aws DEV DAY
ONLINE JAPAN

20-22.10.2020

DEV DAY

In Partnership with intel

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws DEV DAY ONLINE JAPAN

20.10.2020

A - 2

DX時代における最適な開発手法 サーバーレスとDB選択の勘所

SOMPOホールディングス株式会社

チーフエンジニア　杉山祐介

In Partnership with intel

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

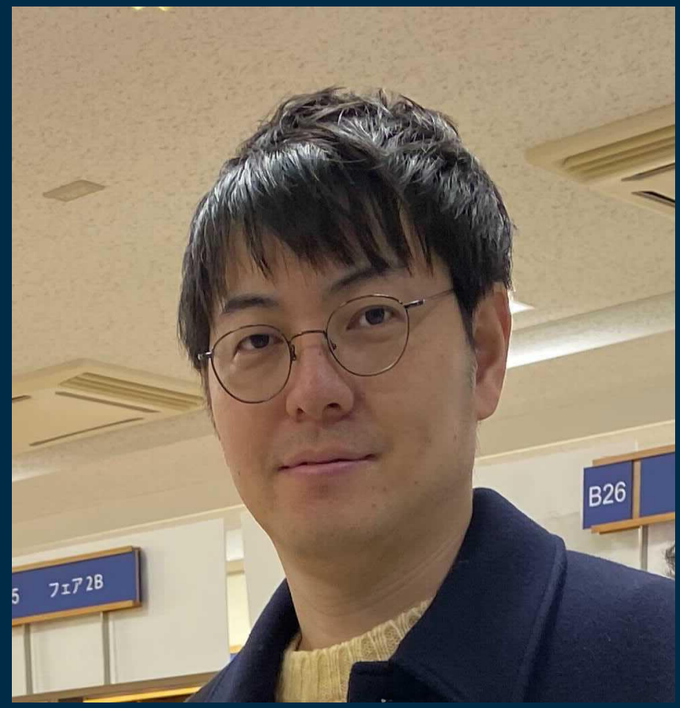
自己紹介

杉山 祐介 (Yusuke Sugiyama)

SOMPOホールディングス株式会社
デジタル戦略部 / チーフエンジニア

様々なPoCを支えるデジタルSandboxを担当

好きなAWSのサービス
Amazon Relational Database Service



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with intel

About SOMPO Sprint Team

- Our Mission
  - デジタルを活用した新たな顧客体験価値の創出
- About Team
  - 2018年7月：デジタル戦略部内に発足
  - 約40名のデザイナー / エンジニア集団
  - 事業部門、企画、デザイナー、エンジニアが一体となったアジャイル開発を実践
  - 完全内製開発部隊
- Our Product
  - 発足から約**50**のPoC向けプロダクトを開発

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with intel

Agenda

- SOMPO Sprint Teamが選んだ標準サーバーレス構成
  - フルマネージドサービスを選択した理由
- DBの異なる事例のご紹介
  - Aurora Serverless 導入事例
  - DynamoDB 導入事例
- RDSとDynamoDBを選択する際の基準
  - SOMPO Sprint Teamとしての選択基準
- Conclusion
  - ふりかえりと今後に向けて

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

本日お伝えしたいこと

1. サーバーレスのすばらしさ

2. RDS or DynamoDB ?

3. SOMPO Sprint Teams Challenge

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

お話の前に

1. 超巨大システムを開発するお話は出てきません。

2. PoC開発ではありますが、本番化とスケーリングも考慮したお話です。

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws DEV DAY ONLINE JAPAN

SOMPO Sprint Teamが選んだ標準サーバーレス構成

In Partnership with intel

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

フルマネージドサービスを選択した理由

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with

Sprint Team としての開発にあたって

- 価値創出のため、ライトにクイックな開発を行いたい
- 本番サービス稼働時に備えてスケーリングも考慮したい
- デザイナーはUI/UXデザイン、エンジニアはコードに集中したい
- 運用コスト、EOS対策など、考えることは多い…

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

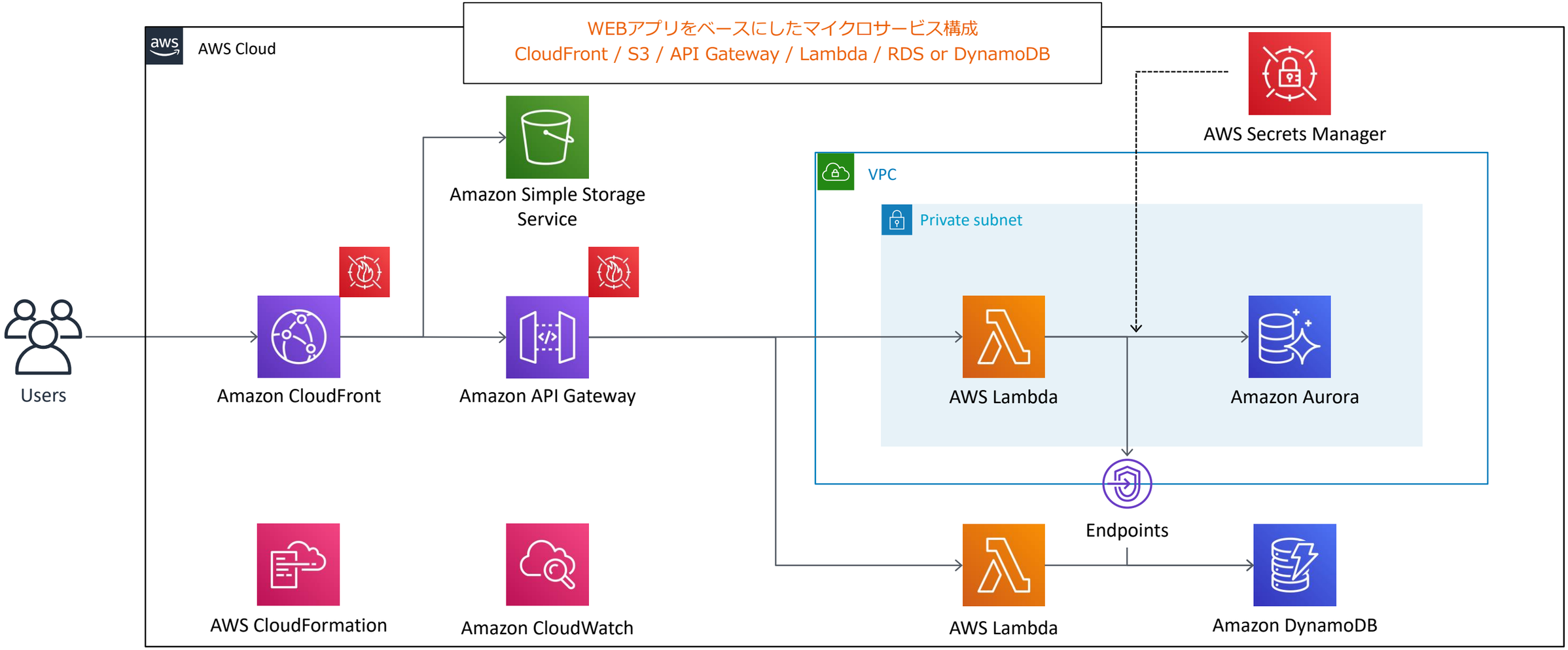
Sprint Team としての開発にあたって

- 価値創出のため、ライトにクイックな開発を行いたい
- 本番サービス稼働時に備えてスケーリングも考慮したい
- デザイナーはUI/UXデザイン、エンジニアはコードに集中したい
- 運用コスト、EOS対策など、考えることは多い…

フルマネージドサービスを選択して
開発効率を最大化

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Sprint Teamが選んだ構成

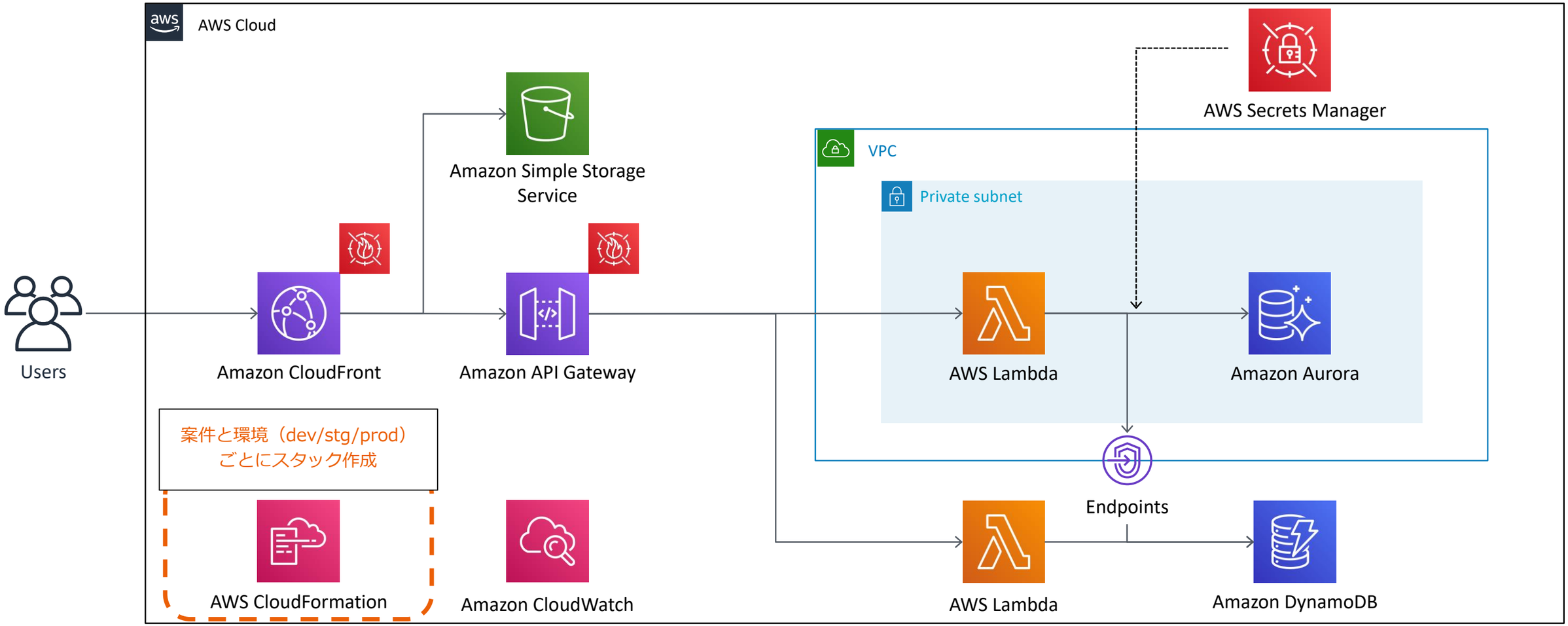


代表的なサービスのみ記載

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with intel

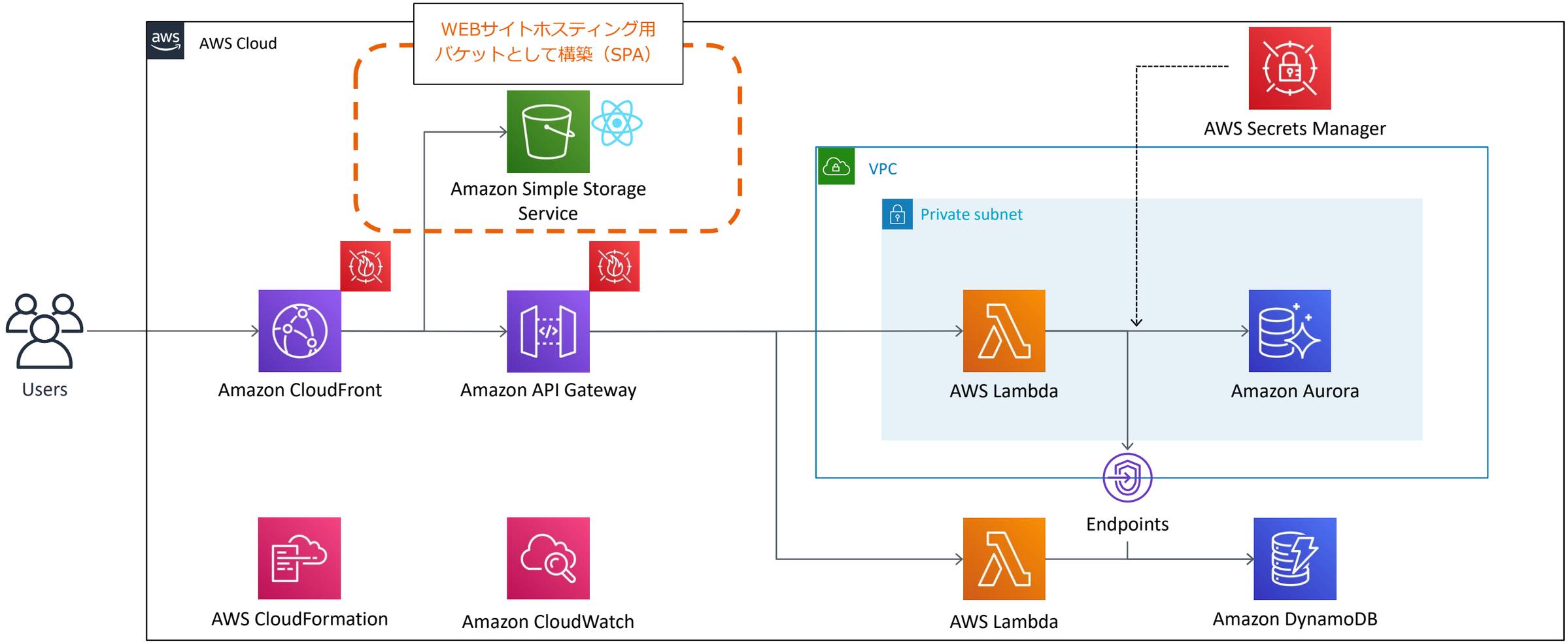
Sprint Teamが選んだ構成



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with intel

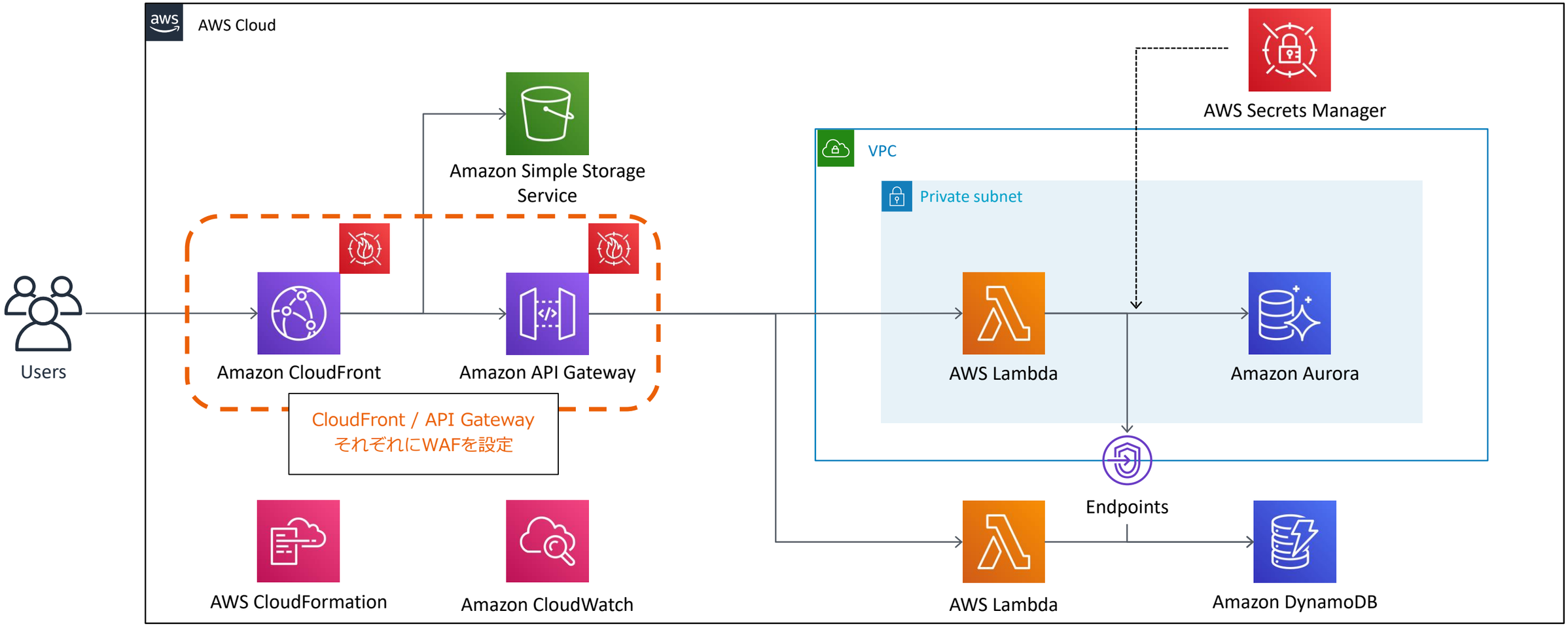
Sprint Teamが選んだ構成



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with intel

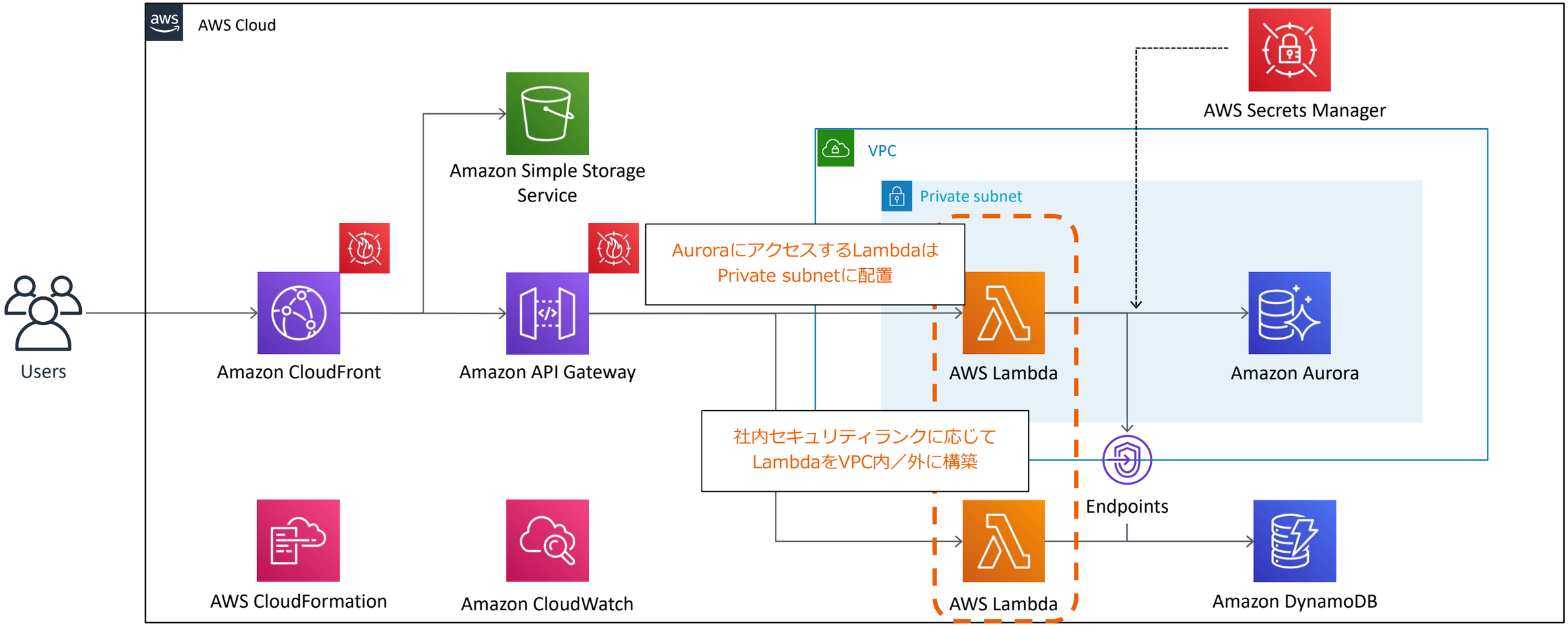
Sprint Teamが選んだ構成



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with intel

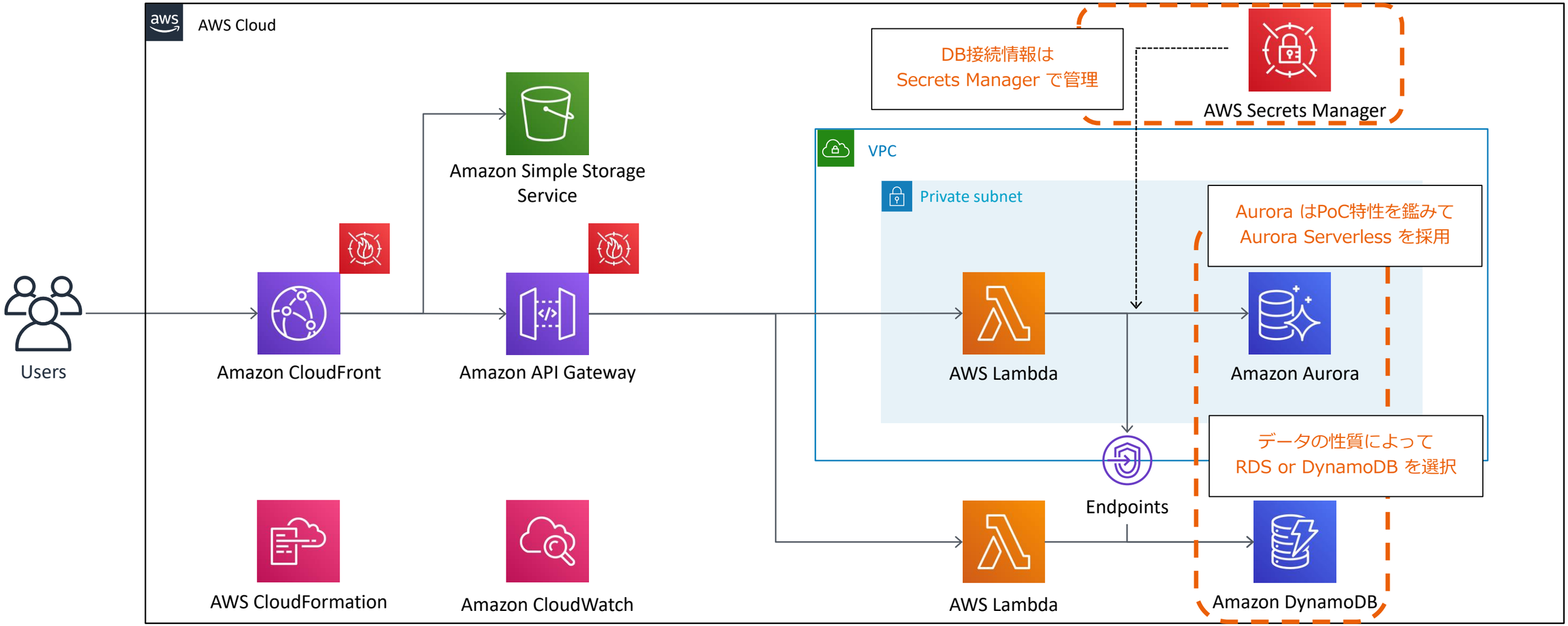
Sprint Teamが選んだ構成



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with intel

Sprint Teamが選んだ構成



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with intel

標準構成を準備して得られた恩恵

事例：車両変更受付フォームの構築

コロナ禍でコールセンターが休業となり、保険契約者が新しい自動車に
契約内容を変更するための、申請受付を行うサイトを開発する必要が発生。

通常であれば二ヶ月以上は掛かる見込み…

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with

標準構成を準備して得られた恩恵

事例：車両変更受付フォームの構築

コロナ禍でコールセンターが休業となり、保険契約者が新しい自動車に
契約内容を変更するための、申請受付を行うサイトを開発する必要が発生。

通常であれば二ヶ月以上は掛かる見込み…

標準構成を利用して開発期間を短縮

三週間でのリリースを実現

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with intel

aws DEV DAY
ONLINE JAPAN

DBの異なる事例のご紹介

In Partnership with intel

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Aurora Serverless 導入事例

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with

事例：自動飛行ドローンの情報連携DB（PoC）

【目的】

- ドローンと専用自動飛行アプリで広域災害時に、迅速かつ効率的に地域や契約者宅の被災情報を収集・把握する。

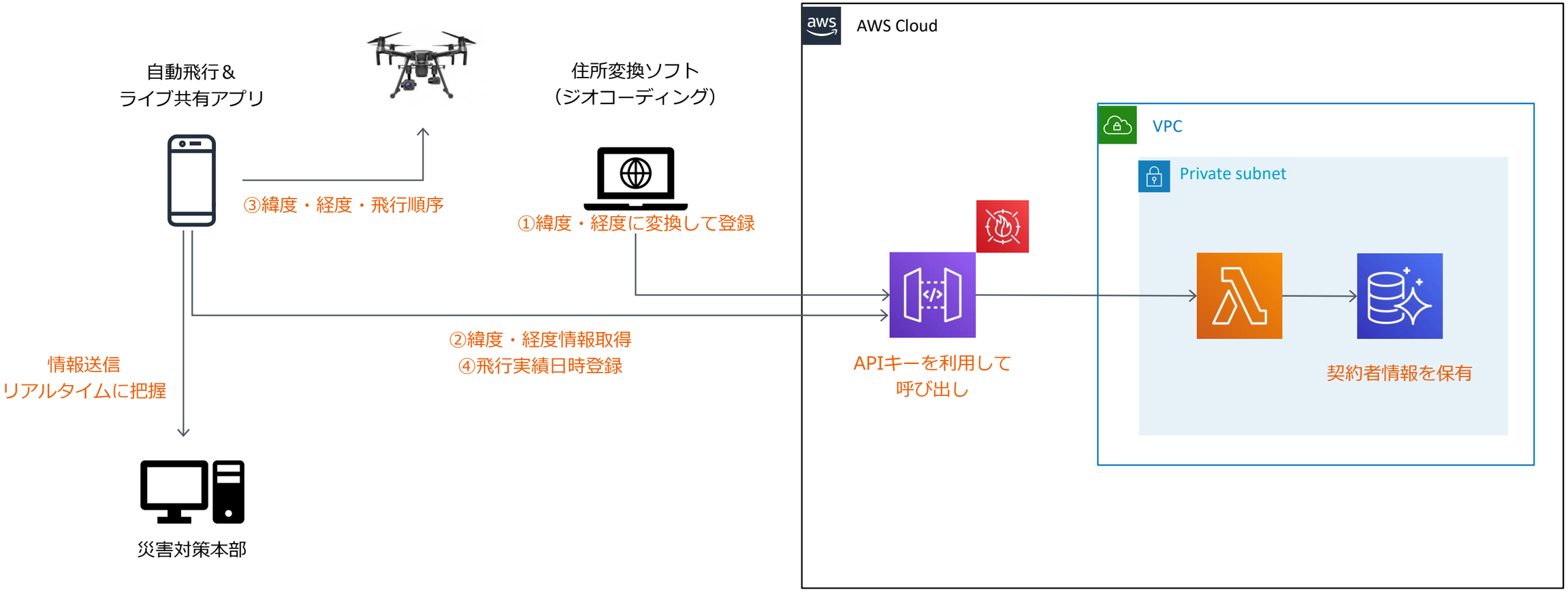
【実施内容】

- 契約者住所情報をジオコーディングし自動飛行アプリに取り込み、取り込んだ各地点を任意選択することで、対地高度も含め自動で飛行ルートが生成され、適切に飛行できるか検証する。

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with intel

構成図



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with intel

Aurora Serverless 導入理由

- 複数項目での検索要件があり、RDBを利用したい
- 有事を中心に利用するものであって、常時利用するものではない
- 常時利用ではないため、コストも最小限に抑えたい

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

DynamoDB 導入事例

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with

事例：位置情報を利用した、とあるサービス

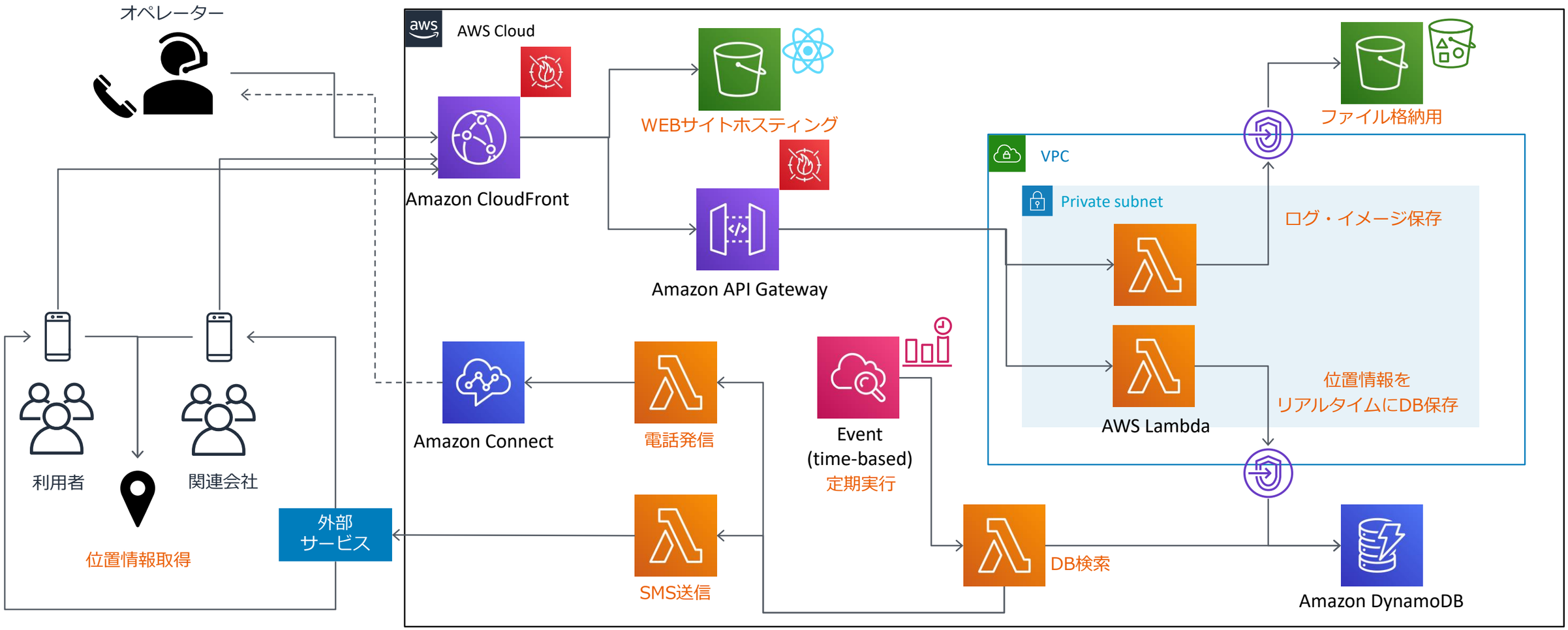
詳細は弊社グループからの正式発表を
お待ち下さい

本日は構成のみご紹介

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with intel

構成図



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with intel

DynamoDB 導入理由

- 位置情報やチャットなどをリアルタイム保存したい
- WEBアプリで扱っているJSON形式でデータ保存が可能
- キー以外の属性に未確定要素が多く、柔軟なデータ構造に対応したい
- 検索要件はシンプル、キーのみでの検索に限定できる
- 上記に加え、クイックに開発を行いたい

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws DEV DAY
ONLINE JAPAN

RDSとDynamoDBを選択する際の基準

In Partnership with intel

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

基準を考える前におさらい

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with

基準を考える前におさらい

- RDBとDynamoDBの特徴をふりかえる
- Sprint Teamが採用しているAurora Serverlessの特徴をふりかえる

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

RDBとDynamoDBの特徴をふりかえる

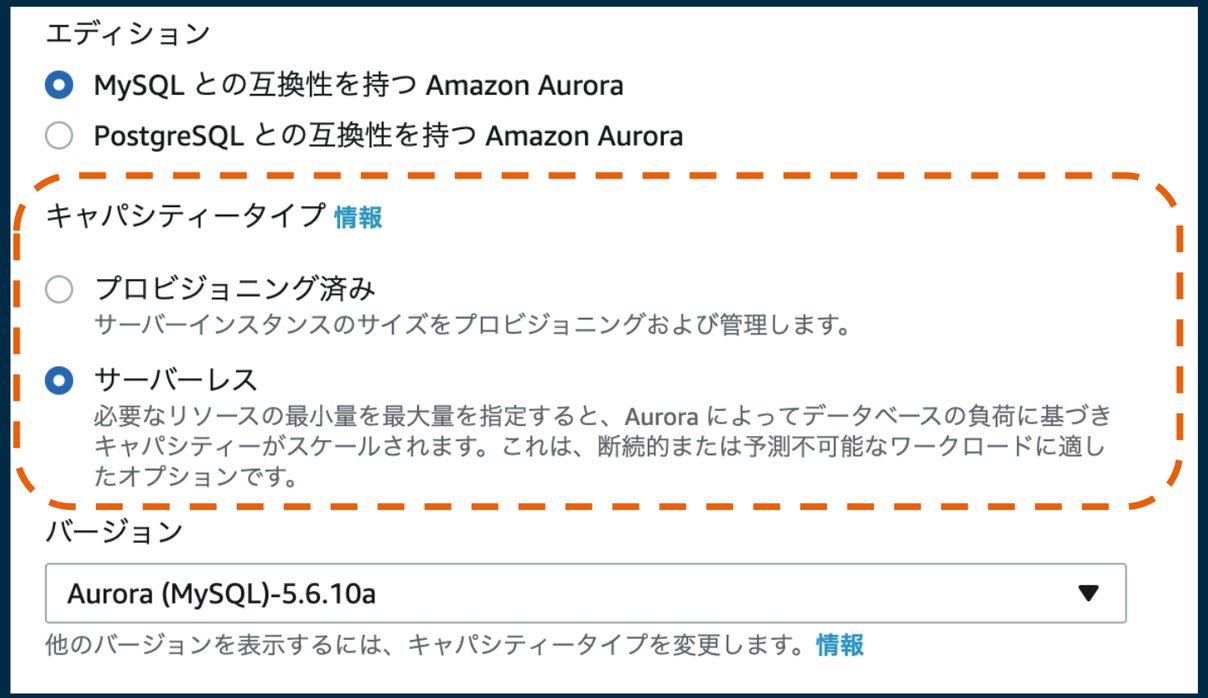
一般的なRDBの特徴とDynamoDB（Key-Value）の特徴について

	RDB	DynamoDB
データ構造	正規化 / 関連性	非正規化 / 階層構造
問い合わせ	SQLを使用 / 複数条件指定	キーを指定したクエリ実行
一貫性・整合性	トランザクション処理による 厳格な一貫性	結果整合性 ※強力な整合性のある読み込みも可
スケーラビリティ	煩雑なスケーリング	高いスケーラビリティ

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Aurora Serverless をふりかえる

- Aurora用のオンデマンドのAuto Scaling設定
  - サービス名というわけではない
- 自動的に起動、シャットダウン、スケールアップ／ダウンが可能
- 不定期、断続的、予測不能なワークロードに最適
- 一方、制約事項も多い



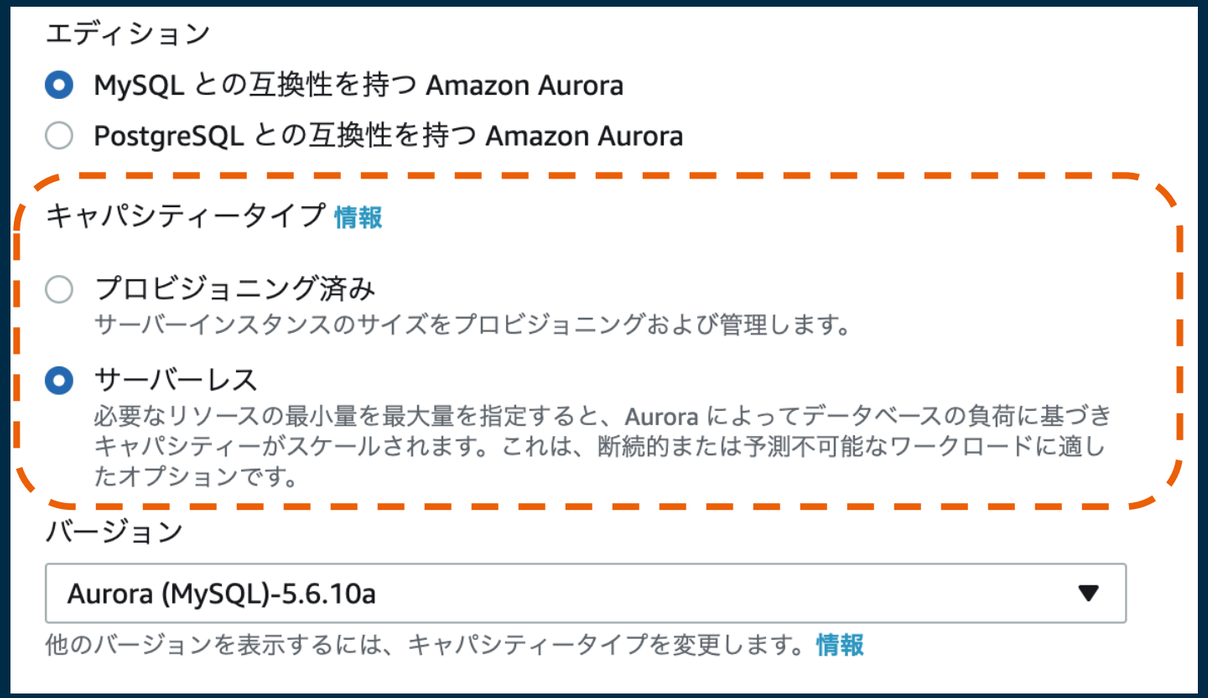
https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/aurora-serverless.html

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with intel

Aurora Serverless をふりかえる

- Aurora用のオンデマンドのAuto Scaling設定
  - サービス名というわけではない
- 自動的に起動、シャットダウン、スケールアップ／ダウンが可能
- 不定期、断続的、予測不能なワークロードに最適
- 一方、制約事項も多い



PoCで予測が難しいキャパシティ問題解決

コストメリットも

https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/aurora-serverless.html

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with intel

データの特性を考える

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with

データの特性を考える

- PoCで取り扱うデータの特性は最初に検討すべき事項
  - 個人情報が含まれるか等、PoC開発着手前に明確にしておかなければならない
  - データの内容だけではなく、誰がどのような経路でアクセスするのかも考える

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

データの特性を考える

- PoCで取り扱うデータの特性は最初に検討すべき事項
  - 個人情報が含まれるか等、PoC開発着手前に明確にしておかなければならない
  - データの内容だけではなく、誰がどのような経路でアクセスするのかも考える

厳格なセキュリティ体制を敷けば

どんなパターンでも問題なし・・・？

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

データの特性を考える

- PoCで取り扱うデータの特性は最初に検討すべき事項
  - 個人情報が含まれるか等、PoC開発着手前に明確にしておかなければならない
  - データの内容だけではなく、誰がどのような経路でアクセスするのかも考える

厳格なセキュリティ体制構築と
柔軟なアジャイル開発のトレードオフ

ジレンマに陥る

厳格なセキュリティ体制を敷けば

どんなパターンでも問題なし・・・？

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with intel

データの特性を考える

- PoCで取り扱うデータの特性は最初に検討すべき事項
  - 個人情報が含まれるか等、PoC開発着手前に明確にしておかなければならない
  - データの内容だけではなく、誰がどのような経路でアクセスするのかも考える

純粋にDB選択に集中したい

他のアプローチはないだろうか？

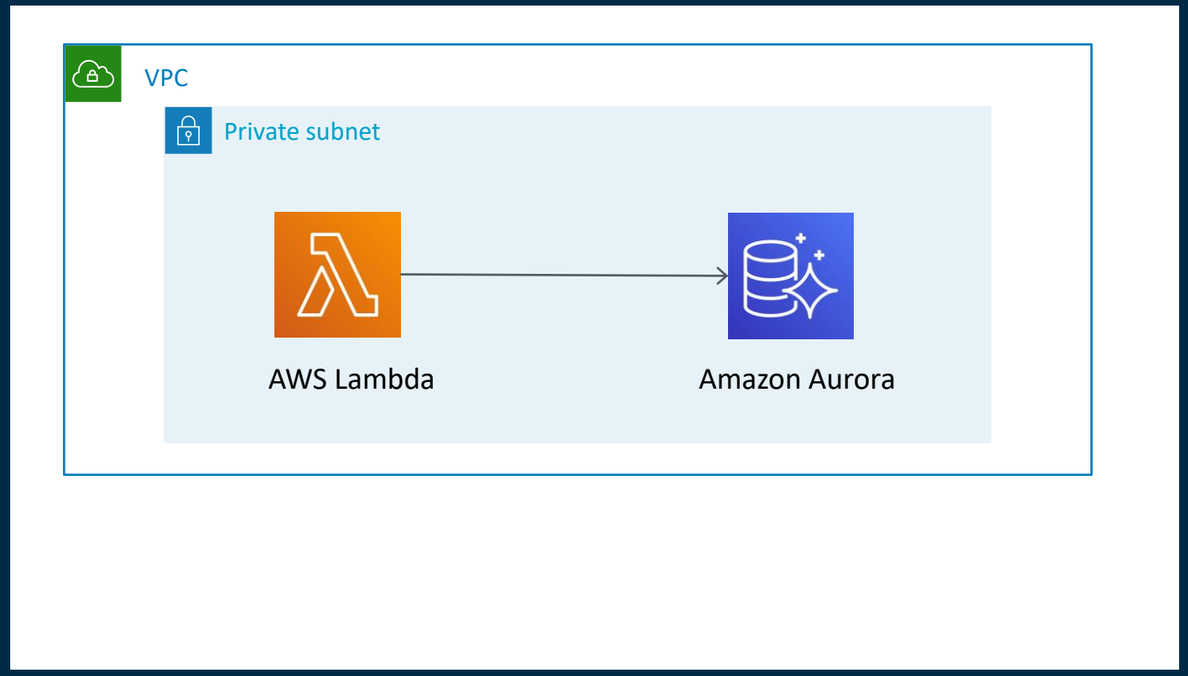
厳格なセキュリティ体制を敷けば

どんなパターンでも問題なし・・・？

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

とりあえず Private subnet 方式

- とりあえずRDSならPrivate subnetに置けるので安全・・・？



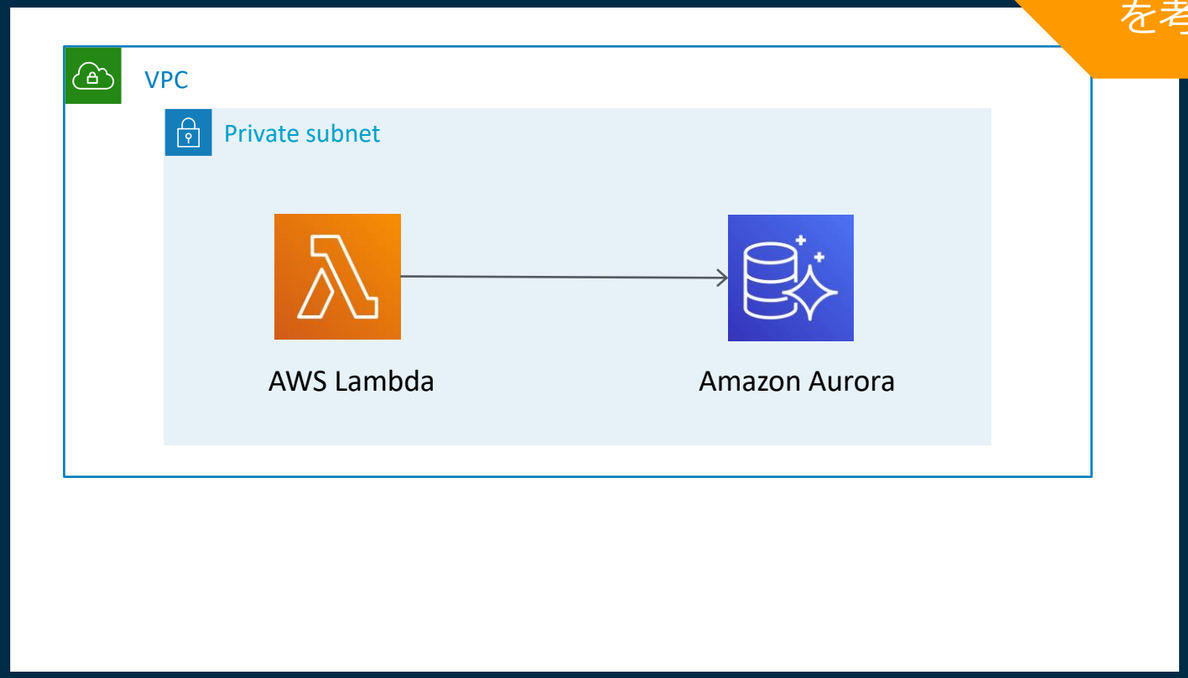
© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with

とりあえず Private subnet 方式

- とりあえずRDSならPrivate subnetに置けるので安全・・・？

もう少し踏み込んで
セキュリティゾーンモデリング
を考える

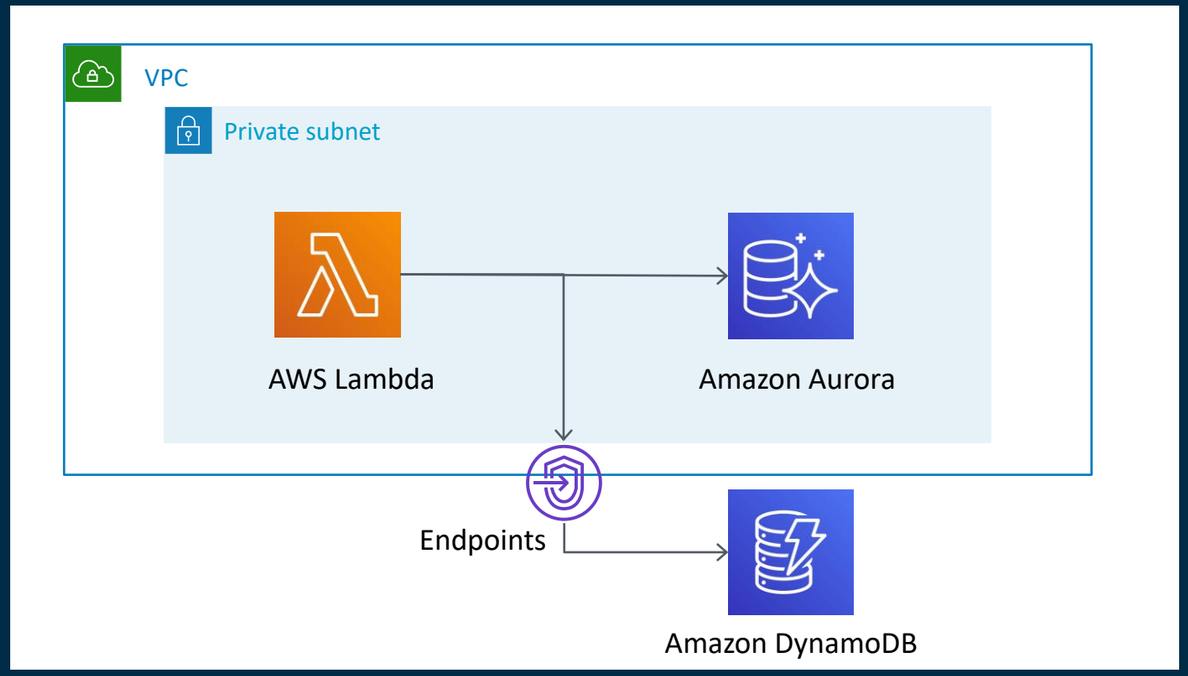


© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with intel

セキュリティゾーンモデリングの設計

- 機密データを扱うDBはセキュアゾーンに配置する
  - ネットワークの隔離を行うことでトラフィックの外部公開を防ぐ
  - RDS：Private subnet に配置してアクセスコントロール
  - DynamoDB：VPCエンドポイントを介することでアクセス元をVPC内部リソースに制限

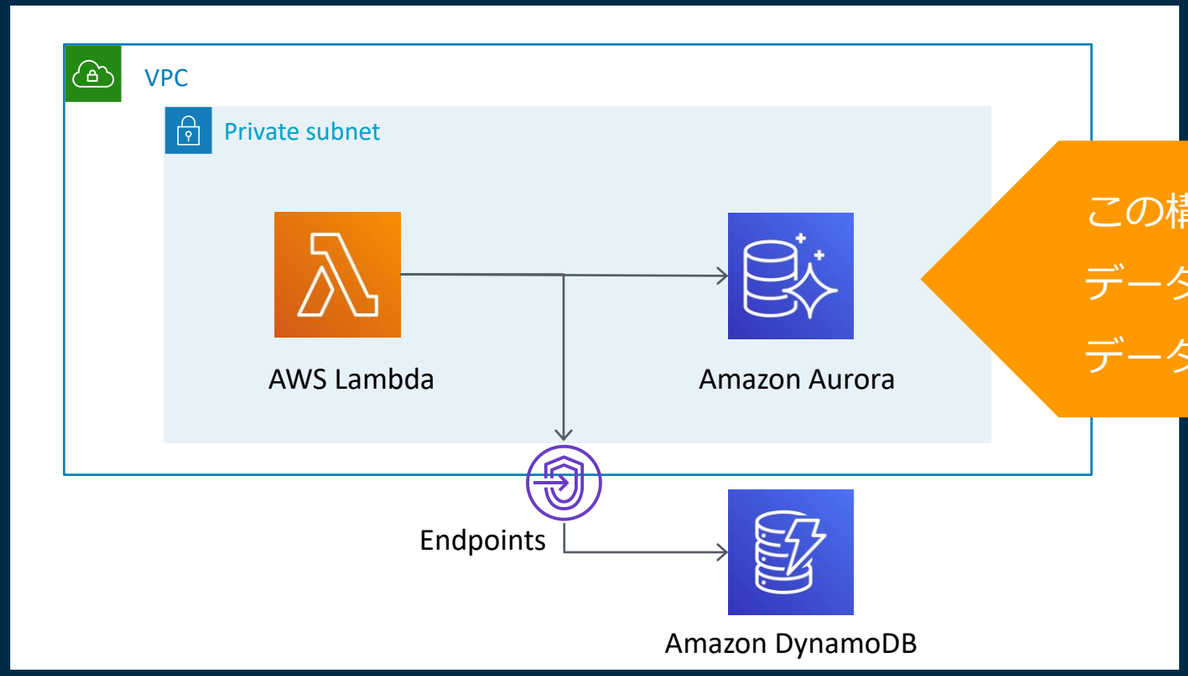


© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with

セキュリティゾーンモデリングの設計

- 機密データを扱うDBはセキュアゾーンに配置する
  - ネットワークの隔離を行うことでトラフィックの外部公開を防ぐ
  - RDS：Private subnet に配置してアクセスコントロール
  - DynamoDB：VPCエンドポイントを介することでアクセス元をVPC内部リソースに制限



この構成を前提とすることで、
データの特性や要件に集中して
データベース選択が可能となる

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with intel

RDS or DynamoDB ?

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

事例：DynamoDBを選択した際の基準とその後

【基準】

- 柔軟なデータ構造に対応したい、クイックに始めたいのでDynamoDBを選択

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

事例：DynamoDBを選択した際の基準とその後

【基準】

- 柔軟なデータ構造に対応したい、クイックに始めたいのでDynamoDBを選択

【その後】

- データ構造変更発生→柔軟に対応可能（選択してよかった）

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

事例：DynamoDBを選択した際の基準とその後

【基準】

- 柔軟なデータ構造に対応したい、クイックに始めたいのでDynamoDBを選択

【その後】

- データ構造変更発生→柔軟に対応可能（選択してよかった）
- 要件が変わり、だんだん検索に耐えられなくなってきた…
  - 複数条件での検索が厳しい

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

事例：DynamoDBを選択した際の基準とその後

【基準】

- 柔軟なデータ構造に対応したい、クイックに始めたいのでDynamoDBを選択

【その後】

- データ構造変更発生→柔軟に対応可能（選択してよかった）
- 要件が変わり、だんだん検索に耐えられなくなってきた…
  - 複数条件での検索が厳しい
- ついにScan不可避に、RDSにしておけば良かったかも

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

事例：DynamoDBを選択した際の基準とその後

【基準】

- 柔軟なデータ構造に対応したい、クイックに始めたいのでDynamoDBを選択

【その後】

- データ構造変更発生→柔軟に対応可能（選択してよかった）
- 要件が変わり、だんだん検索に耐えられなくなってきた…
  - 複数条件での検索が厳しい
- ついにScan不可避に、RDSにしておけば良かったかも
- 検索キー用の属性を追加して何とか対処
  - GSIを再設計するなど、四苦八苦

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

事例：DynamoDBを選択した際の基準とその後

【基準】

- 柔軟なデータ構造に対応したい、クイックに始めたいのでDynamoDBを選択

【その後】

- データ構造変更発生→柔軟に対応可能（選択してよかった）
- 要件が変わり、だんだん検索に耐えられなくなってきた…
  - 複数条件での検索が厳しい
- ついにScan不可避に、RDSにしておけば良かったかも
- 検索キー用の属性を追加して何とか対処
  - GSIを再設計するなど、四苦八苦

皆さんも同じように苦労した経験はないでしょうか・・・？

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

教訓から学んだ Sprint Team の基準

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with

SOMPO Sprint Teamとしての選択基準

【前提】

- 扱うデータ特性として、正規化が必要／リレーションが複雑なことが多い
- 既存DBとの繋ぎ込みも一定数発生する
  - 完全に独立したゼロベースでの開発がないわけではない

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

SOMPO Sprint Teamとしての選択基準

【前提】

- 扱うデータ特性として、正規化が必要／リレーションが複雑なことが多い
- 既存DBとの繋ぎ込みも一定数発生する
  - 完全に独立したゼロベースでの開発がないわけではない

前提をもとに
RDBとDynamoDBの
特徴を改めて確認

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

再掲：RDBとDynamoDBの特徴

一般的なRDBの特徴とDynamoDB（Key-Value）の特徴について

	RDB	DynamoDB
データ構造	**正規化 / 関連性**	非正規化 / 階層構造
問い合わせ	**SQLを使用 / 複数条件指定**	キーを指定したクエリ実行
一貫性・整合性	トランザクション処理による 厳格な一貫性	結果整合性 ※強力な整合性のある読み込みも可
スケーラビリティ	煩雑なスケーリング	高いスケーラビリティ

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

SOMPO Sprint Teamとしての選択基準

【前提】

- 扱うデータ特性として、正規化が必要／リレーションが複雑なことが多い
- 既存DBとの繋ぎ込みも一定数発生する
  - 完全に独立したゼロベースでの開発がないわけではない

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

SOMPO Sprint Teamとしての選択基準

【前提】

- 扱うデータ特性として、正規化が必要／リレーションが複雑なことが多い
- 既存DBとの繋ぎ込みも一定数発生する
  - 完全に独立したゼロベースでの開発がないわけではない

【基準】

- クイックに始めることに目を向けすぎず、RDBでまず考える
  - 検索時にリレーションが必要となった際に、あとで辛くなる→結果としてユーザートライアル開始の遅延は避けたい
  - 小さく始めることについてはAurora Serverless を利用しているため一定の恩恵を受けている
  - 本番化したときのことも考える：PoC終了後、保守する部隊の技術Stackも考慮

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

SOMPO Sprint Teamとしての選択基準

【前提】

- 扱うデータ特性として、正規化が必要／リレーションが複雑なことが多い
- 既存DBとの繋ぎ込みも一定数発生する
  - 完全に独立したゼロベースでの開発がないわけではない

【基準】

- クイックに始めることに目を向けすぎず、RDBでまず考える
  - 検索時にリレーションが必要となった際に、あとで辛くなる→結果としてユーザートライアル開始の遅延は避けたい
  - 小さく始めることについてはAurora Serverless を利用しているため一定の恩恵を受けている
  - 本番化したときのことも考える：PoC終了後、保守する部隊の技術Stackも考慮
- 単一アイテムの格納だけが明確な要件かつ全体構成がシンプルな場合、DynamoDBでもよい
  - この場合はエンジニア同士のレビューを設けてディスカッション
  - 全員の合意を得てDynamoDBを利用する→あとで辛くなっても責任を属人化させない

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

SOMPO Sprint Teamとしての選択基準

【前提】

- 扱うデータ特性として、正規化が必要／リレーションが複雑なことが多い
- 既存DBとの繋ぎ込みも一定数発生する
  - 完全に独立したゼロベースでの開発がないわけではない

【基準】

- クイックに始めることに目を向けすぎず、RDBでまず考える
  - 検索時にリレーションが必要となった際に、あとで辛くなる→結果としてユーザートライアル開始の遅延は避けたい
  - 小さく始めることについてはAurora Serverless を利用しているため一定の恩恵を受けている
  - 本番化したときのことも考える：PoC終了後、保守する部隊の技術Stackも考慮
- 単一アイテムの格納だけが明確な要件かつ全体構成がシンプルな場合、DynamoDBでもよい
  - この場合はエンジニア同士のレビューを設けてディスカッション
  - 全員の合意を得てDynamoDBを利用する→あとで辛くなっても責任を属人化させない

ハイブリッド構成も
もちろんありです！

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

DB選択基準策定までの流れ

RDBとNoSQLの特徴を
再確認する

データの特性を考える

チーム特性と取り扱うデータの
特性、どちらも意識する

基準は基準
柔軟な対応を忘れない

- NoSQLが初めての場合、まずKey-ValueのDynamoDBから検討を始めると、ドキュメントやナレッジも多く進めやすい
- セキュリティゾーンモデリングなどを実施
- 自身のチームや携わる事業など、改めて見直しを行う
- Sprint Team ではPoC開発かつデータ特性を鑑みて、まずRDBとした
- あくまでもクイックに始めるための属人化しない目安、常に見直しを！

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws DEV DAY
ONLINE JAPAN

Conclusion

In Partnership with intel

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

これまでのふりかえり

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with

サーバーレス構成のふりかえり

- よかった点
  - 超短期間のリリース実績をもたらしてくれた
  - サーバー管理から解放され、エンジニアがコードに集中できた
  - 標準構成を採用してから三倍速で案件を回せるようになった（当社比）
- 反省点
  - LambdaのタイムアウトとAPI Gatewayのタイムアウトが噛み合わず不具合が…
    - API GatewayはタイムアウトしたがLambdaはDBに書き込んでしまった
    - WEBアプリとしてAPI Gatewayのタイムアウトに合わせること
    - 各マネジメントサービスの制約をしっかりと確認すること
      - プロダクトがスケールしていけば、メモリや同時実行数など、考えることも増える

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

DB選択のふりかえり

- よかった点
  - PoC開発を中心に考えると、RDB選択時にAurora Serverlessは最適だった
    - 過去のVPC Lambdaにおける課題もAWSさんの改善で解決（ありがとうございます）
      - 最近はRDS Proxyの発表もあり、再検討予定
  - RDBを選択したことによるボトルネックは現状発生していない
    - ただし、書き込みスループットの課題が発生しそうな直近案件はよく考える
- 反省点
  - DynamoDBのシビアな検索要件に対して検討が足りない部分があった
    - 「小さく始める」チームであってもよく考えていきたい
    - ハイブリッド構成等も積極的に考えていく
      - チャレンジは忘れない！

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

今後に向けて

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with

We’re hiring!

- 組織もプロダクトもスケーリング中
  - 本気でDXを実現したい志をお持ちの方
  - 企画、デザイナーが一体となったアジャイル開発が得意な方
  - AWSが大好きな方、ともにアーキテクチャを考えたい方（大募集）
  - 開発だけではなく、監視と運用も得意な方（大大募集）

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In Partnership with intel

aws DEV DAY
ONLINE JAPAN

Thank you!

SOMPOホールディングス株式会社
杉山祐介

In Partnership with intel

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws DEV DAY
ONLINE JAPAN

Please complete the session survey in the mobile app.

In Partnership with intel

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.