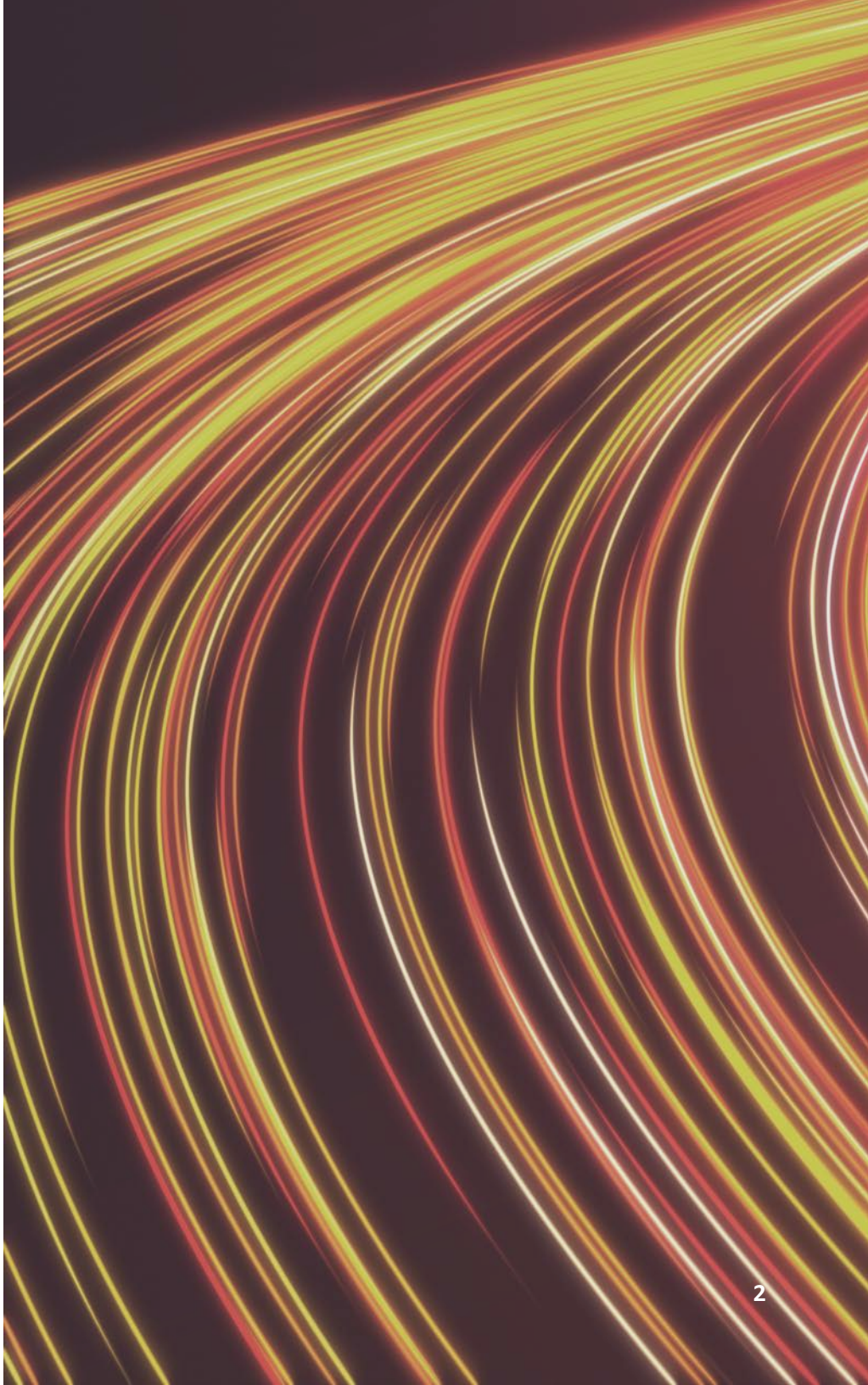

aws modern apps
モダン Dev+Ops
モデルの導入

モダン Dev+Ops を導入すると、開発 (Dev) と運用 (Ops) の役割間のギャップを埋めて、顧客の進化するニーズにより速く、より一貫して対応できます。

DevOps は世界を変えました。箱入りのディスクとして店頭で販売されていたソフトウェアは、Amazon とインターネットの登場により AWS で構築されたサービスとしてデジタルで配信されるようになりました。その中で、数か月～数年だったソフトウェアの更新サイクルを数日へと短縮し、Dev と Ops の関係を緊密にしたのは DevOps でした。

すべての企業は、形態と規模を問わず、ウォーターフォール型の製品管理サイクルを放棄し、アジャイル方式と DevOps を導入することで、イノベーションの加速、セキュリティ、パフォーマンス、レジリエンスの向上、デベロッパーと顧客の満足度の向上を確実に実現しています。ただし、DevOps のすべての利点を実現しても、まだ完全ではありません。DevOps は、クラウド内でのアプリケーションの開発およびホスティングが標準の手法となる前に誕生し、既に長く知られていますが、その解釈は依然として組織ごとに大きく異なります。例えば、DevOps を依然として専門チームと見なす組織が多い一方で、デベロッパーがすべての運用業務を担当することと解釈されることもあります。テクノロジーの進化に伴って、DevOps の定義も進化しています。AWS では、この手法をモダン Dev+Ops と呼び、コンプライアンス、可観測性、レジリエンス、インフラストラクチャなどの運用業務を開発プロセスの早い段階で共有することで開発チームと運用チームの関係を緊密化し、さらに AI と機械学習で強化することを、この手法の主眼としています。

aws modern apps



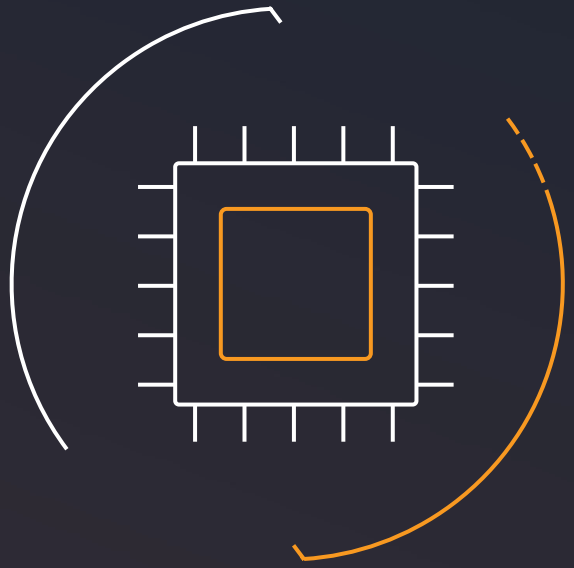
State of DevOps Report によると、DevOps を導入してデプロイ頻度を週 / 月単位から時間 / 日単位に移行した企業では、リードタイムが数か月から数日に短縮され、変更失敗率が 46 ～ 60% から 0 ～ 15% に減少しています。多くの顧客にとって、クラウドでの成功とは、オンデマンドのスケーラブルなインフラストラクチャが導入されたかどうかだけではなく、開発と運用のプラクティスがどのように変革されるかが少なくとも評価の一部となります。現在、Coca-Cola Argentina、3M、Lululemon Athletica、ワシントンポストなどのお客様は、モダン Dev+Ops アプローチに移行することで、AWS の Dev+Ops サービスを活用して開発チームと運用チームの関係を緊密化し、DevOps の利点を一層享受しています。

「AWS を活用することで、デプロイが 6 週間から週 1 回に短縮されました。間もなく 1 日に複数回のデプロイが可能になるでしょう」[1]

— 3M Health Information Systems、アドバンストテクノロジー部門マネージャー、Rick Austin 氏

「AWS CloudFormation テンプレートと AWS CodePipeline を使用すると、新しい本番稼働用アカウントの作成時間を 2 日から数分に短縮できます。そのため、コストもセットアップ時間もほとんどかからない小規模プロジェクトを立ち上げることができます。この俊敏性により、手持ちのリソースで対応できる範囲に留まることなく、最適なソリューションを試して導入できます。AWS を活用すると、新しい機能やアプリケーションを以前よりもはるかに迅速にリリースできます」[2]

– Lululemon Atheletica、プロダクトアーキテクチャ部門ディレクター、Sam Keen 氏



[1] *https://aws.amazon.com/solutions/case-studies/3M-health-information-systems/*

[2] *https://aws.amazon.com/solutions/case-studies/lululemon-athletica/*

DevOps のプラクティスがより高度に進化したチーム

77%

問題の発生後 1 日未満でサービスを復元

60%

セキュリティの脆弱性を 1 日未満で完全に修復

3 倍

変更管理の効率性

aws modern apps

多くのお客様は、Dev+Ops をまだ導入していませんが、より迅速でエラーが少ない配信を容易にするソリューションを求める過程で導入を検討しています。一部のお客様は Dev+Ops を既に導入していますが、期待したレベルのスピードと成功を実現できず苦労しています。多くのお客様にとって導入が難し過ぎるのです。

1. 取り掛かるのが、非常に困難です。Dev+Ops は非常に手ごわい問題に思えるでしょう。学ぶことが多く、変更を実装するために多くの作業を伴います。
2. 現在利用できる業種別ソリューションは、セキュリティ、コンプライアンス、高可用性、アクセス制御などの企業ニーズを十分に考慮していません。その結果、DevOps チームが多大な時間を費やして構築したソリューションを後で置き換える必要があります。
3. 多くの業種別ソリューションは不十分であり、DevOps チームが Dev+Ops に容易に取り組めるレベルに至っていません。例えば、多くのソリューションはリアクティブで、プロアクティブではありません。お客様に通知されるのは障害が既に発生した後であり、事前に警告したり、最適な進め方を提案したりして問題を未然に防止するまでには至っていません。

モダン Dev+Ops の最も難しい部分は、文化を変えることです。全員の認識を統一するには、リーダーシップ、時間、コミットメントを必要とします。デベロッパーが、従来、運用チームが受け持っていた責任を持ち始める場合、共有責任の意識が必要になります。以下に示す方法に従って、文化と行動の両面から作業を進めることで、パフォーマンスの高い Dev+Ops 組織を構築する仕組みが実現します。

この日本語ガイドでは、高いパフォーマンスを実現している Dev+Ops 組織を通じて AWS が特定した、指針となる原則を紹介します。

- 説明責任を習慣づける
- 現状を評価する
- 段階的に改善する
- すべてを自動化する
- 可能な限りコード化する
- 二重対策を行う
- ツールを標準化する

説明責任を習慣づける

Dev+Ops の目標はソフトウェア配信の迅速化ではありますが、当初の意図は開発と運用を緊密化させて、共同して問題を解決することにありました。時間の経過と共に組織の構造が変わり、説明責任の文化を重視することが共通のソリューションになっています。チームや個人にはそれぞれ異なる専門技能と専門分野があるかもしれませんが、問題に取り組んで解決する際に説明責任を共有する必要があります。

現状を評価する

Dev+Ops の最終的な目標は迅速化にあります。ただし、チームはどこから開始するかをわかっていない場合があります。現状を評価することから開始できます。これは、現時点でのリリースまでの時間を評価するということです。ソフトウェア配信プロセスの中で、最も長い時間がかかっている部分とその理由は何でしょうか。改善できるでしょうか。成功の鍵となる他のパフォーマンスメトリクスは何でしょうか。開発 (Dev) と運用 (Ops) の間でサイロ化されているプロセスの領域があるでしょうか。これらのメトリクスの入力と結果を評価することから開始すると、チーム内の信頼を築くような結果を出すために取り組むべき重点領域が見つかります。

- リリースまでの時間
- パイプライン内のボトルネックまたはブロッカー
- リリース速度

段階的に改善する

パフォーマンスの高い Dev+Ops 組織を数日や数週間でゼロから構築することはできません。いくつかの指針となる原則を見極め、小さなことから始めて、着実かつ段階的な改善を積み重ねることが重要です。同様に、大きな変更や大きなリリースにこだわっていると、ソフトウェアプロセスを迅速化することはできません。大規模で複雑な変更に取り組むよりも、小規模な変更を徐々に進めながら、新しいことを少しずつ学ぶほうがはるかに簡単です。

すべてを自動化する

Dev+Ops の目標は、すべてを滞りなく連続して動作させることにあります。自動化により、人的介入に伴う時間とエラーが減ります。Dev+Ops ワークフローに自動化を導入する方法は多くありますが、最も基本的な方法は Dev+Ops CI/CD パイプラインに自動のテスト、デプロイ、ロールバックを追加することです。

可能な限りコード化する

自動化と Dev+Ops を成功させるもう 1 つの要因は、インフラストラクチャやポリシーなどをコード化し、人間と機械の両方が理解できるようにすることです。最も基本的なユースケースは、Infrastructure as Code (IaC) です。これにより、人間が定義したクラウドインフラストラクチャのプロビジョニングが自動化されます。これを手本として、変更を追跡し、組織全体でベストプラクティスを共有できます。別の例は Policy as Code です。この場合、中心となるチームはルールを記述し、自動化によりそのルールが適用されます。

Infrastracture as Code ファイルには、以下が含まれます。

- コンピューティング
- ストレージ
- アイデンティティ、アクセス、セキュリティ
- アプリケーションリソース

二重対策を行う

ベストプラクティスを確実に実行させるには、奨励と強制の両方が必要です。ベストプラクティスを奨励するには、通常、コードとしてのインフラストラクチャ (Infrastructure as Code) を使用し、組織全体でベストプラクティステンプレートを共有します。ベストプラクティスを強制するには、コードとしてのポリシー (Policy as Code) などのプロセスを使用できます。ただし、二重対策をさらに進めるには、ルールと AI を使用してユーザーに影響を与える前に潜在的なエラーを見つけて、ベストプラクティスの実装方法について Dev+Ops チームにガイダンスを提供します。

ツールを標準化する

ツールを標準化すると、組織はスケールできます。ツールセットが異なると、ポリシーを記述したりベストプラクティスを適用したりすることが、不可能ではないものの非常に困難になります。いくつかのベンダーのツールが相互に連携するように設計されている場合は、それらを使用できますが、最終的には標準化することでこそ、時間が節約され、エラーが削減されます。

AWS が選ばれる理由

AWS は、モダン Dev+Ops のプラクティス、ツール、トレーニングのコアセットを提供することにより、お客様がクラウド変革を迅速化するための支援をする立場にあります。お客様は、AWS でクラウドの利点をいち早く活用し、イノベーションを加速させることができます。

1. **Amazon は Dev+Ops とイノベーションのパイオニアである**

   Amazon は、Dev+Ops の一般的なベストプラクティスの多くを先駆けて確立し、パフォーマンスに優れたお客様との 10 年を超える連携を通じて、お客様のニーズに合うソリューションを提供しています。AWS のサービスには、Amazon のベストプラクティスがデフォルトで組み込まれています。

2. **AWS には、最も広範で最も奥の深い、統合されたモダン Dev+Ops サービスのポートフォリオがある**

   コアモニタリングや CI/CD サービスから、コードとしてのインフラストラクチャ、さらにはビッグデータや AI を使用してプロアクティブなアドバイスやレコメンデーションを提供する最新のサービスまで、AWS はパフォーマンスに優れた Dev+Ops チームもしくは組織を構築するために必要なすべてのサービスを備えています。

3. **セキュリティはジョブ 0 (最優先)、可用性はジョブ 1 (その次に優先) である**

   AWS の Dev+Ops サービスには、セキュリティのベストプラクティスがデフォルトで組み込まれています。例えば、CI/CD パイプラインはデフォルトでプロジェクト別に分離されるため、アクセスを許可する対象を該当するユーザーだけに簡単に制限できます。さらに、AWS のすべてのサービスにおいて高可用性が実現されているため、チームの生産性が高まります。

4. **デベロッパーと組織の両方向けに構築されている**

   多くの開発ツールは、組織のニーズを十分に考慮していません。AWS は、デベロッパーと組織の両方のニーズを重視し、ガバナンスとコンプライアンスのコントロールを容易に実装して、開発チームが迅速かつ安全に業務を実行できるようにします。

アクションプラン

モダン Dev+Ops を AWS で実現するには、どうしたらよいでしょうか。以上の指針となる原則を実践するために、すべての形態と規模の組織が使用している、推奨されるベストプラクティスを以下に示します。

GitOps

GitOpsは、コードとしてのインフラストラクチャを Git リポジトリでホストし、アプリケーションソフトウェアコードと同じマージリクエストプロセスに従うプローチです。GitOps は、アプリケーションのインフラストラクチャへの帯域外の変更を排除するように設計されています。コードとしてのインフラストラクチャは、アプリケーションのインフラストラクチャをコード形式で（通常、テンプレートやファイルに）定義し、そのファイル定義に従ってインフラストラクチャのプロビジョニングを自動化する手法です。GitOps では、インフラストラクチャの定義ファイルが Git リポジトリでホストされ、各コミットはアプリケーションソフトウェアコードと似たマージリクエストプロセスに従い、継続的インテグレーションと継続的デリバリー (CI/CD) 自動化を通じて処理されます。GitOps では、インフラストラクチャファイルがテストされ、インフラストラクチャのプロビジョニングが自動化されます。エラーが発生した場合は、前の状態にロールバックされます。

AWS は、AWS CloudFormation に基づいて構築された完全な GitOps ソリューションを提供しています。AWS CloudFormation を使用すると、開発チームと運用チームは、YAML や JSON などの一般的なインフラストラクチャ言語を使用したり、AWS Cloud Development Kit (AWS CDK) を使用したりして、コンソールでInfrastructure as Code ファイルを簡単に定義、プロビジョン、および管理できます。AWS CDK を使用した場合、デベロッパーは、Python、TypeScript、C#、Java、JavaScript などの一般的なプログラミング言語でCloudFormationテンプレートを定義できます。CloudFormation は、自動化された CI/CD ワークフローを実行し、AWS CodeCommit、GitLab、Bitbucket、GitHub など、人気の高いバージョン管理サービスとも統合されていて、AWS CodePipeline と非常によく統合されています。新しいインフラストラクチャ定義ファイルが作成されると、このファイルはバージョン管理された環境でホストされます。CloudFormation テンプレートが変更されると、CI/CD 自動化がトリガーされ、デベロッパーは簡単に変更をデプロイまたはロールバック / フォワードできます。

「Rivian では急成長に伴い、非常にスケーラブルなシステムが必要になっています。5 日間かかっていた変更が今は数分以内で終わります」[3]

— Rivian、3DExperience テクニカルリード、Surendra Balu 氏

[3] *https://aws.amazon.com/solutions/case-studies/rivian-case-study/*

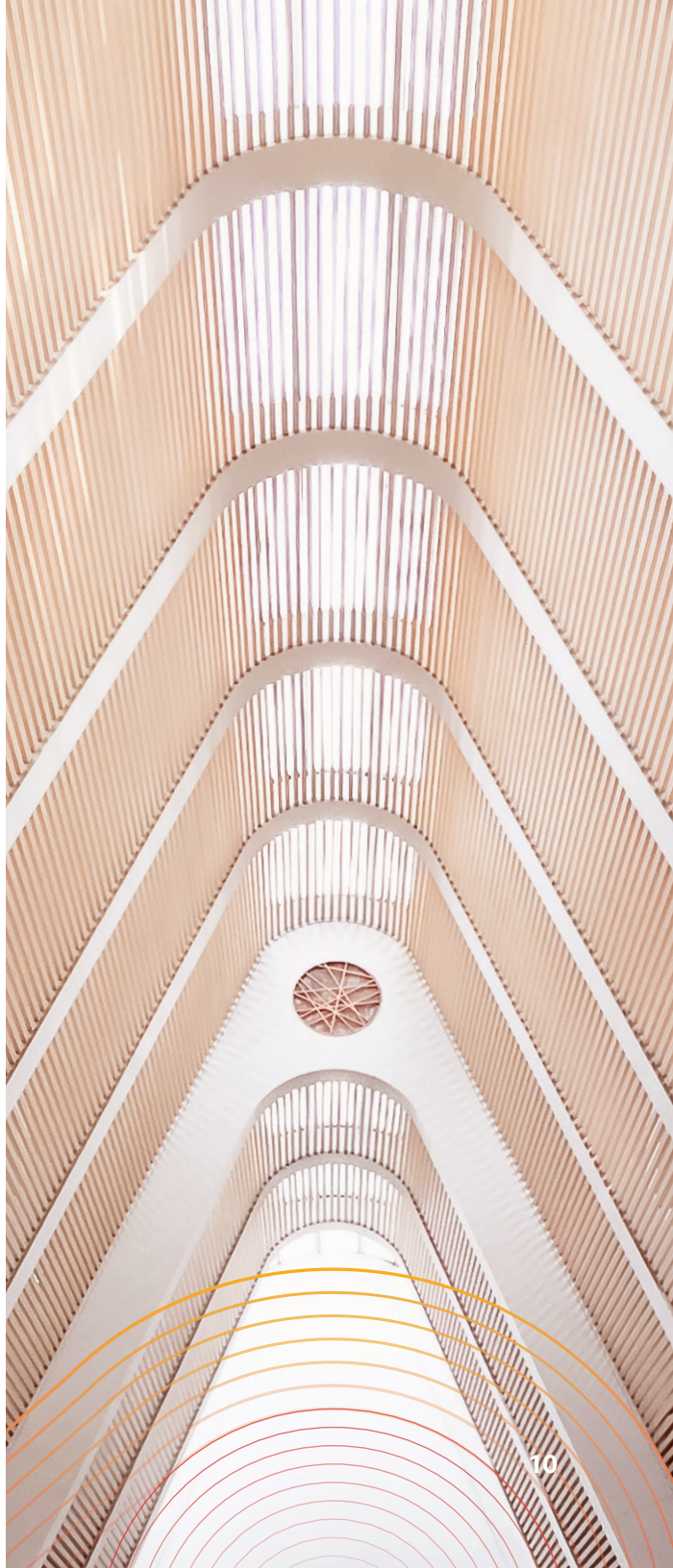
継続的インテグレーションと継続的デリバリー (CI/CD)

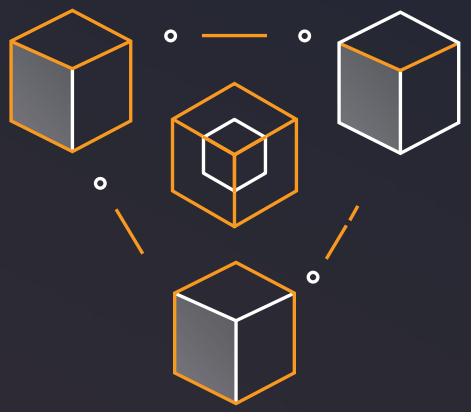
CI/CD は、ビルド、テスト、デプロイの各パイプラインを使用してソフトウェアのリリース準備を自動化するための基礎的なベストプラクティスです。CI/CD は、チームの作業を迅速化して、エラーが発生しやすい手動プロセスと、ソフトウェアリリースに伴う管理作業を排除するのに役立ちます。顧客のリリース期間が長い場合は、CI/CD でシステムを継続的に変更することで、この問題を解決できます。エンジニアは、コード内の潜在的なエラーや違反を検出するテストを記述し、検出された場合は更新をチームに差し戻します。バグが本稼働環境まで見逃された場合でも、チームは安定したバージョンへのロールバックを自動化してアップタイムを維持できます。

AWS は、エンタープライズまたはスタートアップ企業の CI/CD を構築するためのフルマネージドソリューションを提供しています。AWS CodeBuild、AWS CodeArtifact、AWS CodePipeline、および AWS CodeDeploy は、お客様の CI/CD の実装を支援するために構築されたコネクテッド Dev+Ops サービスのセットです。これらのサービスは、Amazon の内部 CI/CD ツールの利点を反映し、デフォルトでベストプラクティスを奨励します。例えば、セキュリティを重視するお客様は、各プロジェクトを単一の CodePipeline パイプラインに分離することで、アクセスを簡単に制限し、該当するユーザーにのみプロジェクトへのアクセスを許可できます。共有ビルドサーバーの脆弱性について心配する必要はありません。さらに、AWS では CI/CD が容易であり、専門知識も少なくて済みます。Amazon CodeGuru Reviewer などのサービスをソフトウェアリリースサイクルのプルリクエストフェーズで利用すると、機械学習を使用したコードレビューにより、開発中の重大な問題、セキュリティの脆弱性、見つけにくいバグを特定できます。デベロッパーは、提供されるガイダンスに従って、新しい機能を本稼働環境にデプロイする前に、問題を解決してコードを改善できます。また、エンジニアリングチームは AWS Proton をサービスプラットフォームとして使用し、開発チームのインフラストラクチャのプロビジョニング、コードのデプロイ、モニタリング、および更新に必要なさまざまなツールを事前に設定できます。

「AWS CodeBuild を使用すると、アプリケーションのビルドは約 10 分で完了します。Jenkins を使用していたときは最大 1 時間かかっていました。同じパフォーマンスを Jenkins で達成しようとすれば、同じ時間でビルドを完了するのに 50 個の Jenkins インスタンスをスピンアップする必要があるため、4 倍のコストがかかるでしょう」

— Coursera、シニアエンジニア、Bryan Kane 氏





AIOps

AIOps は、自動化とプロアクティブなメカニズムを促進し、チームが自信を持ってイノベーションを加速できるようにする変革です。AIOps は、デベロッパーに要求される知識の量を減らし、Dev+Ops ワークフローで AI を活用してデベロッパーの経験を補強するように設計されています。また、問題が発生する前に、役立つインサイトを提供してチームがプロアクティブに対処できるようにし、デフォルトでベストプラクティスを適用して、結果としてイノベーションを加速させます。例えば、デベロッパーは、ベストプラクティスからの逸脱、同時実行の問題、その他の一般的なコーディングバグを、システムに影響を及ぼす前に検出し、実践的なレコメンデーションに従ってシステムのパフォーマンスを向上させることができます。

Amazon CodeGuru Reviewer は機械学習を利用したデベロッパーツールサービスであり、Amazon.com と AWS オペレーショナルエクセレンスで 20 年間にわたって蓄積された情報に基づく AWS 機械学習モデルに従って、コードの分析を容易にし、セキュリティのベストプラクティスがすべての場所で実践されるようにします。

Amazon CodeGuru Profiler を使用すると、お客様はアプリケーションの実行に伴う高コストで非効率な方法を特定し、これらを修正するための実践的なステップを導入して、コストを削減できます。また、Amazon DevOps Guru を使用すると、AIOps ソリューションを導入してアプリケーションの運用パフォーマンスと可用性を向上させることができます。DevOps Guru は、アプリケーションの異常な動作を特定し、停止やサービス中断を引き起こす可能性がある重大な問題を浮上させて、問題を修正するためのレコメンデーションを提供します。

「私たちは運用上の問題の解決にチームが費やす時間を減らす方法を常に模索しています。Amazon DevOps Guru の利用を開始したことで、機械学習から得られたインサイトを活用して運用上の問題をすばやく特定し、相互に関連付けて、修正できるようになりました。Amazon DevOps Guru がもたらすインサイトにより、チームは最初からやり直すことなく、問題をすばやく検出し、根本原因を究明することができるようになりました。IT チームは平均復旧時間 (MTTR) を大幅に短縮し、問題解決の時間を削減しています。一方で、顧客のエンドユーザーエクスペリエンスは最高レベルに維持されています」

— HCL Technologies

継続的な可観測性

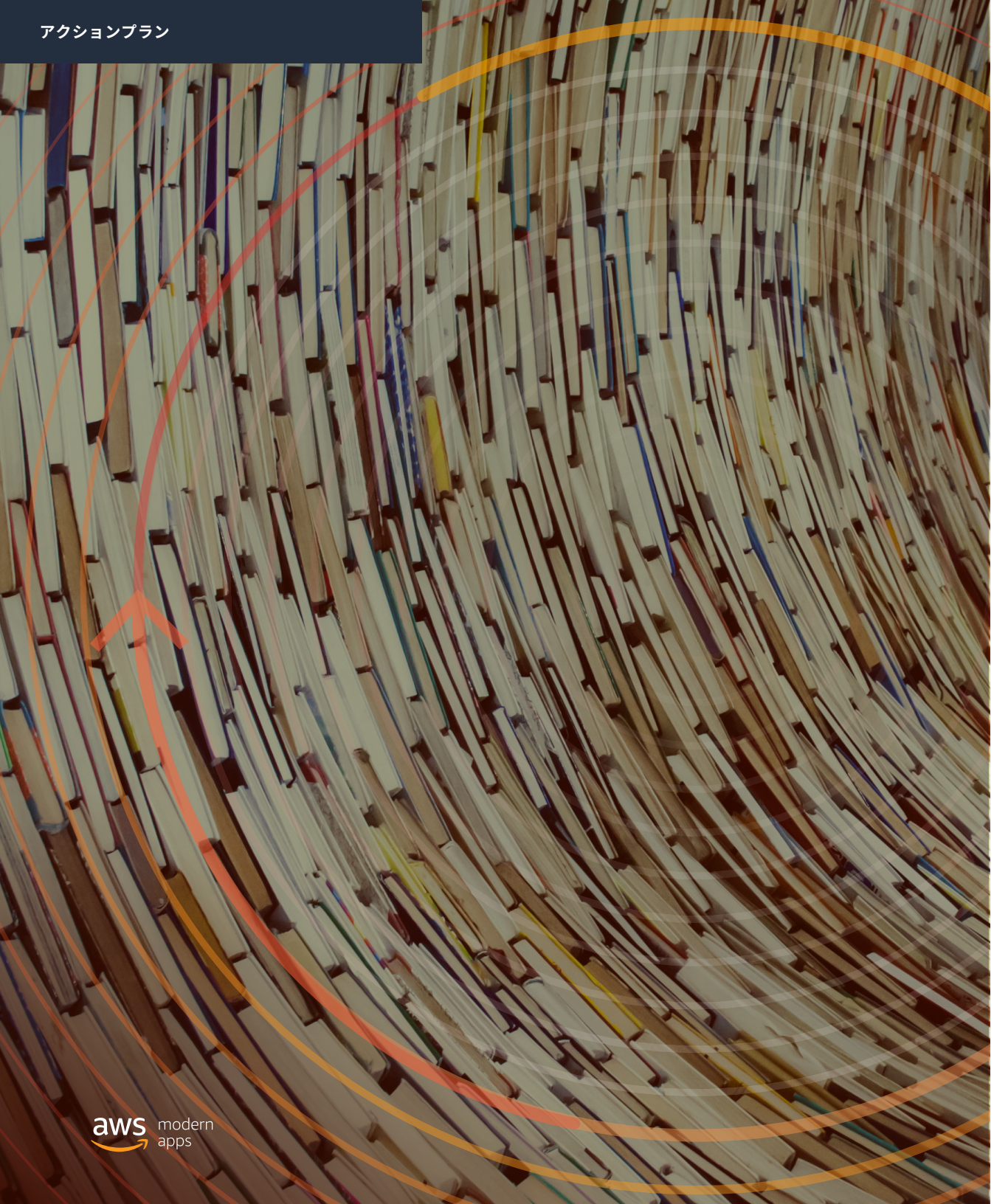
継続的な可観測性があることで、システムのステータスをいつでも確認することが可能になります。可観測性は、問題を検出、調査、修正する能力です。エンジニアリングチームは、継続的な可観測性を利用してアプリケーションのヘルスとパフォーマンスに関するインサイトを取得し、注意を要する問題のトラブルシューティングと解決を行い、最終的にエンドユーザーのエクスペリエンスを改善できます。クラウドのアプリケーションとリソースは、膨大なデータストリームのメトリクス、ログ、トレースを何十億も生成するため、分散アプリケーション全体のパフォーマンスを分析することに困難を伴います。可観測性は、ソースからユーザーに影響を及ぼしている原因を特定したり、破損したコードパスやコストの高いコードパスをすばやく検出したりして、デベロッパーの生産性を高めます。

継続的な可観測性を利用する最初のステップは、Amazon CloudWatch（または好みに応じて Amazon Managed Service for Prometheus、Amazon Managed Service for Grafana、Amazon Distro for OpenTelemetry のいずれか）を使用して可観測性ダッシュボードを構築することです。可観測性ダッシュボードは、クラウドサービスのアクティビティの管理という課題に対処します。これらは、システムへのヒューマンフェイシングの視点です。これらのシステムは、時系列のメトリクス、ログ、トレース、アラームデータを表示することで、システムがどのように動いているのかの要約を提供します。ダッシュボードを構築したら、お客様は CloudWatch のアラームを設定し、クラウド環境の潜在的な問題に関するアラートを継続的に受け取るようにする必要があります。

「私たちは、CloudWatch を使用して Amazon Simple Queue Service 用のアラートと、EC2 Auto Scaling グループのスケーリングアクション用のアラートを作成しています。これらのインサイトに基づいてスケーリングの決定を下し、カスタマーエクスペリエンスに影響を与えることなく、AWS コンピューティングリソースを最も効率的に活用できます。CloudFormation を使用して、顧客向けのアプリケーションで使用する AWS リソースの論理スタックをデプロイできます。これにより、複数の AWS アカウント全体で繰り返して一貫したデプロイを行うことができます。その結果、間違いやすく一貫しない手動の設定に依存しないで、コードとしての設定を使用してアプリケーションスタックを容易に作成できるようになっています」

— Just Eat、シニアエンジニア、Martin Costello 氏

ファクト：
AWS は月平均 1,000 兆のメトリクスの観測結果をモニタリングしている



運用を可視化するための ダッシュボードの構築

この Amazon Builders' Library の記事では、AWS プリンシパルエンジニアの John O'Shea が、システムの状態を理解するためのダッシュボードの構築に関する Amazon DevOps チームの見解を紹介しています。

詳細はこちら ›

継続的なコンプライアンス

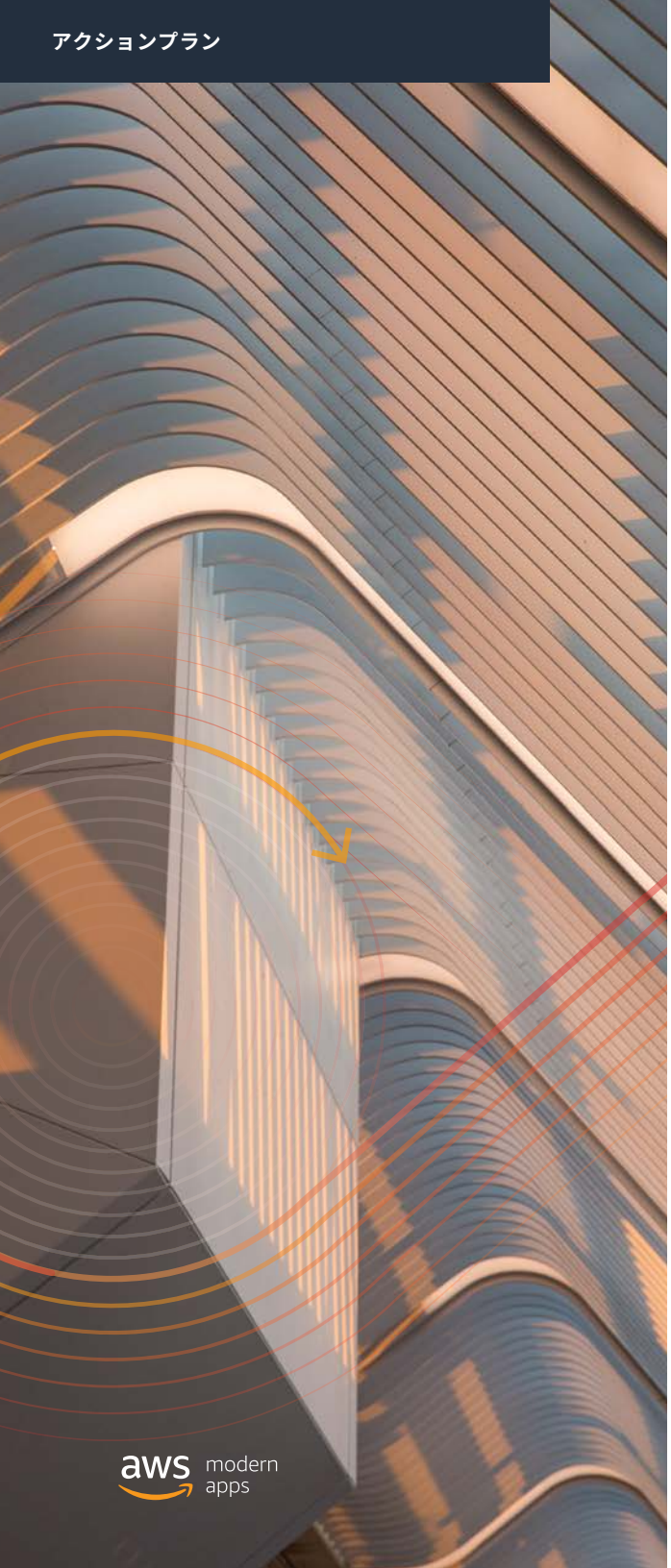
クラウド内のコンプライアンスは、まったく新しいパラダイムです。従来のコンプライアンス手法はクラウドリソースのスピードとスケールに対応できません。スプレッドシートや静的データベースではリソースチャーンを処理できません。成功への鍵は、自動化と簡素化を極限まで進めることです。AWS は、お客様がクラウドコンプライアンスを簡素化するために役立つさまざまなサービスを提供しています。コンプライアンスのためにどのサービスを使用するかを理解するための 1 つの方法は、内部監査手法に関する国際的に認知された機関である The Institute of Internal Auditors (内部監査人協会) の 3 つのラインモデルに従ってサービスを検討することです。このモデルは、組織が 3 つのディフェンスラインに従い、ITの特定の役割と責任を通じてリスクを管理することを明確に示します。第 1 ラインはリスクの管理、第 2 ラインはリスクの監視、第 3 ラインはリスクのアシュアランスの提供を役割とします。

AWS は、これら 3 つのラインのそれぞれに適合するさまざまなサービスを提供しています。例えば、第 1 ラインのリスク管理のサービスとして、AWS Config、AWS CloudTrail、AWS Systems Manager を提供しています。これらのサービスでは、リスクを管理するためのコントロールを個別に定義およびデプロイできます。第 2 ラインについては、AWS Security Hub と Amazon CloudWatch を使用して組織全体のリスクを監視できます。第 3 ラインのリスクのアシュアランスを提供するには、AWS Audit Manager が最適なツールです。お客様は、このツールを使用して監査用にコンプライアンスの証拠を集めるプロセスを自動化できます。

ただし、これらのツールを使用するだけでは不十分です。お客様は、継続的なコンプライアンスでプロセスを自動化する必要があります。継続的なコンプライアンスを導入するには、コードとしてのコンプライアンス (compliance as code) の概念を活用することをお勧めします。コードとしてのコンプライアンスにより、AWS CloudFormation などのツールを使用して 3 つのラインモデルの要素を定義できます。次に、AWS CodePipeline などの自動化されたパイプラインを使用して、これらの要素をテストしてデプロイできます。新しいコントロールや証拠が必要になった場合、これらをアカウントごとに手動で実装する必要はありません。コードはコントロール定義で更新され、パイプラインを通じてプッシュされます。その結果として、リスクを管理するための新しいコントロールが、監査目的の証拠として即座に追加されます。継続的なコンプライアンスには、コントロールが整っており、証拠のコレクションが改ざんされていないことの具体的なアシュアランスを提供するという追加の利点もあります。3 つのラインモデルを活用して継続的なコンプライアンスでプロセスを自動化することは、大規模なコンプライアンスを処理するための最適なアプローチです。

「AWS により、私たちは仕事のやり方を根本的に変革し、コンプライアンスプロセスを自動化することで人員を増やさずにスケールできるようになりました。コンプライアンスは管理しにくい面がありますが、AWS Config のおかげで、コンプライアンスに深い知識がなくても、すぐに使えるテンプレートに従ってコンプライアンスを実装できます。ダッシュボードは、準拠していないリソースなどに関する情報を提供することで、セキュリティ体制を改善するきっかけを提供します。また、どこに着目して修正すべきかについて組織全体で話し合うための開始点ともなります。デジタルグループは、クラウドテクノロジーを通じて社内外のチームにテクニカルリーダーシップを提供するため、組織全体のセキュリティ体制を一元的に監視および管理できることが重要です。パッケージ済みの Config 適合パックテンプレートは、このプロセスを簡素化しており、クラウドへの移行全体を加速させる主要因となっています」

— Baker Tilly Digital、シニアソリューションアーキテクト、Andrew Clark 氏



共有サービスプラットフォームの構築

チームや組織ごとに Dev+Ops を実装する方法はわずかに異なります。多くの組織には、中心となるプラットフォームチームがあり、セキュリティ、ソフトウェア配信、モニタリング、ネットワークを標準化することで開発チームをサポートし、運用上の負担を軽減する役割を果たしています。共有サービスプラットフォームは、コードをデプロイするために合理化された、デベロッパー専用のセルフサービスインターフェイスです。中心となるチームは、セキュリティ、ソフトウェア配信、モニタリング、ネットワークに関する標準を定義して統制します。これらの標準はすべてのデプロイするアプリケーションで使用する必要があります。これにより、デベロッパーの生産性が高まり、プラットフォームチームはより詳細に制御できるようになります。

AWS は、共有サービスプラットフォームを AWS で構築するために必要なすべてのツールを提供しています。本番稼働用の基本的なユースケースの場合、AWS Copilot CLI を使用すると、すぐに使えるマルチアカウント CI/CD、セキュリティグループ、モニタリングを完備した開発環境を簡単に作成できます。より複雑な状況の場合、AWS はフルマネージドサービスである AWS Proton を提供しています。AWS Proton は、コンテナとサーバーレスアプリケーション用の最初のフルマネージド配信サービスです。プラットフォームチームは、AWS Proton を使用して、インフラストラクチャのプロビジョニング、コードのデプロイ、モニタリング、更新に必要なすべての異なるツールを連携させることができます。AWS Proton は、この問題を解決するために、複雑さを管理し、一貫した標準を適用するために必要なツールをプラットフォームチームに提供するとともに、デベロッパーがコンテナやサーバーレステクノロジーを使用してコードを簡単にデプロイできるようにします。参考として、共有サービスプラットフォームをより詳細に制御して構築する必要がある場合は、多くの組織において構築に AWS CloudFormation と AWS Service Catalog を使用し、AWS での使用を承認した IT サービスのカタログを作成、管理することに成ししています。同じように、動的プログラミング言語に慣れているプラットフォームチームを擁しているお客様は、 AWS CDK を使用して開発チーム用のイネーブルメントプラットフォームを構築しています。

「当社の技術者は、AWS Service Catalog のセルフサービスポータルを使用することで、チケットを開くことなく、アプリケーションビルドを本稼働環境に簡単にデプロイできます。その結果、通常は何週間もかかる、新しいアプリケーションスタックの立ち上げを 5 分で完了できます。AWS Service Catalog のおかげで、Wiley では DevOps とオートメーションが機能しています」[4]

— Wiley、プラットフォーム機能部門バイスプレジデント、Meltem Dincer 氏

[4] *https://aws.amazon.com/solutions/case-studies/wiley/*

Dev+Ops の導入が進んでいる組織にも、まだの組織にも、AWS は開発チームと運用チームのギャップを埋めるために設計したさまざまなソリューションを提供します。

AWS デベロッパーツールは、簡単に使い始めることができます。また、セキュリティ、コンプライアンス、高可用性、アクセス制御などの重要なニーズに対応するとともに、事前の警告やベストプラクティスに基づく提案を行うプロアクティブなソリューションをお客様に提供して、未然に問題を特定し、防止します。AWS を使用すると、お客様はクラウドの利点をすばやく実現し、AWS のモダン Dev+Ops のプラクティス、ツール、トレーニングのコアセットを通じてイノベーションを加速できます。

モダン DevOps に関する詳細を参照する ›