sysdig

AWS Fargate チェックリストでコンテナを保護する

本文の内容は、ホワイトペーパー「Securing Containers on AWS Fargate Checklist」を元に日本語に翻訳・再構成した内容となっております。

AWS Fargateチェックリストでコンテナを保護する 3

クラウド上のFargateコンテナを自動的にスキャンする 4

ランタイムの脅威の検出と対応 6

Fargateのポスチャーを継続的に管理する 8

継続的にコンプライアンスを検証 10

Fargateコンテナの監視とトラブルシューティング 12

AWS Fargateチェックリストでコンテナを保護する

AWS Fargateは、Amazon ECSとAmazon EKSの両方で動作する、人気の高いコンテナ用サーバーレスコンピュートエンジンです。Fargateは、アプリケーションの構築に集中することを容易にします。サーバーをプロビジョニングして管理する必要がなくなり、アプリケーションごとにリソースを指定して支払うことができます。サーバーレス環境により、最新のアプリケーション開発に集中できるようになる一方で、いくつかの課題もあります。サーバーレス環境では、基盤となるインフラストラクチャーをDevOpsチームやセキュリティチームから隠す抽象化レイヤーが導入されます。ホストや従来の監視ツールにアクセスできないと、ワークロードのアクティビティに対する可視性が制限され、脅威を発見できなくなる可能性があります。セキュリティやパフォーマンスの問題を特定したら、チームはインシデントに対応し、問題を解決するためにアクティビティの詳細な記録を必要とします。

基盤となるインフラが見えない

脆弱性や脅威を警告できない

Fargateタスクがなくなった後、何が起こったのかを正確に記録する必要がある

重要なのは、クラウドとコンテナの脆弱性を継続的にスキャンし、異常なアクティビティを検出し、クラウドの設定ミスによるリスクを減らし、脅威に優先順位をつけて、Fargate上のコンテナがライフサイクル全体で安全であることを保証することです。これらの5つの重要なワークフローにより、最も重要なセキュリティと可視性の要件に対処することができ、自信を持って安全にAWS Fargate上でコンテナを実行することができます。

クラウド上のFargateコンテナを自動的にスキャンする

コンテナイメージ、バージョン、ビルドの数が増えると、どのソフトウェアが使用されているか、ソフトウェアのアップデートが適用されているかどうかを管理できなくなります。アプリケーションのビルド時にデリバリーパイプラインにセキュリティを組み込むことで、脆弱性の特定と対処を迅速に行い、開発者の生産性を維持することができます。Elastic Container Registry（ECR）内でスキャンを自動的に起動することから始めて、以下の手順で継続的な管理を行うことができます。

- Fargateタスクに関連付けられたイメージを自動的にスキャンし、リスクのあるイメージがデプロイされるのを防ぎます。
- クラウドアカウントのイメージを直接スキャンすることで、機密データを保護します。
- ビルド構成（Dockerfileの指示）とイメージの属性（サイズやラベルなど）を検証します。
- コンテナがデプロイされた後のイメージに影響を与える新しい脆弱性を特定する。
- 公開リポジトリからのイメージや社内で作成されたイメージなど、ワークフローごとに異なるポリシーを作成。
- ワークフローごとに異なるポリシーを作成する。アプリごとに異なるチェックを検討する。
- 問題ごとに適切なチームに警告する（各イメージの所有者に通知し、CI/CDツールと統合して、そのコンテキストでスキャン結果を直接表示する）。

セキュリティ分析とコンプライアンス検証をこのプロセスに統合することで、問題を早期に解決し、導入を遅らせることなく進めることができます。これは、"セキュリティのシフトレフト "と呼ばれています。

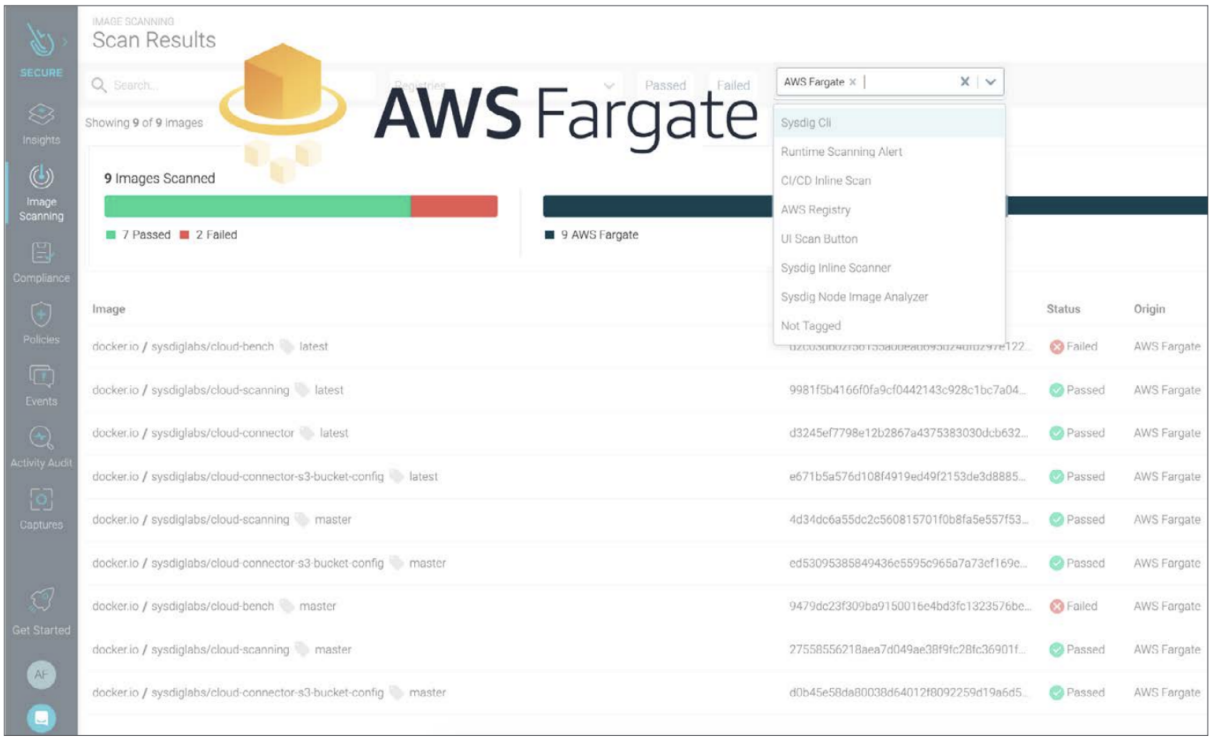


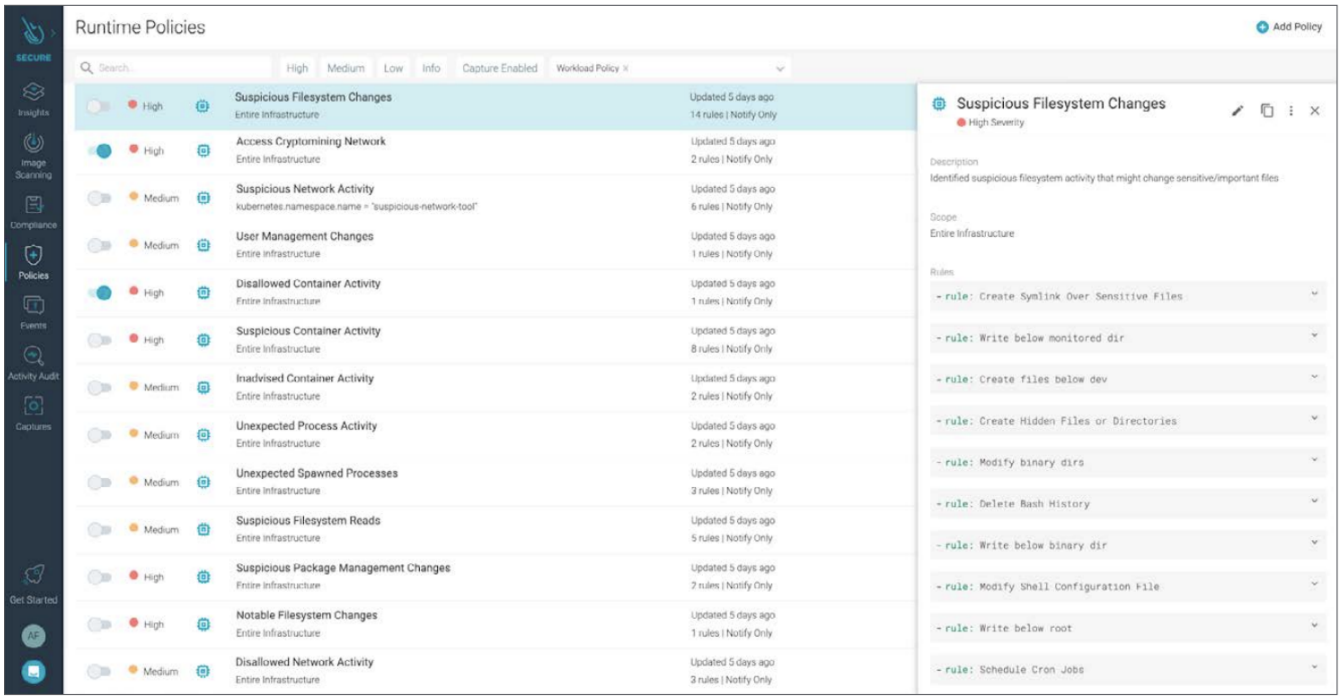
IMAGE SCANNING
Scan Results
AWS Fargate
SECURE
Insights
Image Scanning
Compliance
Policies
Events
Activity Audit
Captures
Get Started
Showing 9 of 9 images
Passed
Failed
AWS Fargate
Sysdig Cli
Runtime Scanning Alert
CI/CD Inline Scan
AWS Registry
UI Scan Button
Sysdig Inline Scanner
Sysdig Node Image Analyzer
Not Tagged
9 Images Scanned
7 Passed
2 Failed
9 AWS Fargate
Image
Status
Origin
docker.io / sysdiglabs/cloud-bench latest Failed AWS Fargate
docker.io / sysdiglabs/cloud-scanning latest 9981f5b4166f0fa9cf0442143c928c1bc7a04... Passed AWS Fargate
docker.io / sysdiglabs/cloud-connector latest d3245ef7798e12b2867a4375383030dcb632... Passed AWS Fargate
docker.io / sysdiglabs/cloud-connector-s3-bucket-config latest e671b5a576d108f4919ed49f2153de3d8885... Passed AWS Fargate
docker.io / sysdiglabs/cloud-scanning master 4d34dc6a55dc2c560815701f0b8fa5e557f53... Passed AWS Fargate
docker.io / sysdiglabs/cloud-connector-s3-bucket-config master ed53095385849436e5595c965a7a73ef169e... Passed AWS Fargate
docker.io / sysdiglabs/cloud-bench master 9479dc23f309ba9150016e4bd3fc1323576be... Failed AWS Fargate
docker.io / sysdiglabs/cloud-scanning master 2755855621 8aea7d049ae38f9fc28fc36901f... Passed AWS Fargate
docker.io / sysdiglabs/cloud-connector-s3-bucket-config master d0b45e58da80038d64012f8092259d19a6d5... Passed AWS Fargate

ランタイムの脅威の検出と対応

アプリケーションを最小限の権限とアクセス許可で構成することにより、ランタイムリスクを低減することができます。また、ポリシーでは、異常な動作や構成の変化を監視する必要があります。アプリケーションを破壊することなく攻撃を防ぐことができるポリシーを作成することは困難です。また、Fargateタスクが終了した後も、インシデント対応のために詳細な記録を取るようにしてください。ランタイムのリスクを減らすために、これらのステップを考慮してください。

- ワークロードの動作を観察し、クラウドのアクティビティを監視し、Fargateの異常なイベントを特定するランタイムセキュリティポリシーを作成、維持します。
    - ツールを活用してポリシーを自動的に構築・カスタマイズしたり、すぐに使えるFalcoルールを使用したりすることができます。
    - Kubernetesとアプリのメタデータを使用して、最小特権と準拠したネットワークポリシーを実装します。
    - トポロジーマップを使って、特定のポッド/サービス/アプリ/タグに出入りするネットワーク通信を時系列で可視化。
    - AWS CloudTrailログのイベントを自動利用して、クラウドサービス上の脅威や設定変更を検知。
- Fargateのタスク使用状況を監視し、異常なアクティビティを検出します。ネットワーク接続を追跡することで、攻撃、ランタイムの動作、および拡散ベクターに関する情報が得られます。一部の攻撃は、セキュリティ違反ではなく監視アラートとして最初に検出されます。
- インシデント対応を合理化し、Fargateの詳細な活動記録を用いてコンテナやクラウドセキュリティの脅威に迅速に対応します。マルチクラウド環境におけるクラウドとコンテナの脅威を統一的に把握できるようにします。システムコールデータに基づくキャプチャーファイルを使用して、コンテナセキュリティインシデントの「いつ」「何を」「誰が」「なぜ」の質問に素早く答えられるようにします。この詳細な記録により、コ

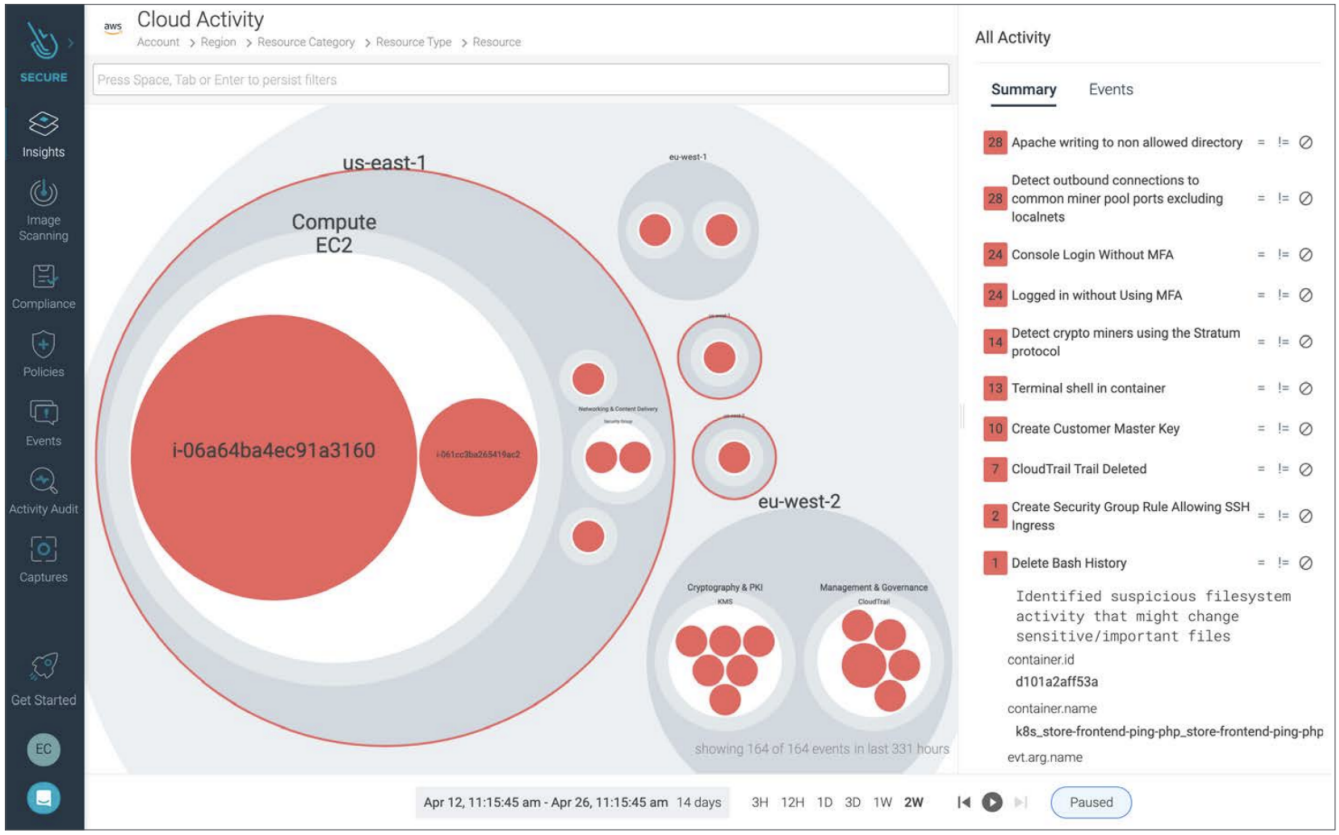
ンテナがなくなった後でも、事後分析を行い、根本原因を特定することができます。



Fargateのポスチャーを継続的に管理する

設定ミスや不審な動作を即座に発見するためには、継続的なクラウドセキュリティが必要です。共有責任モデルによると、アプリケーションとデータを保護するのはAWSユーザーの責任です。以下のステップは、Fargateワークロードのセキュリティ態勢を検証するのに役立ちます。

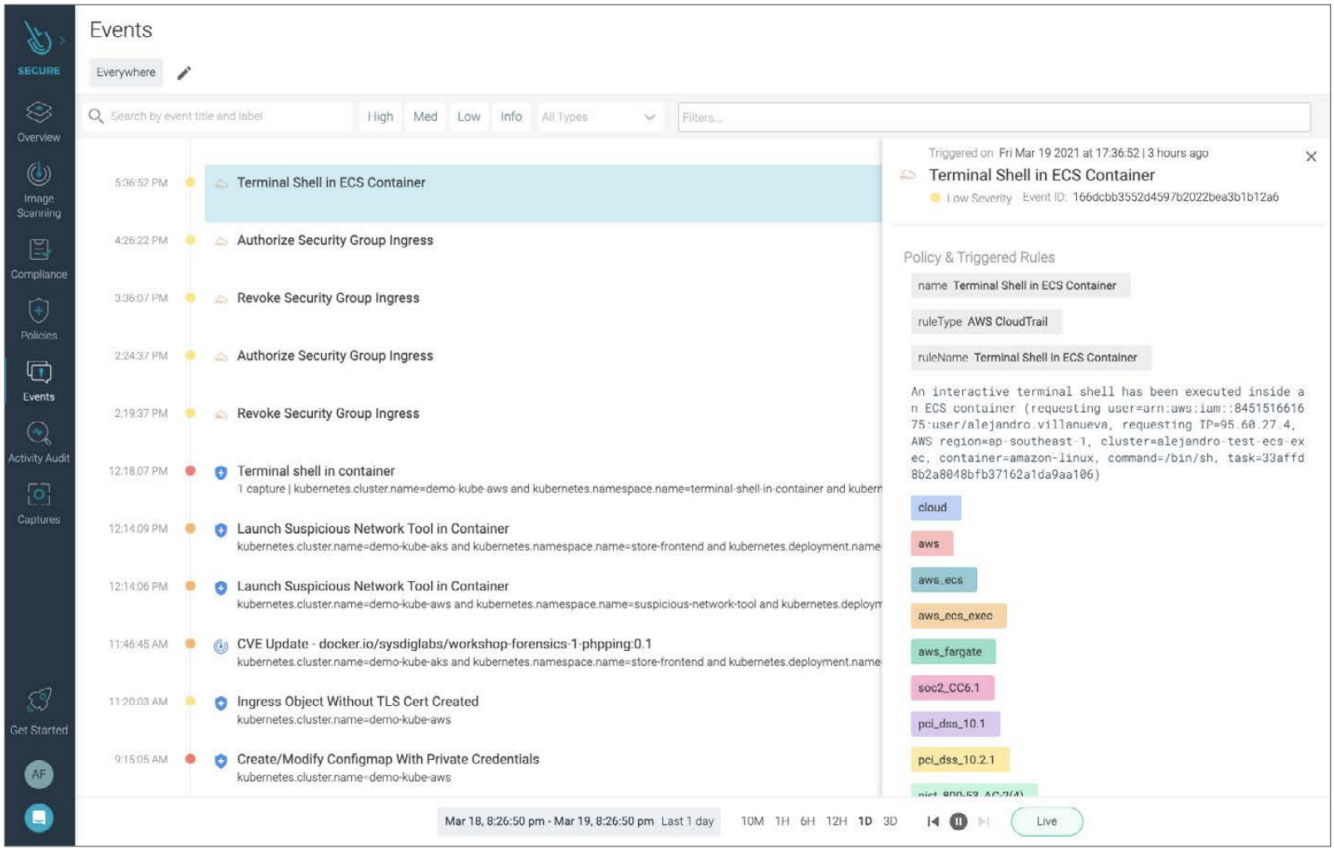
- VPC、RDS、S3バケット、ECS、EKS、Fargateなどのシステム、アプリケーション、サービスを含む環境で稼働しているAWS資産を自動的に発見し、インベントリ化します。
- オープンソースのCloud Custodianルールに基づいて、設定ミスにフラグを立てます。Center for Internet Security（CIS）のベンチマークと比較して、クラウドの構成を定期的にチェックし、リスクのある構成設定（パブリック・ストレージバケット、露出したセキュリティグループやアクセスコントロールなど）を特定し、違反を修復するための措置を講じます。
- オープンソースのFalcoルールを使用してAWS CloudTrailログを解析することで、すべてのクラウドアカウント、ユーザー、サービスにおけるセンシティブなデータへのアクセス権の変更など、予期せぬ変更や疑わしいアクティビティを検出します。特定のユーザーが行ったすべての疑わしいアクティビティを調査し、影響の範囲を確認します。


SECURE
Insights
Image Scanning
Compliance
Policies
Events
Activity Audit
Captures
Get Started
EC
Cloud Activity
Account > Region > Resource Category > Resource Type > Resource
Press Space, Tab or Enter to persist filters
us-east-1
Compute
EC2
i-06a64ba4ec91a3160
eu-west-1
eu-west-2
Cryptography & PKI
KMS
Management & Governance
CloudTrail
showing 164 of 164 events in last 331 hours
All Activity
Summary
Events
28 Apache writing to non allowed directory
28 Detect outbound connections to common miner pool ports excluding localnets
24 Console Login Without MFA
24 Logged in without Using MFA
14 Detect crypto miners using the Stratum protocol
13 Terminal shell in container
10 Create Customer Master Key
7 CloudTrail Trail Deleted
2 Create Security Group Rule Allowing SSH Ingress
1 Delete Bash History
Identified suspicious filesystem activity that might change sensitive/important files
container.id
d101a2aff53a
container.name
k8s_store-frontend-ping-php_store-frontend-ping-php
evt.arg.name
Apr 12, 11:15:45 am - Apr 26, 11:15:45 am 14 days
3H 12H 1D 3D 1W 2W
Paused

継続的にコンプライアンスを検証

コンテナ、Kubernetes、クラウド全体の規制コンプライアンス標準（CIS、SOC2、PCI、NIST 800-53など）を満たすためのコンプライアンスチェックを実施し、ベストプラクティスに対するベンチマークを行う。クラウドサービスを継続的に監視し、コンプライアンスに影響を与える可能性のある構成のドリフトを監視する。スケジュールされた評価と詳細なレポートによって、コンプライアンスの進捗を測定します。

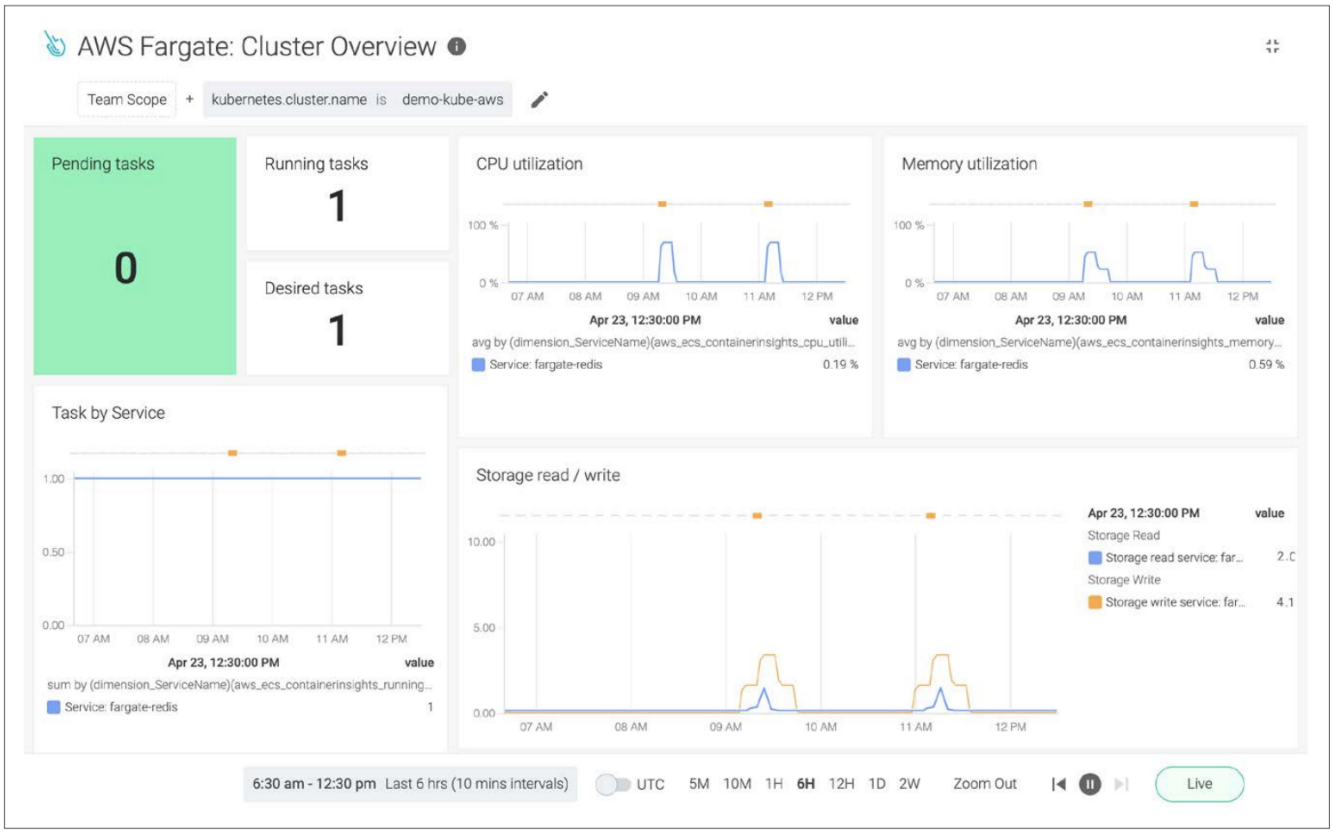
- コンテナとプラットフォームの構成を、AWS、Docker、KubernetesのCISベンチマークと比較してチェックします。
- コンテナイメージのスキャンポリシーを標準（NIST、PCI、SOC2、HIPAAなど）や内部のコンプライアンスポリシー（ブラックリストに登録されたイメージ、パッケージ、ライセンスなど）にマッピングすることで、ビルド時にコンプライアンスを検証する。
- ファイル整合性モニタリング（FIM）を導入し、重要なシステムファイルやディレクトリの改ざん、および不正な変更を検出する。FIMは、多くのコンプライアンス規格の中核となる規制要件です。
- ランタイム時にもコンプライアンスを管理します。ベストプラクティスをチェックし（特権的なコンテナを実行しない、コンテナをルートとして実行しないなど）、既知の敵の戦術や技術を探す。NIST 800-53、PCI DSS、SOC 2、MITRE ATT&CK®、CIS AWS、AWS Foundational Security Best Practicesなどのセキュリティ基準やベンチマークに対するFalcoルールの豊富なセットを通じて、セキュリティフレームワークのマッピングでコンプライアンスを達成・維持する。
- 詳細なフォレンジックデータを組み込んだキャプチャファイルで、コンプライアンスの証明を行います。コンプライアンス監査のためには、ランタイムの変更の監査を含め、構成やポリシーの変更を記録することが重要です。


Events
SECURE
Everywhere
Search by event title and label
High
Med
Low
Info
All Types
Filters...
Overview
Image Scanning
Compliance
Policies
Events
Activity Audit
Captures
Get Started
5:36:52 PM
Terminal Shell in ECS Container
4:26:22 PM
Authorize Security Group Ingress
3:36:07 PM
Revoke Security Group Ingress
2:24:37 PM
Authorize Security Group Ingress
2:19:37 PM
Revoke Security Group Ingress
12:18:07 PM
Terminal shell in container
1 capture | kubernetes.cluster.name=demo-kube-aws and kubernetes.namespace.name=terminal-shell-in-container and kubern
12:14:09 PM
Launch Suspicious Network Tool in Container
kubernetes.cluster.name=demo-kube-aks and kubernetes.namespace.name=store-frontend and kubernetes.deployment.name
12:14:06 PM
Launch Suspicious Network Tool in Container
kubernetes.cluster.name=demo-kube-aws and kubernetes.namespace.name=suspicious-network-tool and kubernetes.deploym
11:46:45 AM
CVE Update - docker.io/sysdiglabs/workshop-forensics-1-phpping:0.1
kubernetes.cluster.name=demo-kube-aks and kubernetes.namespace.name=store-frontend and kubernetes.deployment.name
11:20:03 AM
Ingress Object Without TLS Cert Created
kubernetes.cluster.name=demo-kube-aws
9:15:05 AM
Create/Modify Configmap With Private Credentials
kubernetes.cluster.name=demo-kube-aws
Triggered on Fri Mar 19 2021 at 17:36:52 | 3 hours ago
Terminal Shell in ECS Container
Low Severity Event ID: 166dcbb3552d4597b2022bea3b1b12a6
Policy & Triggered Rules
name Terminal Shell in ECS Container
ruleType AWS CloudTrail
ruleName Terminal Shell in ECS Container
An interactive terminal shell has been executed inside a n ECS container (requesting user=arn:aws:iam::845151661675:user/alejandro.villanueva, requesting IP=95.60.27.4, AWS region=ap-southeast-1, cluster=alejandro-test-ecs-exec, container=amazon-linux, command=/bin/sh, task=33affd8b2a8048bfb37162a1da9aa106)
cloud
aws
aws_ecs
aws_ecs_exec
aws_fargate
soc2_CC6.1
pci_dss_10.1
pci_dss_10.2.1
Mar 18, 8:26:50 pm - Mar 19, 8:26:50 pm Last 1 day
10M 1H 6H 12H 1D 3D
Live

Fargateコンテナの監視とトラブルシューティング

コンテナやクラウドサービスは動的であり、常に変化しています。クラウドアプリケーションの可用性を確保するためには、AWSワークロードとインフラストラクチャーの健全性とパフォーマンスを可視化することが重要です。

- クラウドネイティブなインフラ、アプリケーション、AWSサービスのために構築された監視を実装します。マイクロサービスは複数のインスタンスに分散させることができ、コンテナは異なるリージョンやマルチクラウドインフラストラクチャーで実行することができます。アプリケーションのパフォーマンスを向上させ、問題を迅速に解決するためには、コンテナ、インフラ、サービスの深い可視性と、Kubernetesとクラウドのコンテキストで強化された粒度の細かいメトリクスが必要です。
    - コンテナとクラウドのコンテキストを使用して、問題解決のためのオーナーを即座に特定します。-
    - 過剰なリソースを消費しているポッドを特定し、キャパシティリミットを監視します。
- オープンソースのPrometheusを利用して、AWSサービスやクラウドネイティブアプリケーションを監視。PrometheusのメトリクスをAWS CloudWatch for Fargate経由で抽出し、Grafanaのダッシュボードで表示します。
    - Promcat.ioは、Kubernetesプラットフォームやクラウドネイティブサービスの監視・インテグレーションを厳選し、文書化し、サポートしているPrometheusインテグレーションのリソースカタログで、生産性を高めることができます。
- データを取得して保存することで、問題の調査と解決を迅速に行うことができます。コンテナが死ねば、中のものはすべてなくなります。停止したコンテナにシェルで入って何が起こったかを確認することはできません! 監査ログや詳細なアクティビティ情報があれば、すでに稼働していないコンテナであっても、根本的な原因をうまく特定することができます。


AWS Fargate: Cluster Overview
Team Scope
kubernetes.cluster.name is demo-kube-aws
Pending tasks
0
Running tasks
1
Desired tasks
1
CPU utilization
100 %
0 %
07 AM 08 AM 09 AM 10 AM 11 AM 12 PM
Apr 23, 12:30:00 PM value
avg by (dimension_ServiceName)(aws_ecs_containerinsights_cpu_utili...
Service: fargate-redis 0.19 %
Memory utilization
100 %
0 %
07 AM 08 AM 09 AM 10 AM 11 AM 12 PM
Apr 23, 12:30:00 PM value
avg by (dimension_ServiceName)(aws_ecs_containerinsights_memory...
Service: fargate-redis 0.59 %
Task by Service
1.00
0.50
0.00
07 AM 08 AM 09 AM 10 AM 11 AM 12 PM
Apr 23, 12:30:00 PM value
sum by (dimension_ServiceName)(aws_ecs_containerinsights_running...
Service: fargate-redis 1
Storage read / write
10.00
5.00
0.00
07 AM 08 AM 09 AM 10 AM 11 AM 12 PM
Apr 23, 12:30:00 PM value
Storage Read
Storage read service: far... 2.0
Storage Write
Storage write service: far... 4.1
6:30 am - 12:30 pm Last 6 hrs (10 mins intervals)
UTC 5M 10M 1H 6H 12H 1D 2W Zoom Out
Live

Sysdigがどのようにして AWS Fargate上のコンテナを

継続的にセキュアにするかを深く掘り下げましょう！

プラットフォームの詳細についてはお問い合わせください。

---

sysdig.jp