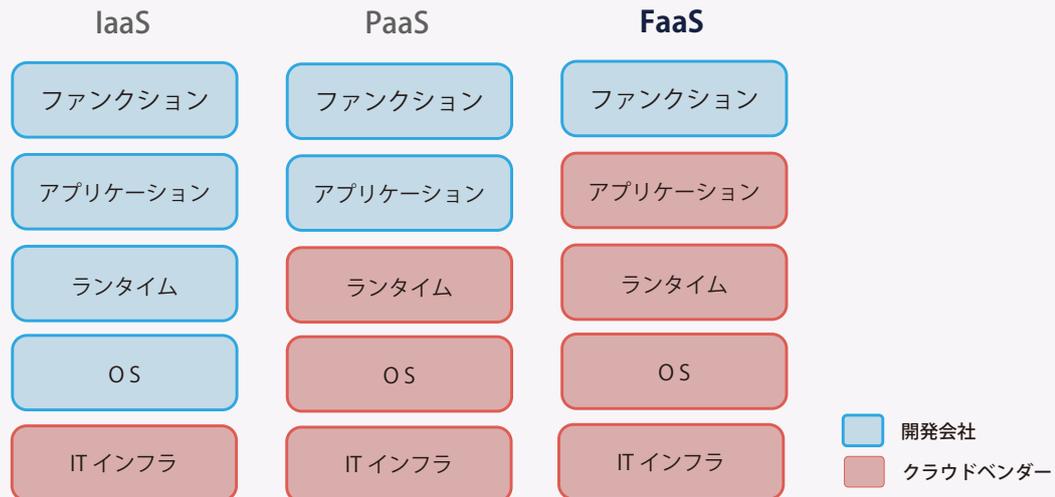


# サーバレスでのアプリ開発

## マイクロサービスに最適な FaaS

マイクロサービスのアプリケーションに適したサーバレスのアーキテクチャー  
アプリケーションの実行に必要なサーバーを自動で振り分け、その都度スケール  
イベント駆動で、実行終了後破棄され、大規模なバッチ処理には不向き



アジャイル開発でマイクロサービスが注目されてくると、それを実行できるプラットフォームを提供できるサービスへの需要が高まります。

クラウドサービスを使って開発をする場合、ユーザが望むことは、マシンパワーを意識せず利用できることも重要ですが、主にインフラ投資や運用スタッフの負担を無くしたいということです。ユーザは煩わしいマシン運用は任せて、開発に集中したいという希望があり、そのためにアプリケーションソフトが稼動するためのハードウェアや OS などのプラットフォーム一式を、クラウドサービスとして提供する PaaS (Platform as a Service) が、最適だといわれ AWS (Amazon Web Services) や、Microsoft Azure などが多くの企業や組織団体で利用されています。

近年では、さらにマイクロサービスによって構築されるシステム向けに最適化された機能 (Functions) 毎に実行環境がサービスとして提供される FaaS (Function as a Service) というアーキテクチャも登場しています。

必要最小限の機能のみを提供するコンポーネントを多数組み合わせるマイクロサービスでは、コンポーネントの分割をさらに細かく行うことで、保守コストやプログラムの複雑性を軽減し、開発効率を向上させるのですが、機能を分割していくと利用頻度の少ない機能が発生してしまい、使用するマシンやプログラムも増えてしまいます。これを、効率よく実装し管理できるようにしたのが、FaaS というサーバレスでアプリケーション開発を行えるサービスです。

FaaS では、WebAPI を経由して、細かな機能をサービスとして提供しており、イベントドリブン方式でサービスのコードを作成し、それを連携させるだけでコンテナ上で実行でき、終了したら破棄されます。

AWS の Lambda、Google の Cloud Functions、Microsoft の Azure Functions などが有名ですが、その多くは、各機能のリクエスト件数に対する課金で、毎月最初の 100 万件などのリクエスト無料枠が設けられています。これにより、導入の初期投資が削減され、使用頻度の少ない機能があっても、コストを気にせず、短期間での開発が可能です。

開発者の多くは、できることなら、コストは気にせずテストや実装ができ、環境に依存したコーディングを気にしないで開発できるサーバレスな環境を望んでいます。つまり、サーバレスとは、サーバの運用や管理が不要ということです。

FaaS を利用すれば、オンプレミスのようにサーバの維持管理に悩まされないサーバレスの環境で、Web アプリケーションなどの開発において、一番効率的で低コストなマイクロサービスの開発を実現できます。