



ANTI-PATTERN INC.

F-1

**AWS Proton を使って(もらって)
快適な開発環境をあげよう(もらおう)！**



DEV DAY

ONLINE JAPAN

SEPTEMBER 28-30, 2021

2021/09/29

株式会社アンチパターン 矢ヶ崎哲宏

矢ヶ崎 哲宏(Akihiro YAGASAKI)
株式会社アンチパターン 取締役 CTO兼COO

役割：日本のソフトウェアエンジニアを憧れの職業にするためのいろいろ

経歴：アマゾン ウェブサービス ジャパン にて
シニアパートナーソリューションアーキテクト
Webメディア/SaaSベンダーにて技術責任者ボードメンバー
大手SIerグループ会社にて情シス責任者
アニメソングのコーラス など

好きな AWS サービス： Amazon Simple Queue Service(SQS)
※ AWS Proton とはこれから仲良くなっていこうと思います。



1975年生まれ





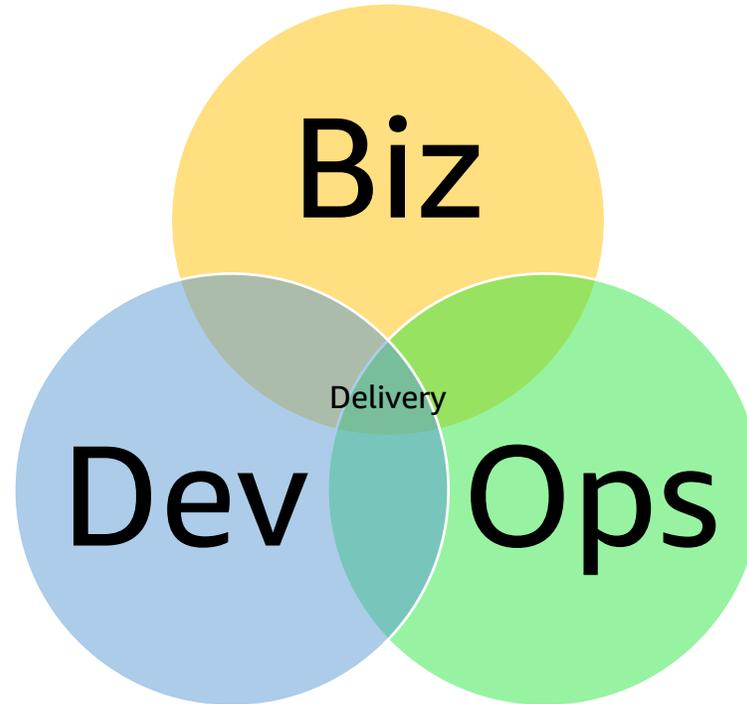
本日のおしながき



ANTI-PATTERN INC.

- 開発と運用・アプリとインフラの各チームの関係性についての所感
- AWS Proton についての所感
- デモ
- まとめ

DevOps っていいですね！



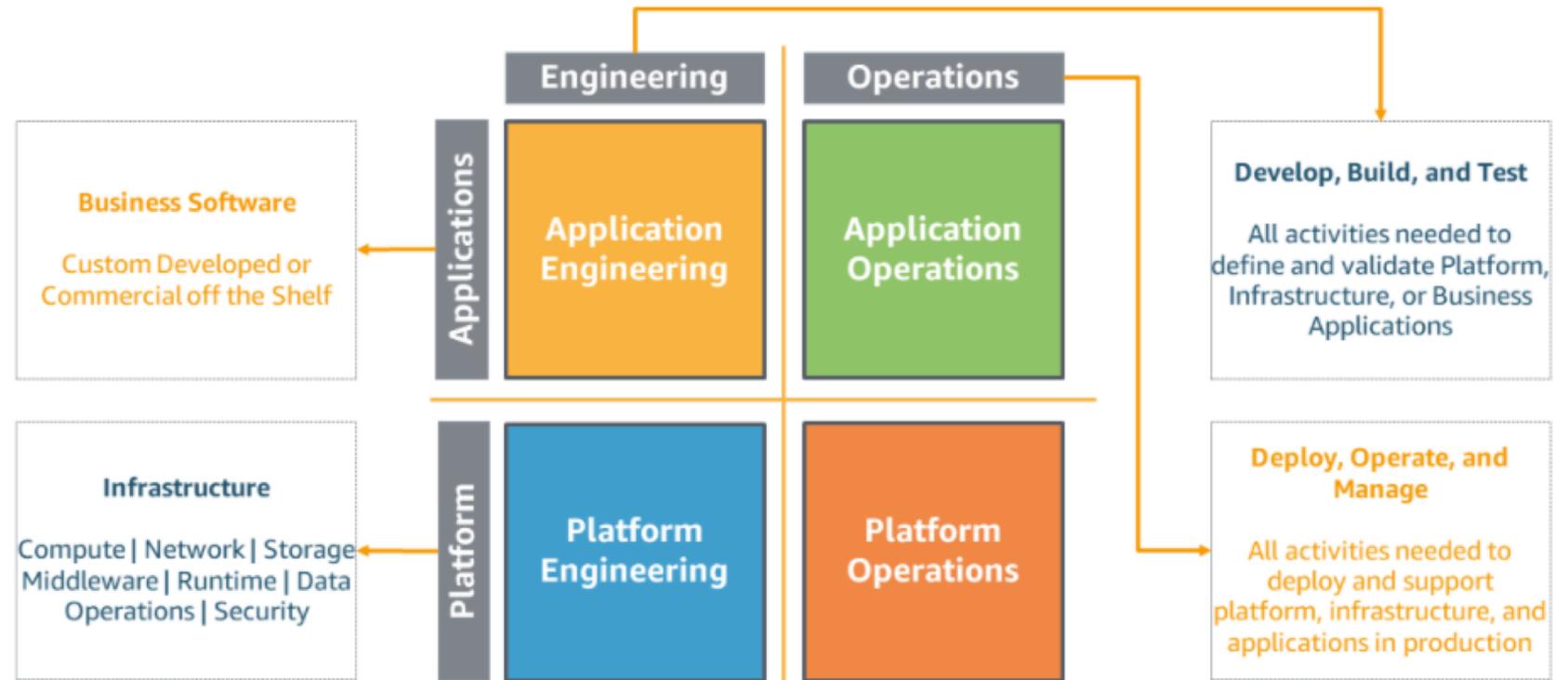
開発と運用の垣根を無くして、アジリティをあげていきましょう！
そのために、文化・手法・ツールを最適なものにしていきましょう！
これができるとよいですね。

特にここでは、Biz(経営と事業)、Dev(開発)、Ops(運用)が合わさって、適切なサービス提供ができるということを BizDevOps として表しています



完全に分離された運用モデル

システムの開発・運用を考えると、DevとOpsという分け方の他にも、他の観点での分け方もあります。AWS Well-Architected フレームワークでは、例えば右の図のように 2x2 に分離して考えられています。縦軸が開発か運用か、横軸がアプリかインフラか、というような感じです。極端な例でいえば、各担当のチームがわかれて 4 種類のチームが出てきます (意識も含んでいます)

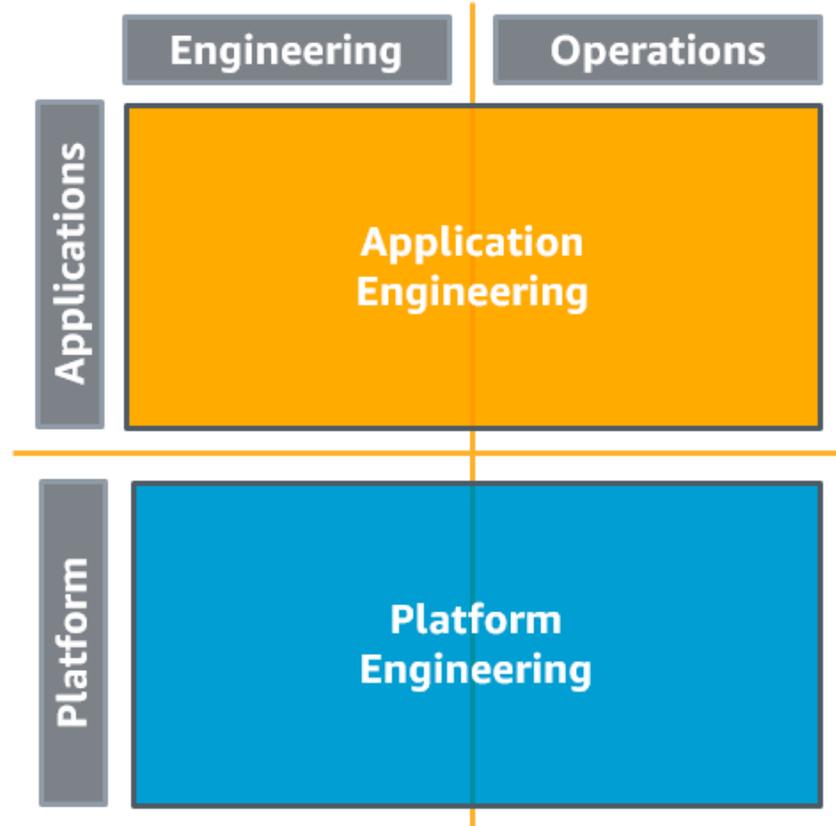


AWS Well-Architected フレームワーク運用上の優秀性の柱 より

https://docs.aws.amazon.com/ja_jp/wellarchitected/latest/operational-excellence-pillar/fully-separated-operating-model.html

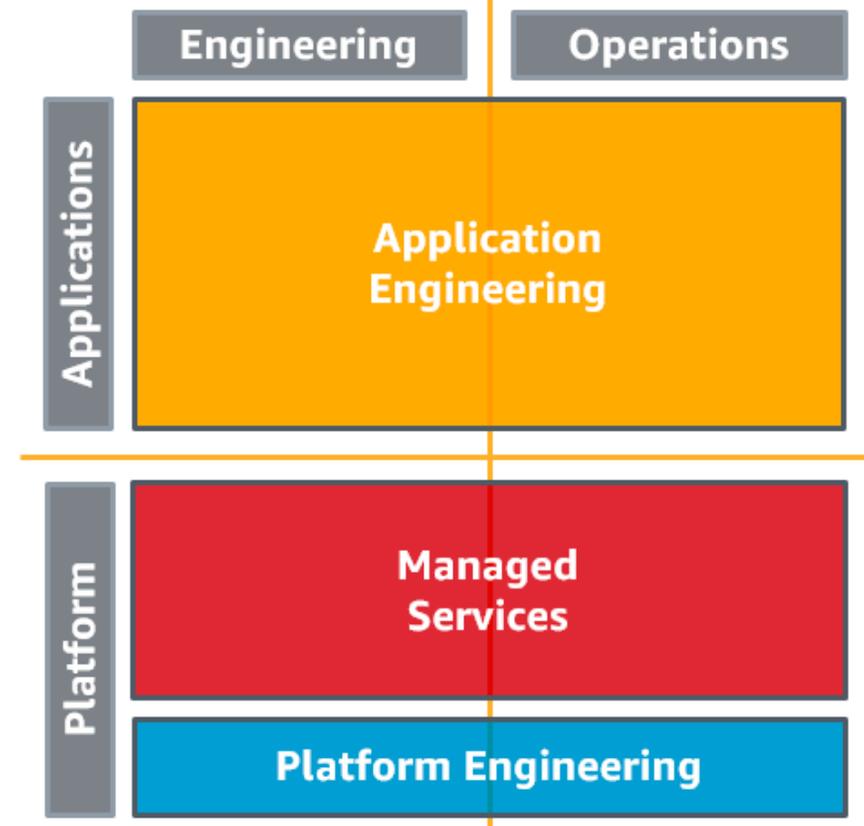


分離されたアプリケーションのエンジニアリングと運用 (AEO)
および一元化されたガバナンスを備えたインフラストラクチャ
のエンジニアリングと運用 (IEO)



https://docs.aws.amazon.com/ja_ip/wellarchitected/latest/operational-excellence-pillar/separated-aeo-and-ieo-with-centralized-governance.html

分離された AEO および一元化されたガバナンスと
サービスプロバイダを備えた IEO

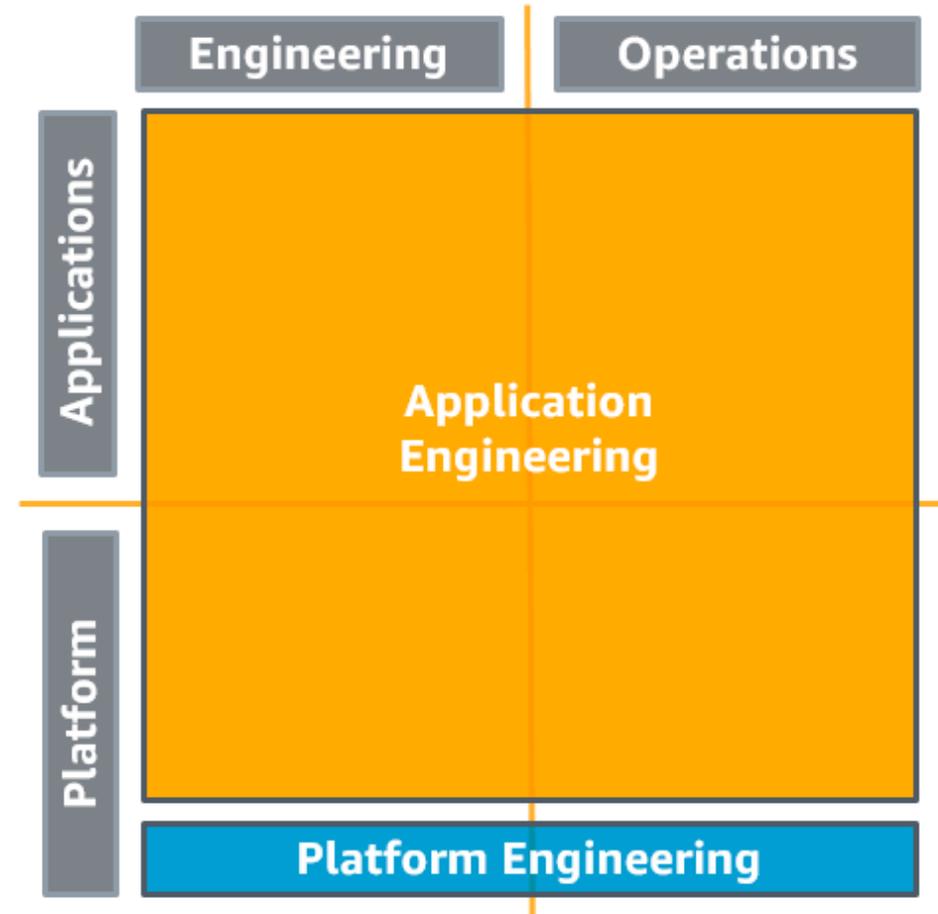


https://docs.aws.amazon.com/ja_ip/wellarchitected/latest/operational-excellence-pillar/separated-aeo-and-ieo-with-centralized-governance-and-a-service-provider.html

AWS Well-Architected フレームワーク運用上の優秀性の柱 より

分離された AEO と一元化されていない ガバナンスを備えた IEO

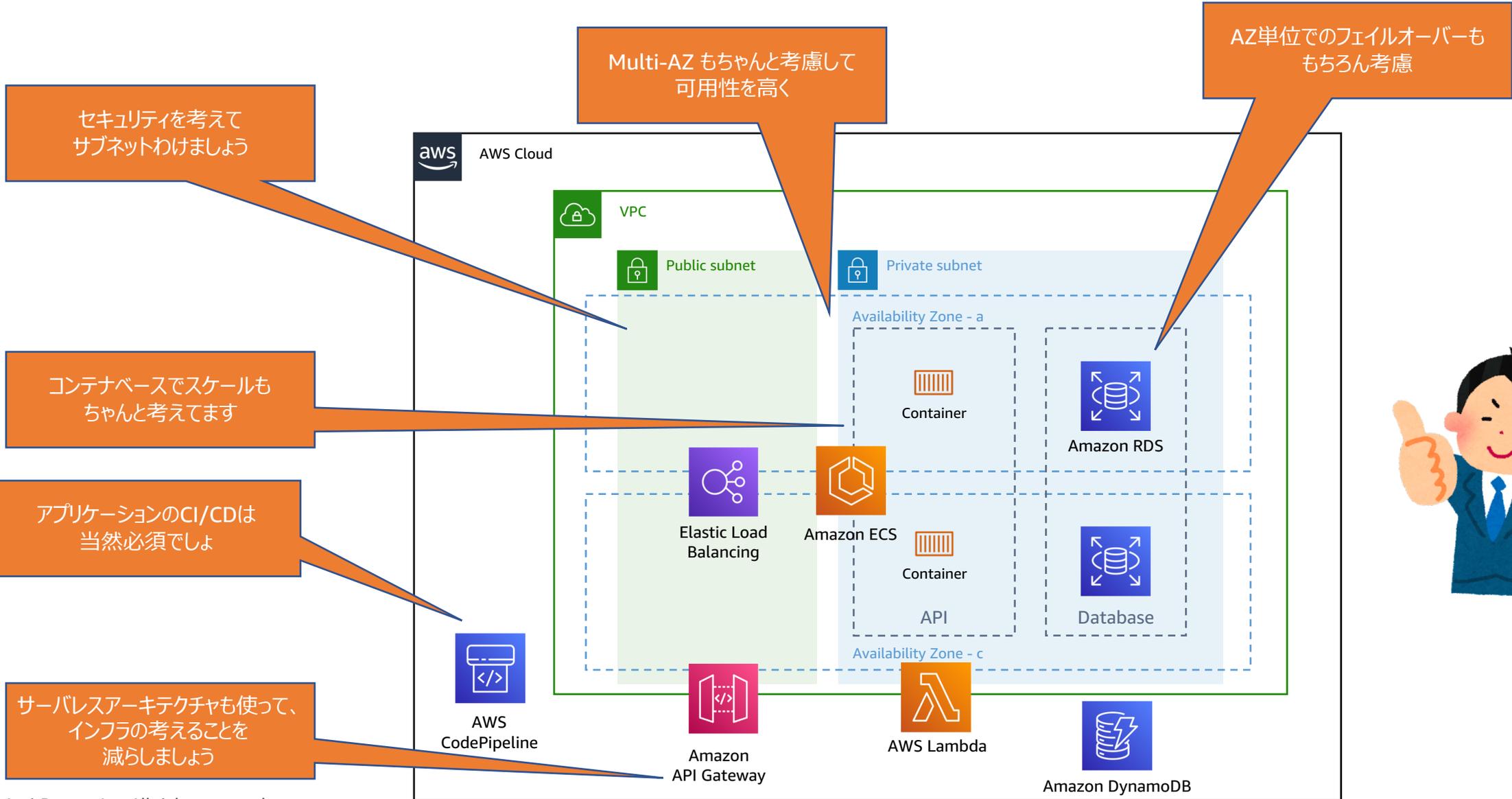
アプリケーションのアーキテクチャは、インフラのアーキテクチャに深く関わってきます。そのため、アーキテクチャのことを中心に考えるとすれば、アプリケーションを作る人がインフラのことも考えるのが、整合性的にも効率的にもアジリティ的にもよいのではないのでしょうか



https://docs.aws.amazon.com/ja_jp/wellarchitected/latest/operational-excellence-pillar/separated-aeo-and-ieo-with-decentralized-governance.html

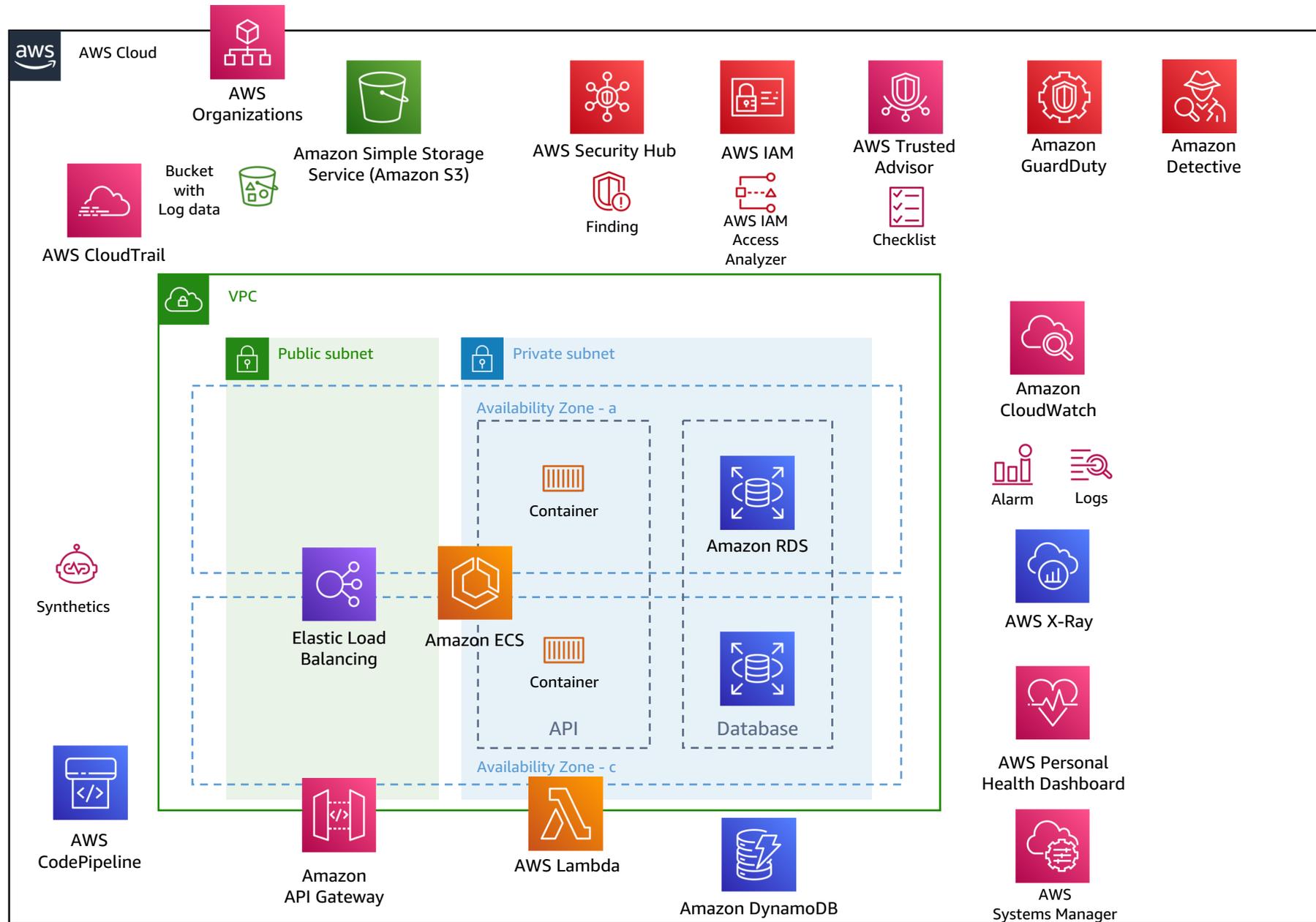
AWS Well-Architected フレームワーク運用上の優秀性の柱 より

デベロッパー的な視点から考えるアーキテクチャ例



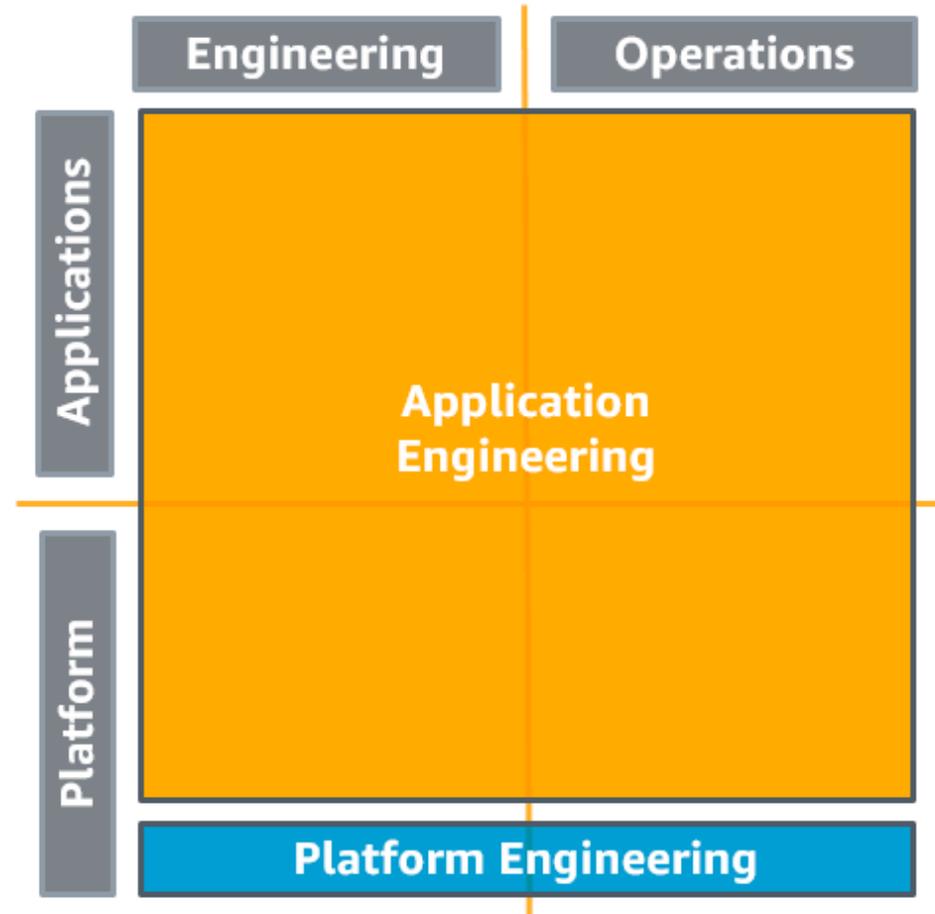
実際の本番環境例

本番環境においては、
機能・性能・可用性のことだけ
ではなく、
運用・セキュリティ・ガバナンス・
マネージメントなど
多くのことを考えてインフラを
構築する必要がある



アプリケーションエンジニアが、本番環境の運用のことまで考えないといけない？
セキュリティや、さらにシステムをまたいだ組織単位でのガバナンスコントロールやマネージメントまでも？

それとも、プラットフォームチームが、AWS Organizations や AWS Control Tower を使ってアカウントの払い出し、ランディングゾーン、Service control policies や CloudFormation StackSets を使ってガバナンスコントロールやマネージメントをする？
それがよさそう！



https://docs.aws.amazon.com/ja_ip/wellarchitected/latest/operational-excellence-pillar/separated-aeo-and-ieo-with-decentralized-governance.html

AWS Well-Architected フレームワーク運用上の優秀性の柱 より

AWS Proton 登場！



AWS Proton

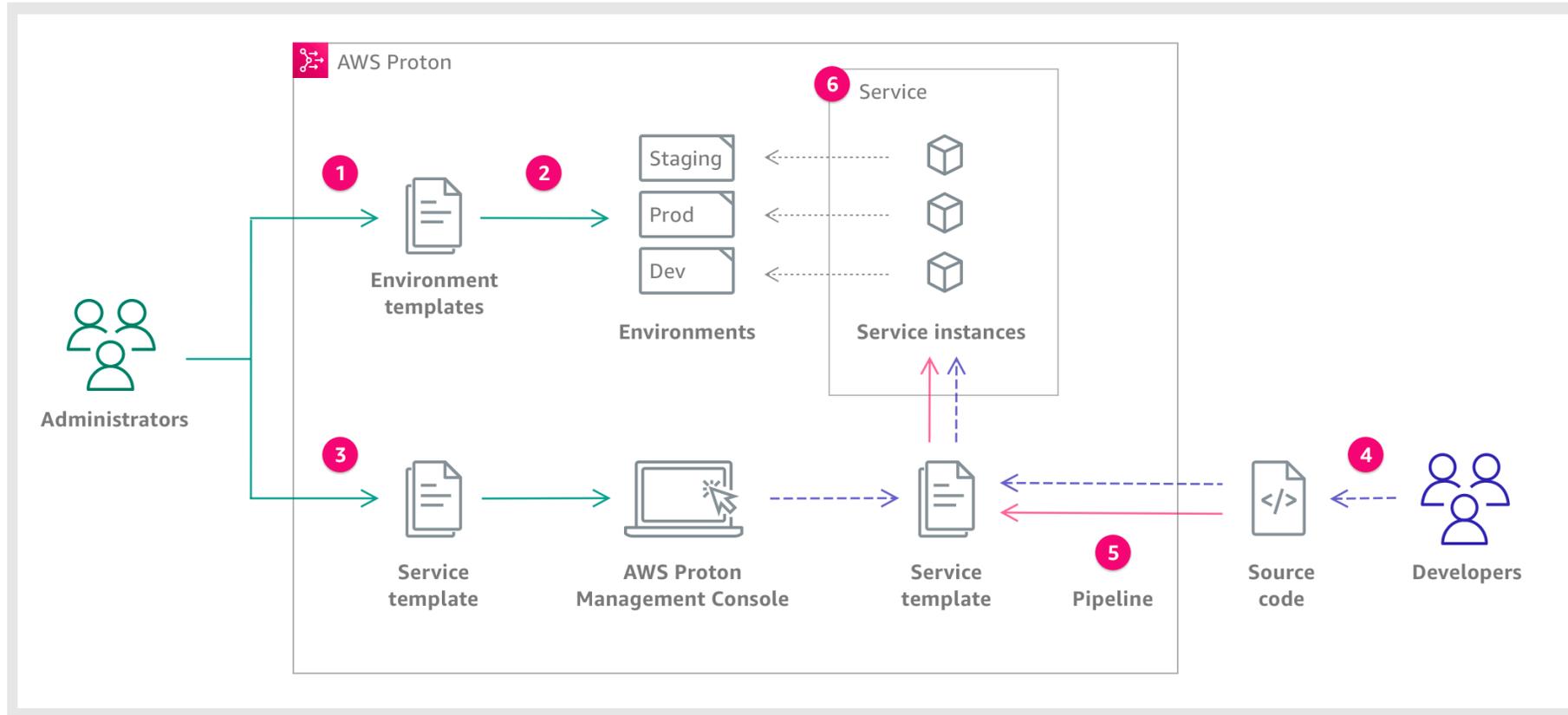
マネジメントとガバナンス

AWS Proton コンテナとサーバーレスデプロ イの自動化

AWS Proton により、プラットフォームチームは、デベロッパーが一貫した標準とベストプラクティスを提供するために必要な管理ツール、ガバナンス、および可視性を使用して、コンテナやサーバーレステクノロジーを使用してコードを簡単にデプロイできるようにします。

ここだけ読むと、「コンテナ・サーバレスのためのサービス」的な意味合いが強く感じますが、今日は「マネジメントとガバナンス」というところを中心にお話します

AWS Proton の登場！

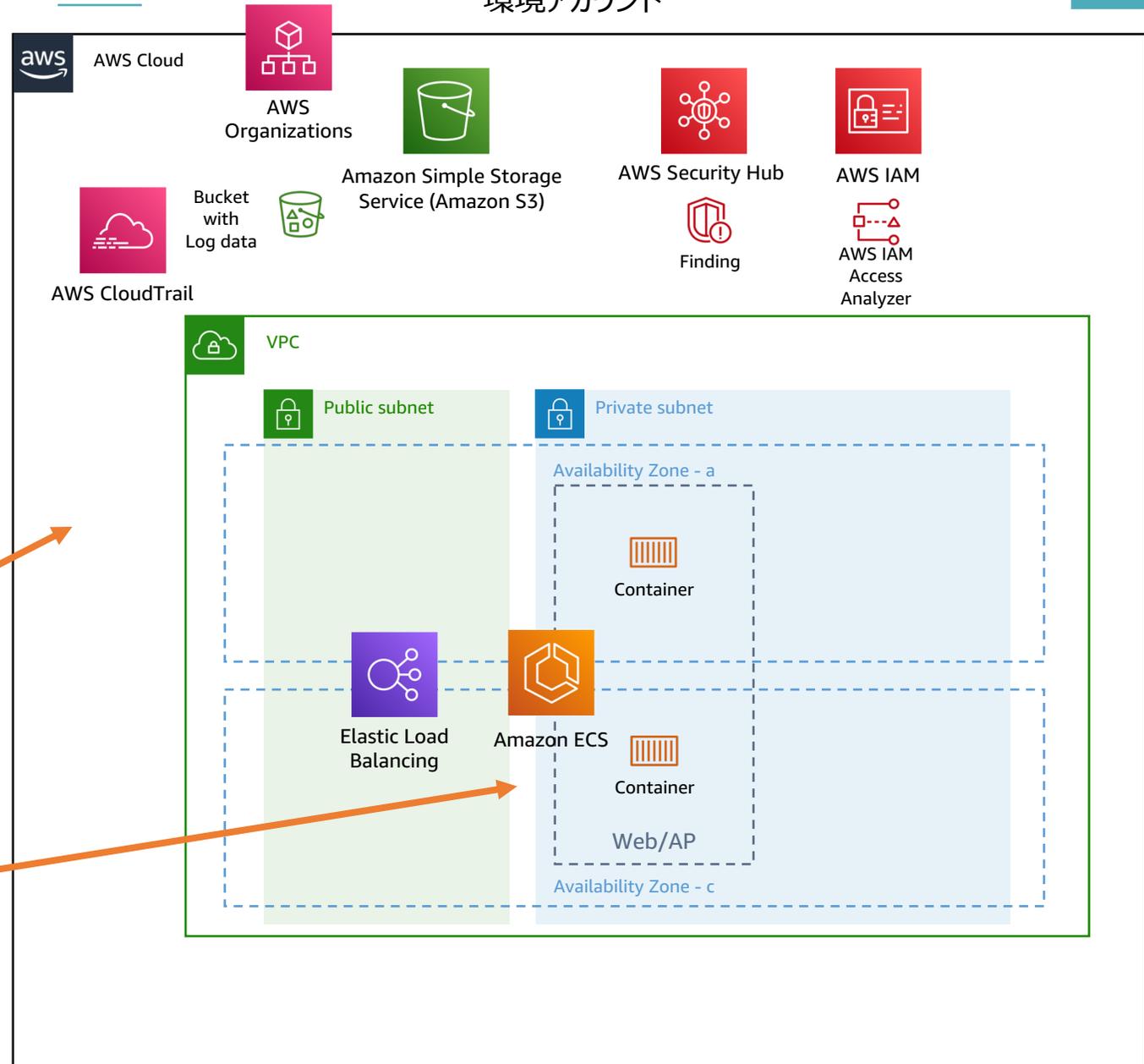


登場人物は2種類。管理者(左)と開発者(右)です。
管理者(インフラのことを考える人)が、CloudFormationのテンプレートを作り、環境(VPCとか)とサービス(インスタンスとかCI/CDとか)を作っておきます。
開発者は、インフラをちょっと意識しつつも、基本的に開発に集中できます。
どの環境のどのバージョンは、どのサービスのどのバージョンで使える。
いま使われているのはどのバージョンのテンプレート。などの管理もできます。

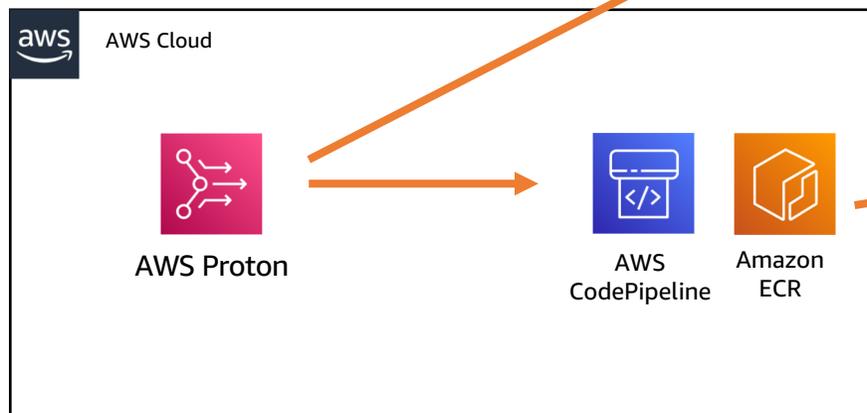
デモの環境

環境アカウント

本日はこのようなものを作ってみます



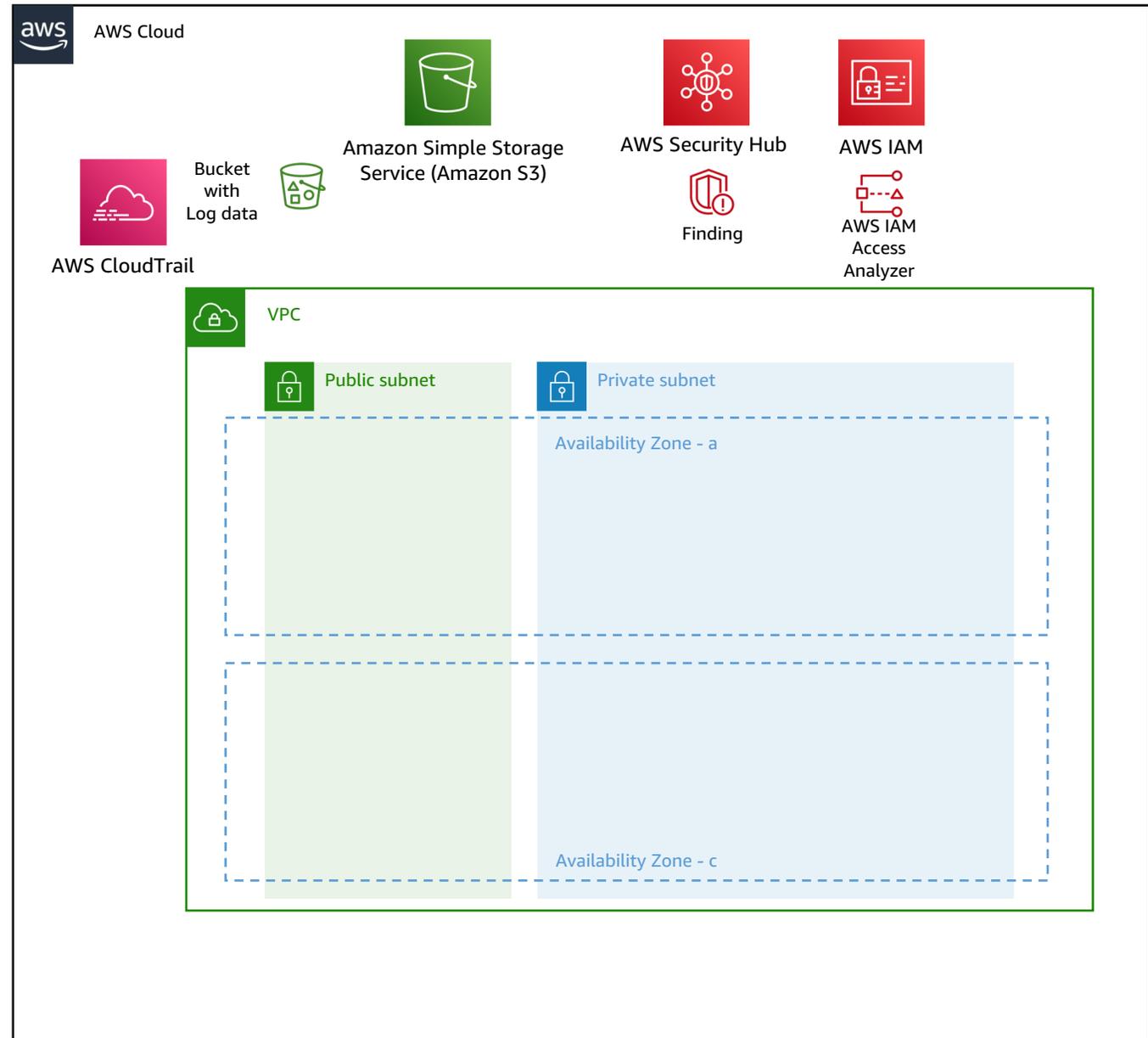
管理アカウント



デモの環境

環境アカウント

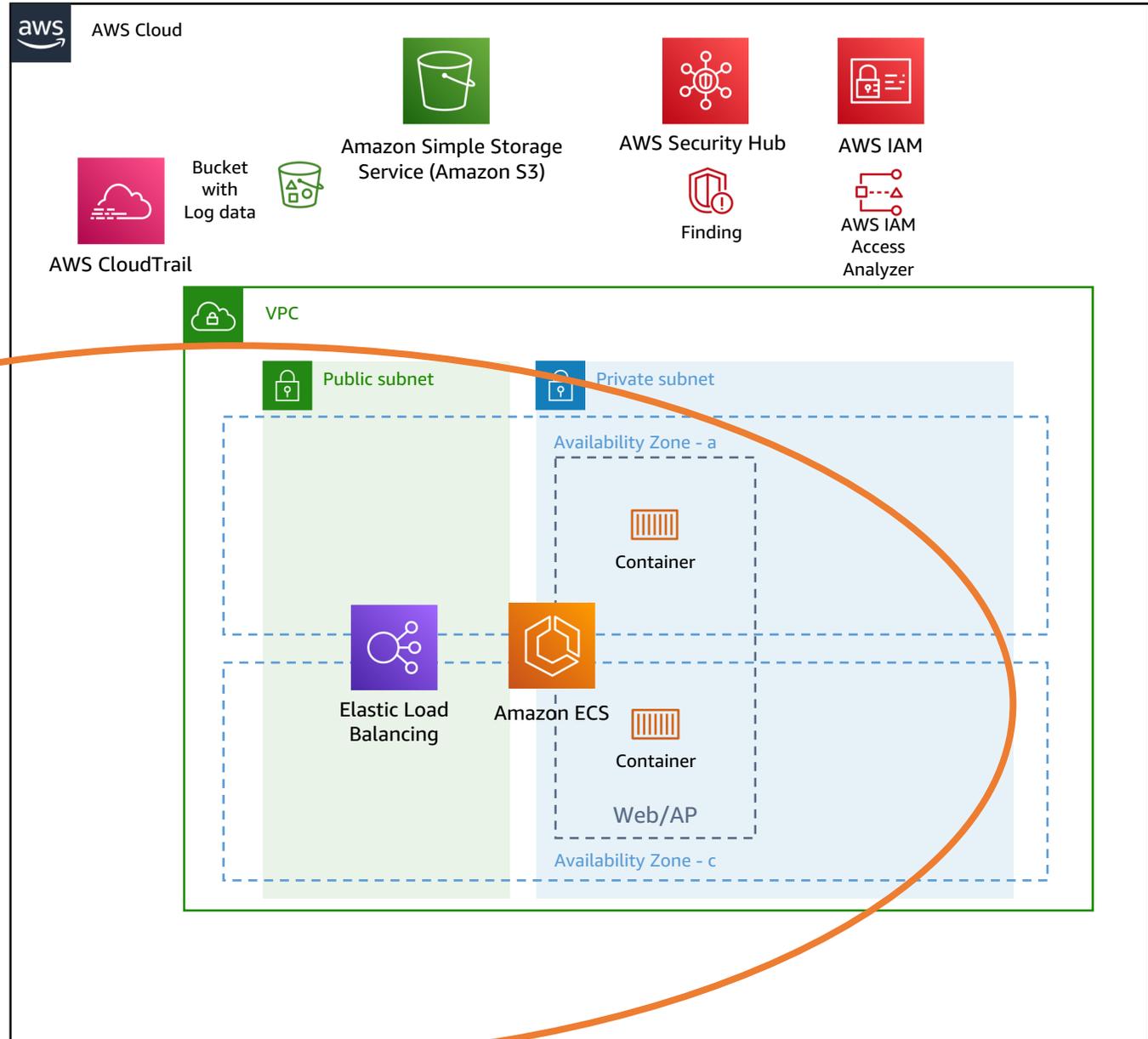
「環境」として作成されるもの



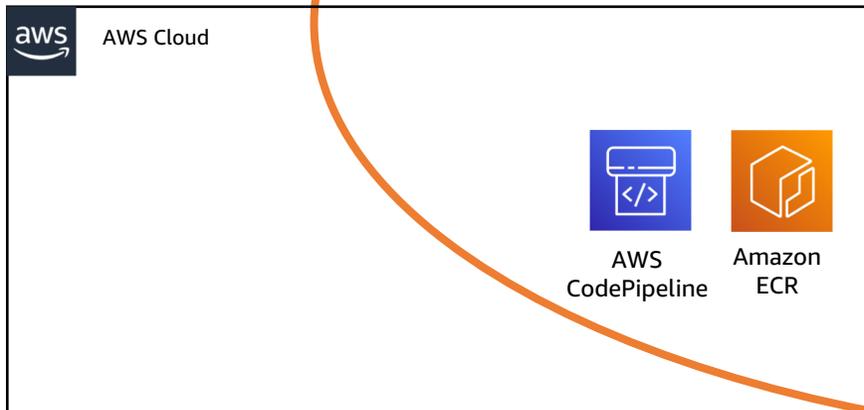
デモの環境

環境アカウント

「サービス」として作成されるもの



管理アカウント

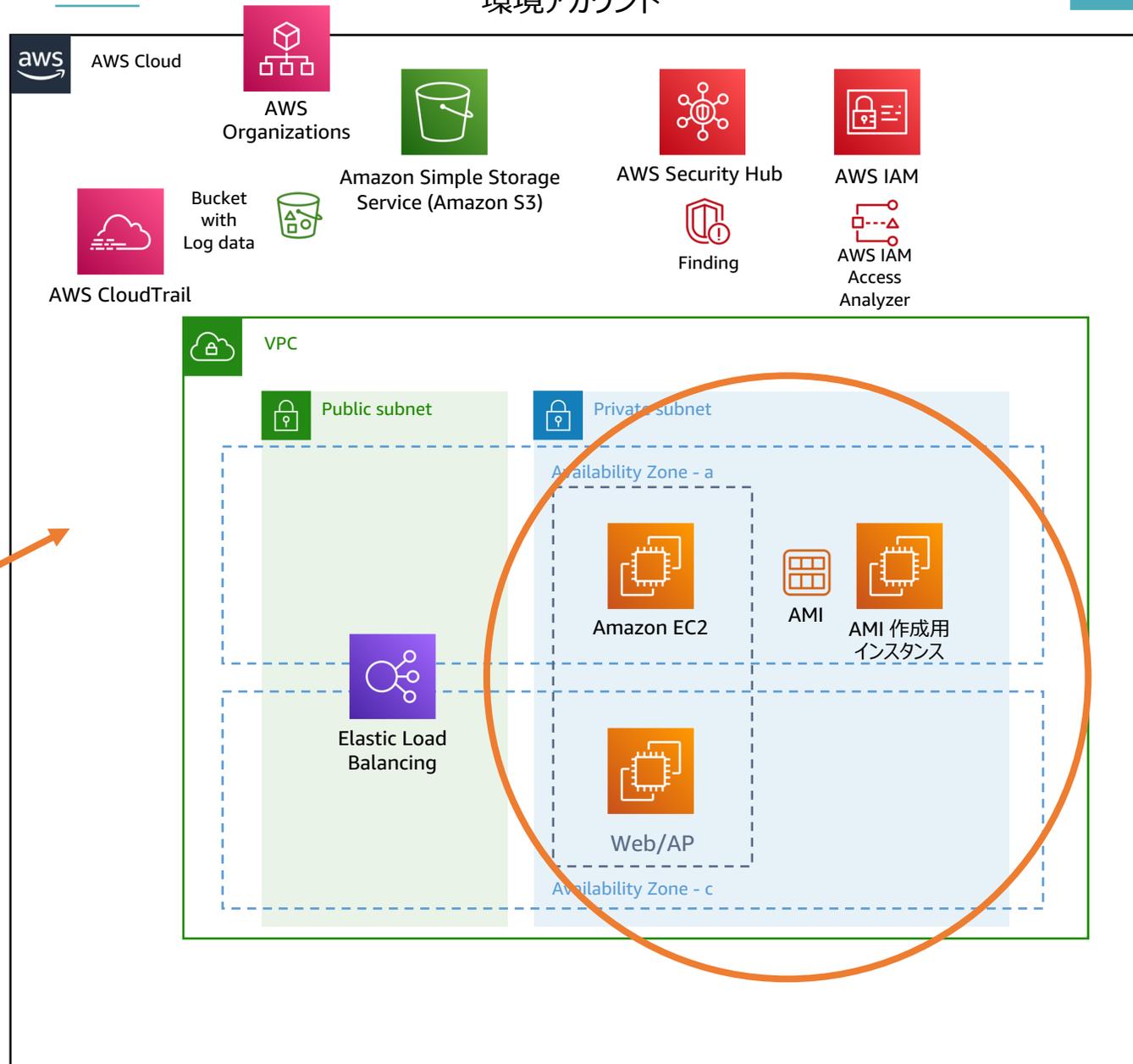


デモ！

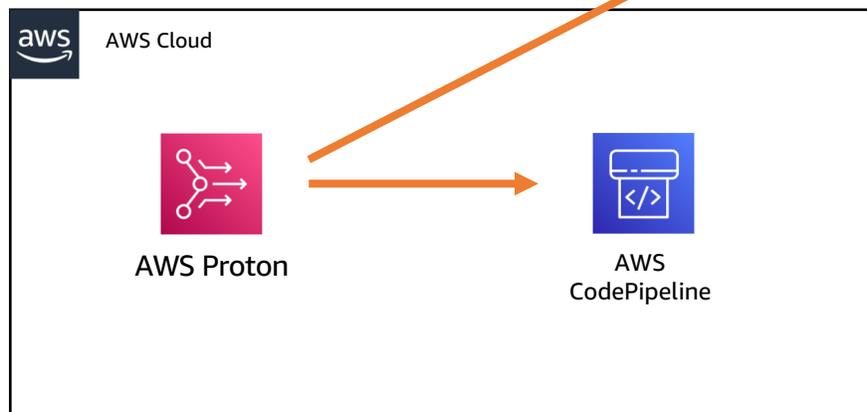
EC2 でもいいじゃないですか

環境アカウント

AWS Proton のサービス紹介では、「コンテナとサーバレスデプロイの自動化」となっていますが、もちろん **EC2 ベース**での構築も可能です。
EC2 ベースでの CI/CD のやり方はいろいろあるかと思いますが、今回は AMI を作成してオートスケーリンググループを更新しています。
(EC2 Image Builder を使うのも良いかもしれません)



管理アカウント



デモ 2 !

- オフィシャルの情報が少ない
 - いまのところ、基本的に英語(と機械翻訳の日本語)の情報しかなさそう
 - 英語の情報もちよっと古い
 - 特に、テンプレートの書き方の情報が少なくてつらい
- 挙動がわかりづらい
 - テンプレートが期待されているものと違ったときに「エラー」としか出なくてわからない
- テンプレートにCDKが使いたい
 - できればTerraformも
- 環境側アカウントから、サービステンプレートを自分で選んで適用できるようになりたい
 - 環境側でサービスの形態に合わせて自分で選びたい
 - 単一アカウントであればできるが、マルチアカウントだとできない
- サンプルの環境テンプレートやサービステンプレートが充実してくるといいな

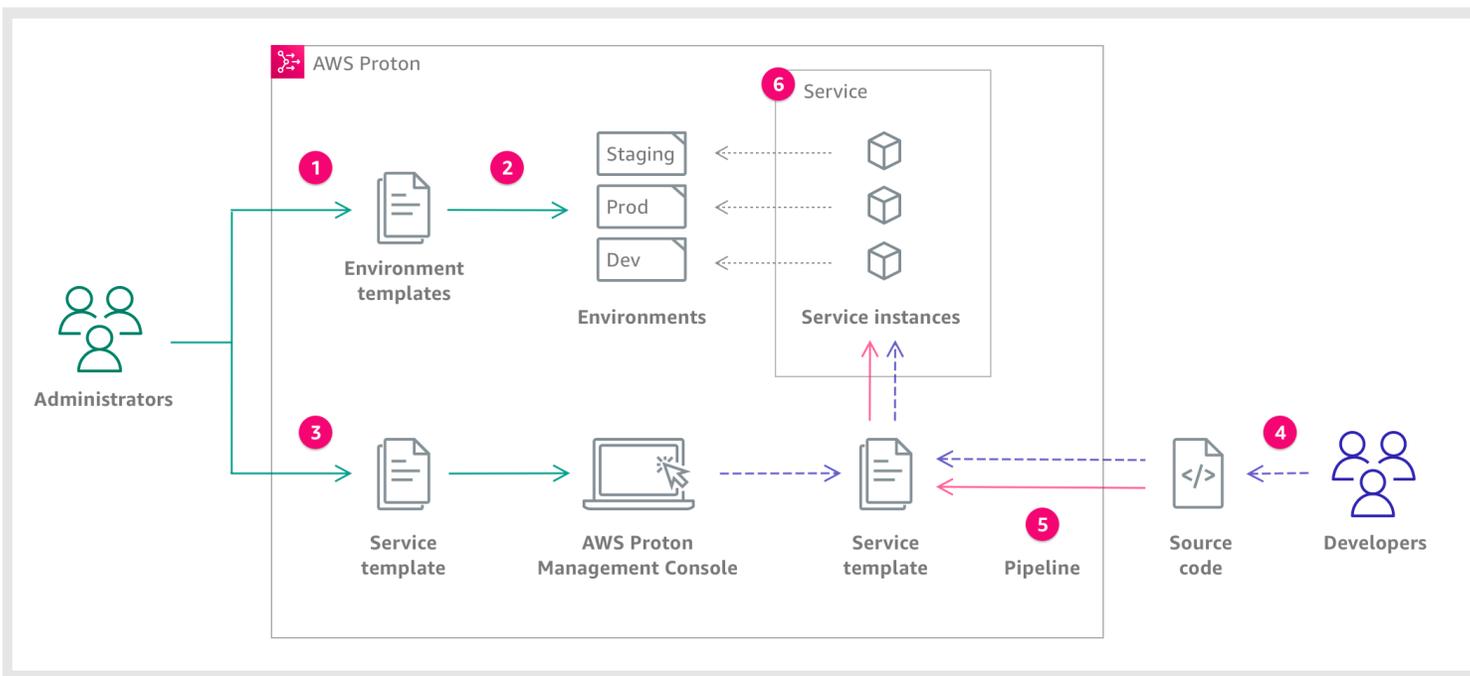
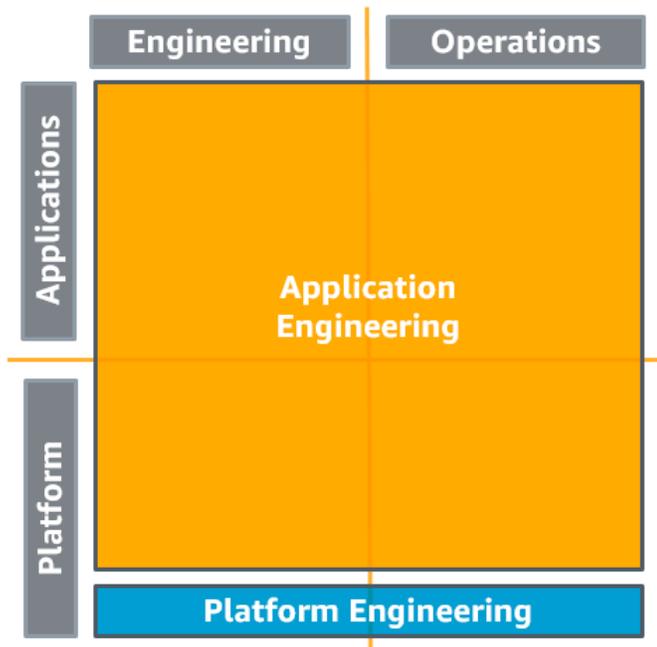
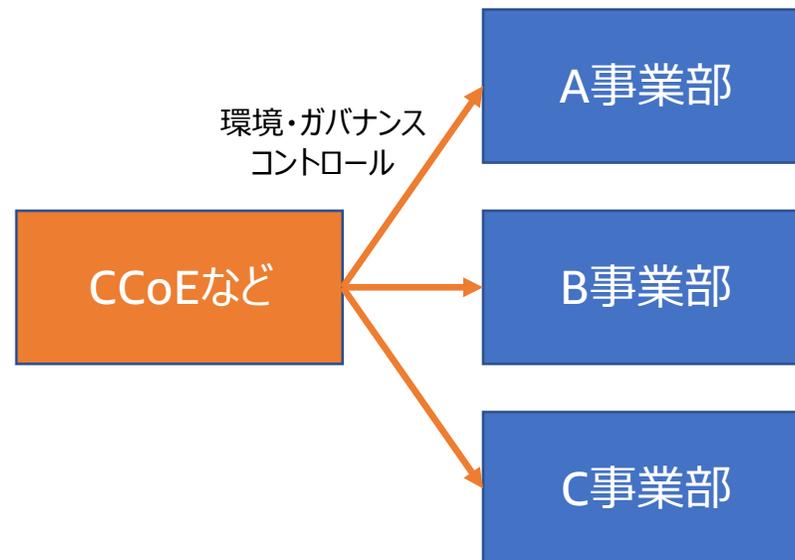
などなど

まとめ

AWS Proton を使うと
プラットフォーム部隊は横断的に
マネジメント&ガバナンスコントロールが
アプリケーション部隊は開発・サービス運用に集中が
できるようになることを期待しています！

本日の内容は合わせて弊社ブログに載せますので、
デモのサンプルコードなどはこちらからどうぞ！

<https://tech.anti-pattern.co.jp/letsuse-awsproton/>





ANTI-PATTERN INC.

“日本のソフトウェアエンジニアを
憧れの職業へ”