# 

# Procédure de déploiement - Application TaskFlow

## 1. Informations générales

**Application :** TaskFlow  
**Version Symfony :** 7.3.4  
**Version PHP :** 8.3.14  
**Base de données :** MySQL 8.0.32  
**Gestionnaire d'assets :** AssetMapper (pas de build requis)  
**Date de dernière mise à jour :** 17 octobre 2025

## 2. Prérequis techniques

### 2.1 Serveur de production

* **PHP :** Version 8.3.14 minimum
* **Extensions PHP requises :**
  + pdo\_mysql
  + intl
  + opcache
  + mbstring
  + xml
  + ctype
  + iconv
  + curl
  + zip
  + gd (si manipulation d'images)
* **Base de données :** MySQL 8.0.32 ou supérieur / MariaDB 10.11.2 ou supérieur
* **Serveur web :** Apache 2.4+ ou Nginx 1.18+
* **Composer :** Version 2.x
* **Git :** Pour récupérer le code source

### 2.2 Configuration Apache

Activer les modules suivants :

mod\_rewrite  
mod\_headers  
mod\_ssl (pour HTTPS)

### 2.3 Permissions système

L'utilisateur du serveur web (www-data, apache, nginx) doit avoir les droits d'écriture sur :

* /var/cache/
* /var/log/
* /var/sessions/ (si utilisation de sessions fichiers)

## 3. Architecture de déploiement

### 3.1 Environnements

| Environnement | URL | Base de données | Usage |
| --- | --- | --- | --- |
| Développement | <http://127.0.0.1:8000> | taskflow\_dev | Tests locaux |
| Staging | <https://staging.taskflow.app> | taskflow\_staging | Tests pré-production |
| Production | <https://taskflow.app> | taskflow\_prod | Application live |

### 3.2 Structure des répertoires sur le serveur

/var/www/taskflow/  
├── current/ # Version actuellement déployée  
│ ├── public/ # Document root  
│ ├── src/  
│ ├── config/  
│ ├── var/  
│ ├── vendor/  
│ └── .env.local  
├── releases/ # Historique des versions  
│ ├── 2025-10-15\_143022/  
│ ├── 2025-10-16\_091544/  
│ └── 2025-10-17\_104523/  
└── shared/ # Fichiers partagés entre versions  
 ├── var/log/  
 └── .env.local

## 4. Configuration des variables d'environnement

### 4.1 Fichier .env.local en production

Créer le fichier /var/www/taskflow/current/.env.local avec :

APP\_ENV=prod  
APP\_SECRET=VOTRE\_SECRET\_GENERE\_64\_CARACTERES  
APP\_URL=https://taskflow.app  
  
DATABASE\_URL="mysql://user\_prod:password\_secure@127.0.0.1:3306/taskflow\_prod?serverVersion=8.0.32&charset=utf8mb4"  
  
MESSENGER\_TRANSPORT\_DSN=doctrine://default?auto\_setup=0  
  
MAILER\_DSN=smtp://smtp.example.com:587?encryption=tls&auth\_mode=login&username=VOTRE\_EMAIL&password=VOTRE\_PASSWORD  
MAILER\_SENDER\_EMAIL=noreply@taskflow.app  
MAILER\_SENDER\_NAME=TaskFlow

**Important :** Ne jamais commiter ce fichier dans Git.

### 4.2 Génération du APP\_SECRET

php -r "echo bin2hex(random\_bytes(32));"

## 5. Procédure de déploiement manuel

### 5.1 Première installation

#### Étape 1 : Cloner le dépôt

cd /var/www/taskflow/releases  
git clone https://github.com/votre-compte/taskflow.git 2025-10-17\_104523  
cd 2025-10-17\_104523

#### Étape 2 : Installer les dépendances

composer install --no-dev --optimize-autoloader --no-interaction

#### Étape 3 : Configurer l'environnement

cp /var/www/taskflow/shared/.env.local .env.local

#### Étape 4 : Créer la base de données

php bin/console doctrine:database:create --if-not-exists --env=prod

#### Étape 5 : Exécuter les migrations

php bin/console doctrine:migrations:migrate --no-interaction --env=prod

#### Étape 6 : Installer les assets (AssetMapper)

php bin/console asset-map:compile --env=prod

#### Étape 7 : Nettoyer et réchauffer le cache

php bin/console cache:clear --env=prod  
php bin/console cache:warmup --env=prod

#### Étape 8 : Configurer les permissions

chown -R www-data:www-data var/  
chmod -R 775 var/

#### Étape 9 : Créer le lien symbolique

ln -sfn /var/www/taskflow/releases/2025-10-17\_104523 /var/www/taskflow/current

#### Étape 10 : Redémarrer PHP-FPM

sudo systemctl reload php8.3-fpm

### 5.2 Mise à jour ultérieure

#### Étape 1 : Sauvegarder la base de données

mysqldump -u root -p taskflow\_prod > /var/backups/taskflow\_$(date +%Y%m%d\_%H%M%S).sql

#### Étape 2 : Créer un nouveau dossier de release

cd /var/www/taskflow/releases  
git clone https://github.com/votre-compte/taskflow.git $(date +%Y-%m-%d\_%H%M%S)  
cd $(date +%Y-%m-%d\_%H%M%S)  
git checkout tags/v1.2.0 # ou la version souhaitée

#### Étape 3 : Répéter les étapes 2 à 9 de la première installation

#### Étape 4 : Mettre en maintenance (optionnel)

touch /var/www/taskflow/current/public/maintenance.html

#### Étape 5 : Basculer vers la nouvelle version

ln -sfn /var/www/taskflow/releases/NOUVELLE\_VERSION /var/www/taskflow/current

#### Étape 6 : Redémarrer les services

sudo systemctl reload php8.3-fpm  
sudo systemctl restart apache2 # ou nginx

#### Étape 7 : Retirer la maintenance

rm /var/www/taskflow/current/public/maintenance.html

## 6. Vérifications post-déploiement

### 6.1 Checklist de validation

* L'application est accessible via l'URL de production
* La page d'accueil se charge correctement
* Les assets (CSS, JS, images) sont bien chargés
* La connexion à la base de données fonctionne
* Les utilisateurs peuvent se connecter
* Les emails sont envoyés correctement
* Les logs ne contiennent pas d'erreurs critiques
* Le certificat SSL est valide
* Les performances sont acceptables (temps de chargement < 2s)

### 6.2 Commandes de vérification

# Vérifier la version Symfony  
php bin/console --version  
  
# Vérifier les exigences système  
php bin/console about  
  
# Tester la connexion base de données  
php bin/console doctrine:query:sql "SELECT 1"  
  
# Vérifier les routes  
php bin/console debug:router  
  
# Consulter les logs  
tail -f var/log/prod.log

## 7. Monitoring et logs

### 7.1 Fichiers de logs

* **Application :** /var/www/taskflow/current/var/log/prod.log
* **Apache :** /var/log/apache2/taskflow\_error.log
* **PHP-FPM :** /var/log/php8.3-fpm.log
* **MySQL :** /var/log/mysql/error.log

### 7.2 Rotation des logs

Configuration dans /etc/logrotate.d/taskflow :

/var/www/taskflow/current/var/log/\*.log {  
 daily  
 missingok  
 rotate 14  
 compress  
 delaycompress  
 notifempty  
 create 0640 www-data www-data  
 sharedscripts  
}

## 8. Gestion des erreurs courantes

### 8.1 Erreur 500 après déploiement

**Solution :**

php bin/console cache:clear --env=prod --no-warmup  
php bin/console cache:warmup --env=prod  
chown -R www-data:www-data var/

### 8.2 Assets non chargés

**Solution :**

php bin/console asset-map:compile --env=prod  
php bin/console cache:clear --env=prod

### 8.3 Erreur de connexion base de données

**Vérifier :**

* Les identifiants dans .env.local
* Que l'utilisateur MySQL a les droits nécessaires
* Que MySQL est démarré : sudo systemctl status mysql

### 8.4 Performances dégradées

**Actions :**

# Activer OPcache  
php -i | grep opcache  
  
# Optimiser Composer  
composer dump-autoload --optimize --no-dev  
  
# Compiler les assets  
php bin/console asset-map:compile --env=prod

## 9. Sécurité

### 9.1 Permissions fichiers

# Propriétaire  
chown -R www-data:www-data /var/www/taskflow/current  
  
# Droits standards  
find /var/www/taskflow/current -type d -exec chmod 755 {} \;  
find /var/www/taskflow/current -type f -exec chmod 644 {} \;  
  
# Droits écriture pour var/  
chmod -R 775 /var/www/taskflow/current/var/

### 9.2 Fichiers sensibles

Vérifier que ces fichiers ne sont pas accessibles via le web :

* .env.local
* composer.json
* composer.lock
* symfony.lock

Configuration Apache dans .htaccess ou vhost :

<FilesMatch "^\.env">  
 Require all denied  
</FilesMatch>

## 10. Contacts et support

**Responsable technique :** [NOM]  
**Email :** [EMAIL]  
**Téléphone :** [TELEPHONE]  
**Heures d'intervention :** [HORAIRES]

**Hébergeur :** [NOM HEBERGEUR]  
**Support hébergeur :** [CONTACT]

## 11. Historique des versions

| Version | Date | Déployé par | Changements majeurs |
| --- | --- | --- | --- |
| 1.0.0 | 2025-10-17 | [NOM] | Déploiement initial |
|  |  |  |  |