

AWS INNOVATE 2020 オンラインカンファレンス

「30分で理解する！SaaSとAWSのサーバーレスなつなぎ方」のクイズおよび解答

AWS INNOVATE 2020のセッションの視聴およびアンケートにご記入頂きありがとうございます。本資料が「30分で理解する！SaaSとAWSのサーバーレスなつなぎ方」で出題されたクイズの解答になります。

問題1：SaaSアプリケーションとの連携において、サーバーレスなアーキテクチャを構成することで得られるメリットには、どのようなものが考えられるでしょうか。

解答1:

SaaSアプリケーションとの連携においてAWSのサーバーレスなアーキテクチャを採用することで得られるメリットには、以下のような内容が考えられます。

- ・SaaSプロバイダーが提供するWebhookやAPIを処理した量に応じて、課金対象となる金額が決まるため、ランニングコストの削減が期待できる。
- ・外部のAPIを定期的にポーリングする仕組みや、Webhookの受け口となるWebサーバーなど、SaaSアプリケーションとの連携に必要なミドルウェアの構築や運用から解放される。
- ・サーバーレスなサービスにあらかじめ組み込まれた、耐障害性に関する仕組みをそのまま利用できるため、高い可用性のもとにSaaS連携を構築できる。
- ・必要に応じて自動的に性能がスケールするため、サーバーを何台配置すべきかなど、スケーラビリティに関して考慮すべき内容が少なくなる。

そのため、SaaSアプリケーションとの連携に関しては、サーバーレスの特徴となる様々なメリットを十分に活用することができます。

問題2：複数の SaaS アプリケーションと、複数の AWS 上のサービスを連携させたいとき、何が課題となり、どのように解決できるでしょうか。

解答2:

SaaS アプリケーションと AWS 上のサービスとの関係が、一対一ではなく多対多になった場合、まず発生するのが「どのサービスどうしを繋ぐか」という問題です。つまり、ある SaaS アプリケーションからの通信を、適切なサービスに対してルーティングするという機能が求められます。

また、一部の通信が、他の通信に悪い影響を与えないよう、SaaS アプリケーションと AWS 上のサービスとの間を非同期、かつ疎結合にすることが求められます。悪影響の例としては、ある SaaS アプリケーションの Webhook が、大量の HTTP リクエストを送信したことで、別の SaaS アプリケーションが応答を返せなくなってしまうようなケースが考えられます。

これら 2 点の問題を解決するために、イベントバスを利用したアーキテクチャパターンの導入、すなわち「イベント駆動アーキテクチャ」を紹介しました。イベントバスは、SaaS アプリケーションで発生したデータを一時的に受け付け、あとで (非同期に) 後段のサービスへ連携することが可能なものになります。SaaS アプリケーションにて連携が必要なデータが発生したことをイベントのトリガーとして、適切なサービスを呼び出すことから、セッション内ではこのような構成をイベント駆動アーキテクチャの一例としてご紹介しました。

サーバーレスなイベントバスサービスとして、AWS では 2019 年より Amazon EventBridge というサービスを提供しています。これにより、サーバーなどの管理を行わずに、イベントバスの仕組みを皆様のアーキテクチャに組み込むことができますので、SaaS 連携が複雑化した際には、導入を検討すると良いと言えます。