モダンアプリケーション: マイクロサービスアーキテクチャ

ビジネスの俊敏性を実現するモダンアーキテクチャ

高速で手頃な価格のコンピューティングが普及したことで、あらゆる規模の企業が社内効率を高め、デジタル製品を通じてより多くの顧客にサービスを提供できるようになりました。しかし、ツールの普及、マーケットへの多数の経路、消費者の嗜好の変化により、企業は競争力を維持するために、これまで以上に迅速に技術革新を行う必要があります。農業から銀行、通信に至るまで、あらゆる業界において、急速なイノベーションの中核にはデジタルがあります。デジタル製品の従来の構築方法は、市場での成功に必要な革新を実現するために十分なスピードを備えていません。

新しいアーキテクチャパターン

マイクロサービスなどの新しいアーキテクチャパターンにより、組織はイノベーションのペースを加速できます。マイクロサービスアーキテクチャで構築されるモダンアプリケーションは、その労力と投資を経時的に複数の小規模チームに分散、テストのスピードを早めて市場に変更を投入、イノベーションを可能にします。モダンアプリケーションにより、リソースをきめ細かく最適化でき、チームは製品の構築方法と運用方法の両方を迅速にスケーリングできます。

マイクロサービスアーキテクチャとは?

特殊化

各サービスは任意の機能セット向けに設計され、特定の問題解決に重点を置いています。デベロッパーによってコードがサービスに徐々に追加され、そのサービスが複雑になった場合は、規模の小さい複数のサービスに分割することができます。

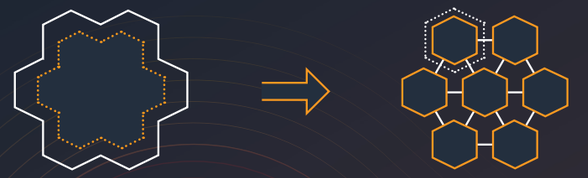
分散型

マイクロサービスアーキテクチャは、アプリケーションを 1 つのプロセスから複数のコンポーネントに分割し、連携させることで価値を提供します。個々のコンポーネント間の通信は、明確に定義された疎結合 API 、またはイベントとメッセージングによって行われます。

自律型

マイクロサービスアーキテクチャ内の各コンポーネントサービスは、他のサービスの機能に影響を与えることなく、開発、デプロイ、運用、スケーリングが可能です。サービスは、コードや実装を他のサービスと共有する必要はありません。それらは自己完結型のブラックボックスとして機能します。

モノリシックアーキテクチャは今のところうまく機能するかもしれませんが、ビジネスの成長に伴って頻繁に課題が生まれてくるでしょう。マイクロサービスは、新しい機能のスケーリングやデプロイなどの一般的な課題への迅速な対応を促進します。



モノリスの分解

モノリスの分解は、なかなか勇気のいることです。あなたのチームに AWS プロジェクトを完成させてもらうことで、モノリシックアーキテクチャを複数のマイクロサービスに分解するプロセスを経験してもらいましょう。

チュートリアルを試す >>

マイクロサービスへの移行

Mobvista 社がマイクロサービスアーキテクチャを採用し、自社のプラットフォームの拡張性と安定性を向上させた方法をご紹介します。導入事例を読む >>

マイクロサービスの優位性

マイクロサービスアーキテクチャは、連携して動作する個別のモジュール式要素で構築されています。モジュール式にはコードの「表面積」が増加するという課題が伴いますが、革新の迅速化、個別のスケーリング、障害時の影響の軽減、分散型コード開発などの重要な優位性を得ることができます。

マイクロサービスのメリット

1.俊敏性: 個々のサービスコンポーネントに取り組んでいる小規模なチームは、他のコンポーネントの制約から解放されるため、より機敏に問題や機会に対応できるようになります。

2.容易なデプロイ: マイクロサービスは変更のサイズを小さくすることで、新しいアイデアを試す、うまくいかない場合はロールバックするなどの作業を容易にします。

3.技術的な自由: マイクロサービスアーキテクチャにより、チームはアプリケーションの各部分を作成するための最適なツールを自由に選択できます。

4.スケーラビリティ: マイクロサービスでは、各サービスを個別にスケーリングして、サポートするアプリケーション機能の需要に対応できます。

5.耐障害性: マイクロサービスは、アプリケーションの 1 つの部分に障害が発生した場合の影響を軽減します。つまり、個々のコンポーネントの障害は、アプリケーション全体のクラッシュではなく、機能の低下のみに抑えられます。

6.再利用可能なコード: ソフトウェアを小さな正確に定義されたモジュールに分割すると、チームはこれらのサービスコンポーネントをアプリケーション内で複数の目的に使用できます。これにより、デベロッパーはコードを最初から記述したり、実装の詳細を解決したりすることなく、API を使って既存のサービスを活用し新しい機能を構築できるようになります。

モダンアプリケーションを実現するマイクロサービス

マイクロサービスは、アプリケーションの耐障害性を向上させ、チームの生産性の最適化を促進します。その結果、開発チームはより迅速に実験と革新を行い、競争優位性をもたらす製品と機能をリリースすることができます。

最初のプロセス

マイクロサービスのアーキテクチャパターンを採用することで、失敗すればすべてを失うような努力をする必要がなくなります。サービス指向アーキテクチャには、(a) 既存のモノリスを API でラップし、それをブラックボックスとして処理しながら、新しい実質的な機能をマイクロサービスとして構築する方法と、(b) ストラングラーパターンを使用してモノリスをマイクロサービスへリファクタリングする方法という 2 つの一般的なアプローチがあります。どちらの方法にもメリットとデメリットがありますが、どちらの方法を選択しても、まず適切な開発インフラストラクチャを設定する必要があります。これには、実行可能なサービスを個別に構築、テスト、デプロイするための自動ソフトウェア配信パイプライン、および分散システムのセキュリティ保護、監視、運用、デバッグを行うためのインフラストラクチャの構築が含まれます。

主要な機能の更新を必要としないスタンドアロンシステムの場合は、モノリスをそのまま機能させることができます。この場合、新たな開発に関する労力のほとんどは新しいマイクロサービスの構築に集中させることができます。このマイクロサービスは単純に API を使ってモノリスに接続させることができます。モノリスを維持または破棄できないが、その一部を書き直す必要がある場合は、最善の方法でストラングラーパターンを使用します。

ストラングラーパターンを使用する開発チームは、既にモノリスから分離できるであろう機能をある程度識別しています。これらの機能は API の後ろでモノリスから分離されるため、マイクロサービスが構築された際に簡単に交換および廃止することができます。つまり、多くのクライアント向けアプリケーションに変更を加える必要がなく、独自のデータストアを必要としない可能性が高い機能が最初の候補にあがります。例えば、e コマースアプリケーションでは、認証、請求、顧客プロファイルなどのサービスを検討できます。初めてマイクロサービスへの分割を行う際は、ほとんどのチームは、機能の最適化ではなく、まずはソフトウェア配信パイプラインや API アプローチのテストと最適化、そしてチームメンバーのスキル向上からスタートしています。

お客様のビジネスにおけるマイクロサービスの可能性については、**こちらをご覧ください。**

© 2020, Amazon Web Services, Inc. or its affiliates.All rights reserved.