Engineer

Retour arrière de déploiement

remettre en service un déploiement antérieur après un échec

Ce que c’est

Un retour arrière de déploiement remet en service une version ou une configuration déployée précédemment après un changement problématique. Ce que l’action fait dépend de l’outil de déploiement. Une plateforme peut renvoyer les requêtes vers une version encore active. Un orchestrateur peut créer de nouvelles instances avec une révision conservée. Dans Kubernetes, le retour arrière d’un Deployment rétablit le modèle de Pod d’une ancienne révision. Les dépendances, dont la base de données, gardent leur propre état.

Annuler un commit dans le dépôt crée une nouvelle révision qui inverse les effets d’une modification du code. Il faut ensuite construire et déployer cette révision pour qu’elle arrive dans le service en cours d’exécution. Un retour arrière du déploiement peut s’appuyer sur un paquet déjà disponible. Après l’intervention, l’équipe doit mettre en accord le résultat voulu et la version déclarée dans sa configuration afin d’éviter une remise en service accidentelle de la version en cause.

L’ancienne version du programme peut trouver des lignes, des fichiers et des événements que la nouvelle a écrits. Si elle peut les traiter, l’équipe peut conserver les opérations récentes des utilisateurs. Restaurer une base de données à une heure antérieure constitue une opération distincte. Définissez le point de récupération, ce qu’il advient des données enregistrées depuis et les preuves attendues. Ces décisions doivent être explicites avant une intervention.

Utilisez le concept pour prendre une décision

Utilisez « Qu'est-ce qu'un retour arrière de déploiement ? » pour éclairer une décision réelle. Nous pouvons vous aider à relier ce concept au système que vous construisez ou achetez. Des changements testés, publiés et réversibles, pour livrer aux clients un mardi après-midi plutôt qu'un vendredi soir tard.

Découvrir nos prestations Contactez-nous →

Vérifiez les hypothèses derrière le terme

Appuyez-vous sur l'explication et les réalisations liées pour formuler les questions de votre projet. Nous pouvons examiner les hypothèses avec vous avant le choix d'un outil. Déploiement SaaS GIS automatisé : GitHub Actions — déploiement, données et reporting automatisés

Déploiement SaaS GIS automatisé : GitHub Actions Contactez-nous →

Pourquoi c’est important

Lorsqu’une erreur apparaît après un déploiement, l’équipe doit chercher l’action qui rendra rapidement le service utile. Un paquet antérieur disponible et une bascule de trafic déjà essayée font du retour arrière une possibilité. Elle peut prendre moins de temps que la préparation d’un correctif urgent. Le changement de version peut limiter les nouveaux incidents. Les opérations déjà touchées peuvent encore demander une réparation des données ou une intervention auprès des utilisateurs.

La récupération se juge sur le service rendu. Une instance peut démarrer correctement alors que le paiement échoue encore ou qu’une file d’attente grossit. Vérifiez l’identifiant de la version qui traite les requêtes et essayez une opération représentative avec un compte de test. Contrôlez les tâches de fond associées et laissez passer assez de temps pour observer le type de trafic qui avait déclenché le problème.

La capacité de revenir dépend aussi de la conservation des ressources. Une révision peut apparaître dans l’orchestrateur alors que l’image correspondante a été supprimée du registre. Il peut manquer un certificat ou un accès à la configuration. Conservez les éléments nécessaires pendant la période de récupération prévue. Exercez le passage à une version antérieure avec les droits et les outils dont la personne d’astreinte disposera réellement.

À quoi cela ressemble en pratique

L’étude de cas sur les déploiements de plateforme dans Azure décrit un parcours défini entre développement, préproduction et production. Cela fournit un cadre pour préparer une récupération, car les membres de l’équipe savent quels environnements et quelles versions ils manipulent. La fiche présente le parcours de déploiement. L’exemple qui suit illustre une répétition possible du retour arrière dans un autre scénario.

Une version B d’un service de commande échoue lorsqu’un client choisit une adresse enregistrée. La version A fonctionnait auparavant et son paquet et sa configuration existent encore. Vérifiez d’abord si la version B a écrit des données

que la version A ne peut plus interpréter. Si les deux savent travailler avec les données actuelles, renvoyez le trafic vers A et passez une commande de contrôle. Confirmez que la version attendue a reçu la requête. Regardez aussi les erreurs traitées dans la file de travail. Vérifiez le sort d'une commande déjà en cours au moment de la bascule. Selon le système, elle peut rester en attente, être terminée correctement ou demander un traitement explicite.

Si B a supprimé une colonne requise par A, le retour immédiat peut faire apparaître d'autres pannes. Une solution possible consiste à suspendre les écritures concernées et à déployer un correctif compatible avec le schéma présent. Pour une modification prévue, ajoutez d'abord les nouvelles structures. Faites migrer les applications qui lisent et écrivent par étapes et différez la suppression de ce que l'ancienne version attend. Vérifiez pendant la transition que les deux versions restent opérationnelles dans le même état de données.

Où les problèmes surviennent

Le succès d'une commande peut annoncer que le modèle a été accepté sans confirmer que les instances répondent ou que le trafic les atteint. Observez la version qui traite vraiment des requêtes. Appliquez les conditions de réussite convenues pendant la répétition et surveillez le comportement qui a motivé l'intervention. Si la panne est encore visible, reprenez le diagnostic.

Un interrupteur de fonctionnalité, une règle de routage ou une tâche planifiée peut agir indépendamment du paquet déployé. Un processus peut continuer à écrire dans le nouveau format même après le remplacement de l'application principale. Recensez ces contrôles à l'avance. Définissez qui les ajuste et dans quel ordre, et ce que feront les clients qui continuent à envoyer les nouvelles requêtes pendant la transition.

Un déploiement automatisé peut réactiver la version problématique après l'intervention si elle figure encore dans la configuration déclarée. Coordonnez le changement de l'état voulu avec celui du paquet et du trafic. Notez les versions, les heures et les effets observés pour les utilisateurs. Déployez la correction par le circuit habituel une fois les vérifications terminées.

Pour aller plus loin

- Kubernetes sur les Deployments et les retours arrière décrit les révisions et le rôle du modèle de Pod.

- DORA sur les mesures de livraison présente les mesures de débit et de récupération.
- PostgreSQL sur la modification des tables documente les opérations sur le schéma.
- Helm rollback explique le choix d'une révision antérieure.

Glossaire

Choisissez selon vos contraintes réelles

Présentez le parcours de déploiement, les risques opérationnels et les personnes d'astreinte. Nous pouvons définir un changement avec des vérifications, une procédure de reprise et une responsabilité claire.

Consulter les tarifs Contactez-nous →

Rendez la terminologie concrète

Demandez comment « Qu'est-ce qu'un retour arrière de déploiement ? » s'applique à votre organisation. Un exemple concret tiré de votre système actuel suffit pour engager la discussion.

Découvrir nos prestations

welcome@engineer.company +45 50 10 61 84 Copenhague, Danemark

Emportez cette page avec vous

Fiche de contact

Créé avec soin et amour par D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Droit d'auteur © 2025 – aujourd'hui · Engineer ApS · Tous droits réservés.

Imprimé depuis https://engineer.company/fr/glossary/deployment-rollback/