

GitHub Actions Runner の運用から見る ニンテンドーシステムズのチーム体制

ニンテンドーシステムズ株式会社
システム開発部

今日の内容

- ▶ 会社紹介
- ▶ 社内 GitHub Actions Self-hosted Runner の紹介
- ▶ チーム・体制について
- ▶ 体制のメリットと課題
- ▶ まとめと展望

会社紹介

ニンテンドーシステムズ株式会社



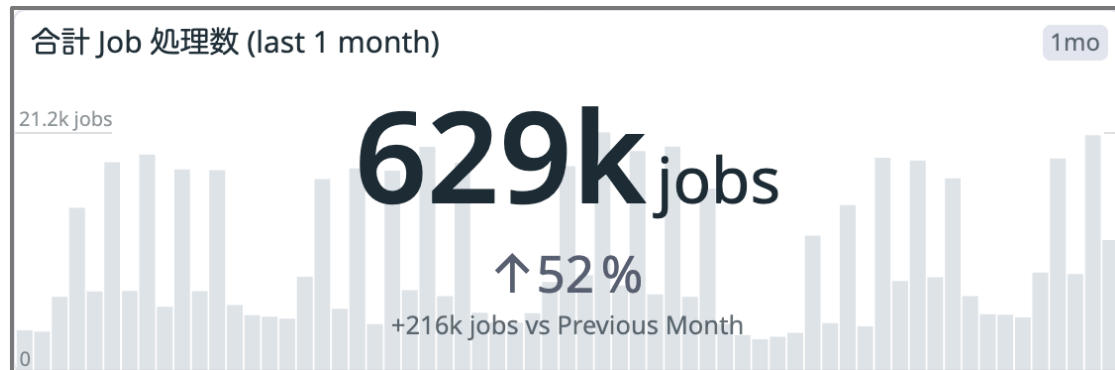
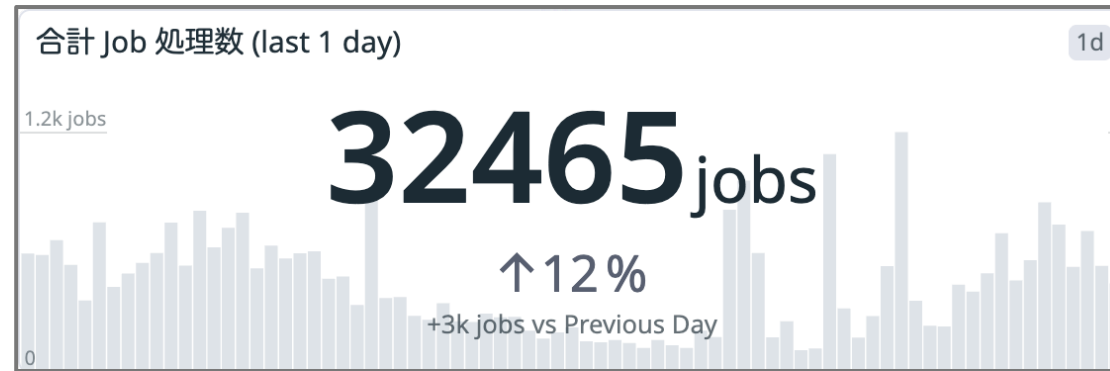
- ▶ 任天堂が展開するビジネスのデジタル部分に関するシステムの開発及び運用
- ▶ 新規サービスの企画、開発及び運用

ニンテンドーシステムズ株式会社

- ▶ 「任天堂が展開するビジネスのデジタル部分に関するシステム」の例
 - ▶ Nintendo Switch 2 / Nintendo Switch 本体向け機能
 - ▶ フレンドリスト、プッシュ通知 など
 - ▶ 汎用ゲームサーバ機能
 - ▶ ランキング、マッチング など
 - ▶ ECプラットフォーム
 - ▶ Nintendo Store、ニンテンドーeショップ
 - ▶ スマートフォン向けアプリ
 - ▶ Nintendo Music
 - ▶ ゲーム連携サービス
 - ▶ ZELDA NOTES、イカリング3 など

社内向け GitHub Actions Self-hosted Runner の紹介

利用状況

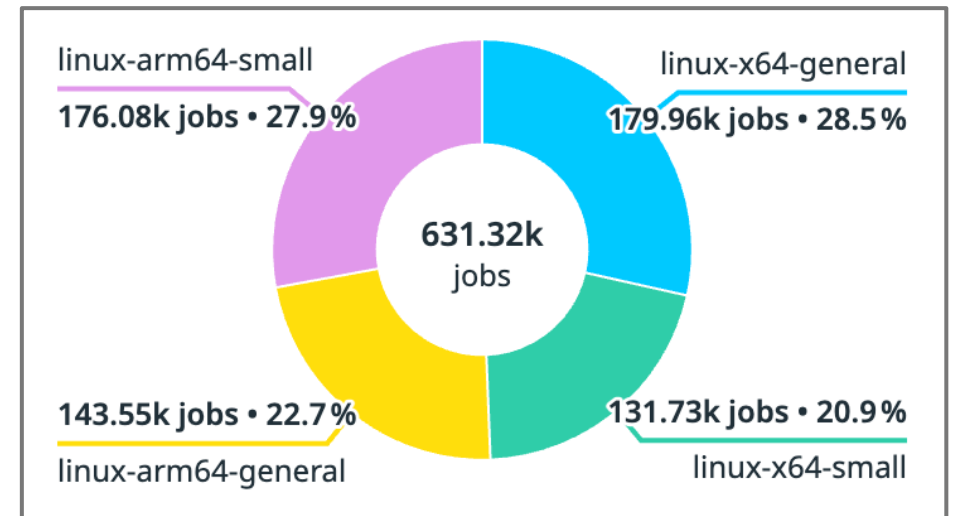


利用例

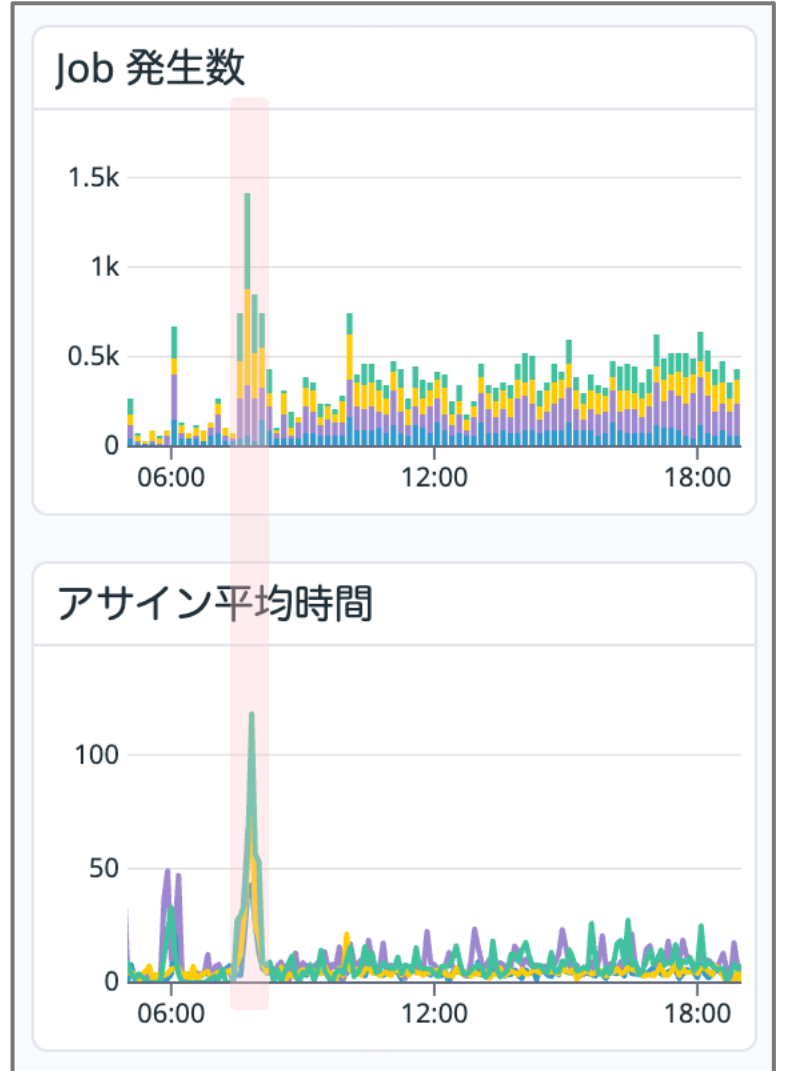
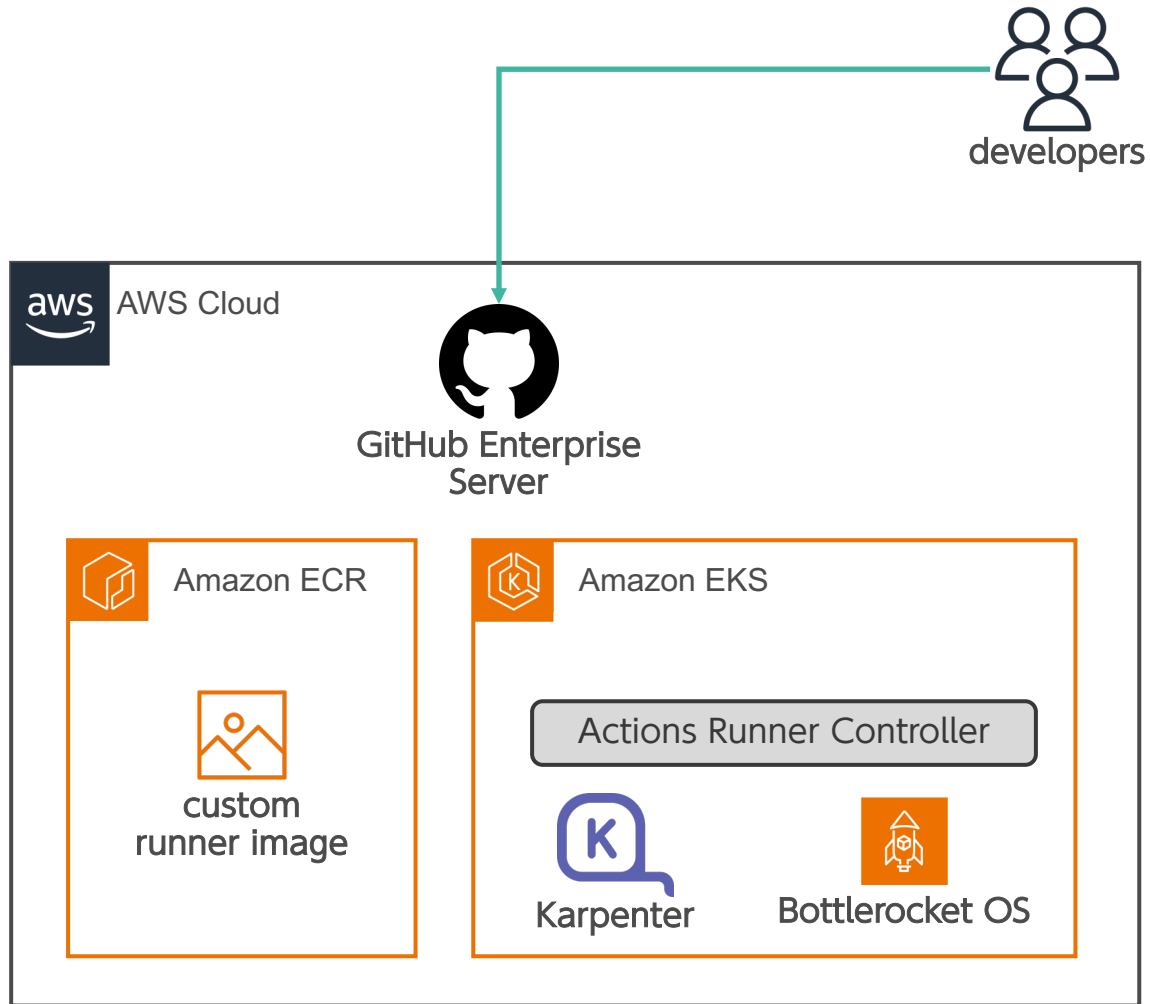
- ▶ アプリケーション CI/CD
 - ▶ Test / Build
 - ▶ Deploy
- ▶ Terraform パイプライン
 - ▶ PR が作成されると自動的に plan 結果がコメントされる
 - ▶ PR 上で apply が可能
- ▶ Jira / Confluence 連携

詳細

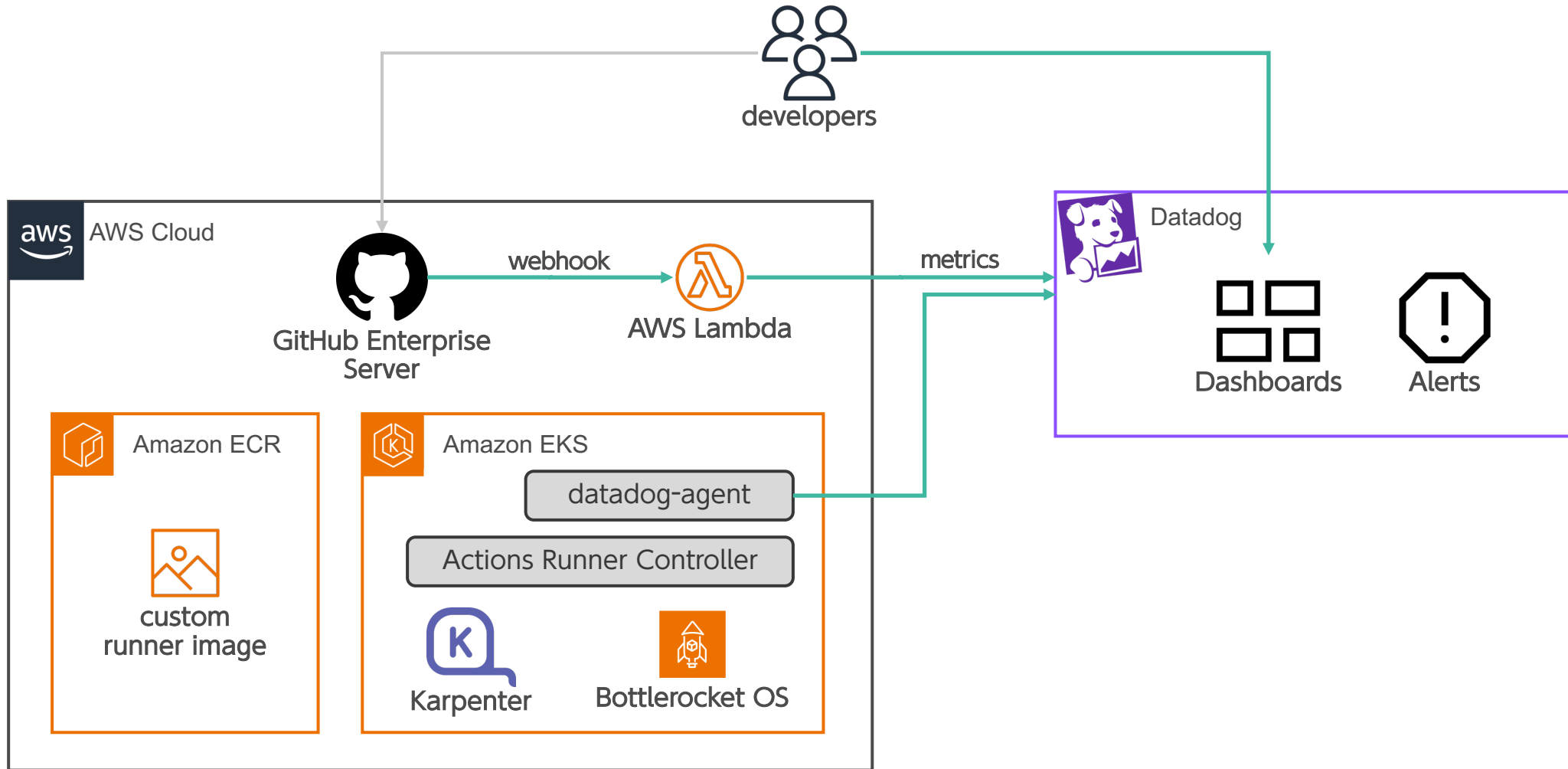
- ▶ なぜ Self-hosted Runner を使うのか
 - ▶ GitHub Enterprise Server を使用しているため
 - ▶ 社内ネットワーク内にあるシステムとの連携を行うため
 - ▶ Runner の IP アドレスを固定するため
- ▶ スペック
 - ▶ ephemeral mode で稼働
 - ▶ Linux x64 / Linux arm64 を提供
 - ▶ small / general の 2 タイプを提供



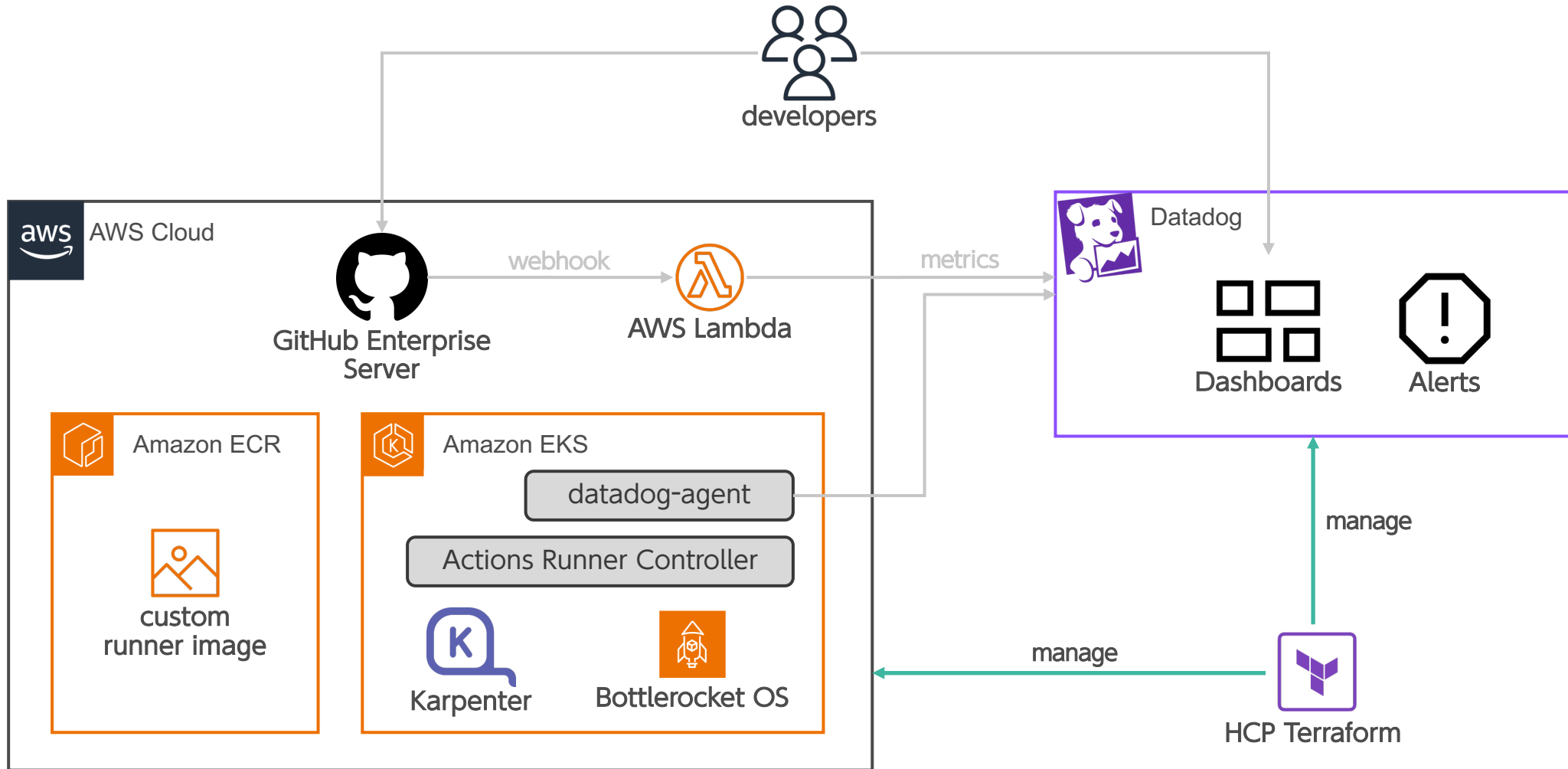
構成と採用技術



構成と採用技術



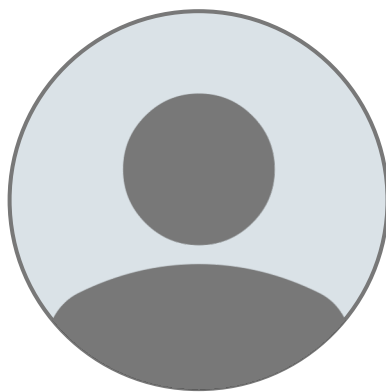
構成と採用技術



チーム・体制について

CI/CD チーム

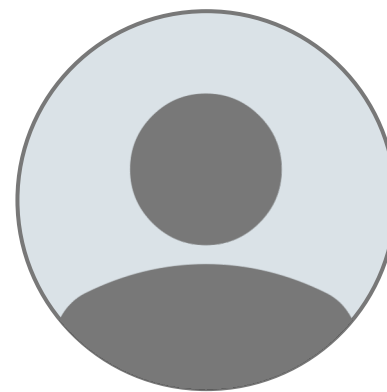
- ▶ Self-hosted Runner の開発・運用を中心に、CI/CD にまつわる仕組みをプロダクト横断で整備
- ▶ 普段は異なるプロダクトの開発を担当する3名が集まったチーム



ゲーム連携サービス
SRE



Switch 2 / Switch 本体機能
サーバ開発者



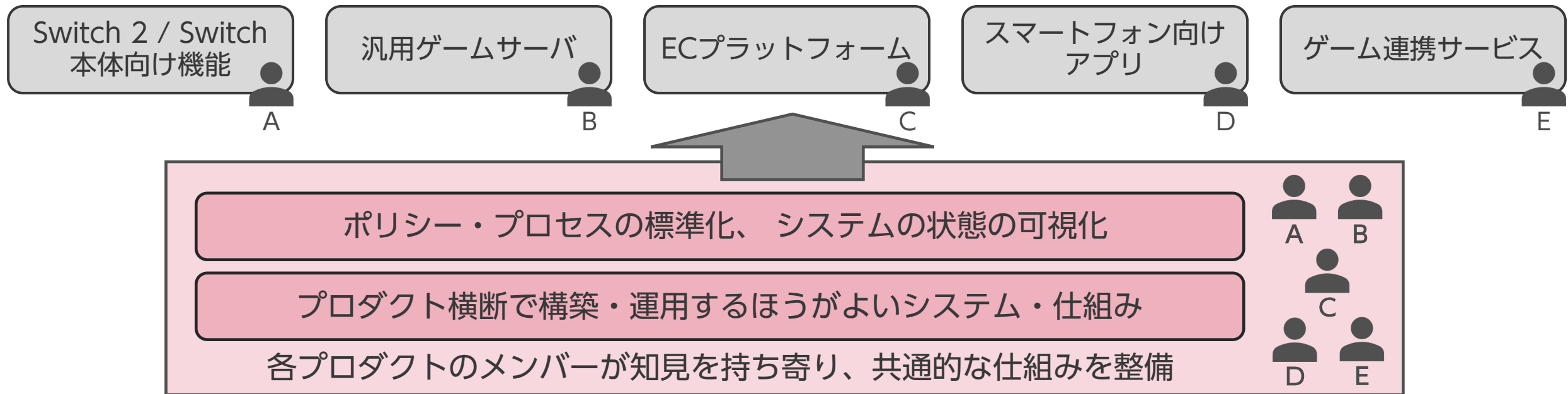
ECプラットフォーム
SRE

ニンテンドーシステムズのプロダクト構造

- ▶ 任天堂の娯楽体験を支える、幅広い領域のサービス開発
 - ▶ 本体向け機能、汎用ゲームサーバ、ECプラットフォームなど
 - ▶ グローバル規模のシステムを長期間・多数運用
- ▶ 各プロダクトが、それぞれに適した方法で開発・運用を行う
 - ▶ プロダクトの特性にあわせた技術選定や運用の判断により素早く柔軟に開発
- ▶ プロダクトごとの最適化が進む一方、全体の状況把握が困難に
 - ▶ 運用基準のバラつき、車輪の再発明などの問題が発生

プロダクト横断の基盤整備

- ▶ 技術選定、SLI/SLO などはプロダクトが主体となって判断
- ▶ ポリシーやプロセス、自動化の仕組みなどは横断的に整備し、各プロダクトの自律的な運用を支援



プロダクト横断の取り組みの例

- ▶ 可視化・標準化
 - ▶ プロジェクト予算・費用の可視化
 - ▶ サービスの利用状況・安定性の可視化
 - ▶ セキュリティ基準
 - ▶ 開発言語選択ルール
- ▶ 開発基盤整備
 - ▶ SaaS の導入および運用管理
 - ▶ GDPR 対応の自動化基盤の開発
 - ▶ CI/CD 基盤の整備
- ▶ など...

取り組みにおける考え方

- ▶ ポリシーや仕組みの整備は行うが、具体的にどうするのかは各プロダクトの判断に任せる
 - ▶ 担当領域について一番考えているのは担当者自身
 - ▶ プロダクトごとの文化の違い・自律性を尊重する
- ▶ CI/CD チームの場合
 - ▶ GitHub Actions はあくまで選択肢のひとつであり、導入は任意
 - ▶ 新規開発の際に手軽に選択できる CI/CD 基盤として提供
 - ▶ 既存システムでも移行が進み、全体の運用負荷の軽減に繋がった

体制のメリットと課題

体制のメリット (1/2)

- ▶ 利用者の要求を先回りして拾うことができる
 - ▶ 運用者自身が、その基盤の利用者でもある
 - ▶ 要望が上がる前に、自分たちの実感や計測をもとに改善できる
- ▶ 例：ストレージ性能の改善
 - ▶ Runner の割り当てに時間がかかっていることに気づく
 - ▶ Disk I/O の詰まりが原因であることに気づき、EBS を強化して対応
 - ▶ インスタンスストア (ローカル NVMe SSD) を導入するなど、先回りして改善

体制のメリット (2/2)

- ▶ 各プロダクトの知見や改善を、より広い範囲に提供できる
 - ▶ 当事者だからこそ、現場の良い工夫や車輪の再発明に気づける
 - ▶ それぞれの事情を考慮して、無理のない形に共通化できる
- ▶ 例：Renovate の共通基盤化
 - ▶ 依存関係の更新 PR を自動作成するツール
 - ▶ Actions 上で実行する仕組みがプロダクトごとに運用されていた
 - ▶ CI/CD チームで共通基盤として整備
 - ▶ 個別の設定は各プロダクトで持てるようにしつつ、実行基盤を共通化
 - ▶ Actions と同様に、新規開発の際に手軽に導入できるようにすることを目指す

体制の課題

- ▶ プロダクト開発との兼務のため、工数を多く割けない
 - ▶ メインのプロダクト開発の状況に左右されやすい
- ▶ 利用規模の拡大に対して、少人数チームでの対応は難しい
 - ▶ 利用者が増えるほど、要望や問い合わせも増えていく
 - ▶ サポートで手一杯になり、新たな改善が進まないことも

課題に対する工夫：運用の効率化

- ▶ 基盤の更新作業をできるだけ自動化
 - ▶ Renovate により Runner のソフトウェア更新 PR を毎週作成
 - ▶ CI/CD チームの規模では、小さな差分を高頻度に取り込む方が運用しやすい
 - ▶ デプロイ作業も PR 上で完結
 - ▶ HCP Terraform と Actions の連携による terraform plan / apply の自動化
 - ▶ plan 結果をレビューし、問題なければ apply を承認するだけ
- ▶ 問題が発生した場合にすぐロールバックできるように
 - ▶ Immutable なイメージタグ運用
 - ▶ 2つのクラスタを使った Blue/Green デプロイ

課題に対する工夫：サポートの効率化

- ▶ Slack に利用者を全員追加したチャンネルを用意
 - ▶ 更新情報のアナウンスや、利用者からの質問を受ける場として利用
 - ▶ 利用者同士の会話で問題が解決することも
- ▶ 基盤のメトリクスをまとめたダッシュボードを公開
 - ▶ ジョブの平均アサイン待ち時間
 - ▶ 正常に稼働しているか、どれくらい混雑しているかがわかる
 - ▶ 時間ごとのジョブ実行数
 - ▶ スケジュール実行のジョブを作る際に、混雑を避けるためのヒントになる

[linux-x64-small] アサイン平均時間

3.04s

[linux-arm64-small] アサイン平均時間

2.99s

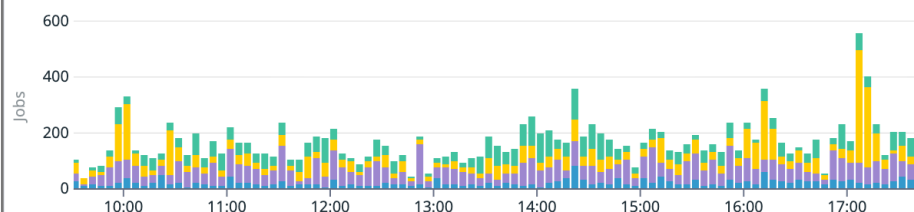
[linux-x64-general] アサイン平均時間

6.83s

[linux-arm64-general] アサイン平均時間

5.92s

[type別] Job 発生数



まとめと展望

まとめ

- ▶ ニンテンドーシステムズの Self-hosted Runner について紹介
 - ▶ クリーンな実行環境を、必要なときに素早く提供
 - ▶ CI/CD、社内ツール連携など、月間60万以上のジョブを実行
- ▶ 各プロダクトのメンバーによる横断的な基盤整備
 - ▶ 利用者に近い立場から、現場の要求に先回りして対応
 - ▶ 各プロダクトの自律性を尊重しながら、全体の開発力を底上げ
- ▶ 少人数・兼務の体制で運用を続けるための工夫
 - ▶ 基盤の更新の自動化、Slack や公開ダッシュボードの活用

展望

- ▶ Runner ラインナップの拡充
 - ▶ 開発環境の多様化に伴う Windows/macOS Runner の要望への対応
 - ▶ Nintendo Switch 2 など、ゲーム専用機を利用した CI/CD のサポート
- ▶ Actions を土台とした仕組みの拡充
 - ▶ Renovate のような、Actions を利用した共通基盤の整備
- ▶ 幅広い領域のメンバーとの協力
 - ▶ 多様なプロダクト開発の視点を取り入れ、実用的な基盤を目指す

Nintendo®

**NINTENDO
SYSTEMS**