

クラウドサービスの概要

サービス

クラウドストラテジー

■ クラウド戦略策定

クラウドのサービスは多岐にわたり、その選択肢は膨大です。また、契約プランが複数あり、利用するワークロードに照らして最適解を見極めるのも至難の業です。特定のクラウドに絞って利用するだけでも困難な作業である上に、これがマルチクラウド／ハイブリッドクラウドとなると、途方も無い選択肢の前に呆然と立ち尽くすしかありません。

お客様にとって最適な選択肢は何なのか？

この問いに対して何のしがりみもなく方向性を示せるベンダーは、実はあまり多くありません。

当社では「クラウドを使わない」という選択肢を提示する場合があります。これは、クラウドを知らないからではなく、クラウドを深く理解しているからこそ言える言葉です。クラウドを使うことを目的化せず、お客様のアジリティを高めるための最適解が何なのか。本当の意味でのクラウド戦略策定の要諦はここにあります。

当社は、お客様が目指すビジネスゴールに向かって、クラウドをいかに使いこなすかという点にフォーカスし、戦略策定をご支援します。

クラウドインテグレーション

■ クラウドネイティブ開発

変化の激しいビジネス環境に迅速かつ柔軟に対応し続けていくためには、ビジネスニーズを支えるアプリケーションも絶えず進化させていくことが求められます。アプリケーション全体を一つの塊として管理するスタイルでは、大規模化・複雑化の傾向にあるビジネス環境に迅速に対応することはできません。

アプリケーションの構築と更改を迅速に行い、品質改善とリスク低減も同時に実現する開発アプローチ、それが「クラウドネイティブ開発」です。クラウドネイティブなアプリケーションのためのオープンソースソフトウェアの開発団体 Cloud Native Computing Foundation(CNCF)は、クラウドネイティブを実現するアプローチとして「コンテナ」、「マイクロサービス」といったキーワードを挙げています。

コンテナは、アプリケーションの仮想化技術の一つです。

コンテナは、CNCF による標準化が進められており、高可搬性・再利用性・柔軟性・自由度といったメリットがあります。しかしながら、その柔軟性・自由度の高さゆえ、運用管理の責任範囲が広がってしまうというデメリットも内在しています。この運用管理の責任範囲を排除するための仮想化技術として、サーバレスが登場しました。運用管理の省力化、インスタンスのオートスケーリングなどのメリットがある一方、コンテナと比較すると標準化が道半ばということもあり、ベンダー独自機能で実装する必要があるなどのデメリットもあります。

コンテナ、サーバレスの技術特性を理解し、ビジネスニーズに応じて適材適所に使い分けて適用することが、これらのクラウドネイティブのアプリケーション開発に求められています。

■ クラウドトランスフォーメーション（クラウドリフト、クラウドシフト）

ビジネスのデジタル化に取り組む企業は着実に増加しつつあり、デジタル化に未着手の企業は少数派となっています。一方、国内の 8 割の企業がレガシーシステムを抱えており、その存在がデジタルトランスフォーメーションの足かせとなっているとのレポートもあります。ここでレガシーシステムとは「業務プロセスを含むシステムの全貌と機能の意義がわからない状態」にあるシステムと定義されています。こうした企業がレガシーシステムから脱却できずにいる背景には、技術面の老朽化、システムの肥大化・複雑化、ブラックボックス化といった課題の解消が困難であることが挙げられます。

レガシーシステムのインフラストラクチャだけを刷新し、アプリケーションは現状維持したままクラウドへ移行しても、ビジネスのデジタル化は実現しません。アプリケーションのクラウドネイティブ化、開発・運用プロセスの効率化といった、先を見据えた戦略が重要です。一方で、ビッグバン的にレガシーシステムをクラウドネイティブ化する難易度は高く、段階を経て移行するクラウドジャーニーを明確にして実施していかなければ、目的であるデジタル化の足かせになりかねません。

レガシーシステムのインフラストラクチャだけを刷新する「リフト」、その後アプリケーションを少しずつクラウドネイティブ化していく「シフト」を組み合わせしていくことが、デジタル化の実現には必要不可欠です。

クラウドオペレーション

■ フルマネージドサービス

「マネージドサービス」とは、機能だけではなく、運用管理も一体となって提供されるサービスです。PaaS 系のサービスが該当します。

しかしながら、クラウドベンダーが提供するサービスには、マネージドサービスではない IaaS 系のものも多くあります。例えば Amazon EC2 ではコンピューティングリソースは提供されますが、OS へのセキュリティパッチの適用やストレージのバックアップ、障害の監視、障害発生時の回復作業といった運用管理は**利用者の責任で実施**しなければなりません。このようなサービスを利用する際には、セキュリティ、ガバナンス、コンプライアンスを確保するため、そのリソースの可視化と管理の強化が必要です。

当社は、クラウドベンダーが提供する監視機能や自動化機能などを駆使して、セキュリティ、ガバナンス、コンプライアンスの確保に向けた取り組みをご支援します。

■ DevOps

DevOps とは、ソフトウェア開発（software development）の Dev と IT 運用（information technology operations）の Ops の 2 つの用語を組み合わせた混成語であり、ソフトウェア開発手法の一つです。システム開発における開発ライフサイクルの短縮化を狙いとしています。

DevOps の目的である開発ライフサイクルの短縮化を実現するためには、プラクティスに基づく技術的な実践および文化的な実践が必要です。技術的な実践としては、コンテナ技術を利用したアプリケーションや開発環境の構成管理であったり、CodePipeline による CI/CD、CloudFormation による Infrastructure as Code (IaC) などが用いられます。文化的な実践としては、メルヴィン・コンウェイが提唱した概念であるコンウェイの法則（※システム設計は、組織構造を反映させたものになる）に基づく組織の設計およびマイクロサービスアーキテクチャの採用であったり、チームビルディングや「チームが自己組織化」するためのアプローチ（ワークショップ、ピアレビュー、レトロスペクティブなど）も必要となります。

当社では、文化的な実践、技術的な実践を通じて、デジタルトランスフォーメーションおよび DevOps の推進をご支援します。

クラウドリセール

■ クラウドリセール

通常 AWS の利用料支払いは米ドル建てのクレジットカード決済です。当社が代行することで、円建ての請求書払いでサービスをご利用いただけます。クラウドの採用を検討する上で、クレジットカード請求がネックになっているお客様は当社にご相談ください。

メリット



利用開始までのフロー

契約成立後、3 営業日以内を目途に AWS アカウントを発行いたします。

請求タイミング

ご利用月の AWS 利用料金を、**日本円に換算**^{※1} し、請求書でご請求いたします。翌月 15 日頃に**請求書**^{※2}を発行しますので、翌月末日までに当社が指定する口座に振り込みをお願いいたします。

※1 米ドルから日本円へは、三菱 UFJ リサーチ & コンサルティングの月中平均をもとに換算いたします。

※2 本サービスはサードパーティーの請求書生成機能を用いて請求書を生成します。そのため AWS コンソールで確認できる料金と差異が生じる点につき、予めご了承ください。

料金プラン

料金プラン	
初期費用	0円
月額費用	月額クラウド利用料＋代行手数料 (3000円または、クラウド利用料の10%のうち高い方) 価格は税抜き表記となります。

ソリューション

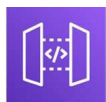
サーバレス

■ サーバレスアーキテクチャによるシステム構築

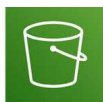
サーバのプロビジョニングや管理、パッチ適用は本来、専任の運用者を必要とします。システムを安定稼働する上で必要不可欠ですが、このことが企業の競争優位性確保や、ビジネス課題の解決といった本来注力すべき機能の開発に集中できない要因となることも多くあります。

そこで今注目されているのが**サーバレスアーキテクチャ**です。サーバレスは利用者側でサーバ管理や運用を必要とせず、アプリケーションやサービスを構築して実行することを指します。クラウド事業者が提供するマネージドサービスや、汎用的な API を組み合わせることにより、サーバレスアーキテクチャを実現します。

AWS 上では、AWS Lambda を中心として複数のサービスを組み合わせることにより、疎結合でスケーラブルなサーバレスアーキテクチャを実現できます。



Amazon API Gateway



AWS Simple Storage Service



Amazon Simple Queue Service



Amazon CloudWatch



AWS Lambda



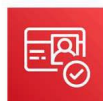
Amazon DynamoDB



Amazon Kinesis



AWS AppSync



Amazon Cognito

■ AWS Lambda の特長

AWS Lambda は AWS が提供するサーバーレスコンピューティングサービスです。Java、Go、PowerShell、Node.js、C#、Python、Ruby など多くの言語をネイティブでサポートしているため、開発者が自由に選択することができます。

- **インフラの管理が不要**

AWS Lambda では事前にサーバや OS をセットアップする必要がありません。パッチ適用をはじめとする保守作業や、サーバのパフォーマンスチューニング、セキュリティアップデートなどの管理は不要です。

- **オートスケーリング**

AWS Lambda ではアクセス数や負荷に応じて自動的にアプリケーションをスケールします。小規模なシステムから大規模なシステムまで利用可能です。

- **実際の処理量に対する従量課金制**

AWS Lambda では、リクエスト数と処理の実行時間に対して課金されます。さらに、AWS 無料利用枠により 1 か月あたり 100 万リクエストまで無期限無料枠で 사용할 ことができます。これによりサーバを所有する場合と比べ、劇的なコストダウンが見込める場合が多々あります。

■ 構成例

当社は多くの AWS 構築・移行・コンサルティングの実績が認められ、「AWS Premier Consulting Partner」ならびに「Well-Architected Partner」に認定されています。コンテナやサーバレスを用いたクラウドネイティブな開発ノウハウを活用することで、企業のデジタルトランスフォーメーションをサポートします。ご要望をもとに、ベストプラクティスに基づいた設計・構築を行います。

営業支援システムの構築

- **概要**

サーバレス基盤の営業支援ツール構築

- **導入の効果**

システム基盤を維持するための体制構築が不要となり、運用コスト・負担を大きく抑制され、業務アプリケーションの構築・改修に専念できるようになりました。



■ サーバレスの向き不向き

サーバレスアーキテクチャを採用するメリットは多々ありますが、システムを採用すればよいというものではありません。レイテンシ、処理時間、連携するサービスとの密接度を考慮した上で採用する必要があります。

AWS Lambda では、コードを実行する前にサーバへのコードを割り当て、コードのデプロイ、コード実行のためのランタイム起動といった処理が行われます。従って、低レイテンシを要求されるアプリケーションにおいては Provisioned Concurrency などを利用したコールドスタート対策が必要です。処理時間や割り当てるメモリにも制限があるため、一つ一つの処理が単純で軽量なものに利用することが適しています。

また、ステートレスな設計が前提となるため、適切な箇所に AWS Lambda を利用することで柔軟なアプリケーションの構築が可能となります。

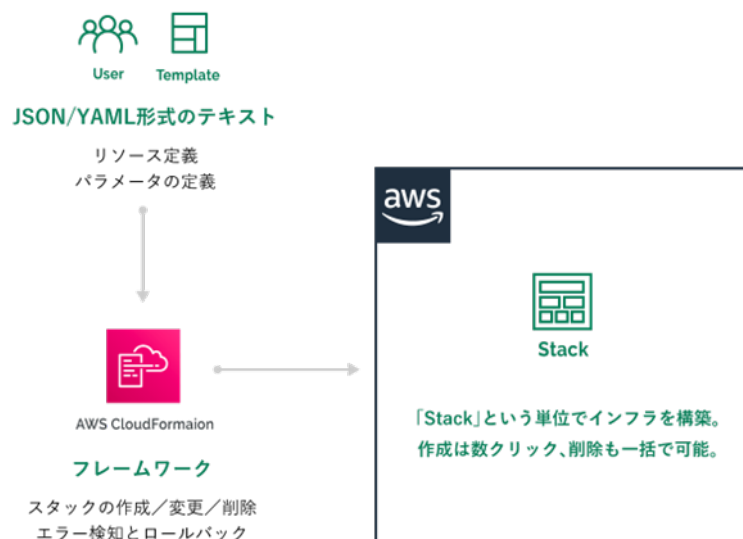
Infrastructure as Code

■ Infrastructure as Code の導入

Infrastructure as Code（以下、IaC）とは、「インフラをコード化して管理」することです。クラウドを利用することにより、システム構築にかかる時間が大幅に削減され、物理サーバは仮想サーバに置き換わり、スペックや台数、さらにアーキテクチャの変更など柔軟に対応できるようになりました。IaC の導入により、パラメータの設定ミスやインフラ更新時の設計書への反映漏れなどのヒューマンエラーの発生を抑え、インフラストラクチャの管理負荷を軽減することが可能です。

インフラストラクチャをコードにより管理する際に前提となる重要な要素は、「冪等性（べきとうせい）」を備えていることです。冪等性とは、IT インフラがどのような状態であっても、インフラ構築・運用のコードを実行したときに、常に同じシステム構成や運用になることを指します。

AWS では、CloudFormation というサービスが提供されており、開発者がアプリケーションコードを記述するように、インフラを開発、展開、維持できます。こうしたサービスが、回復力、明確性および信頼性を実現します。



■ AWS CloudFormation の特長

AWS CloudFormation は AWS において IaC (Infrastructure as Code)を実現する代表的な手段です。AWS におけるサービスのリソースの大部分は CloudFormation で構築・管理できます。

- **リソース全体のモデル化**

インフラストラクチャとアプリケーションリソース全体を、テキストファイルまたはプログラミング言語のいずれかでモデル化できます。組織としてインフラストラクチャコンポーネントの標準化や構成の一貫性を保つことが可能です。

- **リソースのプロビジョニング自動化**

手作業やカスタムスクリプト作成を必要とせずに、インフラストラクチャおよびアプリケーションの構築と再構築が可能になります。また、エラーが検出された場合は、変更が自動的にロールバックされます。

- **コード化**

インフラストラクチャをコード化することで、インフラのバージョン管理やコードの共有・再利用が可能になります。また、構築前にコードを事前検証することもできます。

■ 構成例

当社は多くの AWS 構築、移行、コンサルティングの実績が認められ、「AWS Premier Consulting Partner」ならびに「Well-Architected Partner」として認定されています。IaC を用いたインフラ構築の高速化およびセキュリティの確保を行い、労働生産性を向上することで企業のデジタルトランスフォーメーションをサポートします。また、ご要望をもとに、ベストプラクティスに基づいた設計・構築を行います。

寄付金管理システムの構築

- **概要**

C/S 構成からの AWS への全面移行とコンテナ／サーバレス基盤のシステム構築

- **導入後の効果**

導入後は CloudFormation による管理工数の削減と、統制の取れた戦略的情報発信が可能になりました。また、AWS へ移行し、サーバを冗長構成かつ Autoscaling にすることで、可用性が大幅に向上しました。さらに、AWS では Blue/Green デプロイを実現したことで、ダウンタイム無しでのサービス継続が可能となりました。

コンテナ

■コンテナ

コンテナ技術とは一つの OS 環境を共有し、コンテナと呼ばれる他のユーザから隔離されたアプリケーション実行環境を作り、あたかも独立したサーバのように使う技術を指します。必要最小限の CPU やメモリを使用するため、負荷が小さく高速な動作を実現できます。また、アプリケーションの実行に必要なライブラリ、ミドルウェア、設定ファイルをひとまとめにパッケージングするので、様々なプラットフォームで実行可能であるという特徴を持ちます。このように、コンテナはビジネスニーズに対処するための頻繁な変更にも対応し易い「アジリティ」を備えているため、デジタルトランスフォーメーションを実現する上で重要な技術要素です。

このコンテナ仮想化を実現するエンジンのデファクトスタンダードが Docker です。Docker ではコンテナを実行する時に必要な Docker イメージを Dockerfile と呼ばれるファイルで定義することができます。Docker はコンテナの持つ軽量さというメリットを活かすためにも、Dockerfile の最適化は重要です。蓄積された当社のノウハウをもとにアプリケーションのコンテナ化、および最適な実装をサポートします。

コンテナオーケストレーション

以前からあるコンテナ技術が今広く普及しつつある要因の一つとして、コンテナオーケストレーションの存在が挙げられます。コンテナオーケストレーションツールは多数のコンテナを複数のサーバで構成する際に必要となる、サーバ間のネットワークやストレージの管理、コンテナの配置、状態監視などの機能を備え、コンテナの安定的な運用をサポートします。

その中でもデファクトスタンダードとなりつつあるのが **Kubernetes** です。Kubernetes ではマニフェストファイルと呼ばれるファイルでリソースの具体的な設定情報を管理します。このマニフェストファイルを Kubernetes に登録することで、リソース（アプリケーションコンテナやネットワーク、ジョブなど）の望ましい状態を定義します。宣言した設定と実態に差分がある場合、自動修復する仕組みを備えています。

当社では Docker/Kubernetes（EKS、AKS）を活用したシステム設計・構築・運用の実績を有しています。Kubernetes やクラウドの短いスパンでのアップデートを随時キャッチアップし、それぞれが持つメリットを最大限に引き出すことで、拡張性と信頼性に優れたインフラ基盤を構築します。また、CI/CD や IaC といったほかの技術要素と組み合わせることでコンテナを中心とした迅速なアプリケーションの開発・リリースサイクルをサポートします。

マイグレーション

■クラウドマイグレーション（移行）

クラウドを利用して新たにシステムを構築することはそれほど難しいことはありません。初めからクラウドに最適化されたアーキテクチャを採用することで、クラウドのメリットであるアジリティやスケーラビリティ、イミュータビリティ、コストパフォーマンスを十二分に享受することができるからです。もちろん、そこには**クラウド技術に精通したコンサルタントやエンジニアの存在が不可欠**となります。

一方で、オンプレミスで稼働中のシステムをクラウドに移行することは容易なものではありません。長年の蓄積によりサーバの台数は増大し、場合によっては数百台、数千台というケースも珍しくありません。台数の増加に伴い、システム間の連携は複雑化し、誰も全容を把握できていないということも考えられます。また、アーキテクチャも老朽化して、脆弱なままのソフトウェアを使い続けていたり、EOL（End of Life）を迎えているものもあるかもしれません。

また、稼働中であるということは、システムによっては 24 時間 365 日サービスを提供し続けなければならないものもあり、そういった事情もクラウド移行を難しくする要因となっています。**このように肥大化、複雑化、老朽化したオンプレミスシステムをクラウドに移行するには適切なプロセスを踏むことが必要不可欠となります。**さらに、クラウド技術だけではなく、レガシーアプリケーションやインフラに精通したエンジニアの存在も必要となります。

■ 4つのプロセス

当社では、以下のようにアセスメント、プランニング、マイグレーション、オペレーションの 4 つのプロセスを経ることで、クラウドマイグレーションを成功に導くためのご支援いたします。

● アセスメント

アセスメントでは既存のシステム・業務を分析し、評価します。単にインフラのリソースやソフトウェアを分析するだけではなく、アプリケーションコードやデータ、さらにはビジネスプロセスに至るまでをワンストップで分析・評価します。長年のアプリケーション開発やコンサルティングの実績に加え、最先端の技術をもつアーキテクトやクラウドエンジニア、熟練したインフラエンジニアを有する当社にしかできないアセスメントを提供します。

● プランニング

アセスメントの結果に基づき、移行計画を立てます。後述する移行パターンを的確に見極め、移行戦略を策定します。さらに、移行時および移行後に発生する運用コストの試算を行い、クラウド移行が本当にお客様のビジネスのデジタル化にとって有益か否かといった視点も含めた最適な選択肢をご提案いたします。

● マイグレーション

プランニングに基づき実際の移行作業を行います。クラウドベンダーが提供している移行ツールやマネージドサービスをフル活用し、シームレスなマイグレーションを目指します。

● オペレーション

クラウド移行後の運用フェーズとなります。DevOps の考えに基づく、技術的、文化的な実践によりビジネスの加速を支援します。また、移行時にクラウド最適化できなかったシステムのクラウド最適化を行います。導入から運用までをワンストップでご支援する当社ならではの価値を提供いたします。

■ マイグレーションパターン

クラウドマイグレーションを成功させるには、アセスメントとプランニングのプロセスにおいて、**マイグレーションストラテジー（移行戦略）**を適切に策定することが重要です。そのためには、システムやサーバの特性を理解し、それに応じた適切な移行パターンを見極めることが必要となります。

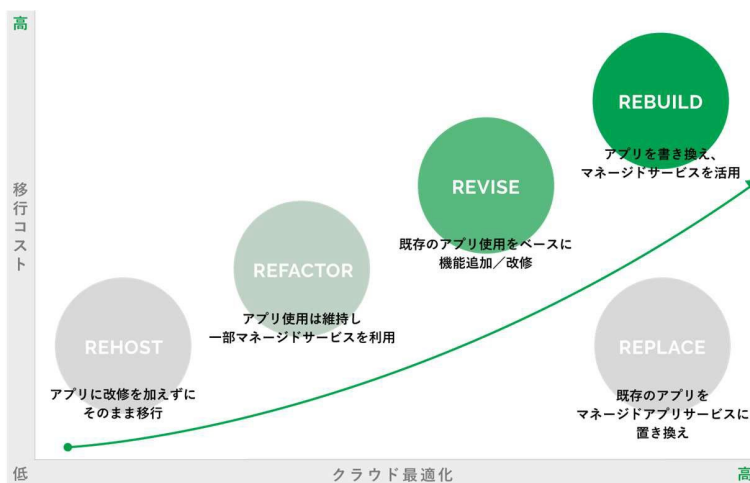
移行パターンについてはクラウドベンダーが各社各様の用語を定義していますが、いずれにおいてもベースとなっているのは米 Gartner 社が 2011 年に定義した「**5 つの R**」と呼ばれるものです。

クラウド最適化度	パターン	意味	概要
1(低)	Rehost	単純移行	アプリに改修を加えずにそのまま移行
2	Refactor	一部置き換え	アプリ使用は維持し、一部マネージドサービスを利用
3	Revise	一部変更	既存のアプリ使用をベースに機能追加／改修
4	Rebuild	作り変え	アプリを書き換え、マネージドサービスを活用
5(高)	Replace	置き換え	既存のアプリをマネージドアプリサービスに置き換え

■ マイグレーションコスト

クラウドマイグレーションを行うにあたり、移行時および移行後に発生する運用コストを試算することは重要です。一般的に、リプレイスが可能な一部のアプリケーションを除けば、アーキテクチャのクラウド最適化の度合いが高まるにつれて、移行後の運用コストは削減されます。その反面、移行時にはアプリケーションの改修にかかるコストは高くなる傾向にあります。

当社では、移行後の運用コストまでを考えた戦略を策定することで、単なるインフラの載せ替えにとどまらない、ビジネスのデジタル化を推進するためのマイグレーションを目指します。



クラウド最適化 ≡ 運用コスト減

ビッグデータ分析

■ クラウドでのデータ活用

近年では大量のデータから知見を見出す「機械学習」が盛んになっています。機械学習を実現する上で欠かせないビッグデータ分析が注目を集めており、企業の利益を左右するビジネスの意思決定に欠かせなくなっています。

インターネットやスマートフォンの普及、IoT 技術の発展により収集・分析するデータ量や種類が大幅に増えています。こうしたデータを人の経験や勘を頼りに分析を行うのは時代遅れとなりました。機械学習を駆使してビッグデータを活用することで精度の高い分析を行うことが可能になり、経営者や責任者がより高い利益を得るための判断を下しやすくなります。

技術の進歩により、ビッグデータを扱うためのデータ基盤のトレンドの変化は加速し続けています。

デジタル化という社会的なニーズが増し、これまで以上にスピーディに柔軟に、多様なデータを扱うことが要求されます。また、夜間バッチで処理し翌日に分析結果を確認するという時代は終わり、蓄積したデータをリアルタイムで参照・分析することが求められています。クラウドという技術的イノベーションの登場により、この要求に応えるインフラストラクチャを提供することが可能になりました。

オンプレミスでデータ分析基盤を構築することも可能ですが、あらゆるフォーマットのデータを蓄積するためには「NoSQL」、「分散 KVS」、「ドキュメントストア」などの様々なデータストアを用意する必要があります。データストアの OSS としては「Apache Hadoop」、「Apache Cassandra」、「MongoDB」が有名ですが、これらの OSS を使って大規模なデータ分析基盤を安定稼働させるには、相応のスキルを持つ技術者を確保する必要があり、構築・運用のハードルが高くなります。

さらに、サーバの大規模なクラスターが必要となり、プロビジョニングやセットアップにかかる時間と負荷も大きくなります。しかし、クラウドでは容易かつ素早くこうした環境をデプロイできます。

CI/CD

■ CI/CD

CI/CD とは、Continuous Integration / Continuous Delivery の略であり、日本語では、「継続的インテグレーション」および「継続的デリバリー」と言います。近年、リリース頻度を増やしたい企業の要求から、開発からリリースまでの速度向上を目的として生まれた方式です。

- **継続的インテグレーションとは**

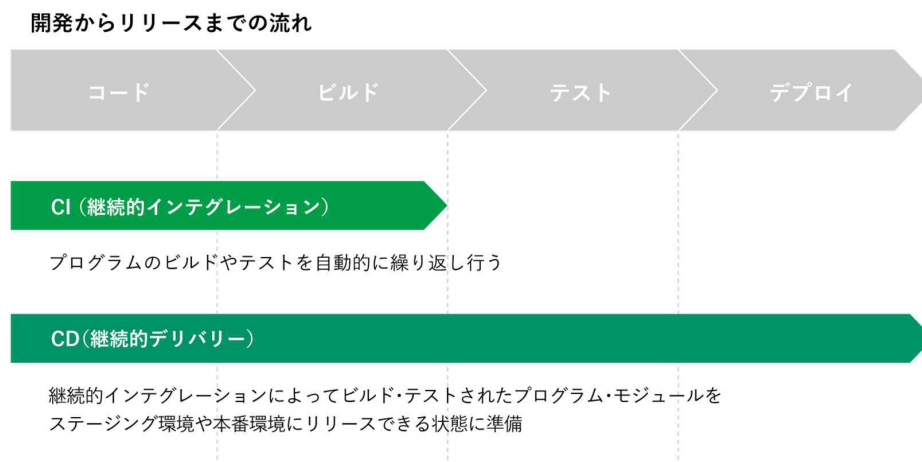
継続的インテグレーションとは、プログラムのビルドやテストを自動的に繰り返し行うことを指します。継続的インテグレーションを行うことによって、プログラムの品質向上および開発・テストの高速化を図ることができます。

- **継続的デリバリーとは**

継続的デリバリーとは、継続的インテグレーションによってビルド・テストされたプログラム・モジュールをステージング環境や本番環境にリリースできる状態に準備することを指します。継続的デリバリーを行うことによって、開発者が個別に各環境にリリースするといったことが不要となり安全なリリースが可能となるため、開発生産性の向上やデプロイ作業自体の高速化を図ることができます。

ビルド・テスト・デプロイを自動的に行うためには、CI/CD のツールやサービスを導入する必要があります。クラウド毎に用意されている CI/CD サービスとして AWS CodePipeline、Azure Pipeline があります。また、Jenkins、Travis CI、CircleCI といった特定のクラウドベンダーに依存しないソフトウェア・サービスで CI/CD パイプラインを構成することも可能です。

実導入にあたっては、ハイブリッドクラウドやマルチクラウドなど将来を見据えた構成を考えた上で CI/CD の方式を検討する必要があります。





セキュリティ

■ セキュリティ

情報システムにおいてセキュリティは最重要テーマであり、取り扱うデータ、システム、資産を保護するための施策が欠かせません。情報セキュリティの基本は、データ、システム、資産にアクセス可能であるユーザや機能・サービスを特定し、それらの認証・認可を適切に行うことによる「機密性の確保」、改ざん防止やデータの破壊からの復旧（バックアップなど）を実現する「完全性の確保」、情報アクセスを停止させないための「可用性の確保」です。

クラウドサービスは、これらの認証・認可、バックアップ、冗長化などの技術がクラウド基盤として提供されています。オンプレミスではサーバ OS やソフトウェアを組み合わせる必要がありましたが、クラウドではこれらがすぐに使えます。

境界防御はクラウドにおいても重要です。ネットワーク機器で実現されていたセキュリティ防御は、クラウドにおいてはソフトウェアで実現されます。昨今は、境界防御では保護しきれないセキュリティ攻撃への対応が必須であり、境界防御のみでは機密性を担保できません。しかし、境界防御はシステムにおける基本的なセキュリティ技術としてクラウドにおいても留意することが必要です。

高度標的型攻撃や内部の人員による犯行の観点から、現時点では完全な防御を実現することは難しいため、情報システムはセキュリティインシデントを早期に検知し、対策を打てる仕組みになっていることが重要です。そのため、脅威インテリジェンスやログの解析による不正検知（SIEM）といった技術が注目されています。また、内部の人員による犯行を防止するためにも、運用時の設定変更やオペレーションが記録、監視されていることが重要です。

パブリッククラウドではこれらの記録、監視を実現するための機能を有しているため、高いレベルの情報セキュリティを確保することができます。

「クラウドはセキュリティが心配」という声を耳にすることがあります。しかし、クラウドはオンプレミスでも取得が難しいとされる各種の第三者認証をパスしています。これはクラウドの基盤費用を利用者全員で分担しているため、高いレベルのセキュリティを全員が享受できていることを意味します。

また、多くのパブリッククラウドでは、HIPAA や HITECH といった機微な情報を取り扱うための基準や、ISO/IEC27017、CS ゴールドマークなどといった第三者認証を取得しています。



About Capgemini

キャップジェミニは、お客様のビジネストラansフォーメーションならびにテクノロジーtransフォーメーションのグローバルパートナーです。企業が“デジタル世界”と“持続可能な世界”への移行を加速できるようご支援し、組織と社会へのインパクトを目に見える形で生み出しています。当グループは、世界 50 か国以上、約 34 万人の使命感あふれる多様性に富んだチームメンバーから成る組織です。55 年にわたり培ってきた豊富な実績を有し、幅広いビジネスニーズへの対応をテクノロジーの力で支援するパートナーとして、お客様から厚いご信頼をいただいています。AI、クラウド、データ領域で市場をけん引する能力を発揮し、各業界への深い専門知識とパートナーエコシステムを組み合わせ、戦略・設計からエンジニアリングまでを網羅する強みを活かして、エンドツーエンドのサービスやソリューションをご提供しています。2023 年のグループ売上高は 225 億ユーロです。

Get the Future You Want - 望む未来を手に入れよう | www.capgemini.com



Copyright © 2024 Capgemini. All rights reserved.