

Serverless Operations

02	資料概要とサーバレスアーキテクチャの背景	17	自社開発プロダクト
04	Serverless Operations, Inc について	19	実績とお客様インタビュー
06	サーバレスアーキテクチャとは	38	ご契約の流れ
08	サーバレスアーキテクチャを採用するメリット	40	メンバー紹介
11	課題例とお手伝いできること	42	認定パートナー
13	強みと特徴	44	会社概要
15	内製化支援とその他のサービス		

Serverless Operations

Introduction

資料概要とサーバーレスアーキテクチャの背景

Serverless Operations

2014年「AWS Lambda」がリリース、 サーバーレスという言葉が一般的に使われはじめ、 世界中へ大きく広がりました

この資料の目的は、サーバーレスアーキテクチャを採用し製品を開発することで企業にもたらす利益やメリットをお客様が理解できるように支援することです。

2014年にアマゾンウェブサービス(以下、AWS)からサーバーレスの代表的なプロダクトとも言えるAWS Lambdaがリリースされ、サーバーレスという言葉が一般的に使われるようになりました。その後、クラウドやOSSのコミュニティが形成され、様々な分野でサーバーレスアーキテクチャが導入されることで、そのムーブメントは世界中で大きく広がりました。

サーバーレスオペレーションズでは多数のお客さまへサーバーレスアーキテクチャの導入支援を行っており、その中でさまざまなビジネス分野やユースケースのソリューションを長年設計してきました。そして、効率が良く、費用対効果が高く、安全で信頼のおけるサーバーレスを中心としたアーキテクチャ上で自社の製品を稼働させることでビジネスの成功の可能性が大いに高まると当社は確信しています。

この資料は主に経営者、エンジニアリングマネージャーといった事業の方針や開発の方向性を決定する層を対象としています。弊社での事例を参考にしながら、サーバーレスアーキテクチャを採用することの事業上のメリットを説明していきます。

Serverless Operations, Inc

Serverless Operations, Incについて

Serverless Operations

サーバーレスで クラウドの価値を最大限に

「サーバーレスでクラウドの価値を最大限に」という理念のもと、様々な企業に AWSを中心としたサーバーレスアーキテクチャの導入・技術支援を行っている会社です。



AWSのサーバーレスアーキテクチャ導入の多数の実績が評価され、2020年よりセレクトコンサルティングパートナーとして認定されています。

また、同年9月にはAWS Lambdaのプロフェッショナルとして、AWSサービスデリバリープログラムのAWS Lambdaパートナーに認定されています。

Serverless Architecture

サーバーレスアーキテクチャとは

Serverless Operations

ユーザー企業の運用管理コストを 最小限にするシステム構成

必要なときにのみAWS Lambdaが立ち上がるため
AWSに支払うコストが最適化されます

サーバーレスアーキテクチャとはAmazon EC2に代表されるクラウド上の仮想マシンを使用すること無くシステムを構成することです。コンピュティングリソースとしては主にAWS Lambdaが中心に使用されます。

AWS Lambdaは様々なイベントをトリガーにして、事前に用意しておいたプログラムを実行します。
例えば、オブジェクトストレージであるAmazon S3にファイルがアップロードされたタイミングやAmazon DynamoDB(マネージドNoSQLデータベース)にデータが追加されたタイミングなどがそれに当たります。
この特性によりAWS Lambdaは必要なときのみ立ち上がるため、Amazon EC2をベースにシステムを構築することに比べてクラウド上のコンピュティングリソースの使用を節約でき、結果、ユーザー企業がAWSに支払うコストは最適化されます。

アップデートもAWSが必要なタイミングで行うため
システム運用管理コストが最小限に抑えられます

また、AWS Lambdaはトラフィックに合わせて自動でスケールアウトします。Amazon EC2のオートスケーリングのように事前にキャパシティを見積もり設定する必要はありません。
リージョンごとの上限値はありますが、突発的なトラフィックに対しても自動でスケールアウトして拝いてくれます。これにより従来インフラエンジニアが実施してきたタスクの大部分を肩代わりしてくれます。

さらにOSやミドルウェアのアップデートもユーザー企業側で実施する必要はありません。
すべてAWSが必要なタイミングで実施してくれます。ユーザー企業側で実施が必要なことはAWS Lambdaが動作するランタイムのアップデートとライブラリなどのアプリケーション上で動作するソフトウェアのアップデートのみとなり、これによりユーザー企業におけるシステムの運用管理コストは最小限となります。

The Advantages of Serverless Architecture

サーバーレスアーキテクチャを採用するメリット

Serverless Operations

Merit 01 — Speed

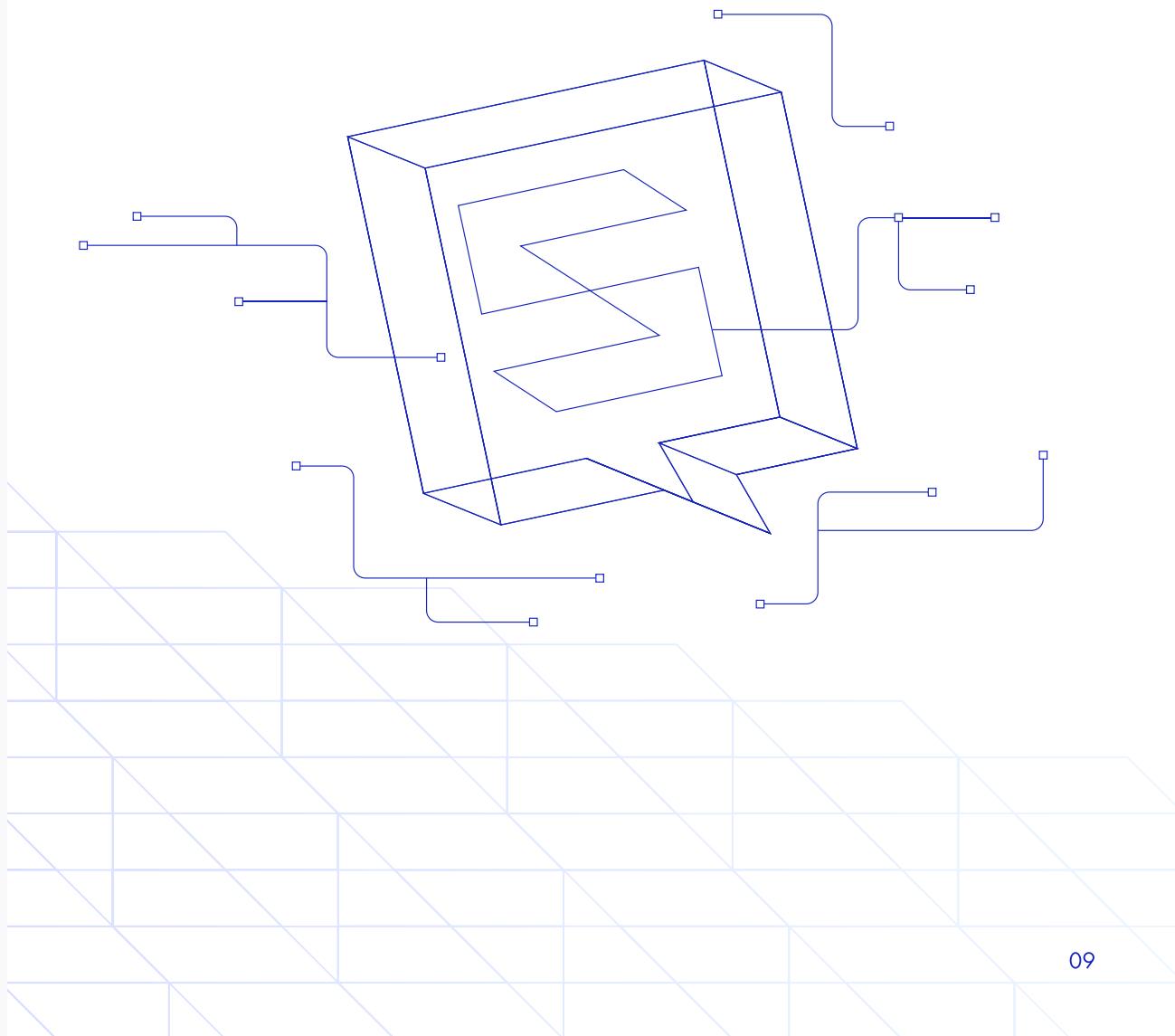
新サービスの迅速な立ち上げによる 企業競争力の上昇

自分たちのアイデアを製品として具現化し、市場にリリースすることで本当に顧客の価値を解決するのかを確かめ、製品として成長させていくサイクルを早く行うことは、企業が競争力を得るために最も重要なことのひとつです。

サーバーレスアーキテクチャによる実装は、CDKやServerless Frameworkといったクラウドにアプリケーションをデプロイするためのツールを使います。それらはコマンドひとつでデプロイされ、ユーザー企業はすぐにアプリケーションが動く環境を手に入れることができます。

また、Amazon EC2を使用する場合に比べて、事前のキャパシティプランニングにかかる時間やコストも最小限にして、アプリケーションを動作させることができます。

このように、サーバーレスアーキテクチャの採用は、ビジネスの課題を解決させるアプリケーションロジックに集中することを助け、顧客の価値を解決するものを作り、それを市場にリリースするサイクルを早めることで、その企業競争力は非常に高くなります。



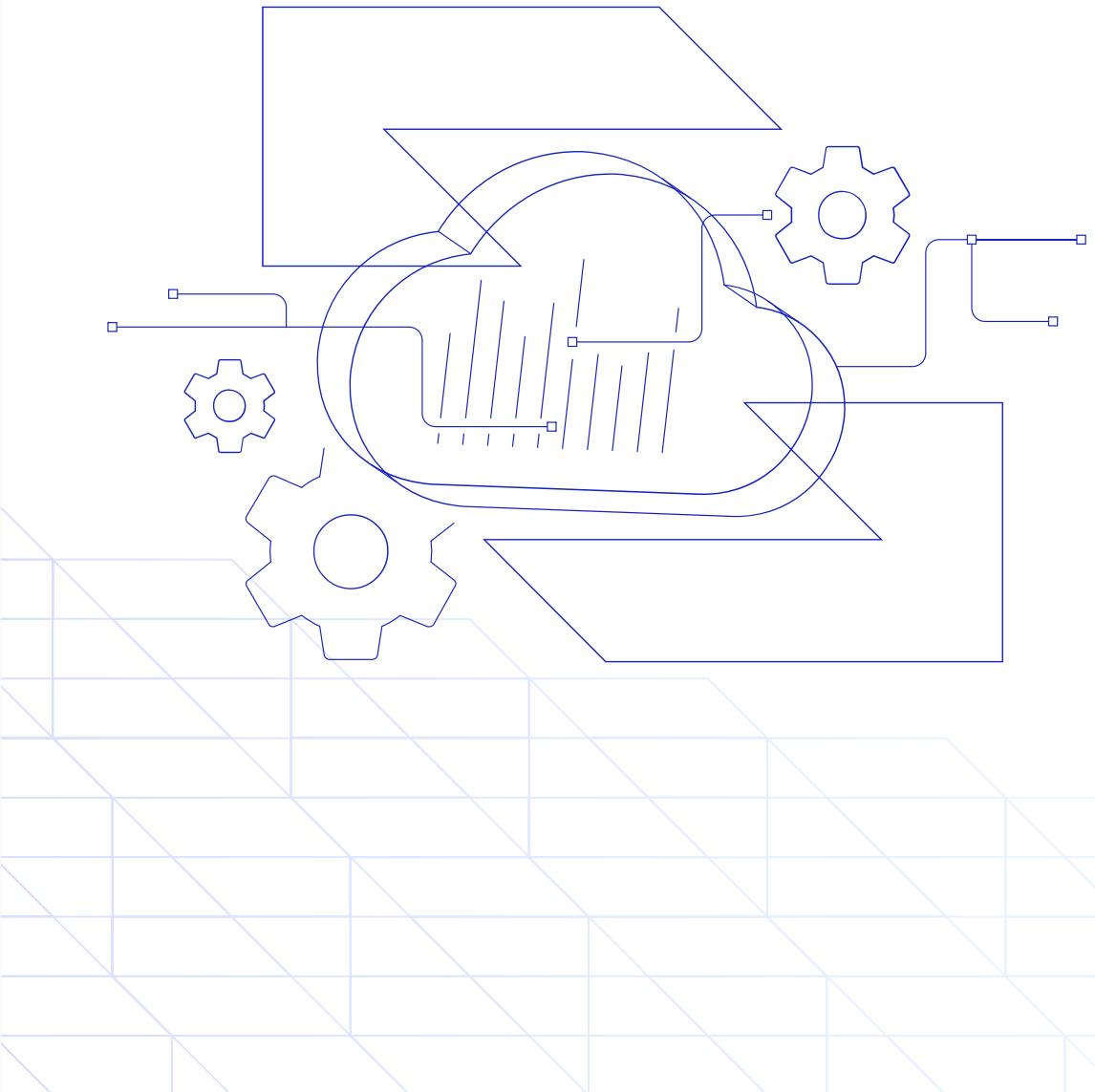
Merit 01 — Cost

従量課金と運用管理が不要になる コストメリット

サーバーレスアーキテクチャで実装する際に使用するAWSのサービスの料金体系はその多くが「使用した分だけ」になっています。つまり製品をリリースしてまだ多くの顧客を獲得できていないフェーズでは、そのAWS使用料金はかなり安くなります。これは特に資金に限りがあるスタートアップ企業には非常に嬉しいでしょう。

また、その事業が大きく成長した後も、従来のAmazon EC2をベースにしたシステムでは、多数のインフラエンジニアがそのシステムを維持するために必要となるでしょう。

しかし、サーバーレスアーキテクチャでは、本来Amazon EC2をベースにしたシステムで必要であった、インスタンスの管理やOSやミドルウェアの管理、キャパシティプランニングなどの多くのインフラ管理タスクをAWSにアウトソース出来る状態となります。結果としてそこにかかる人的コストも最適化されます。



Case Study

課題例とお手伝いできること

Serverless Operations

サーバーレスについてお困りではありませんか？

お客様のチームに入り、 現状を把握しながら課題を共に解決します。



お客様の課題



お手伝いできること



お客様にとっての価値

チームの内製化を行いたい

AWSを効果的に活用してプロダクトやシステム開発、チームの内製化を行いたいが、技術的な経験をもったメンバーがおらず、進めることができないでいる。



サーバーレス開発支援・内製化支援

お客様のチームに入り、内製化をするための課題を洗い出し、実際にお客様とプロダクト開発を行いながら技術習得のお手伝いをします。



価値あるプロダクトを 素早く開発できるスキルを習得

- AWSによるサーバーレスアーキテクチャ及びフロントエンドの技術を習得できる。
- やりたいことを素早く形にしてチャレンジがやっていけるだけのスキルが習得できる。

開発を丸ごと依頼したい

AWSを効果的に活用してプロダクトやシステム開発を行いたいが、社内に開発リソースがなく受託してくれるところを探している。



サーバーレス受託開発

AWSによるサーバーレスアーキテクチャをベースとして、お客様の希望するシステムをフロントエンドからバックエンドまで一気通貫で開発します。



保守性・メンテナンス性の高い システムを獲得

サーバーレスの良さを生かした、コスト効率が高く保守性やメンテナンス性の高いシステムが手に入る。

Features

強みと特徴

Serverless Operations

“共創”とは、 お客様と“共につくる”こと。

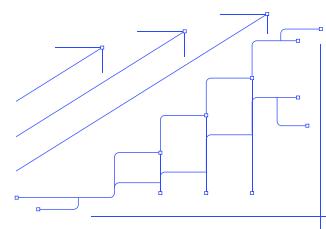
第一線で培った豊富な知見やノウハウを共有し、
お客様と“共に”課題を解決へと導きます。

私たちはこれまで、Serverless Frameworkを中心としたOSSコミュニティで、OSS開発やコントリビューションを続けてまいりました。グローバルの第一線で培ってきたサーバレスアプリケーション開発に関する知見をお客さまに共有しながら、お客様の現状やスケジュール、予算など、ご要望に合わせた支援・サポートを行います。

一時的な開発支援だけでなく、 中・長期的な業務改善へと導きます。

サーバーレスオペレーションズがお客様のチームに加わることでお客様の現状をしっかりと把握し、経験不足や技術力不足を全面的にサポートします。プロジェクトの進行から技術面のアドバイスまで一貫しておこなえるため、目の前の一時的な開発・運用だけでなく、今後の組織・チームづくりや採用、マネジメント等にも活かすことができます。

Optimisation

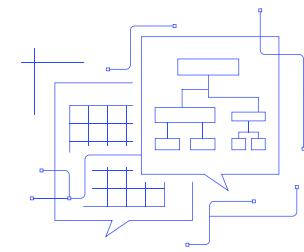


Merit 01 —

質・スピード・コストの最適化

サーバーレスのメリット・デメリットを理解して自社の開発にスムーズに導入し、サービスの質や開発スピード、開発コストの最適化を実現できます。

Self-propelled

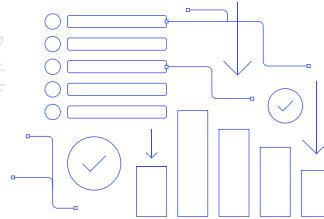


Merit 02 —

自走できるチームへ

共に手を動かすことで専門的な知識や技術が身につき、次回からはサーバーレスアプリケーションの設計や開発が自走できるようになります。

Team Building



Merit 03 —

今後のプロジェクトに活かせる

お客様が今後新しいプロジェクトを進行する際に、スケジュール調整やスタッフのアサイン等がスムーズに進みます。さらに、今後のチームづくりや採用、エンジニア育成に役立つ知識が得られます。

Services

内製化支援とその他のサービス

Serverless Operations

サーバーレス 内製化支援

お客様のチームだけで開発や運用が行えるよう、スキル習得をお手伝いします。

AWSのサーバーレスアーキテクチャを技術的なバックボーンとして、お客様のチームの内製化を支援します。

デジタルトランスフォーメーション時代に重要視される「企業の内製化」

企業がデジタルトランスフォーメーションを進めるためには、企業自身が優秀な人材を獲得し、「内製」で自分たちのプロダクトやシステムを開発することが必要です。

世の中の情報や進化のスピードは非常に早く、旧来日本企業で多く用いられていた外部のベンダーにすべて開発を任せると方法では、十分なスピードを得ることはできず、徐々に競争力は衰えていくと考えられます。

内製でシステムを作り、アジャイルな開発手法でプロダクトをアップデートできるようになれば、市場の変化や顧客の声を素早く取り入れて良いものを市場に送り出していくことが可能となります。

お客様と共にプロダクト開発を行い スキル習得をサポート

弊社では、AWSのサーバーレスアーキテクチャを技術的なバックボーンとして内製化の支援を行います。

お客様のチームに入り、内製化をするための課題を洗い出し、実際にお客さまと一緒にプロダクト開発を行いながらスキルの習得をサポート。

最終的にはお客様のチームだけで、価値のあるプロダクトを素早く開発していくだけのスキル習得をお手伝いします。



Serverless OperationsはAWSの内製化支援推進パートナーです
AWS社と連携し、「AWSパートナー」としてより幅広く効果的な内製化支援サービスを提供します。

その他のサービス

AWSの技術を活用し、サーバーレスアプリケーションの設計・開発・運用までを支援します。

Serverless Operationsでは、AWSを中心とした右記のようなサーバーレスアプリケーションの開発支援が可能です。



Products

自社開発プロダクト

Serverless Operations

サーバーレスアプリケーションの開発をより簡単に

サーバーレスアプリケーションの開発をより簡単に・容易にするために、私たちが独自に開発したプロダクトをご紹介します。



Jeffy

AWS LambdaのPython向けのサーバーレスアプリケーションフレームワーク。Lambdaをよりシンプルに実装するためのツールセットを提供しています。



Costless

サーバーレスのコストをもっと分かりやすくするSaaS。AWSのサーバーレスにおけるコストの詳細を可視化し、継続的なコスト最適化・削減を支援するためのサービスです。



Diggie

Diggie

AWS OpenSearch(Amazon Elasticsearch Service)の操作やクエリを行えるGUIクライアント

Serverless Step Functions

AWS Step Functionsの構成を管理してデプロイするためのServerless Frameworkプラグイン

Serverless API Gateway Service Proxy

API Gatewayと他のAWSサービスをLambda無しで直接インテグレーションを行うAWS Service Proxyの機能を簡単に構成管理するためのServerless Frameworkプラグイン

Serverless Dashboard for Atom

GUIからServerless Frameworkを使いクラウド環境へ構成をデプロイ、可視化するためのAtomエディターアドオン

Serverless Delete Loggroups

Serverless Frameworkのv1.6で発生したLog Groupの変更に対応するためのプラグイン

Works & User Interview

実績とお客さまインタビュー

Serverless Operations



多様で柔軟な働き方を目指し、 サーバーの運用脱却と高可用性を実現するAWSの フルサーバーレスで物流会計システムを構築！

株式会社LIXIL様向けに物流向けの会計仕分けシステムを開発しました。数十万件の物流拠点からの請求に対して仕訳処理を一括で行うためのシステムです。サーバーレスアーキテクチャを採用することで、初期のサーバー等の調達にかかる時間が大幅に削減されると共に、運用フェーズにおいてもメンテナンスコストは必要最小限の人員とコストで実現できています。結果として大幅な効率化とコスト削減を実現することができた事例です。

お話をうかがった方々

猪俣 司 さま（株式会社LIXIL）

物流領域で主に輸配送、貿易業務に関するシステム企画、構築のリーダーを担当。

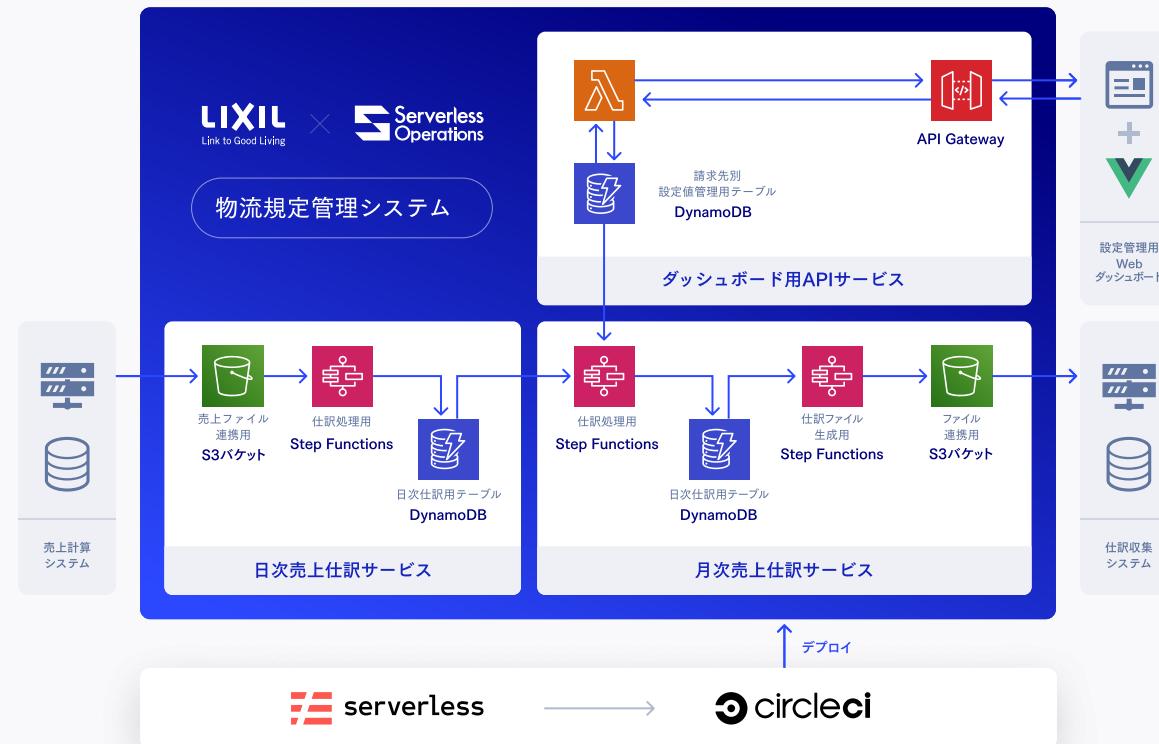
和田 修一 さま（株式会社LIXIL）

主に物流系の各種システム開発を担当。

Facilitator

阿部 文人 | 株式会社necco
CEO・クリエイティブディレクター・デザインエンジニア





お客さまの課題

- システムのサーバー運用、老朽化による障害対応など無駄な作業工数が多い
- 新しく構築するシステムは、可用性および開發生産性、運用効率を高くしたい
- クラウドサービスを使った開発にチャレンジし、サーバーレス開発の進め方を習得したい

お手伝いしたこと

- 要件の概要を出していただき、一緒に仕様を検討
- フロントエンドのUI設計・バックエンドのAWSアーキテクチャ設計
- API開発のレクチャー、ソースレビューなどの開発サポート(フロントエンドと会計処理部分をServerless Operationsが実装し、フロントとDBを繋ぐAPIの開発をLIXILが実施)
- フロントエンド開発・会計処理システム開発・CI/CDパイプラインの設計、設定
- システムUIデザイン改善(担当:株式会社necco)

オンプレミスから クラウドへと社内の流れが 変化するなか、 「サーバーレスって何？」 からのスタートだった

- Serverless Operationsに依頼したきっかけ
- サーバーレスの導入

Serverless Operationsに開発を依頼した きっかけ、経緯を教えてください。

猪俣(LIXIL) 多様で柔軟、効率的に業務ができるインフラ、可用性、開発生産性、運用効率が高いシステムの構築を行ったかったのでサーバーレスのクラウドサービスでのシステム構築を採用し、開発作法やツールの使い方をしっかり学ぶために実績豊富なServerless Operationsさんにたどり着き、声をかけさせていただきました。

和田(LIXIL) まず弊社の役割分担を説明しますと、私は先ほど言った通り物流系のシステム開発を担当する「開発部隊」に所属しており、原田のようにサーバー・ネットワークの環境構築や運用を行う「インフラ部隊」とは別部署になります。私はこれまでサーバーレスについては無知でした。今回もオンプレミスでの開発をするつもりでインフラ部隊に相談したのですが、会社でも「これからはクラウドサービスをメインに使っていこう」という大きな流れがありました。

猪俣 弊社には自社サーバーが約4,600台あるのですが、数が多くすぎて、これ以上作るとメンテナンスや運用できなくなってしまうため本流になりつつあるクラウドサービスで開発できないかという話になりました。

和田 はじめは、それがどういう意味なのか分からなくて。社内にはサーバーレスで開発できる体制がなかったんですね。引き受けくれそうな会社を何社かあたった中で、原田のお付き合いのあるIT会社さんからServerless Operationsの堀家さんを紹介してもらいました。

堀家(Serverless Operations) はい、そんな経緯でLIXILさんに打ち合せに行きました。和田さんと初めて話したとき、サーバーレスの話を聞いてポカンとしてましたね(笑)

和田 全く分かりませんでしたね。サーバーのない意味が理解できませんでした。

弊社でもAWSを使ったシステムはいくつかありますが、どれも仮想サーバーを立てていますから。ただ、私はユーザーさんが使いやすければいいと思っているので、中身はどっちでもいいかなという考えでした。

なるほど。では、LIXILさんのシステムで
サーバーレスを導入したのは初めてですか？

猪俣 以前簡単なシステムをひとつ作ったくらいなので、私のところでは今回で2度目です。今後は確実にサーバーレスが増えそうですね。オンプレミスで開発する場合、環境準備だけでも人手も時間もかかってしまいますし、IT全体のコストを下げられるため、「オンプレミスではなくクラウド」という流れになっているんです。



お客さまとともに つくり上げていく 「共創型開発」で、 技術力不足をカバー

- システムの具体的な内容
- 「共創型開発」で感じたこと
- スキルセットに合わせたレクチャーと作業分担
- 開発の方法について

Serverless Operationsさんに相談されたとき、システムの具体的な内容は決まっていましたか？

和田 はい。ざっくりとしたイメージがあり、猪俣が描いた構成図を用意していました。最初の打ち合わせでは、その資料を堀家さんに見せながら説明しました。

堀家 それをサーバーレスの構成に置き変えてきましたね。

「共創型開発」という方法で一緒にやってみて感じたことを教えてください。

猪俣 堀家さんのやり方がいいなと思ったのは、フランクなコミュニケーションで、われわれLIXILの目線に合わせてやりとりしてくれたところです。短期間の依頼で無茶を言ったかもしれません、私たちの要望にも前向きに迅速に応じてくださり、真摯に取り組んでくれました。技術力も非常に高く、大変助かりました。

和田 主に担当していた弊社側のメンバーは初心者でしたので、まずは作業の基本からレクチャーをいただき半分教育ももらったので助かりました。

堀家さんはどんな風にレクチャーしたんですか。

堀家 はじめは「AWSとは」「サーバーレスとは」という話から入りました。LIXILさんメンバーは各々のスキルセットがバラバラだったので、一人ひとりに合わせてレクチャーしました。

週に1、2回くらいLIXILさんにお邪魔して、手を動かしながら進めました。食堂でも作業しましたね。

和田 話を聞いているだけでは覚えないので、実際に触らせてもらいました。それはよかったのですが、どの作業にどのくらい時間がかかるかが分からなくて。

ひとつひとつの作業に対して、当初の見込みよりも時間がかかってしまいました。あふれた分はServerless Operationsさんのチームに作業してもらいましたね。

堀家 そのあたりはバランスをとりながら進めてきました。納期がありますから。

開発はLIXILさんが担当したのですか？

今まで同じような方法で

開発されたことはありますか？

和田 分担して開発しました。もともとできるメンバーもいたので、分担した作業のなかからバラして進めてきました。このような開発方法はあまり経験がないですね。よくあるのは、開発を一括でお願いして、できあがったら説明だけしてもらうという形です。まるまる新規で、かつ技術的に未知のシステムを一緒に開発したケースは珍しいです。

難しく苦労もありましたが、これから作っていくシステムのことを考えると無駄ではないと思っています。新しいシステムに生かしていきたいですね。

品質を保ちながら素早く 本番へコードを 調達するための仕組みを、 サーバーレスで実現

- 今後の運用スケジュール
- サーバーレスにして感じたメリット

これからどんなスケジュールで 運用などを進めていく予定ですか？

和田 今はまだテストをしている段階で、これから本番で運用できる体制を整えていき、4月から本番運用となります。

このシステムでユーザーさんが入力するのは、主にマスタの登録です。本番に向けて、まず最初に数千件の登録を行います。ひとつひとつ画面に入れていたら大変なので、データをもらって一括で登録する予定です。

4月以降は追加や変更、修正が出てきそうなのでそれに対応します。とにかくマスタが肝ですね。今はメインユーザーが1名、全国の営業を取りまとめてくれることになっています。

サーバーレス アーキテクチャにして 良かったことを教えてください。

猪俣 サーバーを立てるための時間が短縮されたことはかなり大きいですね。今まで着手までに3ヶ月待ち状態でしたから。また運用・保守の省力化、システム開発生産性向上によりコスト削減につながることを今後期待します。

和田 これまで仮想サーバーを立てて、オラクルをインストールする作業などがありました。それも必要なくなり、さらに手間が減りましたね。今回はLambdaを使ったサーバーレスという形で実現しているので、一番手間のかからない作り方になっています。

堀家 そうですね。特に今回ダッシュボードのバックエンドとして作ったAPI GatewayとLambdaの仕組みは、それをウェブのフロントにあてはめればウェブサイトになるし、iPhoneアプリのようなものを別に作れば、すでにAPIがあるのでバックエンドの開発が最小限になります。

ひとつのバックエンドをサービスとして切り出し、あとはフロントインターフェースが変わったとしても対応しやすくなります。

和田 それはAWSの研修でも先生が話していました。そういう話は聞いても、まだ触ってないので実感がなくて。

堀家 今回作ったInfrastructure as Codeでは、Lambdaの構成はこうするとか、API Gatewayの構成はこうする、というのをServerless Frameworkで定義しています。本番にあげる際、今回のような構成にしておくと、毎回確実に同じ構成になることが保証されているんです。APIもそうですし、バッチや料金計算も同様です。

今回はCircle CIを使ってCI/CDパイプラインを構成しています。料金計算は細かくユニットテストを書いています。バグが出た際はそれを再現するテストを書いて修正していくことで、将来的な同じバグの再発のリスクを防いでいます。このように運用していくことで計画的に不具合を潰していくことができます。

テストコードで品質を改善しつつ、寡等性を保ちながら素早いサイクルで本番にコードを調達する。そういう仕組みは、サーバーレスだと非常にやりやすいですね。

今回のプロジェクトでサーバーレスを取り入れた大きなメリットの一つだと思います。

さらに自走力をつけて 今後のシステムづくりに 生かしていきたい

- これから改善したい点
- 今後の向き合い方

これからも一緒に開発をされることがあると思いますが、改善していきたいことはありますか？

猪俣 Serverless Operationsさんには情報交換会などを実施していただき、社内のスキルアップを図っていかなければありがたいですね。弊社だけで進めていると、新しい知識や技術はなかなか入ってきません。

特にクラウドサービスは日進月歩で、新しいサービスがどんどん出てきますし主流も変わってきます。情報交換会やワークショップなどを開催したら、もちろん弊社も情報を提供しますし、相乗効果が生まれてお互いスキルアップにつなげられるはずです。

和田 今後もシステム構築が必要になるはずです。まだまだ勉強が必要なので、堀家さんには引き続きご支援いただきたいですね。

堀家 はい、もちろんです。サーバーレスでシステムを開発するときに考えているのは、「クライアントさんには知識や技術を習得してもらい、内製できるようになってほしい」ということ。僕たちが開発を一括で行ってクライアントさんがシステムを使うだけでは、お互い成長できません。それが共創型開発のメリットの一番大きな部分ですね。のために一緒に作るということにこだわっているので。

和田 個的には時間が取れておらず、まったく吸収しきれていません(笑)。原田や他のメンバーは吸収できているので、私はその人たちから少しづつ吸収したいと思っています。

最近AWSの研修を3日間受けたのですが、去年の夏の段階で受けたければよかったです。そうすれば当時堀家さんが言っていた内容が理解できたかもしれません。

研修のおかげで今はだいぶ理解されているようなので、次が楽しみですね。

和田 自分はたった3日間の研修を受けただけで、モヤモヤしていたものが晴れました。今後新規のシステムを作る人がAWSを使うなら、みんな研修を受けたほうがいいですね。

猪俣 Serverless Operationsさんと一緒に仕事をさせていただき、ノウハウが身につき自走していくようになりました。LIXILでは、今回のように新しい技術や知見を積極的に取り入れ、自分たちで考え最適な環境に発展させていく、そういうクリエイティブな人材を増やし最強のプロ集団を作っていくたいと考えています。 





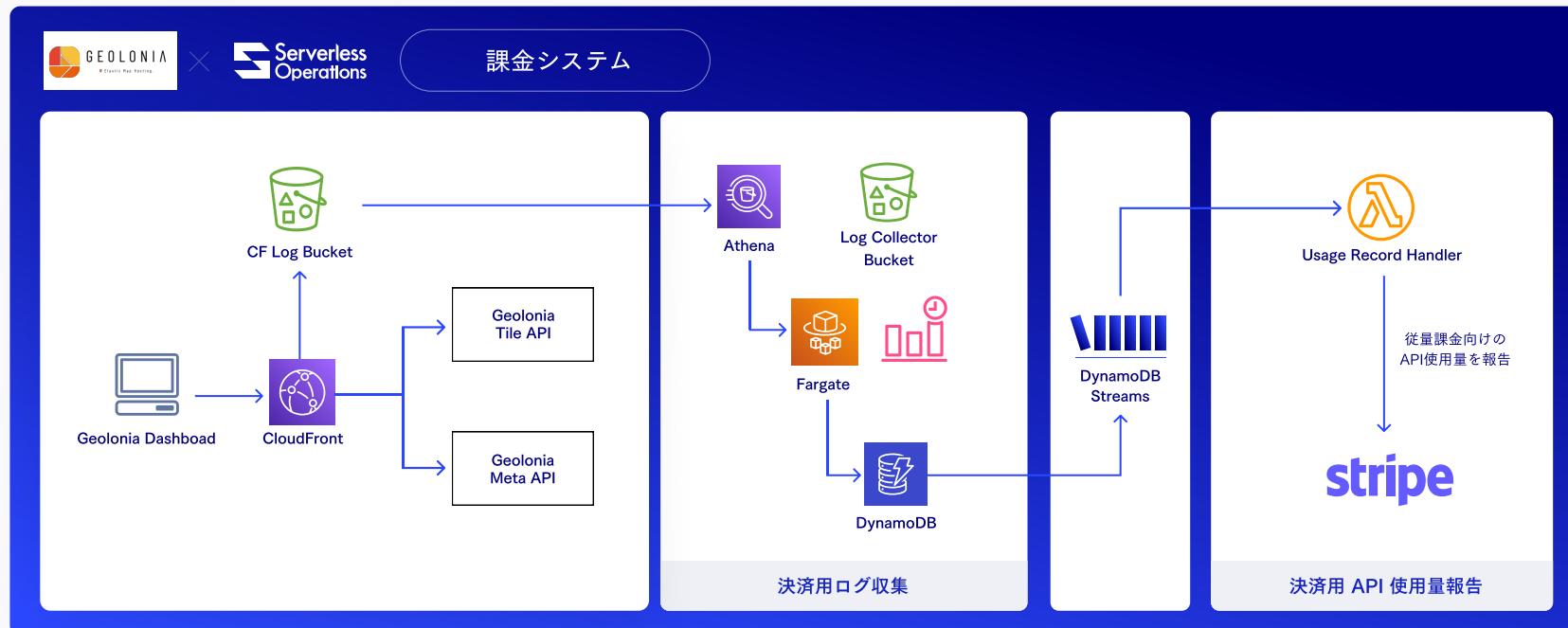
株式会社Geolonia

ウェブ系サービスに必要なAPI課金システムを サーバーレスで構築、スタートアップ企業のスピード感と コストとニーズを満たすためAWSとStripeを活用

株式会社Geolonia様向けにAPI課金用の決済システムを開発しました。サーバーレスアーキテクチャ及び決済のSaaSであるStripeを利用して実現しています。これらのサービスを使用せずに構築すると半年以上はかかるであろうシステムをたった2ヶ月というスピードで開発を行いました。サーバーレスアーキテクチャの採用することで、新しいビジネスを早く、安く、シンプルに開始することができた事例です。

お話をうかがった方
宮内 隆行 さま（株式会社Geolonia）
株式会社Geolonia 代表取締役 CEO





お客さまの課題

- 地図のサービスを展開するうえで、地図のデータは非常に大きくサーバーをメンテナンスしてサービスを提供するやり方は避けたかった
- すべてをサーバーレスにしようとすると、技術的な制約事項などで不安な部分が多くあった
- 課金システムは不具合があるとお客さまにも影響が大きいので、信頼できる外部の事業者に協力してもらいたかった

お手伝いしたこと

- 課金システムをサーバーレスで実装するためのAWSサービス及びサードパーティのサービス(Stripe)の選定
- 選定したサービスを元にしたアーキテクチャ設計やDB設計
- Geoloniaの課金体系を実現するためのStripeの設計
- Geoloniaの開発チームと分担しての実装とテスト

サーバーレスにしたいから 選んだ開発会社

- はじめに
- Serverless Operationsへのアプローチ
- 地図サービスに必要な課金システム

はじめに

株式会社Geoloniaは、提供している地図サービスの本格的な事業化に向け、課金システムの整備に着手。その設計と構築にあたって、Serverless Operationsにご依頼いただきました。

地図サービスの課金システムの開発にあたって、 GeoloniaからServerless Operationsへどのようなアプローチがあったのでしょうか。

宮内(Geolonia) そもそも、弊社が地図のサービスを展開するうえで、サーバーをなくしたかったんです。これまでのようサーバーを置いて、それがすべてのトラフィックをさばくやり方だと、地図データではインフラが大変なことになる。100GBを超えるような、すごく大きなデータなので。でも、サーバーレスでやろうとすると、わからないこと、不安なことがたくさんあります。そこで堀家さんに相談しました。

いろんな開発会社があるなかで、Serverless Operationsだったのは、以前に堀家さんと一緒に仕事をしたことがあって、彼が「Serverless Framework」というオープンソースプロジェクトで活躍されていることを知っていたからです。だから、サーバーレス開発のことなら、堀家さんに聞くのが日本で一番確実だなと。

課金システムの以前に、そもそもサービス開発から一緒にされていたんですね。

宮内 はい。Serverless Operationsの協力もあって地図のサービスが形になってきたので、サービス開始に必要な課金システムを作ることにしました。自分たちで作ることも考えたのですが、社内のエンジニアが少ないのと、課金システムは不具合があ

るとお客様にも影響が大きいので、信頼できる外部の事業者に協力してもらうことにしました。

それに、特定のエンジニアだけにまかせるとブラックボックスになりがちなので、係わるエンジニアを増やして多くの目に触れる状態にしたかったんです。そこで、もとの地図サービスを一緒に開発した、堀家さんに課金システムもお願いしました。

堀家(Serverless Operations) 宮内さんからは最初「地図サービスの課金システムを作りたいんだけど、どうしたらいいかな」という、依頼というよりも相談みたいな形でお話がありました。そこで、どのくらいの予算かうかがったうえで、開発における役割分担を切り分けを考えました。

Serverless Operations側が主に設計やバックエンドの開発などを担当し、Geolonia側が主に実装や展開(CI/CD)、それにダッシュボード(利用者側のインターフェイス)の開発などを担当するという体制を提案して、それで進めることになりました。

インフラに Amazon Fargate を 選んだ理由

- 開発の体制
- 課金システムの仕組み
- Fargateを使うメリット

具体的にはどういった体制で開発したのですか。

堀家 Geoloniaからは、鎌田さんというエンジニアが仕様決定や取りまとめ役として立ちました。Serverless Operationsからは、僕が設計とサービス選定、デモ開発を担当しました。僕から課金システムに利用するクラウドインフラや決済サービスを提案して、実際に動くデモを作って、鎌田さんがそれを見て決定する、という進め方でした。

後半からは、Geoloniaから小林さんというエンジニアも参加して、お客様にクレジットカード情報の登録してもらうダッシュボードの開発をしてもらい、Serverless Operationsからは私と、途中でもうひとりのエンジニアに交代してバックエンドを作りました。エンジニアは最大で4名がコミットして、期間は2020年の10月に開発を始め、2021年の2月いっぱいで終わりました。

1月はほとんど作業をしなかったので、実質的には3ヶ月です。

課金システムは、どのような仕組みなんでしょうか。

堀家 Geoloniaの地図サービスは、利用者がAPIを呼び出していくもので、サービスのアクセスログはAmazon S3(AWSのストレージサービス)に保存されています。そのアクセスログからAmazon Athena(S3上のデータをSQLで操作できるAWSのサービス)で必要なデータだけを取り出し、AWS Fargate(AWSのコンテナ向けサーバーレスコンピューティングエンジン)に設置したプログラムで集計処理をして、結果をAmazon DynamoDB(AWSのNoSQLデータベースサービス)に保存します。

地図サービスのお客様への請求と決済には、Stripe(ネット向けの決済プラットフォームを提供する事業者)を使っています。あらかじめStripeに請求先やサイクル、料金体系、締め日などを設定しておくと、DynamoDBに保存した集計結果を元に月次の料金を計算して、請求日に請求書を自動的に発行してくれます。

サーバーレスで開発するには多くのサービスがあり、AWSだけでも複数の選択肢があります。今回、Fargateを選択した理由はなんでしょうか。

堀家 コンピューティングエンジンに関しては、AWSに3つのサービスがあります。Lambda、Fargate、EC2と選択肢が3つあって、後にいくほど手間が増えていくんです。その替わり、できることの柔軟度が上がっていく。

Lambdaは、考えることが少なくて済みますが、ひとつのプログラムが15分以上動かないんです。だから、15分以内で処理を終わらせるようにしないといけない。逆にEC2は、CPUをどれにするか、どのくらいの数を立ち上げるか、メモリをどうするか、とか初期設定で考えないといけないことがたくさん増える代わりに、Lambdaのような制限がなくて何でもできるんです。Fargateは、その中間でEC2ほどではないけれど初期設定が必要だけど、Lambdaほどの制限はない。

僕は、サーバーレスで開発する場合、基本的にまずLambdaを検討するんです。今回の場合はログの集計処理で、ログはユーザー数や利用状況によってデータ量が大きく変動するので、データ量が大きい場合は15分を超てしまう可能性がある。並列処理にすると工夫することで対応できますが、データ量で処理時間が変化するので15分を過ぎるリスクが常にあります。それを考慮して、今回はFargateにしました。

Fargateも内部は結局EC2ですが、あらかじめサービスの側である程度設定を上手くやってくれます。コンテナをFargateにポンと投げれば、それだけで動いてしまう。だから、普通にウェブでサービスを作る場合はFargateが一番多く選ばれていて、最初からEC2でやるというケースは少ない気がします。

Stripeは開発者が使いやすい決済インフラ

- 決済サービスにStripeを選んだ理由
- 開発者にとってStripeの長所とは
- 開発にあたり重視したポイント

決済サービスについては、なぜStripeを選んだのですか。

堀家 決済サービスについては現状、Stripe一択という状況なので、特に迷うことはありませんでした。以前は、他にもサービスがあったのですが、そこも終了してしまったので、本当に選択肢がなっています。ただ、開発者にとってStripeはとても使いやすいんです。課金を処理するためのAPIが非常に使いやすかったり、サブスクリプションから従量制までいろんな課金の仕組みに対応できたり、開発に必要なドキュメントの整備であったり、サポートが体制が充実していたりするんです。

具体的にどういった点が、開発者にとって使いやすいのでしょうか。

堀家 サブスクリプションサービスで課金をする上で必要な機能が揃っているだけでなく、途中でプランを切り替えた場合の差額の処理といった、自分たちでゼロから考えて実装するにはかなり大変なところも、あらかじめ用意されているところですね。そうした場合、締め日に合わせて日割りでやるのか、それとも申し込み日に合わせて締め日を変えるのか、とか考えることがものすごく多いんです。それをStripeなら、サービスの側で上手くやってくれる。だから、Stripeしか選択肢がないというのは、ネガティブな意味ではなくてポジティブな本音です。Stripeに依存してベンダーロックインされているところもないです。ユーザーCommunityも熱心に活動していて、ミートアップなども定期的に開催しているようです。こういうCommunityがあるプロダクトは、個人的に信頼できると思っています。

宮内 地図サービスは、APIを介して利用するサービスなので、最初から漠然と「これとこれを数えておけば課金ができる」というイメージはあったんです。

そこを数える部分、ログの集計の部分をお願いしたんですけど、そこがすごいうまく動いているので、結果的にはどんな課金モデルでも対応できることになったので、良かったですね。

課金システムの開発にあたって、重視したポイントはなんでしたか。

堀家 開発する上では、集計などでエラーが起きたときにどうするか気を付けました。エラー対応にはリカバリーのセオリーがあるので、それに沿った開発を行っています。ログ集計系については、先ほど述べたようにログの大きさでエラーが起きないように、コンピューティングエンジンを選んでいますし、決済系についてはStripeの信頼性が非常に高いので、そこに投げるデータさえ間違ってなければ問題ないです。クラウドサービスでの課金は、よく利用者側のミスで莫大な金額が請求されてしまうことがありますよね。そういうことがないようになっています。

宮内 課金で怖いのは、仕組みが間違っていてお客様に多く請求してしまったり、少なく請求してしまったりすることです。少ない場合は、しょうがないですが(笑)。ただ、多く請求することは信用を失いますし、返金することになるとその手間も掛かる。それに、開発を相談した時点では、地図サービスの料金体系がまだ確定していなくて、流動的だったんですよね。地図サービスには、グーグルマップという大きな競合がいるので、その料金体系は常に意識しないといけないんです。だから、正式な料金体系は、つい先日にやっと決まったばかりです。

課金システムを開発するのに料金モデルが決まらないので、サービスを開始するまで何ヶ月も開発に付き合ってもらうことになると思っていました。それが、思っていたよりもすんなりといけたのは良かったです。課金モデルも、Geolonia側からダッシュボード上の設定変更で対応できるので楽で助かります。





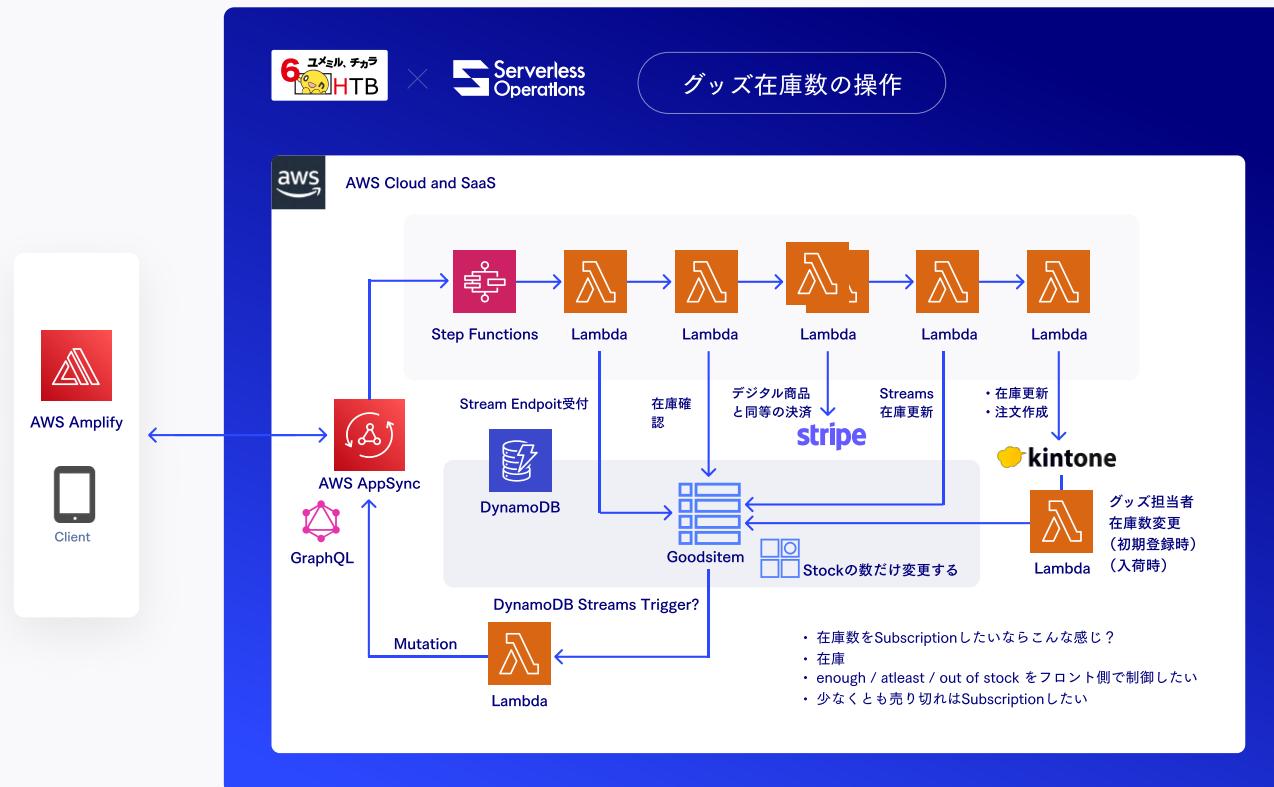
新たな収益源として独自サービスの内製化を目指した地方局が求める「完全サーバーレス」の実現を支援

北海道テレビ株式会社様向けに「onライン劇場」という、有料映像配信サービスとECとが融合したオリジナルサービスの開発支援を行いました。2020年に新型コロナウィルス感染症の拡大によって、各種のオフラインイベントが潰れてしまったので、その代替施策として考えられたサービスです。

開発は北海道テレビ株式会社様の開発チームメインで行い、弊社ではReactを使ったフロントエンドの開発支援、DynamoDBを使ったデータ設計の支援やAuth0とAWSのAppSyncやLambdaを使用したインテグレーション方法の支援を行いました。

お話をうかがった方
三浦 一樹さま（北海道テレビ株式会社）
コンテンツビジネス局ネットデジタル事業部





お客さまの課題

- 部署にサーバーサイドエンジニアやネットワークエンジニアと呼ばれる人材がおらず、システム構築の際は出来る限りコーディング以外のことをしたくなく、サーバーレスであることが自然と必須の要件となった
- サーバーレスでサービスを構築するにあたって、内製だけでは技術的に不安な部分があり、プロフェッショナルによる開発支援が必要であった

お手伝いしたこと

- お客さまで設計したAWSやAuth0のアーキテクチャレビュー
- フロントエンドの設計やソースコードレビュー
- DynamoDBを使った在庫管理の仕組みの設計
- 週1回のミーティングを通じた各種開発に関する相談

開発する上で サーバーレスが絶対条件

- はじめに
- コンテンツビジネス局が展開するサービス
- 開発支援を依頼したシステム

はじめに

北海道テレビ放送株式会社(以下、HTB)は、札幌に本拠地を置く北海道を対象とした放送局です。東京や大阪に拠点を置き全国で放送される番組を作りいわゆるキー局とはことなり、地域に向けた番組作りを行っていますが、そうした番組のひとつ「水曜どうでしょう」は徐々に人気を獲得し、今では全国的に知られるまでになりました。

同局の三浦一樹さんは、コンテンツビジネス局ネットデジタル事業部に所属し、同社のストリーミングサービスやECサイトの開発や運営を主導する立場。しかし、三浦さん自身もチームのメンバーもエンジニアではないため、サービス開発ではしばしば行き詰まることがありました。

そこで開発支援をServerless Operationsに依頼しました。



HTBは放送局ですが、三浦さんはそこでどのようなサービスを提供されていますか。

三浦 コンテンツビジネス局のネットデジタル事業部というところに所属しておりますので、コンテンツビジネス局全体が放送外収入を支える事業部というイメージですね。放送局は企業などに広告枠を買っていただいて、CMを流すことが主な収益になるんで

すが、私の部署はそれ以外で収益を上げるところです。

例えば、弊社には「onちゃん」というオリジナルキャラクターがいるんですが、そのグッズを販売したり、「水曜どうでしょう」のグッズを売ったり、独自のVODサービスで弊社が製作した番組をストリーミング配信したり、他局が弊社のコンテンツを二次利用する時の権利処理とかをしています。そのなかで私は開発を担当しています。

今回、Serverless Operationsには開発支援を依頼されたそうですが、どのようなシステム開発に取り組んでいたのですか。

三浦 「オンライン劇場」という、有料での映像配信とグッズ販売のECと一緒に始めたサービスです。2020年に新型コロナウイルス感染症の拡大によって、各種のオフラインイベントが潰れてしまつたので、その代替施策として考えたものです。

もともと2019年に「水曜どうでしょう祭 FESTIVAL in SAPPORO2019」というオフラインのイベントを実施した際、一緒に有料での動画配信の仕組みを作っていたので、それを活用したかったんです。その動画配信の仕組みをテコ入れして、配信ページにShopifyの購入ボタンを貼り付けただけのような形で、最初のオンライン劇場がスタートしました。

その当時、動画を見ながら関連グッズを買えるというサービスが見当たらなかったので、自分で作るしかないと思ったのが、アイデアのきっかけです。結果的にある程度、うまくいったので、社内を説得して開発費が出ることになり、不満だった部分を作り直すところから、Serverless Operationsに開発支援で入ってもらいました。

冷や汗をかいた過剰販売を二度と起こさない

- Serverless Operationsを選んだ理由
- 具体的に行った支援

なぜ、最終的にServerless Operationsを選んだのでしょうか。

三浦 最初のVODサービスの開発を依頼したところも含めて、いくつかの開発会社に相談をしていて、Serverless Operationsもそのひとつでした。もともと堀家さんとは、あるユーザー主体の勉強会で知り合いました。勉強会のライトニングトークで動画配信について話したら、その後の懇親会で堀家さんが声をかけてくれて、それがとても印象に残っていたんです。

今回の開発にあたって大前提だったのは、サーバーレスで行うことでした。そのわけを私はよく冗談で「宗教上の理由で」といっていますが、それというのも我々の部署にはサーバーサイドエンジニアもネットワークエンジニアもいないんですね。だから、できるだけ開発するときは、コーディング以外のことをしたくないので、サーバーレスという条件は絶対的なものなんです。

私自身、HTBの入社当初は「放送監視」という、放送されている番組を一日中見るという仕事で、それまで開発経験はありませんでした。他のメンバーも電波塔で電源工事をしたり、壁の補修でモルタルを塗ったりと、まったくITとは無関係の経歴の人間ばかり。そういうメンバー達と一緒に、勉強しながら開発しているんです。だから、できるだけ余計なことはしたくないのですが、開発会社に相談すると、ほとんどがサーバを立てることが前提なんですね。僕らは、クラウドの中には極力、なにも置きたくないのに、すぐにサーバを立てることが必要な提案をしてくる。まるでサーバーレスという要望がわがままなことかのように言われると、残念な気持ちになってしまいます。

そこで、AWSのパートナーになっている会社にもいくつか話を聞きましたが、フロントもバックエンドもできるところや、Auth0などの外部のAPI連携に詳しいところとなるとなかなかなくて、そのような別無しに相談に乗ってくれたところはServerless Operationsだけだったんです。

Serverless Operationsは、具体的にどのような支援を行ったんですか。

金(Serverless Operations) 開発にあたりいろいろな質問を受け、それに対して回答するという形で支援しました。最初は不定期でしたが、途中からは週1回の定例会議を設定して、そこまでまとめて受け付けるようになりました。

一番印象的だったのが、EC部分の在庫管理の仕組みのところですね。グッズの販売で在庫管理が上手くいかず、在庫数以上の販売を行ってしまうというトラブルに対し、DynamoDBの条件付き書き込みという機能を使って解決しました。

三浦 2020年の10月に、オンライン劇場で「水曜どうでしょう」の福袋を限定販売したんですね。結果的に400個くらいが数分で売り切れて喜んでいたんですが、販売後にShopifyの管理画面を開いたら、在庫がゼロではなくマイナス10になっていたんです。つまり、実際に在庫している数よりも10個も多く売ってしまったんです。調べたら、Shopifyが標準で用意している決済方法なら問題ないはずが、JCBに対応するための別の決済方法を使ったのが原因でした。過剰販売分はキャンセルてしまえば対応も簡単でしたが、お客さんが楽しみにしているからそれはしたくなかった。結局、あちこちにお願いして回って、なんとか商品が確保できましたが、めちゃくちゃ肝が冷えた出来事でした。

それで、ECでの発注処理のデータベースの更新処理も自分たちでコードを書くことにしたんですが、絶対に在庫がマイナスにならないようにするためににはどうしたらよいのか、いろいろやってもできなかったんです。

金 そのタイミングで相談を受けて、僕も1週間くらい悩んで、DynamoDBの条件付き書き込みを使えば上手く行くことに気がつきました。DynamoDBの条件付き書き込みは、トランザクション処理ではなくて、単一レコードに対する排他制御です。DynamoDBにはそもそも楽観的排他制御っていう手法はあるん

蓄積したノウハウで将来は他の地方局の支援にも

- DynamoDBを使うメリット
- TV業界にありがちなスパイクに対応するために

ですが、いずれにしても結果整合にフォーカスした対応なので、今回のような在庫だと単一レコードに対する強い一貫性のある排他制御が必要です。そこで、多少のパフォーマンスは犠牲にして、強い一貫性の書き込みモードを維持しつつ、在庫数をチェックしながら書き込みを行うということをしています。SQLが使えるリレーションナルデータベースなら簡単にできることなんですが、DynamoDBの本来の使い方からするとアレな機能です。ただ、古くからある機能なので、それを使ってサンプルコードも作成して上手く行きました。

今回の要望をDynamoDBで実現するのは、そんなに珍しいケースなんでしょうか

堀家 一般的にはリレーションナルデータベースを使った方が簡単にできます。ただし、AWSでリレーションナルデータベースを使うにはAmazon RDSというサービスになるんですが、これを利用するにはAWSのなかでネットワークの設定が必要になるので、考えることが一気に増えます。DynamoDBならネットワークのことを一切考えなくてよいですから、すべてサーバーレスで開発するなら、Amazon RDSの利用は躊躇します。結局、レイヤーの違うサービスなんですね。

三浦 SQLなんてメンバーの誰も書いたことがないし、まずリレーションナルデータベース自体をよく知らない。DynamoDBのようにAPIのレスポンスだとイメージが付くんんですけど、SQLだとよくわからないんですよ。

堀家 強力な整合性のある読み込みを行うケースは、リレーションナルデータベースであってもあまりない要件です。実際には、リレーションナルデータベースでも、リードレプリカが存在する場合に書き込みしたデータが全てのレプリカに完全に伝搬するのに少しの時差があるんですけど、そこの時差で困るケースってあまりないですか。

ECでここまでトラフィックが殺到したのは、「水曜どうでしょう」のような人気コンテンツだからですよね。

三浦 ただ、テレビ放送がきっかけになってウェブサーバーが落ちるということはよくある話で、TV業界は「スパイク」を見やすい業界と良く言われています。AWSの認定試験でも、「TVCMを受けてスパイクがたくさん来る場合はどうするか」という問題がよく出されます。

単純なコンテンツサーバーだけではなくて、ECの場合でもDynamoDBを使ったサーバーレスで構築できて、スパイクにも対応できるというのは、いざというときに切れるカードを持てるようになったので、すごく大きいことです。最終的には、TV局でのこうしたウェブ周りのシステムのノウハウをつかって、他の地方局をお手伝いできるようになれたらしいと思っています。そのためにも、金さんのようなエンジニアの考え方や行動指針をもっと身につけて、最終的には誰にも頼らずに自社内だけで考えたり、作ったりできるようになっていけばいいと思っています。まだまだ、金さんには頼ることが多いですが、すごく成長させていただいて、その目標に向けた大きな一歩を踏み出せたのはServerless Operationsのおかげです。 

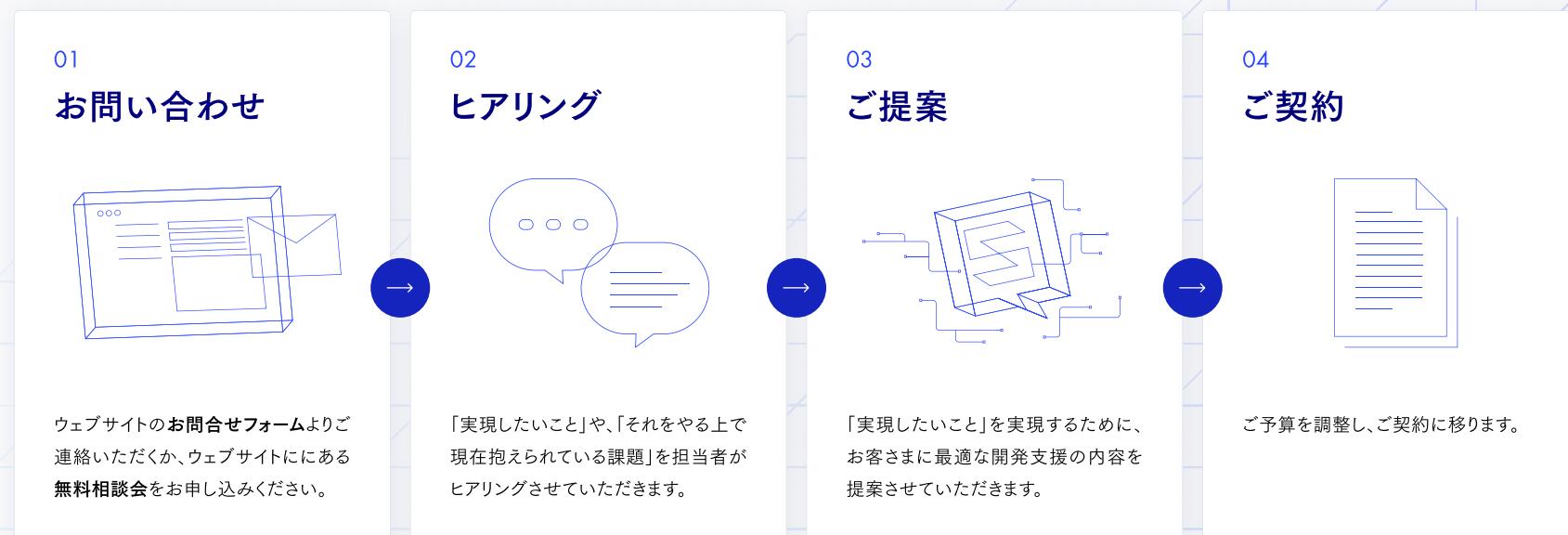
Contract

ご契約の流れ

Serverless Operations

ご契約までの流れ

ウェブサイトのお問い合わせホームまたは無料相談会にて、
まずはお気軽にご相談ください。



プロジェクトの進行について

プロジェクトのキックオフからリリースまで、
お客様のチームに入り、課題を共に解決します。



Member

メンバー紹介

Serverless Operations

代表取締役／CEO

堀家 隆宏

Takahiro Horike



大学院を卒業してからソフトウェアエンジニアとして新卒で就職をしてから10年のキャリアになります。もともと、勤めていた会社でAWSは利用していたのですが、2014年にAWS LambdaとAPI Gatewayがリリースされ衝撃を受けました。私はこれらのサーバーレスのテクノロジーにのめり込みました。そしてServerless Frameworkというサーバーレスの開発を助けるOSSのコントリビューションや国内でのミートアップ開催などのコミュニティ活動を通して、サーバーレスの開発をより簡単に、より身近なものにするための活動をするようになりました。その結果、コミュニティ活動だけでなく自分自身でも事業を興すようになりました。現在に至ります。

主な経歴・仕事歴

- 2008 関西大学大学院電気工学科修了
- 2008 株式会社神戸デジタル・ラボ入社
- 2012 株式会社デジタルキューブ入社
- 2018 フリーランスとして独立
- 2018 株式会社Serverless Operations創業

活動実績

- 2016 JAWS DAYS（東京）2016 登壇
- 2017 Serverlessconf（東京）2017 登壇
- 2019 AWS Meetup Amsterdam（オランダ）2019 登壇
- 2019 AWS DevDay Tokyo（東京）2019 登壇
- 2019 ServerlessDays Fukuoka（福岡）2019 登壇
- 2020 JAWS DAYS（東京）2020 登壇
- 2020 Serverless Meetup London（イギリス）2020 登壇
- 2021 Serverless Days China（中国）2021 登壇

取締役／COO

金 仙優

Sonu Kim



韓国出身で大学から日本へ留学、新卒でソニーに入社、主にB2C領域におけるAWSサーバーレスアーキテクチャを用いた設計・開発を得意として、ソニーグループ内での様々な製品・サービスをリリースした実績があります。ウェブ技術全般における豊富な知識を持っており、グローバルなプロジェクト推進と開発経験を生かして、今後は Serverless Operations での新サービス開発とグローバル活動を進めていきます。

主な経歴・仕事歴

- 2014 早稲田大学教育学部卒業
- 2014 ソニーグローバルソリューションズ株式会社入社
- 2014 - 2020 数々のソニー製品及びIoTサービスのアプリケーション開発に従事
- 2020 株式会社Serverless OperationsのCOOに就任

活動実績

- 2017 Serverlessconf（東京）2017 登壇
- 2018 Serverlessconf（東京）2018 登壇
- 2018 SPAJAM 東京予選 優秀賞
- 2018 Node学園祭（東京）登壇
- 2020 Serverless Meetup Japan Virtual #4 登壇
- 2021 Stripe Meetup Online Meetup #2 登壇

Partner

認定パートナー

Serverless Operations

Serverless Operationsは、 さまざまなサービスでパートナー認定されています。

パートナー認定されたプロに相談することで、お客さまの状況に合わせた最適なサービスの導入ができます。
導入後のイメージや不安、疑問点を相談することもできますので、まずはお気軽にお問い合わせください。

	<p>内製化支援推進 AWS Partner ユーザー企業の内製化を支援するための「AWS内製化支援推進パートナー」に認定されました。</p>
	<p>AWS Lambda Partner AWS Partner Network (APN)の セレクトコンサルティングパートナー、 AWS Lambdaパートナーの認定を受けています。</p>
	<p>ISV Partners AWSの新たなパートナープログラムである 「ISVパートナーパス」に認定されました。</p>
	<p>APN Select Consulting Partner AWSのAPNセレクトコンサルティングパートナーに 認定されました。</p> <p>OpenSearch Official Partner Amazon OpenSearch Serviceの オフィシャルパートナーになりました。</p> <p>Digitalcube Shifter Partner Network Serverless Static WordPress ホスティングサービス「Shifter」のPartner Networkの一員です。</p>

Company Profile

会社概要

Serverless Operations

Company Profile

社名	株式会社Serverless Operations	決算月	10月
社名（英語）	Serverless Operations, Inc	資本金	100万円
代表	代表取締役 堀家 隆宏	URL	serverless.co.jp
所在地	〒107-0061 東京都港区北青山2丁目7番13号 プラセオ青山ビル3F	Mail	info@serverless.co.jp
設立	2018年11月	事業内容	クラウドサービスを使ったServerlessアプリケーションの開発運用支援

Contact

サーバーレスアプリケーションの設計・開発・保守運用等についてお困りの方は、
ウェブサイトのお問い合わせフォームよりお気軽にお問い合わせください。

サーバーレスで クラウドの価値を最大限に

© Serverless Operations

