



マルチクラスタ戦略を無邪気に採用するとどうなるか

夏のAWS Kubernetes 祭り！ - 2022/08/04



Takada Soh

**高田 颯**

free株式会社 プロダクト基盤本部

SRE Platform team

## 経歴

2020年5月～ 人事労務freee開発

2021年4月 新卒入社

2021年7月～ SRE

## お悩み

Kubernetesにバグ修正のPRを4月に  
送ったものの途中から返信がなく悩み中

マルチクラスタ戦略を  
無邪気に採用するとどうなるか



マルチクラスタ戦略を無邪気に採用するとどうなるか

(運用が) 爆発します。

# 目次

## 何が爆発したのか

- freeeのcluster運用の歴史

## 現在の戦略

- マルチクラスタ（シングルテナント）

- EKS version blue/green upgrade

- インフラの運用を丸ごと引き受けるSRE

## 改善の取り組み

- upgrade方式の整備

- terraform moduleの作成

- 開発チームへの移譲

## 今後の展望

- まとめ

何が爆発したのか



# freeeのcluster運用の歴史



# 何がつらいのか

マルチクラスタ戦略 x EKS version blue/green upgrade戦略  
インフラの面倒をほぼすべてSREが見ている



# マルチクラスタ（シングルテナント）戦略

freeではマルチクラスタ（シングルテナント）戦略を採用

## マルチクラスタのメリット

障害の影響範囲を狭められる

セキュリティ的な権限範囲を明確に区切ることができる

## マルチクラスタのデメリット

利用料金が増える

運用の人的コストが増える

# マルチクラスタ（シングルテナント）戦略

freeではマルチクラスタ（シングルテナント）戦略を採用

## マルチクラスタのメリット

障害の影響範囲を狭められる

セキュリティ的な権限範囲を明確に区切ることができる

## マルチクラスタのデメリット

利用料金が増える

運用の人的コストが増える

とてもつらい



# EKS version b/g upgrade 戦略

マルチクラスタの運用作業で一番工数がかかる

## 背景

EKSは約3回/年のペースで新バージョンリリース

サポート期間はリリースされてから14ヶ月 or 直近4バージョン

少なくともおよそ1年に1回程度version upgrade作業が発生

## 対応

inplace upgradeだと切り戻しができないので安全性のためblue/green方式を採用

# EKS version upgrade 戦略 比較

	工数	安全性
inplace	○ Control Plane : ワンクリック その他 : 要作業	× 切り戻し不可
blue / green	△ 準備・切り替え・削除で 3人日程度	△ 切り戻し可能 IP, DB等共有リソースに注意

# 運用工数が爆発した結果

SRE Platform teamが実質cluster upgrade専門チームに  
EKS 1.18 クラスタ群をupgradeし切るのに1Qかかりきりになった

# インフラの運用を丸ごと引き受けるSRE

EKS移行プロジェクト開始当初は各プロダクト開発チームにサービス運用を移譲すると言っていたが.....

マルチテナントからシングルテナントなEKSに移行した実例



## EKS 移行プロジェクト

- ・ 1月下旬頃からスタート
- ・ 3月13日までに全プロダクトを移行する
- ・ **シングルテナントに変更**する
- ・ 必要な**AWSリソースは開発チーム主導**で用意してもらう
- ・ **Kubernetesクラスタも開発チーム主導**で構築してもらう

SREから開発チームへ権限委譲を果たし、**開発チームにサービスの運用をおまかせ**することが最大のミッション

77



# インフラの運用を丸ごと引き受けるSRE

EKS移行プロジェクト開始当初は各プロダクト開発チームにサービス運用を移譲すると言っていたが.....

マルチテナントからシングルテナントなEKSに移行した事例



## EKS 移行プロジェクト

- ・ 1月下旬からスタート
- ・ 3月13日まで全プロダクトを移行する
- ・ シングルテナントに変更する
- ・ 必要なAWSリソースは開発チーム主導で用意してもらう
- ・ Kubernetesクラスタも開発チーム主導で構築してもらう

**できてない！！**

SREから開発チームへ権限委譲を果たし、開発チームにサービスの運用をおまかせすることが最大のミッション

77



# インフラの運用を丸ごと引き受けるSRE

EKS移行プロジェクト開始当初は各プロダクト開発チームにサービス運用を移譲すると言っていたが.....

マルチテナントからシングルテナントなEKSに移行した実例



## EKS 移行プロジェクト

- ・ 1月下旬頃からスタート
- ・ 3月13日までに全プロダクトを移行する **ここだけはできてる**
- ・ **シングルテナントに変更**する
- ・ 必要な**AWSリソースは開発チーム主導**で用意してもらう
- ・ **Kubernetesクラスタも開発チーム主導**で構築してもらう

SREから開発チームへ権限委譲を果たし、**開発チームにサービスの運用をおまかせ**することが最大のミッション

77





# 開発チームへの移譲

## 原因

当初のミッションを忘れたこと

当初のミッション：SREから開発チームへ権限委譲を果たし、開発チームにサービスの運用をおまかせする

実際のミッション：EC2からEKSへサービスを移行することそのもの

## 起きたこと

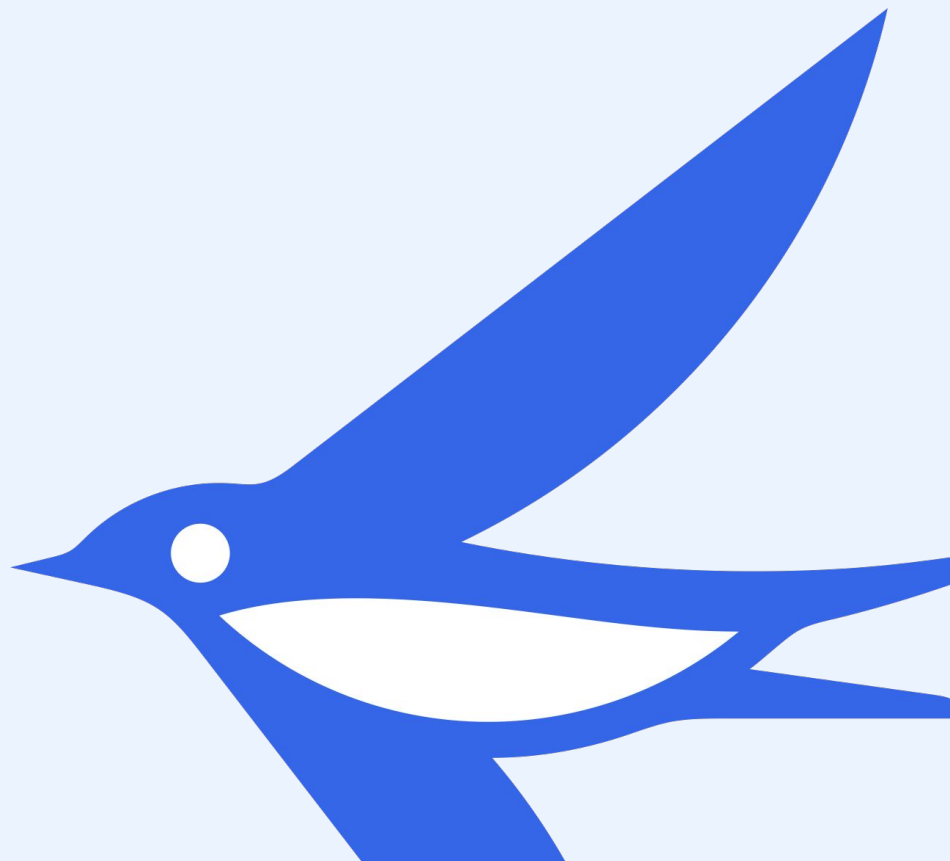
SREが知識・作業移譲を行わないまま主導・ひたすら移行作業

## 結果

開発チーム側にインフラ・運用の意識・技術が育たないまま今に至っている

100近いEKSクラスタ・インフラの面倒をSREが見続けている（ほぼパンク状態）

# 改善の取り組み



# upgrade方式の整備

## 背景

Clusterの作成・削除の手順が煩雑だった

旧方式のterraform-eksctl-provider(eksctl)はいくつか難があった

特定の条件下でnodegroupの追加・削除を行うとaws-authが壊れて既存NodeGroupがClusterから切り離されるバグ

create/update時に何が起こるか確認できなかった

基本的にimmutableでnodegroupの変更などができなかった

## やったこと

terraform-aws-modules/eks をラップした社内terraform modulesを作成

簡単にクラスタ作成・削除を行えるように改善

ArgoCDのApplicationSetを使い、新旧クラスタへのK8s manifestのApplyを容易に

暗黙知だったクラスタ作成・削除手順の整理を行い、より簡単・安全に行えるように

事前チェック内容のリスト化・自動化



# EKS version up b/g 作業内訳 新旧差分

## 準備

プロダクト側チームと日程調整

- TargetGroup作成
- /+ Cluster作成(module化でパッケージング)
- 新クラスタ用helmfile(K8s manifests)の作成
- ArgoCD用の認証情報の作成
- ALB ListenerRuleの差し込み準備
- /+ 手順書作成(テンプレート化)

## 切り替え

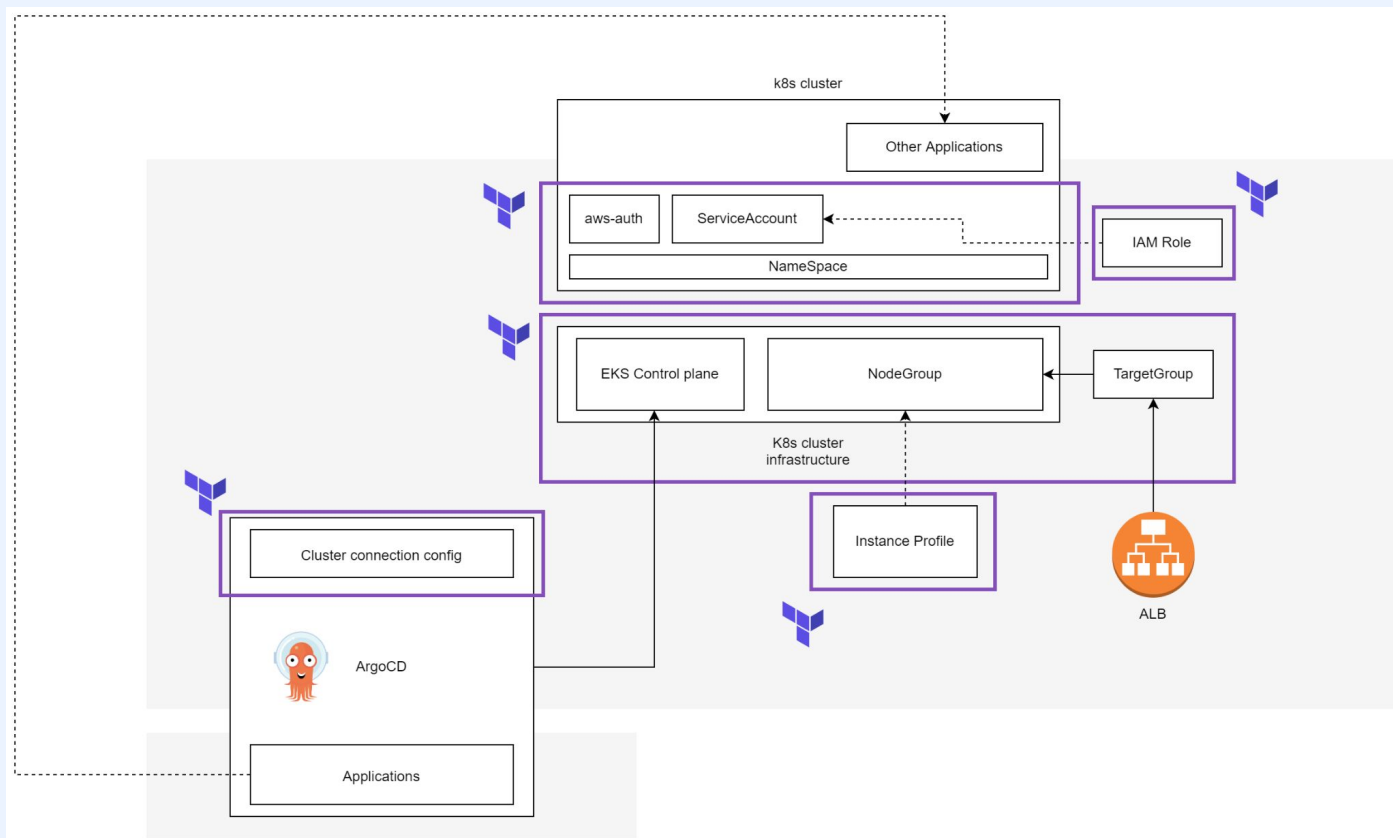
- + インフラレイヤーの疎通テスト  
アプリケーションデプロイ
- /+ ListenerRule切り替え(半自動化)  
動作確認
- cronjob suspend

## 後片付け

- 旧クラスタのNodeを落とす  
旧クラスタ削除



# 参考) terraform modules概観



# 今後の展望

## 背景

開発チームが自律して開発運用を行えるようにしていきたい  
とはいえ余計な手間をかけさせることはPlatform側としても本意ではない

## 対応案

マルチテナント（シングルクラスタ）の検討

過去議論して却下したが再検討の余地あり

EKS version up作業の自動化

クラスタ管理用クラスタを作り、クラスタ立ち上げ・切り替え作業・クラスタ削除までを自動でできる仕組みを作る  
現在PoC作成中

## これからEKSを導入・拡大しようとしている方々へ

EKSのversion upgrade工数を甘く見て「運用でカバー！」とか言っていると

1, 2年後に爆発します

upgrade手順の整理、開発チームへの移譲などはお早めに

**EKS upgradeは計画的に！**

