sysdig

Sysdig 2021

コンテナセキュリティと使用状況レポート

本文の内容は、2021 Container Security And Usage Report(https://dig.sysdig.com/c/pf-2021-container-security-and-usage-report?x=u_WFRi）2021年1月13日時点を元に日本語に翻訳・再構成した内容となっております。

コンテンツ

エグゼクティブサマリー

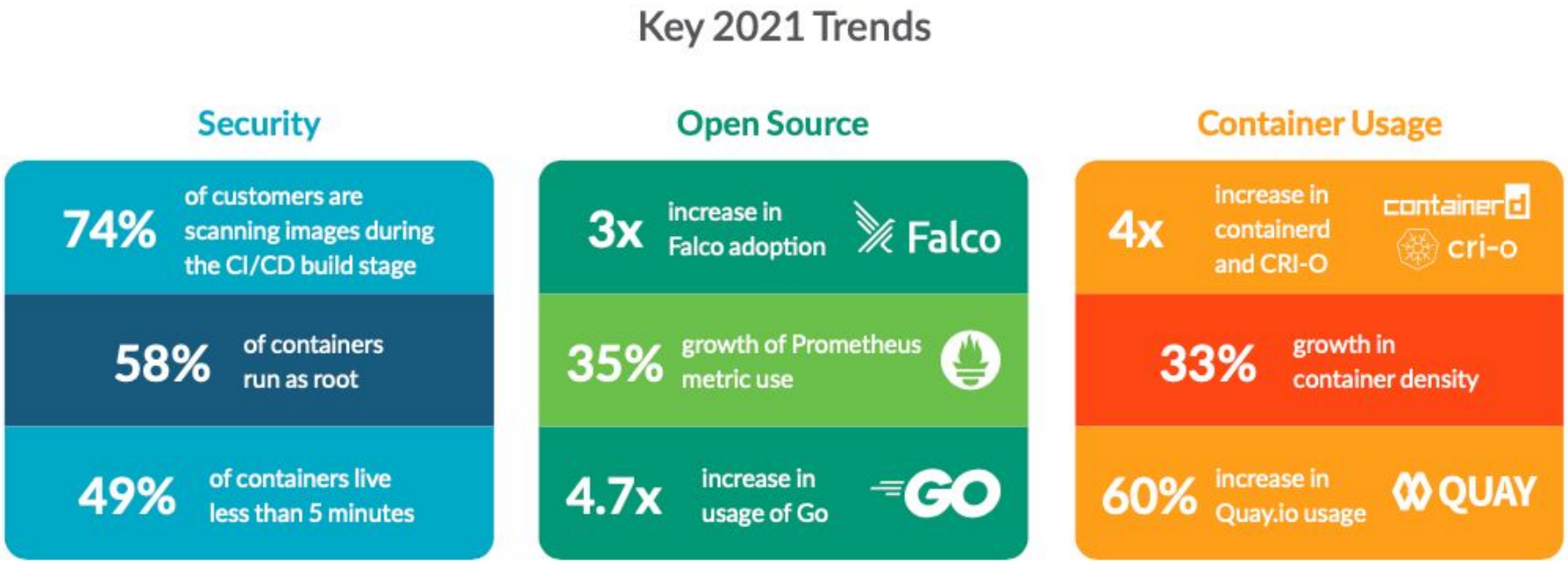
過去4年間、当社はリアルタイムの実世界の顧客データを通じて、コンテナの使用状況についての洞察を提供してきました。当社のセキュリティと監視機能が成長するにつれ、当社独自の視点から、企業がどのようにセキュリティとコンプライアンスに対処しているか、また、インフラ、アプリケーション、コンテナの使用状況が時間の経過とともにどのように進化しているかなどの詳細を発見することができます。これらの洞察に基づいて、Sysdig 2021 コンテナセキュリティと使用状況レポートをお届けします。

当社のお客様からは、コンテナ環境を取り巻くセキュリティとコンプライアンス上の懸念は、その刹那的な性質上、最重要課題であるとのお声をいただいております。昨年のレポートと同様に、コンテナの約半数が5分未満で稼働しています。これは、インシデント対応、フォレンジック、トラブルシューティングに使用できる詳細な記録の必要性を浮き彫りにしています。このことを念頭に置き、お客様が直面している課題を明らかにするために、セキュリティの状態をより深く見ていくことを追加しました。当社の分析では、多くの企業でシフトレフトの傾向がKubernetesセキュリティにまで及んでおり、4分の3の組織がデプロイ前のCI/CDビルドフェーズでコンテナイメージをスキャンしていることが明らかになりました。多くのチームは脆弱性を特定することに高い意識を持っていますが、その設定ミスが攻撃者に扉を開いたままにしています。実際、私たちの分析によると、コンテナイメージの大部分は依然として過度に寛容に設定されており、58%がrootとして実行されており、セキュリティに深刻な影響を与えていることがわかりました。コンテナ環境が成熟するにつれ、組織はスキャンだけでは十分ではないことを認識しています。また、継続的な脅威に対処するためのランタイムセキュリティも必要です。これらの懸念に対処する方法として、Cloud Native Computing Foundation (CNCF) Falcoプロジェクトの採用が大幅に増加しています。

コンテナオーケストレーションのためのKubernetesの使用は2020年も変わりませんでしたが、組織がDockerからcontainerdやCRI-Oへと移行していく中で、コンテナランタイムの選択に明確なシフトが見られます。実際、Kubernetesプロジェクトは2021年後半にDockerの使用を正式に非推奨とすることを発表しました。今年もコンテナ密度が高まる中、組織はこれらの環境を監視する標準的な方法としてPrometheusにシフトしています。顧客の間でのPrometheusメトリクスの使用は前年比で35%増加し、GitHubの統計によると、エクスポーターのトップ3はnode-exporter、blackbox-exporter、jmx-exporterとなっています。Quay レジストリは今年も採用が増加し、クラウドネイティブ開発者に人気のプログラミング言語である Golang は組織内での利用が急増しました。

このレポートのデータは、当社の顧客の一部が毎日実行している数百万個のコンテナと、過去1年間に当社の顧客が実行してきた10億個近くのユニークなコン

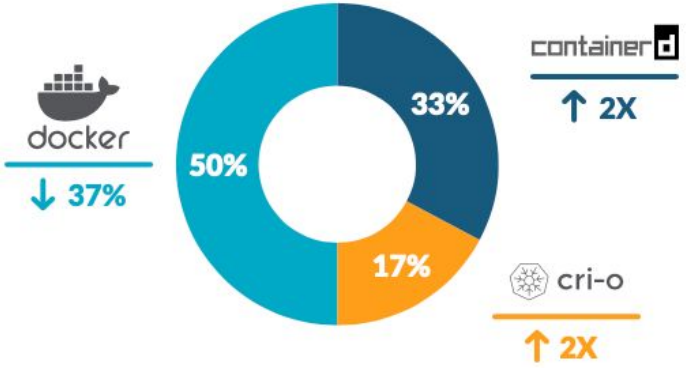
テナの分析に基づいています。このレポートでは、セキュリティ、コンプライアンス、サービス、アラート、およびKubernetesの使用パターンについてさらに詳しく説明しています。これらの情報は、幅広い業種の世界中の企業におけるコンテナ環境のセキュリティや利用状況の実情を把握するのに役立ちます。



どのようなコンテナプラットフォームがデプロイされているか？

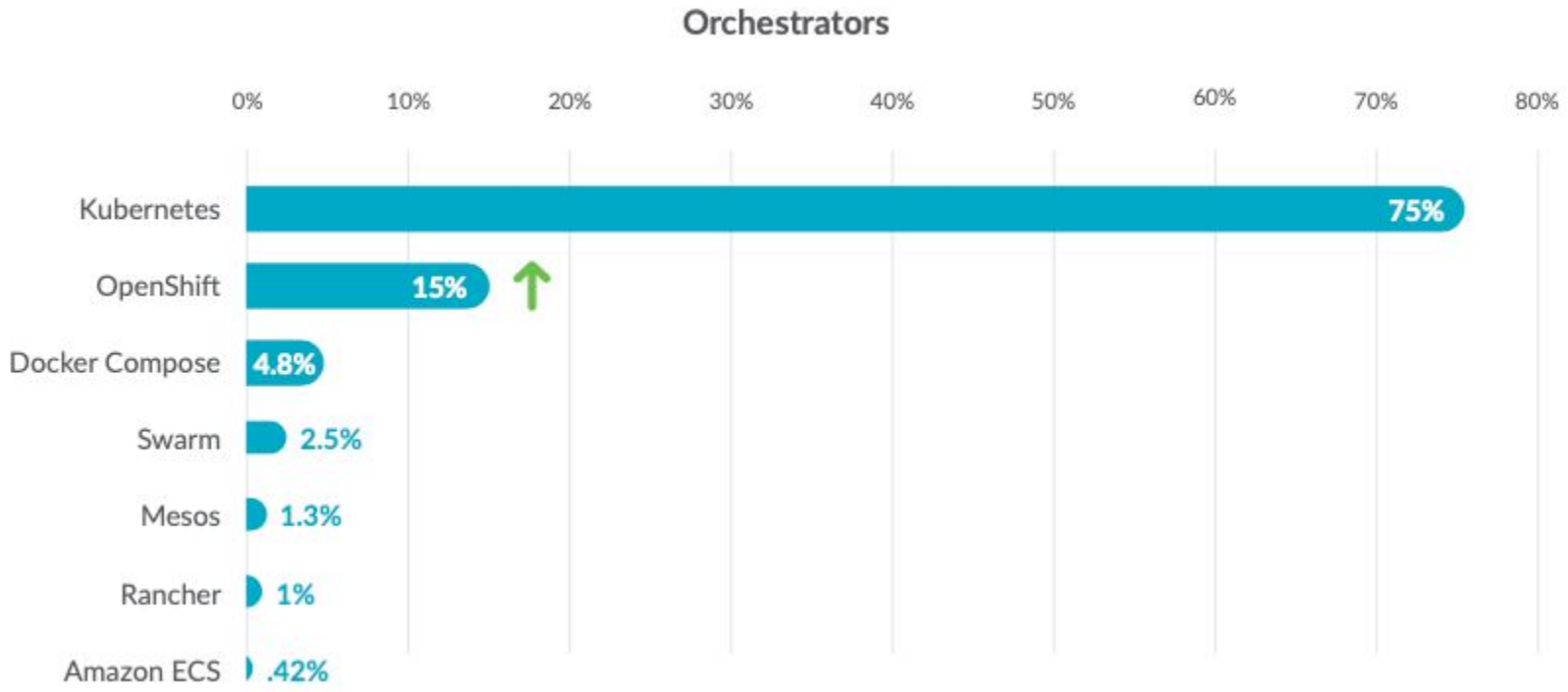
コンテナの実行時間

昨年は79%であったDockerが今年は50%まで落ち込んでいますが、この1年間でcontainerdとCRI-Oの両方が大きく伸びています（それぞれ18%、昨年は4%増）。また、Kubernetesプロジェクトが2021年後半にDockerの使用を正式に非推奨とすることを発表したことも注目に値します。公平に言うと、Dockerではcontainerdが使用されていることに注意が必要です。Dockerエンジンは以前、高レベルと低レベルのランタイム機能の両方を実装していました。これらは現在、containerdプロジェクトとruncプロジェクトに分かれています。いくつかのオプションが出てきているので、どのコンテナランタイムを使うかは少し不明確に思えるかもしれません。様々なソリューションは、オーバーヘッドの削減、安定性、拡張性、コンテナレジストリの互換性などを利点として挙げています。しかし現在では、オープンスタンダードのおかげで、間違った選択をしてロックインしてしまうことへの懸念は消えています。これをさらに簡単にするために、OpenShift、GKE、および IKS のような一般的なプラットフォームは、複数のコンテナランタイムの並列使用をサポートしており、一般的には選択したランタイムで設計されているため、どれを使用するかを決定するためにサイクルを費やす必要はありません。



コンテナオーケストレーションプラットフォーム

Kubernetesは現時点で他のオーケストレータを抑えて安定したリードを保っており、昨年のレポートからはわずかにシフトしています。ページ下部のチャートは、現在の内訳を示しています。驚くべきことに、SwarmとMesosは昨年と比較して、それぞれ5%と4%とほぼ同じ利用レベルにとどまっている。OpenShiftが9%から15%へと最も大きく跳ね上がっているのは、複数のクラウド環境で実行できることから、OpenShiftに頼るユーザーが増えているようです。Docker Composeは、Kubernetesのようなマルチホストオーケストレーターの直接の関連性とは考えられないかもしれませんが、1つのホストのみで複数のコンテナを管理するために使用されていますが、今年は追加されました。



セキュリティとコンプライアンス

組織がコンテナワークロードを本番環境に移行するにつれ、セキュリティとコンプライアンスをDevOpsのワークフローに統合する必要性を認識するようになってきています。"セキュリティを左へシフトする "というのは、コンテナの脆弱性をスキャンすることを指すことが多いバズフレーズになっています。公開レジストリから引き抜かれたコンテナイメージの割合が高く、スキャンされたイメージの障害率が高いことを考えると、スキャンは明らかに重要です。しかし、このデータは、リスクを軽減するためのコンプライアンスチェックと厳格なランタイムポリシーの必要性も浮き彫りにしています。Kubernetesとクラウドネイティブ環境におけるセキュリティとコンプライアンスの状態についての洞察を提供するために、脆弱性スキャン、ランタイムセキュリティ、コンプライアンスを含むデータポイントを分析しました。

イメージスキャン

コンテナイメージのソースにかかわらず、本番環境にデプロイする前にイメージスキャンを行い、既知の脆弱性を特定することが重要です。脆弱性のリスク範囲を定量化するために、7 日間にわたってスキャンしたイメージの合格率と不合格率をサンプリングしました。半数以上のイメージが不合格である。つまり深刻度が高いかそれ以上の既知の脆弱性が発見されました。

"スキャンをすればするほど、欠陥の特定と修正が早くなります。CI/CD パイプラインでスキャンを行うことで、リスクを軽減し、セキュリティがアプリの高速配信の妨げにならないことを確認できました。"

- ナトネル・テフェリ（Natnael Teferi）
FISのリードDevSecOpsクラウドセキュリティアーキテクト

ＯＳ脆弱性スナップショット

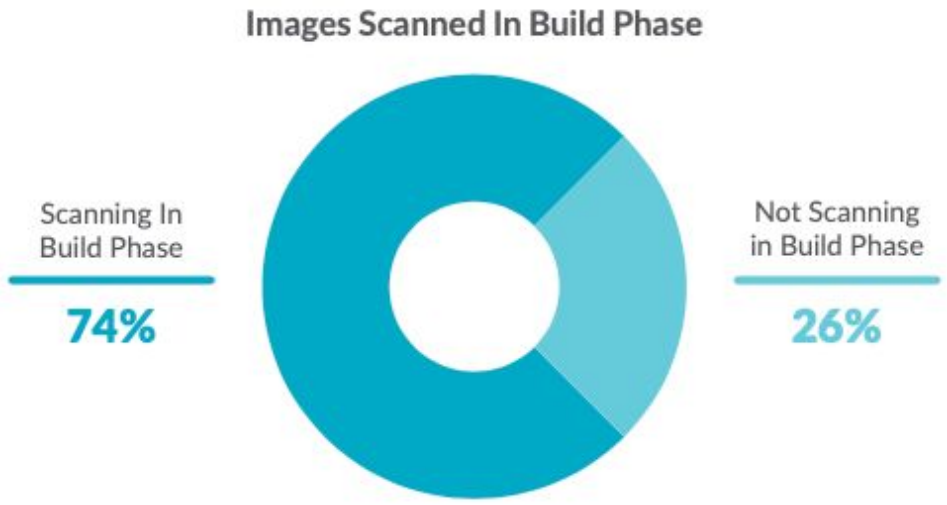
OSの脆弱性のうち、4%が高レベルまたは重大な脆弱性であることに気がつきました。これは低いと思われるかもしれませんが、OSの脆弱性が悪用されると、イメージ全体が危険にさらされ、アプリケーションがダウンする可能性があります。このため、特にレジストリスキャンの一部としてこの機能を提供しているクラウドプロバイダー（ECR、GCRなど）では、OSの脆弱性をスキャンすることに重点が置かれています。

非OSの脆弱性スナップショット

多くのチームがチェックしていないのは、サードパーティ製ライブラリの脆弱性です。非OSパッケージの53%が高レベルまたはクリティカルレベルの深刻度の脆弱性を持っていることがわかりました。開発者は無意識のうちに、Python PIPやRuby Gemなど、これらの非OSオープンソースパッケージから脆弱性を引っ張ってきて、セキュリティリスクを導入している可能性があります。

ビルドフェーズでのスキャニング

DevOpsチームは、開発ライフサイクルの早い段階で重要な影響を考慮し始めることを目標に、「シフトレフト」している。これらの懸念事項の中でも、セキュリティは重要です。分析の一環として、CI/CD パイプラインのビルドフェーズの一部としてイメージのスキャンを行っている組織と行っていない組織の数を調べました。当社の顧客の74%は、実際にデプロイ前にスキャンを行っています。これは、ビルドフェーズでスキャンを行うことで、本番前にイメージを使って潜在的なセキュリティリスクに対処することができることを示しています。



スキャンはどこで行われるか: インライン Vs. バックエンドスキャン

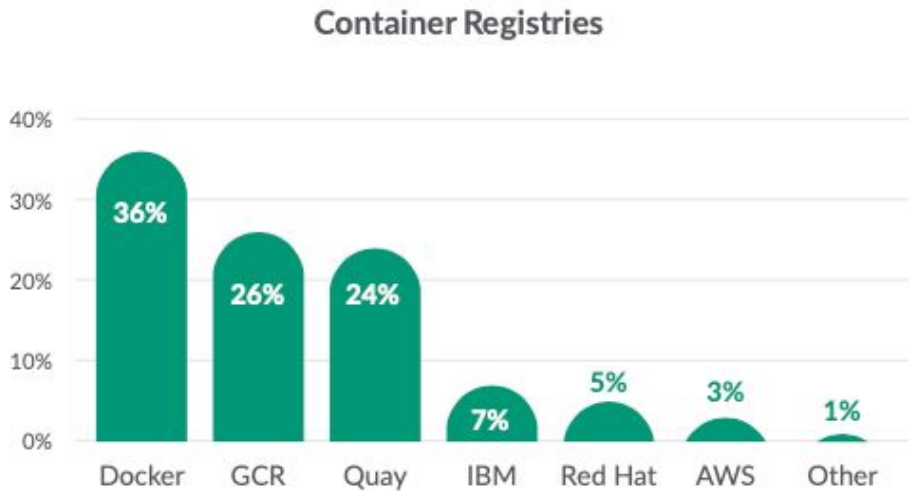
お客様がイメージをスキャンするには、2つの基本的なアプローチがあります。

バックエンドスキャン - バックエンドスキャンを使用する場合（すなわち、UI内で直接、またはsdc-cliを使用する統合を介して）、Sysdigバックエンドはレジストリからイメージ全体を引き出し、イメージ分析（インストールされているパッケージ、バージョン、ファイル属性、Dockerfile命令などのイメージメタデータの抽出）と評価（OS/非OSの脆弱性、設定ミス、不正なセキュリティ慣行の検出）の両方を実行します。多くのチームはバックエンドスキャンを活用しています。インラインスキャンは、より優れたセキュリティを提供するためのゴールかもしれませんが、より高度なステップです。

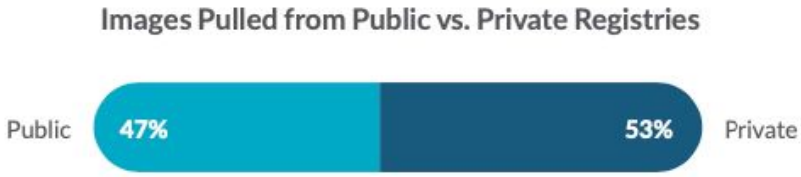
インラインスキャン - インラインスキャンを使用する場合、イメージ分析フェーズはCI/CDパイプライン、レジストリ、または実行時に直接行われます。結果として得られたメタデータは評価のためにSysdigバックエンドに送られ、ポリシーの評価はワーカーに送られます(すなわち、PDFまたはJSONアーティファクトとして)。これにより、イメージコンテンツを共有したり、レジストリの認証情報を外部に公開したりすることなく、イメージデータを完全に管理することができます。スキャン結果はSysdigで直接見ることができます。

パブリックおよびホストされたコンテナレジストリ

コンテナレジストリは、コンテナイメージをホスティングして管理するためのリポジトリを提供します。Dockerレジストリーは最も頻繁に使用されており、34%のお客様に共通して使用されています。このメトリクスには、プライベートホスティングされたリポジトリとパブリックリポジトリの両方が含まれます。クラウドプロバイダーがホストするレジストリソリューションは、ますます人気が高まっています。過去数年と同様に、Google Cloud Registryが再びトップのパブリッククラウドリポジトリとなり、当社のSysdigユーザーの26%が利用しています。しかし、Quayは昨年から若干の伸びを拾い上げており、14%から24%に増加しています。

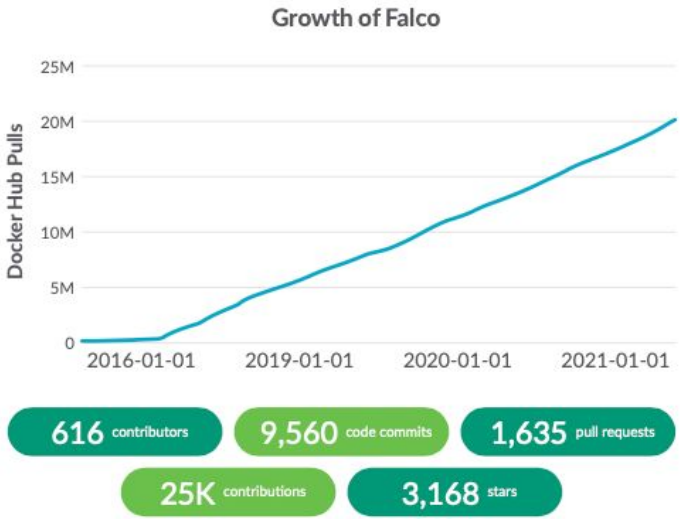


これらのさまざまな仕組みの中で、公開リポジトリとプライベートリポジトリからプルされたコンテナの割合を調べました。パブリックソースの信頼度は、昨年の40%から今年は47%に増加していることがわかりました。パブリックリポジトリからコンテナイメージを使用するリスクは、セキュリティ脆弱性の検証やチェックが行われているものが少ないことです。しかし、Kubernetes環境のセキュリティ手順やプロセスを改善する企業が増えているため、パブリックリポジトリを利用する利便性がリスクを上回る可能性があります。当社のお客様は、組織内で使用が承認されているコンテナレジストリを定義するためのポリシーを作成しています。



ランタイムセキュリティの脅威

コンテナライフサイクルのビルドフェーズで既知の脆弱性に対処した後、チームはランタイム時に異常な動作を検出してセキュリティアラートをトリガーするポリシーを設定する必要があります。Kubernetesのランタイムセキュリティは、組織が取り組み始めたばかりのものです。Sysdigがコントリビュートした CNCFオープンソースプロジェクトであるFalcoは、以下のプロジェクト統計に見られるように、急速に勢いと関心を増しています。このプロジェクトの Docker Hubのプル数は現在2,000万を超えており、昨年の252%と比較して 300%の伸びを示しています。Falcoは、セキュリティ違反を検出してアラートを生成するランタイムポリシーの定義を可能にします。



"コンテナ環境のセキュリティに対する懸念が高まる中、Falcoの継続的な成長は、より多くのユーザーがコミュニティベースのルールを活用することを意味します。Falcoプロジェクトが成長するにつれ、悪質なアクターに対して集団で協力することで、Kubernetesのセキュリティは強化されていきます。"

- CNCF、CTO、クリス・アニシュチク

ルートとして動作するコンテナ

チームは脆弱性をスキャンする必要性を理解していますが、一般的な設定ミスをスキャンしていない可能性があります。私たちが確認したのは、イメージの 58% が root として実行されており、特権コンテナが侵害される可能性があるということです。root で実行されるべきコンテナ（セキュリティやシステムデーモンなど）もありますが、これはコンテナ全体のごく一部です。顧客と話をしていると、実際には、実行時に危険な設定が検出されたとしても、チームは迅速なデプロイを継続するためにコンテナを停止させません。その代わりに、猶予期間内に実行して、その後に是正措置を決定します。

ランタイムポリシー違反のトップ

顧客が受け取っているアラートの量によって測定されるポリシー違反を調べました。これは、コンテナユーザーが最も頻繁に発見しているランタイムセキュリティリスクの種類を示しています。今年は、疑わしいファイルシステムと疑わしいコンテナの違反が増加しました。

以下の違反はそれぞれ、Sysdig Secureでデフォルトで有効になっているFalcoのセキュリティポリシーによって検出されます。以下では、頻度の高い順に上位7つの違反を、それぞれの説明とともに提供し、可能性のある脅威を説明しています。

違反	それは何か	なぜセキュリティ脅威なのか
etc以下への書き込み	etc ディレクトリ以下のファイルへの書き込みを試みる。	etcにあるファイルを追加したり変更したりすると、アプリケーションの動作を変更しようとしている可能性があります。
特権コンテナの起動	特権コンテナの起動	特権コンテナは、ホストシステムのデバイスと相互作用したり、ホストOSに危害を加えたり、他のコンテナにアクセスしたりすることができます。
root以下への書き込み	/ または、/root ディレクトリ以下への書き込みを試みる。	これらのディレクトリ内のデータを変更すると、コンテナにソフトウェアをインストールしようとする不正な試みになる可能性があります。
不審なファイルシステムの変更	機密な/重要なファイルを変更する可能性のある不審なファイルシステムの活動を新たに確認しました。	攻撃者は機密データにアクセスしようとしているかもしれません。
センシティブマウントコンテナの起動	センシティブなホストディレクトリからファイルシステムをマウントしているコンテナを起動します。	コンテナが、機密ファイルを含む可能性のあるデータボリュームにアクセスしていることを示します。
不審なコンテナの活動	不審なコンテナ関連の活動（コンテナへのexec等）を識別。	コンテナシステム内での侵害の指標になるかもしれません。
コンテナ内のターミナルシェル	シェルが、端末が接続されたコンテナへのエントリポイント/execポイントとして使用されました。	攻撃者がシステムを操作したり、マルウェアをダウンロードしたり、その他の悪意のある活動を開始したりすることを可能にします。

コンプライアンス

今日の企業は、PCI-DSS、HIPAA、GDPRを含む多くのガバナンスと規制遵守の要件に直面しているため、規制を遵守するためにはベストプラクティスに従うことが不可欠です。Sysdigプラットフォームは、監視対象のクラスターに対してコンプライアンスチェックを実行し、ホスト、コンテナ、および環境の他の側面を、定義されたベストプラクティスのセットに照らし合わせてチェックします。これには、Center for Internet Security（CIS）のベンチマークテスト、KubernetesのCISベンチマーク、DockerのCISベンチマークが含まれます。CIS benchmark for Dockerの80以上のベンチマークルールの中からサンプルを選び、Sysdigのユーザーとこれらのベストプラクティスに対するコンプライアンスの状況を浮き彫りにしました。8つのベンチマークでは、各ホストに存在するコンテナイメージを評価し、組織がリスクにさらされる可能性のあるパーミッション、セキュリティツール、構成に関連した構成の問題について評価しています。私たちは、これら8つのコンテナチェックのそれぞれについて、中央値のスコアを取りました。この場合のスコアは、テストに失敗し、リスクを低減するための推奨ベストプラクティスを遵守していないホストごとのコンテナの尺度です。

"脆弱性やセキュリティ問題が発生したときにそれを検知し、検知できることは、お客様に安全でライブなオブザーバビリティSaaSソリューションを提供する上で非常に重要です。自動化されたランタイム・チェックを使用してコンプライアンスを検証することで、ポリシー違反に迅速に対処することができます。

SOC 2に準拠しているため、当社のスピードを落とすことなく、より良いサービスを提供することができます」と述べています。

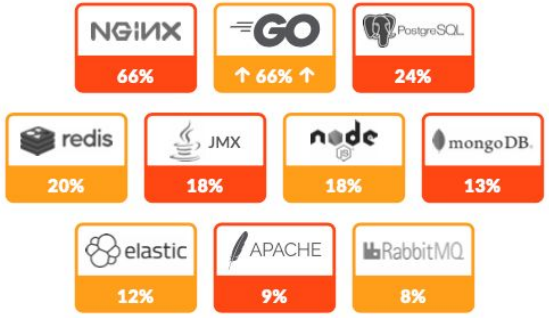
- Geeta Schmidt, CEO Humio

ベンチマーク	ホストあたりの脆弱性のあるコンテナの数の中央値	なぜ脅威なのか
ヘルスチェック命令が設定されていないコンテナ	49	重要なセキュリティコントロールは可用性です。コンテナイメージにヘルスチェック命令を追加することで、Dockerエンジンが定期的に実行中のコンテナインスタンスをその命令と照らし合わせてチェックし、コンテナがまだ動作中であることを確実にします。ヘルスチェックの結果に基づいて、Dockerエンジンは正しく応答しないコンテナを終了させ、新しいコンテナをインスタンス化することができます。
専用の cgroup を持たないコンテナ	34	実行時には、元々定義されているものとは別のcgroupにコンテナをアタッチすることが可能である。別の cgroup にアタッチすることで、過剰なパーミッションとリソースがコンテナに付与されているため、セキュリティ上のリスクがあることを証明することができます。
デフォルトの seccomp プロファイルが無効になっているコンテナ	34	大量のシステムコールがすべてのユーザとプロセスに公開され、その多くはプロセスの全生涯にわたって使用されません。ほとんどのアプリケーションでは、これらのシステムコールをすべて必要としないため、利用可能なシステムコールのセットを減らすことで利益を得ることができます。システムコールのセットを減らすことで、アプリケーションにさらされるカーネルのサーフィスを減らし、アプリケーションのセキュリティを向上させることができます。
無制限に再起動できるコンテナ	29	コンテナを無期限に起動しようとし続けると、ホスト上のサービス拒否につながる可能性があります。特に同じホスト上に多くのコンテナがある場合は、分散型のサービス拒否攻撃を簡単に行うことができます。さらに、コンテナの終了ステータスを無視して常にコンテナを再起動しようとすることは、コンテナが終了した根本的な原因を調査しないことを意味します。
デフォルトの ulimit を持つコンテナ	29	ulimit はシェルが利用可能なリソースとそれによって起動されるプロセスをコントロールします。システムリソースのリミットを慎重に設定することで、サービス拒否状態から保護することができます。場合によっては、正当なユーザやプロセスが誤ってシステムリソースを過剰に使用し、システムの劣化や応答不能を引き起こす可能性があります。
AppArmorプロファイルのないコンテナ	28	SELinuxの代替として、AppArmorはほとんどのLinuxディストリビューションでデフォルトで利用可能です。AppArmorは、各アプリケーションへのセキュリティプロファイルの関連付けを可能にし、基盤となるシステムへのアクセスを制限します。
読み書きモードで /root がマウントされているコンテナ	28	コンテナのルートファイルシステムは、Docker runのread-onlyオプションを使用して「ゴールデンイメージ」として扱われるべきです。これにより、コンテナ実行時にコンテナのルートファイルシステムへの書き込みを防ぎ、不変のインフラストラクチャーの原則が適用されます。

顧客はどのようなサービスを実行しているのか？

コンテナで稼働しているオープンソースソリューションのトップ10

オープンソースは、エンタープライズコンピューティングの顔を変えました。それはインフラストラクチャーだけでなく、特にアプリケーションの開発においてもイノベーションを後押ししています。コンテナ内のプロセスを自動検出するSysdigの能力は、お客様が本番で実行しているクラウドネイティブサービスを構成するソリューションを瞬時に把握することができます。以下は、Sysdigのお客様が導入しているオープンソース技術のトップ10です。



2021年のリストには、幅広いサービスが含まれています - それぞれが現代のアプリケーションの機能に不可欠なもので、以下のようなものがあります。

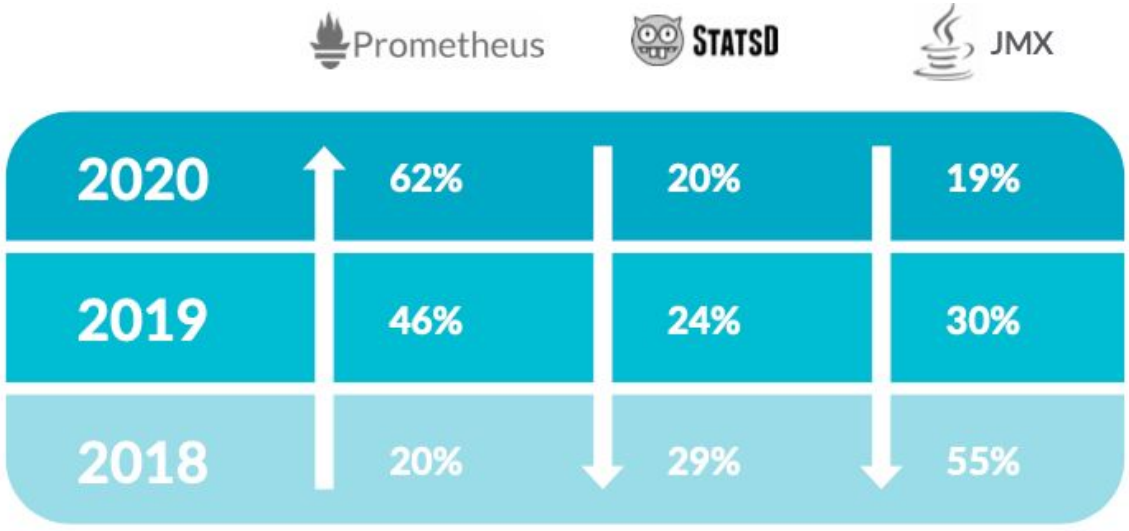
- HTTPサーバーとリバースプロキシソリューション - NGINX
- NoSQL、リレーショナル、インメモリデータベースソリューション
- MongoDB、Postgres、Redis
- ロギングとデータ分析 - Elasticsearch
- プログラミング言語とフレームワーク - node.js, Go, Java/JVM
- メッセージブローカーソフト - RabbitMQ

オープンソースコミュニティで利用可能な幅広い選択肢があることを考えると、私たちのリストにあるサービスが過去3年間でかなり一貫していることは驚くべきことです。我々は、Falcoと同様にetcdやfluentdのようなKubernetesコンポーネントを意図的に省略しました。これらはデフォルトでデプロイされているので、すべてのKubernetesユーザーにとってリストの一番上で終わってしまいます。昨年は、Node.jsとGo（別名golang）の両方がJavaの使用を追い越しました。今年は、Goの使用率が14%から66%に急上昇し、470%の増加となりました。Googleのエンジニアによって作られたGoは、急速にクラウドネイティ

ブアプリケーションを開発するための言語として選ばれるようになってきています。上記のトップ10のソリューションは、広くデプロイされ、信頼されているサービスです。同様のサービスを市場に求めているのであれば、これらのオープンソースソリューションが提供するサービスを利用することに越したことはありません。しかし、利用可能なソフトウェアソリューションはロングテールです。

カスタムメトリクス

カスタム・メトリクス・ソリューションは、開発者やDevOpsチームがコードを計測して独自のメトリクスを収集する方法を提供します。このアプローチは、本番環境のクラウド環境でアプリケーションを監視するための一般的な方法となっています。JMX、StatsD、Prometheusの3つの主力ソリューションのうち、2年連続で獲得したのはPrometheusでした。Prometheusのメトリクスの使用率は、昨年の46%に対し、前年比で62%に増加しました。新しいプログラミングフレームワークの使用が拡大しているため、JMXメトリクス（Javaアプリ用）やStatsDのような代替メトリクスは、前年比でそれぞれ35%と15%減少しており、引き続き減少しています。



Metric types in use on average

Prometheusのトップエクスポーター

CNCFから生まれた最も成功したオープンソースプロジェクトの1つであるPrometheusは、クラウドネイティブモニタリングの代名詞となっています。現在では、Kubernetes、OpenShift、Istioなどのプロジェクトでメトリクス標準として広く採用されています。また、さまざまなサードパーティ製ソリューションにメトリクス出力を提供する「エクスポーター」も増えてきています。特にSysdigが大規模環境向けにPrometheusの完全な互換性を提供するようになったことから、顧客ベース内でのPrometheusの人気は今後も高まると予想しています。

このランキングのために、prometheus.ioにリストアップされている各githubプロジェクトを調べ、各プロジェクトの課題数、スター数、フォーク数を測定し、結果をDockerhubや他のリポジトリのプル数と相関させました。

"PrometheusはSysdigとの組み合わせで、スクレイピングをニーズに合わせて適応させることができ、重要なものをフィルタリングすることができます。その上、PromQLのおかげで、ユーザーに強力な可視化とアラートを提供することができ、監視システムを大幅に改善してくれました。

- マリオ・シムコ、SAP Concur オブザーバビリティチームリーダー

Top 10 Prometheus Exporters

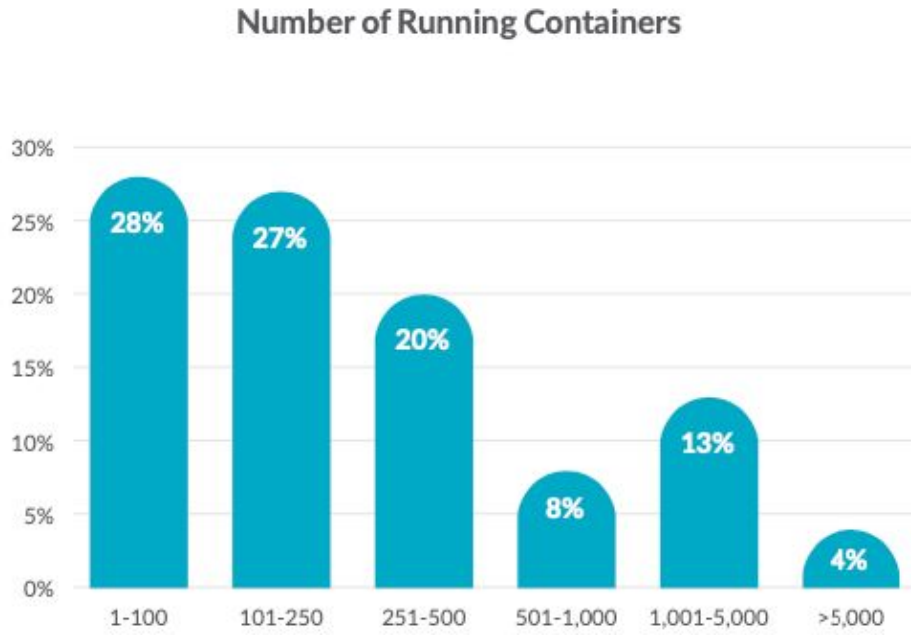
Name	Maintainer
node_exporter	prometheus
blackbox_exporter	prometheus
jmx_exporter	prometheus
redis_exporter	oliver006
windows_exporter	prometheus
postgres_exporter	wrouesne
elasticsearch_exporter	justwatchcom
mysqld_exporter	prometheus
snmp_exporter	prometheus
kafka_exporter	danielqsi

コンテナ

年毎に、密度や寿命など、コンテナの数や活動に特有の詳細を見ていきます。これにより、採用率についての洞察を得ることができるだけでなく、達成されている規模や効率性についても説明しています。

組織ごとのコンテナ

企業が現在運用している規模を把握するために、各顧客がインフラストラクチャーで運用しているコンテナの数を調べてみました。半数以上の顧客が250個以下のコンテナを運用しています。10% ハイエンドでは、5,000個以上のコンテナを管理している顧客はわずか4%でした。小規模から導入するのが一般的で、ソフトウェアのデリバリーを加速する手段としてコンテナ化を推進する開発者から生まれることもあります。DevOpsやクラウドチームの報告によると、メリットが証明されると、新しいプラットフォームを導入しようとするビジネスユニットが増えるにつれ、採用が加速します。しかし、稼働しているコンテナの生の数は、コンテナのサイズ（右図参照）とともに考慮に入れる必要があります。



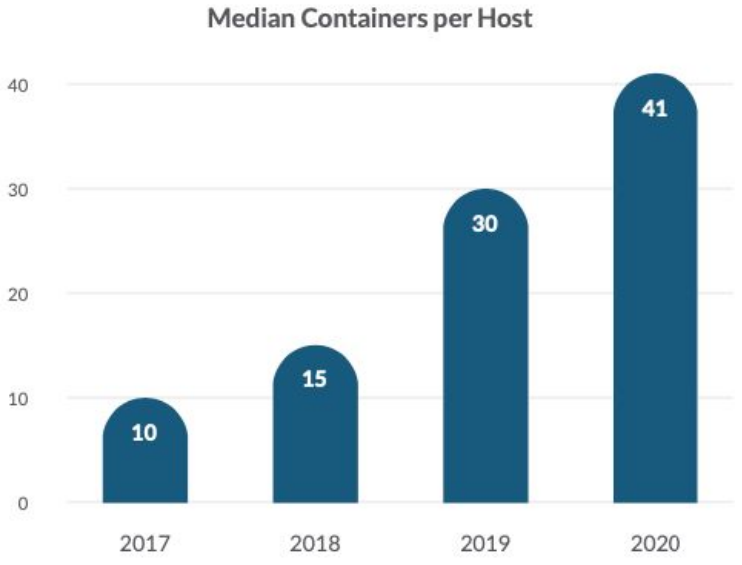
イメージのサイズはどのくらいですか？

イメージサイズはアプリケーションによって異なりますが、私たちのデータによると、平均的なイメージサイズは376MBです。10GBの大きなAlpineイメージは例外的なものと思われます。大きなイメージは、デプロイに時間がかかり、リリース速度が遅くなるだけでなく、攻撃の機会を増やすことにもなります。

コンテナ密度

ホストあたりのコンテナ密度が33%増加!

過去4年間、ホストあたりのコンテナ数の中央値はすべてのレポートで増加していました。しかし、昨年の100%の増加に対し、今年は前年比33%の増加にとどまりました。今後も微増が続く可能性はありますが、その密度は全体のイメージサイズを犠牲にしていると思われます。コンテナの第一の目的は開発とデプロイのスピードアップですが、コンテナの効率化によりハードウェアリソースの利用率が向上したことで、多くの組織が恩恵を受けています。



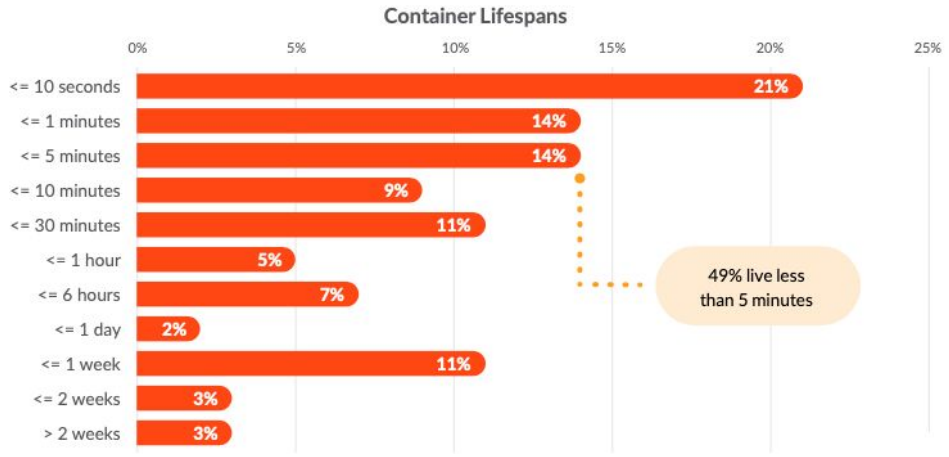
コンテナ、イメージ、サービスの寿命

コンテナ、コンテナイメージ、およびサービスがどれだけ長く（またはどれだけ短く）生きているかの指標は、2019年のレポートで最も人気のあるデータポイントの1つでした。これは、開発とランタイムの両方の観点から、現代のアプリケーションがいかにダイナミックであるかを反映しています。

コンテナは短命

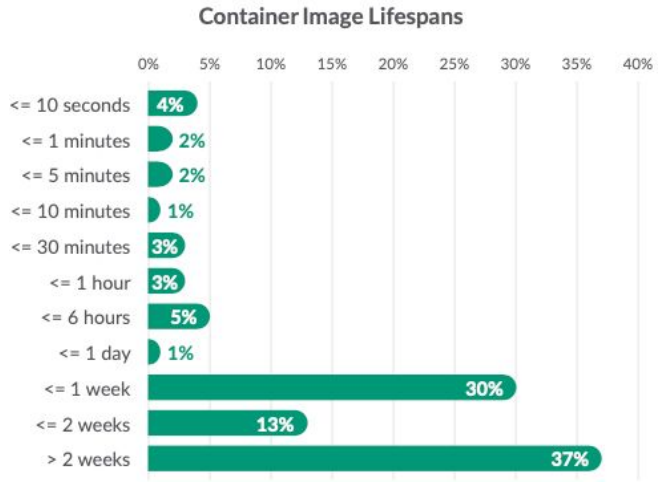
コンテナの寿命を前年比で比較すると、コンテナの大部分が1週間未満であるという同様のパターンが見られます。実際、最新のデータサンプルによると、10秒間生きているコンテナの数は、昨年の22%と比較して21%と比較的変化がありませんでした。

多くのコンテナは、関数を実行するのに十分な時間だけ生きていて、関数が完了したら終了する必要があります。数秒は短いように見えるかもしれませんが、プロセスによってはそれだけで十分なのです。コンテナの刹那的な性質は、テクノロジーのユニークな利点の1つであることに変わりはありませんが、監視、セキュリティ、コンプライアンスの面で考慮すべき新たな問題を提示しています。クラウド・サーバーレス技術の採用が進むにつれ、短命なプロセスやサービスがコンテナからホストされた機能へと移行していくことで、振り子が逆に揺れる可能性があります。しかし、これは環境で稼働しているワークロードの全体的な構成の変化を意味するものではない。むしろ、短命なワークロードがあるテクノロジーから別のテクノロジーへと移行する可能性があります。



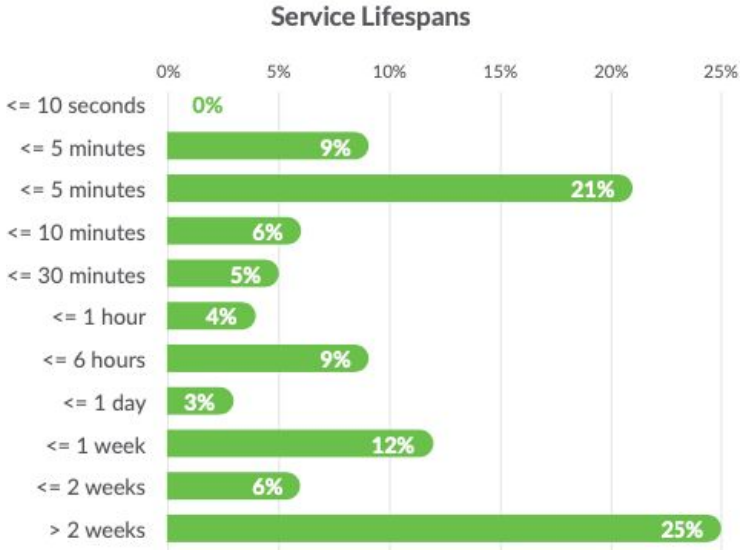
継続的な開発とイメージの寿命

コンテナはアジャイルな動きに最適なコンパニオンであり、コードの開発とリリースを加速し、多くの場合はコンテナ化されたマイクロサービスとして利用されています。当社のイメージの寿命データは、コードリリース間の時間のシフトと、CI/CD パイプラインが開発者チームがこれまで以上に速いペースでソフトウェアアップデートを提供するのに役立っているという現実を反映しています。このデータによると、コンテナイメージの半分以上が1週間以内に入れ替わっていることがわかります。今日のビジネスのほとんどにとって、市場投入までのスピードは重要であり、競争力を維持する上で大きな違いとなっています。コードのデプロイはより頻繁に行われるようになっており、それは新しいコンテナイメージを意味しています。コンテナは、素晴らしいアイデアを迅速に現実のものにするために必要なものをサポートします。



サービス寿命

ライフスパンに関する最後の見解として、サービスと稼働時間に関するデータを調査しました。サービス - データベース・ソフトウェア、ロードバランサー、カスタム・コードなどのアプリケーションの機能的なソフトウェア・コンポーネント - は、継続的に改善されているかもしれません。しかし、同時に、サービスを24時間稼働させ続けることは（少なくともほとんどの24時間365日のビジネスにとっては）重要です。当社の顧客サービスの半分以上が2週間以上稼働していた過去数年とは異なり、今年は特に5分未満の範囲で稼働時間にばらつきが見られました。



アラート

お客様が設定したアラートの種類に応じて傾向を分析することで、ユーザーがどのような状況にあるのかを把握し、コンテナの運用に支障をきたす可能性が高いと判断していることがわかります。

アラート条件のトップ10

現在、当社のお客様には800以上のユニークなアラート条件が使用されています。以下の図は、最も一般的に使用されているアラート条件と、それぞれを使用している顧客の割合を示しています。これらのアラートの構成は前回のレポートから変化しており、Kubernetesノードの可用性を重視する一方で、リソースの利用率と稼働時間にはわずかに注目していません。

アラートスコープ

Sysdigのアラートは、特定のタグやKubernetes / クラウドのラベルに「スコープ」することでカスタマイズをサポートしています。例えば、上記のアラートの例を使用して、"istio-system "のような個々のネームスペースに対してmemory.used.percentアラートを指定したり、そのネームスペース内の "envoy "のような特定のPod名に対してアラートを指定したりすることができます。タグ付けとラベル付けは、クラウドネイティブ環境では重要な役割を果たし、アイテムの整理と分離に役立つ一意の識別子を提供します。この場合、タグ付けでは "注目すべきもの "のグループを指定しています。Kubernetesのラベルでアラートを指定することは、現在では最も一般的なプラクティスの1つであり、ネームスペース、クラスター、デプロイメント、ホストが上位5位にランクインしています。エージェントタグ - デプロイ時にSysdigエージェントに添付されるメタデータ - は、Sysdigユーザー全体で最もポピュラーなアラートスコープになっています。

2019

Scope label type	% of users
Kubernetes namespace	94%
Agent tags	78%
Kubernetes cluster	76%
Kubernetes Deployment	71%
Kubernetes Pod	38%

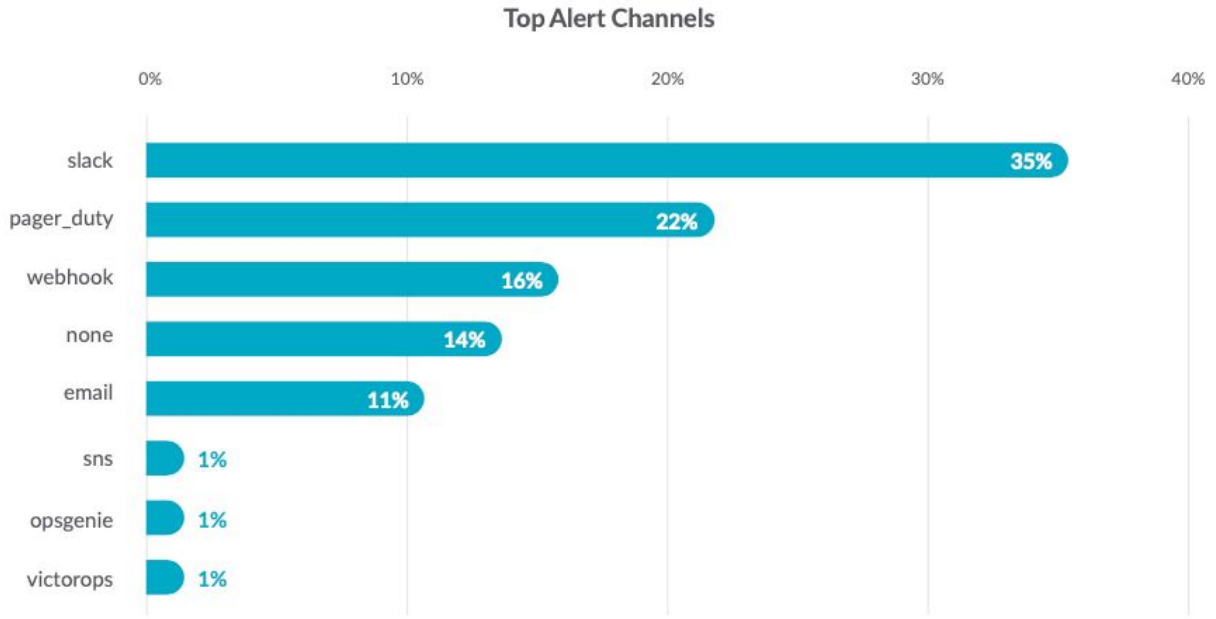
2020

Scope label type	% of users
Agent tags	88%
Kubernetes namespace	75%
Kubernetes cluster	52%
Kubernetes Deployment	37%
Host	31%

下の言葉の雲は、他のタグやラベルの内訳を示していますが、お客様がアラートのスコープに使用していたものです。

アラートチャネル

ユーザーがアラートを受信するために設定したコミュニケーションチャネルを調べました。 Slackがトップの座を獲得し、専用のインシデント対応プラットフォームやEメールさえも上回っています。 例えば、PagerDutyやOpsgenieとは異なり、Slackはインシデント対応プラットフォームとはみなされていないため、この結果は興味深いものでした。 PagerDutyのようなソリューションは "ベッドから人を起こす "ために使用されているのに対し、Slackは就業時間中に処理される非クリティカルなアラートに使用されている可能性があります。今年は、通知チャネルが設定されていないアラートのカテゴリーを含めることにしました。 これは、アラートが情報提供のみを目的としたものであったためか、Sysdigのプラットフォーム自体が問題のアラートの要求を満たすのに十分な情報を提供していたためかもしれません。



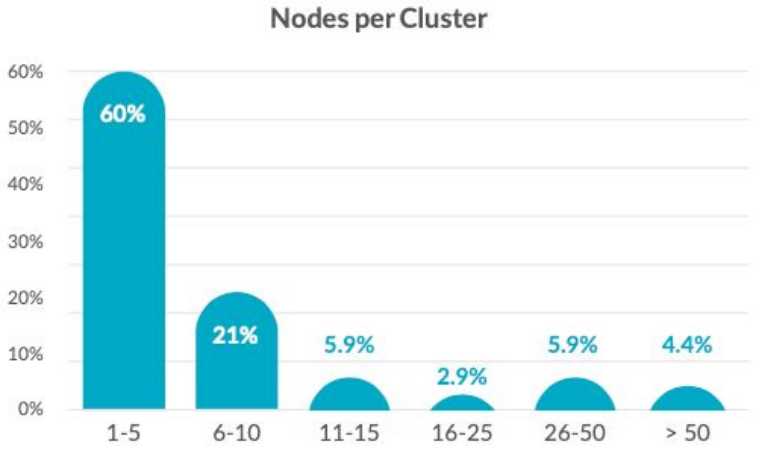
Kubernetesの利用パターン

顧客は何台のクラスターを運用しているのか？ ノードごとに稼働しているポッドの数は？ このセクションでは、これらの質問などに答えます。 クラスターからReplicaSetsまで、顧客がKubernetesを使って何をしているのか、様々な詳細を見ていきます。 SysdigはKubernetesのラベルとメタデータを自動的に収集しているため、パフォーマンスメトリクスやアラートからセキュリティイベントまで、発見したすべてのデータインサイトにクラウドネイティブなコンテキストを提供することができます。この同じ機能により、クラスターからポッドやコンテナに至るまで、次のような使用メトリクスを簡単なクエリーで取得することができます。

Kubernetesのクラスターとノード

顧客の中には、いくつかのクラスタ（小規模なものもあれば、大規模なものもあります）を維持している人もいれば、さまざまな規模の多数のクラスターを保有している人もいます。 右のグラフは、Sysdigプラットフォームを利用しているユーザーのクラスター数とクラスターあたりのノード数の分布を示しています。 顧客ごとの単一クラスターの数が多く、ノード数が比較的少ないことは、多くの企業がまだ Kubernetes の使用には早いことを示しています。 また、パブリッククラウドでのマネージドKubernetesサービスの利用も、これらのデータポイントに影響を与えるもう一つの要因であると認識しています。 Amazon Elastic Kubernetes Service (EKS)、Google Kubernetes Engine (GKE)、Azure Kubernetes Service (AKS)、IBM Cloud Kubernetes Service (IKS)などのサービスを利用すれば、ユーザーは必要に応じて迅速にクラスタをスピンアップしたり、ティアダウンしたりすることができます。


Number of Clusters
60%
50%
40%
30%
20%
10%
0%
60%
19%
15%
4.4%
1.5%
1
2
3-5
6-10
11-25


Nodes per Cluster
60%
50%
40%
30%
20%
10%
0%
60%
21%
5.9%
2.9%
5.9%
4.4%
1-5
6-10
11-15
16-25
26-50
> 50

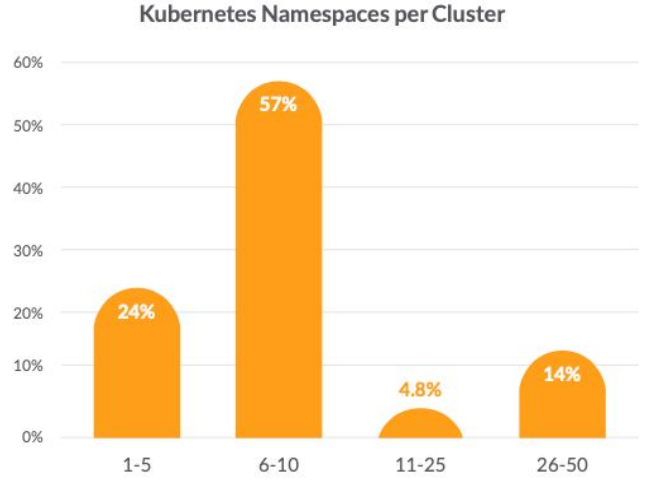
Kubernetes ネームスペース、デプロイメント、およびポッド

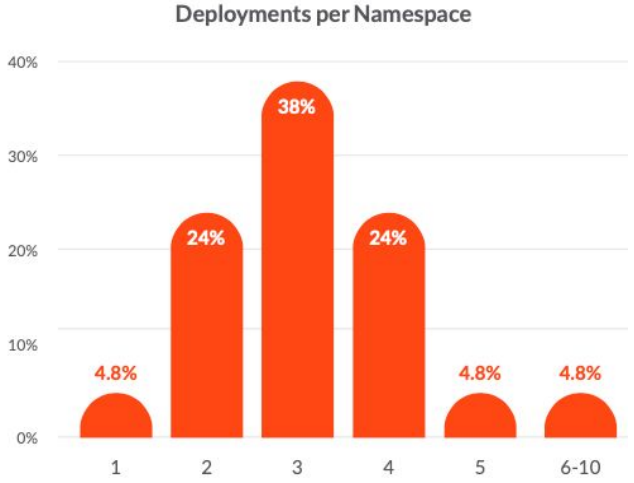
クラスターごとのネームスペース

Kubernetes ネームスペースは論理的な分離を提供し、複数のユーザー、チーム、またはアプリケーション間でクラスターリソースを整理するのに役立ちます。 Kubernetesは、default、kub-system、kubepublicの3つの初期ネームスペースから始まります。 ネームスペースの使用方法は組織によって異なりますが、クラウドチームではアプリケーションごとに固有のネームスペースを使用するのが一般的です。

ネームスペースごとのデプロイメント

デプロイメントは、ポッドとReplicaSetsの望ましい状態を記述し、アプリケーションの1つ以上のインスタンスがユーザーのリクエストに対応できるようにするのに役立ちます。 デプロイメントは、NGINX、Redis、Tomcatのデプロイメントのように、一意の識別子を持たない複数の同一のポッドのセットを表します。ネームスペースごとのデプロイメントの数は、ユーザーのマイクロサービスアプリケーションを構成するサービスの数を示しています。今年は、ネームスペースごとのディプロイメント数の減少に向けて若干のシフトが見られました。ネームスペースごとに環境へのアクセスをセグメント化するのは最も簡単ですが、各ネームスペースでのデプロイ数を減らすことで、チームが担当するアプリケーションへのアクセスを制限するために、より多くのセグメント化を行うことができます。



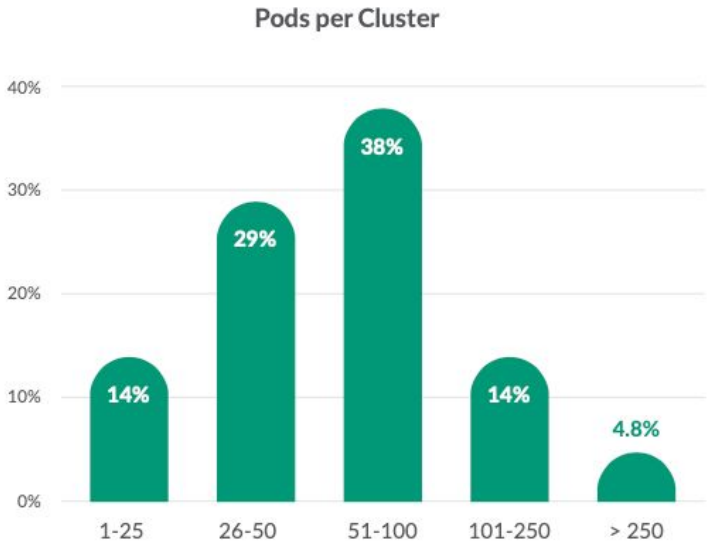


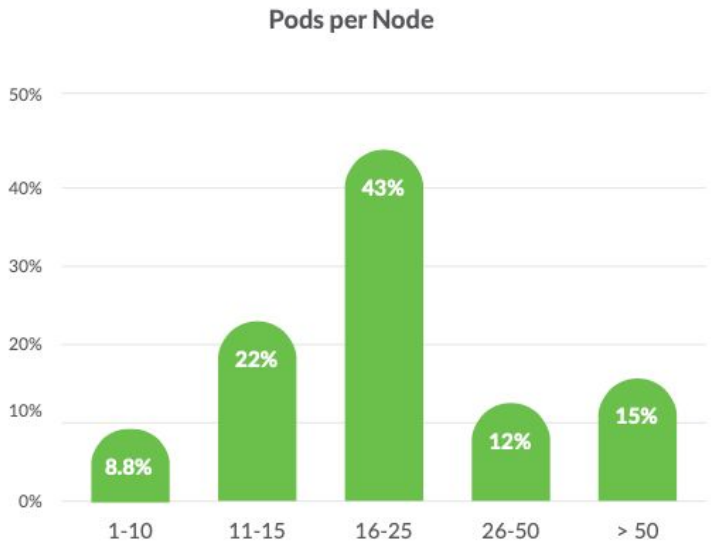
クラスターごとのポッド

ポッドはKubernetesの中で最も小さなデプロイ可能なオブジェクトです。 ポッドには、ストレージとネットワークを共有する1つ以上のコンテナと、コンテナを実行する方法の仕様が含まれています。

ノードごとのポッド

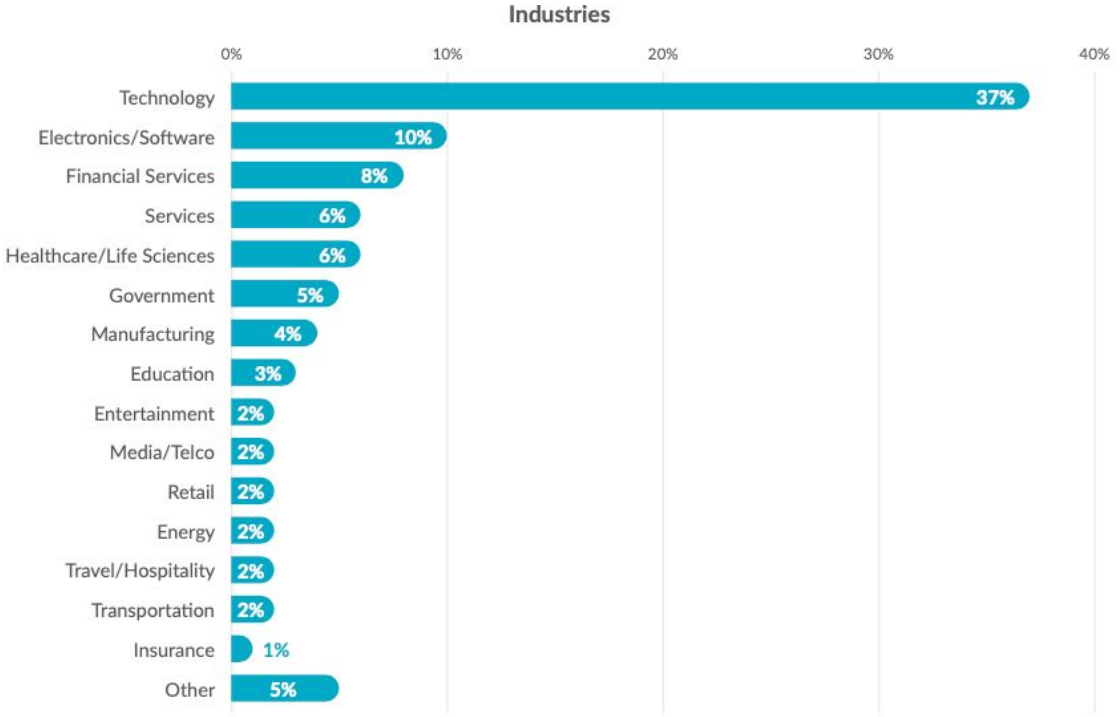
ポッドは、処理が完了するまでノード上に残り、ポッドが削除されたり、リソースが不足してポッドがノードから追い出されたり、ノードが失敗したりします。

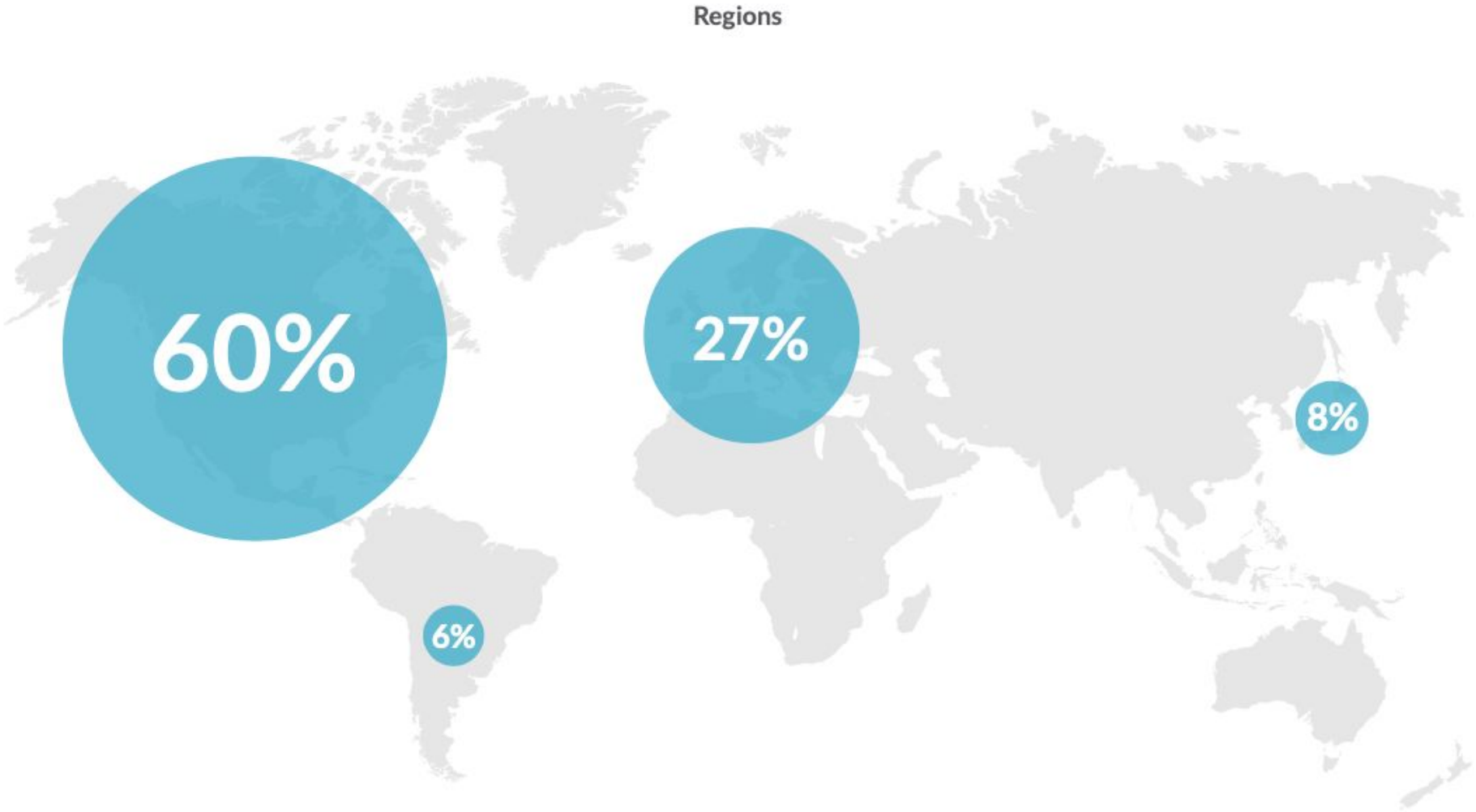




統計とデータソース

本レポートのデータは、お客様が日常的に稼働させているコンテナのサブセットである約200万個のコンテナの分析から得られたものです。 また、GitHub、Docker Hub、CNCFなどの公開データソースからも抽出しています。 このデータは、中堅から大企業までの組織規模を持つ、幅広い産業や地域に渡るコンテナのデプロイメントから得られたものです。




Regions
60%
27%
8%
6%

まとめ

コンテナ技術は、組織がアプリケーションを提供する方法を変革する上で、その役割を拡大し続けています。DevOpsチームの間ではKubernetesのセキュリティに対する関心が高まっており、チームがビルドプロセス中にセキュリティを実装しているのは良いことです。 しかし、可能性のある脆弱性が本番環境に侵入するのを防ぐためには、コンテナ自体のセキュリティを確保するための作業がより多く必要とされています。第4回年次レポートの主な傾向は、コンテナ環境の継続的な成長と、コンテナ環境を実行するためのオープンソースベースのソリューションへの依存度が高まっていることを強調しています。

- 組織は、ビルドフェーズでイメージをスキャンし、インラインスキャンを活用することで、Kubernetes環境をシフトレフトしています。 脆弱性は依然として盛んであり、特定、監査、コンプライアンスの検証を行うためには、堅牢なセキュリティツールが必要とされています。
- Kubernetesは依然として明確なオーケストレーターとして選択されていますが、containerdとCRI-Oの急速な成長に伴い、コンテナランタイムの選択はシフトしています。 コンテナの密度が高まる中、組織はKubernetesネイティブツールに投資して、スケールでの運用を簡素化する必要があります。
- オープンソースは、Kubernetes環境のコアコンポーネントとして成長しています。 Falco、Prometheus、Goの成長は、短命なコンテナの確保と監視の問題を解決するための高品質なオープンソースソリューションへの欲求を示しています。

Sysdig 2021 コンテナセキュリティと使用状況レポートをお読みいただきありがとうございます。来年のコンテナ市場の進化を追いかけ、ドキュメント化することを楽しみにしています。 それではまたお会いしましょう!

30日間の無料トライアルにサインアップして、
セキュリティ、コンプライアンス、モニタリングの可視性を数分で手に入れましょう！
https://sysdig.jp/company/free-trial/

www.sysdig.jp