scalable, une approche systématique de résolution est requise. Les problèmes suivants sont les plus fréquemment rencontrés :

Problème	Solution Technique
CSS ou JS cassé	Vérifier les variables d'environnement PS_CSS_THEME_CACHE et PS_JS_THEME_CACHE. Vider le cache du conteneur et reconstruire le fichier .htaccess.
Images produits manquantes	Vérifier la persistance des fichiers dans /var/www/html/img/p/ sur EFS. Si elles existent, exécuter rebuild_thumbs.php pour régénérer les miniatures.
Redirection vers l'adresse IP privée	Confirmer que PS_DOMAIN dans parameters.php et dans le script user_data est correctement défini sur le DNS de l'ALB.
Erreur 404 sur des ressources	S'assurer que le module mod_rewrite est actif dans Apache, que les droits d'accès sont corrects pour www-data, et que le cache est bien vidé.



# Restauration et Résilience des Données

***L'architecture assure une forte résilience. Même en cas de destruction complète de l'infrastructure (terraform destroy), les données critiques de l'e-commerce sont conservées par les services de stockage et de base de données gérés.***

## Persistance des Fichiers (EFS)

***Tous les fichiers de PrestaShop (thèmes, images téléchargées, modules) sont persistants sur EFS. Ce système de fichiers géré n'est pas affecté par la destruction des instances EC2, assurant la préservation de la couche front-end.***

## Durabilité de la Base de Données (RDS)

***La base de données RDS doit être sauvegardée via des snapshots avant toute suppression majeure. En cas de restauration, il est possible de recréer l'instance RDS à partir du dernier snapshot viable.***

## Processus de Restauration

- ***Exécuter terraform apply pour recréer l'infrastructure (ALB, ASG, RDS, EFS mounts).***
- ***Les nouvelles instances EC2 se reconnectent automatiquement au volume EFS existant et à l'instance RDS.***
- ***PrestaShop redémarre avec l'état exact (fichiers et données) précédant la destruction, grâce à l'automatisation du user\_data.sh.***

# Analyse Détaillée des Coûts Estimés (Mensuels)

L'estimation des coûts est essentielle pour la planification budgétaire. Ce tableau présente une projection mensuelle pour une architecture de taille modérée (deux instances t3.medium en ASG).

Service	Détails de l'Estimation	Coût Estimé Mensuel
Application Load Balancer (ALB)	0,0225 \$/h + LCU (Charge Computing Units) estimé	32,76 \$
Instances EC2 (2 x t3.medium)	0,0416 \$/h × 2 instances (utilisation 24/7)	60,00 \$
Amazon RDS (db.t3.micro)	Multi-AZ, 0,017 \$/h (pour une instance de développement/petite prod)	24,48 \$
Amazon EFS (5 To)	0,03 \$/Go × 5 000 Go (coût de stockage)	150,00 \$
NAT Gateway	0,045 \$/h + 1 To de données traitées	50,00 \$
Coût Total Estimé	Infrastructure de Production Modérée	≈ 317,24 \$

Il est impératif de surveiller régulièrement les coûts AWS, particulièrement pour le stockage EFS et la NAT Gateway (frais de transfert de données), et d'ajuster la taille des instances (t3.medium) selon le volume réel de trafic.

