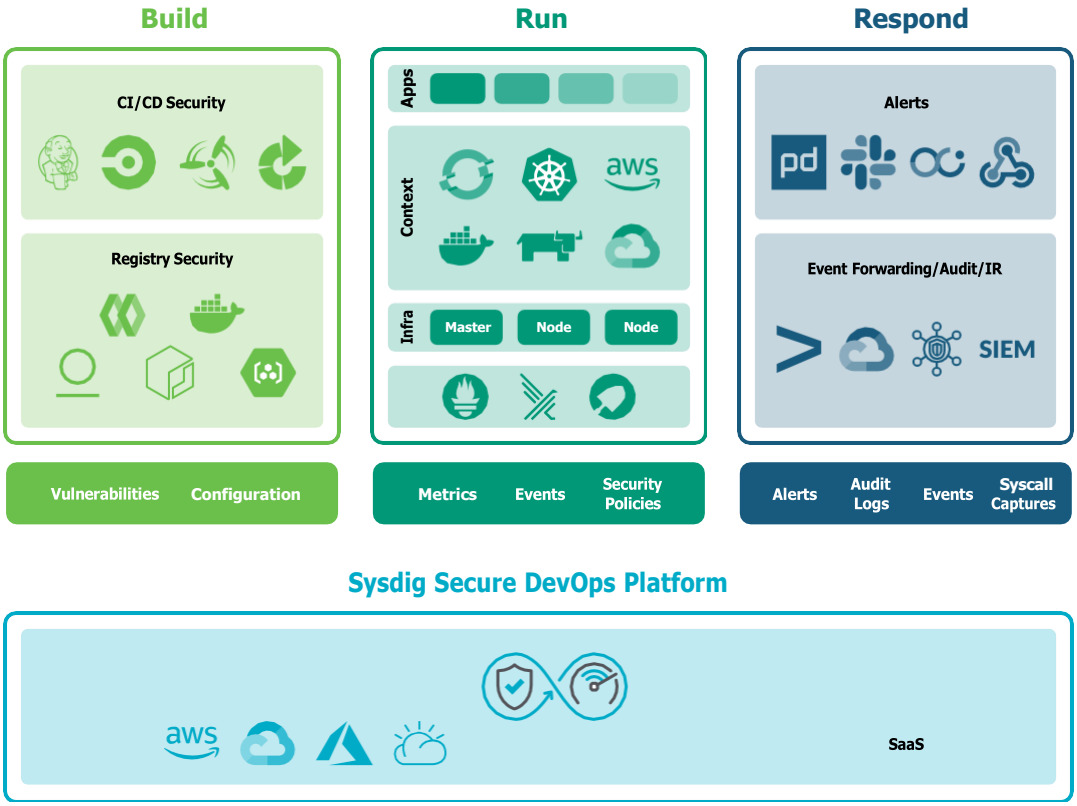
Sysdig Platform Architecture Guide

Contents

Sysdig Secure DevOps Platform

クラウドとコンテナがアプリケーション展開の標準になるにつれて、DevOpsの実践が不可欠になります。 クラウドチームは、アプリケーションのセキュリティ、コンプライアンス、パフォーマンス、および可用性について責任を負います。 チームには、展開を加速し、本番環境でクラウドアプリケーションに自信を持って実行するために、安全なDevOpsワークフローをサポートするツールが必要です。

クラウドネイティブなエコシステムへの統合



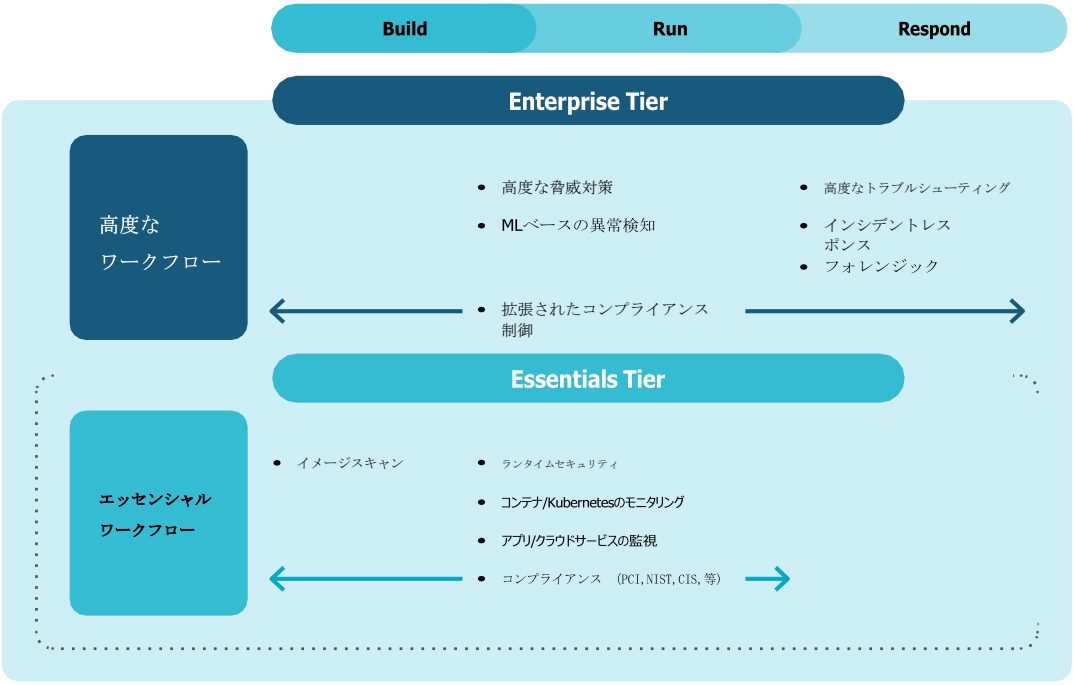
Sysdig Secure DevOps Platformは、セキュリティ、コンプライアンス、モニタリングをDevOpsのワークフローに組み込みます。コンテナのセキュリティとモニタリングを統合した唯一のものです。Sysdigは、単一の情報源を持つことで、開発、DevOps、セキュリティチーム間の情報のサイロ化を解消します。この統一されたデータプラットフォームにより、DevOpsチームはインシデントを正確にトリアージし、インシデントの原因が設定ミスなのか悪意のある試みなのかを迅速に判断し、コンテナがなくなった後でもフォレンジックを行うことができます。お客様のDevOpsチームは、特定のネームスペースやクラスタの実行中のイメージに影響を与える脆弱性を報告することができます。たとえば、新しいCVEが登場した場合、Sysdig Secure DevOps Platformは、特定のパブリッククラウドのリージョン、ネームスペース、クラスタなどで影響を受けたイメージと、その修正プログラムを所有するチームを迅速

に特定するのに役立ちます。このアプローチにより、組織は粒度の高いシステムデータを自動的に分析することで、問題を迅速に解決することができます。

クラウドやKubernetesのコンテキストに自動的に関連付けられた粒度の高いシステムデータを分析することで、問題を迅速に解決することができます。

Sysdigは、信頼性と安全性の高いクラウドアプリケーションの提供を支援し、マルチクラウドの本番環境でKubernetesやコンテナを実行する際に不可欠なユースケースに対応します。

クラウドネイティブなエコシステムへの統合



ユニークなContainerVisionとサービスビジョン

- **ContainerVision** - Sysdig独自のインストルメンテーションにより、侵襲的なインストルメンテーションを行うことなく、コンテナ、ネットワーク、アプリケーション、およびシステムのアクティビティを深く可視化することができます。この詳細なシステムコールデータにアクセスすることで、インシデント対応とトラブルシューティングの両方が加速し、DevOpsチームが根本原因を特定して問題を迅速に解決できるようになります。
- **ImageVision –** CI/CDパイプラインやレジストリのスキャンを自動化、レジストリのスキャンをインラインで実装することで、脆弱性や設定ミスを特定します。本番前に脆弱性をブロックし、ランタイムに新しいCVEをモニタリングします。重要な脆弱性をアプリケーションや開発チームにマッピングすることができます。
- **CloudVision -** クラウドログを使用したクラウドアクティビティの統合ビューを可能にします。AWS CloudTrailのようなクラウドのアクティビティログを、ランタイムの脅威検知に使用できるようにします。FalcoでCloudTrailログを分析することで、AWSのユーザーパーミッション、S3バケット、アクセスキーなどの変更を警告します。イベント監査を可能にし、脅威となるイベントを検出し、通知を上げることで、それらのセキュリティイベントに迅速に対処することができます。
- **ServiceVision -** ContainerVisionで収集したすべてのデータに対して、Kubernetesやクラウドサービスから提供され

るメタデータを使用してリッチなコンテキストを提供します。ダッシュボード、メトリクス、イベントのドリルダウン表示により、ネームスペース、デプロイメント、ポッドなど、Kubernetesの論理オブジェクト全体のセキュリティステータスを柔軟に表示できます。これにより、どのチームが脆弱性やアラートの解決に責任を持っているかを迅速に特定し、そのチームに関連するデータに集中することができます。

DevOpsパイプラインに容易に統合できる、プラットフォームに依存しないセキュリティおよびモニタリング機能

- **プラットフォームに依存しない** - クラウドやインフラを問わず、複数のクラスタのモニタリング、セキュリティ、トラブルシューティング、フォレンジックを統合することで、一貫したビューを実現します。
- **DevOpsパイプラインの統合** - 現在使用しているツールとすぐに統合できるため、ライフサイクル全体で時間を節約できます。
- **共通のデータプラットフォーム** - セキュリティとモニタリングの両方のワークフローの基盤となるのは同じデータで、共通のエージェントによって収集され、共通のバックエンドで管理されます。

導入・拡張が容易

- **SaaSファーストのデリバリーモデル** - 数分で立ち上げ、オンプレミスのソフトウェアを管理する手間を省くことができます。
- **ガイド付きの導入と使用** - 洗練されたワークフローとカスタマイズ可能なダッシュボードにより、すぐに価値を提供し、導入を促進します。
- **規模**- Sysdigは、パフォーマンスと安定性を損なうことなく、世界最大規模のクラウドのデプロイメントをサポートします。
- **チームによる隔離** - サービス、アプリケーション、インフラごとに特定のユーザーグループを定義できるため、DevOpsチームは、異なるユーザーロールをサポートするための排他的なパーミッションを与え、データアクセスを隔離して保護することができます。

投資保護のためのオープンソース標準に基づいて構築

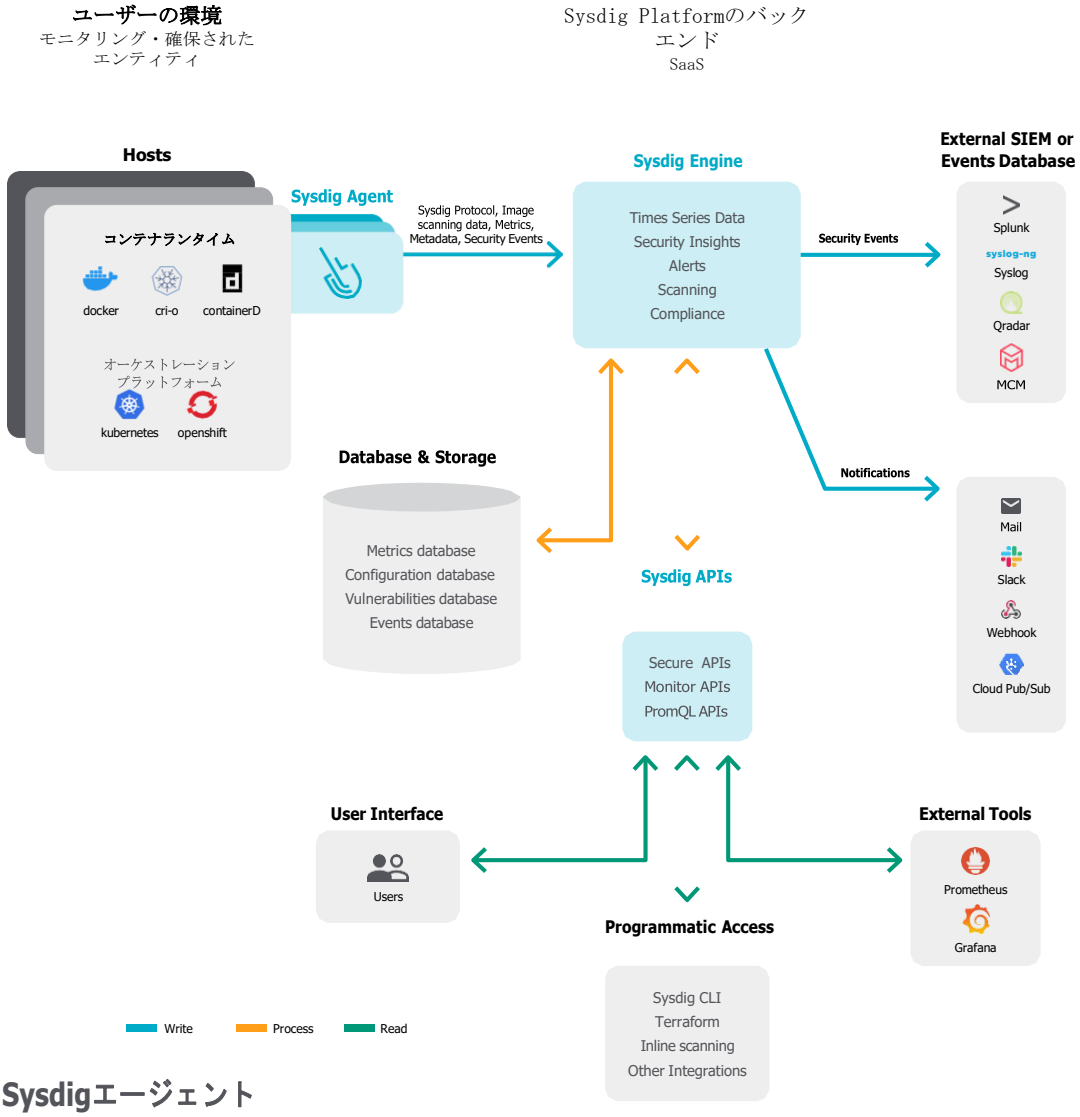
- **Prometheusとの完全な互換性** -Sysdigは、Prometheusメトリクスの規模と長期的なメトリクス保持を実現します。Prometheus Query Language (PromQL)のサポートを含む完全な互換性を維持することで、開発者はPrometheusへの投資を活かし、既存のダッシュボードやスクリプトを継続して使用することができます。
- **Falcoランタイム脅威検知** - Falcoオープンソースプロジェクトは、ランタイム脅威検知とコンプライアンス検証のためのルールエンジンを提供します。
- **Sysdigオープンソースプロジェクト** - Sysdigオープンソースプロジェクトは、1,000万以上のダウンロード数を誇る広く知られたプロジェクトで、低レベルの問題のトラブルシューティングにコミュニティで使用されています。
- **Anchore engine** - オープンソースのコンテナイメージスキャンツール「Anchore engine」は、コンテナイメージに存在するパッケージやサードパーティのライブラリを分析し、既知のソフトウェアの脆弱性を発見し、コンテンツやライセンスに関するレポートを作成します。

本稿では、Sysdigがどのように、どこで、なぜ、クラウドネイティブアプリケーションへの移行を加速・容易にし、ソフトウェア開発ライフサイクル全体に継続的なメリットをもたらすのかを説明する枠組みを提供します。

また、この論文では、当社のKubernetesの可視化およびセキュリティプラットフォ

ームのアーキテクチャを深く掘り下げます。その仕組みを理解するだけでなく、Sysdig独自のアプローチがお客様のクラウドネイティブな旅を効果的にサポートする方法を理解するのに役立ちます。

Sysdig Platformの主な要素



Sysdigエージェント

Sysdigは、対象となるすべてのエンティティからモニタリングおよびセキュリティ情報を収集します。これを実現するためには、各ホストに1つのSysdigエージェントを配備する必要があります。ホストは以下のようなものです。

- KubernetesやOpenShiftのクラスタを構成するノード
- 仮想マシンまたはベアメタル
- お客様のデータセンターのオンプレミスに常駐

Sysdigエージェントは、Helmチャートや Kubernetesオペレーターなどを使って、コンテナそのものとしてインストールすることができます。エージェントをインストールするプロセスは、プラットフォームのオンボーディングインストラクションで詳細に説明されている手順に従って完全に自動化されています。

エージェントがホストにインストールされると、自動的に以下から情報収集を開始します。

- 実行中のコンテナ
- コンテナランタイム
- オーケストレーションAPI（Kubernetes、OpenShiftなど）
- 定義されたPrometheusのエンドポイントからのメトリクス
- 自動検出された JMX ソース
- StatsD
- アプリチェックによる統合
- カーネルヘッダやモジュール、extended Berkeley Packet Filter（eBPF）を用いたホスト自身の検出

Sysdigエージェントは、Sysdigバックエンドとの通信チャネルを維持し、モニタリングメトリクス、インフラメタデータ、セキュリティイベントを含むメッセージをカプセル化するために使用されます。このチャネルは、標準的なTLS暗号化を使用して保護され、バイナリメッセージを使用してデータを転送します。このチャネルを使用することで、エージェントはデータを送信するだけでなく、セキュリティランタイムポリシーやベンチマークなどの追加設定をバックエンドから受け取ることができます。

Sysdigエージェントのリソース消費量は、ホストのサイズと負荷に左右されます。最低でも、エージェントは総CPUの2%と512MiBのメモリを必要とします。

Sysdigバックエンド

Sysdigのバックエンドは、SaaS版を直接利用してSysdigが透過的に管理することもできますし、お客様の施設にインストールすることもできます。この違いは、後述するプラットフォームの実際の運用には影響しません。

エージェントメッセージがバックエンドで受信されると、それらは処理され、プラットフォームで利用可能なデータ（時系列、インフラとセキュリティイベント、インフラのメタデータ）に抽出されます。

Sysdigのバックエンドは、メトリクスの容量など個々のデータニーズの負荷に対応するために垂直方向に拡張できるだけでなく、何千ものエージェントのニーズや何百万ものメトリクスの取り込みなど、処理やスケーラビリティのために水平方向に拡張することができます。

Sysdig Platformの主なデータの流れ：

- エージェントからのメトリックデータの抽出と後処理により、必要なインフラのメタデータを含む完全な時系列データをユーザーが利用できるようにします。
- インフラのメタデータ（特にKubernetesの状態）を維持することで、すべてのイベントと時系列を充実させ、正しくグループ化することができます。
- 時系列データおよびイベントデーを保存します。
- 時系列データを処理してアラートトリガーを算出します。
- エージェントによってトリガーされたセキュリティイベントをキューに入れ、イベントフィードに表示したり、通知チャネルにプッシュしたり、イベントフォワーダを介してSplunk、Syslog、IBM MCM／Qradarなどのプラットフォームに転送したりします。
- コンテナプロファイルやセキュリティベンチマークの結果を生成するために使用されるコンテナフィンガープリントのような、その他のセキュリティデータを集約し、後処理を行うことができます。
- 後処理されたデータはSysdig Platformの内部データベースに保存され、APIサービスによってダッシュボード、イベントフィード、脆弱性レポート、セキュリティベンチマークなどを作成することができます。

Sysdig API

Sysdig Platformでは、内部データを利用、表示するためのいくつかの方法を提供しています。すべてのAPIはRESTfulで、HTTP JSONベース、TLSで保護されています。Sysdigのフロントエンドや**sdc-cli**などのAPIクライアントも同じAPIを使用しています。

これらのAPIは、以下のようなユースケースを可能にします。

- ユーザーはSysdigのユーザーインターフェースを使ってプラットフォームにアクセスします。
- プログラムによるデータの入力や抽出など
- ユーザーの自動作成
- 設定状態を保存または回復するためのTerraformスクリプト
- CI/CDパイプラインからスキャン結果をプッシュするインラインスキャン
- sdc-cliを使用したインストルメンテーション
- GrafanaなどのPromQL互換ソリューションに接続するために使用できるPromQL APIインターフェース

Sysdigのアーキテクチャ：オープンソースをベースに構築

Sysdig Secure DevOps Platformは、本番環境でコンテナを効果的に運用することは、基本的にデータの問題であるという考えからスタートしています。私たちは当初から、複雑で分散したエフェメラルなコンテナ環境を根本的に異なるアプローチで可視化することができれば、モニタリング、ランタイムセキュリティ、脆弱性管理、インシデント対応など、多くの運用上の課題を長期的に解決できると考えてきました。

同時に、最初はDocker、現在はKubernetesに牽引され、クラウドネイティブの動きはオープンソースのビルディングブロックに大きく根ざしています。このコミュニティの努力により、企業はほとんどコストをかけずに始めることができ、主要なプロジェクトを常に改善する巨大なコミュニティを提供しています。Sysdigも例外ではありません。

Sysdigは、クラウドネイティブエコシステムにおいて、オープンソースツールは不可欠な要素であると考えています。開発者がクラウドネイティブ戦略を考える際に、実験やツールの使用のために前払いを要求されるべきではありません。その代わりに、オープンソースを通じて主要なビルディングブロックにアクセスできるようにすべきです。Sysdigは、同名のオープンソースのLinuxトラブルシューティングフォレンジックプロジェクト（sysdig）をベースにしています。このオープンソースのプロジェクトでは、1つのホスト上のすべてのシステムコールを、プロセス、引数、ペイロード、接続に至るまで確認することができます。これから説明するように、このデータはコンテナ、マイクロサービス、クラウド、オーケストレータに動的にマッピングされます。

このデータをコンテナ、マイクロサービス、クラウド、オーケストレータに動的にマッピングすることで、強力かつシンプルに利用できるようになっています（商用サポートされているSysdig Platform）。

透明性が必要になるインストルメンテーション

静的環境や仮想環境では、ホスト上でエージェントを実行し、関連するアプリケーションに基づいてエージェントを設定することが簡単でした。しかし、コンテナ環境では、この方法は以下で通用しません。

- コンテナの重要な価値である「シンプルさ」を損なわずに、各コンテナ内にエージェントを配置することはできません。
- アプリケーションとコンテナが行き来する中で、アプリレベルの関連メトリクスを収集するためにエージェントプラグインを手動で設定することはできません。また、サービスごとのセキュリティポリシーを手動で設定することもできません。

このような新たな要求に応えるためには、インストルメンテーションの行為が可能な限り透過的であり、人間の介入を可能な限り少なくする必要があります。対象となるシステム上のイベントやアクション、インフラメトリクス、アプリケーションメトリクス、サービスレスポンスタイム、カスタムメトリクス、リソース／ネットワーク使用率などを、コンテナ内で何の努力もせずに取り込むことができなければなりません。また、コンテナを追加するたびに作業を行う必要もありません。

これを実現するには、2つのアプローチが考えられます。まず、Kubernetesが生み出した概念であるポッドです。ポッドとは、共通のネームスペースを共有するコンテナのグループです。そのため、ポッド内のコンテナは、同じポッド内の他のコンテナが何をしているかを見ることができます。インストルメンテーションエージェントでは、これを「サイドカー」コンテナと呼ぶことが多いです。

これは、Kubernetesでは比較的簡単にできることだと思います。

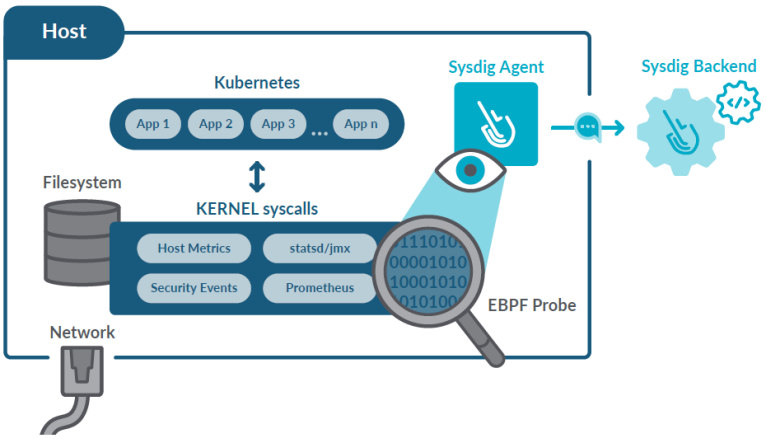
しかし、デメリットは以下の気になるところです。

- 1台のマシンに多くのポッドを配置すると、リソースの消費量が多くなります。これは、プロセスごとにモニタリングエージェントを配置するようなものです。
- また、ポッドには依存関係があり、攻撃対象も増えます。つまり、モニタリング用のサイドカーにパフォーマンスや安定性、セキュリティの問題があると、アプリケーションに大打撃を与える可能性があるということです。
- サイドカー方式の別のバリエーションとして、ld_preload 方式があります。これは、すべてのコンテナにエージェントを自動的にインストールするのと同じように、コンテナにバイナリを注入します。この方法は、エージェントベースとサイドカーベースのインストルメンテーションの両方の欠点を持ち、コンテナの制御されていない／テストされていない変更による安定性の問題をしばしば引き起こします。

2つ目のモデルは、ホスト単位の透明性のあるインストルメンテーションです。アプリケーション、コンテナ、カスタムメトリクス（Prometheus、StatsD、JMX）、ホストメトリクスのすべてを1つのインストルメンテーションポイントで捕捉し、ホストごとのコンテナに送って処理、転送するものです。これにより、多くの人が頼りにしている、すべてをカスタムメトリクスにする必要がなくなりました。

サイドカーモデルとは異なり、ホスト毎のエージェントは、モニタリングエージェントのリソース消費を大幅に削減し、アプリケーションコードの修正も必要ありません。ただし、特権的なコンテナとカーネルモジュールが必要になります。

Sysdigは後者を選択しました。私たちは、この技術的なアプローチを「ContainerVision」と呼んでいます。このモデルを選択することは、私たちが行った最も重要な設計上の決定の一つであることがわかります。



Sysdigエージェントの構築は複雑になりますが、このアプローチにはいくつかの利点があると考えました。

- セキュリティ、モニタリング、トラブルシューティング、フォレンジックなど、本番システムに必要な多くの共通運用活動を統合します。
- 少ないリソースでより多くのデータを収集します。
- 小規模でアジャイルなチームの運用オーバーヘッドを削除します。

また、私たちのインストルメンテーション手法は、複雑なネットワークや高密度のエージェントが存在する環境に対する脅威を軽減するものであると確信していました。懸念を軽減するために、私たちはSysdig Linuxとコンテナ可視化コマンドラインツールの一部としてカーネルモジュールをオープンソース化しました。

ContainerVisionの詳細

ContainerVisionは、Sysdigのアプローチを異なるものにしている中核的な要素です。私たちのアーキテクチャは、tcpdumpやWiresharkと非常によく似ています。これは偶然ではありません。SysdigはWiresharkの共同開発者の一人によって作られました。まず、イベントは、sysdig-probeと呼ばれる小さなドライバによってカーネル内で捕捉されます。このドライバは、tracepointsと呼ばれるカーネルの機能を利用します。トラップポイントは、カーネル内の特定の関数から呼び出される「handler」をインストールすることを可能にします。

現在、Sysdigでは、プロセスのスケジューリングイベントだけでなく、入出力時のシステムコールに対してもtracepointsを登録しています。Sysdig-probeのhandlerは、イベントの詳細を共有の読み取り専用リングバッファにコピーし、後で消費できるようにエンコードするだけです。handlerをシンプルに保つ理由は、想像できると思いますが、パフォーマンスです。なぜなら、元のカーネルの実行は、handlerが戻るまで「frozen」するからです。このfrozenはナノ秒のオーダーで、それがドライバのすべてです。残りのマジックはユーザーレベルで行われます。

イベントバッファはユーザースペースにメモリマップされているので、コピーなしでアクセスでき、CPU使用率とキャッシュミスを最小限に抑えることができます。libscapとlibsinspという2つのライブラリが、イベントの読み取り、デコード、解析をサポートします。具体的には、libscapはトレースファイル管理機能を提供し、libsinspは洗練された状態追跡機能（たとえば、FD番号の代わりにファイル名を使用することができる）のほか、フィルタリング、イベントデコード、プラグインを実行するためのLua JITコンパイラなどがあります。

最後に、オープンソースのSysdigは、これらのライブラリのシンプルなラッパーとしてトップを飾っています。一方、商用プラットフォーム向けのエージェントは、同じコアを使用していますが、同時に(a)データをバックエンドに転送し、(b)セキュリティポリシーを実施します。

eBPF（extended Berkeley Packet Filter）は、Linuxネイティブのカーネル内仮想マシンで、アプリケーションのパフォーマンスやイベントの観察・分析のために、安全でオーバーヘッドの少ないトレースを可能にします。

ContainerVisionを eBPF に接続する動機はいくつかあります。一つは、既にベースオペレーティングシステムの一部となっている成熟した技術を利用することです。これにより、オブザーバビリティの管理が非常に簡単で摩擦のないものになります。

eBPFが理にかなっているもう一つの理由は、コンテナに最適化されたオペレーティングシステムの出現です。Google Cloud PlatformのCOS（Container-Optimized OS）やAWS Bottlerocketのようなソリューションは、カーネルモジュールを完全に排除した不変的なインフラストラクチャアプローチを特徴としています。eBPFを利用することで、Sysdigが以前から提供しているカーネルモジュールと同じレベルのコンテナオブザーバビリティを実現することができます。
ContainerVisionは、Sysdigのアーキテクチャの重要な利点です。Sysdigのインストルメンテーションのトレースオーバーヘッドは非常に予測しやすく、本番環境での実行に適しています。これについてもう少し詳しく知りたい方は、**DTrace vs strace vs sysdig：A technical discussion**をご覧ください。
Sysdigオープンソーストラブルシューティングツールは、1つのホストについてContainerVisionで収集されたすべてのリッチなデータに対して、コマンドラインインターフェースとcursesベースのインターフェースの両方を提供します。
モニタリングアプリケーションでは、これらの同じシステムコールを使用して、スタックの上下に関連するメトリクスを抽出し、分散環境全体でhtopのようなインターフェースを作成します。また、セキュリティアプリケーションでは、同じエージェント、同じデータ、同じバックエンドを使用しますが、セキュリティ違反を表すイベントに焦点を当てます。
しかし、私たちは、コンテナを扱う場合、パフォーマンス指標やセキュリティイベントだけでは十分ではないと考えています。次は、そのすべてを理解できるかどうかを見てみましょう。

ServiceVisionがContainerVisionにリッチなコンテキストを追加

コンテナやマイクロサービスの環境では、Linuxのシステムコールは、コンテナ、Kubernetes、ホスト、アプリケーションの健全性に関する豊富なデータ源となります。システムカーネルトレースを利用することで、Sysdigエージェントは、カーディナリティや精度を損なうことなく、すべてのプロセス、すべてのコンテナ、すべてのサービス、すべてのアクション、すべてのメトリクスを独自の多次元的な方法で把握することができます。Sysdigエージェントは、Prometheus、statsD、コンテナ、オーケストレータのイベントなど、追加のデータソースを自動検出し、自動診断することができます。この豊富なデータソースは、開発から本番までのライフサイクル全体を通して、コンテナ化されたアプリケーションの健全性とセキュリティに関する独自の洞察を提供します。そのため、悪意のあるプレイヤーや、一般的な設定ミスによって生じる脆弱性から、コンテナ化されたアプリケーションを保護することができます。

きめ細かいシステムコールデータは、商用でサポートされているSysdig Platformですぐに強力で簡単に使用できる方法で、コンテナ、マイクロサービス、クラウド、オーケストレータに動的にマッピングされます。優れたメトリクスの豊富なセットがあっても、関連性がなければ意味がありません。**ServiceVision**では、粒度の高いデータをコンテナオーケストレータにマッピングして、システムの構築方法を理解することができます。ここでの主な利点は、グループの設計やサービスのレイアウトに時間を費やす必要がないことです。これはKubernetesですでに行われていることです。また、データやビューを探す際に、見慣れたキーや名前を参照することができます。何百万ものデータの中から、関連性のあるいくつかのメトリクスを素早く見つけ出すこのユニークな機能は、複雑なKubernetesを大規模に管理するために

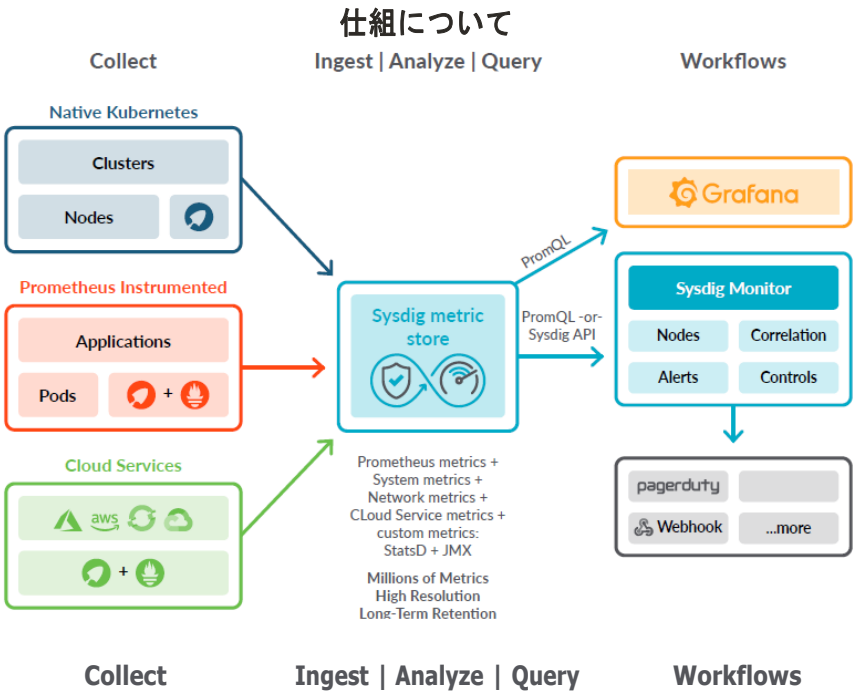
Sysdigが選ばれる主な理由の1つです。 このメタデータを取得してビューに適用する機能は、データを様々なユーザーに関連付けるための非常にシンプルな方法です。何千、何百万もの異なるオブジェクト（セキュリティイベント、コンテナ、メトリクスなど）をユーザーに見てもらう代わりに、ユーザーに関連のあるものだけをフォーカスして表示することができます。これにより、時間を節約し、健全な状態のスナップショットを簡単に提供し、根本原因の分析を加速します（つまり、修正までの時間を短縮します）。私たちは、このメタデータの収集と適用の機能をServiceVisionと呼んでいます。

Kubernetes Nativeのモニタリングとセキュリティ

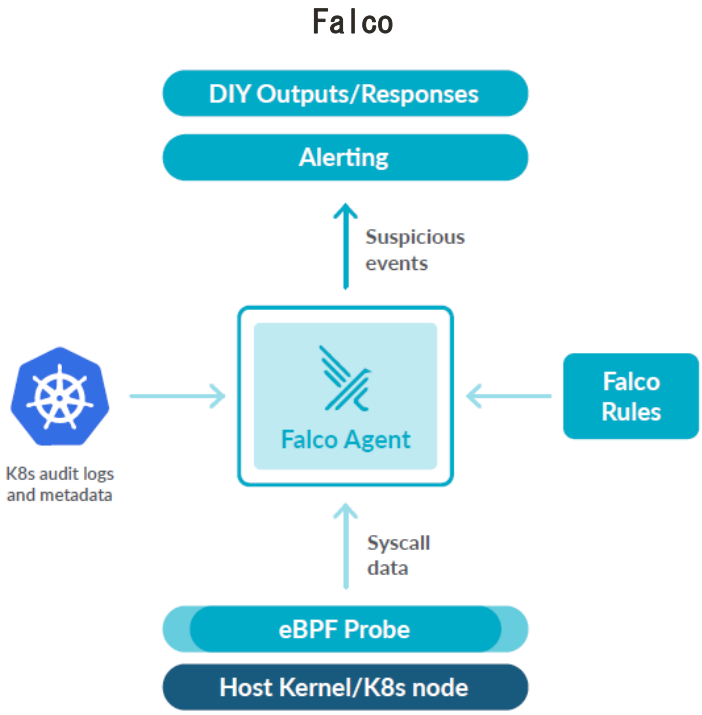
先ほど強調したServiceVisionは、Kubernetesとの統合の多くで核となるものです。手動でグループ分けをして、それぞれに個別にポリシーを適用する必要はありません。Sysdigでは、すでに構築されたものを利用して、今日デプロイしたものだけでなく、明日デプロイするものにも適応する柔軟な方法でポリシーやルールを適用することができます。ポリシーに「production」とタグ付けすると、このキーを使用する新しいワークロードは自動的に継承されます。本番クラスタを保護すれば、そのクラスタで今後発生するすべてのことが保護されます。

アプリケーションとクラウドサービスのモニタリング

Prometheusモニタリングは、オープンソースコミュニティのもう一つの重要な構成要素であり、メトリクスフォーマットとアプリケーションからメトリクスをスクレイピングする方法を提供しています。SysdigがPrometheusを開発したわけではありませんが、私たちは積極的にプロジェクトに貢献し、その機能を私たちのプラットフォームに取り入れています。たとえば、標準的なPromQLクエリを使用して、Sysdigに保存されているメトリクスを表示、分析することができます。Sysdigは Prometheusコミュニティのためのウェブサイト**promcat.io**を作成しました。ここではユーザーがPrometheus用のアプリケーションモニタリングリソースを見つけることができ、Sysdig Platformや他のオープンソースツールですぐに使用することができます。これらのリソースは、献身的なエンジニアのチームによって収集され、積極的にテストされ、維持されています。



当初のプロジェクトで生み出された独自の可視性をさらに活用するために、私たちはFalcoというオープンソースのセキュリティツールを構築しました。Falcoは、オープンソースのSysdigの可視性と、ランタイムにポリシー違反がないかシステムイベントを常時モニタリングするルールエンジンを組み合わせたものです。私たちが提供するエンタープライズ製品は、この豊富なデータに基づいて、ポリシーの施行、コンプライアンス、モニタリングを可能にします。



ランタイム脅威検知のためのFalco

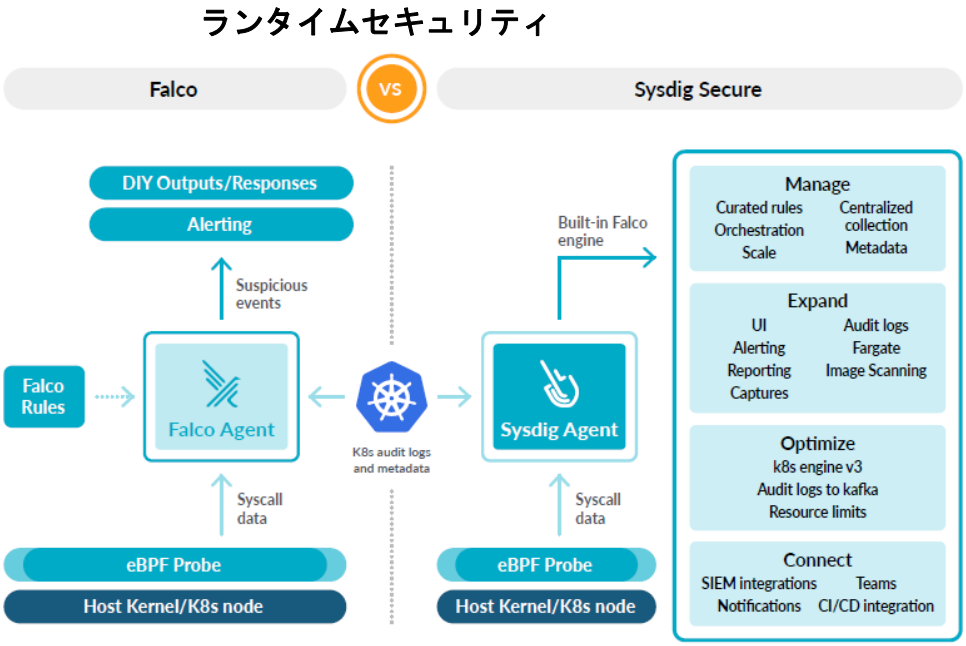
Falcoは、ランタイムセキュリティ検出のための標準的なCNCFプロジェクトとなりつつあり、すでに何千ものクラスタで広く採用されています。また、Falcoには、製品の開発や改良を支援するだけでなく、新しいルールをクラウドソースで提供したり、既存のルールを継続的に検証したりする広範なコミュニティがあります。Falcoのコミュニティは、（Falcoを最初に開発した）Sysdigのオープンソースチームに励まされており、コミュニティやCNCFと協力してツールの将来を推進し続けています。

Falcoのデータソースは、低レベルのシステムコール（ホストやコンテナレベルでのSSH接続の検出、データベースサーバの送信ネットワーク接続の検出、メッセージバスのプロセス分岐やコマンドラインスクリプトの実行など）と、Kubernetesのモニタリング API（ユーザーによる特権コンテナの起動、クリアテキストの認証情報を含むコンフィグマップ、機密性の高いホストファイルシステム領域がコンテナにマッピングされているなど）の両方です。

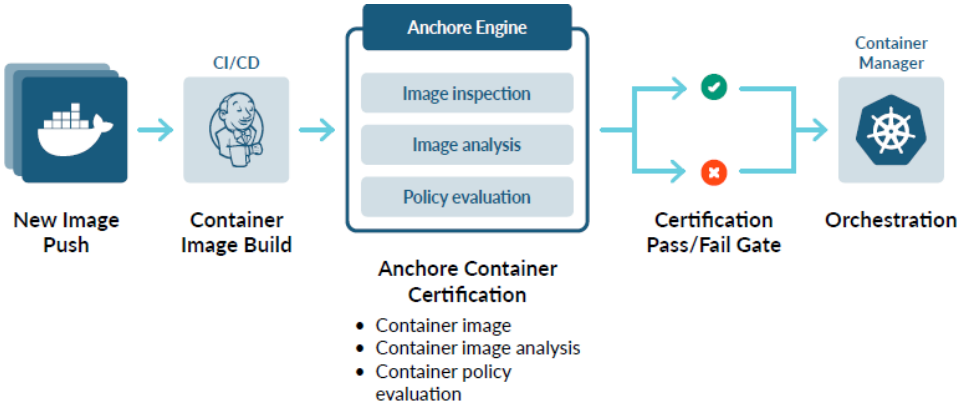
Falcoは、Sysdig Secure DevOps Platformに組み込まれており、ランタイムセキュリティの可視性を提供します。このプラットフォームでは、オープンソースの機能が拡張されており、シンプルなUI、簡単なルールエディタ、関連するルールを素早くフィルタリングするためのルールタグが用意されています。

ルールのタグ付けが可能で、Sysdigネイティブの Kubernetesインテグレーションに

よって拡張され、さまざまなKubernetesオブジェクトに迅速かつ柔軟にルールを適用することができます。Falcoのルールは、異常な活動を検出し、コンプライアンス上の問題点を指摘することができます。Sysdigは、Secure DevOps Platformで使用されるFalcoルールの完全な商用サポートを提供します。

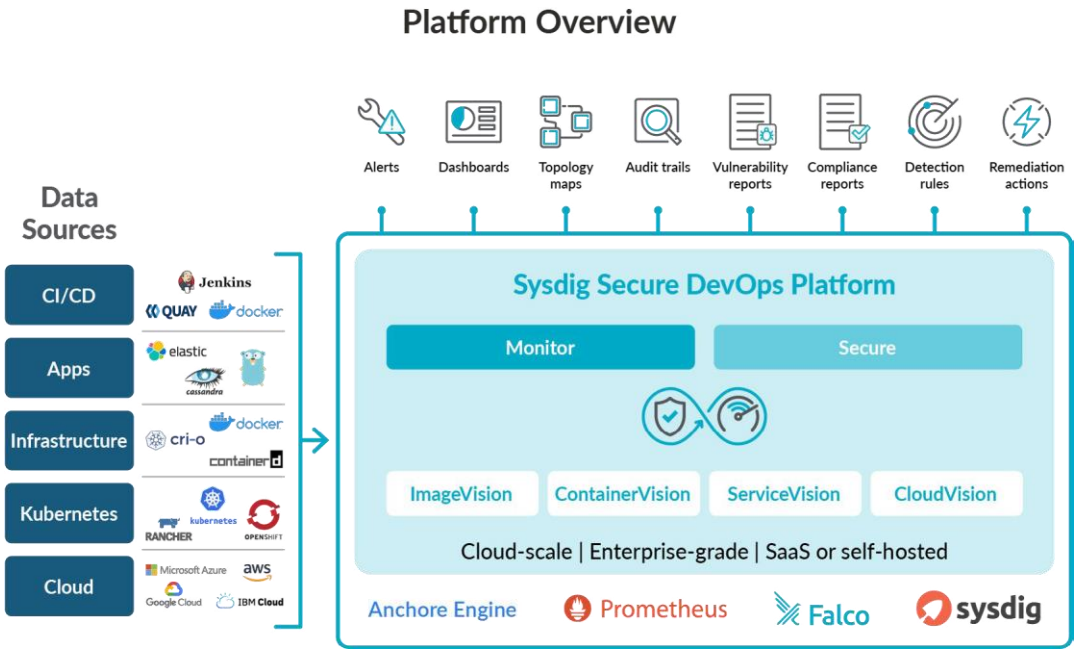


お客様の環境を保護するためのデータをさらに充実させるために、SysdigはAnchoreをプラットフォームに統合しました。Anchoreは、Falcoがランタイムに提供するものと同様に、ビルドタイムにも提供します。Anchoreを使用することで、脆弱性管理ポリシーを実装・実施し、コンテナイメージを本番前にスキャンすることができます。



Sysdig Secure DevOps Platformは、包括的で実用的な脆弱性情報のソースであるVulnDBと統合し、サードパーティのライブラリや依存関係にある脆弱性に関する豊富な知見を提供します。Sysdig Platformがチェックする広範な脆弱性データベースと組み合わせることで、**VulnDB**からの包括的なデータは、より効果的にセキュリティリスクを特定、追跡、低減することができます。

Sysdigの商用サービスは、お客様のすべての運用データを統合し、インサイトに変換します。Sysdigのプラットフォームは、すべてのアプリケーション、コンテナ、ホストの何千ものメトリクスとイベントを基に、データをリッチ化し、アプリケーションやマイクロサービスの正確なインコンテクストなビューを提供します。Sysdigは、お客様のワークフローの実現を支援するために、主要なビジュアライゼーションを提供するアプリケーションを提供します。



Sysdig Secure DevOps Platformは、お客様の環境全体のリスク、健全性、パフォーマンスを統合的に把握することができます。環境内で最も問題が多く、深刻度が高いサービスやコンポーネントをインテリジェントに表示するように設計されています。

- **Sysdig Monitor**は、コンテナ、アプリケーション、オーケストレーション、クラウドから収集した深いテレメトリデータを用いて、パフォーマンスとヘルスモニタリングを実現します。
- **Sysdig Secure**は、脆弱性管理、コンプライアンス、セキュリティポリシーの施行、モニタリング、ランタイムプロテクションを実現します。

これらの機能は、統一されたホストインストルメンテーション、統一されたデータバックエンド、強力でシンプルなユーザーインターフェースによって提供され、DevOps、DevSecOps、サービスオーナー、開発者の各チームは利用可能なすべてのリッチデータを活用することができます。ここでは、コンテナライフサイクルの各側面において、Sysdigがお客様のユースケースにどのような影響を与え、サポートしているかを理解していただくために、さらに詳しく説明します。

開発 – 構築：

- 開発者は、脆弱性のないアプリケーションを確実にプッシュする必要があり、 CI/CDパイプラインの早い段階でセキュリティスキャンを行う責任があります。
- Sysdigは、単一のワークフローで脆弱性や設定ミスの検出をサポートします。

操作 – 実行：

- Sysdigは、特定のネームスペースやクラスタなどで稼働中のイメージに影響を与える脆弱性をレポートする機能も提供しています。たとえば、新しいCVEが登場した場合、Sysdigは特定のAWSリージョン、ネームスペース、クラスタなどで影響を受けるイメージや、修正プログラムを所有するチームを迅速に特定することができます。
- Sysdigは、パフォーマンスに影響を与えることなく、実行時に脅威を予防・検出することができます。
- また、Sysdigは、クラウドアプリケーション、インフラをモニタリングし、可用性、パフォーマンスを最大限に高めることができます。

インシデントレスポンスとトラブルシューティング：

- お客様のDevOpsチームは、インシデントを正確にトリアージし、それが設定ミスなのか悪意のある試みなのかを迅速に判断する必要があります。
- Sysdigは、コンテナがなくなった後でも、インシデント対応やトラブルシューティング、フォレンジックを行うことができます。

コンプライアンスをエンドツーエンドで検証：

- コンテナがrootとして実行されているか、コンテナ内でsshが実行されていないかなど、構成がCISのベストプラクティスに適合しているかを確認します。
- NIST、PCI、SOC2などのアプリケーションコンプライアンスを確保します。
- Falcoルールを使用して、異常なアクティビティを検出し、コンプライアンス上の問題を指摘します。

これらのエリアに加えて、次はアーキテクチャをさらに掘り下げて、各パーツをより詳細に見ていきます。

Build: セキュアな開発を加速する

企業は、開発ライフサイクルの早い段階でセキュリティを組み込む必要があります。コンテナによって開発プロセスがどのように変化するかを考えると、コンテナ化されたアプリケーションやマイクロサービスベースのアプリケーションの開発における積極的なセキュリティ対策は、必要不可欠なものとなります。また、コードデリバリーが劇的に高速化していることを考えると、開発者がこれらの追加要件を満たすために、コードが本番に入る前にパフォーマンスやセキュリティの状態を可視化するためのリソースやツールの充実を期待するのは当然のことでしょう。Sysdigは、開発チームがより信頼性の高い、安全なコードをより効率的に提供するための重要なツールです。

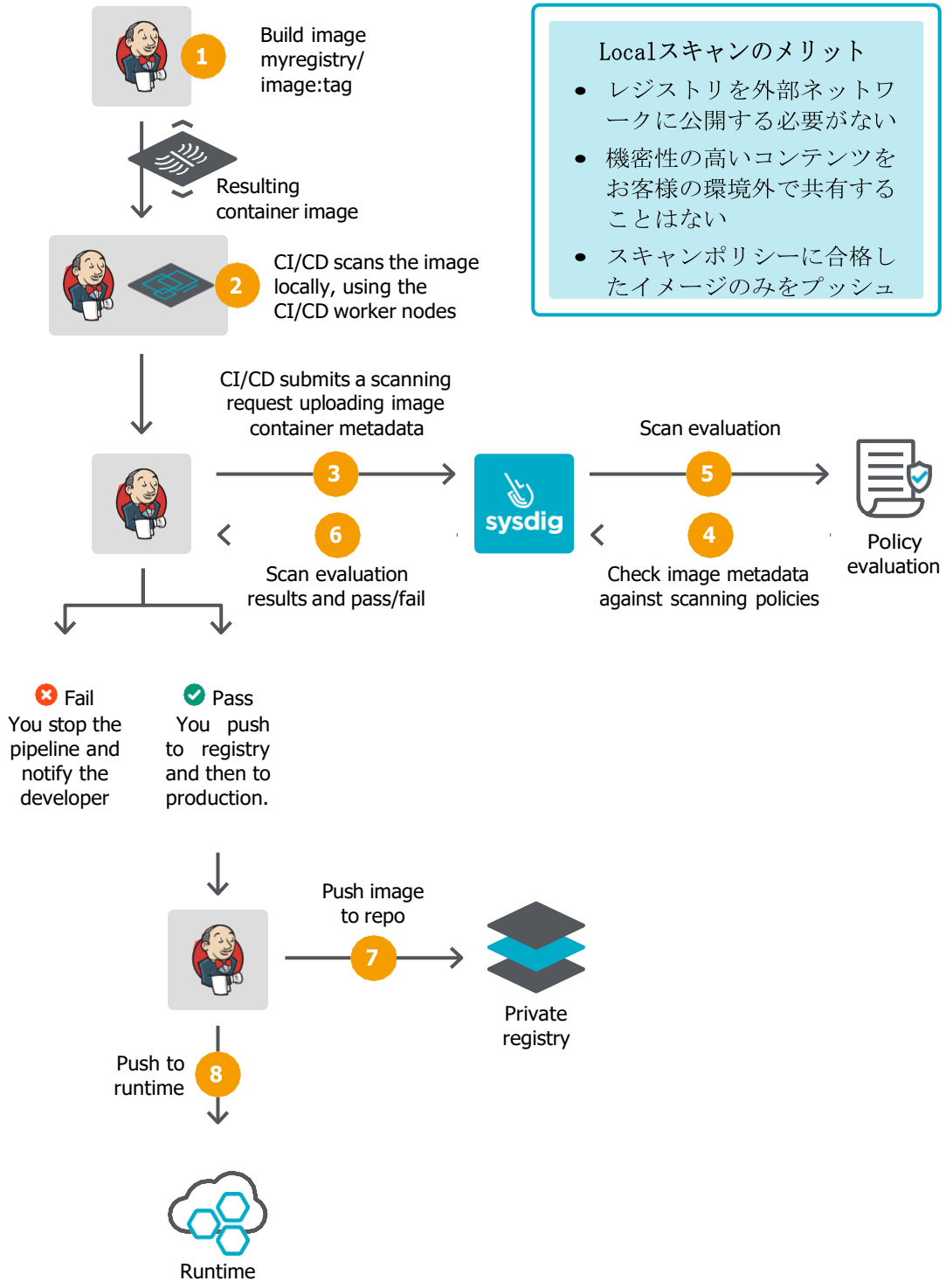
Shifting Security Left:
イメージスキャンと脆弱性管理を開発プロセスに導入

Sysdigセキュアの脆弱性管理機能は、アプリケーションのセキュリティ、コンプライアンス、品質を開発者の身近なものにします。ソフトウェアデリバリーチェーンの一般的なツールとのネイティブな統合により、Sysdigセキュアは、ビルドが完了する前やコンテナがデプロイされる前に、セキュリティ問題をスキャンし、ブロックし、修正することを可能にします。Sysdigセキュアは、コンテナイメージのスキャンから始まり、コンテナイメージの検査を行い、以下のようなコンテナイメージの内容の詳細な分析を生成します。

- OS公式パッケージ
- 非公式のOSパッケージ
- 設定ファイル
- 言語モジュール - NPM、PiP、GEM、およびJavaアーカイブ
- コンテナイメージのメタデータなど

Sysdigセキュアは、イメージの内容と脆弱性フィードを自動的に関連付けて、既知の脆弱性を持つパッケージやファイルを把握します。脆弱性フィードは、OSベンダー、パッケージリポジトリ、VulnDB、National Vulnerability Database (NVD)などから継続的に更新される脆弱性やパッケージのデータです。

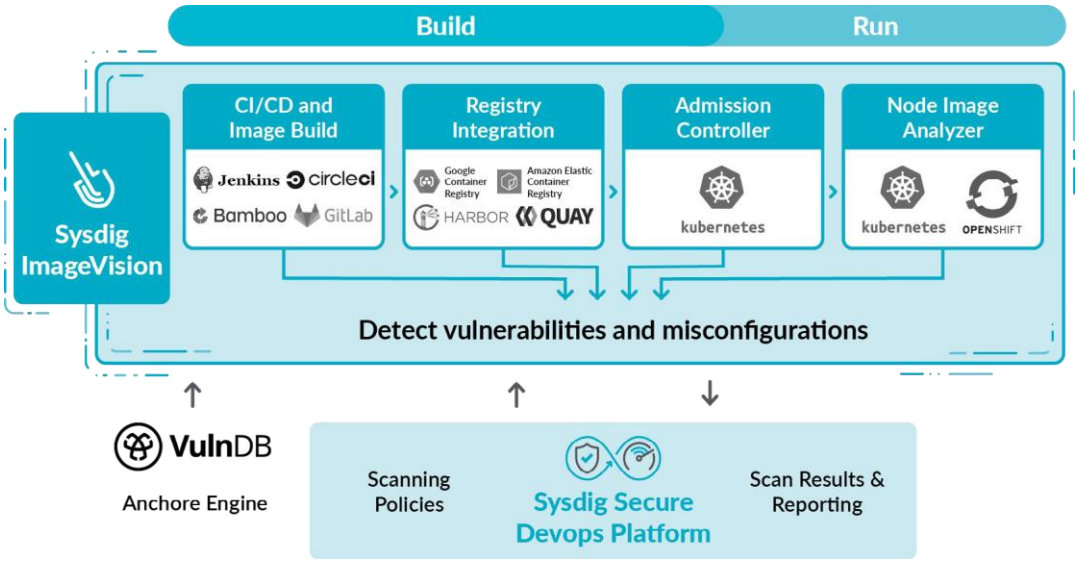
Inline Scanning



Localスキャンのメリット

- レジストリを外部ネットワークに公開する必要がない
- 機密性の高いコンテンツをお客様の環境外で共有することはない
- スキャンポリシーに合格したイメージのみをプッシュ

コンテナイメージのライフサイクル全体を通じたコンテナイメージスキャンとガバナンスの統合



Jenkinsのネイティブプラグインや**API**を使って、ビルドプロセスの一環として自動的にコンテナイメージをスキャンするよう**Sysdig**セキュアを簡単に設定できます。コンテナがビルドプロセスを通過するたびにイメージスキャンを行うことで、ビルドの失敗、警告の発動、コンプライアンスの強化を簡単に行うことができます。

Sysdigセキュアは、CoreOS Quay、Amazon ECR、Docker Private Registries、Google Container Registry、JFrog Artifactory、Microsoft ACR、SuSE Portus、VMware Harborなど、Docker V2と互換性のあるレジストリに保存されているイメージをスキャンします。

このソリューションでは、ビルドパイプラインやレジストリにポリシーを簡単に設定することができ、脆弱性、オペレーティングシステムパッケージ、サードパーティパッケージ、ソフトウェアライブラリ、**Dockerfile**チェック、ファイルの内容、構成ファイル、イメージの属性に関するユーザー定義のポリシーに照らしてイメージを評価することができます。

これらの手順はすべて警告に結びつきます。スキャンされていないイメージが本番環境にデプロイされた場合、本番で稼働しているイメージのパッケージに新たな脆弱性が発見された場合、あるいは稼働中のイメージのスキャンステータスが変更された場合には、プロアクティブに通知されます。

すべてのランタイムの脆弱性管理について、**Sysdig**はスキャンされていないイメージやスキャン結果の情報を**Kubernetes**のクラスタ、ネームスペース、デプロイメントに結びつけ、リスクを分類し、イメージのパッチやアップグレードの優先順位を決定します。このように**Kubernetes**のメタデータによって問題を切り分ける独自の機能により、適切な場所に適切なタイミングで作業を行うことができます。

開発・テスト時のソフトウェアのパフォーマンス

たとえ最新のソフトウェアに脆弱性がないとしても、そのソフトウェアのパフォーマンスがどの程度のものかをご存知でしょうか。結局のところ、安全なソフトウェアを提供するだけの開発チームは仕事をしていないことになります。彼らは、企業の競争力を高め、顧客をより幸せにするような、高性能で信頼性の高い安全なソフトウェアを提供する必要があります。

ソフトウェアを構築し、一連のテストを実行し、そのパフォーマンスを代替バージョンや以前のバージョンのコードと比較するという、このパフォーマンステストのプロセス（あるいは回帰テストの逆の見方）は、開発者にはよく理解されています。開発者が行う典型的な質問には以下のようなものがあります。

- 私のサービスの応答時間はどのくらいですか？
- 一般的なアクティビティでは常にエラーが発生していますか？
- 以前のバージョンと比較して、コードの基本的なリソース使用率（CPU、メモリ、ディスク）はどうなっていますか？
- 私のサービスで最も遅いAPIエンドポイントはどこですか？驚くことではありませんが、これらは運用者が本番環境でモニタリングする質問と同じ種類のものであり、これらの情報を取得するためのプロセスも非常によく似ています。ここですべてを繰り返すのではなく、本番環境でのソフトウェアのモニタリングについて話してみると、同じテクニックが開発／テストでも使用できることがすぐにわかるでしょう。

Run: 本番環境でのコンテナの運用
モニタリング、検出、エンフォースメント、コンプライアンス

企業の本番環境を大規模にモニタリングすると、本番環境でコンテナを運用する際の複雑なデータの課題が見えてきます。

- データをアプリケーション、ホスト、コンテナにマッピング
- オーケストレータの活用
- 保存するデータの決定

Sysdig ServiceVisionを使って、モニタリングやセキュリティデータをアプリケーション、ホスト、コンテナ、オーケストレータに関連付ける方法

環境が複雑になるにつれ、メタデータに基づいてメトリクスやポリシー違反をフィルタリング、セグメント化、グループ化する機能が不可欠になります。タグを使用することで、コンテナが稼働している物理的な状況に加えて、サービスやアプリケーションアーキテクチャの論理的な設計図を表現することができます。たとえば、データを開発／本番別、サービス別、ポッド別、コンテナ別、クラスタ別、データセンター別、ホスト別に分けて見ることができます。

データを調査する際には、メトリクスのラベルやタグを動的に選択して、目下の関心事や問題に必要な適切なビューを作成できるようにする必要があります。これらのタグを分類してグループ化することで、サービス全体のパフォーマンスを表示したり、デプロイメントやコンテナまで掘り下げて表示したりすることができます。また、動的に選択することで、組織内のさまざまなユーザーや役割の人が、データをすばやく可視化し、アプリケーションスタックの各部分に関する質問に答えることができます。

メトリクスのタグ付けについては、2つの考え方があります。

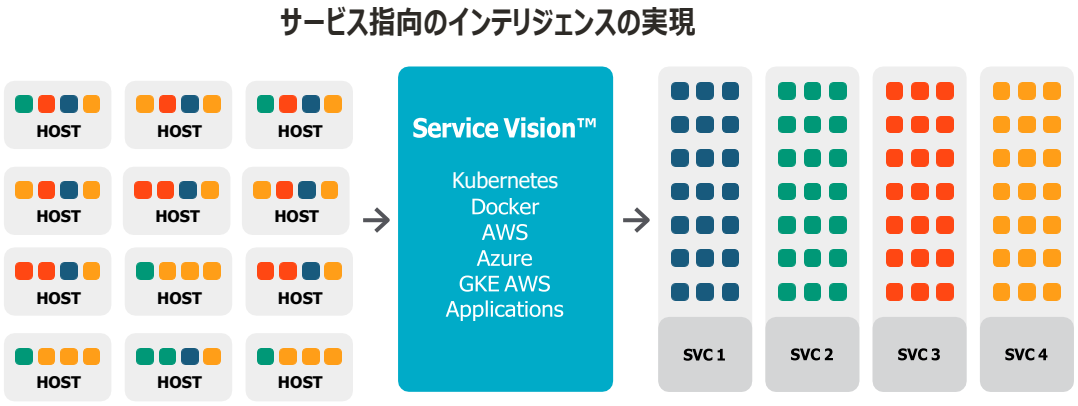
- 明示的 - アノテーションして保存しておきたい属性
- 暗黙的 - Kubernetesの記述子のようなオーケストレータのタグ（ネームスペース、ポッドなど）

チームの誰もが必要に応じてタグを追加できるように、明示的なタグを使用するためのメカニズムとベストプラクティスを用意しておく必要があります。そして暗黙的なタグはデフォルトでキャプチャーする必要があります。

オーケストレータは、コンテナのスケジューリング管理手法を根本的に変え、その過程でユーザーのモニタリング戦略にも影響を与えます。Kubernetesであれ、DC/OSであれ、Mesosであれ、Nomadであれ、必要とされるモニタリングのアプローチには同様の変化が見られるでしょう。個々のコンテナの重要性は低くなり、サービスのパフォーマンスがより重要になります。サービスは潜在的に多くのコンテナで構成されており、さらに重要なのは、オーケストレータがパフォーマンスや健全性の要件を満たすために、必要に応じてそれらのコンテナを動かすことができることです。

オーケストレータがアプリケーションに何をしているかを動的かつ自動的に理解できることは非常に重要であり、私たちはServiceVisionと呼ばれる機能を構築しました。この機能により、ユーザーが入力しなくても、Sysdigがすべてのタグ付けを管理することができます。これにより、システムがどれだけ分散していても、動的であっても、システムのパフォーマンスを把握することができます。

サービスコンテキストによる測定値の関連性と実用性の向上



プラットフォームの進化に伴い、デフォルトでは、任意の指標やイベントに対して12～25個のタグを発見し、追加することになりました。当社製品のパワーユーザーは、これよりもはるかに多くのタグを持っているかもしれません。タグのユニークな組み合わせは、保存、処理、そして正確なビューを抽出するためにオンデマンドで呼び出す必要のある個別のラインアイテムと考えてください。

- セキュリティ、モニタリング、フォレンジックなどのシステムでは、オーケストレータのメタデータに従って、すべてのメトリクスやイベントに暗黙のタグ付けを行う必要があります。
- このタグ付けは、イベント、ユーザーコマンド、入力、システムメトリクス、コンテナメトリクス、アプリケーションコンポーネントメトリクス、さらにはカスタムメトリクスにも適用されます。

  カスタムメトリクスについては、繰り返し説明します。Prometheus、StatsD、JMX、Golang expvar のいずれであっても、開発者はカスタムメトリクスを簡単に出力でき、モニタリングシステムはそのメトリクスがどこから来たのかを記録しておく必要があります。以下では、カスタムメトリクスとタグ付けの課題を回避するために、私たちが推奨する方法を紹介します。
- エージェントは、手動で入力しなくても、ホスト上やコンテナ内で実行されているアプリケーションコンポーネントを発見できる必要があります。
- オーケストレータはいつでもコンテナを移動できるため、モニタリングシステムの負担は、実行中のものを自動検出して正しいメトリックを収集することになります。
- 自動検出のためには、カスタムコード用にカスタムの「check」を記述する必要があるかもしれません。この作業は一度で済むべきですが、繰り返したり、いつ実行するかを手動で指示したりする必要はありません。

Kubernetesとオーケストレータのモニタリングについては**KubernetesMonitoring Guide**をご覧ください。

アラート機能とセキュリティポリシーの施行です。オペレーションシステムは、オペレーターの業務を簡素化するように設計されるべきです。そのための最も直接的な方法の一つは、システムが以下のことを可能にすることです。

- 問題を発見します
- ルール違反やパフォーマンスがしきい値を超えた場合に警告を発します
- 必要に応じて、問題を停止または軽減するための自動アクションを実行します

Sysdigは、これらの機能をSecure DevOps Platformに搭載しました。これらはすべて、ContainerVisionとServiceVisionを念頭に置いて構築されています。つまり、Sysdigはホスト、コンテナ、アプリケーションを見ているだけでなく、個々のコンテナだけでなく、サービス全体のアクションを自動的に管理することができます。より少ないアクションを管理し、それがダイナミックでエフェメラルなコンテナ環境に適応していることを確信できます。

- Metric Alertsは、システム内の任意のメトリクスに基づいてトリガーすることができます。メトリクスのしきい値は、合計、平均、およびレートごとに手動で設定できます。また、ブーリアンロジックを用いて複数の必須条件を設定し、アラートを作動させることができます。アラートは、Slack、PagerDuty、Eメールなど、さまざまなダウンストリームツールに送信できます。
- Event Alertsでは、メトリクスに基づく計算ではなく、個別のシステムイベントを使用します。たとえば、KubernetesのCrashLoopBackOff、Docker Kill、ユーザーがコンテナ内でシェルを起動した場合などです。特定の数のイベントが発生した後にアラートをトリガーします。
- Anomaly detectionは、手動でしきい値を設定するのではなく、Sysdigのアルゴリズムに依存して正常な動作を判断し、その範囲を超えた場合にアラートを出すというものです。Sysdigは、過去の指標パターンに対する異常値と、グループ内の異常値（例：ホストのグループ）の両方を検出することができます。
- Action triggersは、どのアラートにも結びつけることができます。典型的にはウェブフックを使用して、オペレーターはスケジューラを起動し、配置を変更したり、スクリプトを実行したり、その他のほぼすべてのアクションを実行することができます。
- Policy enforcementは、アクションとは若干異なるアプローチをとります。ランタイム中に機密性の高いセキュリティポリシーに違反した場合、システムは問題のあるコンテナを一時停止または強制終了させ、さらなる悪意のある行動や侵入を防ぐことができます。これは、先ほど説明したビルド時のポリシー施行に加えて行われます。
- Capturesは、インシデント発生時の詳細なフォレンジックとトラブルシューティングを可能にします。キャプチャーの価値については、次のセクションで詳しく説明します。キャプチャーはアラートによってトリガーされます。

ファイアウォールや、コンテナの不正使用を防ぐための別の方法は必要ありません。Sysdigは、適材適所のツールを使うことが非常に重要だと考えており、Kubernetesにすでに用意されているツールを採用しています。これにより、セキュリティ管理の効率的な展開、トラブルシューティングの簡素化、パフォーマンスのオーバーヘッドの最小化が可能になります。Sysdigは、以下のようなKubernetes内で既に提供されているツールの活用を支援します。

- Admission Controllers - 特定のポリシーに違反したコンテナイメージを使用しているポッドの起動を阻止します。
- Pod Security Policy - ワークロードを分析したり、ワークロードに対する影響をモデル化したりして、ワークロードに関するネイティブなセキュリティコントロールを作成します。
- Kubernetes Audit API - ユーザーが何をしているのか、どのワークロードをデプロイしているのか、どのようなアクションをとっているのかを理解し、分析することができます。
- Kube-State-metrics - Kubernetesはすでに自身状態について数百のデータポイントを収集しています。

私たちは、サービスやアプリケーションごとに小規模で分離されたバックエンドを展開することは合理的ではないと判断しました。その代わりに、メトリックとデータストレージのための水平方向にスケーラブルなアプローチを構築することを決定しました。

複数のバックエンドを用意することは、クラウド環境で運用する当社にとっても、オンプレミスで展開するお客様にとっても、管理しやすいアプローチとは思えませんでした。なぜこのようなことを考慮する必要があるのかというと、いくつかのオープンソースのモニタリングプロジェクトでは、デフォルトで分離型バックエンドモデルを採用しているため、スケーラビリティに関する懸念をユーザーに押し付けてしまい、管理作業が大幅に増えてしまうからです。私たちは、水平方向に拡張可能なバックエンドを構築し、ユーザーやサービスに応じてデータ、ダッシュボード、アラートなどを分離できるようにしたいと考えました。

また、時間に縛られないリテンションを実現するために、データを長期的に蓄積することにしました。フル解像度のデータを6時間保存し、それ以降はデータの集約を開始します。

エンタープライズグレードスケール

バックエンドは進化し続けていますが、現在はCassandra（メトリクス）、ElasticSearch（イベント）、MySQL（設定データ）、Apache Kafka（モニタリングログ）、Redis（サービス内仲介）の水平方向にスケーラブルなクラスタで構成されています。これらのコンポーネントをベースに構築することで、高い信頼性とスケールを実現し、長期的なトレンド分析のために数年分のデータを保存することができます。

エンタープライズスケールは、**Sysdig**の導入を促進する重要な要因の一つです。**Sysdig**は、パフォーマンスと安定性を損なうことなく、世界最大規模のクラウドのデプロイメントをサポートするためにスケールアップします。**Sysdig**プラットフォームは、オーケストレータメカニズムを使用して個々のコンポーネントをスケーリングし、何千ものプロダクションクラスタで負荷テストを行っています。**Sysdig**は、メトリクスの容量のような個々のデータのニーズの負荷を処理するために深くスケーリングできるだけでなく、何千ものエージェントのニーズや何百万ものメトリクスのインジェストのような処理とスケーラビリティのために広くスケーリングすることができます。

Sysdigが収集、保存するデータはすべてREST APIでアクセス可能です。このスケーラブルなバックエンドは、Sysdigのクラウドサービスで使用されていますが、セキュリティや分離性を高めるために、企業がプライベートクラウドで運用するソフトウェアとして導入することも可能です。Sysdigは、モニタリング用のシステムと、セキュリティ、長期分析、データ保持、コンプライアンスなどのためのシステムを別々に運用する必要がないように設計されているため、急成長している企業であれば、ビジネスに合わせてシステムを拡張・成長させることができます。

さらに、私たちはシンプルにしたいと考えています。Sysdigには、状態を示すダッシュボードやアラート、セキュリティポリシーなどが標準で装備されています。通常、お客様は導入後数分でSysdigの可視性の恩恵を受け始め、ビューやポリシーをパーソナライズするための調整には最小限の時間しかかかりません。

Respond: MTTR（Mean Time To Respond）の短縮

コンテナ環境でのトラブルシューティングとフォレンジック

コンテナは、小型、軽量、分散型に設計されています。これは、デプロイのしやすさと再現性にとっては素晴らしいことですが、パフォーマンスの問題やセキュリティイベントが発生した場合に、可視性を得る能力に影響を与えます。

古くから関係のある ssh、top、ps、ifconfig などを覚えていますか？あなたのコンテナにはそれらがないかもしれません。管理されたPaaS環境で運用している場合は、これらのツールが利用可能であってもアクセスできないかもしれません。そして、コンテナはもう存在しないかもしれないと言いましたか？オーケストレータがその役割を果たしていれば、トラブルシューティングを行う前に、影響を受けたコンテナは消えてしまっているでしょう。

開発者に戻ってホストからのtcpdumpを再度依頼することもできません。

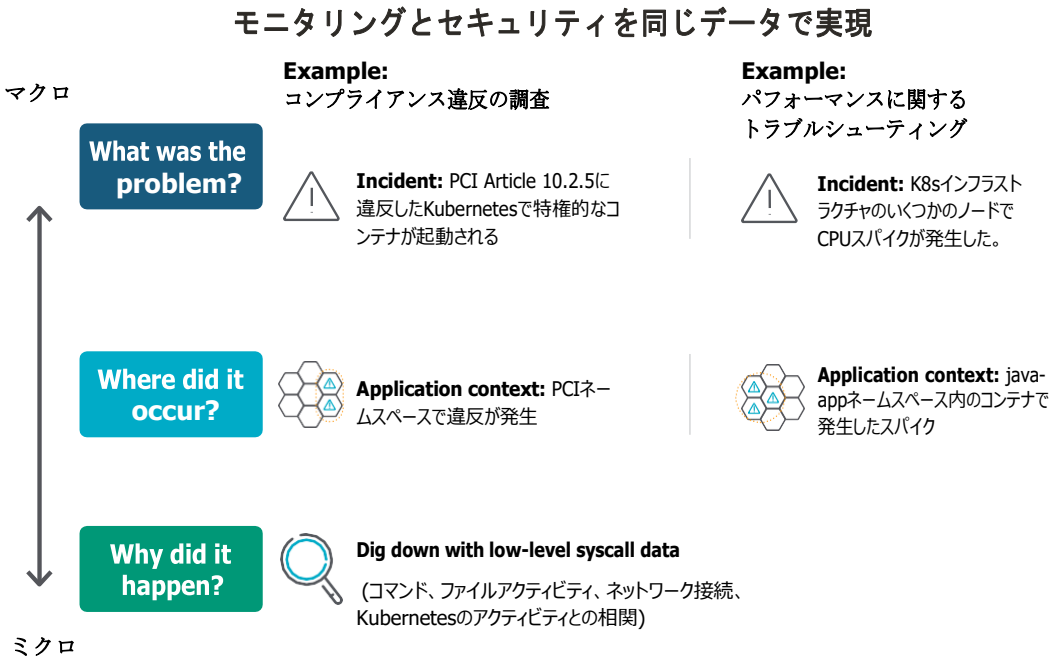
要するに、必要な情報を得るためには複雑な作業が必要になるのです。その上、トラブルシューティングのためにオーケストレータから適切なコンテキストを得ることは不可欠です。したがって、開発者がこのような詳細な情報を取得できるようにすることが不可欠であり、理想的には本番環境を汚染することなく取得することができます。トラブルシューティングを簡素化することは、コンテナワークロードのモニタリングを可能にすることと同様に重要であると判断したため、この問題に取り組む必要がありました。

ここでSysdigのコンテナトラブルシューティング機能が活躍します。ホスト上のすべてのシステムコールをキャプチャーする機能は、問題やイベント発生時にアプリケーション、コンテナ、ホスト、ネットワークがどのように動作していたかを深く可視化します。Sysdigのキャプチャーは、詳細なインシデント対応やフォレンジックのためのメカニズムを提供します。イベント発生時やその前からホストのシステムコール情報をキャプチャーすることで、キャプチャーは、コンテナの刹那的なワークロードのアクティビティの永続的なコピーを提供します。コンテナはとっくに破壊されているかもしれませんが、コンテナが何をしていたのか、何と相互作用していたのか、どんなエラーを発生させたのかなどのコピーがSysdig Secure DevOps Platform内に安全に保存され、後で検索して分析することができます。

すべてのシステムコールアクティビティをスタンドアロンファイルに記録するSysdigの機能は、本番環境からデータをキャプチャーして、ラップトップでトラブルシューティングを行うことを可能にします。また、コンテナがなくなった後もこの作業を行うことができるため、髪の毛に火がつかなくなっても適切な事後調査を行うことができます。

オーケストレーションマスターとの自動統合と通信により、関連するメタデータをリアルタイムに収集し、個々のマシンの状態だけでなく、分散システムのコンテキストと状態を把握することができます。

Sysdigのキャプチャーは、その場限りのトラブルシューティングのために手動で作成することもできますが、さらに重要なのは、パフォーマンスのしきい値、ネットワークアクティビティ、セキュリティイベントなどのカスタムイベントトリガーに基づいて完全に自動化することです。また、これらのキャプチャーをSIEMプラットフォームと統合することで、複数のシステムやアプリケーションを横断する、より複雑なインシデントレスポンスワークフローを実現することができます。これにより、アプリケーションのトラブルシューティングの根本原因分析や、セキュリティインシデントのフォレンジック分析を迅速に行うことができます。



たとえば、特定のサービスのデータベースがアウトバウンド接続を生成していることを示すアラートを受け取ったとします。Sysdigのアラートは、そのホスト上の必要な時間のすべてのシステムコールを記録するキャプチャーのトリガーとなります。これらのキャプチャーには、違反の後だけでなく、違反前のデータも含まれます。これをSysdig Inspectで確認することで、コンテナのコンテキストを正しく把握し、ネットワーク接続を掘り下げることができます。

Sysdig Inspectで何ができるかを知るには、**Introduction toSysdig Inspect**をお読みください。

インフラのコンプライアンスの確保

企業は、ホストやノードからコンテナ内のサービス設定ファイルに至るまで、インフラ全体の設定がCISベンチマークなどの標準に準拠していることを確認する必要があります。

Sysdig Platformのもう一つの付加価値は、docker-benchやkube-benchのようなコンテナやKubernetesのコンプライアンスベンチマークを実行してレポートする機能で、インフラのすべての論理層で構成を検証することができます。KubernetesやDocker CISベンチマークの設定がドリフトした場合（ベストプラクティスに準拠していないことを意味します）、Sysdigはコンテナのコンプライアンスを維持するために必要な変更を推奨するガイド付きの改善策を提供し、問題発生時にセキュリティ専門家やDevSecOpsの時間を節約します。

監査活動

トラブルシューティングに加えて、コンテナ環境で誰が何をしたかの監査証跡を残しておくことは、違反やその他の違反があった場合に監査人の要求に応えるために不可欠です。繰り返しになりますが、コンテナの動的で一時的な動作は、以前の世代のインフラよりもこのタスクをはるかに困難にしています。

Sysdigは、すべてのコンテナレベルのアクティビティを深く掘り下げ、シスコールの監査証跡とKubernetesの監査ログを組み合わせて、コンテナ環境のすべてのアクティビティを時系列で表示する独自の機能を持っています。

この監査データは、外部の当事者に提供、別のイベントストレージシステムや通知ツール（Slack、Webhook、Mailなど）、セキュリティ情報・イベント管理SIEM（Splunk、Syslog、Qradar、MCMなど）に供給することができます。

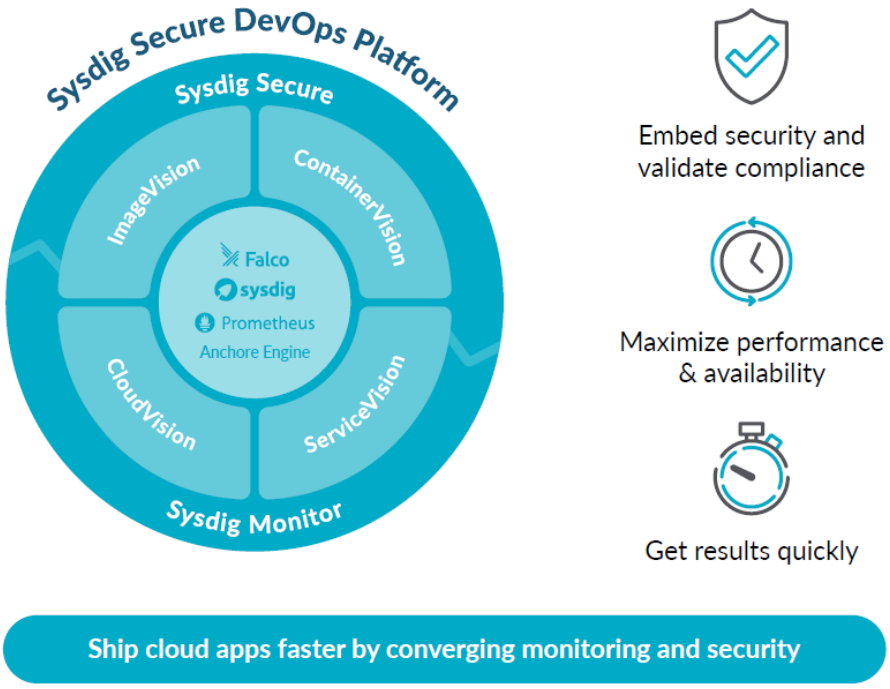
数回クリックするだけで、悪質な行為者のデータを隔離することができます。悪質な行為者が隔離された侵害の場合、Sysdigはその行為者の活動範囲を把握し、どのマイクロサービスやデータが影響を受けたかを即座に理解する強力な手段を提供します。

結論

コンテナの可視化とセキュリティのために、拡張性の高い分散型データプラットフォームを構築することは、簡単なことではありません。適切なデータを適切なコンテキストで取得することは、コンテナのライフサイクル全体にわたって開発者、オペレーター、セキュリティ専門家にサービスを提供できる堅牢なシステムを構築するために不可欠です。

セキュアなDevOpsを実現するためのツールを検討する際に重要な質問

- 最も信頼性が高く、安全なコードだけを本番環境に投入することをどのようにして実現するのか？
- モニタリングとセキュリティのために、運用中のコンテナをどのように計測するのか？
- データに適切なコンテキストを与えるためのオーケストレータとのインターフェースはどうするのか？
- コンテナにセキュリティとコンプライアンスのポリシーをどのように実装するのか？
- アプリケーションデータやカスタムメトリクスの収集をどのように簡素化するのか？
- どのようなデータを保持するのか？
- また、ダイナミックでエフェメラルなコンテナ環境におけるトラブルシューティングやフォレンジックを可能にするにはどうすればよいのか？



Sysdig.comのリソースセクションでは、コンテナへの移行を支援するより詳細なガイドをご覧いただけます。お客様のニーズに合わせて、アカウントチームが概念実証をサポートします。

Sysdig Secure DevOps Platformの無料トライアルを利用して、自信を持ってKubernetesの本番運用を開始してください。

sysdig.com/company/free-trial