

活発化するクラウドネイティブ開発のアーキテクチャ解説と導入事例紹介 ～ クラウドの特長を最大限に活かし、保守性の高い基盤構築や迅速なサービス提供を実現 ～

昨今、ビジネス環境の変化に対応するために各社が DX を推進しています。それを受け、クラウド活用もますます活発化しており、クラウドを使うことを前提にシステムの設計や開発をする「クラウドネイティブ」に注目が集まっています。本レターでは、クラウドネイティブの考え方とその事例についてご紹介いたします。

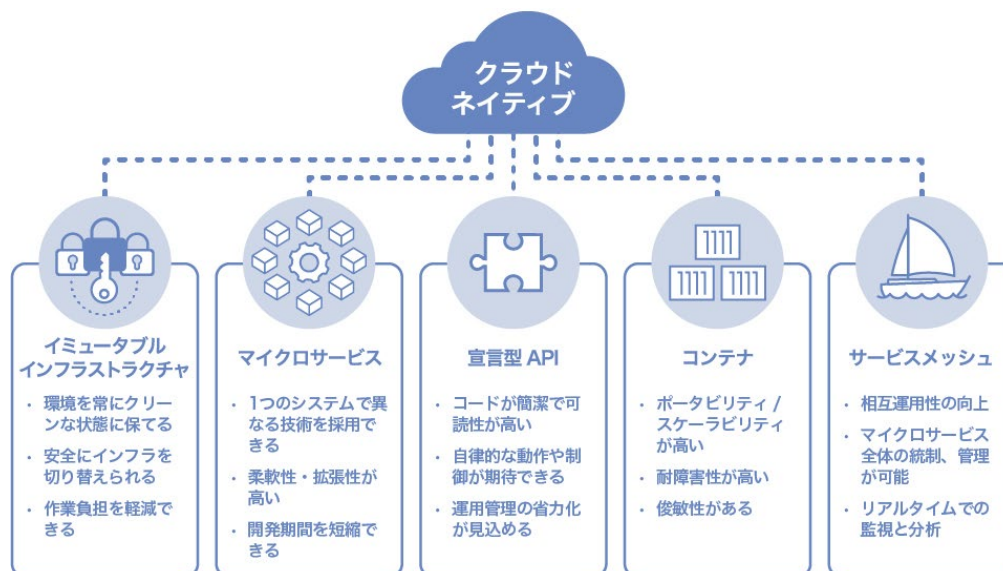
クラウドネイティブとは

クラウドネイティブコンピューティング技術を定義・推進している非営利団体である Cloud Native Computing Foundation（以下、「CNCF」）は、クラウドネイティブテクノロジーを「パブリッククラウド、プライベートクラウド、ハイブリッドクラウドなどの近代的でダイナミックな環境において、スケーラブルなアプリケーションを構築、実行するための能力を組織にもたらす」と定義しています。

クラウドネイティブは、システムを使う方の需要や要望に合わせて規模を調整できる性質を持っているため、柔軟性に優れています。また、頻繁にアプリケーションが変更されることを前提としているため、柔軟で自由度の高いシステムを構築することも可能です。

クラウドの特長を最大限に活用・最適化するクラウドネイティブアーキテクチャ

スケーラブルなクラウドネイティブアプリケーションを構築および実行するために使用するソフトウェアコンポーネントの組み合わせをクラウドネイティブアーキテクチャと呼びます。CNCF では、クラウドネイティブアーキテクチャの技術的要素として、イミュータブルインフラストラクチャ、マイクロサービス、宣言型 API、コンテナ、サービスメッシュを挙げています。これらのアーキテクチャ要素を採用することで、疎結合で復元力があり、管理しやすく、オプザバビリティ（可観測性）に優れたシステムを構築することが可能になります。



イミュータブルインフラストラクチャ

一度構築したら、その後はパッチ適用の管理を行わずに、インフラ全体を特定の状態で固定化します。インフラ構成の変更等が必要な場合には、新規にインフラ環境を構築し、既存環境から新環境に全体を切り替えて運用します。IaC（Infrastructure as Code）という手法を採用してインフラ構築をすべてコード化することにより、構築や廃棄、起動などの作業の負担を軽減することが可能となります。

マイクロサービス

複数の独立した小規模なサービスを組み合わせて、一つのアプリケーションを構築するためのアーキテクチャです。アプリケーションのスケールを容易にし、開発期間を短縮できるため、イノベーションの実現と新機能の市場投入の加速に効果的と言われています。

宣言型 API

サービスがどのような状態にあるべきかを宣言する API です。例えば、コンテナ数を制御する場合、従来の命令型 API ではコンテナ数の調整をその都度実行する必要がありました。それに対して、宣言型 API を利用してコンテナ数を維持するように宣言することで、自立的なコンテナ数の増減が可能となり、結果としてアプリケーションの柔軟性が高まり、サービス運用の複雑さも軽減されてシステムがシンプルになります。

コンテナ

アプリケーションを実行するための依存関係をコンテナイメージとしてパッケージ化し、任意の場所で実行されるコンテナランタイムによりコンテナとして実行するテクノロジーです。コンテナを利用することで、アプリケーションが可搬性を得るため、開発環境や本番環境など様々な環境で一貫性のある形でアプリケーションを実行することができます。また、起動時間の短縮、リソースの使用率の向上、アプリケーションのライフサイクル管理の改善も期待できます。

サービスメッシュ

分散コンピューティング向けのサービスマネージャの一種で、マイクロサービスなどのサーバー側アプリケーションを構成する小さなコンポーネント群が互いに動的にリンクされ、1つのアプリケーションとして機能できるようになります。サービスメッシュが提供する代表的な利点として、分散システムのセキュリティ、オブザーバビリティ、トラフィック制御の向上があげられます。

クラウドネイティブな開発により、保守性の高い基盤構築や迅速なサービス提供を可能に

クラウドネイティブな開発により、保守性の高い基盤の構築やプロセスを自動化し迅速なサービス提供を可能にした事例をご紹介します。

事例 1：ブロックチェーンシステムのクラウド基盤の開発支援（金融企業）

ブロックチェーンは、データの改ざんなどが難しく、安全性も高いという特長がある一方で、分散型のため従来のシステムと比べるとデータ処理に時間がかかるというデメリットがあります。日々変化する状況に対応するためには、迅速なサービスの提供が求められるため、それを実現するために既存のシステムの一部をクラウドに移行する支援をおこないました。アプリケーションの実行基盤として Kubernetes をコンテナオーケストレーターとして構築し、Lambda でサーバーレス基盤を構築して基盤系の運用作業の効率化を図りました。結果、基盤障害からの回復やリソース使用状況に応じた柔軟なスケールリングを実現しました。

事例 2：サプリメント・ヘルスケアサービスにおけるクラウド基盤の維持（健康食品企業）

サービスのリリース頻度を増やすための開発スピードの加速化や品質の向上が求められるなか、クラウド基盤の維持と保守開発（SRE 業務）をスクラム開発でおこないました。SRE 業務では、機能の拡張に向けて既存のサービスへの機能追加や性能の向上を図ったほか、クラウドリソースの最適化や 2 週間に 1 回の定期的なリリースをしました。使用していた製品のサポートやサービス期間が終了する EOSL（End of Service Life）対応もおこない、サービスが安定的に稼働し、品質を維持しました。さらに、開発スピードを速めリリース頻度を高めたことに加え、品質の向上にも貢献しました。

事例 3：既存サイトのアーキテクチャの刷新（サービス業）

昨今のサービスにおいては、ビジネス環境の変化にすばやく対応し、ユーザーが求める価値をいち早く届けることが求められます。それに対応するために、既存サイトのアーキテクチャの刷新をおこないました。マネージドサービスを積極的に活用し、構築の自動化により保守性の高い基盤構築を実現しました。アプリケーションの実行基盤には ECS/Fargate 構成のコンテナモデルを採用し、IaC による自動構築をおこないました。また、迅速にサービスを提供することを可能にするためにテストやリリースを自動化するための CI/CD 環境の構築を GitHub Actions と AWS マネージドサービスを利用して実現したほか、クラウド特性を損なわないようにするために SaaS を導入した統合モニタリング方式を採用しました。その結果、パッチ適用などの一部の運用を AWS にオフロードすることによる負荷軽減や、計画停止を伴わないリリース頻度の向上、3rdParty ツール(SaaS) への情報集約や可視化により早期の状況把握を可能にし、リソース状況に応じたリソースのスケール検討などに繋がりました。

今後も NTT データ ニューソンでは、クラウドを用いたインフラ技術だけではなく、アプリケーション技術、運用技術にも熟知した開発者がクラウド特性を最大限に活かすクラウドネイティブな開発を支援いたします。

【参考情報】

・NTT データ ニューソンのクラウドインテグレーションサービスについて

<https://www.newson.co.jp/services/ITInfra/CloudIntegration/>

＜関連コラム＞

・第 1 回：クラウドのマーケットと現状の課題

<https://www.newson.co.jp/column/2023102403/>

・第 2 回：クラウド特性を最大限にいかすクラウドネイティブ

<https://www.newson.co.jp/column/2024012301/>

・第 3 回：クラウド特性を活かし、柔軟性と拡張性を兼ね備えたアプリケーションを開発し安定稼働させるには

<https://www.newson.co.jp/column/2024053001/>

■ NTTデータ ニューソンについて

株式会社NTTデータ ニューソンは、情報システムの企画、設計、開発、保守をしています。1974 年に設立以来、システムインテグレータとしてソフトウェアとハードウェアの融合を行っております。2005 年にNTTデータと資本業務提携を行い、2017 年にNTTデータグループの一員となり、2024 年に創立 50 周年を迎えました。

NTTデータ ニューソンは、NTTデータグループの各社と連携し、“情報技術で、新しい「しくみ」や「価値」を創造し、より豊かで調和のとれた社会の実現に貢献すること”を目指しています。

NTTデータ ニューソンに関する詳細な情報については、<https://www.newson.co.jp/> をご覧ください。

＜ 本件に関するお問い合わせ先 ＞

株式会社NTTデータ ニューソン 営業推進室

E-mail : pr_newson@newson.co.jp