

Déploiement BACKEND

Script de Déploiement Backend (Pré-production)

Ce script Bash est conçu pour automatiser le déploiement d'une application backend sur un environnement de pré-production. Il gère la mise à jour du code, la sauvegarde des versions précédentes, la configuration spécifique à l'environnement et le déploiement via Docker Compose.

? Fonctionnalités

- **Sauvegarde Automatique** : Archive la version précédente du déploiement avant toute mise à jour.
- **Clonage Git Sécurisé** : Récupère la dernière version du code depuis un dépôt Git privé via SSH.
- **Gestion des Variables d'Environnement** : Copie le fichier `.env` de la version précédente pour maintenir la configuration spécifique à l'environnement.
- **Déploiement Dockerisé** : Construit et démarre le service Docker Compose spécifié.

?? Prérequis

Avant d'exécuter ce script, assurez-vous que les éléments suivants sont configurés sur le serveur de déploiement :

- **Git** : Installé et configuré.
- **Docker & Docker Compose** : Installés.
- **Clé SSH** : Une clé SSH (`brian_git` dans l'exemple) doit être configurée pour accéder au dépôt Git (`git@github.com:Eiffage-AFC/WegoBackend.git`). Le chemin de la clé est spécifié par la variable `SSH_KEY`.
- **Permissions `sudo`** : L'utilisateur exécutant le script doit avoir les permissions `sudo` pour les opérations de création de répertoire, déplacement de fichiers, clonage Git et commandes Docker.

?? Variables Clés

Modifiez ces variables au début du script pour l'adapter à votre projet :

- `FRONT_REPO` : URL SSH du dépôt Git de votre backend.
- `BRANCH` : Branche Git à déployer (par exemple, `preprod` ou `main`).
- `DEPLOY_DIR` : Chemin absolu où l'application sera déployée sur le serveur.
- `DOCKER_SERVICE` : Nom du service Docker à construire et démarrer (défini dans votre `docker-compose.yml`).
- `PREV_DIR` : Répertoire où les sauvegardes des déploiements précédents seront stockées.
- `SSH_KEY` : Chemin vers la clé SSH privée utilisée pour l'authentification Git.

? Étapes du Déploiement

1. **Préparation du dossier de sauvegarde** : Crée un répertoire pour stocker les anciennes versions du déploiement.
2. **Sauvegarde** : Si une version de l'application est déjà présente dans `DEPLOY_DIR`, elle est déplacée vers `PREV_DIR` avec un horodatage pour backup.
3. **Clonage du dépôt** : Le script clone la branche spécifiée (`preprod`) du dépôt Git vers le `DEPLOY_DIR`. L'authentification SSH est utilisée.
4. **Configuration de l'environnement** : Le fichier `.env` de la version précédente sauvegardée est copié dans le nouveau dossier de déploiement. **(Il est essentiel que le `.env` de la première installation soit présent dans le dossier `prev` pour être copié).**
5. **Build et Déploiement Docker** : Le script navigue vers le `DEPLOY_DIR`, construit l'image Docker du service spécifié et démarre (ou redémarre) le conteneur en mode détaché.

? Utilisation

Pour exécuter ce script, connectez-vous à votre serveur cible via SSH et lancez-le :

```
bash nom_du_script.sh
```

```
#!/bin/bash

# -----
# Déploiement BACKEND Angular (SSH)
# -----

# Variables
FRONT_REPO="git@github.com:Eiffage-AFC/WegoBackend.git"
```

```
BRANCH="preprod"
DEPLOY_DIR="/var/wego/WegoBackend"
DOCKER_SERVICE="backend"
PREV_DIR="/var/wego/prev/backend"
TIMESTAMP=$(date +%d%m%Y)
SSH_KEY="$HOME/.ssh/brian_git"

echo "?? Début du déploiement BACKEND"

# 0?? Préparer le dossier prev pour les backups
sudo mkdir -p "$PREV_DIR"

# 1?? Sauvegarder l'ancienne préprod si elle existe
if [ -d "$DEPLOY_DIR" ]; then
    echo "?? Sauvegarde de l'ancienne préprod dans $PREV_DIR/$TIMESTAMP"
    sudo mv "$DEPLOY_DIR" "$PREV_DIR/$TIMESTAMP/"
fi

# 2?? Cloner la branche preprod depuis Git via SSH
echo "?? Clonage de la branche $BRANCH via SSH"
sudo GIT_SSH_COMMAND="ssh -i $SSH_KEY -o StrictHostKeyChecking=no" git clone -b "$BRANCH"
"$FRONT_REPO" "$DEPLOY_DIR"

# 3?? Configuration Backend pour la production

sudo cp "$PREV_DIR/$TIMESTAMP/.env" "$DEPLOY_DIR/.env"

echo "Fichier .env Sauvegarder"

# 4?? Build Docker et déploiement
echo "?? Build Docker et démarrage du service $DOCKER_SERVICE"
cd "$DEPLOY_DIR"
sudo docker compose build "$DOCKER_SERVICE"
sudo docker compose up -d "$DOCKER_SERVICE"

echo "? Déploiement Backend terminé !"
```

Revision #1

Created 4 December 2025 08:06:10 by kazmierowski

Updated 4 December 2025 08:14:57 by kazmierowski