Engineer

デプロイのロールバック

不具合のあるデプロイを取り消し以前の構成に戻すこと

どのようなものか

デプロイのロールバックは、変更によって問題が起きたときに、以前のバージョンや設定を再び使える状態にすることです。具体的な操作はデプロイを管理する仕組みによって異なります。すでに待機している旧バージョンへトラフィックを切り替える場合もあれば、保存したリビジョンを使ってアプリケーションのインスタンスを作り直す場合もあります。KubernetesでDeploymentをロールバックすると、以前のリビジョンのPodテンプレートが復元されます。データベースなど、外部にある状態はその操作と別に扱います。

リポジトリでコミットを打ち消すと、以前の変更を取り消す新しいリビジョンができます。それを運用中のサービスに反映するには、ビルドとデプロイが必要です。ロールバックでは、保存していた成果物をその場で使えることがあります。緊急対応が終わったら、稼働させたいバージョンと宣言された設定をそろえます。次の自動デプロイで問題のあったバージョンが戻らないようにするためです。

コードを戻した後にも、新しいバージョンが書いたレコード、イベント、ファイルは残っている場合があります。以前のコードがそれらを処理できれば、利用者が直前に行った作業を残せます。データベースを過去の時点に復元する操作には、対象時刻とデータへの影響を見極める別の手順が必要です。どの部分を戻し、どのデータを保存するかを先に決めます。

概念を判断に生かす

「デプロイのロールバックとは？」を実際の判断に生かしましょう。その概念を、開発または購入するシステムに結び付けて考えるお手伝いをします。 変更を検証し、リリースし、元に戻せる仕組みを整え、金曜の深夜ではなく火曜の午後にお客様へ届けられるようにします。

サービスを見る お問い合わせ →

用語の背後にある前提を検証する

解説とリンク先の実績を参考に、プロジェクトについての質問を整理してください。ツールを選ぶ前に、前提を一緒に検討できます。 GitHub ActionsでGIS SaaSのデプロイを自動化 — デプロイ、データ、報告を手作業から解放

GitHub ActionsでGIS SaaSのデプロイを自動化 お問い合わせ →

なぜ重要なのか

デプロイ後に障害が発生すると、まずどうすればサービスが実際に使えるようになるかを判断します。以前の成果物が残り、トラフィック切り替えを試したことがあるなら、ロールバックが候補になります。急いで修正を作るより復旧が早い場合があります。旧版への切り替えで新たな失敗は止まっても、すでに誤った状態で記録されたデータには別の対応が必要です。何を復旧の目標とするかを担当者間で共有します。

プロセスの起動は一つの手掛かりです。購入手続きに影響があったなら、テスト用アカウントで一件の注文を試します。どのバージョンが応答したかを確認し、関連するキューやバックグラウンド処理のエラーも見ます。特定の条件でだけ発生した問題なら、その条件を再現するか、影響を受けた利用状況を観察します。復旧したと判断する前に観察時間を決めておくと、慌てずに評価できます。

以前のリビジョンが一覧にあっても、参照先のイメージがレジストリから削除されていることがあります。証明書や認証情報が失効している可能性もあります。復旧できると約束した期間だけは必要なパッケージと設定を使える状態にします。当番の担当者が実際に持つ権限で試し、通常の管理画面が使えない場合にどこから手順を読めるか確かめます。

実際の活用例

Azureへのデプロイを一貫した手順で運用した事例では、開発、事前検証、本番の順に変更を送っています。環境間の経路が明確であれば、どの場所でどの版を試すか整理できるため、復旧も計画しやすくなります。この事例はデプロイの経路を説明しています。次は、別の状況でロールバックを試すための例です。

注文サービスのバージョンBを入れた後、保存済みの住所を選ぶとエラーが出るとします。直前まで使っていたバージョンAの成果物と設定が残っています。まずBがAに読めないデータを書いたかを調べます。どちらのバージョンも現在のデータを使えるなら、Aにトラフィックを戻してテスト注文を行います。応答したバージョンとキューの状態を確認します。切り替え前に始まった注文が、その後どうなったかも追跡します。

BがAに必要な列を消していたら、すぐに戻すとさらに障害が起きるかもしれません。該当する書き込みをいったん止め、現在のデータを扱える修正版を入れる方法もあります。計画した変更では先に新しい構造を追加し、読み取り側と書き込み側を段階的に移行します。古い構造を削除するのは、復旧に使うバージョンもそれを必要としなくなってからです。移行中の状態で両方のバージョンを試します。

問題が起きるところ

管理コマンドの成功は、新しい設定が受け付けられたことしか示さない場合があります。新しいインスタンスがまだ起動中だったり、トラフィックが切り替わっていなかったりします。どのバージョンが実際に応答するかを見て、事前に合意した条件を満たすまで観察します。元のエラーが続くなら、原因の調査に戻ります。

機能のスイッチ、トラフィックの規則、定期実行のジョブは、アプリケーションのイメージとは別に反映されることがあります。旧版に戻っても、別の処理が新しい形式で書き続けるかもしれません。関係する制御を洗い出し、切り替える順番と担当を決めます。新しい形式のリクエストを送るクライアントや、処理中のジョブについても対応を考えます。

定期的に設定へ収束する仕組みがあると、手動操作の後で問題のあったバージョンに戻る場合があります。宣言された構成の更新と、成果物やトラフィックの切り替えを調整します。どのリビジョンを使い、いつ変更し、利用者にどんな影響があったか記録します。修正ができたら、通常の確認を経て新たにデプロイできます。

参考資料

- KubernetesのDeploymentとロールバックは、リビジョンとPodテンプレートの復元範囲を説明します。
- DORAのデリバリー指標は、変更の流れ、障害、復旧の指標を扱います。
- PostgreSQLのテーブル変更は、スキーマの変更方法とその範囲を説明します。
- Helm rollbackは、以前のリビジョンを選択する操作を説明します。



用語集

実際の制約に合わせて選ぶ

デプロイの流れ、運用上のリスク、緊急対応の担当者をお知らせください。検査、復旧手段、明確な責任の所在を伴う変更の範囲を定められます。

料金を見る お問い合わせ →

用語を実務に結び付ける

「デプロイのロールバックとは？」がお客様の組織にどう当てはまるかをご相談ください。現在のシステムの具体例が一つあれば、話を始められます。

サービスを見る

welcome@engineer.company +45 50 10 61 84 デンマーク、コペンハーゲン

このページを持ち歩く

連絡先カード

心を込めて丁寧に制作： D.

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

著作権 © 2025 – 現在 · Engineer ApS · すべての権利を留保します。

印刷元： https://engineer.company/ja/glossary/deployment-rollback/