

BTS SIO — Option SLAM

Procédure Technique

Mise en place de l'environnement de préproduction & Pipeline CI/CD GitHub Actions

Candidat	Cyrille Cour
Établissement	Aristée
Entreprise d'accueil	UPV (Union Patronale du Var)
Date	Juillet 2026

Sommaire

1. Objectif
2. Étape 1 — Infrastructure & Nom de Domaine (OVH / Evolix)
 - 2.1 Création du compte et espace Web
 - 2.2 Configuration DNS (OVH)
3. Étape 2 — Migration de la Base de Données (BigDump)
 - 3.1 Configuration de BigDump
 - 3.2 Transfert et Import
4. Étape 3 — Déploiement des Fichiers Symfony (WinSCP)
 - 4.1 Fichiers sources (Transfert direct Prod → Préprod)
 - 4.2 Gestion du dossier var/ (Cache et Logs)
5. Étape 4 — Configuration de l'Environnement (.env)
 - 5.1 Isolation de la base de données
 - 5.2 Activation du mode Debug (Symfony)
6. Pipeline CI/CD — GitHub Actions
 - 6.1 Repo Git
 - 6.2 Authentification SSH
 - 6.3 Workflow GitHub Actions
7. Résultat & Récapitulatif

1. Objectif

Créer un environnement de préproduction (preprod-papier.upv.org) totalement isolé, à partir d'un clone de l'environnement de production existant sur le serveur Evolix (upv-www02.evolix.net), puis automatiser les déploiements via un pipeline CI/CD GitHub Actions.

2. Étape 1 — Infrastructure & Nom de Domaine (OVH / Evolix)

preprod-papier	preprod-papier.upv.org	8.1	Activé	Alias - Serveur - PHP - Let's Encrypt - Supprimer
----------------	------------------------	-----	--------	---

2.1 Création du compte et espace web

Création effectuée auprès d'Evoadmin pour l'ouverture d'un accès cloisonné et d'un répertoire dédié au nouvel utilisateur preprod-papier. Cela garantit une isolation complète par rapport à la production.

2.2 Configuration DNS (OVH)

Ajout d'un sous-domaine preprod-papier.upv.org pointant vers l'adresse IP du serveur d'hébergement Evolix.

☐ preprod-papier.upv.org.	3600	A	31.170.11.201	(...)
--	------	---	---------------	-------

3. Étape 2 — Migration de la Base de Données (BigDump)

En raison des restrictions de sécurité SSH bloquant les imports massifs en ligne de commande, l'outil BigDump a été utilisé pour injecter la base de données via HTTP sans coupure de session.

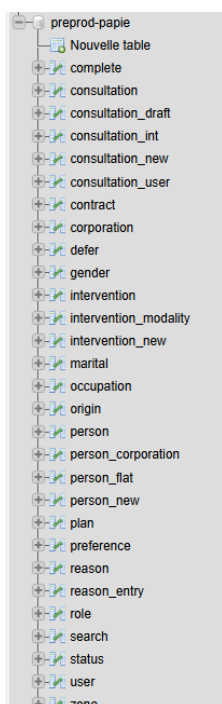
3.1 Configuration de BigDump

Modification du fichier bigdump.php avec les nouveaux identifiants de la base de données de préproduction (preprod-papier). Pour éviter le dépassement du max_execution_time de PHP, le script a été bridé pour injecter le fichier SQL par micro-paquets :

```
$linespersession = 500;  
$delaypersession = 100; // pause de 100ms entre chaque session
```

3.2 Transfert et Import

- Envoi du script bigdump.php et du dump (sf_papier.sql, ~21 Mo) via WinSCP dans le dossier web public
- Lancement de l'import depuis le navigateur
- Validation de l'intégrité des tables dans PhpMyAdmin
- Suppression immédiate de bigdump.php du serveur après finalisation



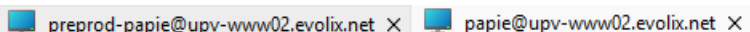
4. Étape 3 — Déploiement des Fichiers Symfony (WinSCP)

Pour contourner les restrictions PHP de l'hébergeur (putenv() désactivé, bloquant Composer global), une stratégie de duplication hybride a été appliquée via WinSCP, en copiant directement depuis la production.

4.1 Fichiers sources (Transfert direct Prod → Préprod)

Duplication directe des répertoires de code et fichiers de configuration essentiels :

- Répertoires : assets/, bin/, config/, migrations/, public/, src/, templates/, translations/, vendor/
- Fichiers : .htaccess, composer.json, composer.lock, package.json, yarn.lock, symfony.lock, webpack.config.js, routes.yaml



4.2 Gestion du dossier var/ (Cache et Logs)

Le dossier var/ de la production a été exclu du transfert pour éviter de corrompre l'environnement avec du cache figé de production. Un répertoire var/ vide a été créé manuellement, avec les droits d'écriture nécessaires à Symfony :

```
chmod -R 775 var
```

5. Étape 4 — Configuration de l'Environnement (.env)

Modification du fichier .env sur le serveur de préproduction pour basculer l'application dans son nouvel état.

5.1 Isolation de la base de données

Modification de la variable DATABASE_URL pour pointer sur la base clonée preprod-papie, sans impact sur la production :

```
DATABASE_URL="mysql://preprod-papie:motdepasse@127.0.0.1:3306/preprod-papie?serverVersion=mariadb-10.3.39&charset=utf8mb4"
```

5.2 Activation du mode Debug (Symfony)

Passage de l'environnement production à développement pour activer la Web Debug Toolbar et les logs détaillés :

```
APP_ENV=dev
```

```
1  ###> symfony/framework-bundle ###
2  APP_ENV=dev
3  APP_SECRET=censure
4  ###< symfony/framework-bundle ###
5
6  ###> doctrine/doctrine-bundle ###
7  # Format described at https://www.doctrine-project.org/projects/doctrine-dbal/en/latest/reference/configuration.html#connect
8  # IMPORTANT: You MUST configure your server version, either here or in config/packages/doctrine.yaml
9  #
10 # DATABASE_URL="mysql://app:!ChangeMe!@127.0.0.1:3306/db_name?serverVersion=8"
11 DATABASE_URL="mysql://preprod-papie:sievriwirAvCektep8@127.0.0.1:3306/preprod-papie?serverVersion=mariadb-10.3.39&charset=utf8mb4"
12 ###< doctrine/doctrine-bundle ###
13
14
15 ###> symfony/mailer ###
16
17 #MAILER_DSN=smtp://upvsocial_mailcatcher_1:1025
18 ###< symfony/mailer ###
19
20 ###> symfony/mailjet-mailer ###
21 # MAILER_DSN=mailjet+api://824c11fc8d70ddb3e64eb14ca339290b:0a488bc53072cfdc807254a31801fc6e@api.mailjet.com
22 MAILER_DSN=mailjet+smtp://824c11fc8d70ddb3e64eb14ca339290b:0a488bc53072cfdc807254a31801fc6e@in-v3.mailjet.com
23 ###< symfony/mailjet-mailer ###
24 SYNC_TOKEN=censure
```

❏ **Note :** Le fichier .env ne doit jamais être versionné avec des credentials réels. Sur la préprod, un .env.local peut surcharger les valeurs sans être commité dans Git.



6. Pipeline CI/CD — GitHub Actions

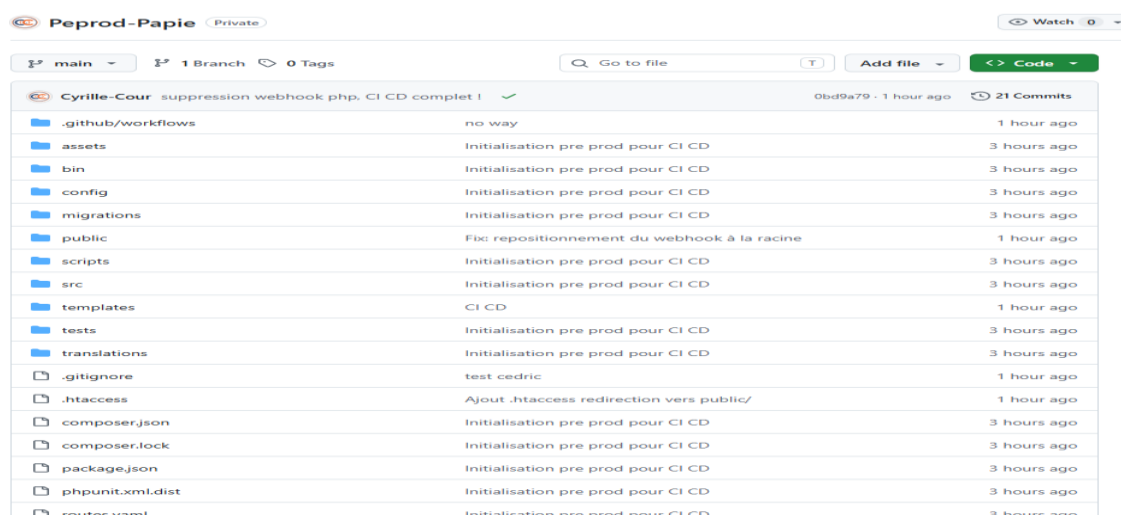
Pour automatiser les déploiements et éviter les transferts manuels via WinSCP, un pipeline CI/CD a été mis en place avec GitHub Actions. Chaque push sur la branche main déclenche automatiquement le déploiement sur le serveur de préproduction.

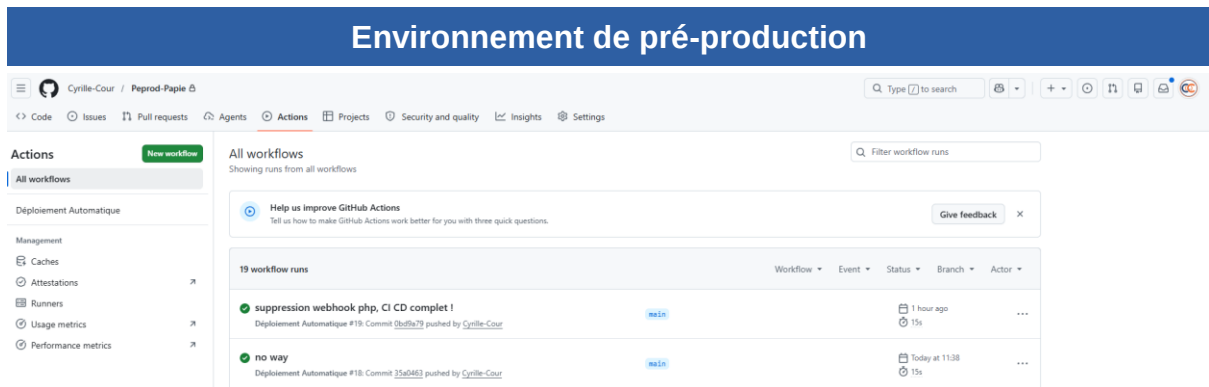
6.1 Repo Git

- Création du repo GitHub Preprod-Papie et push du code source Symfony
- Ajout du .htaccess à la racine du projet pour rediriger vers public/ (le DocumentRoot Apache pointant sur www/) :

```
RewriteEngine On
RewriteRule ^(.*)$ public/$1 [L]
```

- Ajout de index.php et deploy-webhook.php au .gitignore





6.2 Authentification SSH

Le déploiement s'effectue via SSH. Une paire de clés ED25519 a été générée pour permettre à GitHub Actions de se connecter au serveur sans mot de passe :

```
ssh-keygen -t ed25519 -C "github-actions-preprod" \
-f "$env:USERPROFILE\.ssh\github_preprod"
```

- Clé publique ajoutée dans /home/preprod-papier/.ssh/authorized_keys sur le serveur
- Clé privée ajoutée dans GitHub → Settings → Secrets and variables → Actions → SSH_PRIVATE_KEY

Repository secrets

[New repository secret](#)

Name 	Last updated
 SERVER_HOST	2 hours ago  
 SERVER_USER	2 hours ago  
 SSH_PRIVATE_KEY	2 hours ago  

6.3 Workflow GitHub Actions

Le fichier .github/workflows/deploy.yml déclenche le déploiement à chaque push sur main. Points clés : port SSH 2222 (port 22 bloqué par le firewall Evolix), git reset --hard pour forcer la synchronisation, et vidage du cache Symfony.

```
name: Déploiement Automatique

on:
  push:
    branches: [main]

jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - name: Déploiement SSH
        uses: appleboy/ssh-action@v1.0.0
        with:
          host: preprod-papie.upv.org
          username: preprod-papie
          key: ${ secrets.SSH_PRIVATE_KEY }
          port: 2222
          script: |
            cd /home/preprod-papie/www
            git fetch --all
            git reset --hard origin/main
            rm -rf var/cache/*
```

❏ **Note :** exec() étant désactivé sur le serveur Evoadmin pour des raisons de sécurité, l'approche webhook PHP a été abandonnée au profit de la connexion SSH directe depuis GitHub Actions.

7. Résultat & Récapitulatif

Étape	Action	Détail
1. Repo Git	Création du repo GitHub Preprod-Papie et push du code source Symfony	Ajout du .htaccess racine pour redirection vers public/
2. Clé SSH	Génération d'une paire de clés ED25519 sur le poste développeur	Clé publique → authorized_keys serveur / Clé privée → GitHub Secret SSH_PRIVATE_KEY
3. GitHub Actions	Création du fichier .github/workflows/deploy.yml	Déploiement SSH automatique sur push main (port 2222)

□ **Note :** Toute modification du code est désormais déployée automatiquement en préproduction après un simple git push. Un collaborateur peut cloner le repo, coder dans VSCode, pousser sur main et vérifier directement sur preprod-papie.upv.org — sans configuration locale requise.