100台のサーバー運用からの脱却を目指して
初めてのAWSサーバーレス環境構築の裏側

株式会社ゼンリンデータコム

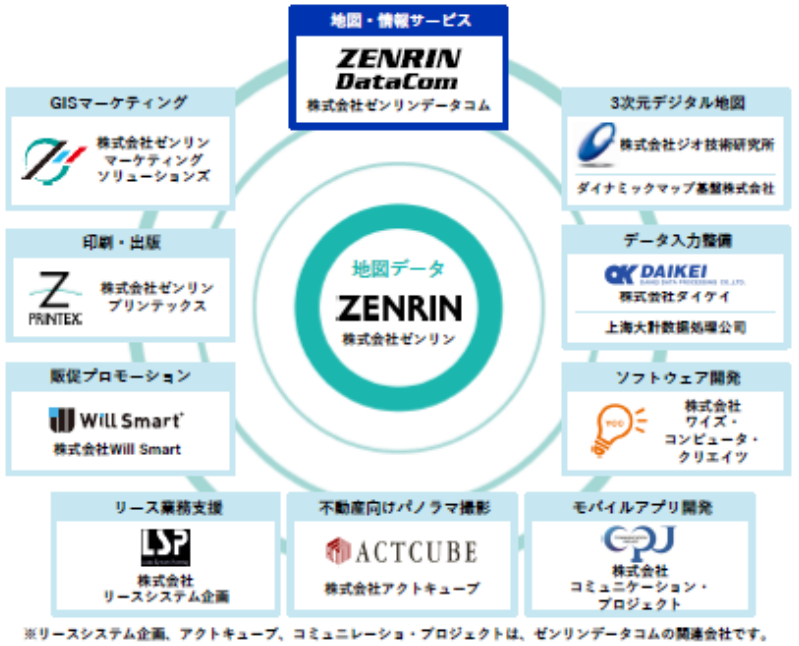
ZENRIN DataCom CO., LTD.

自己紹介

- 名前 ： 新谷 亮人（しんたに あきと）
- 所属 ： 株式会社ゼンリンデータコム
  開発統括本部プロダクト第一開発部 Area Markerチーム
- 担当 ： 自社サービス 店舗案内パッケージ の開発・運用
- 好きなサービス ： AWS Amplify

会社概要

設　　立	2000年4月13日
本　　社	東京都港区芝浦三丁目１番１号 msb Tamachi 田町ステーションタワーN 22階
代表取締役社長	清水 辰彦
資 本 金	2,283,010千円
従業員数	391名（2022年4月1日現在）
事 業 所	田町オフィス　大阪オフィス 名古屋オフィス　福岡オフィス
株　　主	株式会社ゼンリン

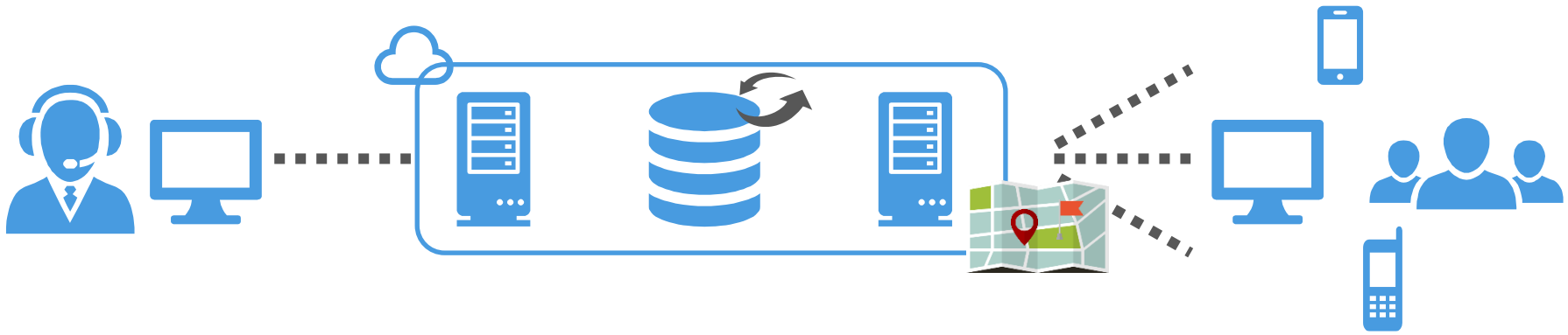


※リースシステム企画、アクトキューブ、コミュニケーション・プロジェクトは、ゼンリンデータコムの関連会社です。

背景：店舗案内パッケージ

- ゼンリンデータコムの「店舗案内パッケージ」サービス

  - お客様がお持ちの店舗情報をホームページ上でご案内・社内業務システム等でご利用いただくことが可能。
  - 各種デバイス（PC・スマートフォン）向けの地図表示に加え、様々な検索機能までがパッケージとなっており、お客様のご要望に沿ったデザイン・機能改修にも対応。
  - お手持ちの店舗情報をCSV形式にて投入が可能、簡単に管理することができる。
    ポイント ＞ 住所から座標情報の付与（ジオコーディング）またその逆も可能



15年以上の歴史

300社以上多くの企業様に採用実績

月間ページビュー約200,000,000

背景：サーバー台数の増加と運用負荷

- 顧客毎の個別要件も増え続け仮想サーバーはついに100台を超える…

  - **OS、ミドルウェアのEOL対応**
    アップデートやバージョンアップの計画と実行が複雑に
    すべてのサーバーで同じバージョンのOSやミドルウェアが動作しているとは限らず、統一性を保つのが困難

  - **脆弱性パッチ適用の工数増加**
    すべてのサーバーで一貫性の確保が困難

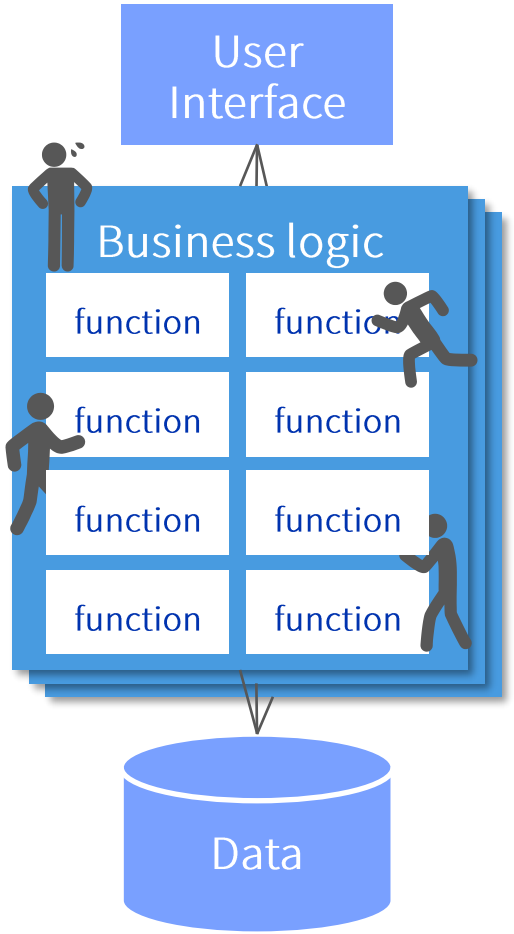
  - **通信障害発生ポイントの増加**
    サーバー台数や通信ポイントが増えると障害の発生確率も上がる。
    単一障害点でサービスが停止しないよう、冗長性を確保する設計が求められる

  - **スケーラビリティでの非効率**
    特定の機能がリソースを多く消費していても全体をスケールアウトやスケールアップする必要がある

背景：モノリスな環境によるエンハンスのスピードダウン

- いわゆるWEB三層構造のLAPP構成

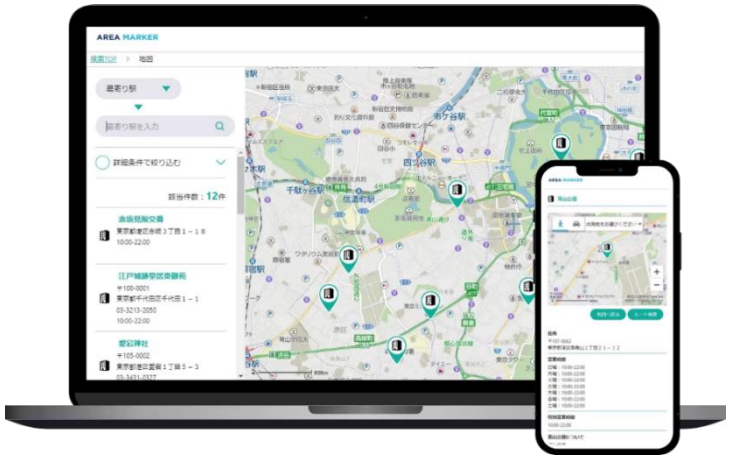
  - **複雑性の増加**
    機能が増えるたび、コードの複雑性が増加
  - **協調作業**
    複数チームが同じコードベースに取り組む場合、作業の調整やマージの競合が頻発
  - **テストの難易度**
    一部のコード変更だけでも、全体のテストを再度行う必要がある場合が多い
  - **デプロイメント時のリスク**
    小さな変更であっても、モノリス全体をデプロイし直す必要
  - **技術的負債**
    既存の技術スタックに縛られ、新しい技術やツールの導入が困難



新しい店舗案内パッケージへ：新機能とサーバーレス化でリニューアル

- 後継版店舗案内パッケージ「Area Marker」

  - モバイルフレンドリーなUI/UX（レスポンシブデザインを採用）
  - AWS サーバーレスアーキテクチャを積極採用し可用性向上
  - 新たな付加機能（危機管理・災害情報確認）

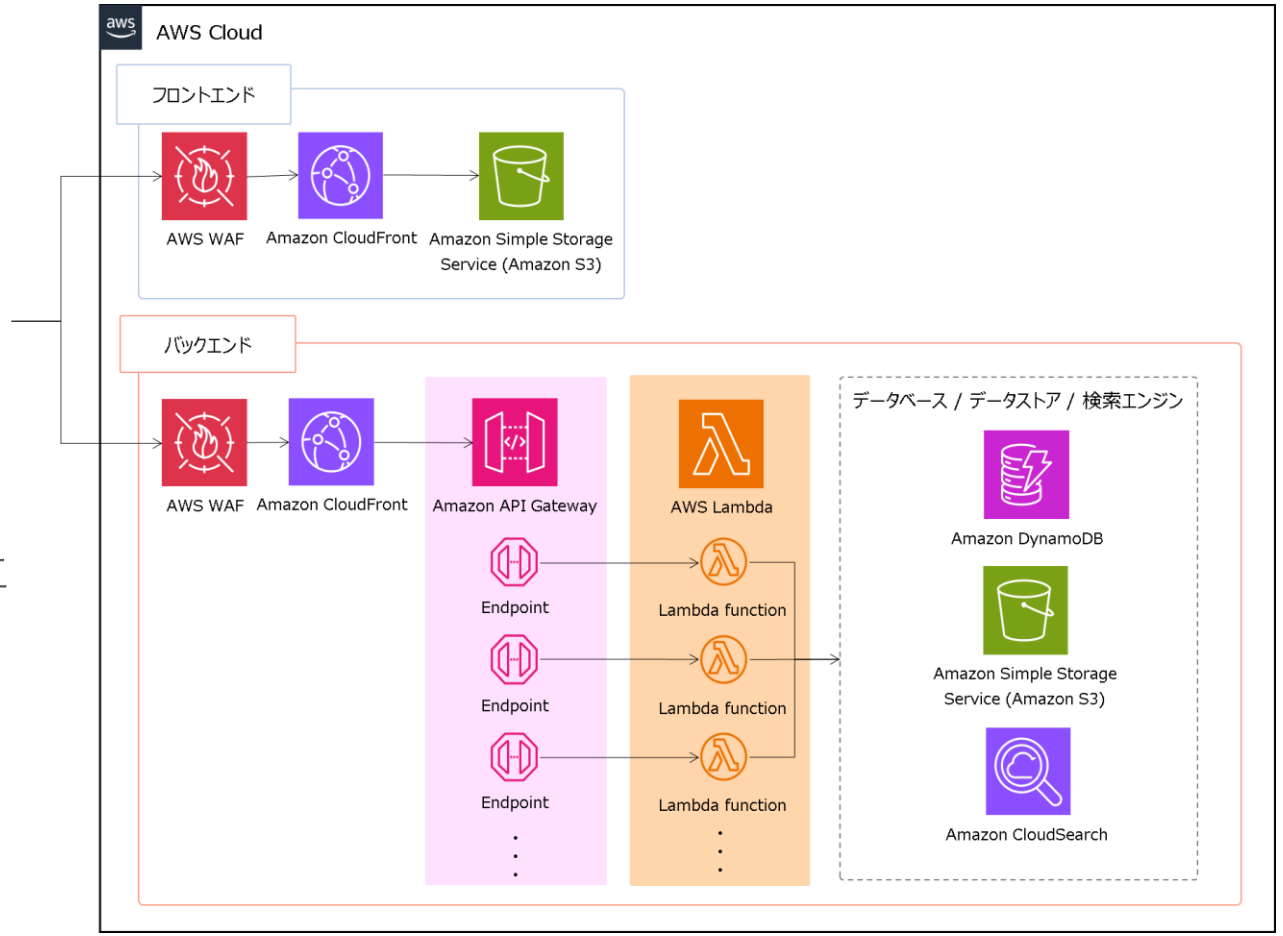


新しい店舗案内パッケージへ：基本構成

- フロントエンド
  - Angularフレームワークでアプリを SPA化
  - 静的ファイルを Amazon S3 へ格納し Amazon CloudFront でキャッシング

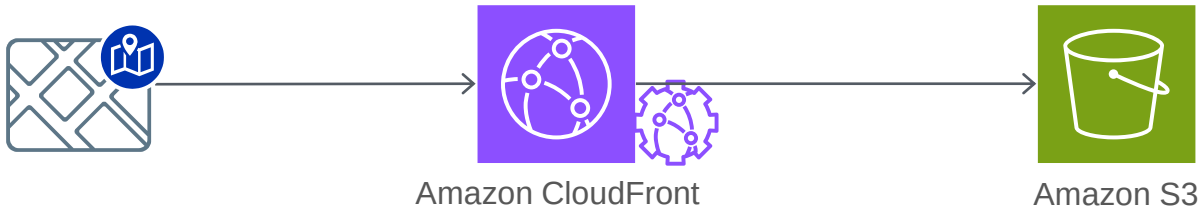
- バックエンド
  - 動的データ取得は Amazon API Gateway・AWS LambdaでWeb APIを構築。データストレージには Amazon DynamoDB を採用
  - またAmazon CloudSearchを利用し全文検索エンジンを構築

※基本部分の抽出となります。
一部Amazon RDS ・ Amazon EC2等も使用しています。



新しい店舗案内パッケージへ：AWS のサービス選定および改善点

- フロントエンドでの改善点

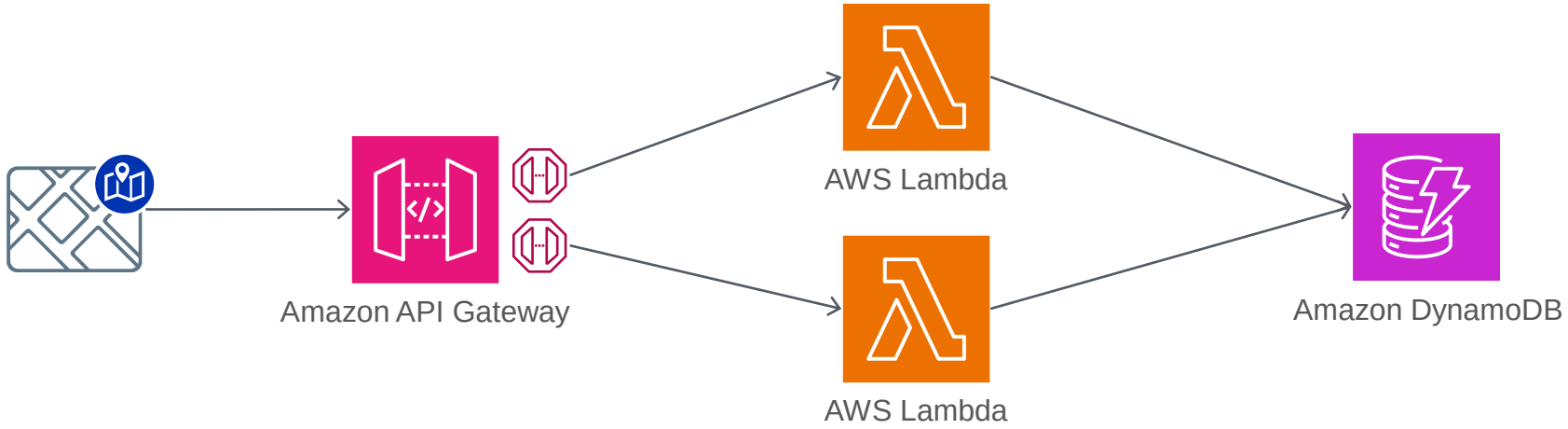


- 静的コンテンツのホスティングには高い耐障害性を持つAmazon S3とAmazon CloudFrontを採用。
  WEBサーバーのスペックや台数を意識せず、OS・ミドルウェアのバージョン管理からも解放されます。

- CloudFrontではCDNとしてのキャッシュ用途以外に、ビヘイビアを使用したpathパラメータによるオリジンの向き先変更や、CloudFront Functionsをアクセス制御等の用途で活用しマルチテナント上で各お客様の個別のご要望に対応しています。

新しい店舗案内パッケージへ：AWS のサービス選定および改善点

- バックエンドでの改善点

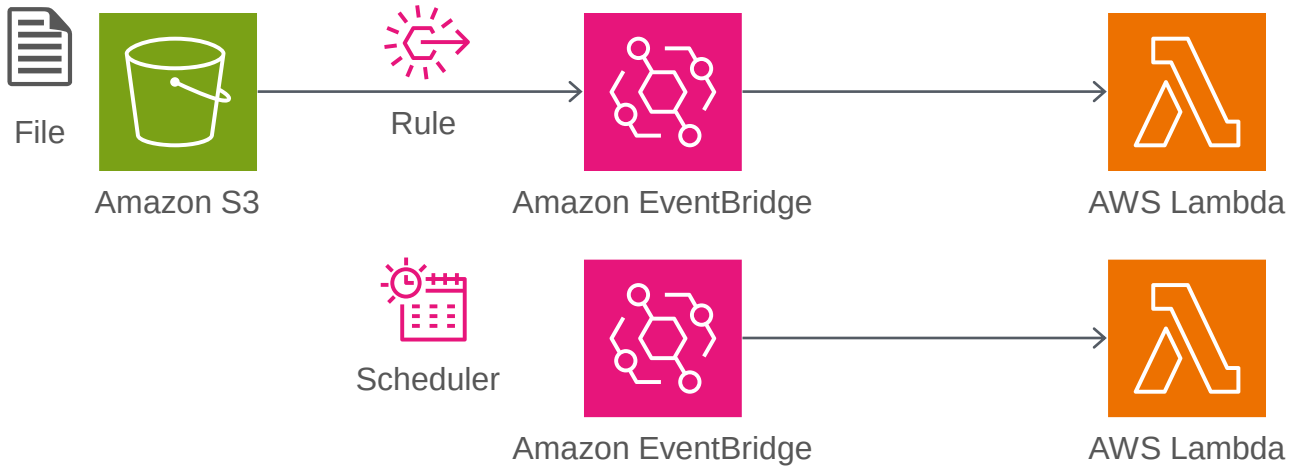


- API の作成にはOpenAPI 定義ファイルを利用して迅速に開発可能な Amazon API Gateway を利用。
  CloudFront同様スペックや台数を意識せず安心してサービス提供可能となりました。

- ロジック部にAWS Lambdaを採用しCRUD処理等エンドポイント毎に個別のLambda関数で実装されており、
  デプロイはLambda関数単位で可能となりました。
  これによりデプロイメント時の影響範囲を最小限に低減しています。

新しい店舗案内パッケージへ：AWS のサービス選定および改善点

- バッチ処理での改善点

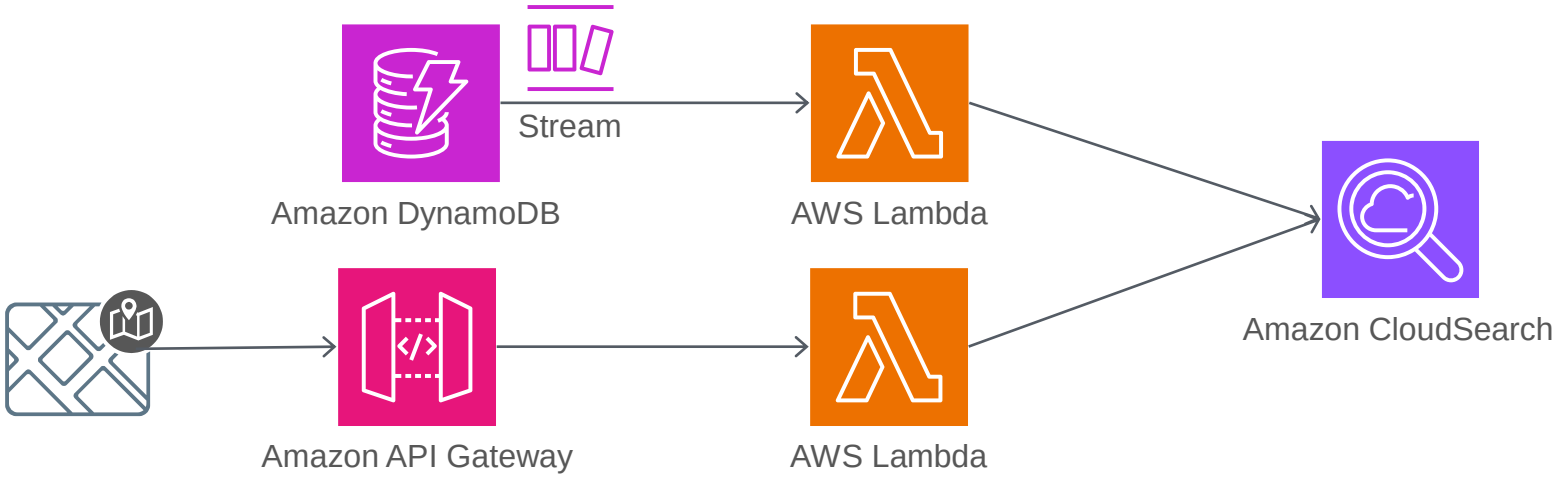


- AWSの様々なサービスをイベントによって連携する Amazon EventBridge を使用し、イベント駆動型のアーキテクチャを実現しています。

- Area Markerご利用のお客様によって投入される店舗情報CSVファイルがS3バケットにPUTされた事をトリガーに処理を開始することや、夜間定期実行される処理をスケジュール設定しLambda関数を必要に応じて起動させることにより、これまで常時稼働していた仮想サーバーの台数を減らすことが可能となりました。

新しい店舗案内パッケージへ：AWS のサービス選定および改善点

- Amazon DynamoDB および Amazon CloudSearch

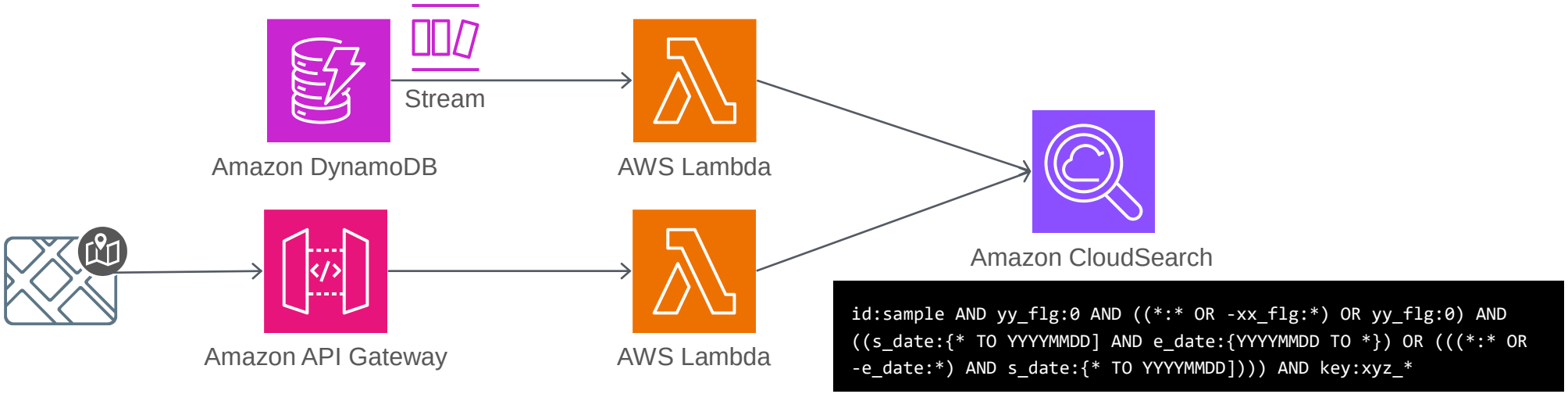


- データストレージには結果整合性モデルにより高いスケーラビリティとパフォーマンスを持つNoSQLデータベースサービスAmazon DynamoDB を採用しました。

- Area Markerはお客様毎に店舗に付随する様々な情報を効率的に格納する要件がありました。DynamoDB はドキュメントストア型のデータベースとしても利用可能なため、柔軟なデータモデリングが可能となりました。

新しい店舗案内パッケージへ：AWS のサービス選定および改善点

- Amazon DynamoDB および Amazon CloudSearch

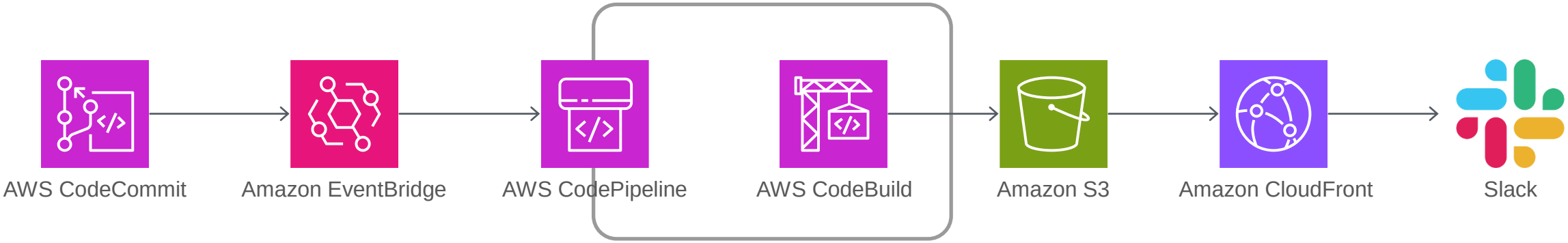


- DynamoDB のデータモデリングが自由な反面、柔軟なクエリや表結合は困難なため、セカンダリインデックス等を含めテーブル設計はややハードルが高くなってしまう状況でした。
  Area Markerでは、バックエンドの検索エンジンとして Amazon CloudSearchを利用しており、CloudSearchの強力なクエリ機能を使い多様な検索パターンを実現させています。

- DynamoDB から CloudSearch へのデータアップロードはDynamoDB StreamsよりLambda関数を経由して自動化されています。

新しい店舗案内パッケージへ：AWS のサービス選定および改善点

- フロントエンド開発のデプロイ作業においての取組



- ソースコードリポジトリにAWS CodeCommitを使用し、特定ブランチの変更(マージ)をEventBridgeを通して検知しAWS CodePipelineを実行しています。

- AWS CodeBuildのビルドプロジェクトでは、Angular-CLIによるtest & buildが実行された後、 AWS CLIを使用して公開用S3バケットにデプロイを行います。さらにCloudFrontビヘイビアのキャッシュクリアを実施、全てが正常終了（または途中異常終了）した際にSlackへ通知するパイプラインを構築しています。

AWSサーバーレスアーキテクチャを活用する事で享受されるメリット

- 多数の仮想サーバーを稼働することなく、機能毎のマイクロサービス化を実現

  - OS、ミドルウェア（Apache・NGINXなど）のEOL対応から解放
  - プラットフォームに対するパッチ適用はAWSに責任を委譲できる

  - 単一障害点の減少
  - 必要に応じて必要な箇所のみをスケールアウトでき、
    従量課金制によるコスト抑制（平時の稼働を抑える）、インフラ費用は約90%削減

  - 新メンバー追加時の導入スピードアップ
  - 機能毎のリリースが可能になり、テストの効率化や影響範囲を抑制できる

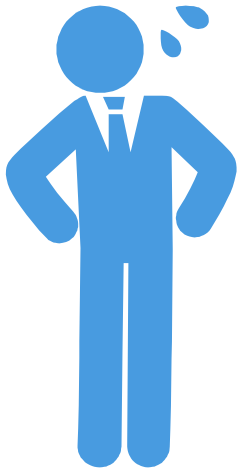
一方で新たな課題も…

- サーバーレス・マイクロサービス化によって発生する新たな課題

共通ロジックと個別機能の分離

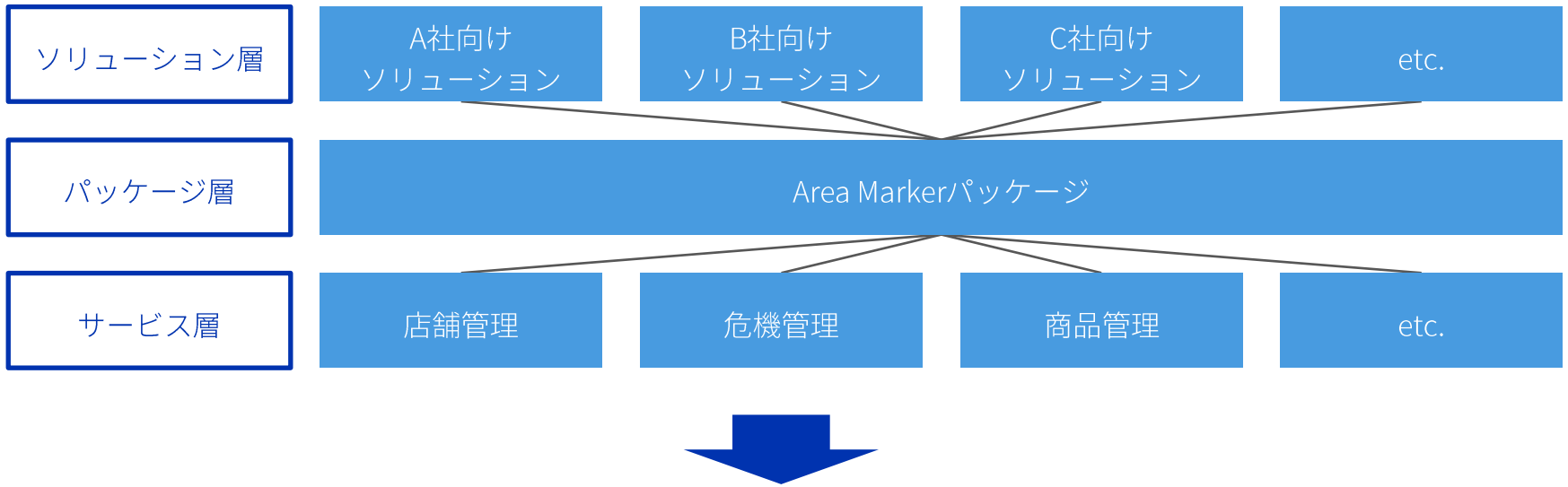
ステートの管理（大量データの処理時間）

大量に増えたLambda関数の管理

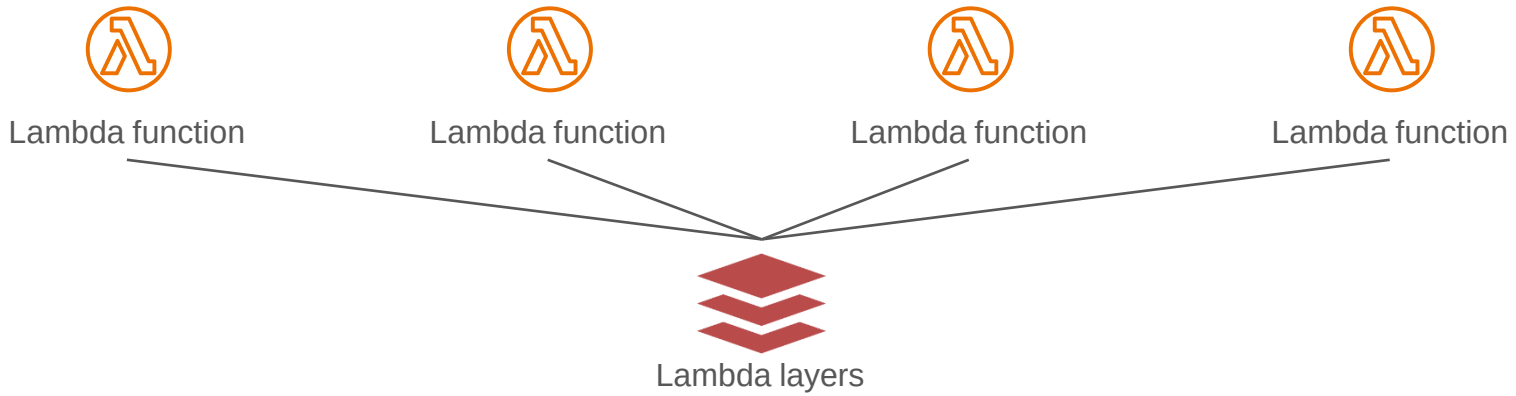


新たな課題：共通ロジックと個別機能の分離

- 座標演算等の共通機能を頻繁に処理し、マルチテナント上で各社の個別機能を扱うArea Marker

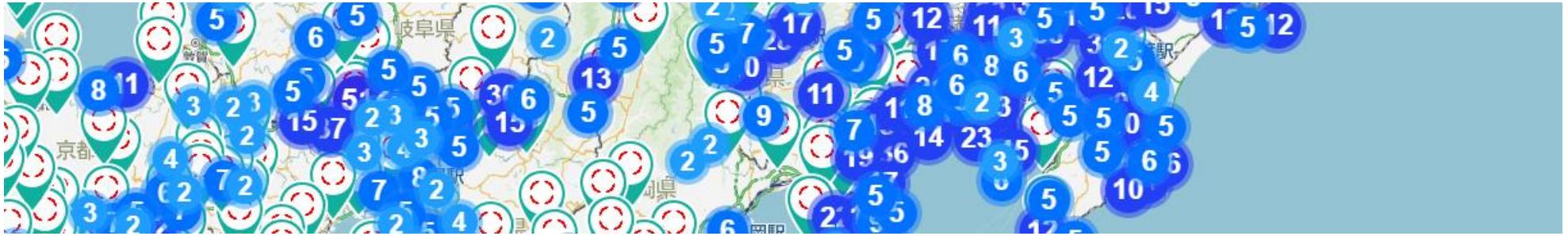


- 共通ライブラリやカスタムランタイムをアーカイブできるLambdaレイヤーを効率良く活用

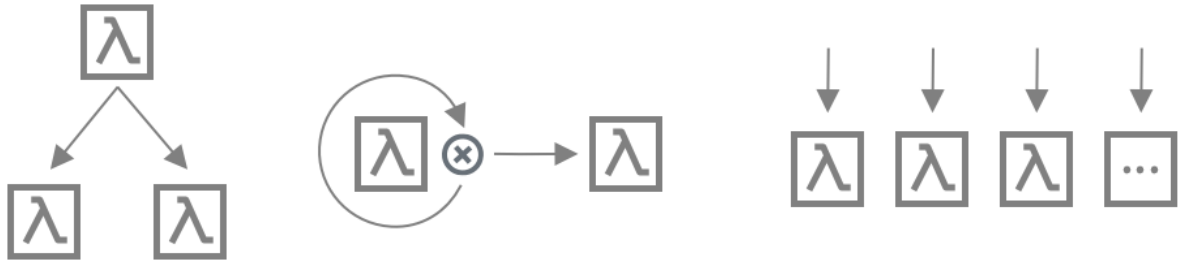


新たな課題：ステートの管理（大量データの処理時間）

- お客様の店舗情報が十万件以上に至る場合もあり、一連の処理が長時間に及ぶケースが発生
  - Lambda関数の最長実行時間（15分）やオーケストレーション



- LambdaなどAWSのサービス機能と統合してアプリケーションを構築できるAWS Step Functionsを活用
  - 複数のLambda関数やSQSを組み合わせ状態を管理

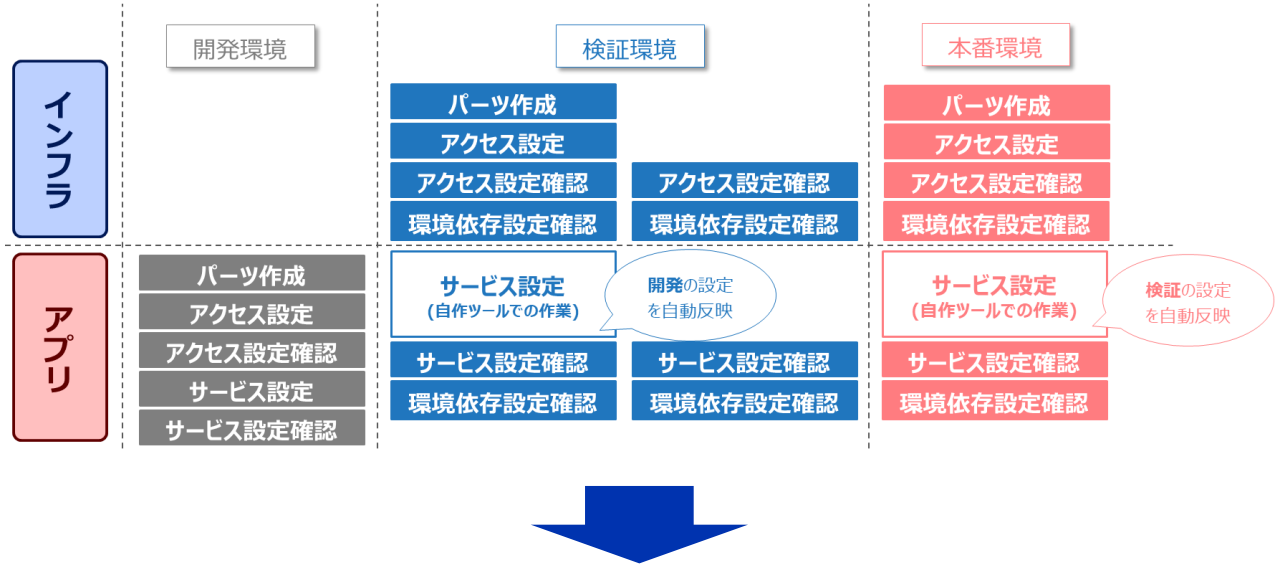


条件分岐　　エラー処理　　並列処理　　etc.

画像引用：https://docs.aws.amazon.com/step-functions/latest/dg/welcome.html

新たな課題：大量に増えたLambda関数の管理

- マイクロサービス化によりLambda関数が乱立（関係性や設定の管理）。作業も煩雑に…



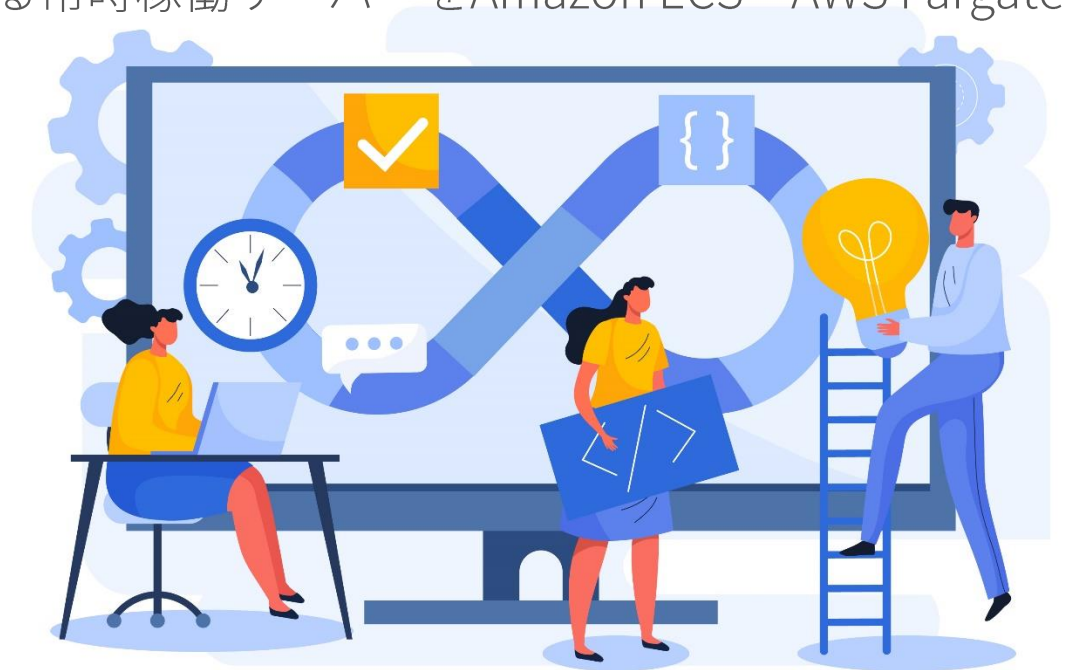
- AWS CDK（AWS Cloud Development Kit）を利用し機能単位でまとまったリソース群を定義・プロビジョニングまで。
  - プログラミング言語（チームではPythonを選択）を使用してInfrastructure as Code（IaC）を実現。
  - VSCode等IDEと連携して強力な補完機能を活用できる。

今後の展望

- CI/CDサイクルとDevOpsの文化を根付かせ質の高いシステムを迅速かつ継続的に届ける

  - E2Eテスト自動化をパイプラインへ導入
    - MagicPod・ Playwright等を検証中

  - 残るEC2のコンテナ化
    - 気象災害等を処理する常時稼働サーバーをAmazon ECS・AWS Fargate等によりコンテナ化



著作者：Freepik

サーバーレスアーキテクチャ選定の注意点

- AWSサービスクォータに気を付け、必要に応じて上限緩和申請を検討する。
  - AWS Lambda: 同時実行関数数/関数のデプロイパッケージのサイズ/関数の実行タイムアウト 等
  - Amazon API Gateway: デフォルト/バーストのレート制限/ステージあたりのデプロイメント数 等
  - Amazon DynamoDB: テーブルやインデックスの最大サイズ/単一パーティションでサポートされる最大読み取り/書き込み容量単位 等
  - Amazon S3: シングルPUTでアップロードできるオブジェクトの最大サイズ 等
  - AWS Step Functions: ステートマシンの最大ステート数/ステートマシン実行の最大履歴 等

- アイドル状態が続いたLambda関数は非アクティブになります。
  - Amazon RDS 、 Amazon EC2インスタンスなどのサービスにアクセスできるようVPCに接続するようにLambda 関数を設定した場合、非アクティブにならない（定期的に実行するなど）工夫が必要です。

- 全てのバージョン管理から解放されるわけではない
  - Lambda関数のランタイムやAmazon RDSのデータベースエンジンなどはサポート期限があります。
    日頃の継続的インテグレーションにより対応インパクトを縮小する取り組みは必要です。

まとめ

長い歴史の中で進化を経て、ゼンリンデータコムの「店舗案内パッケージ」サービスは
AWSのサーバーレスアーキテクチャとマイクロサービスの採用により、より柔軟でス
ケーラブルなSaaSとして成長しました。
サーバーレス技術への移行は部分的に開始することも可能です。皆様のプロジェクトにお
いても、弊社実例が検討の一助やインスピレーションとなることを心より願っています。
ありがとうございました。