

# セルフホストランナーのAWS費用を30% 削減するまでの道のり

GitHub Actions Meetup Tokyo#4サイボウズ株式会社

開発本部 生産性向上チーム

加瀬 健太 (@Kessin11)

# 自己紹介

■ 加瀬 健太 (@Kessin11)

■ 経歴

■ ~2023/06 株式会社DeNA SWET第二グループ

■ 2023/07 サイボウズ株式会社 生産性向上チーム

■ 業務

■ Github Actionsセルフホストランナーの運用など

■ 日課はgithub.blog/changelogを見ること



 @Kessin11

 @Kessin11

 @kesin11.bsky.social

# Productivity Weekly

- 「1週間の間に発見された開発者の生産性向上に関するネタを共有する会」を社内で開催した内容を [zenn](#) で公開
- 100回を突破して現在160回！
- CI/CD関連のネタが多いのでぜひ



サイボウズ 生産性向上チーム 🍌 Publication

サイボウズ株式会社 開発本部 生産性向上チームです。サイボウズエンジニアの生産性を上げるために日々活動しています 🍌

174 Followers

フォロー中 📡

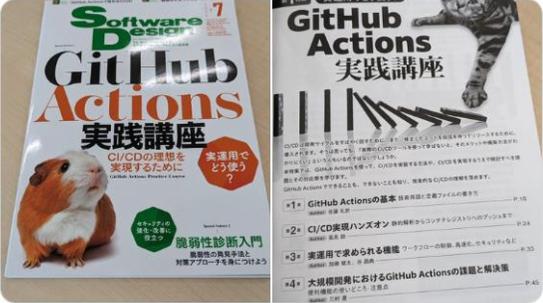
会社情報 🗨️ チーム紹介 🗨️ 採用 🗨️

<p>IDEA </p> <p>開発生産性系の話多し   Productivity Weekly(2024-07-17)</p> <p>Futa Hirakoba 4時間前 🍌 4</p>	<p>IDEA </p> <p>GitHub Projectsのwebhookの対応イベントが増えたなど   Productivity Weekly(2024-0...</p> <p>Futa Hirakoba 6日前 🍌 6</p>	<p>IDEA </p> <p>attestationsやfeature flagの話など   Productivity Weekly(2024-06-26)</p> <p>Futa Hirakoba 13日前 🍌 7</p>
<p>IDEA </p> <p>GitHubのカスタムプロパティのアップデートの話など   Productivity Weekly(2024-06-19)</p> <p>Futa Hirakoba 18日前 🍌 8</p>	<p>IDEA </p> <p>VSCoDe Chat Extensionやセキュリティ話など   Productivity Weekly(2024-06-12)</p> <p>Futa Hirakoba 28日前 🍌 9</p>	<p>IDEA </p> <p>ActionsのArm64ランナーがパブリックベータに。LFS料金体系変更   Productivity Weekly(202...</p> <p>Futa Hirakoba 1ヶ月前 🍌 8</p>

# Software Design 7月号の特集に寄稿しました！！

**Kenta.Kase** @Kesin11

今日発売のSoftware Design 7月号のGitHub Actions特集を生産性向上チームで書かせて頂きました！入門から実務で使われている玄人向けテクニックまで幅広く紹介していますので、ぜひお手にとってみてください！



午前10:55 · 2024年6月18日 · 2,464 件の表示

← **ポストする**

**mimu** @r4mimu

弊チームでGitHub Actions特集を書きました！自分も実践で遭遇する課題とその対処法 Tips を担当しました！何卒〜

📖 技術評論社販売促進部 @gihyo\_hansoku · 6月4日

【次号予告】2024年6月18日発売『Software Design 2024年7月号』本体1,420円+税、gihyo.jp/magazine/SD/ar...



**uta8a** @uta8a · 6月7日

GitHub Actions実践講座の3章後半を寄稿させていただきました。

ぜひSoftware Design 2024年7月号を買って、GitHub Actionsへ入門して実践レベルまで学んじましょう！

よろしくお祈りします！

📖 技術評論社販売促進部 @gihyo\_hansoku · 6月4日

【次号予告】2024年6月18日発売『Software Design 2024年7月号』本体1,420円+税、gihyo.jp/magazine/SD/ar...  
[DeepLで翻訳する](#)



**cangoxina** @Shitimi\_613 · 6月18日

#SoftwareDesign 2024年7月号で、「GitHub Actions 実践講座」の5章に寄稿させていただきました。大規模開発におけるGitHub Actions ワークフロー設計に関するtipsを紹介しております。

7月号は本日発売です！ぜひお手にとってください〜



**SoftwareDesign** @gihyosd

社内の開発チームの生産性向上をテクノロジーで支援する——サイボウズにはそんな役割を担ったチームがあります。その名もズバリ「生産性向上チーム」。

GitHub Actionsを活用してさまざまな業務の自動化に取り組まれているとのことで、7月号の第1特集は同チームの皆さんに寄稿していただきました。

1/6



CI/CDの理想を実現するために  
実運用でどう使う？

第1特集

CI/CDは開発サイクルをすばやく回すために、また、修正したコードを自信を持ってリリースするために、導入されます。そうは言っても、「実際のCI/CDツールを使って学ばないと、そのメリットや構築方法がわかりにくい」という人もいるのではないのでしょうか。

本特集では、GitHub Actionsを使って、CI/CDを実装する方法や、CI/CDを実現するうえで検討すべき課題とその対応策を学びます。

GitHub Actionsでできることも、できないことも知り、現実的なCI/CDの理解を深めます。

- 1\* **GitHub Actionsの基本** 技術用語と定義ファイルの書き方  
Author: 佐藤 礼弥 P.18
- 2\* **CI/CD実現ハンズオン** 静的解析からコンテナレジストリへのプッシュまで  
Author: 高見 訓 P.24
- 3\* **実運用で求められる機能** ワークフローの制御、高速化、セキュリティなど  
Author: 加藤 賢史、谷 昌典 P.33
- 4\* **大規模開発におけるGitHub Actionsの課題と解決策** 便利な機能の使いどころ・注意点  
Author: 三村 遼 P.45
- 5\* **大規模開発におけるワークフロー設計** 保守性や運用面の要件に応えるためのテクニック  
Author: 平本場 風太 P.52

午後0:22 · 2024年6月18日 · 1.6万 件の表示

# Cybozuの成熟したセルフホスランナー基盤の 最近の運用とコスト削減

# 目次

- Cybozuのセルフホストランナーの近況
- AWS費用の移り変わり
- Cybozuのセルフホストランナー基盤のアーキテクチャ紹介
- 無駄なプールサイズ削減
- スポットインスタンスの費用削減
- 今後：NAT Gatewayの料金について
- まとめ

# Cybozuのセルフホストランナーの近況

- 1年前の時点で既に機能は十分で安定運用



 cybozu

**philips-labs/terraform-aws-github-runner  
による  
GitHub Actions セルフホストランナーの大規模運用**

CI/CD Test Night #6  
サイボウズ株式会社 開発本部 生産性向上チーム  
宮田 淳平 (@miyajjan)

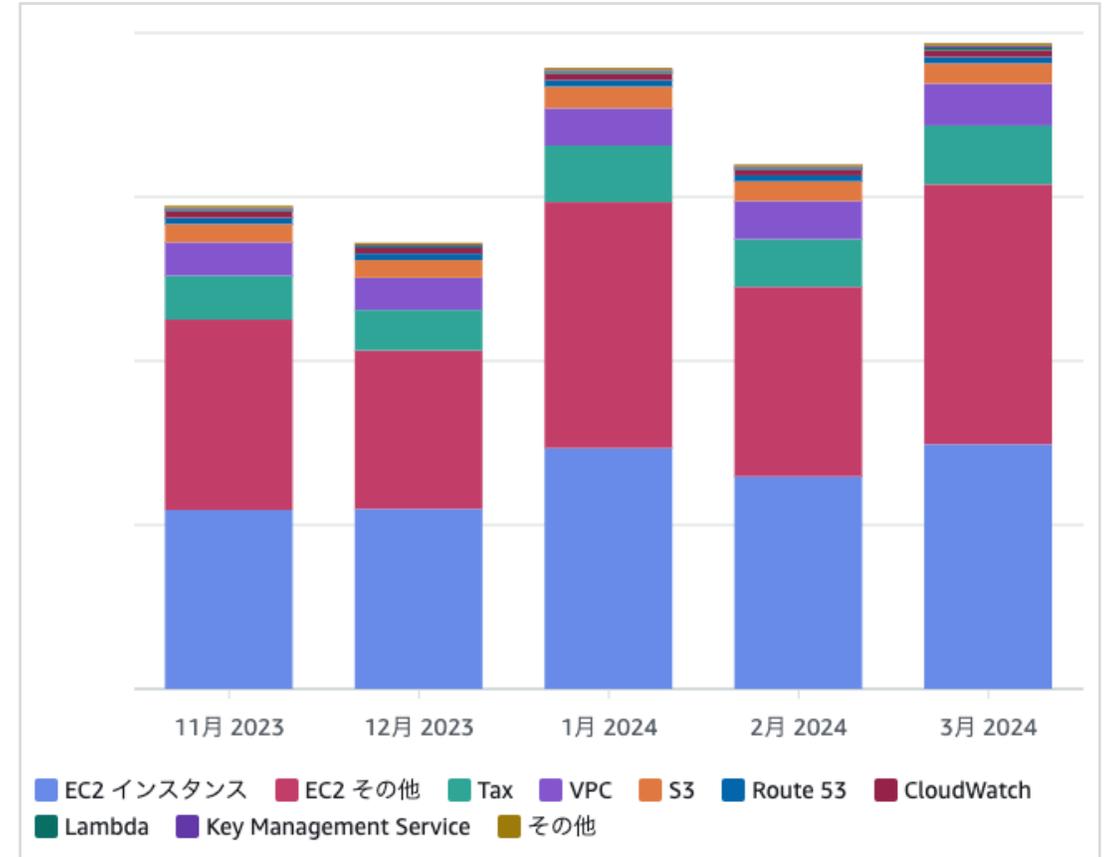
# Cybozuのセルフホストランナーの近況

- Terraformの運用が苦しい
  - tfstateが巨大すぎる問題
  - Terragrunt, tfcmt導入で運用の手間はかなり改善
- AWSのAPIリミットが苦しい
  - 上限緩和を申請できるものは申請して改善した
  - SSM ParameterStoreがたまにAPIリミットに当たるが対策が難しいので後回し
    - philips-labs/terraform-aws-github-runner が実質的にKVストアとして使っている
- 利用チームの増加
  - **AWSの費用増加がさすがに無視できなくなってきた**

# AWS費用の移り変わり

## AWS費用の移り変わり（削減前）

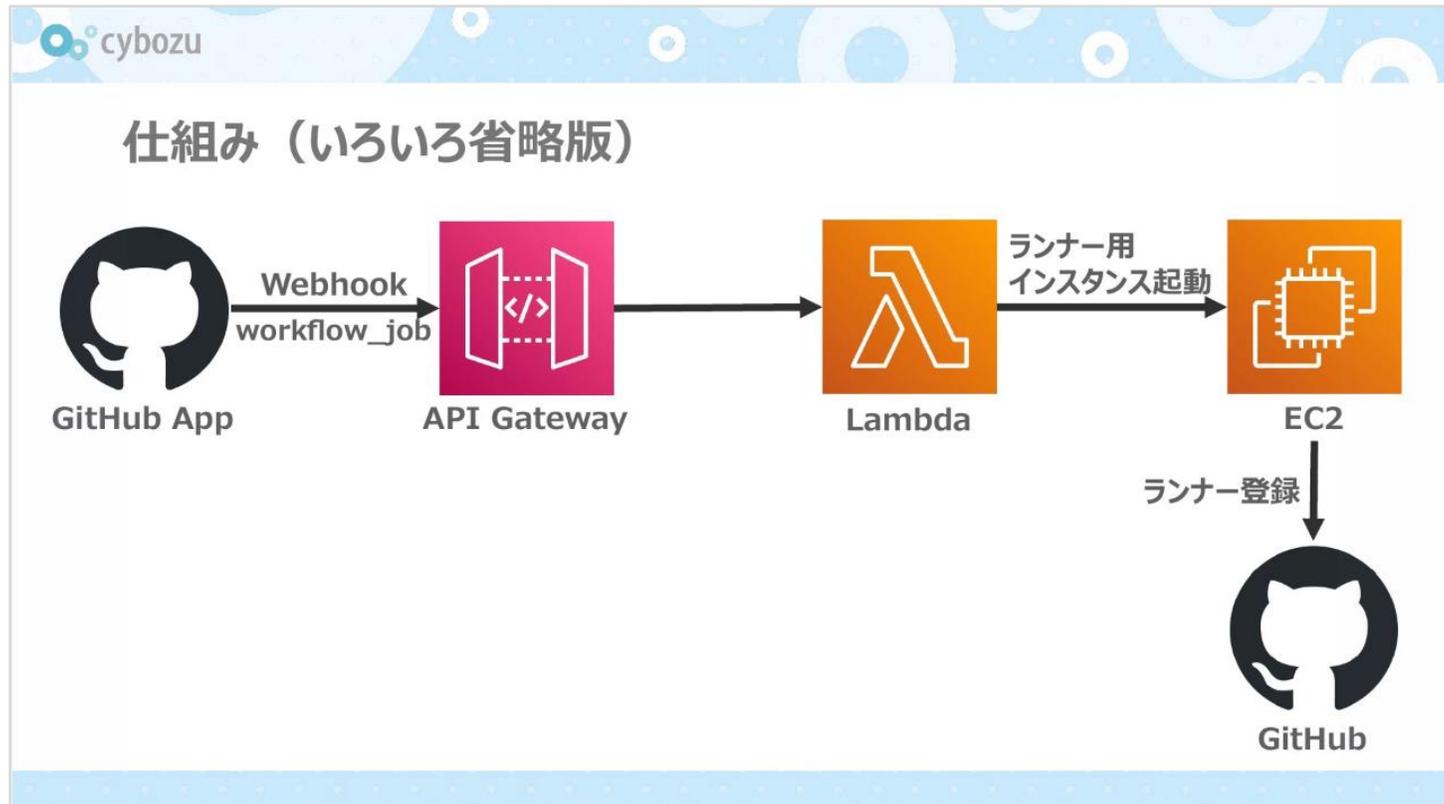
- 利用チームの増加や主力プロダクトのE2Eテストなどで右肩上がり
- 内訳はEC2とEC2その他（ほぼNAT Gateway）が2/3を占める



# Cybozuのセルフホストランナー基盤のアーキテクチャ

# Cybozuのセルフホストランナー基盤のアーキテクチャ

- OSSを利用 [philips-labs/terraform-aws-github-runner](https://github.com/philips-labs/terraform-aws-github-runner)

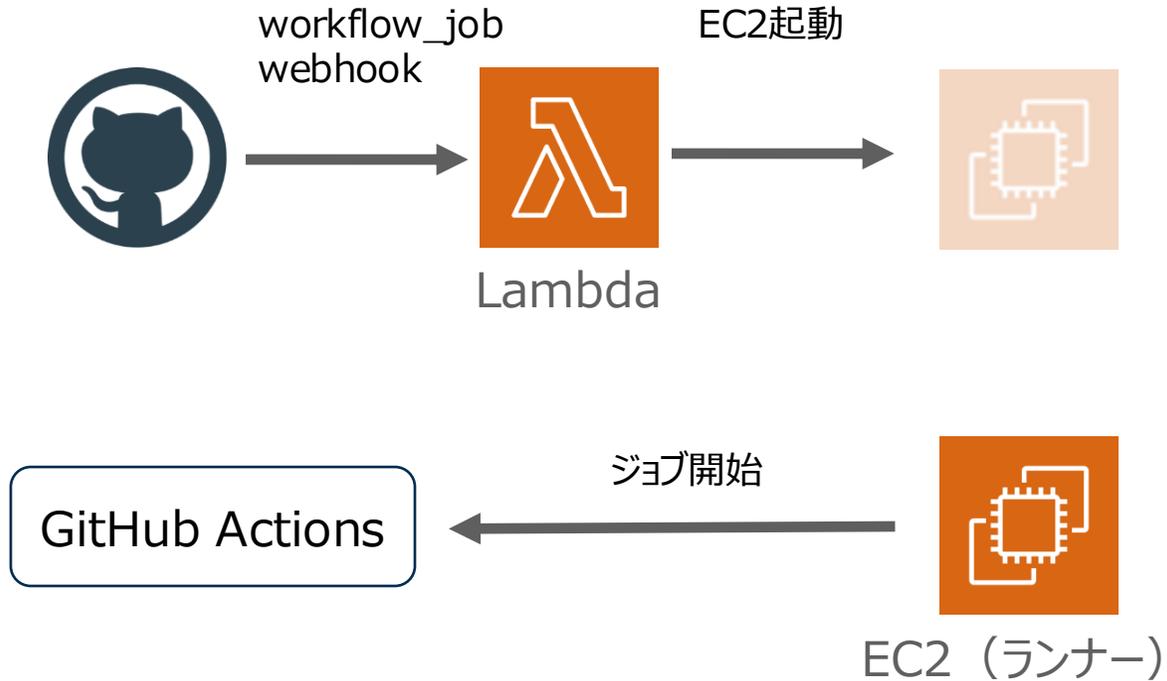


## Cybozuのセルフホストランナー基盤のアーキテクチャ

- OSSを利用 [philips-labs/terraform-aws-github-runner](https://github.com/philips-labs/terraform-aws-github-runner)
- ジョブごとに飛ぶwebhookでLambdaからEC2を起動する
- あらかじめEC2を起動しておくランナープール機能もあり（後述）
- Repo or Orgランナーしか作れない（**×** Enterpriseランナー）
  - Orgごとにランナーが分離されてしまうのでリソース集約効率が悪い

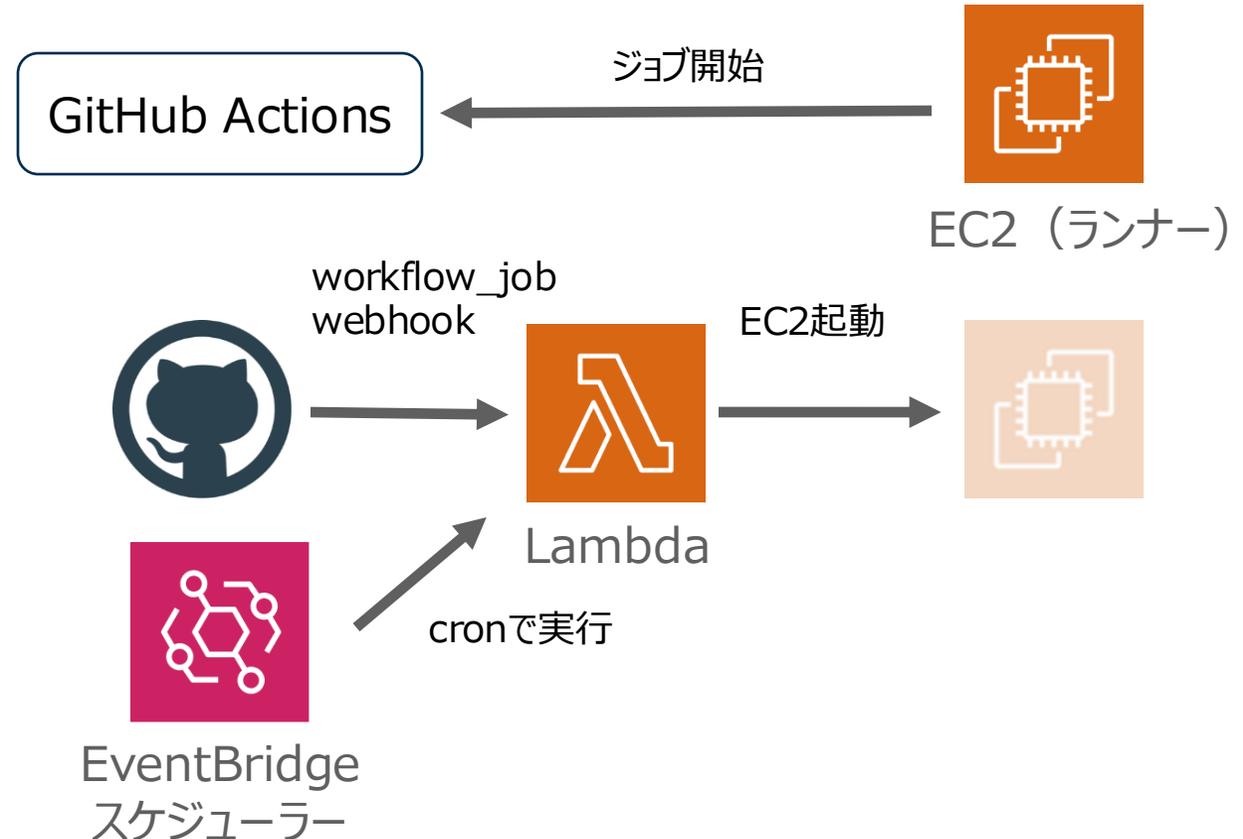
# ランナープール = 待機状態のランナーを用意しておく方式

ランナープールなしの構成



時系列

ランナープールありの構成



# Cybozuでのランナープール

- Orgごとに最低1台、最大は現在のところ60台
- 高スペックランナーのプールはE2Eを頻繁に動かすOrgのみ
- 固定費的な部分なので地道に無駄を削った
  - 非アクティブなOrgのプール数を削減
  - 高スペックランナーを夜間・休日にスケールインしていなかったミス修正
    - Terraform定義の凡ミス
    - 高スペックランナーをプールしているOrgは僅かだったので見逃していた

# スポットインスタンスの費用削減

## スポットインスタンス復習

- 圧倒的に安い（最大90%とAWS公式は言ってる）
- 起動中に中断される可能性がある
- スポットプールに複数のインスタンスタイプを混ぜることが可能（m5.large, m6i.largeなど）
- 基本的にスポット価格が安く、かつ在庫が余っているインスタンスタイプが起動される
  - AWS推奨のprice-capacity-optimized戦略の場合

# Cybozuのランナーでのスポットインスタンスの利用

- 初期から全てのランナーがスポットインスタンス
  - スポット中断=ジョブの謎のfail は許容してもらっている
- 中断対策としてインスタンスタイプを増やしていた（2023/07）
  - 初期はm5, m6iのみ
  - m5d, m5n, m6id, m6inを追加
    - n: ネットワークが強化されたバリエーション
    - d: ローカルSSD付き

# スポットインスタンスの種類と価格の変動

## ■ AWSのスポットリクエストから確認可能



EC2 > スポットリクエスト

スポットリクエスト

🔍 リクエストを検索 < 1 > ⚙️

リクエ...	リクエスト...	インスタ...	状態	容量	ステ...	永続性	作成日
--------	----------	---------	----	----	-------	-----	-----

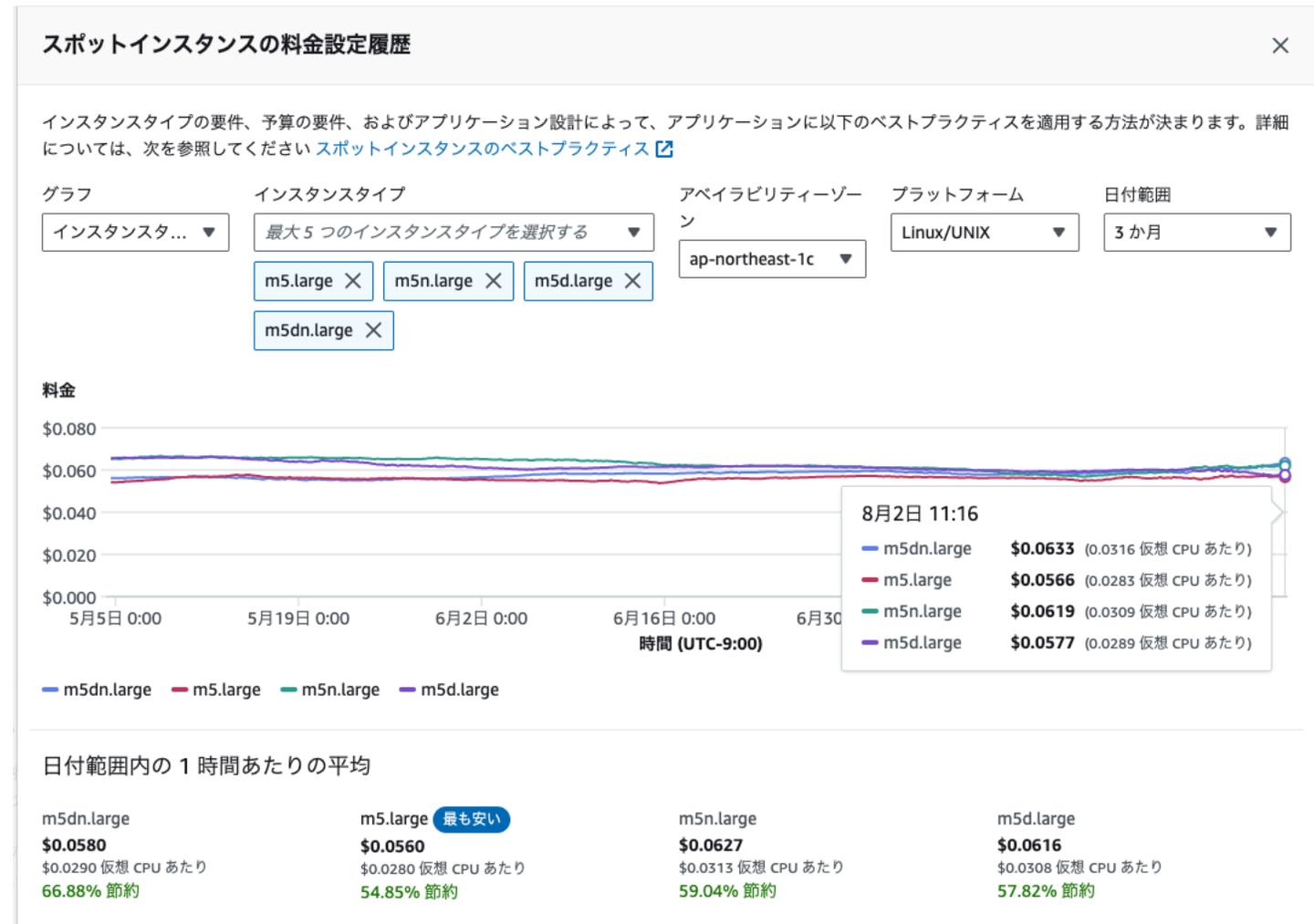
現在、このリージョンにはリソースがありません。

EC2 スポットインスタンスを初めて使用する場合はこちら。 [「開始方法」ページにアクセスしてください](#)

[Request Spot Instances] ボタンをクリックして、スポットインスタンスを起動します。

# スポットインスタンスの種類と価格の変動（m5系バリエーション）

- m5に対してm5n, m5d, m5dnなどは大体順当に高い
- nやdの付加価値はセルフホストランナーとしてはメリットがないので割高

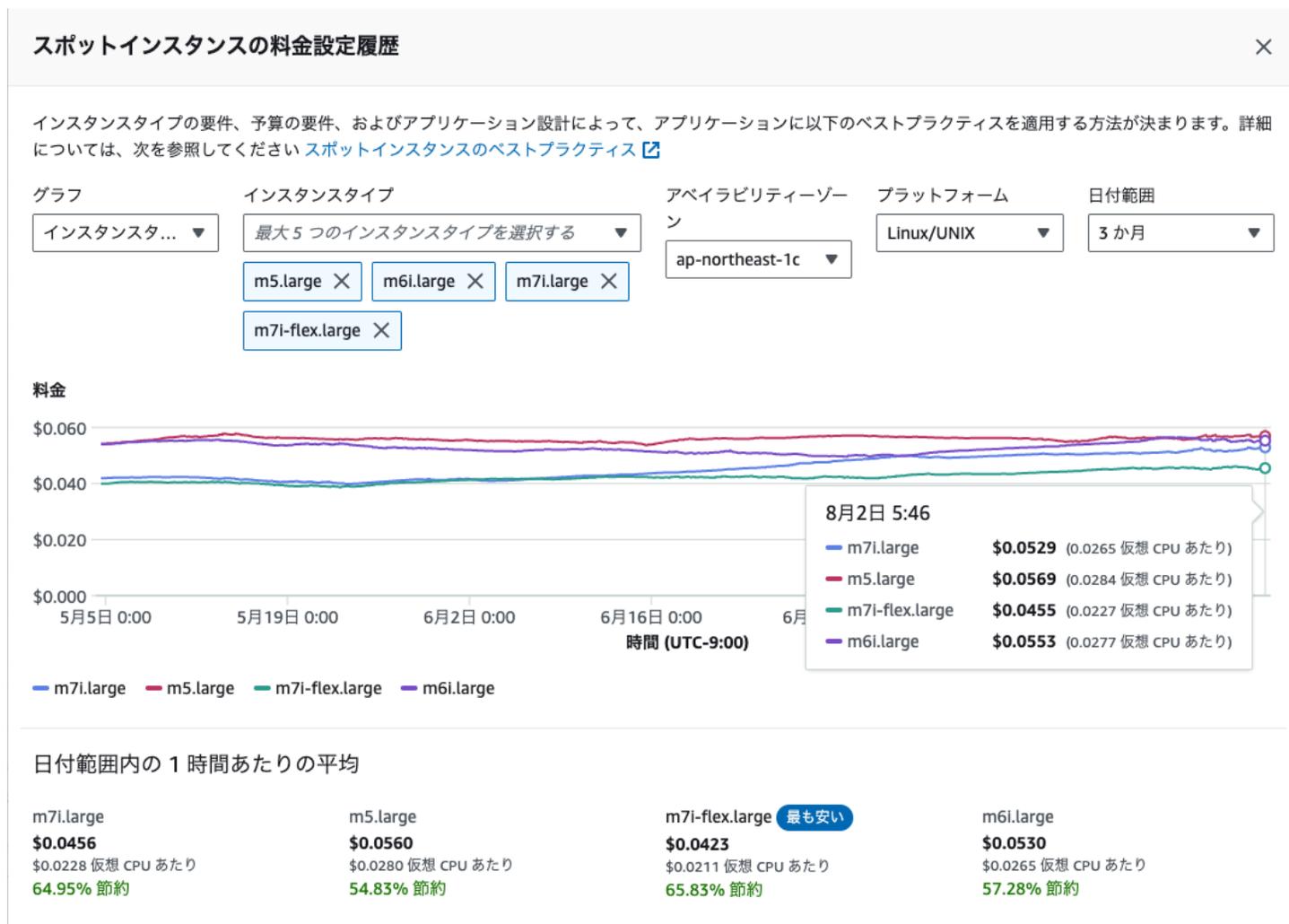


## EC2の費用について考えていたある日

- 社内の別チームがm7iのスポットインスタンスを利用していることを知る
- 最新のm7は高そうだが、m5nやm5dと入れ替えることでスポット中断率を維持できるならば高スペックな分お得なのでは？ 🙄
- ついでに前から気になっていたAMD系のインスタンスも調べてみるか

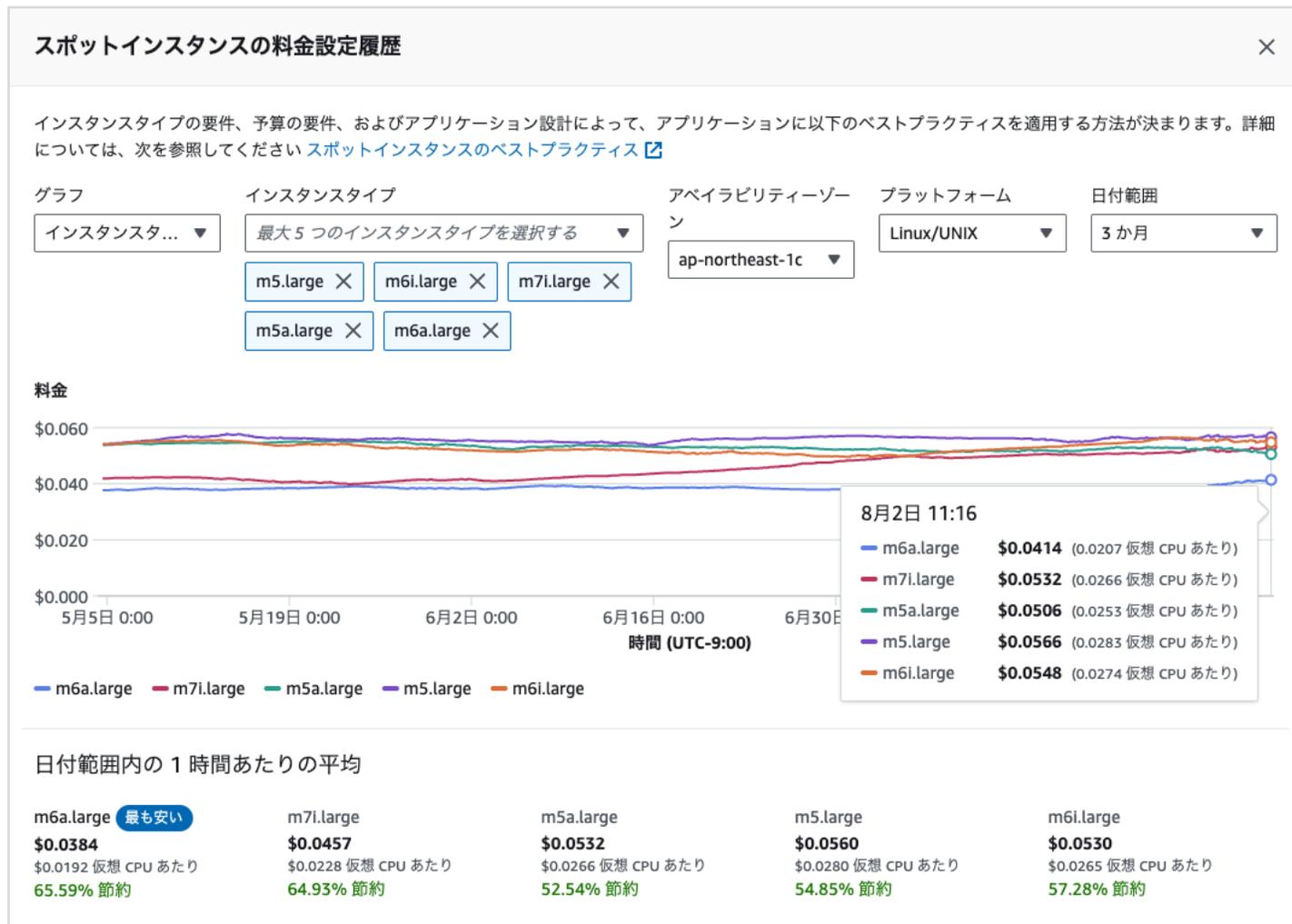
# スポットインスタンスの種類と価格の変動（世代別）

- スペックと価格が比例していないことがある
- オンデマンド料金は  $m5 = m6i < m7i\text{-flex} < m7$
- 8/2時点では  $m7i\text{-flex}$ が最安値  $m5$ が最高値



# スポットインスタンスの種類と価格の変動（世代別+AMD系）

- さらにAMD系のm5a, m6aを追加すると・・・
- m6a安すぎでは（褒め言葉）



## スポットプールの調整

- スポットプールに最新世代のm7系を追加
  - m7i, m7aを追加
    - インスタンスガチャのSSR
    - m7i-flexは調べても性能特徴がよく分からなかったのでスキップ
  - m5a, m6a, m7aのAMD系x86 CPUを追加
    - Intel系と同じx86なので互換性はあるはずだが心配はしていた
    - AMD系由来の問題は今のところ社内から聞いていないので結果オーライ

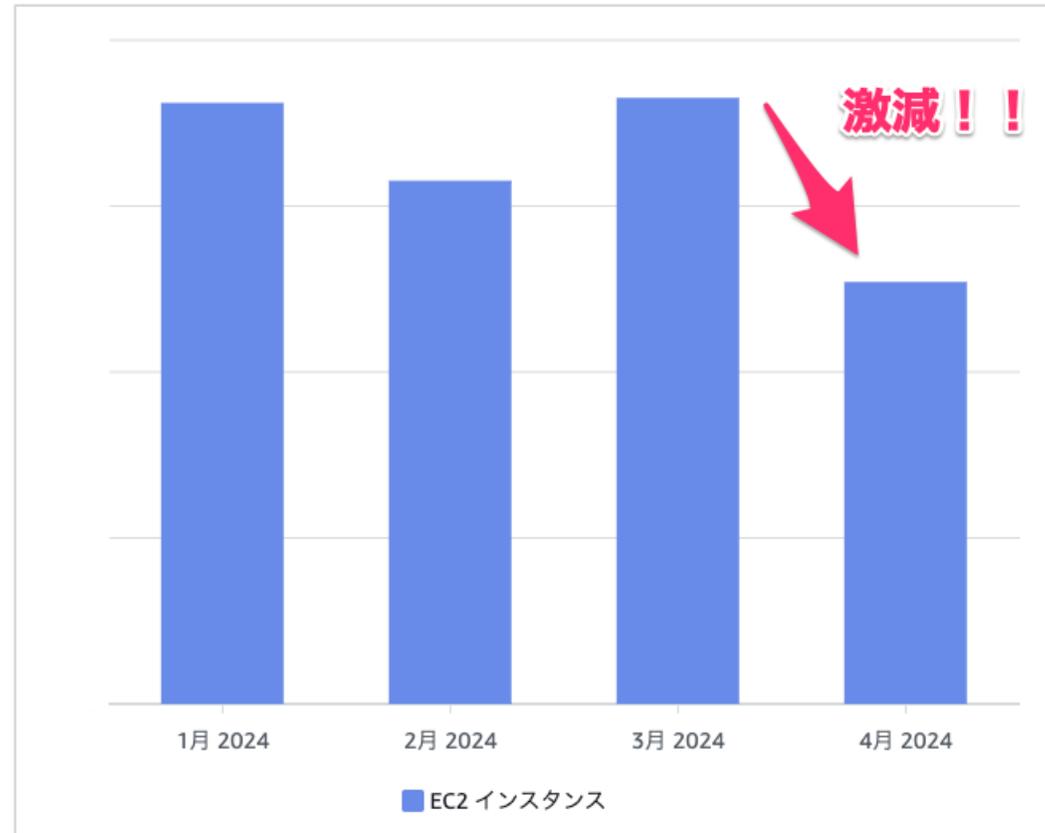
## スポットプールの調整

- 現在のスポットプールのリスト→
- large, xlarge, 2xlargeの  
3種類のスポットプールを用意
  - = 3種類のスペックのセルフホストランナーを提供

```
instance_types = {  
  large = [  
    "m5.large",  
    "m5d.large",  
    "m5a.large",  
    "m5ad.large",  
    "m5n.large",  
    "m5dn.large",  
    "m5zn.large",  
    "m6i.large",  
    "m6a.large",  
    "m6id.large",  
    "m6in.large",  
    "m6idn.large",  
    "m7i.large",  
    "m7a.large",  
  ]  
}
```

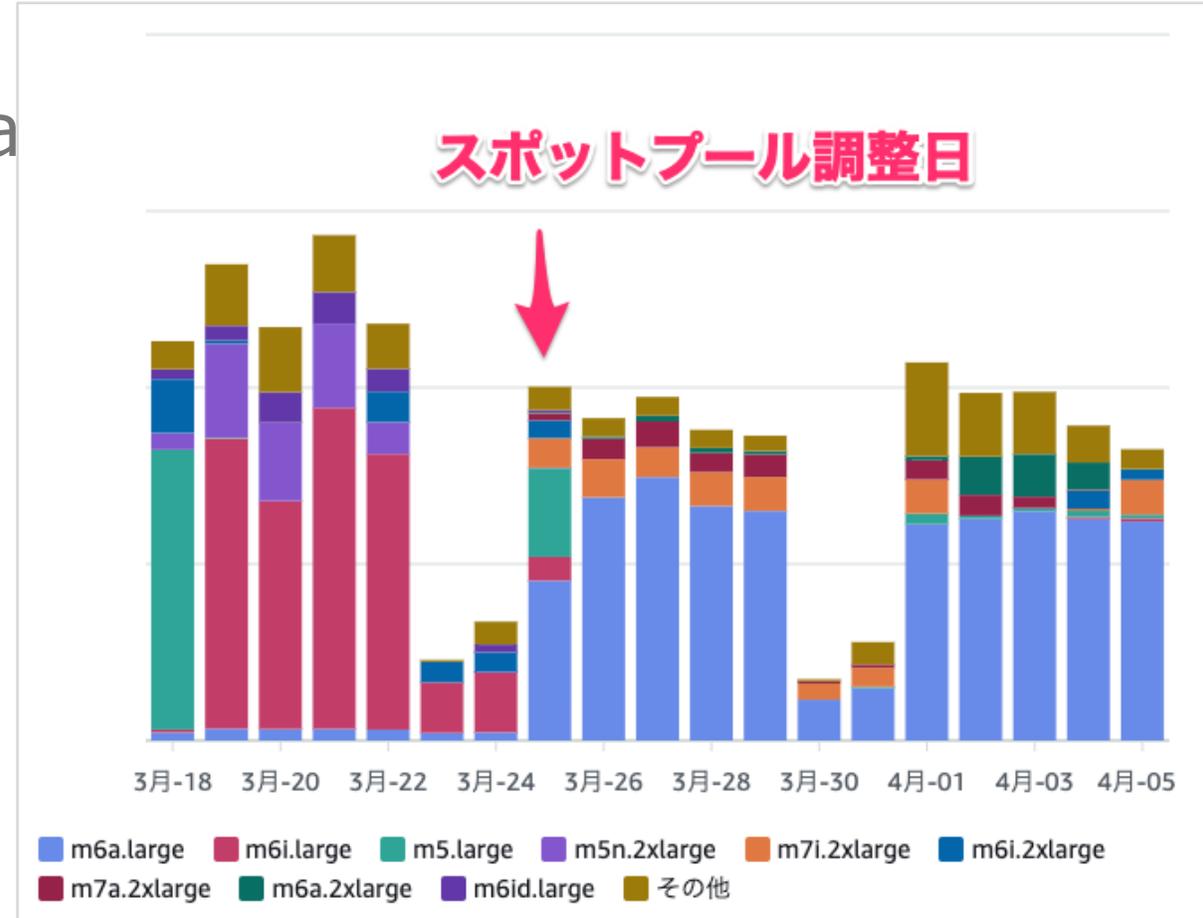
# スポットインスタンスの費用削減

- スポットプールに追加してみた結果・・・EC2費用が激減！



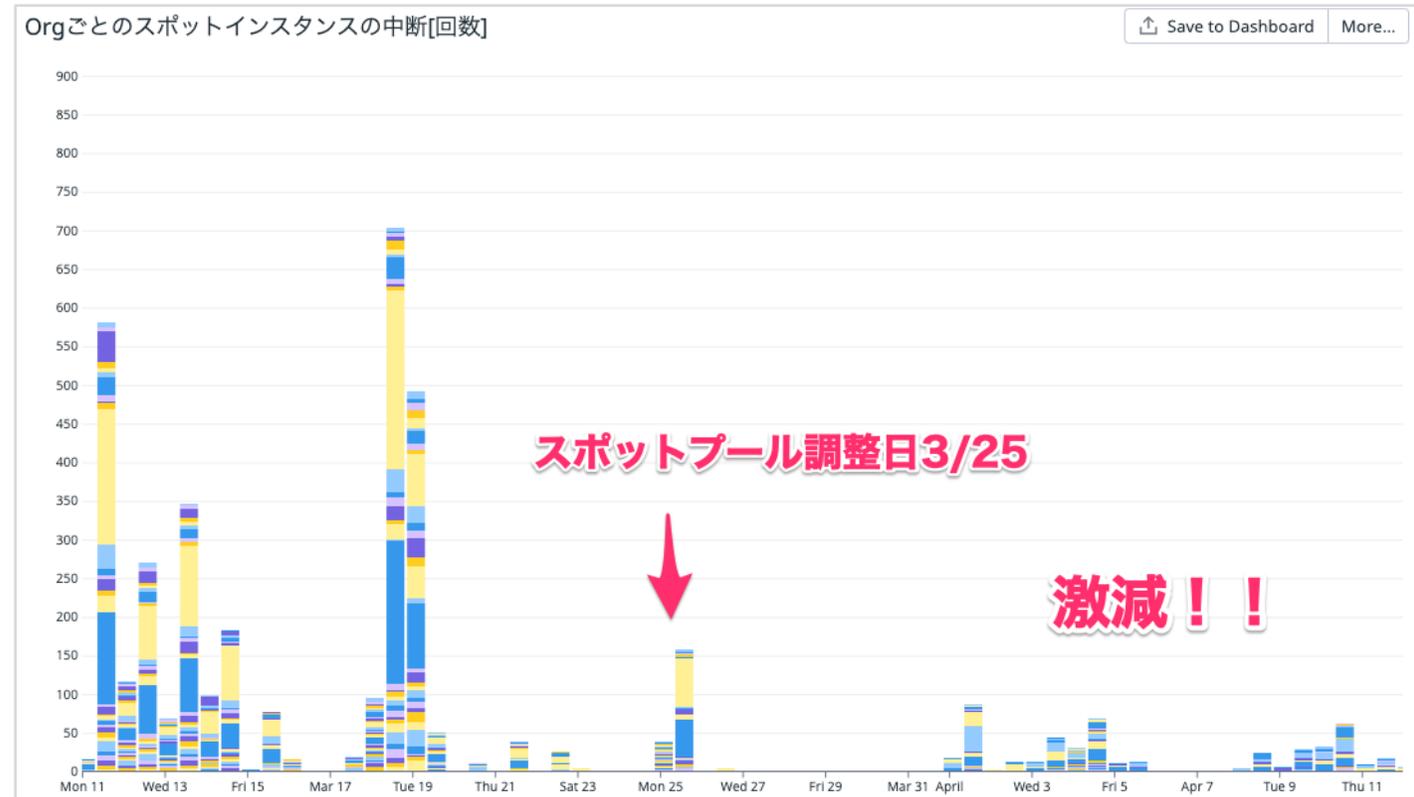
# スポットインスタンスの費用削減

- インスタンスの内訳を見るとほぼm6a切り替わっている
- AWS内で余ってる説・・・？



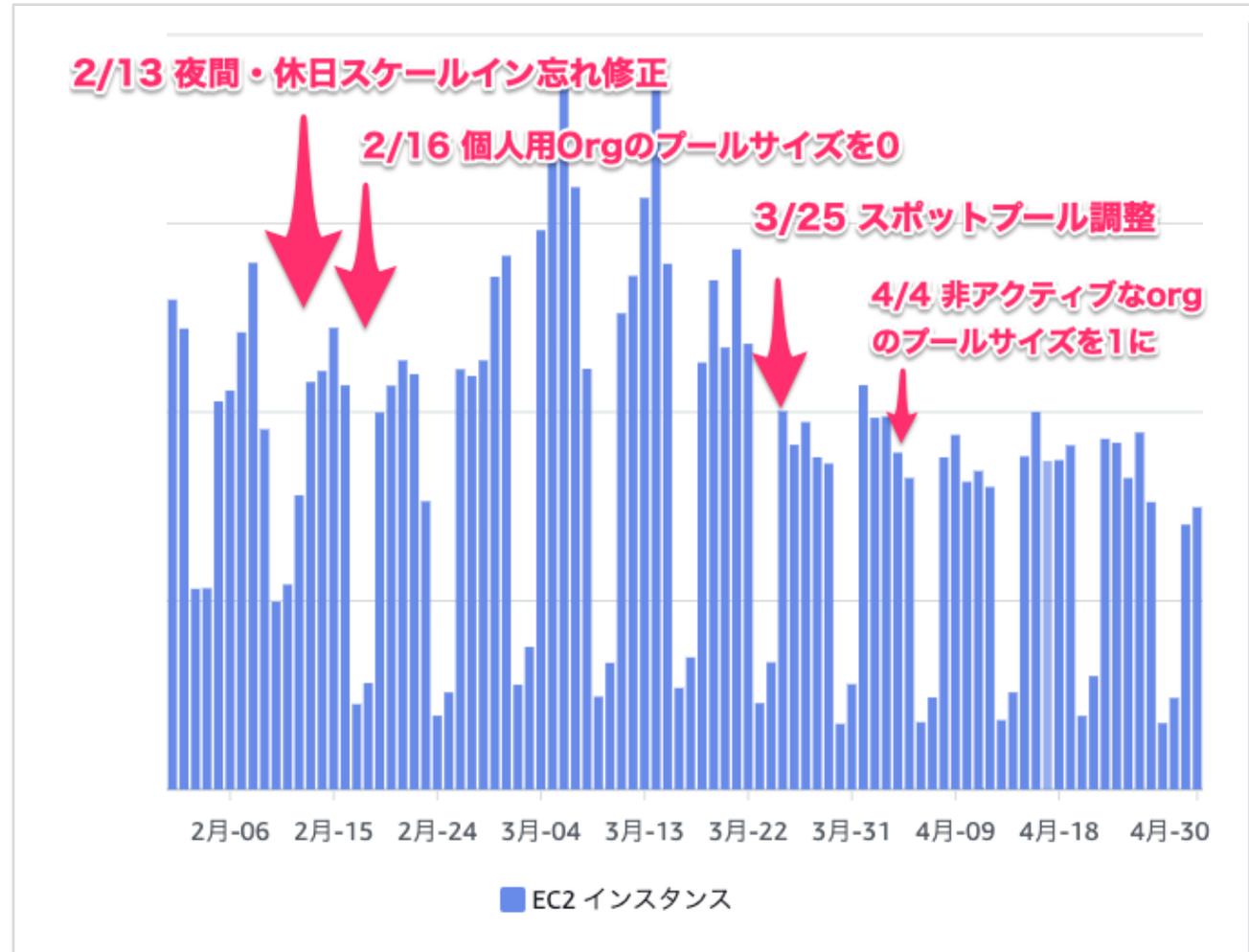
# スポット中断も改善

- 変更後に中断回数が激減
- 以前からジョブが謎に中断される話は聞いていたが、予想以上に中断されてた 😨
- ついでに改善されて社内のユーザー体験も向上



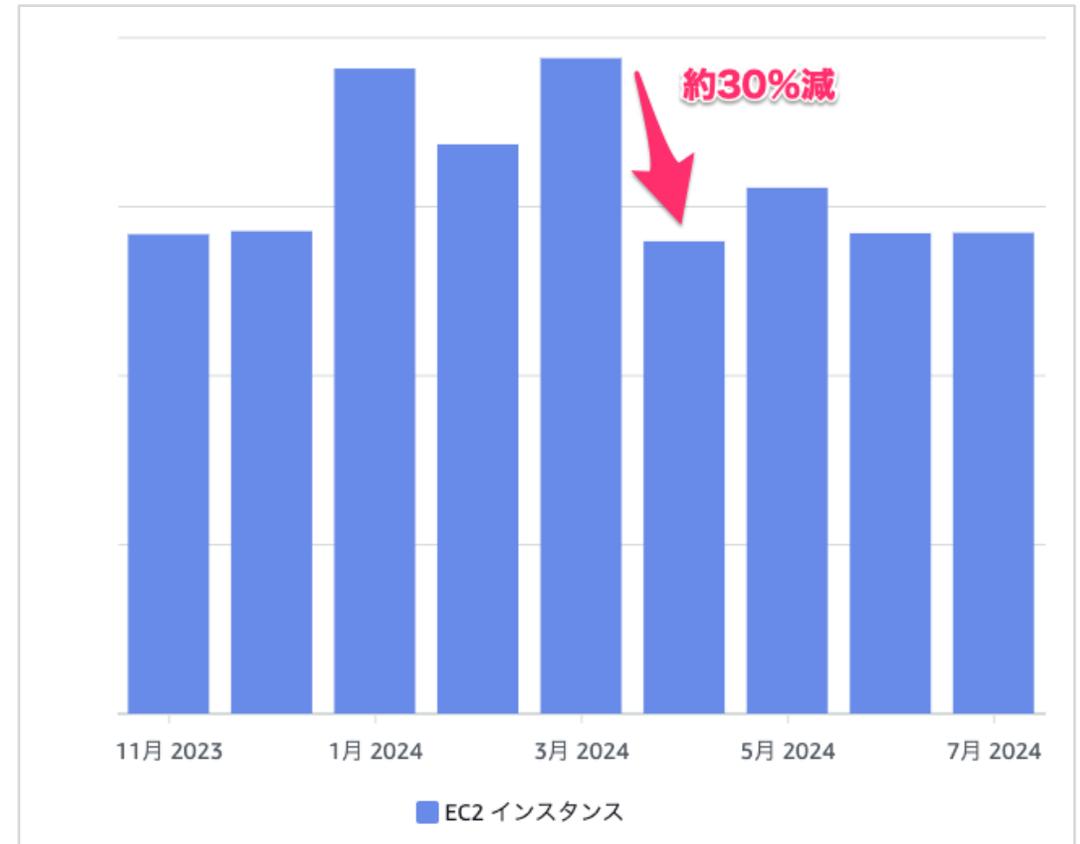
# 最終的な費用削減の効果測定

- 対策ごとに削減効果が出ている
- スポットの調整が圧倒的成果



## 最終的な費用削減の効果測定

- EC2費用が3月->4月で30%削減
- 11月水準まで戻り、横ばいを維持





今後

# 今後：NAT GWの料金を削減したい

- EC2その他（NAT GWのネットワークの費用）は高止まりしたまま
- 3つの対策を予定
  - **VPCフローログの分析**
    - どこへの通信が多いのかを計測可能にする
    - AthenaからS3にクエリする際のパーティション設定とBIツールの選定で苦戦中
  - **Dockerのミラーレジストリ構築とpull-throughキャッシュ**
    - 先行調査でDocker Hubへの通信が多いことが判明している
    - pullしたイメージを自動キャッシュするレジストリサーバーをVPC内に構築し、NAT GWの通信を減らす予定
  - **IPv6 + Egress-Only Internet Gateway**
    - IPv6であればNAT GWを通らずに外向きの通信が可能らしい
    - Docker Hub以外への通信も減らせるのではと期待している
    - 参考：AWS Summit Japan 2024 [“IPv6 on AWS ～Public IPv4 アドレス削減に向けてできることできないこと～（AWS-20）”](#)

## まとめ

- Cybozuのセルフホストランナーの近況紹介
- 最近ではAWS費用削減に取り込み中
  - 無駄にプールしているOrganizationのランナー数を削減
  - スポットプールにm7世代、AMD系CPUを追加
- 主にスポットプール調整の効果でEC2費用が30%削減
- 今後はNAT GW周りの費用改善を予定