EKS + マルチAWSアカウントを用いたマイクロサービス開発

日本経済新聞社 大塚 恭平

自己紹介

- 名前：大塚 恭平
- 所属
  - 2017-: 日本経済新聞社
- 業務内容
  - 法人向けサービス開発
  - インフラ構築・運用
  - SRE

本日の内容

- AWS上で稼働している法人向けサービスとその設計
- マルチAWSアカウント下におけるEKSの活用
- まとめ

本日の内容

- AWS上で稼働している法人向けサービスとその設計
- マルチAWSアカウント下におけるEKSの活用
- まとめ

AWS上で稼働している法人向けサービス

2018年　2月	インフラチーム発足
9月	日経リスク&コンプライアンス リリース
10月	日経バリューサーチ AWS移行
2019年	基盤強化（検索・コンテンツ拡充・レコメンド）
2020年　3月	NIKKEI The KNOWLEDGE リリース
6月	日経テレコン AWS移行

+ 継続的なサービスの機能開発

法人向けサービスのシステム

AWSアカウント数	コンテナ稼働数
20+	700+

※ 2020年9月時点

AWS上でスケールするシステム設計を強く意識

システム設計時に考慮したこと

- 開発の効率化
  - 複数の法人向けサービスをAWS上で構築することが早い段階から計画
- サービス横断で用いる共通機能の横展開
  - 検索処理・請求処理・ID基盤など
- 権限管理・セキュリティ
- システムごとに異なる要件
  - エンドユーザーに近いシステム => QA・ABテスト・カナリアリリースなど
  - 基幹システム => 高い稼働率・安定した運用

マイクロサービスアーキテクチャの採用

機能ごとに細分化されたアプリケーションでサービスを構成する手法

メリット

- デプロイ容易性
- 回復性・スケール
- 異なる技術選定が可能

デメリット

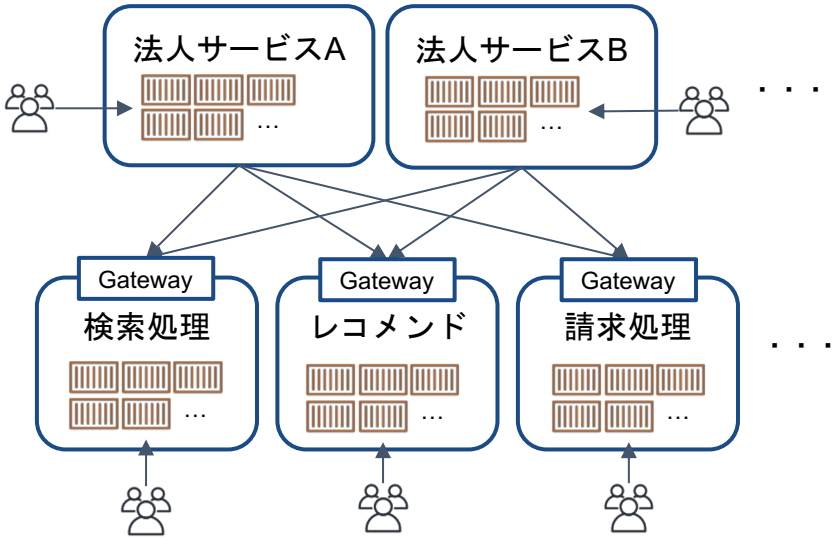
- ネットワークのオーバーヘッド
- 設計が困難

マイクロサービスによる開発の効率化

開発チームは、担当するビジネスドメイン内でマイクロサービスを構築

ドメインごとのインターフェースとなる
ゲートウェイを介して機能提供

- インターフェースを満たす限り
  独立した開発・デプロイが可能
- インターフェースを介して利用する側は
  背後のシステムを意識せずに良い



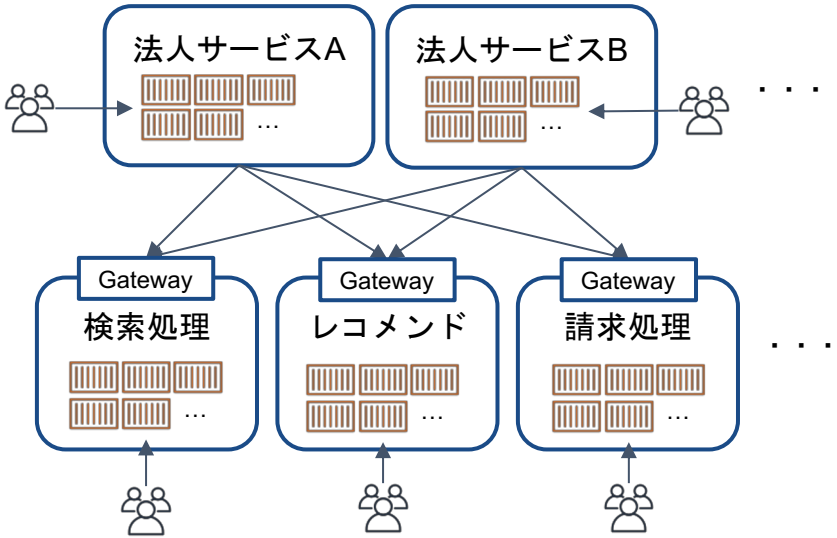
マイクロサービスによる開発の効率化

開発チームは、担当するビジネスドメイン内でマイクロサービスを構築

ドメインごとのインターフェースとなる
ゲートウェイを介して機能提供

- インターフェースを満たす限り
  独立した開発・デプロイが可能
- インターフェースを介して利用する側は
  背後のシステムを意識せずに良い

➡ **開発効率の良い、共通機能の横展開を
可能とするシステム構成を実現**



マイクロサービスにおける権限管理

システムを細分化するため、AWSリソースもその分作成される

- 意図しない権限が開発者に付与されるリスク
  - 本来許可されないデータへのアクセス（故意）
  - 誤ったリソースの設定変更・消去（不注意）
- 最小権限でIAMを設定するのは困難
  - RBAC / ABACに対応していないAWSリソースもある

マイクロサービスにおける権限管理

システムを細分化するため、AWSリソースもその分作成される

- 意図しない権限が開発者に付与されるリスク
  - 本来許可されないデータへのアクセス（故意）
  - 誤ったリソースの設定変更・消去（不注意）
- 最小権限でIAMを設定するのは困難
  - RBAC / ABACに対応していないAWSリソースもある

➡ **ビジネスドメインごとにAWSアカウントを分けることで権限を管理**

システムごとに異なる要件

- エンドユーザーに近いシステム => QA・ABテスト・カナリアリリースなど
- 基幹システム => 高い稼働率・安定した運用

システムごとに異なる要件

- エンドユーザーに近いシステム => QA・ABテスト・カナリアリリースなど
- 基幹システム => 高い稼働率・安定した運用

AWSでは様々なワークロードが用意されている

AWS Elastic Beanstalk

AWS Lambda

Amazon Elastic Container Service

AWS Batch

Amazon Elastic Kubernetes Service

➡ **システムの要件次第でAWSサービスを使い分ける**

本日の内容

- AWS上で稼働している法人向けサービスとその設計
- マルチAWSアカウント下におけるEKSの活用
- まとめ

Amazon Elastic Kubernetes Service (EKS)

Amazon Elastic
Kubernetes Service

AWSが提供するフルマネージド型のKubernetesサービス

メリット

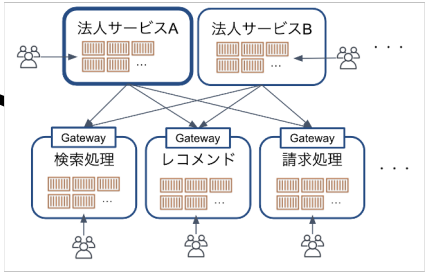
- コンテナ化したアプリケーションの運用管理と自動化
- エコシステムのOSSの恩恵を受けられる
  - Istio (Service Mesh), Argo Workflows, ArgoCD, etc.

    => 複雑な要件に応えることが可能
- Kubernetes自体のライフサイクルに追従
  - 新機能・バグ修正に積極的

デメリット

- データプレーン（EC2/Fargate）とは別にクラスタごとに料金が発生

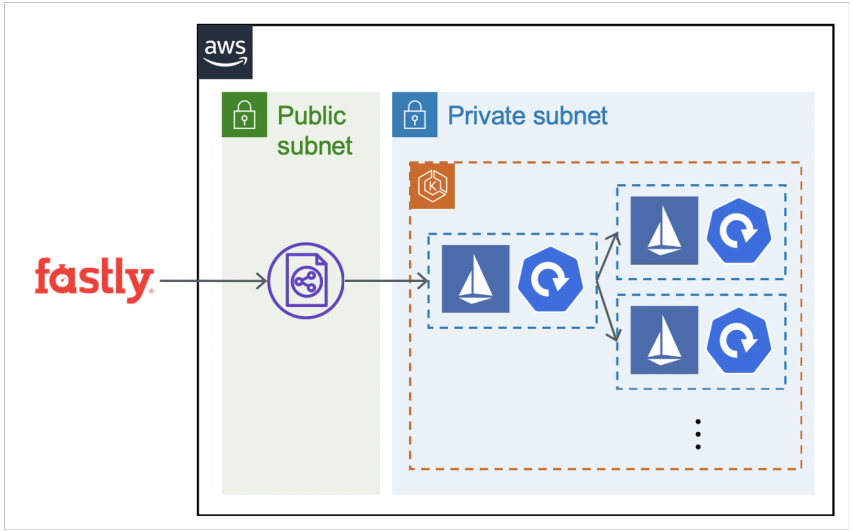
EKS事例1: Webサービスアカウント



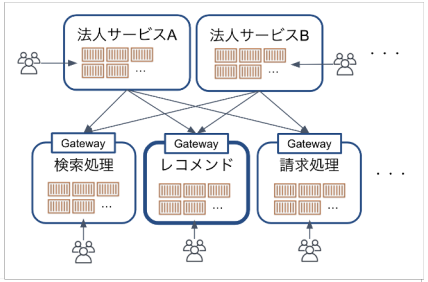
NIKKEI The KNOWLEDGEではEKSを導入

Istioを活用したリリース管理

- 安全なアプリケーションのリリース
- スタンバイ環境でのQAテスト
- 一貫したアプリケーションの監視



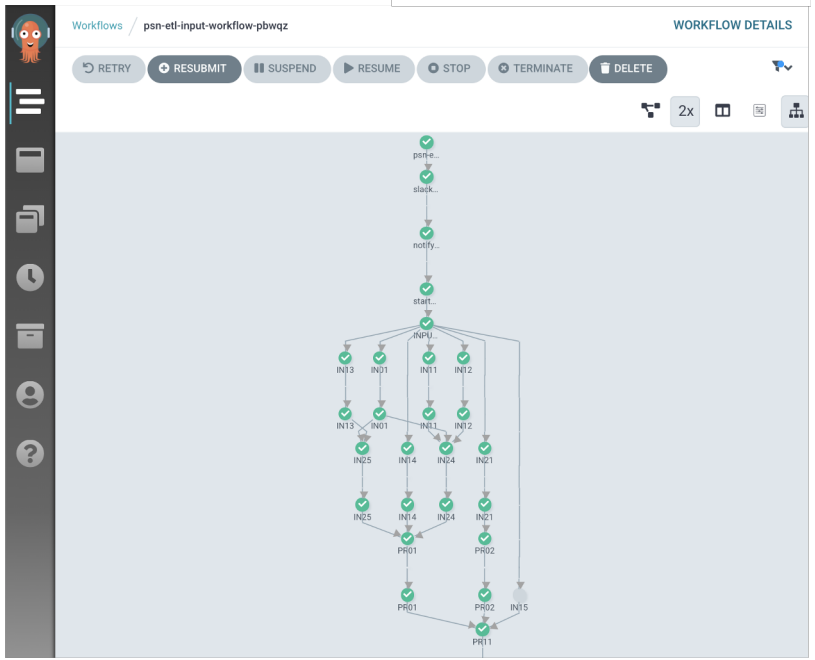
EKS事例2: レコメンド基盤



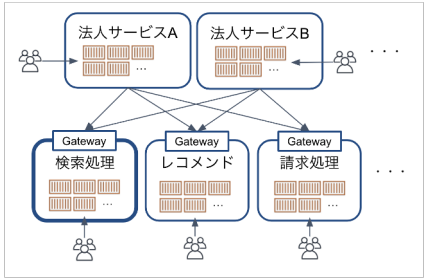
機械学習の学習処理に加えて学習データの抽出や前処理など、多くのジョブを動かすことが求められる

Argo Workflowsを用いたジョブ管理

- 複雑なジョブパイプラインの定義
  - 並列処理・順序制御
- 宣言的なパイプライン管理



EKS事例3: コンテンツ周り



メタデータ修正ツール等、コンテンツに関連する小規模なアプリケーションの作成需要は高い

- アプリケーションの追加コストを下げる
  - namespace追加のみで素早くアプリケーション開発に着手
  - より多くの試行錯誤に繋がる
- 一貫したアプリケーション監視の実現
  - クラスタ単位で共通のアラートやログ転送の設定を入れる

EKSを導入して良かったこと

- Kubernetes導入が容易
- Kubernetesのエコシステムの力を借りて様々なシステム要件を実現
  - リリース管理・ジョブパイプライン・コンテンツ周り
- マイクロサービスの追加コストの減少
  - より多くの試行錯誤を重ねることに繋がっている
- 社内ナレッジの蓄積

マルチAWSアカウント戦略との相性

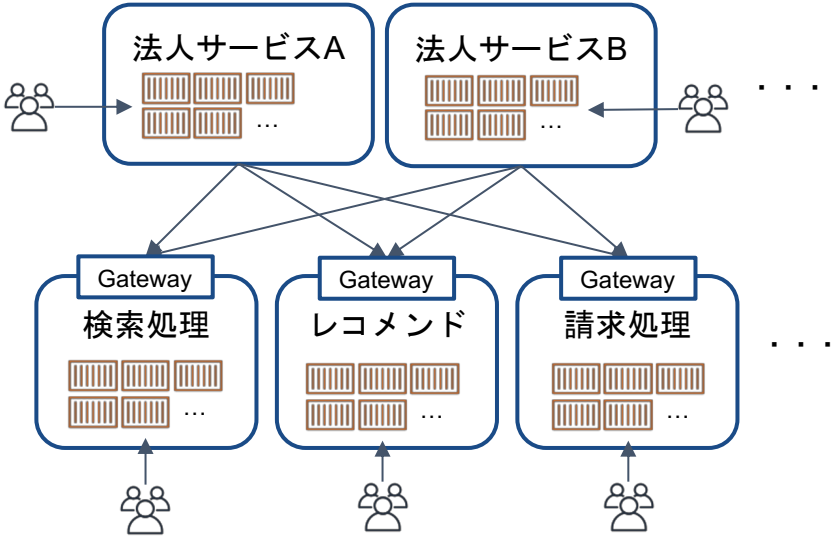
運用・管理コストの観点で相性は良くない

- 管理対象のEKSクラスタが増加傾向
  - アカウント単位のリソース分離とのトレードオフ
- AWSアカウントごとのクラスタでは高集積の恩恵を受けづらい
- EKSクラスタのアップデートコスト
  - オペレーションの効率化（Infrastructure as a Code・GitOps）で緩和することは可能

EKSを導入する観点

機能追加が多く、柔軟に
要件に応えることが求められる
➡ **EKS**

安定した運用・高い稼働率
➡ **ECS**



本日の内容

- AWS上で稼働している法人向けサービスとその設計
- マルチAWSアカウント下におけるEKSの活用
- まとめ

まとめ

- マルチAWSアカウントでマイクロサービス開発を行うことでスケールする開発体制
- 法人向けサービス開発で用いている一部のAWSアカウントにEKSを導入
- EKSとそのエコシステムにより、様々なシステムを実現
- 機能追加の頻度が高く、柔軟に要件に応えたいシステムにEKSは向いている