

製品オプションを識別する

100 XP

5 分

サーバーレス コンピューティングの概要については、こちらの動画をご覧ください。

動画の中で述べられているように、サーバーレス コンピューティングはクラウドでホストされる実行環境で、ユーザーのコードを実行しますが、基になるホスティング環境は抽象化されています。"サーバーレス コンピューティング" という用語は、適切ではありません。結局のところ、コードまたは必要な機能を実行するサーバー (またはサーバーのグループ) は "存在します"。

重要なのは、そのサーバーの設定や保守について、ユーザーは責任を負わないという考えです。需要が増加してもスケーリングについて心配する必要はありません。また、停止する心配もありません。ユーザーに代わってクラウド ベンダーが、メンテナンスやスケーリングに関するあらゆる懸念に対処します。

ユーザーは、サービスのインスタンスを作成してから、自分のコードを追加します。インフラストラクチャの構成やメンテナンスを求められることはなく、許可さえされません。イベントに応答するようにサーバーレス アプリを構成します。イベントとしては、REST エンドポイント、定期的なタイマー、別の Azure サービスから受信したメッセージなどがあります。サーバーレス アプリは、イベントでトリガーされた場合にのみ実行されます。スケーリングとパフォーマンスは自動的に処理され、使用したリソースに対してのみ課金されます。リソースの予約さえも必要ありません。

通常、サーバーレス コンピューティングは、"バックエンド" のシナリオを処理するために使用されます。つまり、サーバーレス コンピューティングには、あるシステムから別のシステムへのメッセージの送信または、他のシステムから送信されたメッセージの処理を行う役割があります。ユーザー向けのシステムに使用されるのではなく、バックエンドで動作します。

このモジュールでは、2 つの Azure サーバーレス コンピューティング サービス (Azure Functions と Azure Logic Apps) について説明します。

Azure Functions

Azure Functions サービスを使用すると、クラウド内で一般的なプログラミング言語を使用して、イベントに応答して実行される 1 つのメソッドまたは関数をホストすることができます。イベントの例としては、HTTP 要求、キューの新しいメッセージ、タイマーのメッセージなどがあります。

Azure Functions は、そのアトミックな性質により、アプリケーションの設計でさまざまな目的に使用することができます。関数は、C#、Python、JavaScript、Typescript、Java、PowerShell など、多くの一般的なプログラミング言語を使用して記述できます。

Azure Functions は自動的にスケーリングされ、関数がトリガーされたときにのみ課金が発生します。こうした品質があるので、需要が変化する場合には Azure Functions が確実な選択肢になります。

す。たとえば、すべての配送車両を監視する IoT ソリューションからメッセージを受信しているとします。業務時間中は受信データが増える可能性があります。Azure Functions では、これらのより多忙な時間に対応するためにスケールアウトできます。

Azure 関数はステートレス環境です。関数は、イベントに応答するたびに再起動されたかのように動作します。この機能は、受信データを処理するために最適です。また、状態が必要な場合は、関数を Azure Storage アカウントに接続することができます。

Azure Functions では、Durable Functions と呼ばれる拡張機能を使用してオーケストレーション タスクを実行できます。これにより、開発者は状態を維持しながら Functions を連結できます。

基盤となるプラットフォームやインフラストラクチャではなく、サービスを実行しているコードのみに関心がある場合には、Azure Functions ソリューションが最適です。Azure Functions は、イベントに応答して作業を実行する必要がある場合に最も一般的に使用されます。多くの場合、REST 要求、タイマー、または別の Azure サービスからのメッセージを介して、その作業をすばやく (数秒以内に) 完了できるときに、これを実行します。

Azure Logic Apps

Logic Apps は、クラウド サービスとしてホストされるローコードまたはノーコード開発プラットフォームです。このサービスを使用すると、企業全体または組織全体でアプリ、データ、システム、サービスを統合する必要がある場合に、タスク、ビジネス プロセス、ワークフローを自動および調整することができます。Logic Apps によって、クラウド、オンプレミス、またはその両方で、スケーラブルなソリューションを簡単に設計および構築できるようになります。このソリューションは、アプリ統合、データ統合、システム統合、エンタープライズ アプリケーション統合 (EAI)、および企業間 (B2B) 統合を対象としています。

Azure Logic Apps は、Web ベースのデザイナーで設計されており、コードを記述せずに Azure サービスによってトリガーされるロジックを実行できます。アプリをビルドするには、コネクタを使用してトリガーをアクションにリンクします。トリガーは、タイマーなどのイベントです。これによって、アプリを実行する、新しいメッセージをキューに送信する、または HTTP 要求を送信することができます。アクションは、実行可能なタスクまたはステップです。ほとんどのプログラミング言語で見られるようなロジックアクションがあります。アクションの例としては、変数、デシジョンのステートメントとループ、データを解析および変更するタスクの操作などがあります。

Azure Logic Apps を使用してエンタープライズ統合ソリューションをビルドするには、200 以上のコネクタのギャラリーから選択でき、この数は増え続けています。ギャラリーには、Salesforce、SAP、Oracle DB、ファイル共有などのサービスがあります。

必要なアクションまたはコネクタが見つからない場合は、カスタム コードを使用して独自にビルドすることができます。

これらのサービスの相違点

Azure Functions を Azure Logic Apps から呼び出すことができ、その逆も可能です。2 つのサービスの主な相違点は、その目的です。Azure Functions はサーバーレス コンピューティング サービスであり、Azure Logic Apps はサーバーレス オーケストレーション サービスを目的としています。Azure Functions を使用すると、さまざまな接続が関係する実行時間の長いビジネス プロセスを調整できますが、これは、設計時には主要なユース ケースではありませんでした。

また、2 つのサービスの料金は異なります。Azure Functions の料金は、実行回数と、各実行の実行時間に基づいています。Logic Apps の料金は、実行回数と、利用されるコネクタの種類に基づいています。