SOLUTION BRIEF

Fortinet のコンテナセキュリティ

概要

クラウドネイティブなテクノロジーを導入して新しい製品やサービスの提供を行うことで、企業はビジネスの重要な領域を素早くに変革することができるようになりました。このようなテクノロジーの例として、マイクロサービスアーキテクチャでのコンテナの使用があり、これにより、アプリケーションの構築、テスト、デプロイ、再デプロイの方法を効率化できます。

開発者はコンテナテクノロジーを利用することで、モジュール性が高く柔軟な新しい手法でアプリケーションを設計、開発できるようになります。これは、アプリケーションの異なる論理機能を別個のコンテナに落とし込むことで達成でき、これにより、管理とスケーリングが容易になります。

また、コンテナテクノロジーではアプリケーションを個別の論理機能（マイクロサービス）に分解できるため、さまざまなパブリックおよびプライベートクラウド環境におけるマイクロサービスやアプリケーションのポータビリティを強化できます。

そのような利点があるものの、従来型のセキュリティツールには、コンテナワークロードを保護するための本格的な機能がありません。コンテナセキュリティにおいては、コンテナライフサイクルのすべての段階での可視性と保護機能が求められます。

Fortinet のコンテナセキュリティ戦略では、コンテナベースアプリケーションのライフサイクル全体を対象とした、複数のソリューションが用意されており、コンテナベースアプリケーションがさらされるさまざまな脅威ベクターに関連する脅威に対抗できる、包括的な保護機能が提供されます。

コンテナセキュリティの主な特長

- コンテナを認識できるセキュリティ
- コンテナに対応したセキュリティ
- コンテナ統合セキュリティ
- コンテナレジストリのセキュリティ
- シフトレフトセキュリティ：ソフトウェア開発ライフサイクルにセキュリティを組み込む

チームは、オンプレミスかクラウドかを問わずに、モジュール性の高いアプリケーションをコンテナで迅速に開発できるようになる必要があります。

アプリケーションそのものと同程度のポータビリティ、一貫性をセキュリティに持たせるべきです。

コンテナテクノロジーのメリット

多くの組織が、コンテナの使用を開始してアプリケーションのデプロイを加速しています。2022 年までに、グローバル組織の75% 以上が本番環境でコンテナ化されたアプリケーションを実行するようになると見積もられており、また、2024 年までにエンタープライズアプリケーションの15% 近くがコンテナ環境で実行されるようになると予想されています（2020 年には5% 未満）。[1]

コンテナの導入が加速されている主な要因として、次のようなものがあります：

- **シンプルさとポータビリティ:** コンテナは、軽量な自己完結型のソフトウェアアプリケーションバンドルであり、ホストのオペレーティングシステムからは独立しています。そのため、プラットフォームやクラウドを問わずに一貫した形で実行することが可能です。
- **俊敏性:** コンテナは、より迅速なアプリケーション開発、デプロイサイクルを可能とします。
- **優れた効率性:** 仮想マシン（VM）とは異なり、コンテナはシンプルで軽量なソフトウェアスタックです。要する起動時間が短いため、リソースを効率化して、コストを削減できます。

コンテナでは、自律的機能を持つ小さなオブジェクトにアプリケーションが分解されます。

それぞれに求められるパフォーマンスのレベルは異なるため、それぞれのオブジェクトは別個に維持され、個別にスケーリングされます（バージョンやバグ修正も別個に維持されます）。これらのオブジェクトは通常、サービスと呼ばれます（最近では、マイクロサービスと呼ばれることもあります）。

サーバー仮想化環境で、VM には、ハイパーバイザーまたは仮想インフラストラクチャレベルで確認できる一連のメタデータ属性（一般的に「タグ」と呼ばれます）が付与され、イメージのリポジトリの管理に使用されます。

同様に、コンテナには関連するメタデータ属性が付与され、これは一般的に「ラベル」と呼ばれます。

現時点で、最も一般的なコンテナフォーマットの標準はDocker であり、オープンソースと商用利用の両方があります。

コンテナテクノロジーを使用するアプリケーションを構築するには、相互にやり取りを行う、相互に依存した複数のサービスを用意する必要があります（これは一般にポッドと呼ばれます）。アプリケーションを構築するには複数のコンテナが必要です。コンテナがポッドにグループ化されない場合もありますが、アプリケーションを適切に動かすためには、すべてのコンテナを相互に接続しなければなりません。

コンテナは、サービス/ アプリケーションのコンポジションにより接続されます。これは通常、コンテナ化されたアプリケーションを起動するオーケストレーションプロセスの一環として実行されます。

このオーケストレーションプロセスは、さまざまなサービスのアドレスを動的に割り当て、互いに使用する異なるサービスに対してサービス/名前解決機能を提供します。ここで、Kubernetes が登場します。

Kubernetes は、アプリケーションコンポジション、サービスの依存関係、サービススケールの要件、サービス可用性の要件などの記述に必要な機能とインストルメンテーションを備えており、最も一般的なコンテナオーケストレーションシステムとなっています。

Kubernetes はまた、アプリケーション全体の可用性を損なうことなく、それぞれのサービスのライフサイクル、スケーラビリティ、可用性、パフォーマンスを個別に管理するために必要なツールも提供しています。

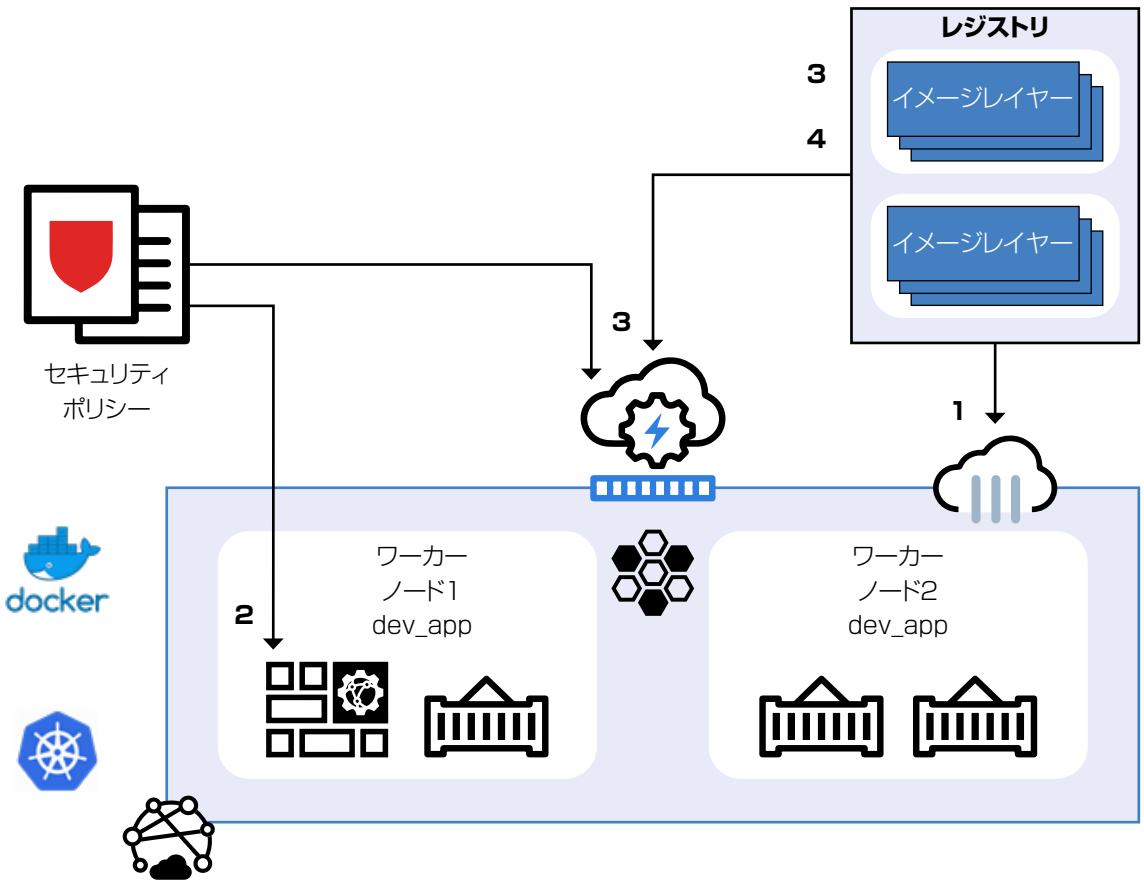

図1 :アプリケーションライフサイクル全体でのアプリケーションコンテナの保護。

アプリケーションコンテナのライフサイクル全体を対象とする、Fortinet のセキュリティ

ビジネスでアプリケーションをデプロイ、使用する方法を変革したコンテナの人気は急速に高まりました。その使用率は増加し続けるとされており、IT リーダーの86% が、より多くのアプリケーションでのコンテナの使用を優先しています[2] が、セキュリティが導入時の最大の懸念事項のひとつとなっています。事実、最近の調査によれば、組織の94% が、Kubernetes およびコンテナ環境で少なくとも一度のセキュリティインシデントを過去12 か月に経験したと回答しています。[3]

そのダイナミックな性質上、コンテナ化されたアプリケーションを保護することは簡単ではありません。従来型のセキュリティツールには、コンテナワークロードを保護するための本格的な機能がありません。

Fortinet のコンテナセキュリティソリューションは、次のような形でこのような課題を解決します：

1. **コンテナを認識できるセキュリティ：FortiGate（NGFW）**は、コンテナ管理レイヤーと効果的に接続して、さまざまなコンテナのラベルを学習します。セキュリティポリシーはラベルを認識することが可能であり、セキュリティポリシーでこのラベルを使用してオブジェクトを記述できます。このソリューションの主な役割は、コンテナインフラストラクチャを出入りするトラフィックを保護すること、つまり、north-south のセキュリティを担うことです。FortiGate NGFW には、ネイティブKubernetes、AWS EKS、GCP GKE、Azure AKS、OCI OKE などの大手のコンテナオーケストレーションシステムとのインターフェースとなるファブリックコネクタが用意されており、メタデータをセキュリティポリシーオブジェクトとして活用できます。トラフィックがコンテナ化された環境の境界を離れると、そのトラフィックは、コンテナの役割に基づいてポリシーを適用するFortiGate NGFW を通過します。またFortiGate は、FortiSandbox の統合を介して侵入防止システムと高度なマルウェア保護機能を使用し、内向きおよび外向きのコンテナトラフィックをスキャンして、脆弱性やファイルベースの脅威を検出します。
2. **コンテナに対応したセキュリティ：FortiWeb（WAF）**を、アプリケーションチェーンに組み込めるコンテナイメージとして活用することもできます。ウェブサービスベースのアプリケーションでマイクロサービスを作成することはきわめて一般的であるため、ウェブアプリケーションとアプリケーションプログラミングインターフェース（API）の保護をマイクロサービスベースのアプリケーションと組み合わせる機能は、そのようなアプリケーションを構築する組織にとっての大きな利点となります。開発者は、アプリケーション開発ライフサイクルにセキュリティ管理を展開して、アプリケーションライフサイクル全体を通じて、アプリケーションセキュリティを他のアプリケーションサービスとともにさまざまな環境に組み込めます。FortiWeb は現在、ネイティブDocker コンテナとして、また、AWS EKS マーケットプレイスのサービスとして提供されています。FortiWeb はFortiSandbox とも統合できるため、送受信されるトラフィックを対象としたゼロデイ脅威保護の追加のレイヤーを組み込めます。
3. **コンテナ統合セキュリティ：FortiCNP** で既知の脆弱性がビルドプロセスに入り込むことを防止することで、ソフトウェア開発ライフサイクル全体にわたりセキュリティを組み込めます。デベロッパーツールチェーンとの統合により、継続的インテグレーション/ 継続的デプロイメント（CI/CD）パイプラインが自動化およびビルドされます。FortiCWP Container Guardian は、CIS Kubernetes ベンチマークポリシーを使用してセキュリティガバナンスを推進し、安全ではないワークロードがデプロイされることを防止し、リスクポスチャを継続的にモニタリングして、進化する脆弱性を特定します。コンテナが実行されている間、コンテナベースの内部アプリケーションのトラフィックのほとんどはコンテナホスト内部で発生し、ネットワークインフラストラクチャからは確認できません。各トラフィックフローが検査されるようにするには、サービス間のトラフィックフローを、アプリケーションまたはサービス挿入メカニズム内で修正する必要があります。FortiCNP では、各 Kubernetes クラスタのコンポーネント、および、コンテナ間のトラフィック接続に関する分析情報を取得できます。
4. **コンテナレジストリのセキュリティ：**コンテナイメージは通常、レジストリと呼ばれる公開リポジトリに保管されます。新しいコンテナイメージのレジストリへの公開に関する制限はほとんどありません。そのため、アプリケーション開発者がレジストリから簡単に「プル」できる、悪意のあるコードが意図的または誤って組み込まれたコンテナイメージが作成されてしまうことがよくあります。このような状況になると、アプリケーション開発プロセスが不必要なリスクにさらされることになります。FortiCNP は、コンテナイメージがビルドサイクルに統合される前に、その脆弱性をスキャンします。**FortiSandbox** には、コンテナベースアプリケーションの開発者のニーズに特に対応するための、API や統合機能が用意されています。これにより、アジャイル開発の手法で導入される潜在的なリスクを低減できます。

セキュアかつ包括的なコンテナ戦略を実現する

コンテナテクノロジーは、アプリケーションインフラストラクチャや開発テクノロジーとして急速に普及しています。

ただ、これに伴い生じるリスクには、従来型のセキュリティツールでは対応できません。パブリックまたはプライベートクラウド環境で何かしらのアプリケーションをコンテナインフラストラクチャにデプロイする組織にとって、さまざまなコンテナオーケストレーションシステムとの互換性を持つ、包括的なコンテナセキュリティソリューションを用意できる能力が重要になります。

Fortinet のコンテナセキュリティソリューションであれば、拡大する攻撃対象領域に完全な形で対処できます。セキュリティをコンテナアプリケーションライフサイクルに組み込み、安全性の高いアプリケーションを提供することが可能となります。

[1] Michael Warrilow, "Forecast Analysis: Container Management (Software and Services), Worldwide," Gartner、2020 年5 月29 日

[2] "Cloud Container Adoption In The Enterprise," Forrester,2020 年6 月

[3] "State of Kubernetes Security Report," Red Hat、2021 年

フォーティネットジャパン合同会社

〒106-0032
東京都港区六本木7-7-7 Tri-Seven Roppongi 9階
www.fortinet.com/jp/contact

お問い合わせ