aws modern apps

AWS のコンテナサービスでモダナイゼーションを促進

現在、世界中の企業がデジタルトランスフォーメーションに取り組んでいます。企業はアプリケーションのモダナイゼーションを行うことで、より優れたサービスを顧客に提供し、競争の激しい市場で生き残ることができます。AWS は、企業のモダナイゼーション支援の 1 つとして、コンテナの実装により開発を効率化すること、つまり企業文化の改革のきっかけを作ってきました。この日本語ガイドでは、コンテナ化のベストプラクティスと、AWS でどのようにコンテナを使用するのかを説明します。

競争の中で生き残る

ここ数年、世界中でデジタルトランスフォーメーションの動きが急激に加速しています。中小企業から大手企業まであらゆる企業が、俊敏性を高め、顧客の要求により優れた対応をするための新たな手段を探しています。変わり続ける環境の中で生き残ろうとする各社のニーズが、この動きを加速させています。今はデジタルか破綻かの時代です。クラウドネイティブな企業が業界を破壊し、レガシービジネスを追い抜いています。企業のデジタルトランスフォーメーションは多くの場合、アプリケーションのモダナイゼーションを行い、オートメーション環境をクラウドで活用することから始まります。モダナイゼーションにより企業は次のことが可能になります。

伸縮性 : 顧客のニーズの急速な増加に対応できる

可用性 : いつでもどこでも顧客の要望に対応できる

俊敏性 : 問題を迅速に解決したり、顧客の求める新たな機能をすばやくデプロイできる

コンテナは目標を達成するためのツール

デジタルトランスフォーメーションには時間がかかりますが、最終的には生産性が向上し利益がもたらされます。企業のモダナイゼーションのためのツールは数多く出回っていますが、その中でもコンテナは、アプリケーションを効率的にパッケージ化しデプロイするための有力なソリューションとして、デベロッパーの間で継続的に人気が高まっています。Cloud Native Computing Foundation (CNCF) による 2020 年の調査によると、本番環境におけるコンテナの使用は、2016 年から 300% 増加しています。これは、2 つの異なるコンテナ技術である Docker と Kubernetes が今日どれほど広く使用されているかを見れば明らかです。クラウド上にホストされた Kubernetes ワークロードのうち、82% が AWS で実行されています [1]。

開発はコンテナでシンプルになる

コンテナは、アプリケーションを簡単に実行しどこからでもスケーリングできる、ポータブルで安定した軽量のソフトウェア環境を提供します。アプリケーションはそのライフサイクルにおいて、テストや本番稼働といった初期ステージや、オンプレミスの仮想マシンから、移行時のクラウドまで、多様な環境で実行されます。コンテナが開発される以前は、IT チームは新たな環境の互換性を考慮し、アプリケーションが適切に機能するようコードを書き足さなければなりませんでした。コンテナが開発され、アプリケーションを依存関係や設定ファイル、インターフェイスと一緒にパッケージ化できるようになり、デベロッパーは、異なるホスト間をシームレスに移動する 1 つのイメージを利用できるようになりました。これにより、インフラストラクチャ管理の差別化につながらない重労働は開発チームから取り除かれました。

コンテナがあれば、デベロッパーは新機能の追加や最新のセキュリティの追加など、アプリケーションの構築に集中できます。さまざまな環境間の互換性の管理に時間をとられることはありません。

さらにコンテナは、アプリケーションのモノリシックな従来のアーキテクチャを打破する上で不可欠です。コンテナによって、マイクロサービスに移行し、より扱いやすいスケールを実現することが可能になります。マイクロサービスを使用すれば、それぞれのアプリケーションの構成要素を独立したサービスとして実行できるため、デベロッパーは、さまざまな側面に個別に対応することができます。

[1] *Nuclear Research による 2019 年のレポート*

コンテナは開発プラクティスにおいて持続可能なカルチャーシフトを促す

コンテナは、アプリケーションのモダナイゼーションを行うためのツールであるだけでなく、開発プラクティスの向上にも大きく貢献します。コンテナでは、デベロッパーに開発するアプリやコードの品質管理を積極的に委ねることで、従来の開発サイクルを革新します。これまでデベロッパーはアプリケーションの構築のみに集中し、パッケージ化やデプロイの成功についてはほとんど監督することはありませんでした。コンテナにより、デベロッパーは「シフト」と呼ばれる現象を経験します。これにより、開発プロセスにおける品質管理の優先順位がかなり低い順位から前面へと押し出され、デベロッパーはこの責任を持つようになります。

デベロッパーが自身の成果物の品質に責任を持つようになったら、次に必要なのは、初期に失敗し、その失敗から学ぶという文化を育てることです。コンテナ化では、コードの統合とデプロイを自動化し、企業の俊敏性を総合的に高めることによりこれを実現します。

老舗企業でもモダナイゼーションは可能

創業 100 年を迎えたある大手企業が、グローバル市場において自社アプリケーションのモダナイゼーションに成功しました。この企業の開発チームは、こうしたクラウドベースのアプローチを導入することで高い俊敏性を達成し、はるかに迅速なイテレーションを可能にしました。コンテナと、AWS によって自動化される環境を利用することで、新しいプラットフォームをわずか 6 週間で概念からプロトタイプ、プロトタイプからデプロイへと進めることができました。

今すぐ AWS のコンテナでモダナイゼーションに取り掛かる

AWS には、コンテナ化をシームレスに行うためのソリューションがすべてそろっています。インフラストラクチャのプロビジョニング、オーケストレーション、セキュリティ、ネットワーク、オートメーション、モニタリングのための優れたツールをすぐに活用し、コンテナを使ったモダナイゼーションに着手できます。

プロビジョニング

基盤となるインフラとリソースをシームレスにプロビジョニングする

コンテナの実行には、基盤となるインフラストラクチャのプロビジョニングが必要です。AWS では、目標とする管理範囲やオートメーション範囲に応じて、2 つのソリューションを用意しています。

- AWS Fargate による基盤インフラストラクチャのプロビジョニングの自動化
- Amazon Elastic Compute Cloud (Amazon EC2) インスタンスによるインフラストラクチャのコンピューティング、ストレージ、ネットワーク機能の手動の定義

オーケストレーション

Docker や Kubernetes のコンテナのスケーリングと管理を行う

- Amazon Elastic Container Registry (Amazon ECR) を利用し、Amazon Elastic Container Service (Amazon ECS) をオーケストレータとして使用した、Docker イメージの保存と管理
- Amazon Elastic Kubernetes Service (Amazon EKS) の導入による Kubernetes コンテナのオーケストレーション

セキュリティ

コンテナの脆弱性を保護、スキャン、検出する

- AWS Identity and Access Management (IAM)、タグ付け、Amazon EC2 インスタンスのセキュリティグループ、Amazon Virtual Private Cloud (VPC) によるコンテナの保護
- イメージスキャンのソリューションによる、Docker コンテナイメージの脆弱性の検出

ネットワークと接続

コンテナ間のアプリケーショントラフィックを分散する

- AWS Elastic Load Balancing を使用した、コンテナおよびサーバーレス環境間でのアプリケーショントラフィックの分散
- AWS Global Accelerator および AWS Elastic Load Balancing を使用した、コンテナで実行中のグローバルに分散されたアプリケーション用のトラフィックのルーティング
- AWS Global Accelerator および AWS Elastic Load Balancing を使用した、アプリケーションパフォーマンスの改善
- AWS App Mesh を使用した、サービス間の通信とセキュリティの管理

オートメーション

継続的インテグレーションと継続的デリバリー (CI/CD) を使用し、コードを自動的にデプロイする

- AWS CodeCommit を使用したソースコードリポジトリの作成
- AWS CodePipeline を使用した CI/CD パイプラインの設定
- AWS CodeBuild のデプロイによる、独自のコンテナイメージの構築
- AWS App Runner でのコンテナ化されたウェブアプリケーションの構築、デプロイ、実行

観測とモニタリング

コンテナで実行されるサービスが、正常に機能し、期待どおりの相互コミュニケーションを行うようにする

- AWS App Mesh のデプロイによるログ記録、メトリクス、追跡の可視化、およびロードバランシングとトラフィック形成の実現
- Docker コンテナイメージのヘルスチェックによる、コンテナとアプリケーションの実行の確認
- Amazon CloudWatch Application Insights を使用した、Amazon EC2 上の Amazon ECS、Amazon EKS、または Kubernetes でデプロイされたコンテナで実行中のアプリケーションのヘルスおよび正常性のモニタリング

プロビジョニング

AWS は、Amazon EC2 と AWS Fargate により、インフラストラクチャをプロビジョニングするオプションを提供します。この 2 つのツールの主な違いは、コンテナアプリケーションを実行する基盤インフラストラクチャに対して可能な維持管理のボリュームにあります。

- **AWS Fargate:** サーバーやクラスターを管理することなくコンテナを実行できます。アプリケーションをコンテナにパッケージ化して、CPU とメモリの要件を指定し、ネットワークと IAM のポリシーを定義したら、あとはアプリケーションを起動するだけです。

- **Amazon EC2:** 以前からある起動タイプで、インスタンスにコンテナを実行させることができ、コンテナアプリケーションを実行するインフラストラクチャを、サーバーレベルできめ細かく管理できます。

オーケストレーション

アプリケーションをコンテナ化したら、次はコンテナを本番環境で実行します。アーキテクチャをスケールするには、オーケストレーションツールが必要です。AWS は、オンプレミスでもクラウドでも、お客様のニーズに合ったオーケストレーションプラットフォームを提供します。

Amazon ECS および Amazon ECS Anywhere:

Amazon Elastic Container Service (Amazon ECS) は、コンテナ化したワークロードの AWS でのデプロイを簡単にします。Amazon ECS の強力なシンプルさにより、1 つの Docker コンテナから成長して、エンタープライズアプリケーションポートフォリオ全体の管理を実現できます。コントロールプレーンまたはノードを管理する複雑さがなく、アベイラビリティーゾーン、クラウド上、オンプレミス全体でコンテナワークロードを実行およびスケールできます。Amazon ECS は、マネージド型の、無限にスケーラブルなコントロールプレーンを備えています。このタイプのオーケストレーションツールは、独自のオペレーティングシステムを使用する企業や、自社のインフラストラクチャの管理機能を必要とする企業に最適です。AWS Copilot は、コンテナ化されたアプリケーションを AWS ですばやく起動し、簡単に管理することができるコマンドラインインターフェイス (CLI) です。AWS Copilot は、お客様の迅速なデプロイを支援するために組み込まれたサンプルやガイド付きの操作を含む、シンプルな宣言コマンドのセットを提供します。本番環境で使用できるサービスを導入するために必要なものは、AWS Copilot、AWS アカウント、コードのみです。

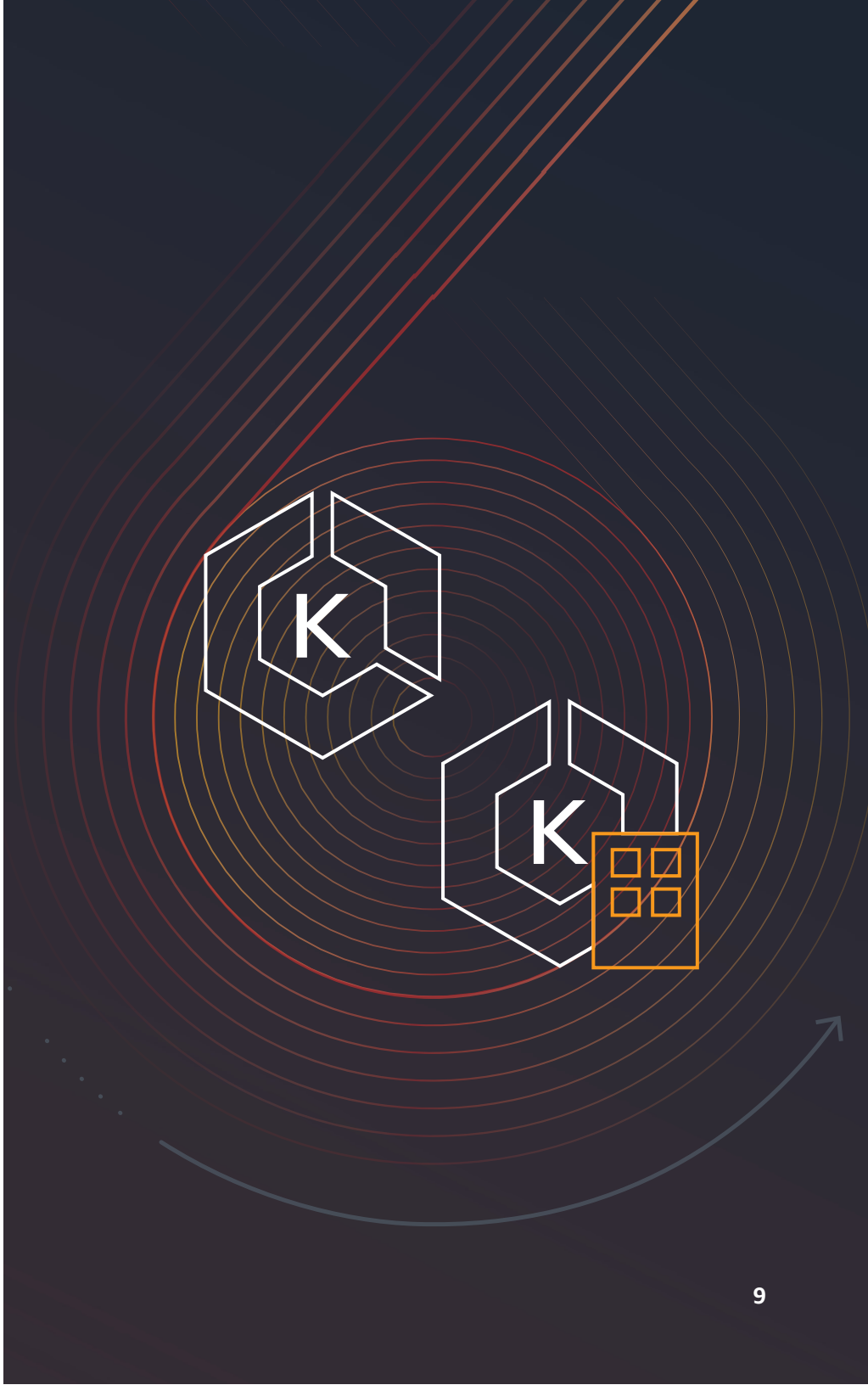
Amazon ECS Anywhere を使用すれば、使い慣れた同じ Amazon ECS コンソールとオペレーターツールにより、オンプレミスコンテナワークロードを管理し、コンテナベースのアプリケーション全体で一貫したエクスペリエンスを実現できます。AWS Systems Manager (以前の SSM) との統合により、オンプレミスハードウェアと AWS コントロールプレーン間で自動的かつ安全に信頼を確立できます。

Amazon EKS および Amazon EKS Anywhere:

Kubernetes は、大規模なコンテナの実行を容易にするために開発された、急成長中のオープンソースのコンテナ管理プラットフォームです。Amazon EKS は、複数の AWS のアベイラビリティーゾーンにまたがり Kubernetes の管理インフラストラクチャを実行します。

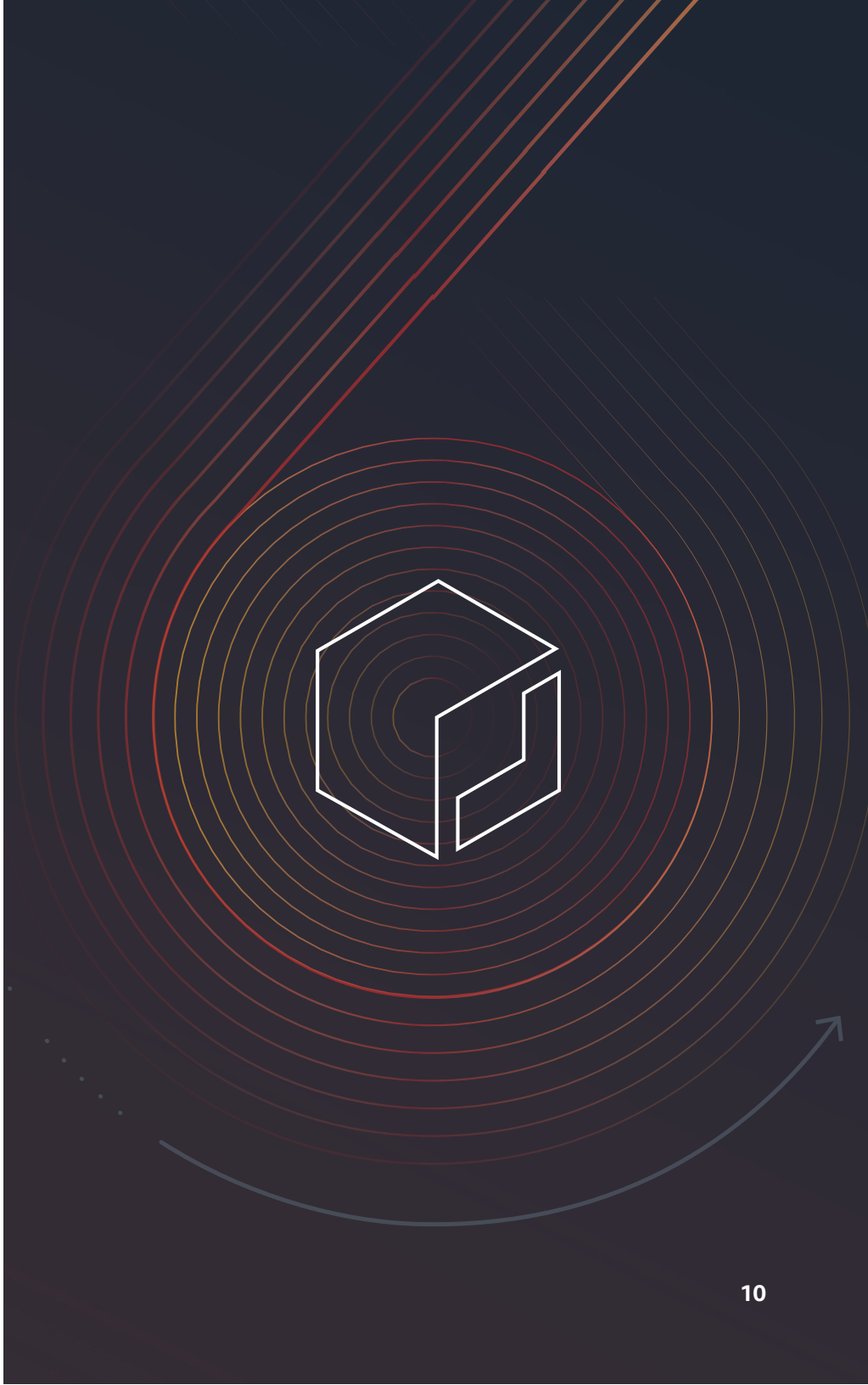
AWS のお客様は、AWS で Kubernetes を実行することのメリットは驚くほど大きいと感じています。Amazon EKS は、コンテナのスケジュール、アプリケーションの可用性の管理、クラスターデータの保存、その他の主なタスクを行う Kubernetes コントロールプレーンノードの可用性とスケーラビリティを自動的に管理します。Kubernetes アプリケーションは、コンテナ用サーバーレスコンピューティングを提供する Amazon EC2 と AWS Fargate のどちらでも実行できます。

Amazon EKS Anywhere により、独自の仮想マシン (VM) とベアメタルサーバーを含めて、オンプレミスで Kubernetes クラスター (Amazon EKS Distro でのソフトウェアによる構築) を簡単に作成および運用できます。EKS Anywhere は、Kubernetes クラスターを管理するための独自のツールを構築およびサポートする複雑さをなくします。EKS Anywhere は、ログ記録、モニタリング、ネットワーク、ストレージのデフォルト設定を備えた、ベアメタル、vSphere、クラウド仮想マシンなどのインフラストラクチャ上でのクラスターの作成、管理、運用を簡単にするオートメーションツールを提供します。その一方で、クラスターのインストールとライフサイクル管理、可観測性、クラスターのバックアップ、ポリシー管理など、本番環境で Kubernetes を実行するために必要となる、自律的なツールや追加のコンポーネントを提供します。Amazon EKS Distro は、オンプレミスでのお客様独自のインフラストラクチャで使用するために AWS 上の EKS で使用されるのと同じ、オープンソースの Kubernetes ソフトウェアディストリビューションをパッケージ化します。EKS Distro クラスターは、お客様独自のツールまたは Amazon EKS Anywhere を使用して管理できます。



Amazon ECR:

Docker コンテナを使用するために必要な最初のものは、Docker イメージです。このイメージは、コンテナのインスタンスを作成するための設計図の役割を果たします。Amazon ECR は、Docker コンテナイメージを簡単に保存、管理、デプロイできるようにするためのフルマネージド Docker コンテナレジストリです。Amazon ECR は Amazon ECS に統合されているため、開発から本番までのワークフローを簡略化できます。Amazon ECR を使用することにより、独自のコンテナリポジトリを運用したり、その基盤インフラストラクチャをスケーリングしたりする必要がなくなります。

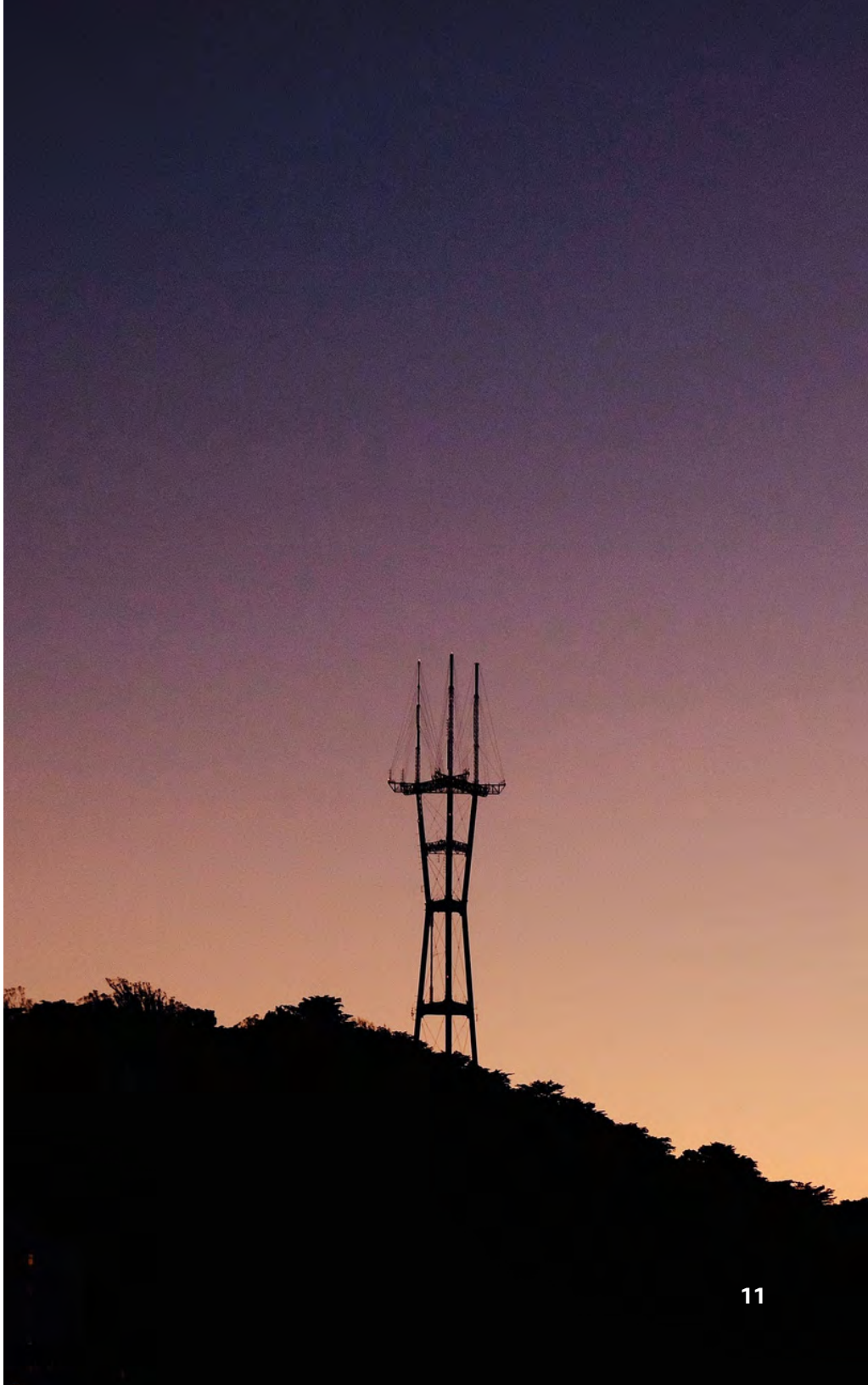


セキュリティ

品質管理と同様に、セキュリティの課題対応も開発サイクルの初期段階に移行しています。より高い自主性により、デベロッパーはすばやくコードを適用して新たなセキュリティの脅威に対処できます。ただし、セキュリティは組織全体での優先事項とする必要があります。セキュリティの取り組みをできる限り透明化し、セキュリティのベストプラクティスとツールを考慮したアーキテクチャを最初に定義することで、このような企業文化のシフトを支援することができます。

AWS は、コンテナにアクセスできるユーザーを管理するための複数のツールを提供しています。AWS IAM を使用して、誰を認証 (サインイン) し、誰にリソースの使用を承認する (アクセス許可を持たせる) かを決定できます。Amazon VPC を使用すれば、定義した仮想ネットワークでコンテナタスク (Docker) やポッド (Kubernetes) を論理的に隔離できます。セキュリティグループを定義すれば、EC2 インスタンス間で仮想ファイアウォールを作成できます。

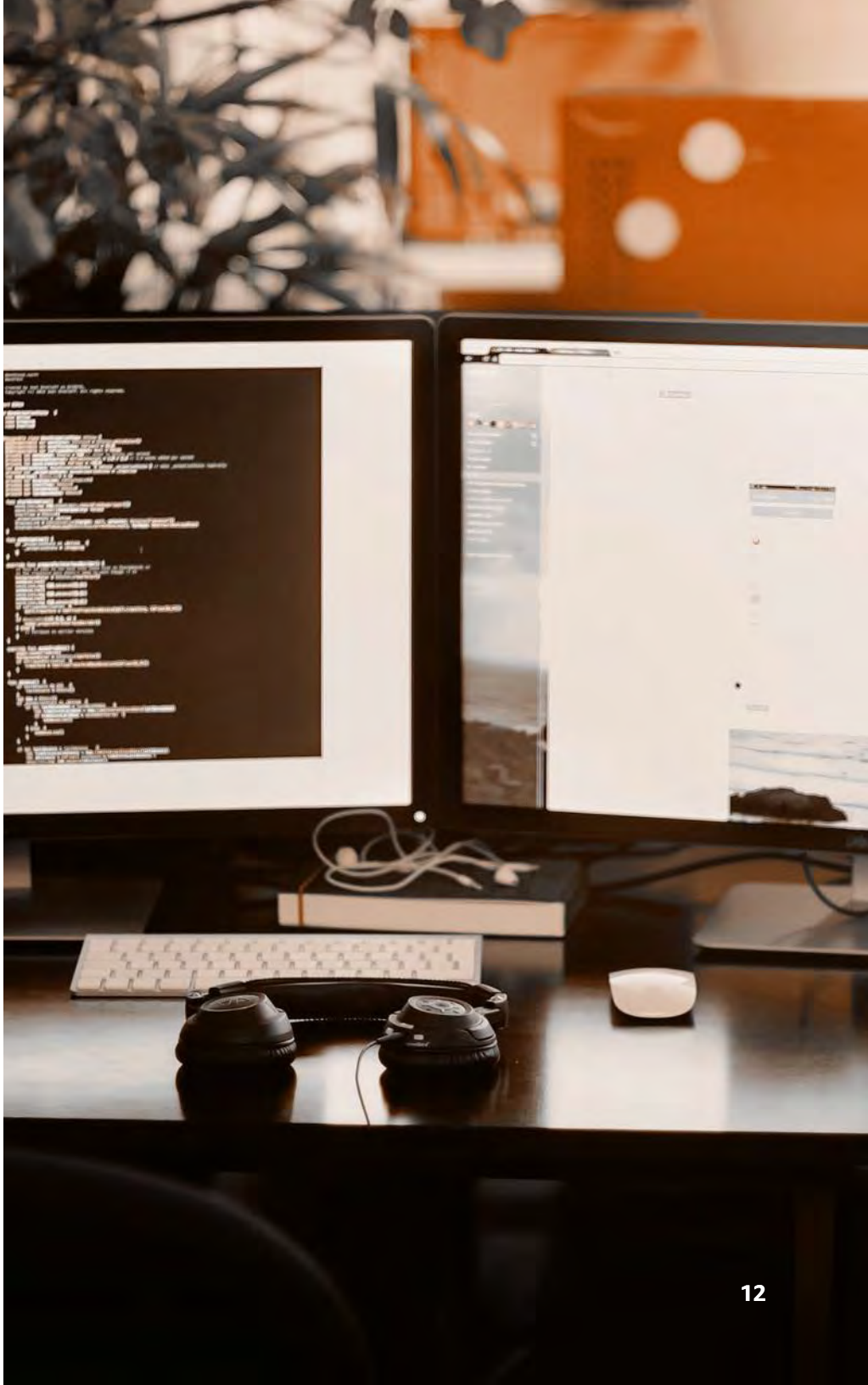
イメージスキャンのソリューションでは、コンテナイメージの脆弱性やイメージの依存関係を検出できます。セキュリティチームは、コンテナやイメージの依存関係をスキャンしてから事前承認済みのリソースを公開することができるため、デベロッパーは安心してこれらのリソースを使用できます。



ネットワークと接続

アプリケーションを実行したら、トラフィックが常にコンテナ全体で分散されるようにし、エンドユーザーが中断なくアプリを使用できるようにする必要があります。

- **Elastic Load Balancing** は、AWS コンテナサービスとネイティブに統合された高度なロードバランシングアルゴリズムを使用して、ユーザーが簡単にアプリケーションにアクセスできるよう支援します。
- **Amazon Global Accelerator** は、アプリケーションのパフォーマンスが高く、アプリケーションが世界中のユーザーがアクセス可能で、ユーザーに最も近い AWS リージョンから提供されるよう支援します。
- **AWS App Mesh** は、コンテナサービス間の通信を管理し、きめ細かなセキュリティと高い可視性を実現します。



オートメーション

これから 2024 年までは、ほとんどの形態の PaaS サービスの競合増加に連動して、ハイブリッド統合機能の需要が高まるでしょう [2]。Amazon EKS Anywhere と Amazon ECS Anywhere を導入すれば、シンプルで使い慣れたツールを使用しながら、お客様のインフラストラクチャ上でワークロードを実行して、コンプライアンス、データグラビティ、その他のビジネス要件を満たすことができます。ローカルコントロールプレーンをインストールして運用する代わりに、同じハイパースケールで、信頼されたフルマネージドのコントロールプレーンを、オンプレミスコンテナワークロード用に使用できます。お客様独自のハードウェア上のエッジロケーションでコンテナ化されたデータ処理ワークロードを実行し、エンドカスタマーに近い場所から対応して、低いレイテンシーを維持できます。

オートメーション環境では、コードを手動でデプロイする必要がありません。インフラストラクチャに無数のコンテナが含まれている場合、継続的インテグレーションと継続的デリバリー (CI/CD) を使用して自動化することで、人的ミスのリスクを最小限に抑えつつ、迅速にスケーリングや措置対応を行うことができます。AWS CodeCommit、AWS CodePipeline、そして AWS CodeBuild はソースコードリポジトリの作成、CI/CD パイプラインの設定、コンテナイメージの構築を可能にします。

AWS App Runner は、デベロッパーが、コンテナ化されたウェブアプリケーションと API の大規模なデプロイおよび実行をわずかなクリック操作で簡単に行えるようにする、フルマネージドアプリケーションサービスです。App Runner により、インフラストラクチャの設定と管理の必要がなくなります。ソースコード、コンテナイメージ、またはデプロイパイプラインを提供するだけで、App Runner はウェブアプリケーションの構築とデプロイ、ネットワークトラフィックのロードバランシング、需要に応じたキャパシティーの拡大と縮小、アプリケーションヘルスのモニタリング、デフォルトでのトラフィックの暗号化を行います。App Runner を使用し、コンテナの可搬性、効率性、コスト削減効果を活用するために、過去のコンテナの経験は必要ありません。

[2] Gartner Forecast による分析、2021 年 2 月

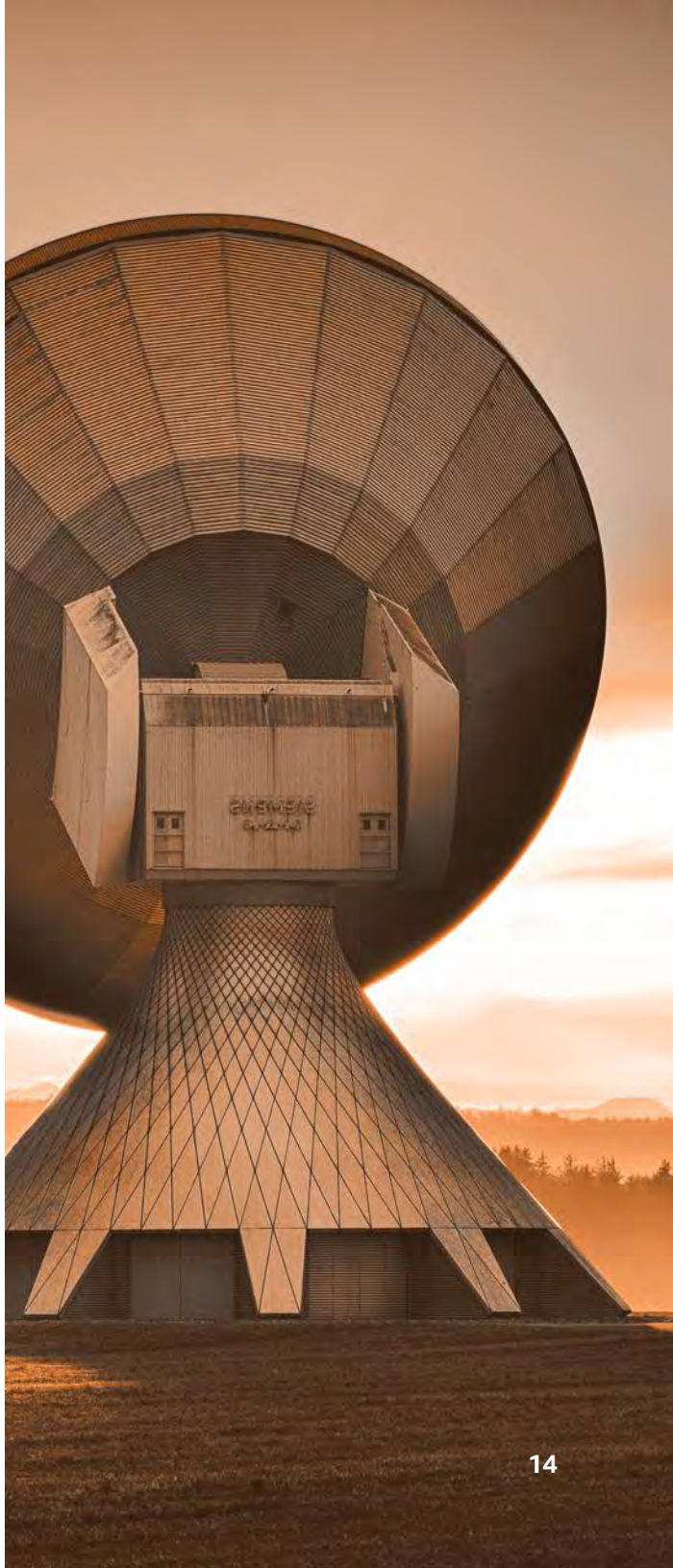
可観測性とモニタリング

コンテナを使ってマイクロサービスをデプロイしたら、コンテナのヘルスをモニタリングし、各サービス間で確実に期待どおりのコミュニケーションが行われるようにする必要があります。

- **AWS App Mesh** は、さまざまなタイプのコンピューティングインフラストラクチャにまたがり構築されたサービスに対し、一貫した可視性およびネットワークトラフィック管理を提供することで、サービスの実行を容易にします。また、監視データの収集方法やサービス間のトラフィックのルーティング方法を変更する際にアプリケーションコードを更新する必要はありません。App Mesh は、各サービスが監視データをエクスポートするように設定し、アプリケーション全体に一貫したコミュニケーション制御ロジックを実装します。これにより、エラーの起きた場所をすばやく正確に特定し、障害がある場合やコード変更のデプロイが必要な場合には、ネットワークトラフィックを自動で再ルーティングすることを簡単にできるようにします。App Mesh はオープンソースの Envoy Proxy を使用しているため、AWS パートナーやオープンソースのツールに幅広く対応しています。

- Docker コンテナイメージを使用すると、コンテナやアプリケーションが正常かどうかを検証できます。簡単なヘルスチェックコマンドを実行するだけで、Docker ファイルがコンテナをチェックし、正常に機能しているかを確認します。ウェブサーバーが無限ループに陥り、サーバープロセスが実行されているのに新しい接続に対処できない場合は、これを検出することも可能です。

- **Amazon CloudWatch Application Insights** はコンテナのモニタリングをサポートします。**Amazon ECS**、**Amazon EKS**、AWS で実行中の **Kubernetes on EC2** コンテナにデプロイされたアプリケーションのモニタリング、アラーム、ダッシュボードを簡単にセットアップできます。AWS 上のコンテナで実行中のアプリケーションのメトリクス、テレメトリー、ヘルスと正常性のモニタリングログを取得するためのモニタリング階層オプションが利用可能になりました。



今すぐ、コンテナを活用してみませんか。

AWS と圧倒的な規模のパートナーエコシステムが、ニーズに応じたツールで、モダナイゼーションのどの段階にあるお客様をもサポートします。

AWS 日本担当チームまたは AWS パートナーに、いつでもお気軽にご相談ください »

コンテナの詳細については、aws.amazon.com/containers をご参照ください。

aws modern apps