

Procédure de rollback - Application TaskFlow

1. Introduction

Ce document décrit les procédures de retour arrière en cas de problème après un déploiement de l'application TaskFlow. Le rollback permet de revenir rapidement à une version stable et fonctionnelle de l'application.

Date de dernière mise à jour : 17 octobre 2025

2. Quand effectuer un rollback

2.1 Critères déclencheurs

Un rollback doit être envisagé dans les situations suivantes :

- Erreurs critiques empêchant l'accès à l'application
- Perte de données utilisateur
- Problèmes de performance majeurs (temps de réponse > 10 secondes)
- Erreurs de migration de base de données irréversibles
- Failles de sécurité découvertes après déploiement
- Incompatibilité avec l'environnement de production
- Dysfonctionnement des fonctionnalités critiques métier

2.2 Processus de décision

Délai de décision : Maximum 30 minutes après détection du problème

Responsables autorisés à déclencher un rollback :

- Responsable technique
- Chef de projet
- Administrateur système
- Développeur senior de garde

Procédure de validation :

1. Identification du problème
2. Évaluation de l'impact
3. Vérification qu'un correctif rapide n'est pas possible
4. Décision de rollback
5. Communication aux parties prenantes

3. Types de rollback

3.1 Rollback applicatif (sans base de données)

Cas d'usage :

- Erreurs dans le code PHP
- Problèmes de configuration
- Dysfonctionnement des assets
- Erreurs de templates Twig

Temps estimé : 5 à 10 minutes

3.2 Rollback applicatif + base de données

Cas d'usage :

- Migrations de base de données défectueuses
- Corruption de données
- Changements de schéma problématiques

Temps estimé : 15 à 30 minutes

3.3 Rollback complet (système)

Cas d'usage :

- Problèmes de configuration serveur
- Incompatibilités PHP/extensions
- Problèmes de permissions critiques

Temps estimé : 30 à 60 minutes

4. Procédure de rollback applicatif simple

4.1 Prérequis

- Accès SSH au serveur de production
- Droits sudo ou accès root
- Vérifier qu'une version précédente existe dans `/var/www/taskflow/releases/`

4.2 Étapes manuelles

Étape 1 : Se connecter au serveur

```
ssh utilisateur@serveur-production.taskflow.app
```

Étape 2 : Vérifier les versions disponibles

```
cd /var/www/taskflow/releases  
ls -lt
```

Exemple de sortie :

```
2025-10-17_104523 # Version actuelle (défectueuse)  
2025-10-16_153012 # Version précédente (stable)  
2025-10-15_091844
```

Étape 3 : Identifier la version actuelle

```
readlink /var/www/taskflow/current
```

Étape 4 : Basculer vers la version précédente

```
ln -sfn /var/www/taskflow/releases/2025-10-16_153012 /var/www/taskflow/current
```

Étape 5 : Nettoyer le cache

```
cd /var/www/taskflow/current  
php bin/console cache:clear --env=prod --no-warmup  
php bin/console cache:warmup --env=prod
```

Étape 6 : Redémarrer les services

```
sudo systemctl reload php8.3-fpm  
sudo systemctl reload apache2
```

Étape 7 : Vérifier le fonctionnement

```
curl -I https://taskflow.app
```

Vérifier que le code de réponse est 200 ou 302.

4.3 Utilisation du script automatisé

```
cd /var/www/taskflow  
./deploy.sh rollback
```

Le script effectue automatiquement toutes les étapes ci-dessus.

5. Procédure de rollback avec restauration de base de données

5.1 Avertissement

La restauration de la base de données entraîne la **perte de toutes les données** créées ou modifiées depuis le déploiement défectueux. Cette opération doit être validée par au moins deux personnes responsables.

5.2 Étapes détaillées

Étape 1 : Mettre l'application en maintenance

```
cd /var/www/taskflow/current/public  
echo "Application en maintenance. Retour prévu dans 15 minutes." > maintenance.html
```

Configurer Apache/Nginx pour afficher cette page.

Étape 2 : Lister les sauvegardes disponibles

```
ls -lh /var/backups/taskflow/
```

Exemple de sortie :

```
db_backup_20251017_103000.sql.gz  # Avant déploiement défectueux  
db_backup_20251016_150000.sql.gz
```

Étape 3 : Identifier la bonne sauvegarde

La sauvegarde à restaurer doit être celle créée **juste avant** le déploiement problématique.

Étape 4 : Décompresser la sauvegarde

```
cd /var/backups/taskflow  
gunzip -c db_backup_20251017_103000.sql.gz > restore_temp.sql
```

Étape 5 : Arrêter les processus accédant à la base

```
sudo systemctl stop php8.3-fpm
```

Étape 6 : Restaurer la base de données

```
mysql -u user_prod -p taskflow_prod < restore_temp.sql
```

Attention : Toutes les données actuelles seront écrasées.

Étape 7 : Vérifier l'intégrité de la base

```
mysql -u user_prod -p taskflow_prod -e "SHOW TABLES;"  
mysql -u user_prod -p taskflow_prod -e "SELECT COUNT(*) FROM users;"
```

Étape 8 : Effectuer le rollback applicatif

Suivre la procédure de la section 4.

Étape 9 : Redémarrer les services

```
sudo systemctl start php8.3-fpm  
sudo systemctl reload apache2
```

Étape 10 : Retirer la maintenance

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

Étape 11 : Nettoyer les fichiers temporaires

```
rm /var/backups/taskflow/restore_temp.sql
```

6. Vérifications post-rollback

6.1 Checklist de validation

Après un rollback, vérifier systématiquement :

- ☐ L'application est accessible via l'URL principale
- ☐ La page d'accueil se charge sans erreur
- ☐ Les utilisateurs peuvent se connecter
- ☐ Les fonctionnalités principales sont opérationnelles
- ☐ Les logs ne contiennent pas d'erreurs critiques
- ☐ La base de données répond correctement
- ☐ Les emails sont envoyés (tester avec un email de test)
- ☐ Les sessions utilisateurs fonctionnent
- ☐ Les assets (CSS, JS, images) se chargent correctement
- ☐ Les performances sont normales (temps de réponse < 2s)

6.2 Tests fonctionnels rapides

Test de connexion base de données

```
cd /var/www/taskflow/current  
php bin/console doctrine:query:sql "SELECT COUNT(*) FROM users"
```

Test de cache

```
php bin/console cache:pool:clear cache.global_clearer
```

Vérifier Les routes

```
php bin/console debug:router
```

Consulter Les derniers Logs

```
tail -n 50 var/log/prod.log
```

6.3 Surveillance post-rollback

Maintenir une surveillance accrue pendant **2 heures** après le rollback :

- Consulter les logs toutes les 15 minutes
- Surveiller les métriques de performance
- Vérifier les retours utilisateurs
- Monitorer l'utilisation des ressources serveur

7. Communication

7.1 Notification interne

Modèle d'email à envoyer à l'équipe :

Objet : [URGENT] Rollback effectué - Application TaskFlow

Bonjour,

Un rollback a été effectué sur l'application TaskFlow en production.

Détails :

- Date et heure : 17/10/2025 à 14h30
- Version déployée (défectueuse) : v1.2.0 (release 2025-10-17_104523)
- Version restaurée : v1.1.8 (release 2025-10-16_153012)
- Raison : Erreurs critiques dans le module de facturation
- Impact : Aucune perte de données (rollback applicatif uniquement)
- Durée d'indisponibilité : 8 minutes

Statut actuel : Application opérationnelle

Actions à suivre :

- Investigation de la cause racine
- Correction du bug identifié
- Tests approfondis avant nouveau déploiement

Cordialement,

[NOM DU RESPONSABLE]

7.2 Communication utilisateurs

Si l'incident a impacté les utilisateurs :

Objet : Incident technique résolu - TaskFlow

Chers utilisateurs,

Nous avons rencontré un problème technique temporaire sur TaskFlow entre 14h22 et 14h30 aujourd'hui.

Le service est maintenant pleinement opérationnel.

Aucune donnée n'a été perdue. Si vous constatez un comportement anormal, n'hésitez pas à nous contacter.

Nous nous excusons pour la gêne occasionnée.

L'équipe TaskFlow

8. Post-mortem

8.1 Analyse obligatoire

Après chaque rollback, une réunion post-mortem doit avoir lieu dans les **48 heures**.

Participants requis :

- Responsable technique
- Développeur ayant effectué le déploiement
- Testeur QA
- Chef de projet

8.2 Points à analyser

1. Cause racine du problème

- Qu'est-ce qui a mal fonctionné ?
- Pourquoi le problème n'a-t-il pas été détecté en pré-production ?

2. Processus de détection

- Combien de temps pour détecter le problème ?
- Comment a-t-il été détecté ?
- Les alertes ont-elles fonctionné ?

3. Processus de rollback

- Le rollback s'est-il déroulé comme prévu ?
- Quelles difficultés ont été rencontrées ?
- Le temps de résolution était-il acceptable ?

4. Impact

- Nombre d'utilisateurs affectés
- Durée de l'incident
- Perte de données éventuelle
- Impact métier

5. Actions correctives

- Tests supplémentaires à mettre en place
- Améliorations du processus de déploiement
- Formation nécessaire

- Outils de monitoring à ajouter

8.3 Template de rapport post-mortem

Post-mortem : Rollback du [DATE]

Résumé

[Description courte du problème et de la résolution]

Chronologie

- HH:MM - Déploiement de la version X
- HH:MM - Détection du problème
- HH:MM - Décision de rollback
- HH:MM - Début du rollback
- HH:MM - Fin du rollback
- HH:MM - Vérification et validation

Cause racine

[Description détaillée]

Impact

- Utilisateurs affectés : X
- Durée : X minutes
- Perte de données : Oui/Non
- Services impactés : [Liste]

Ce qui a bien fonctionné

- [Point 1]
- [Point 2]

Ce qui doit être amélioré

- [Point 1]
- [Point 2]

Actions correctives

Action	Responsable	Échéance	Statut
[Action 1]	[Nom]	[Date]	En cours
[Action 2]	[Nom]	[Date]	À faire

Leçons apprises

[Résumé des apprentissages]

9. Prévention des rollbacks

9.1 Bonnes pratiques de déploiement

Pour réduire la nécessité de rollbacks :

1. Tests exhaustifs en pré-production

- Tests unitaires (couverture > 80%)
- Tests d'intégration
- Tests fonctionnels
- Tests de performance

2. Environnement de staging identique à la production

- Même version PHP
- Même configuration serveur
- Données de test représentatives

3. Déploiements progressifs

- Déployer d'abord sur un serveur de test
- Valider pendant 24h minimum
- Déployer en production aux heures creuses

4. Monitoring et alertes

- Surveillance des erreurs 500
- Alertes sur les performances dégradées
- Monitoring des logs en temps réel
- Surveillance de l'utilisation mémoire/CPU

5. Code review systématique

- Revue de code par un autre développeur
- Validation par le tech lead
- Vérification des migrations de base de données

6. Sauvegardes automatiques

- Sauvegarde base de données avant chaque déploiement
- Conservation de 30 jours de sauvegardes
- Vérification régulière de la restaurabilité

9.2 Checklist pré-déploiement

Avant chaque déploiement, vérifier :

- ☐ Les tests automatisés sont tous au vert
- ☐ Le code a été revu et approuvé
- ☐ La documentation est à jour
- ☐ Les migrations ont été testées
- ☐ Une sauvegarde récente existe
- ☐ L'équipe est disponible pour le support
- ☐ Le déploiement est planifié en heures creuses
- ☐ Les utilisateurs ont été prévenus si nécessaire
- ☐ Le plan de rollback est prêt

10. Contacts d'urgence

10.1 Équipe technique

Rôle	Nom	Téléphone	Email
Responsable technique	[NOM]	[TEL]	[EMAIL]
Développeur senior	[NOM]	[TEL]	[EMAIL]
Administrateur système	[NOM]	[TEL]	[EMAIL]
DBA	[NOM]	[TEL]	[EMAIL]

10.2 Astreintes

Horaires d'astreinte : 24h/7j

Délai d'intervention :

- Critique : 15 minutes
- Urgent : 1 heure
- Normal : 4 heures

10.3 Support hébergeur

Hébergeur : [NOM]

Support technique : [TELEPHONE]

Email : [EMAIL]

Portail client : [URL]

11. Annexes

11.1 Commandes utiles

Lister toutes les releases disponibles avec leur date
`ls -lt /var/www/taskflow/releases/`

Voir quelle version est actuellement déployée
`readlink /var/www/taskflow/current`

Tester la connexion à la base de données
`php bin/console doctrine:query:sql "SELECT 1"`

Vérifier les logs en temps réel
`tail -f /var/www/taskflow/current/var/log/prod.log`

Vérifier l'état des services
`systemctl status php8.3-fpm`
`systemctl status apache2`
`systemctl status mysql`

Vérifier l'utilisation des ressources
`htop`
`df -h`
`free -h`

11.2 Numéros de version et dates de déploiement

Version	Date déploiement	Date rollback	Raison rollback
v1.0.0	2025-10-01	-	-
v1.1.0	2025-10-08	-	-
v1.1.8	2025-10-16	-	-
v1.2.0	2025-10-17	2025-10-17	Erreur module facturation

Prochaine révision prévue : Janvier 2026