Kubernetes の説明

最新のアプリケーションは、コンテナー (依存関係と構成がパッケージ化されたマイクロサービス)
を使用して構築されることが増えています。Kubernetes ("クーバネティス" と発音します) は、大
規模なコンテナーをデプロイおよび管理するためのオープンソース ソフトウェアです。また、ギリ
シャの言葉で船の操舵手やパイロットの意味もあります。Kubernetes ("k8s" または "k-eights" と
呼ばれることもあります) を使用して、コンテナー化されたアプリのビルド、配信、拡張を迅速に行
うことができます。

こちらの簡単なラーニング パスで Kubernetes の詳細を学ぶ

Kubernetes のしくみ

アプリケーションが、複数のサーバーにデプロイされた複数のコンテナーにまたがるようになるにつれて、アプリケーションの運用が複雑化します。この複雑さを管理するために、Kubernetes はコンテナーを実行する方法と場所を制御するオープンソース API を提供します。

Kubernetes は、仮想マシンのクラスターを調整し、利用可能なコンピューティング リソースと各コンテナーのリソース要件に基づいて、それらの仮想マシンで実行するコンテナーをスケジュールします。コンテナーはポッド (Kubernetes の基本的な運用単位) にグループ化されており、これらのポッドは目的の状態に合わせてスケーリングされます。

また Kubernetes では、サービスの検出の管理、負荷分散の実現、リソースの割り当ての追跡、コンピューティングの使用率に基づいたスケーリングが自動的に行われます。また、個々のリソースの正常性の確認や、コンテナーの再起動またはレプリケートを自動的に行うことにより、アプリが自己回復できるようにします。

Kubernetes の基本に関するビデオを見る

Kubernetes のしくみを見る　一般的な Kubernetes シナリオを見る

Kubernetes を使用する理由

コンテナー化されたアプリの稼働状態を維持することは、複雑な作業になる可能性があります。これは多くの場合、多くのコンテナーが異なるコンピューターにデプロイされていることが理由です。Kubernetes を使用すると、これらのコンテナーをスケジュール設定してデプロイすることが

き、さらにそれらを目的の状態にスケーリングしたり、ライフサイクルを管理したりすることができます。Kubernetes を使用して、移植性や拡張性に優れ、スケーラブルな方法でコンテナーベースのアプリケーションを実装することができます。

ワークロードの移植性を高める

コンテナー アプリはインフラストラクチャとは切り離されているため、それらを Kubernetes 上で実行することで移植性が高まります。環境間の一貫性を維持しながら、オンプレミス、ハイブリッド、および複数のクラウド環境において、ローカル コンピューターから運用環境に移行することができます。

コンテナーを簡単にスケーリング

Kubernetes を使用すると、複雑なコンテナー化アプリケーションを定義し、それを 1 つのクラスター サーバー、さらには複数のクラスターにデプロイすることができます。Kubernetes は必要な状態に応じてアプリケーションをスケーリングし、コンテナーの正常性を自動的に監視して維持します。

拡張可能なアプリの構築

発者や企業で構成される大規模なオープンソース コミュニティは、セキュリティ、監視、管などの機能を追加する拡張機能やプラグインを Kubernetes で積極的に構築しています。さに、Kubernetes ソフトウェア適合認証プログラムでは、それらのコミュニティのオファリンを簡単に使用できるようにするための API を、すべての Kubernetes バージョンでサポートることが求められています。

e-Book: Kubernetes の利用を開始するための実用的なスキルを学ぶ

Kubernetes の概要

コンテナー化されたアプリケーションをデプロイして管理する方法をご確認ください。

詳細はこちら

ıーニング パスをフォロー

ubernetes のコンポーネント、機能、ソリューションについて、実践的に体験できます。

ガイドを入手する

完全な Kubernetes プラットフォーム上に構築

Kubernetes 自体に、移植性、拡張性、拡張性が備わっており、そこにエンドツーエンドの開発、運用、セキュリティ制御を追加することで、セキュリティや信頼性を損なうことなく、より迅速に更新プログラムをデプロイし、インフラストラクチャ管理の時間を減らすことができます。Kubernetes を採用するときは、次のような実装についても検討してください。

プロビジョニング、修正プログラムの適用、アップグレードなどのルーチン タスクを排除するためのインフラストラクチャ自動化またはサーバーレス Kubernetes。

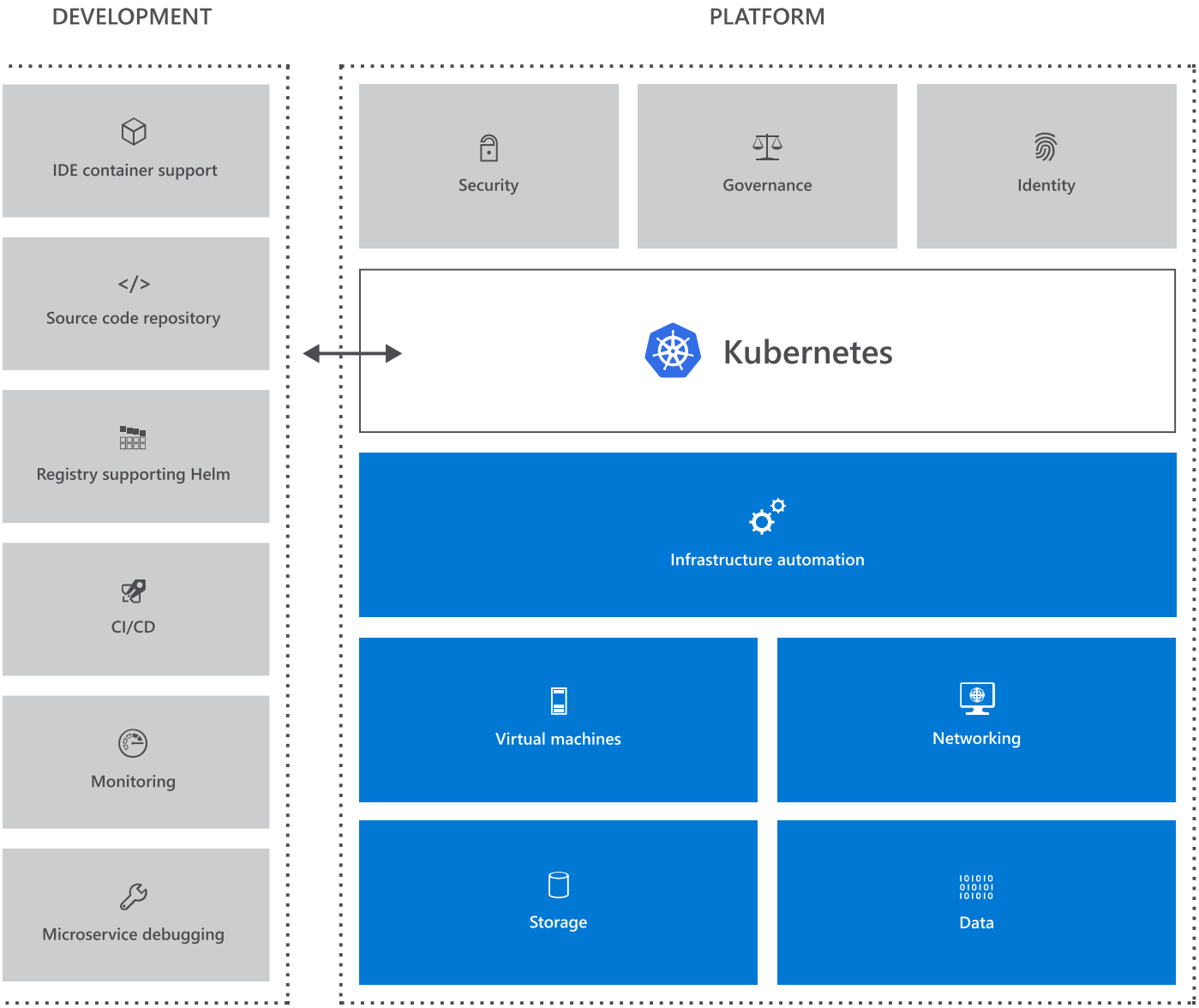
サンプルの確認

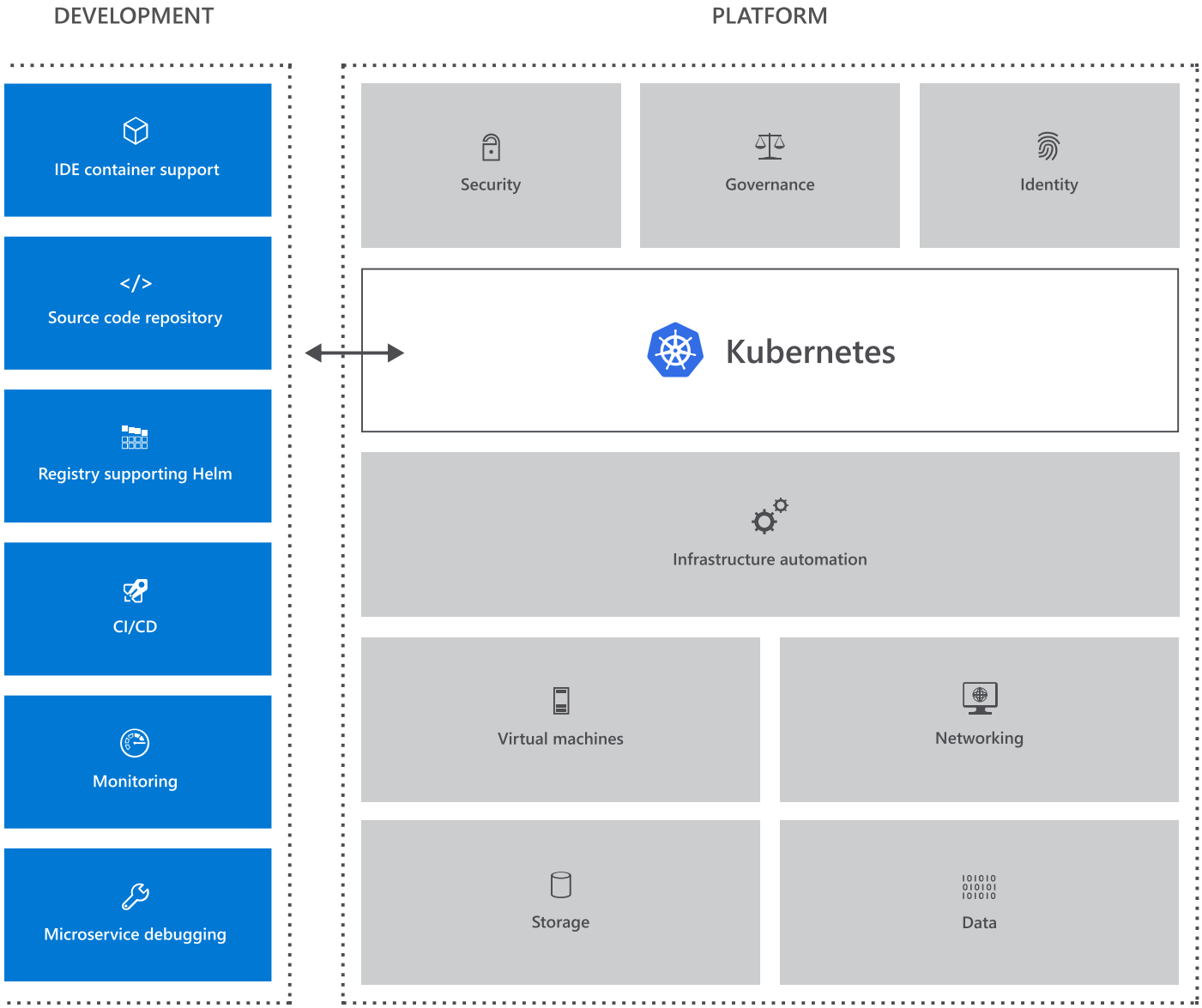
コンテナー化されたアプリの開発用ツールと、継続的インテグレーションおよび継続的デプロイ (CI/CD) ワークフロー。

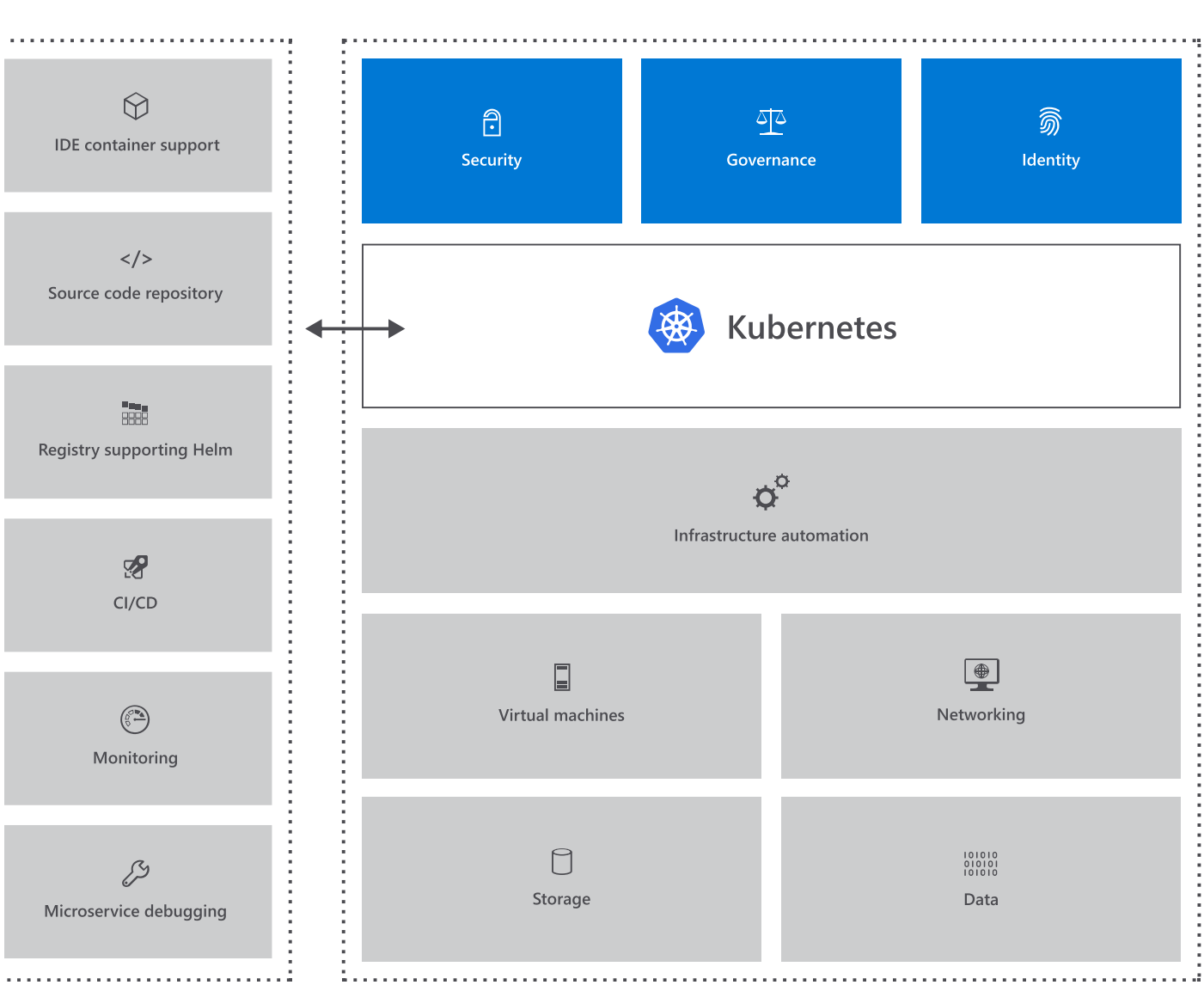
サンプルの確認

セキュリティ、ガバナンス、ID、アクセスを管理するサービス。

サンプルの確認




DEVELOPMENT
PLATFORM
IDE container support
Source code repository
Registry supporting Helm
CI/CD
Monitoring
Microservice debugging
Security
Governance
Identity
Kubernetes
Infrastructure automation
Virtual machines
Networking
Storage
Data



DevOps プラクティスでの Kubernetes の活用

コンテナー、環境、チームが追加されて Kubernetes アプリが大きくなるにつれ、リリースの頻度が増える傾向になり、また開発や運用も複雑になります。Kubernetes 環境で DevOps プラクティスを採用することで、セキュリティを強化しながら大規模かつ迅速に移行することができます。

CI/CD を使用してコードをより迅速に配信する

コンテナーにより、開発チームと運用チーム間のコラボレーションを容易にする一貫したアプリケーション パッケージ形式が提供されます。CI/CD では、それらのタスクを自動化することで、コードからコンテナー、さらに Kubernetes クラスターへの移行を数分で実行できます。

Kubernetes の CI/CD を設定する

コードとしてのインフラストラクチャを使用してリソースを効果的に管理する

コードとしてのインフラストラクチャにより、チーム間でコンピューティング リソースの一貫性と可視性が高まり、人的エラーの可能性が低減されます。このプラクティスは、Helm を利用した Kubernetes アプリケーションの宣言的な性質を使用することで機能します。この 2 つを組み合わせることで、追跡可能で繰り返し可能な信頼性のある方法で、アプリ、リソース、構成を定義できます。

Terraform を使用して Kubernetes クラスターをデプロイする

常時監視によってフィードバック ループを加速する

コンテナーの正常性監視からログの一元化まで、リソース、クラスター、Kubernetes API、コンテナー、コードの全体像を把握することで、バグの検出から修正までの時間を短縮することができます。これにより、リソースのボトルネックを防ぎ、悪意のある要求をトレースし、Kubernetes アプリケーションの正常性を維持できるようになります。

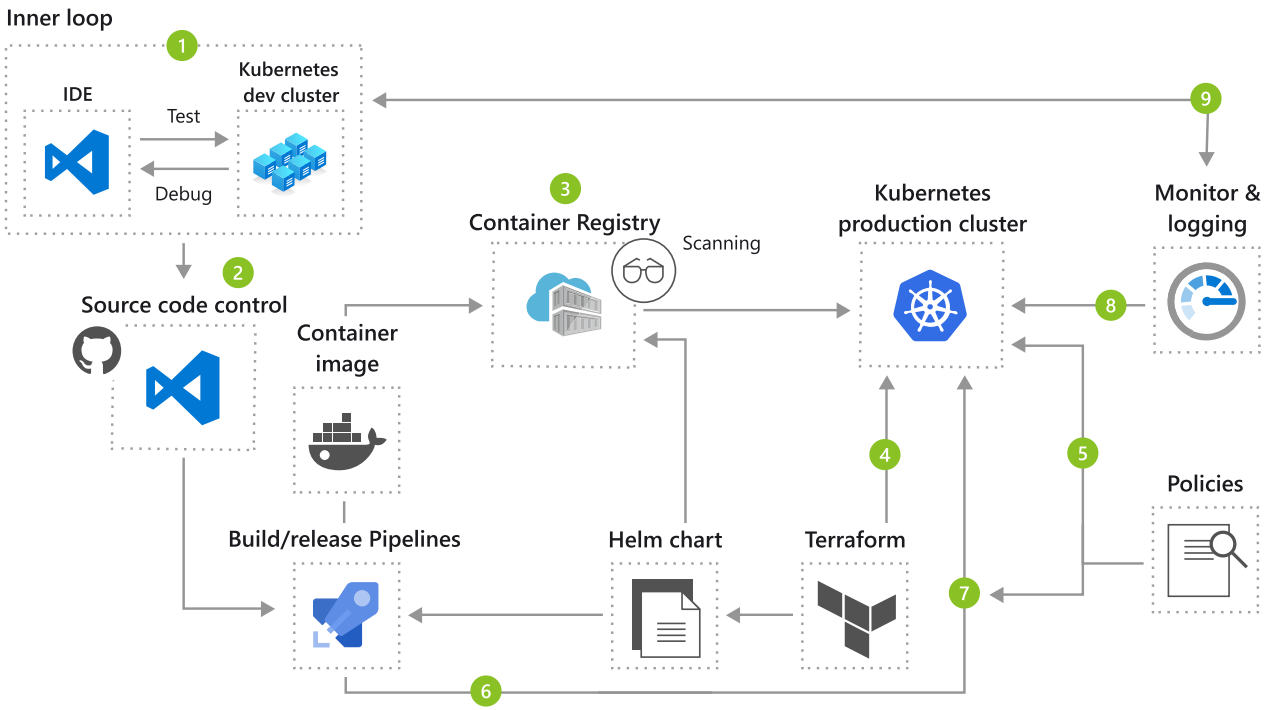
コンテナーのリアルタイムの分析情報のしくみを確認する

evOps を使用してスピードとセキュリティのバランスを取る

度を犠牲にすることなく、DevOps ワークフローをリアルタイムで観測することができま
。コンプライアンス チェックと再構成を自動的に適用して、ビルドとリリースのパイプライ
をセキュリティで保護することで、Kubernetes アプリケーションを保護します。

続的なセキュリティの実例を見る

Kubernetes での DevOps ワークフローの例



1 同じ Kubernetes クラスターで、アプリケーションのさまざまな部分の反復処理、テスト、デバッグを同時にすばやく行います。

2 継続的インテグレーションのために、コードをマージして GitHub リポジトリにチェックインします。次に、継続的デリバリーの一環として、自動化されたビルドとテストを実行します。

3 コンテナー イメージのソースと整合性を確認します。イメージはスキャンに合格するまで検疫に保持されます。

4 Terraform などのツールを使用して、Kubernetes クラスターをプロビジョニングします。Terraform によってインストールされた Helm チャートにより、アプリのリソースと構成の望ましい状態が定義されます。

5 Kubernetes クラスターへのデプロイを制御するポリシーを適用します。

6 リリース パイプラインにより、事前定義されたデプロイ戦略が各コードで自動的に実行されます。

7 CI/CD パイプラインにポリシー監査と自動修復を追加します。たとえば、リリース パイプラインだけが、Kubernetes 環境で新しいポッドを作成するためのアクセス許可を持ちます。

8 アプリのテレメトリ、コンテナーの正常性監視、リアルタイムのログ分析を有効にします。

9 分析情報を使用して問題に対処し、次回のスプリントの計画を通知します。

Kubernetes のデプロイ戦略の詳細情報

Azure で Kubernetes の長所を活用

フル マネージドの Microsoft Azure Kubernetes Service (AKS).を使用して、プロビジョニング、アップグレード、監視、スケーリングを自動化しましょう。サーバーレス Kubernetes、開発から運用までのよりシンプルなエクスペリエンス、エンタープライズ レベルのセキュリティとガバナンスを利用できます。

AKS の詳細を確認する

Kubernetes コミュニティからのインスピレーションやイノベーションを採用する

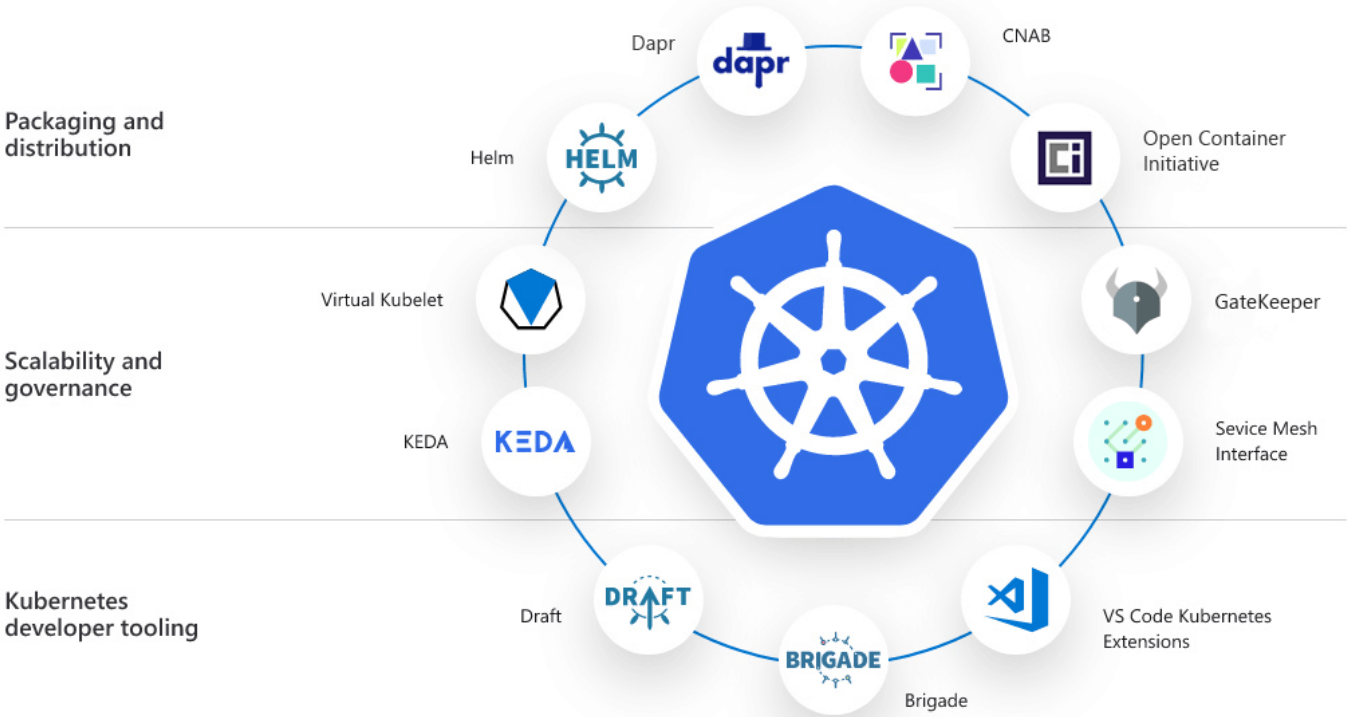
Kubernetes は、数千人の個人と数百社の企業がプロジェクトに注いできた英知、コード、努力の結晶として生まれ、発展しています。情熱のこもった継続的な貢献を活用して、ソフトウェアを成功に導きましょう。

35,000
共同作成者

180,000
コミット

トップ プロジェクト

GitHub で提供
Kubernetes に対する Microsoft の貢献



企業で独創的なオープンソースを実現する

ubernetes を企業が採用しやすく、開発者が使用しやすいものにするため、Microsoft は、このオープンソース プロジェクトに参加する従業員数をわずか 3 年で 3 倍に増員しました。Microsoft 、第 3 位の貢献企業として、さまざまなお客様との協力関係から得られた最新の知見とベスト プクティスを Kubernetes コミュニティに還元することにより、Kubernetes を企業にとってより用しやすいものにすることに取り組んでいます。

FAQ - Kubernetes

何から始めればよいですか?

Kubernetes の一般的なユース ケースを教えてください

Kubernetes のベスト プラクティスを教えてください

Kubernetes のデプロイとは何ですか?

DevOps 手法を使用して Kubernetes にデプロイするにはどうすればよいですか?

Kubernetes および Docker とは何ですか?

リソース

Kubernetes についての詳細情報

Kubernetes の基本を学ぶ

Kubernetes のベスト プラクティスを確認する

コンテナーの詳細を確認する

AKS についての詳細情報

Azure Kubernetes Service (AKS) の詳細を確認する

AKS のビデオとオンデマンドの Azure ウェビナーで、デモ、主要な機能、技術セッションをご覧ください。

マイペースで進められる Azure Kubernetes ワークショップに参加する

Kubernetes 用 Azure クイック スタート テンプレートを参照する

AKS のリージョン別の提供状況を確認する

GitHub、KubeCon、または近くの Kubernetes Meetup で、他の AKS ユーザーと交流しましょう。

AKS のステップバイステップのチュートリアルに従います。

アプリケーションからコンテナー イメージを作成する

Azure Container Registry にコンテナー イメージをアップロードする

AKS クラスターをデプロイする

Kubernetes でコンテナー イメージを実行する

アプリケーションおよび Kubernetes インフラストラクチャをスケーリングする

Kubernetes で実行されているアプリケーションを更新する

AKS クラスターをアップグレードする