

決定の条件を分析する

100 XP

5 分

実行可能なサーバーレス オプションが 2 つあるため、どちらがジョブに最適であるかを判断するのが難しい場合があります。このユニットでは、エキスパートが特定のビジネス ニーズに対してどのサーバーレス サービスを使用するかを選択するときに採用する条件を分析します。この条件を理解することは、製品の微妙な違いをより深く理解するためにも役立ちます。

よく知られている API 全体でオーケストレーションを実行する必要がある場合

前述のとおり、Azure Logic Apps は、Web ベースのビジュアル コンフィギュレーターから価格モデルまで、オーケストレーションを考慮して設計されました。Logic Apps は、ワークフローの多くのステップでデータを渡して処理するために、API を介して多数の異なるサービスを接続することに優れています。

Azure Functions を使用して同じワークフローを作成できますが、呼び出す API とその呼び出し方法を調べるにはかなりの時間がかかる可能性があります。Azure Logic Apps にはこれらの API 呼び出しが既にコンポーネント化されているため、わずかな詳細を指定するだけで済み、必要な API を呼び出す詳細は抽象化されています。

カスタム アルゴリズムを実行したり、特殊なデータ解析やデータ検索を実行したりする必要がある場合

Azure Functions の場合、プログラミング言語のすべての表現力をコンパクトな形式で使用できます。これにより、複雑なアルゴリズムや、データのルックアップと解析操作を簡潔に構築できます。ユーザーは、コードの保守、回復性の高い例外処理などに対して責任があります。

Azure Logic Apps からはロジック (ループ、決定など) を実行できますが、複雑なアルゴリズムを必要とするロジック集中型のオーケストレーションがある場合、そのアルゴリズムの実装はより詳細になり、読むことも困難になります。

命令型プログラミング言語で記述された既存の自動化タスクがある場合

C#、Java、Python などの一般的なプログラミング言語で記述されたオーケストレーションまたはビジネス ロジックが既にある場合、Azure Logic Apps を使用してコードを再作成するよりも、ご利用のコードを Azure Functions の関数アプリの本文に移植する方が簡単な場合があります。

ビジュアル (宣言型) ワークフローと記述 (命令型) コードのどちらを使用するか

この選択は、最終的に宣言型環境と命令型環境のどちらで作業したいかによって決まります。命令型プログラミング言語の専門知識を持つ開発者は、命令型の思考法から自動化とオーケストレーションについて考えることを好む可能性があります。IT プロフェッショナルやビジネス アナリストは、より視覚的なローコードまたはノーコードの (宣言型) 環境で作業することを好む可能性があります。