

confidential

許可なき配布・参照はお断りしております



AWSマルチアカウント戦略を採用した サーバーレスアプリケーションの管理と運用

木村情報技術株式会社
徳山 鐘三



本日本話すること

- Docker on LambdaでサーバーレスなRailsアプリケーションを構築、運用してみた
- サーバーレスなRailsアプリケーションをマルチアカウントで運用してみた

自己紹介

氏名：徳山 鐘三

所属：木村情報技術株式会社

好きなAWSサービス：

AWS Lambda

Amazon S3

Amazon DynamoDB

Amazon Cloudfront

AWS ElementalMediaシリーズ

Amazon Elastic Transcoder



会社紹介

会社名

木村情報技術株式会社

設立

2005年7月29日

主要取引先

[企 業] 製薬会社 約120社、大手企業約100社
[学 会] 日本臨床腫瘍学会、日本臨床整形外科学会、佐賀県作業療法士会
佐賀県臨床工学技士会、他
[教育機関] 順天堂大学、佐賀大学、京都薬科大学、他

代表者

木村 隆夫

従業員

431名（他 取締役4名、監査役1名）

事業内容

- 人工知能(AI)活用事業および人工知能サービスの研究・開発
- Web講演会運営・配信事業、収録およびコンテンツ制作
- ウェビナー情報一括管理システムの開発・運営
- プラットフォーム事業（オンライン学会、医薬品情報、地方銀行×企業）
- メタバース事業
- eスポーツ事業
- 医薬品業界向け出版及び研修コンサルティング事業
- サガン鳥栖スマホの販売とエクスモバイルの代理店販売
- [Zoom] および [Remo] 会議システムの代理店販売
- 他、各種システム・Webサイトの構築及び保守

その他

プロサッカーJ1リーグチーム「サガン鳥栖」、
プロeスポーツチーム「Sengoku Gaming」、
プロeスポーツチーム「ALBA E-sports SAGA」のスポンサーに就任



今回の事例

 木村情報技術株式会社
KIMURA INFORMATION TECHNOLOGY Co., Ltd.

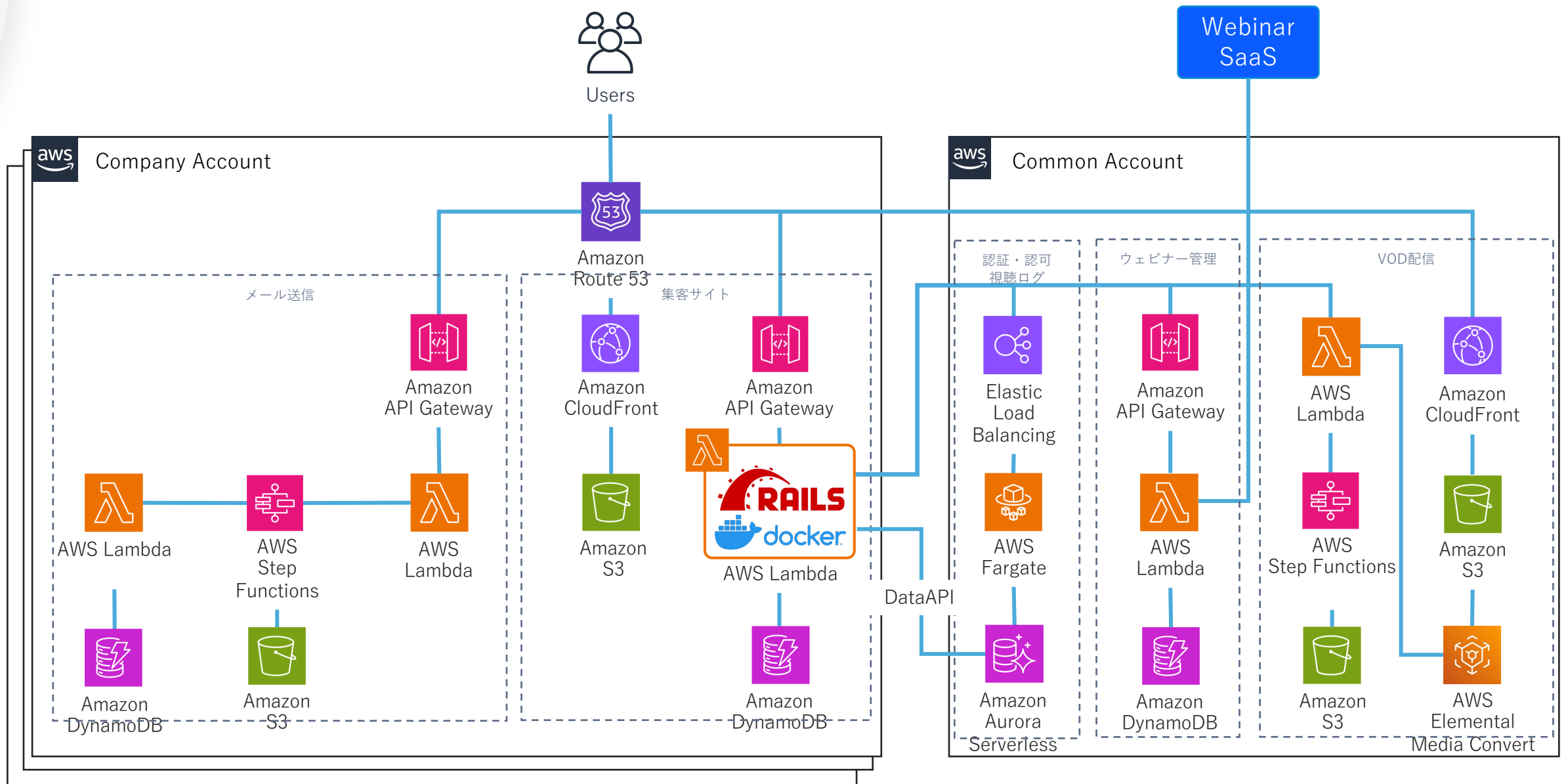
すべてのウェビナーの視聴情報の把握と分析をこれ一つで

ウェビナー情報一括管理システム

Biznar

(ビズナー)

Biznar アーキテクチャ概要



Docker on Lambdaでサーバーレスな
Railsアプリを構築、運用してみた

Docker on Lambdaについておさらい

- AWS Lambdaのデプロイパッケージ
 - .zipファイルアーカイブ
 - コンテナイメージ
- Lambda関数のコンテナイメージの種類
 - LambdaのAWSベースイメージ
 - カスタムランタイムのAWSベースイメージ
 - 非AWSベースイメージ

Docker on Lambdaを採用した背景

- 初期検討よりサーバーレスアーキテクチャでの構築を想定
 - アプリケーション自体（集客サイトCMS）の規模自体があまり大きくない
 - コストパフォーマンス
 - 可用性
- 当初はRuby on Railsを利用する想定だった
 - 「そもそも、RailsのDockerイメージをそのままLambdaで動かせば良いのでは？」となった

RubyのAWSベースイメージをカスタマイズし、Ruby on RailsのDockerイメージとしてAWS Lambdaで実行する方向とした

Docker (Rails) on Lambdaの管理・運用してみた

- メリット
 - ローカルでもAWSでも同じコンテナを起動できるので、開発/本番のギャップがない
 - 通常のRailsアプリケーションと同じ感覚でサーバーレスアプリケーションが構築できる
 - サーバーレスとコンテナの美味しいとこどり
 - 外部のミドルウェアの管理が楽
 - サーバー管理不要
 - やっぱり安価
- デメリット
 - コンテナ、サーバーレスどちらの学習も必要
 - Lambdaの実行時間の都合上、Active Jobなど、Rails単体で非同期Jobが使いづらい
 - インフラ自体はAPI Gateway+Lambdaなので、どちらの制限事項も考慮する必要がある (タイムアウトや実行時間など)

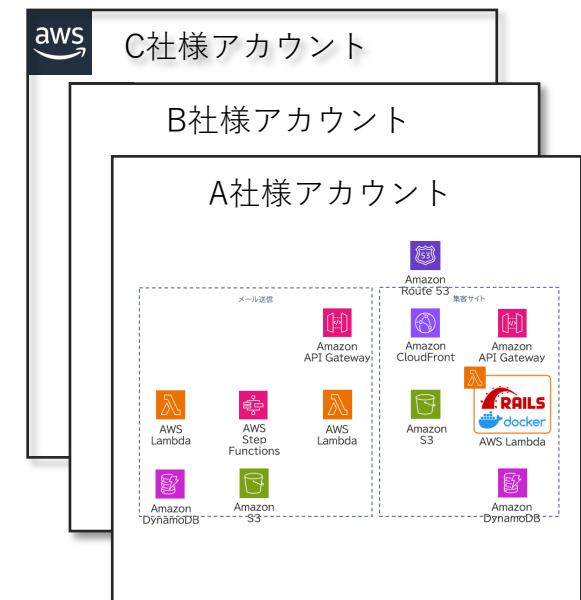
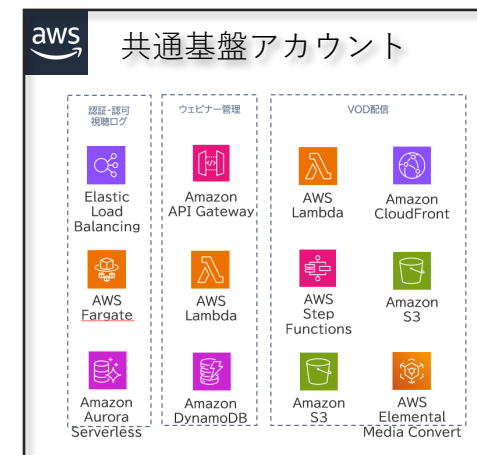
Docker (Rails) on Lambdaの管理・運用してみた

- 苦労した（している）ところ
 - デプロイ実行マシン（AWS CodeBuild）へ渡す権限調整
 - IaCにserverless frameworkを利用
 - DataAPI
 - RDS Proxyを検討したがVPC Lambda初期起動時間などネックだった
 - 2024年10月までにAurora Serverless v2へ更新が必要
 - Aurora Serverless v2にはDataAPIがないので、アーキテクチャ見直しが発生するのが確定している
 - Aurora Serverless v 2 へDataAPIがくること、ずっと待っています

サーバーレスな Rails アプリケーションを
マルチアカウントで運用してみた

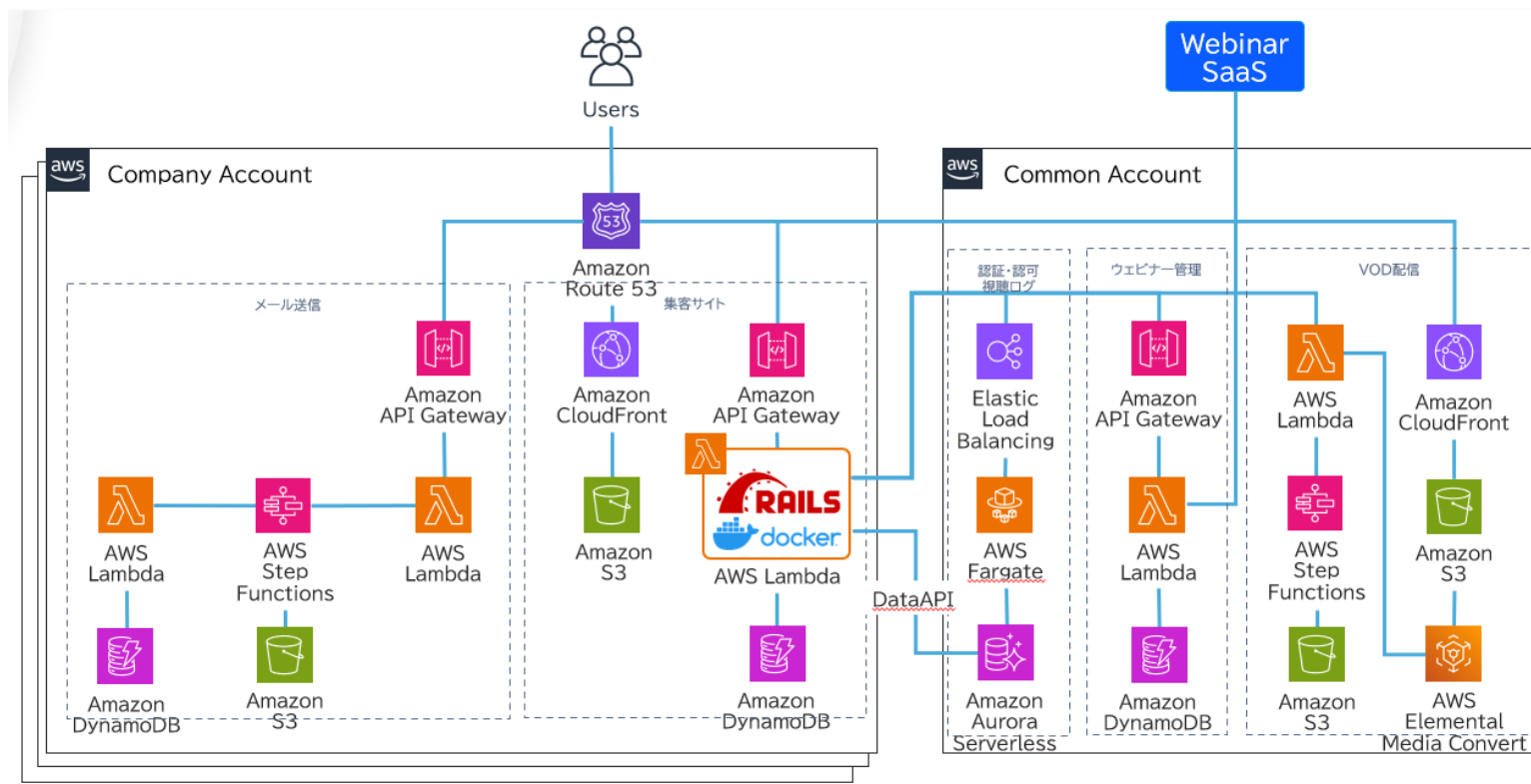
マルチアカウント戦略についておさらい

- マルチアカウント戦略とは？
 - 特定の単位や基準でAWSアカウントを分割する戦略のこと
 - 開発用、本番用、などなど
- 本事例では以下のようにアカウントを分割
 - 共通基盤アカウント
 - 本番用
 - 開発用
 - 各顧客用本番アカウント
 - 各顧客用開発アカウント
 - AWS Control Towerも利用しているため Audit, Log, 管理アカウントなども存在



なぜマルチアカウントとしたか

- サーバーレスアーキテクチャを採用しており、アプリケーションの多くの部分で Lambda を利用しているため、アカウント単位での同時実行数などの制限を考慮
- 集客サイト部分は顧客ごとに分離し、シングルテナントのように扱いたい



マルチアカウントの管理・運用について

- アカウント毎にユーザーのアクセス権限を分離できているのはやはりとても良い
 - 同一アカウント上での権限調整の運用コストがなくなる
 - うっかり別サービスの設定を触ってしまう、などのミスがなくなる
- 各AWSサービスの制限をあまり気にしなくて良い
- 各顧客アカウント毎の請求が分かりやすい
- AWS Config、AWS Security Hubなどのガードレール設定
- Control TowerのAccount Factoryを使用することで、アカウント作成、設定も自動反映できる

全体まとめ

- Docker on LambdaでサーバーレスなRailsアプリケーションを構築、運用してみた
 - 小～中規模のアプリケーションには有用
 - 出来るだけ低コストに抑えたいアプリケーション向け
 - LambdaやAPI Gatewayなどの制限をしっかりと把握しておく
 - サーバー運用コストなどが抑えられ、アプリケーション開発に集中できるのはとても良い
- サーバーレスなRailsアプリケーションをマルチアカウントで運用してみた
 - アカウントを分離することで、権限分けが容易になり、事故を防げるようになったので、皆幸せになる
 - 特に細かい要件がなければ、AWS Organizations、Control Towerも使っていくととても便利

私たちの仕事は、
常にアイデアをもとに新しいツールを創造し、
お客様にお役立ちできることを目指してまいります。
すべては、人に喜ばれるため、
日々進化する需要にお応えするために。

always new idea.