

# GreenShop DevOps & Cloud Migration

Show Image

## Table des matières

1. [Présentation](#)
  2. [Prérequis](#)
  3. [Installation rapide](#)
  4. [Installation détaillée](#)
  5. [Arborescence du projet](#)
  6. [Administration et utilisation](#)
  7. [Surveillance et sécurité](#)
  8. [Troubleshooting](#)
  9. [Contribuer](#)
  10. [Licence](#)
- 

## Présentation

Ce projet permet de migrer et d'automatiser l'infrastructure de l'application e-commerce **GreenShop** sur AWS, en suivant les meilleures pratiques DevOps et Cloud Native. Une fois déployée, l'infrastructure inclut tous les composants nécessaires pour faire fonctionner l'application de manière scalable, sécurisée et observable.

## Technologies utilisées

- **Infrastructure as Code** : Terraform
- **Configuration Management** : Ansible & SaltStack
- **CI/CD** : Jenkins + Docker
- **Monitoring & Observabilité** : Prometheus, Grafana, Loki
- **Sécurité** : Scan d'images Docker avec Trivy

Le code source de l'application GreenShop (PHP/Apache + MariaDB) est disponible dans [docs/www/greenshop/](#).

---

## Prérequis

- Système d'exploitation : Ubuntu 24.04 LTS à jour
  - Un compte AWS avec les droits nécessaires
  - Un accès git pour cloner le dépôt
- 

## Installation rapide

bash

*# Cloner le dépôt*

```
git clone git@github.com:ttnmeoeayhrnefr/Projet--GreenShop-DevOps-Cloud-Migration-.git
```

*# Se déplacer dans le répertoire du projet*

```
cd Projet--GreenShop-DevOps-Cloud-Migration-
```

*# Installer make si nécessaire*

```
sudo apt update -y && sudo apt install make -y
```

*# Installer les dépendances*

```
make install
```

*# Configurer l'environnement*

```
make configure
```

*# Configurer le token de session AWS (si vous utilisez MFA)*

```
make configure-aws_session_token aws_session_token=VOTRE_TOKEN
```

*# Déployer GreenShop*

```
make greenshop
```

---

## Installation détaillée

### 1. Cloner le dépôt

bash

```
git clone git@github.com:ttnmeoeayhrnefr/Projet--GreenShop-DevOps-Cloud-Migration-.git
```

```
cd Projet--GreenShop-DevOps-Cloud-Migration-
```

### 2. Installer les dépendances

Le Makefile automatise l'installation de tous les outils nécessaires :

bash

```
make install
```

Cette commande installe :

- Terraform (v1.0+)
- AWS CLI v2
- Ansible (v2.9+)
- SaltStack (v3000+)
- Docker & Docker Compose
- kubectl (pour Kubernetes)
- Trivy (scanner de sécurité)

### 3. Configurer l'environnement

```
bash
```

```
make configure
```

Cette étape :

- Configure les credentials AWS
- Initialise Terraform
- Prépare l'environnement Ansible
- Configure Docker

### 4. Configurer l'authentification AWS avec MFA (si nécessaire)

Si vous utilisez l'authentification multi-facteurs (MFA) :

```
bash
```

```
make configure-aws_session_token aws_session_token=VOTRE_TOKEN
```

### 5. Déployer GreenShop

```
bash
```

```
make greenshop
```

Cette commande :

- Déploie l'infrastructure via Terraform
- Configure les serveurs avec Ansible
- Démarre les conteneurs Docker
- Déploie la stack de monitoring

text

```
Projet--GreenShop-DevOps-Cloud-Migration-/  
├─ Makefile                # Automatisation des tâches  
├─ README.md               # Documentation principale  
├─ ansible/  
│   ├─ inventory.yml       # Inventaire des hôtes  
│   ├─ greenshop.yml       # Playbook principal  
│   └─ vault.yml           # Variables chiffrées  
├─ salt/  
├─ terraform/  
├─ docker/  
├─ jenkins/  
├─ prometheus+grafana/  
├─ loki/  
├─ trivy/  
├─ docs/  
│   └─ www/greenshop/  
└─ aws/  
# Ressources AWS supplémentaires
```

---

## Administration et utilisation

### Commandes courantes

bash

*# Démarrage complet*

**make** greenshop

*# Connexion SSH aux serveurs*

**make** ssh-adm1     *# Serveur d'administration*

**make** ssh-db1      *# Base de données*

**make** ssh-web1     *# Serveur web GreenShop*

*# Mise à jour de la configuration*

**make** update

*# Redémarrage des services*

**make** restart-web

**make** restart-db

**make** restart-monitoring

*# Vérification de l'état de l'infrastructure*

**make** status

*# Destruction de l'infrastructure*

**make** destroy

## Accès aux interfaces web

- **Application GreenShop** : `http://<IP_WEB1>`
  - **Jenkins** : `http://<IP_ADM1>:8080`
  - **Grafana** : `http://<IP_ADM1>:3000`
  - **Prometheus** : `http://<IP_ADM1>:9090`
- 

## Surveillance et sécurité

Notre stack de surveillance comprend :

- **Prometheus** : Collecte des métriques système et applicatives
  - Exporters installés : `node_exporter`, `mysql_exporter`, `apache_exporter`
  - Alertmanager configuré pour les notifications
- **Grafana** : Visualisation des données
  - Tableaux de bord pré-configurés pour le système, MySQL et Apache
  - Alertes visuelles et notifications
- **Loki** : Agrégation et analyse des logs
  - Tous les logs centralisés et indexés
  - Requêtes et alertes sur les patterns de logs
- **Trivy** : Analyse de sécurité des conteneurs
  - Intégré dans le pipeline CI/CD
  - Rapports de vulnérabilités automatisés

Pour accéder aux dashboards :

```
bash
```

```
make open-grafana
```

---

## Troubleshooting

### Problèmes courants

- **Erreur d'authentification AWS :**

```
bash
```

```
make configure-aws_session_token aws_session_token=VOTRE_NOUVEAU_TOKEN
```

- **Erreur de déploiement Terraform :**

```
bash
```

```
make terraform-clean
```

```
make terraform-apply
```

- **Problèmes d'accès aux serveurs :**

```
bash
```

```
make fix-ssh
```

- **Génération des logs de diagnostic :**

```
bash
```

```
make diagnostics > diagnostic-$(date +%Y%m%d).log
```

---

## Contribuer

1. Forker le projet
2. Créer une branche : `git checkout -b feature/ma-feature`
3. Commit & push : `git commit -am "Description de la feature" && git push origin feature/ma-feature`
4. Ouvrir une Pull Request

Avant de soumettre votre PR, exécutez les tests :

```
bash
```

```
make test
```

---

## Licence

Ce projet est distribué sous licence MIT. Voir le fichier [LICENSE](#) pour plus d'informations.