

どのようなときに Azure Functions を使用するか

100 XP

10 分

Tailwind Traders の何人かの開発者と話し合った結果、アプリケーションのロジックの一部はイベントドリブンであると判断しました。つまり、アプリケーションは、処理を実行する前に、特定の入力を長い時間待っています。コストを削減するため、アプリケーションが入力を待機している時間に対して料金を支払う必要がないようにしようと考えます。この点を念頭に置いて、Azure Functions が役に立つかどうかを調べることにしました。

"サーバーレス" コンピューティングは、サーバー、インフラストラクチャ、オペレーティングシステムを抽象化したものです。サーバーレス コンピューティングを使用すると、Azure によって、サーバー インフラストラクチャや、必要に応じたリソースの割り当てと割り当て解除が管理されます。ユーザーにはインフラストラクチャに関する責任はありません。スケーリングとパフォーマンスは自動的に処理されます。厳密に使用したリソースに対してだけ課金されます。容量を予約する必要もありません。



サーバーレス コンピューティングには、サーバーの抽象化、イベントドリブンのスケーリング、マイクロビリングが含まれます。

- **サーバーの抽象化:** サーバーレス コンピューティングでは、実行先のサーバーが抽象化されます。ユーザーが明示的にサーバー インスタンスを予約することはありません。それは、プラットフォームによって自動的に管理されます。各関数の実行を、異なるコンピューティング インスタンスで処理できます。この実行コンテキストは、コードに対して透過的です。サーバーレス アーキテクチャを使用すると、コードをデプロイすると、高可用性で実行されます。
- **イベントドリブン スケール:** サーバーレス コンピューティングは、受信イベントに応答するワークロードに最適です。イベントには次のものによるトリガーが含まれます。
 - タイマー。たとえば、毎日午前 10:00 UTC に関数を実行する必要がある場合。

- HTTP。たとえば、API や Webhook のシナリオ。
- キュー。たとえば、注文処理によって。
- その他にもたくさんあります。

開発者は、アプリケーション全体を記述するのではなく、トリガーとバインドに関するコードとメタデータの両方を含む関数を作成します。プラットフォームにより、実行する関数が自動的にスケジュールされ、受信イベントの割合に基づいてコンピューティング インスタンスの数がスケーリングされます。トリガーにより、関数を呼び出す方法が定義されます。バインドにより、コード内からサービスに接続するための宣言型の方法が提供されます。

- **マイクロビルディング**: 従来のコンピューティングの場合、Web サイトのホスティングに対する月単位または年単位の料金のように、時間のブロックに対して請求されます。この課金方法は便利ですが、コスト効率に優れているとは限りません。顧客の Web サイトのヒット数が 1 日に 1 回だけの場合でも、1 日分の可用性に対する料金が発生します。サーバーレス コンピューティングでは、コードが実行された時間に対してのみ課金されます。アクティブな関数の実行が発生しなかった場合は、課金されません。たとえば、コードが 1 日に 1 回、2 分間実行される場合、1 回の実行と 2 分間のコンピューティング時間に対して課金されます。

Azure のサーバーレス コンピューティング

Azure には、2 つのサーバーレス コンピューティングの実装があります。

- **Azure Functions**: Functions を使用すると、ほぼすべての最新の言語でコードを実行できます。
- **Azure Logic Apps**: 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。

Azure Functions

サービスを実行しているコードのみに関心があり、基になるプラットフォームやインフラストラクチャには関心がない場合は、Azure Functions が最適です。Functions は、イベント (多くの場合 REST 要求による)、タイマー、または別の Azure サービスからのメッセージに応答して処理を実行する必要があり、数秒以内にすばやく処理を完了できる場合に、よく使用されます。

Functions は必要に応じて自動的にスケーリングされます。そのため、需要が変化する場合、堅実な選択肢となります。たとえば、配送車群を監視するために使用されている IoT ソリューションからのメッセージを受信するような場合があります。業務時間中は受信データが増える可能性があります。

仮想マシン ベースのアプローチを使用すると、仮想マシンがアイドル状態であってもコストが発生します。関数を使用する場合、Azure では、トリガーされたコードが実行され、関数が完了したとき、リソースの割り当てが自動的に解除されます。このモデルでは、関数の実行中に使用された CPU 時間に対してのみ課金されます。

Functions は、ステートレスまたはステートフルにできます。ステートレス (既定) の場合は、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。ステートフル (Durable Functions と呼ばれます) の場合は、前のアクティビティを追跡するために、コンテキストが関数に渡されます。

Functions は、サーバーレス コンピューティングの主要なコンポーネントです。また、任意の種類のコードを実行するための一般的なコンピューティング プラットフォームでもあります。開発者のアプリのニーズが変化した場合は、サーバーレスではない環境にプロジェクトをデプロイできます。この柔軟性により、スケーリングを管理し、仮想ネットワーク上で実行し、関数を完全に分離することさえできます。

Azure Logic Apps

Logic Apps は Functions に似ています。どちらも、イベントに基づいてロジックをトリガーすることができます。Functions の場合はコードが実行されますが、Logic Apps の場合は、ビジネス シナリオを自動化するために設計され、事前定義されたロジック ブロックから構築された "ワークフロー" が実行されます。

Azure ロジック アプリのすべてのワークフローは、トリガーによって起動され、トリガーは特定のイベントが発生するか、新たに利用可能になったデータが特定の条件を満たした時点で生成されます。多くのトリガーには、開発者がワークロードの実行頻度を指定できる基本的なスケジューリング機能が備わっています。トリガーが起動されるたびに、Logic Apps エンジンによって、ワークフロー内でアクションを実行するロジック アプリ インスタンスが作成されます。これらのアクションにはデータ変換のほか、条件文、switch ステートメント、ループ、分岐などのフロー制御が含まれることもあります。

Azure portal または Visual Studio でビジュアル デザイナーを使用し、ロジック アプリのワークフローを作成します。ワークフローは、既知のワークフロー スキーマを利用し、JSON ファイルとして保持されます。

Azure からは、さまざまなサービスとやりとりするために、200 を超えるさまざまなコネクタと処理ブロックが提供されています。これらのリソースには、最も人気のあるエンタープライズ アプリが含まれています。やりとりする必要があるサービスが含まれていない場合、カスタムのコネクタやワークフロー ステップを作成することもできます。次に、ビジュアル デザイナーを使用してコネクタとブロックをリンクします。ワークフロー経由でデータを渡し、カスタム処理を実行します。多くの場合、コードを記述することなくすべて行われます。

たとえば、Zendesk にチケットが届くとしします。以下のような方法で対処できます。

- Cognitive Services を利用し、メッセージの意図を検出します。
- SharePoint で項目を作成し、問題を追跡します。
- まだデータベース内に存在しない顧客は、Dynamics 365 CRM システムに追加します。
- フォローアップ メールを送信し、要求が届いたことを知らせます。

それらのすべてのアクションをビジュアル デザイナーで設計できるため、ロジックの流れを簡単に確認できます。このため、ビジネス アナリストの役割に最適です。

Functions と Logic Apps

Functions と Logic Apps では、いずれも複雑なオーケストレーションを作成できます。オーケストレーションは、複雑なタスクを遂行するために実行される関数またはステップの集まりです。

- Functions を使用して、各ステップを完了するためのコードを記述します。
- Logic Apps では、GUI を使用して、アクションと、それらが相互にどのように関係するかを定義します。

オーケストレーションを構築するときは、ロジック アプリから関数を呼び出したり、関数からロジック アプリを呼び出したりしながら、さまざまなサービスを組み合わせることができます。この 2 つには一般的な違いがいくつかあります。

| | Functions | Logic Apps |
|----------|-----------------------------------------------|---------------------------------------------------------------------|
| 状態 | 通常はステートレスですが、Durable Functions ではステートが与えられます。 | ステートフル。 |
| 開発 | コード優先 (命令型)。 | デザイナー優先 (宣言型)。 |
| 接続 | 組み込みのバインド型 (約 10 種類)。カスタム バインドのためにコードを書きます。 | コネクタの豊富なコレクション。B2B 向けの Enterprise Integration Pack。カスタム コネクタを作成します。 |
| アクション | 個々のアクティビティが Azure 関数です。アクティビティ関数のコードを書きます。 | 既製のアクションの豊富なコレクション。 |
| 監視 | Azure Application Insights。 | Azure portal、Log Analytics。 |
| 管理 | REST API、Visual Studio。 | Azure portal、REST API、PowerShell、Visual Studio。 |
| 実行コンテキスト | ローカルまたはクラウドで実行できます。 | クラウドでのみ動作します。 |