

CI導入事例

2018/5/24

株式会社トライビート

畔蒜 真弘<kurone masahiro>

アジェンダ

1. CI導入経緯
2. CI構築図
3. 気をつけたこと
4. はまったところ
5. 反省点
6. CI導入してみても
7. 今後の課題

CI導入経緯

きっかけ

- システムの横展開が控えてる
- 機能の改修が増えている
- 開発リソースの拡大
- 成長と技術的負債の解消

CI導入経緯

導入前のシステムの状態

- 開発、運用に関して問題なく、一定の品質が担保されています
 - 致命的なバグが連発してしまうようなコードではありません
 - システムに関してお客様の不満はありません
 - レガシーコードにもなってません
 - リリースの手順書はあります
- ．．．．今すぐ開発、運用の見直しが必要な状態ではありません

CI導入経緯

見えていた問題点

- リリース作業が手作業になっている
 - ヒューマンエラーがいつおきてもおかしくない
- 個人のローカル環境からリリース作業が行われている
 - 環境が変わった場合、リリースできなくなる可能性
- ユニットテストの未実施
 - プロジェクトが大きくなった場合のコードの品質低下の可能性

リリース作業に対してCI導入を行ったほうがよい

CI構築図

サービス環境

- インスタンス
 - Azure App Service
- DB
 - SQLServer
- ソースコード
 - Github

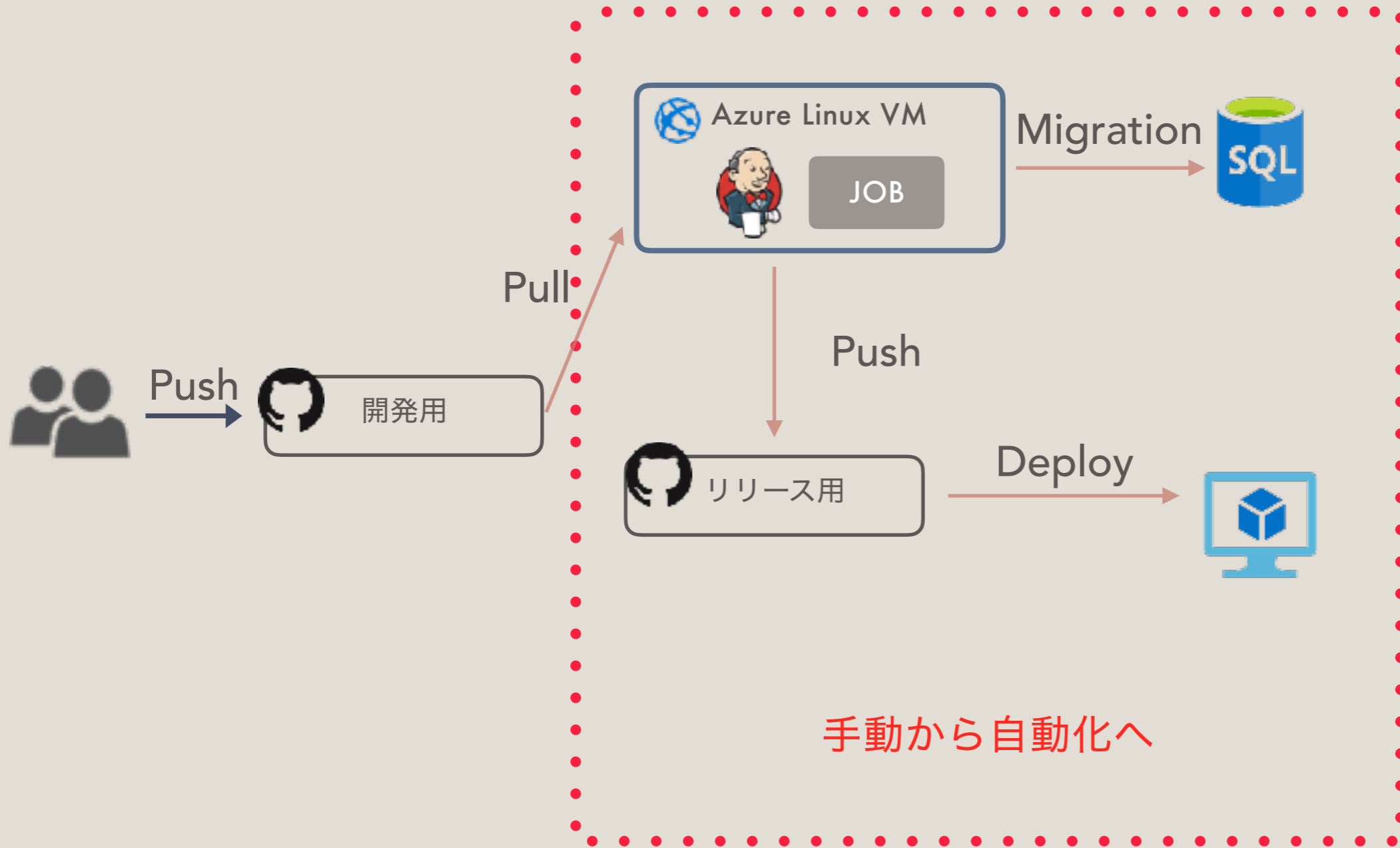
CI環境

- インスタンス
 - Azure Linux VM
- CIツール
 - Jenkins
- ジョブ(シェルスクリプト)
 - Github

Jenkins on AzureでCI環境を構築

- AzureがJenkinsの公式サポート(テンプレートですぐに構築可能)
- リリース作業は単純なコマンドで構成されており手順書もあるため、Jenkinsのジョブに設定/実行/確認が素早く出来る

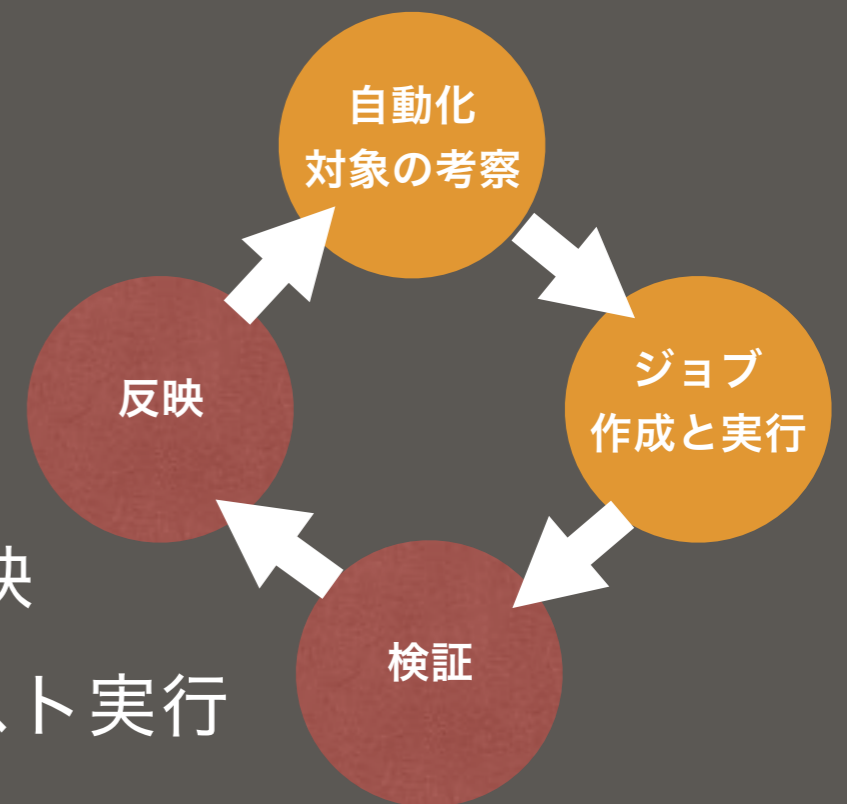
CI構築図



気をつけたこと

スモールスタートで行う

- リリースの自動化から着手
 - 簡単な作業から徐々に自動化していく (Gitコマンドとか)
 - 作業内容ごとに区切ってジョブを登録していく
 - 開発環境リリースのジョブ
 - 1 : Gitからビルド用ソースを取得
 - 2 : ビルドコマンドの実行
 - 3 : リリース用GithubへPush
 - 4 : GithubのPushを検知して、開発環境へ反映
 - 5 : 本番のみの実行内容を開発環境向けにテスト実行

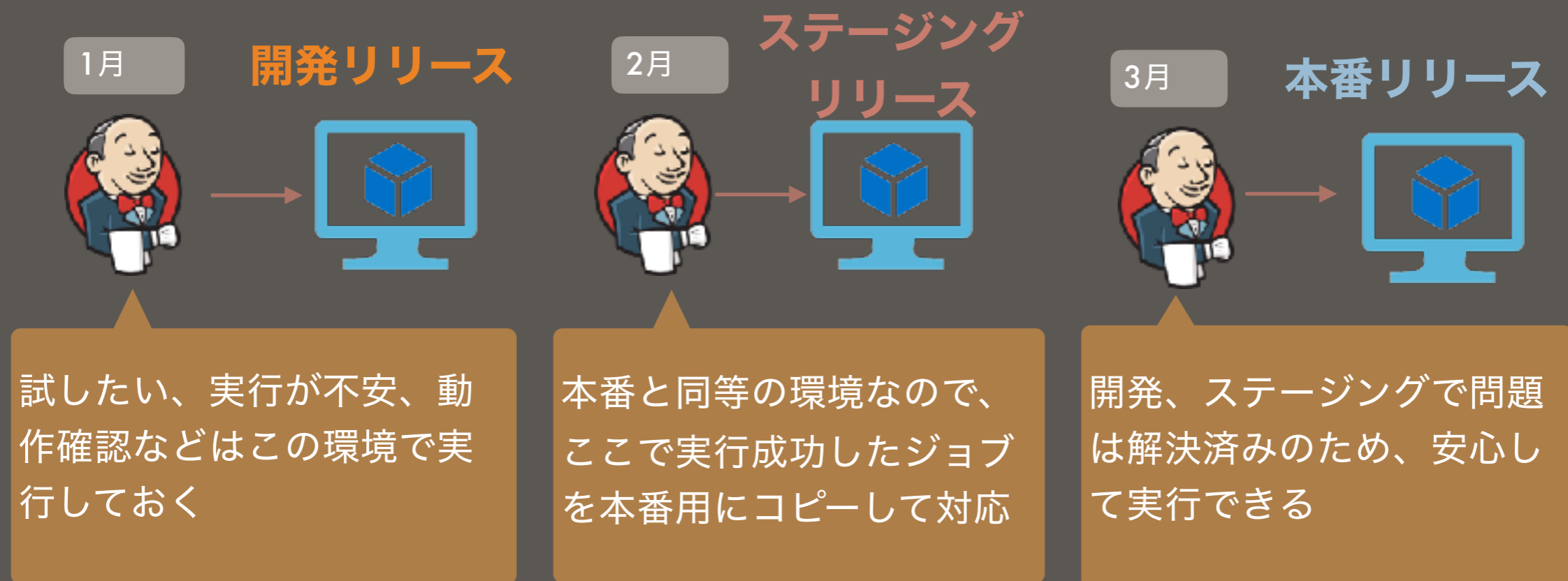


気をつけたこと

致命的な不具合を本番で発生させない

- 段階的にCIを導入していく
 - 開発(壊れてもいい環境)、ステージング(本番反映前の確認)、本番の3環境で運用していたため

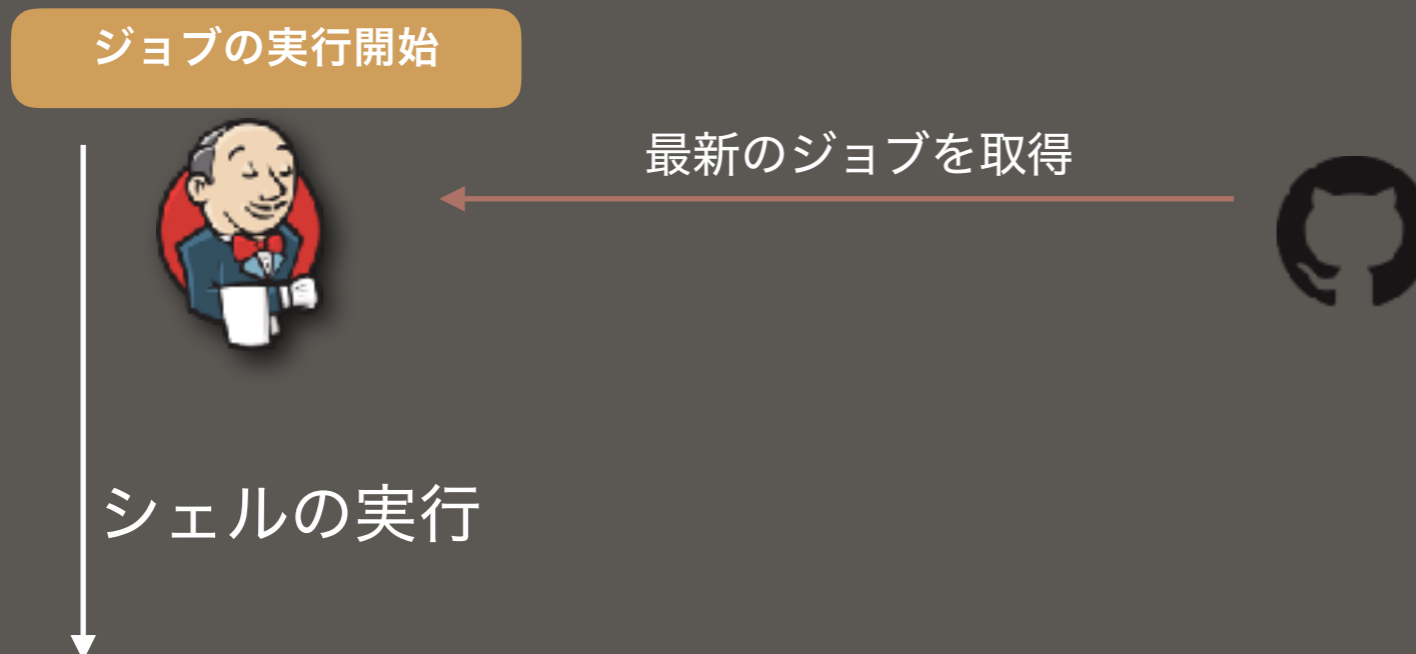
開発→ステージング→本番の順でリリース時にCIを導入



気をつけたこと

実行内容の可視化(JenkinsジョブのGithub管理)

- JenkinsジョブのシェルスクリプトをGithubで管理する
 - Github上で関係者はいつでもジョブの内容を確認できるようになる
 - Githubにジョブの更新をPushしておくだけで済む
 - 各ジョブの実行時にGithubから最新のシェルスクリプトを取得する



はまったところ

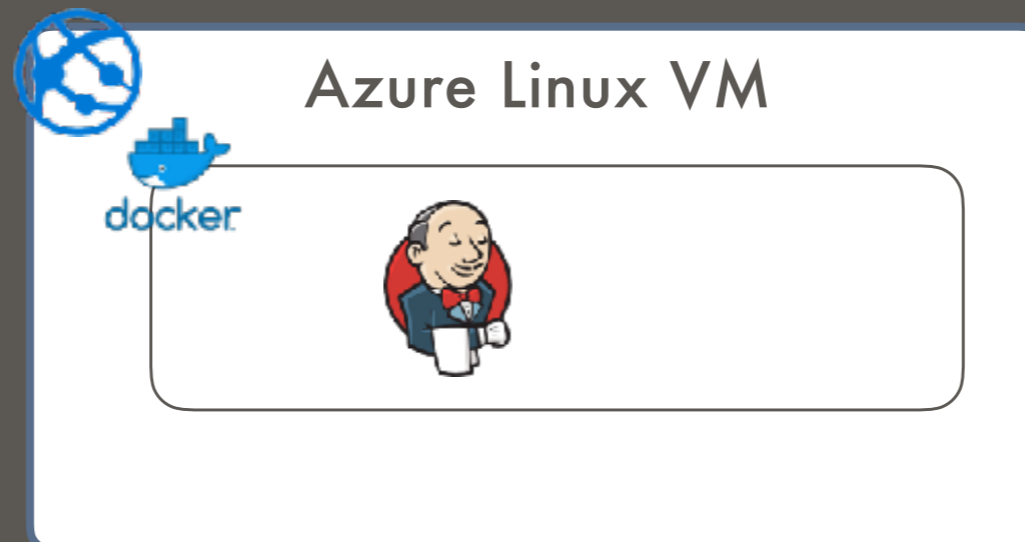
はまったところ

- 不具合時の実行内容確認
 - 予期せぬエラーが発生した場合など、ログから追えない場合などがある
 - ジョブ実行時のログに必要な情報を全て出力するよう作成する
- メモリ不足
 - 実行コマンドの一部でメモリが足りなくなってしまう、
動かない/実行速度が遅くなってしまった
 - 想定より料金が高くなってしまったので、環境の起動時間を減らし対応

反省点

反省点

- CI環境に直接Jenkinsインストールしての運用
 - 環境が壊れかけた時があり、環境を1から作り直すことになりそうに・・・
 - 環境にJenkins用のDockerを構築して運用する方法もあった
 - 壊れても復元が容易なため
- 環境作成時に実行したコマンドから作成出来るように準備中



CI導入してみても

CI導入してみても

- リリース作業をしていた担当者の負担が減った
- 他の業務に時間がさけるようになった
- 作業内容、履歴が残るようになった
- 属人的ではなくなった
- 各開発者のローカル環境のライブラリのバージョンなどが統一された

★良いことしか起きてません！

CI導入してみても

運用時の課題、要望が上がってきている

- 運用後に見えてくる課題
 - 通知、連携、制御など
- 追加処理の要望
 - 便利になったから他の作業も対応したい
 - 逆に手間が増えてしまった作業に対してのフォロー
- 優先順位を決めて、徐々に対応していく
- なるべく設定済みジョブに対して更新は行わず、新規のジョブで対応する

ご清聴ありがとうございました