

ランサーズSRE 1年間の取り組みとこれから

Lancers

自己紹介

日吉 陽介

ランサーズ株式会社

プロダクト開発部 アーキテクトグループ

マネージャー

2022年1月入社

CakePHP/Laravel/Ruby on Rails/Vue.js/
Nuxt.js/Flutter/Docker/AWS/GCP etc...

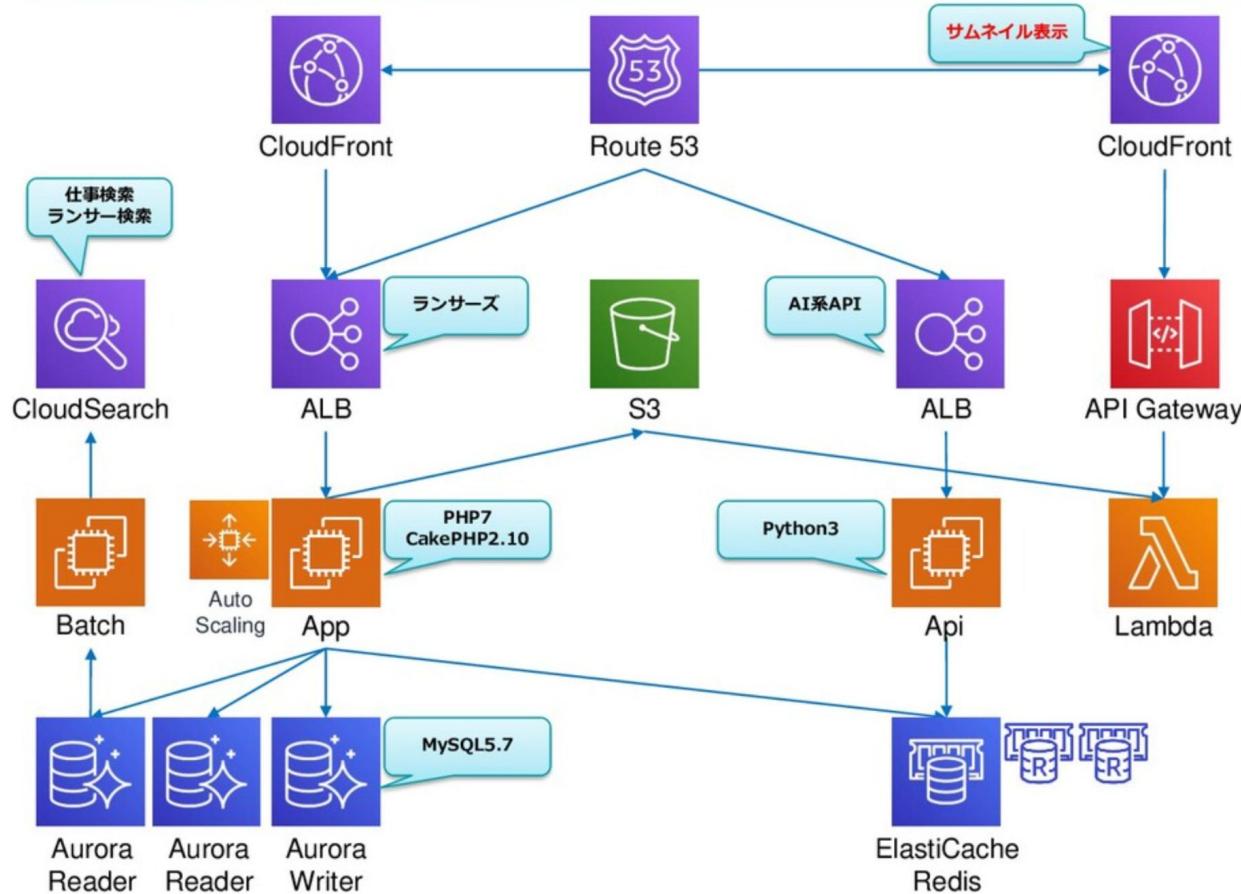
 yosukehiyoshi



今回お話したいこと

- ・これまでのインフラ構成
- ・App、ならびに Batch をコンテナに移行
- ・長く使われてきた自前のデプロイシステムを廃止
- ・CI/CD を CircleCI に移行
- ・移行後のレスポンス、負荷
- ・ランサーズSREチームの課題と解決策
- ・これからの取り組み

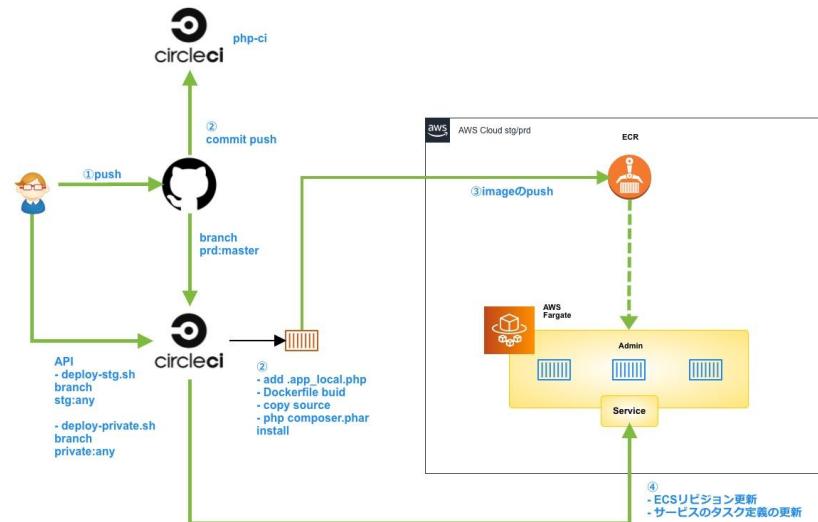
ランサーズSRE 1年間の取り組み



移行するにあたって

- まずは管理画面を **ECS/Fargate** に移行
- Batch は **Arm**
- App は Arm に移行できず Intel
- ※ 当時 Datadog の **dd-trace-php** が
Arm をサポートしていなかった

管理画面のデプロイ構成



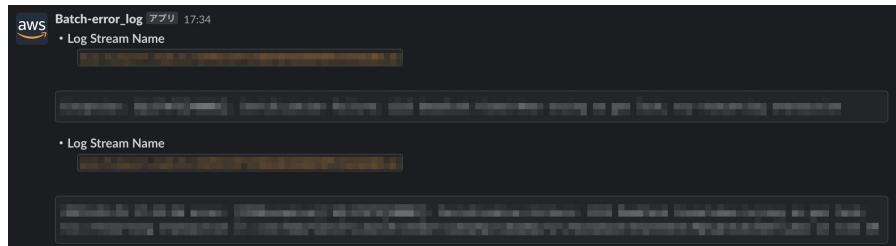
フロントエンドのネックと改善

- ・ソースコードが **5GB** 以上
 - `yarn install` で **30分** 以上かかる
 - コンテナデプロイで時間が**膨大**にかかる
- ・フロントエンドを分割してから移行してみては？
 - ソースの構造上、パスによる処理分割が困難で工数がかかる
- ・あらかじめローカルでビルドしたファイルをGit管理する運用
 - `yarn install` は不要になり、CI時間が**大幅に短縮**
- ・App コンテナ移行のための条件が整った！

Batch をコンテナ + EventBridge へ移行

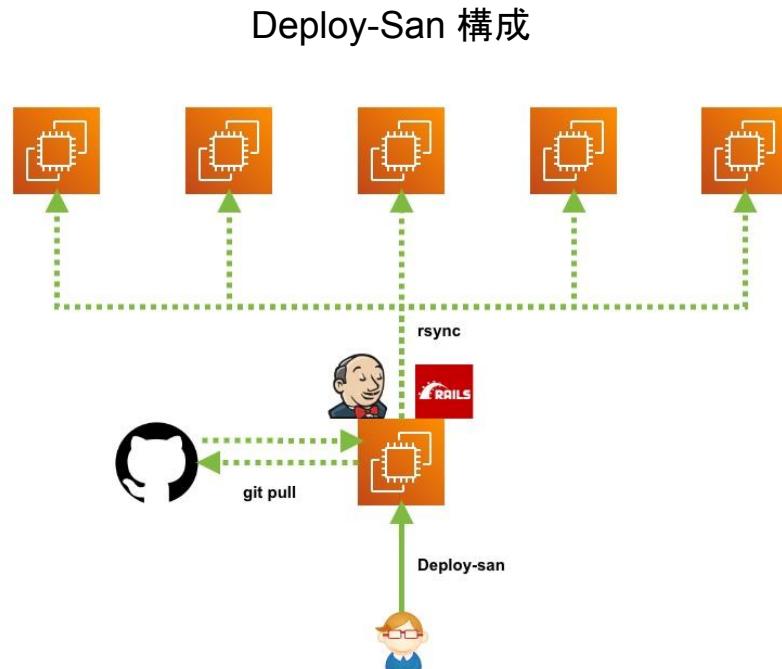
- ・バッチは100個以上ある
 - 冗長化せずEC2 1台のみで運用
 - サーバーダウンやAZ障害が起きるとサービスに影響が出る状況
- ・開発メンバーがTerraformで管理可能に
- ・CloudWatchLogs、SNS、Lambdaでエラー通知

Batch エラー時の Slack 通知



自前のデプロイシステム「Deploy-San」

- ・2015年より使われている
 - Rails + Jenkins
 - 独自シェルで rsync
- ・オートスケール時
 - Deploy-San から最新のコードを取得
 - 単一障害点
- ・リリース時間 - 差分更新で1分以内

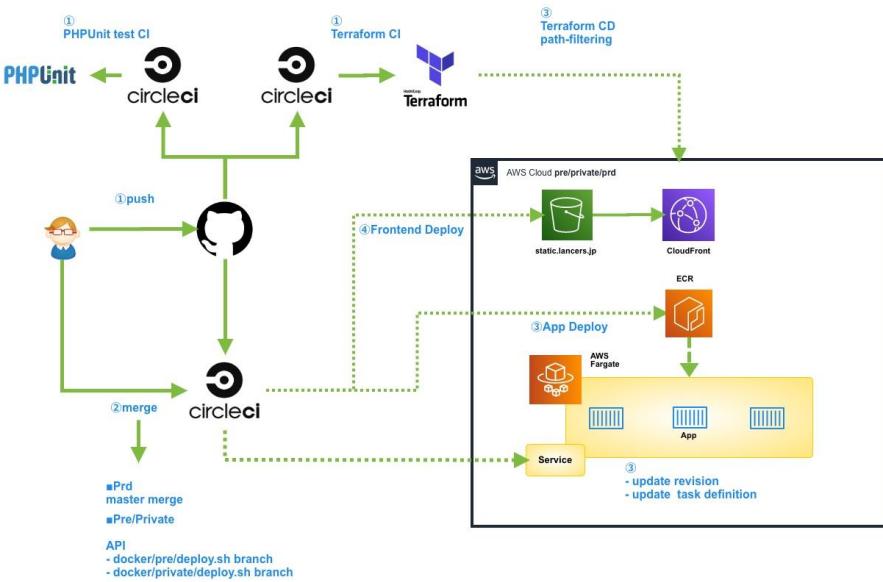


CI/CD を CircleCI に移行

- ・**基本フロー**

- **本番** → master マージ
- **ステージング/カナリア** → API 経由
- ・**コンテナデプロイの高速化**
 - .dockerignore、checkout_shallow
 - remote-docker-layer-caching
- ・**リリース時間5分以内まで縮めることができた**

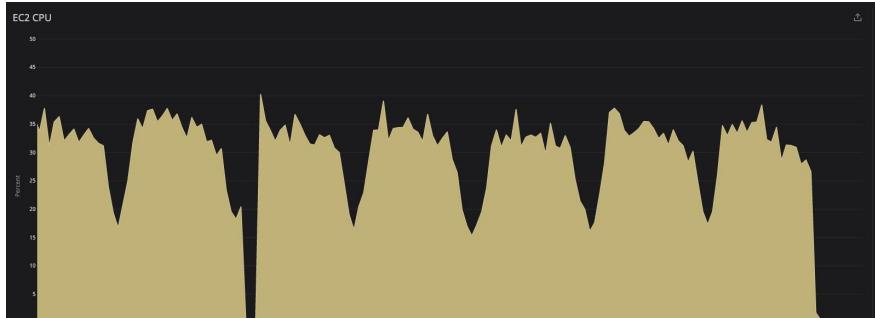
移行後の構成



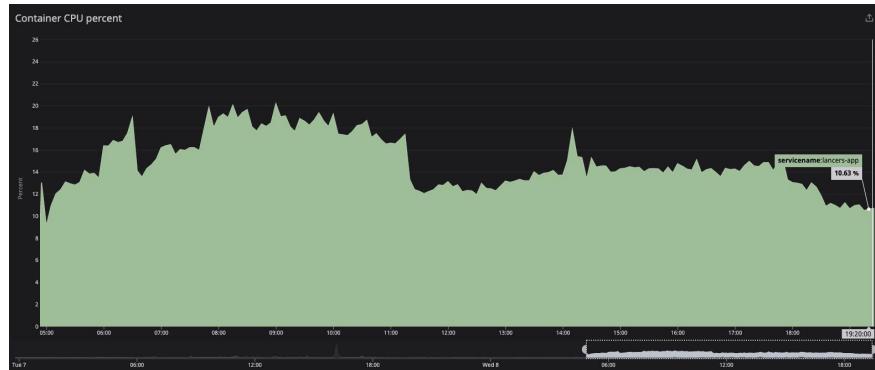
移行後のレスポンス

- ・ランサーズは負荷のピークが日中
- ・サーバレスポンスは数十ms程度
- ・現在の設定
 - CPU 60% で増加、40% で減少
 - オートスケール 7台

EC2の負荷



ECSの負荷



SREチーム 目下の課題

- ・関与する領域が広すぎるがゆえに、SREチームに**責務が集中**
- ・開発チームと連携の際にSREチームが**ボトルネック**となってしまう
- ・本来SREチームが注力すべきことに**注力できない**
- ・組織サイロ発生の要因となりかねない

課題に対する解決策

- ・SREチームがやらないことを定義、委譲を進める
 - アカウント発行業務やデータ分析基盤の運用サポートなど
- ・開発チーム内の自己解決力を上げ、委譲範囲を広げる
 - リファレンス、How to guideの整備
 - 逆 Embedded SRE 人材の育成

ランサーズSRE これからの取り組み

- ・引き続きSREチームの課題解決を進行
 - 本来あるべきSRE業務に注力できる状態にする
- ・前進させたい取り組みはまだ多い
 - CircleCI から GitHub Actions への移行
 - フロントエンド関連の環境をモダン化
 - 開発チームのエンジニアが気軽にAWS に触れることのできる **Sandbox** 環境
 - セキュリティ強化
 - インフラコスト削減 etc...

まとめ

- ・障壁は多かったが、インフラのモダン化をやりきることができた
 - 自前のデプロイシステムから CircleCI への移行、フロントエンドの改善 etc...
- ・コンテナ化と CI/CD 移行を複合でおこなうことにより、**生産性の向上**が期待できる
 - SRE のトイル削減
 - 開発部全体のリリースフローを**改善**できた
- ・今後はSREチームの課題解決と組織化を進めつつ、事業全体のさらなる**生産性向上**を前進させていく

ご清聴ありがとうございました

Lancers

Lancers