Red Hat Advanced Cluster Security for Kubernetes 4.5

運用

Red Hat Advanced Cluster Security for Kubernetes の運用

Last Updated: 2024-09-16

Red Hat Advanced Cluster Security for Kubernetes 4.5 運用

Red Hat Advanced Cluster Security for Kubernetes の運用

法律上の通知

概要

このドキュメントでは、ダッシュボードの使用、コンプライアンスの管理、セキュリティーリスクの評価、セキュリティーポリシーおよびネットワークポリシーの管理、イメージの脆弱性検査、違反への対応など、Red Hat Advanced Cluster Security for Kubernetes で一般的な操作タスクを実行する方法を説明します。

目次

第1章 ダッシュボードの表示 5
1.1. ステータスバー 5
1.2. ダッシュボードフィルター 5
1.3. ウィジェットのオプション 5
1.4. 操作可能なウィジェット 6

第2章 RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES での COMPLIANCE OPERATOR の使用 8
2.1. COMPLIANCE OPERATOR のインストール 8
2.2. SCANSETTINGBINDING オブジェクトの設定 9

第3章 コンプライアンスの管理 11
3.1. コンプライアンス機能の概要 11
3.2. ワークロードとクラスターのコンプライアンスの監視 12
3.3. コンプライアンススキャンのスケジュールとプロファイルコンプライアンスの評価 (テクノロジープレビュー) 18

第4章 セキュリティーリスクの評価 26
4.1. リスクビュー 26
4.2. リスクビューからのセキュリティーポリシーの作成 26
4.3. リスクの詳細の表示 30
4.4. デプロイの詳細タブ 31
4.5. プロセス検出タブ 32
4.6. プロセスベースラインの使用 34

第5章 アドミッションコントローラーの適用の使用 37
5.1. アドミッションコントローラーの適用について 37
5.2. アドミッションコントローラーの適用の有効化 38
5.3. アドミッションコントローラーの適用の回避 39
5.4. アドミッションコントローラーの適用の無効化 39
5.5. VALIDATINGWEBHOOKCONFIGURATION YAML ファイルの変更 41

第6章 セキュリティーポリシーの管理 43
6.1. デフォルトのセキュリティーポリシーの使用 43
6.2. 既存のセキュリティーポリシーの変更 44
6.3. ポリシーカテゴリーの作成と管理 45
6.4. カスタムポリシーの作成 46
6.5. セキュリティーポリシーの共有 73

第7章 デフォルトのセキュリティーポリシー 75
7.1. 重大度のセキュリティーポリシー 75
7.2. 重大度の高いセキュリティーポリシー 76
7.3. 重大度が中程度のセキュリティーポリシー 80
7.4. 重大度の低いセキュリティーポリシー 85

第8章 ネットワークポリシーの管理 89
8.1. ネットワークグラフ 89
8.2. ネットワークグラフを使用したネットワークポリシーの生成およびシミュレート 96
8.3. ネットワークグラフでのネットワークベース化について 101

第9章 ビルド時のネットワークポリシーツール 104
9.1. ビルド時のネットワークポリシージェネレーターの使用 104
9.2. ROXCTL NETPOL CONNECTIVITY MAP コマンドを使用した接続マッピング 106
9.3. プロジェクトバージョン間での許可される接続の違いの確認 109

第10章 リスニングエンドポイントの監査 **113**

第11章 クラスター設定の確認 **114**
11.1. CONFIGURATION MANAGEMENT ビューの使用 114
11.2. KUBERNETES ロールの設定ミスの特定 114
11.3. KUBERNETES シークレットの表示 115
11.4. ポリシー違反の検索 116
11.5. 失敗した CIS コントロールの検索 116

第12章 イメージの脆弱性の調査 **118**
12.1. RHACS SCANNER V4 について 119
12.2. イメージのスキャン 120
12.3. イメージスキャン委譲へのアクセス 124
12.4. スキャンの設定 127
12.5. 脆弱性について 128
12.6. 言語固有の脆弱性スキャンの無効化 129
12.7. 関連情報 129

第13章 イメージの署名の確認 **130**
13.1. 署名統合の設定 130
13.2. ポリシーでの署名検証の使用 131
13.3. 署名の検証の実施 132

第14章 脆弱性の管理 **133**
14.1. 脆弱性管理の概要 133
14.2. 脆弱性の確認と対処 135
14.3. 脆弱性レポート 151
14.4. 脆弱性管理ダッシュボードの使用 (非推奨) 157
14.5. RHCOS ノードホストのスキャン 164

第15章 違反への対応 **170**
15.1. 違反ビュー 170
15.2. 違反の詳細の表示 171

第16章 デプロイメントコレクションの作成と使用 **176**
16.1. 前提条件 176
16.2. デプロイメントコレクションについて 176
16.3. デプロイメントコレクションへのアクセス 179
16.4. デプロイメントコレクションの作成 179
16.5. コレクションへのアクセススコープの移行 181
16.6. API を使用したコレクションの管理 182

第17章 検索およびフィルタリング **184**
17.1. 検索構文 184
17.2. オートコンプリートの検索 185
17.3. グローバル検索の使用 185
17.4. ローカルページのフィルタリングの使用 186
17.5. 一般的な検索クエリー 186
17.6. 属性の検索 188

第18章 ユーザーアクセスの管理 **193**
18.1. RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES での RBAC の管理 193
18.2. PKI 認証の有効化 202
18.3. 認証プロバイダーについて 204
18.4. アイデンティティープロバイダーの設定 206

18.5. 管理者ユーザーの削除 218
18.6. 短期間のアクセス権の設定 219
18.7. マルチテナンシーについて 221

第19章 システムヘルスダッシュボードの使用 224
19.1. システムヘルスダッシュボードの詳細 224
19.2. 製品の使用状況データの表示 225
19.3. RHACS ポータルを使用した診断バンドルの生成 226

第20章 管理イベントページの使用 228
20.1. 異なるドメインのイベントログにアクセスする 228
20.2. 管理イベントページの概要 228
20.3. 特定のドメインのイベントに関する情報を取得する 229
20.4. 管理イベントの詳細の概要 229
20.5. 管理イベントの有効期限の設定 230

第1章 ダッシュボードの表示

Red Hat Advanced Cluster Security for Kubernetes (RHACS) ダッシュボードを使用すると、必要なデータに素早くアクセスできます。追加のナビゲーションショートカットと、簡単にフィルタリングおよびカスタマイズできるアクション可能なウィジェットのパネルが含まれているため、最も重要なデータに集中できます。環境内のリスクレベル、コンプライアンスステータス、ポリシー違反、一般的な脆弱性と露出 (CVE) に関する情報をイメージで表示できます。

注記

初めて RHACS ポータルを開くと、空のダッシュボードが表示される場合があります。Sensor を少なくとも 1 つのクラスターにデプロイすると、ダッシュボードに環境のステータスが反映されます。

以下のセクションでは、Dashboard コンポーネントを説明します。

1.1. ステータスバー

Status Bar には、ひと目で把握できる主要なリソースの数値カウンターがあります。カウンターには、ユーザープロファイルに関連付けられたロールで定義された現在のアクセススコープで表示できるものが反映されます。これらのカウンターはクリック可能で、以下のように必要なリストビューページに迅速にアクセスできます。

カウンター	宛先
クラスター	Platform Configuration → Clusters
ノード	Configuration Management → Application & Infrastructure → Nodes
Violations	Violations のメインメニュー
デプロイメント	Configuration Management → Application & Infrastructure → Deployments
イメージ	Vulnerability Management → Dashboard → Images
シークレット	Configuration Management → Application & Infrastructure → Secrets

1.2. ダッシュボードフィルター

ダッシュボードには、すべてのウィジェットに同時に適用されるトップレベルフィルターが含まれるようになりました。1 つ以上のクラスターと、選択したクラスター内の 1 つ以上の namespace を選択できます。クラスターまたは namespace が選択されていない場合、表示が自動的に **All** に切り替わります。フィルターへの変更はすべてのウィジェットで即座に反映され、データの表示は選択されたスコープに制限されます。ダッシュボードフィルターは **Status Bar** には影響しません。

1.3. ウィジェットのオプション

一部のウィジェットは、特定のデータにフォーカスできるようにカスタマイズ可能です。ウィジェットにはさまざまな制御があり、データのソート、データのフィルター、ウィジェットの出力のカスタマイズに使用できます。

ウィジェットでは、さまざまな側面をカスタマイズする 2 つの方法を使用できます。

- **Options** メニュー (存在する場合) は、そのウィジェットに適用される特定のオプションを提供します。
- **dynamic axis legen**(存在する場合) を使用すると、1 つ以上の軸カテゴリーを非表示にしてデータをフィルタリングできます。たとえば、**Policy violations by category** ウィジェットでは、重大度をクリックして、データから選択した重大度の違反を包含または除外できます。

注記

個々のウィジェットのカスタマイズ設定は有効期間が短く、ダッシュボードを離れるとシステムのデフォルトにリセットされます。

1.4. 操作可能なウィジェット

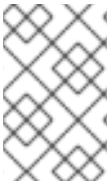
以下のセクションでは、ダッシュボードにある操作可能なウィジェットを説明します。

1.4.1. 重大度別のポリシー違反

このウィジェットでは、ダッシュボードでフィルタリングされたスコープの重大度レベル全体における違反の分布が表示されます。チャートで **severity level** をクリックすると、その重大度およびスコープでフィルタリングされた **Violations** ページに移動します。また、ダッシュボードのフィルターで定義したスコープ内で、**Critical** レベルのポリシーに対する再審の違反 3 件がリスト表示されます。特定の違反をクリックすると、その違反の **Violations** 詳細ページに直接移動します。

1.4.2. 最もリスクの高いイメージ

このウィジェットでは、ダッシュボードでフィルター処理されたスコープ内の上位 6 つの脆弱なイメージが、計算されたリスクの優先度と、それらに含まれる重大および重要な CVE の数で並べ替えて一覧表示されます。イメージ名をクリックすると、**Vulnerability Management** の **Image Findings** ページに直接移動します。**Options** メニューを使用して、修正可能な CVE に焦点を当てるか、アクティブなイメージにさらに焦点を当てます。



注記

ダッシュボードフィルターでクラスターまたは namespace が選択されている場合、表示されるデータは、アクティブなイメージ、またはフィルタリングされたスコープ内のデプロイメントで使用されるイメージにフィルタリングされています。

1.4.3. 最もリスクのあるデプロイメント

このウィジェットは、環境内で危険にさらされている上位のデプロイメントに関する情報を提供します。リソースの場所 (クラスターと namespace) やリスク優先度スコアなどの追加情報を表示します。さらに、デプロイメントをクリックして、ポリシー違反や脆弱性などのデプロイメントに関するリスク情報を表示することもできます。

1.4.4. イメージの有効期限

古いイメージにはすでに対処されている脆弱性が含まれる可能性があるため、セキュリティーリスクが

高くなります。古いイメージがアクティブであれば、デプロイメントが不正使用される可能性があります。このウィジェットを使用すると、セキュリティー体制を迅速に評価し、問題のあるイメージを特定することができます。デフォルトの範囲を使用するか、独自の値で期間をカスタマイズできます。非アクティブなイメージとアクティブなイメージの両方を表示するか、ダッシュボードフィルターを使用してアクティブなイメージの特定領域に焦点を当てることができます。このウィジェットで有効期限グループをクリックすると、該当するイメージのみを **Vulnerability Management → Images** ページに表示できます。

1.4.5. カテゴリー別のポリシー違反

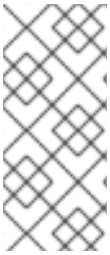
このウィジェットは、どのタイプのポリシーの違反が他よりも多いかを分析することにより、組織が直面しているセキュリティーポリシーの準拠に関する課題の洞察を得るのに役立ちます。ウィジェットには、関心の高い 5 つのポリシーカテゴリーが表示されます。データを切り取るさまざまな方法については、**Options** メニューを確認してください。データをフィルタリングして、デプロイまたはランタイム違反のみにフォーカスできます。

また、並べ替えモードを変更することもできます。デフォルトでは、データは重大度が最も高い違反の数で並べ替えられます。そのため、重要なポリシーを持つすべてのカテゴリーは、重要なポリシーを持たないカテゴリーの前に表示されます。他の並べ替えモードは、重大度に関係なく違反の合計数を考慮します。一部のカテゴリーには重要なポリシーが含まれていないため (“Docker CIS” など)、2 つの並べ替えモードは大幅に異なるビューを提供し、追加の洞察を提供します。

グラフの下部にある重大度レベルをクリックし、そのレベルをデータに含めるか、除外します。異なる重大度レベルを選択すると、上位 5 つの選択またはランキング順序が異なる場合があります。データは、ダッシュボードフィルターで選択されたスコープにフィルタリングされます。

1.4.6. 標準によるコンプライアンス

標準ウィジェットによるコンプライアンス をダッシュボードフィルターと共に使用して、最も重要な領域に焦点を当てることができます。ウィジェットには、並べ替え順序に応じて、上位または下位 6 件のコンプライアンスベンチマークが一覧表示されます。**オプション** を選択して、カバレッジパーセンテージで並べ替えます。ベンチマークラベルまたはグラフのいずれかをクリックして、ダッシュボードスコープと選択したベンチマークでフィルタリングされた **Compliance Controls** ページに直接移動します。



注記

Compliance ウィジェットには、コンプライアンススキャンの実行後にのみ詳細が表示されます。

詳細は、インフラストラクチャーのコンプライアンスステータスの確認 参照してください。

第2章 RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES での COMPLIANCE OPERATOR の使用

Compliance Operator を使用して、OpenShift Container Platform クラスターでコンプライアンスのレポートと修正を行うように RHACS を設定できます。Compliance Operator の結果は、RHACS コンプライアンスダッシュボードに報告されます。

Compliance Operator は、多数の技術実装のレビューを自動化し、それらを業界標準、ベンチマーク、およびベースラインの特定の側面と比較します。

Compliance Operator は監査人ではありません。このようなさまざまな標準に対する準拠または認定を実現するには、Qualified Security Assessor (QSA)、Joint Authorization Board (JAB)、または業界で認められたその他の規制当局など、認定監査機関と協力して、環境を評価する必要があります。

Compliance Operator は、このような標準に関する一般に入手可能な情報とプラクティスに基づいて推奨事項を作成し、場合によっては修復を支援します。ただし、実際の準拠はお客様の責任となります。標準への準拠を実現するには、認定監査人と協力する必要があります。

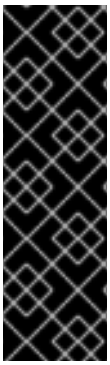
最新の更新は、Compliance Operator リリースノート を参照してください。

2.1. COMPLIANCE OPERATOR のインストール

Operator Hub を使用して Compliance Operator をインストールします。

手順

1. Web コンソールで、**Operators → OperatorHub** ページに移動します。
2. **compliance operator** を **Filter by keyword** ボックスに入力して、Compliance Operator を検索します。
3. **Compliance Operator** を選択して、詳細ページを表示します。
4. Operator に関する情報を読み、**Install** をクリックします。



重要

- コンプライアンス機能を使用する場合は、RHACS を使用してコンプライアンススキャンスケジュールを作成し、スキャンをスケジュールできます。
  コンプライアンス機能を使用してコンプライアンススキャンをスケジュールする方法の詳細は、「コンプライアンススキャンのカスタマイズと自動化」を参照してください。
- スキャンのスケジュールを作成する場合、Compliance Operator に **ScanSettingBinding** を作成する必要は **ありません**。

次のステップ

- ScanSettingBinding オブジェクトの設定

関連情報

- コンプライアンススキャンのカスタマイズと自動化

2.2. SCANSETTINGBINDING オブジェクトの設定

openshift-compliance namespace に **ScanSettingBinding** オブジェクトを作成すると、コマンドラインインターフェイス (CLI) またはユーザーインターフェイス (UI) から **cis** および **cis-node** プロファイルを使用してクラスターをスキャンできます。

重要

この例では **ocp4-cis** および **ocp4-cis-node** プロファイルを使用しますが、OpenShift Container Platform にはその他のプロファイルもあります。

詳細は、「Compliance Operator について」を参照してください。

前提条件

- Compliance Operator がインストールされている。

手順

- CLI から **ScanSettingBinding** オブジェクトを作成するには、次の手順を実行します。
  a. 次の内容を使用して、**sscan.yaml** という名前のファイルを作成します。

```
apiVersion: compliance.openshift.io/v1alpha1
kind: ScanSettingBinding
metadata:
  name: cis-compliance
profiles:
  - name: ocp4-cis-node
    kind: Profile
    apiGroup: compliance.openshift.io/v1alpha1
  - name: ocp4-cis
    kind: Profile
    apiGroup: compliance.openshift.io/v1alpha1
settingsRef:
  name: default
  kind: ScanSetting
  apiGroup: compliance.openshift.io/v1alpha1
```

  b. 次のコマンドを実行して、**ScanSettingBinding** オブジェクトを作成します。

```
$ oc create -f sscan.yaml -n openshift-compliance
```

  成功すると、次のメッセージが表示されます。

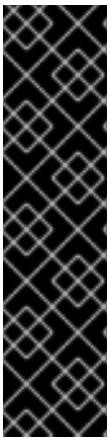
```
$ scansettingbinding.compliance.openshift.io/cis-compliance created
```

- UI から **ScanSettingBinding** オブジェクトを作成するには、次の手順を実行します。
  a. アクティブなプロジェクトを **openshift-compliance** に変更します。
  b. **+** をクリックして、**Import YAML** ページを開きます。
  c. 前の例の YAML を貼り付けて、**Create** をクリックします。

検証

1. RHACS でコンプライアンススキャンを実行します。
   コンプライアンス機能を使用してコンプライアンススキャンを実行する方法の詳細は、「インフラストラクチャーのコンプライアンスステータスの確認」を参照してください。

2. **ocp4-cis** および **ocp4-cis-node** の結果が表示されていることを確認します。



重要

- CLI を使用している場合は、Dashboard ページからコンプライアンススキャンの結果を表示できます。
  Dashboard ページからコンプライアンススキャンの結果を表示する方法の詳細は、「環境全体のコンプライアンス標準の表示」を参照してください。

- UI を使用している場合は、ダッシュボードと Coverage ページの両方からコンプライアンススキャンの結果を表示できます。
  Coverage ページからコンプライアンススキャンの結果を表示する方法の詳細は、「クラスター全体のプロファイルコンプライアンスの評価」を参照してください。

関連情報

- Compliance Operator について
- Compliance Operator のスキャン
- インフラストラクチャーのコンプライアンスステータスの確認
- 環境全体のコンプライアンス標準の表示
- クラスター全体のプロファイルコンプライアンスの評価

第3章 コンプライアンスの管理

3.1. コンプライアンス機能の概要

コンプライアンス機能は、Kubernetes クラスターを業界標準と規制要件に準拠させるためのものです。自動コンプライアンスチェックを実行し、CIS、PCI-DSS、HIPAA などの定義済みベンチマークに照らしてクラスターを継続的に監視できます。

この機能には、管理者がコンプライアンスの問題を迅速に特定して解決するのに役立つ詳細なレポートと修復ガイダンスが含まれています。Red Hat Advanced Cluster Security for Kubernetes (RHACS) ポータルのコンプライアンス機能を使用すると、クラスターに関連するコンプライアンスの結果を表示できます。

コンプライアンス機能は、次のセクションに情報をまとめます。

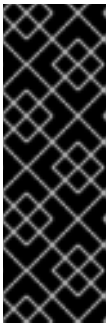
- **Dashboard** (旧称 **Compliance 1.0**) は、すべてのクラスターから収集されたコンプライアンス情報をまとめます。ワークロードとインフラストラクチャーのコンプライアンスを対象としています。

重要

RHACS でコンプライアンススキャンを実行すると、Kubernetes インフラストラクチャーとワークロード全体を監視し、必要な標準を満たしていることを確認できます。コンプライアンスダッシュボードを使用して、フィルタリングや詳細なレポートを作成できます。

詳細は、ワークロードとクラスターのコンプライアンスの監視 を参照してください。

- **Schedules** および **Coverage (テクニカルプレビュー)** (旧称 **Compliance 2.0**) は、Compliance Operator を使用してスケジュールされたスキャンの実行後、コンプライアンス情報を 1 つのインターフェイスにまとめます。



重要

Compliance Operator がインストールされた Red Hat OpenShift クラスターがある場合は、Schedules ページで RHACS に直接コンプライアンススキャンスケジュールを作成および管理できます。Coverage ページには、ベンチマークとプロファイルに関連するスキャン結果が 1 つのインターフェイスに表示されます。

詳細は、コンプライアンススキャンのスケジュールとプロファイルコンプライアンスの評価 (テクノロジープレビュー) を参照してください。

3.1.1. RHACS を使用したコンプライアンス評価とレポート

Dashboard ページで、さまざまなセキュリティーおよび規制フレームワークの該当する技術的なコントロール項目について、コンテナー化されたインフラストラクチャーおよびワークロードのコンプライアンスを評価し、レポートすることができます。

次の業界標準に基づいて、すぐに使用できるコンプライアンススキャンを実行できます。

- **Center for Internet Security (CIS) Benchmarks**for **Kubernetes**
- **Health Insurance Portability and Accountability Act**(HIPAA)

- **National Institute of Standards and Technology (NIST) Special Publication 800-190**
- **NIST Special Publication 800-53**
- **Payment Card Industry Data Security Standard** (PCI DSS)
- **OpenShift Compliance Operator Profiles**: Compliance Operator は、OpenShift Container Platform の Kubernetes API リソースと、クラスターを実行しているノードの両方のコンプライアンスを評価します。Compliance Operator のインストールの一部として利用可能なプロファイルは複数あります。
  利用可能なプロファイルの詳細は、サポートされているコンプライアンスプロファイル を参照してください。

これらの標準に基づいて環境をスキャンすることで、次のことが可能になります。

- 規制コンプライアンスについてインフラストラクチャーを評価できます。
- Kubernetes オーケストレーターを強化できます。
- 環境全体のセキュリティー体制を把握し、管理できます。
- クラスター、namespace、ノードのコンプライアンスステータスの詳細を概観できます。

3.2. ワークロードとクラスターのコンプライアンスの監視

コンプライアンススキャンを実行すると、RHACS でインフラストラクチャー全体のコンプライアンスステータスを確認できます。コンプライアンスダッシュボードで結果を表示し、データをフィルタリングして、クラスター、namespace、ノード全体のコンプライアンスステータスを監視できます。

詳細なコンプライアンスレポートを生成し、特定の標準、コントロール、業界ベンチマークに注目することで、環境のコンプライアンスステータスを追跡して共有し、必要なコンプライアンス標準をインフラストラクチャーが満たしていることを確認できます。

3.2.1. インフラストラクチャーのコンプライアンスステータスの確認

コンプライアンススキャンを実行すると、すべてのコンプライアンス標準について、インフラストラクチャー全体のコンプライアンスステータスを確認できます。コンプライアンススキャンを実行すると、Red Hat Advanced Cluster Security for Kubernetes (RHACS) によって環境のデータスナップショットが作成されます。データスナップショットには、アラート、イメージ、ネットワークポリシー、デプロイメント、および関連するホストベースのデータが含まれます。

Central は、クラスターで実行されている Sensor からホストベースのデータを収集します。その後、Central は各 Collector Pod で実行されているコンプライアンスコンテナーからさらにデータを収集します。

コンプライアンスコンテナーは、環境に関する以下のデータを収集します。

- コンテナーデーモン、コンテナーランタイム、コンテナーイメージの設定。
- コンテナーネットワークに関する情報。
- コンテナーランタイム、Kubernetes、OpenShift Container Platform のコマンドライン引数とプロセス。
- 特定のファイルパスの権限。

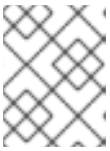
- Kubernetes および OpenShift Container Platform コアサービスの設定ファイル。
- データ収集が完了すると、Central はデータをチェックして結果を判定します。ユーザーはコンプライアンスダッシュボードで結果を表示し、結果に基づいてコンプライアンスレポートを作成できます。

注記

- コンプライアンススキャンに関連する用語は次のとおりです。
  - **コントロール** とは、業界標準または規制規格の 1 項目を表します。これは、情報システムが該当する標準に準拠しているかどうかを評価するために、監査人によって使用されるものです。RHACS は、1 つ以上のチェックを実行して、1 つのコントロールに準拠している証拠を検証します。
  - **チェック** は、1 つのコントロール評価中に実行される 1 回のテストです。
- コントロールによっては、複数のチェックが関連付けられています。コントロールに関連付けられたチェックの 1 つが失敗した場合、コントロールの状態全体が失敗としてマークされます。

手順

1. RHACS ポータルで、**Compliance → Dashboard** をクリックします。
2. オプション: デフォルトでは、すべての標準に関する情報がコンプライアンス結果に表示されます。
   特定の標準に関する情報のみを表示するには、次の手順を実行します。
   a. **Manage standards** をクリックします。
   b. デフォルトでは、すべての標準が選択されます。非表示にする特定の標準のチェックボックスをオフにします。
   c. **Save** をクリックします。
   選択されていない標準は、ウィジェットを含むダッシュボードの表示、ダッシュボードからアクセスできるコンプライアンス結果テーブル、および **Export** ボタンを使用して作成した PDF ファイルには表示されません。ただし、結果を CSV ファイルとしてエクスポートした場合は、すべてのデフォルトの標準が含まれます。
3. **Scan environment** をクリックします。



注記

環境全体のスキャンが完了するまでに約 2 分かかります。この時間は、環境内のクラスターおよびノード数によって異なる可能性があります。

検証

1. RHACS ポータルで、**Configuration Management** をクリックします。
2. **CIS Kubernetes v1.5** ウィジェットで、**Scan** をクリックします。
3. RHACS にコンプライアンススキャンが進行中であることを示すメッセージが表示されます。

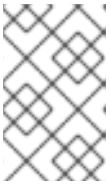
3.2.2. 環境全体のコンプライアンス標準の表示

コンプライアンスダッシュボードには、環境内のすべてのクラスター、namespace、ノードにおけるコンプライアンス標準の概要が表示されます。また、潜在的なコンプライアンスの問題を調査するためのグラフやオプションも表示されます。

コンプライアンススキャン結果は、個々のクラスター、namespace、またはノードごとに表示できます。コンテナー化された環境のコンプライアンスステータスに関するレポートを生成することもできます。

手順

- RHACS ポータルで、**Compliance → Dashboard** をクリックします。



注記

コンプライアンスダッシュボードを初めて開くと、空のダッシュボードが表示されます。コンプライアンススキャンを実行してダッシュボードにデータを入力してください。

3.2.3. コンプライアンスダッシュボードの概要

コンプライアンススキャンを実行すると、コンプライアンスダッシュボードに、環境のコンプライアンスステータスとして結果が表示されます。このダッシュボードからコンプライアンス違反を直接表示できます。環境が特定のベンチマークに準拠しているかどうかを確認するには、詳細ビューをフィルタリングし、コンプライアンス標準をドリルダウンします。

コンプライアンスダッシュボードの右上にあるショートカットを使用して、クラスター、namespace、ノードのコンプライアンスステータスを確認できます。これらのショートカットをクリックすると、コンプライアンススナップショットを表示し、クラスター、namespace、またはノードの全体的なコンプライアンスに関するレポートを生成できます。

3.2.3.1. クラスターのコンプライアンスステータスを表示する

クラスターのコンプライアンスステータスを表示することで、必要なコンプライアンス標準にクラスターが準拠していることを監視し、確認できます。

コンプライアンスダッシュボードで、すべてまたは個々のクラスターのコンプライアンスステータスを表示できます。

手順

- 環境内の全クラスターのコンプライアンスステータスを表示するには、次の手順を実行します。
  - RHACS ポータルで、**Compliance → Dashboard → clusters** タブをクリックします。
- 環境内の特定クラスターのコンプライアンスステータスを表示するには、次の手順を実行します。
  - RHACS ポータルで、**Compliance → Dashboard** をクリックします。
  - **Passing standards by cluster** ウィジェットを探します。
  - このウィジェットで、クラスター名をクリックすると、そのコンプライアンスステータスが表示されます。

3.2.3.2. namespace のコンプライアンスステータスを表示する

namespace のコンプライアンスステータスを表示することで、必要なコンプライアンス標準に各 namespace が準拠していることを監視し、確認できます。

コンプライアンスダッシュボードでは、すべてまたは 1 つの namespace のコンプライアンスステータスを表示できます。

手順

- 環境内の全 namespace のコンプライアンスステータスを表示するには、次の手順を実行します。
  - RHACS ポータルで、**Compliance → Dashboard → namespaces** タブをクリックします。
- 環境内の特定 namespace のコンプライアンスステータスを表示するには、次の手順を実行します。
  - RHACS ポータルで、**Compliance → Dashboard → namespaces** タブをクリックします。
  - **namespace** テーブルで、namespace をクリックします。右側にあるサイドパネルが開きます。
  - サイドパネルで namespace の名前をクリックし、コンプライアンスのステータスを表示します。

3.2.3.3. 特定の標準のコンプライアンスステータスを表示する

特定の標準のコンプライアンスステータスを表示することで、業界および規制のコンプライアンス要件に環境が準拠していることを確認できます。

Red Hat Advanced Cluster Security for Kubernetes (RHACS) は、NIST、PCI DSS、NIST、HIPAA、CIS for Kubernetes、および CIS for Docker コンプライアンス標準をサポートしています。1 つのコンプライアンス標準のコンプライアンスコントロールをすべて表示できます。

手順

1. RHACS ポータルで、**Compliance → Dashboard** をクリックします。
2. **Passing standards across clusters** ウィジェットを探します。
3. 標準をクリックすると、その標準に関連するすべてのコントロールに関する情報が表示されます。

重要

CIS Docker のコントロールの多くは、各 Kubernetes ノードの Docker エンジンの設定を参照しています。多くの CIS Docker コントロールは、コンテナーを構築して使用するためのベストプラクティスでもあり、RHACS にはそれらの使用を強制するポリシーがあります。

詳細は、「セキュリティーポリシーの管理」を参照してください。

関連情報

- セキュリティーポリシーの管理

3.2.3.4. 特定のコントロールのコンプライアンスステータスを表示する

特定のコントロールのコンプライアンスステータスを表示することで、環境が詳細なコンプライアンス要件を満たしていることを確認できます。

選択した標準の特定のコントロールについてコンプライアンスステータスを表示できます。

手順

1. RHACS ポータルで、**Compliance → Dashboard** をクリックします。
2. **Passing standards by cluster** ウィジェットを探します。
3. 標準をクリックすると、その標準に関連するすべてのコントロールに関する情報が表示されます。
4. **Controls** テーブルで、コントロールをクリックします。右側にあるサイドパネルが開きます。
5. サイドパネルでコントロールの名前をクリックし、その詳細を表示します。

3.2.4. コンプライアンスダッシュボードに表示されるデータの量の制限

コンプライアンスデータをフィルタリングすることで、一部のクラスター、業界標準、合格または不合格のコントロールに注目し、コンプライアンスダッシュボードに表示されるデータの量を制限できます。

手順

1. RHACS ポータルで、**Compliance → Dashboard** をクリックします。
2. **clusters**、**namespace**、または **nodes** タブをクリックして、詳細ページを開きます。
3. 検索バーにフィルタリング条件を入力し、**Enter** をクリックします。

3.2.5. 環境のコンプライアンスステータスの追跡

コンプライアンスレポートを生成することで、環境のコンプライアンスステータスを追跡できます。これらのレポートを使用して、さまざまな業界の義務でコンプライアンスステータスを他のステークホルダーに伝えることができます。

次のレポートを生成できます。

- ビジネス面に重点を置き、コンプライアンスステータスのグラフと概要を含む PDF 形式の **エグゼクティブレポート**。
- 技術的な側面に重点を置き、詳細な情報を含む CSV 形式の **エビデンスレポート**。

手順

1. RHACS ポータルで、**Compliance → Dashboard** をクリックします。
2. **Export** タブをクリックし、次のいずれかのタスクを実行します。
   - エグゼクティブレポートを生成するには、**Download Page as PDF** を選択します。
   - エビデンスレポートを生成するには、**Download Evidence as CSV** を選択します。

ヒント

Export オプションは、すべてのコンプライアンスページおよびフィルターされたビューに表示されます。

3.2.5.1. エビデンスレポート

Red Hat Advanced Cluster Security for Kubernetes からの包括的なコンプライアンス関連のデータは、エビデンスレポートとして CSV 形式でエクスポートできます。このエビデンスレポートは、コンプライアンス評価に関する詳細情報が記載されています。このレポートは、コンプライアンス監査人、DevOps エンジニア、セキュリティー担当者などの技術職向けにカスタマイズされています。

エビデンスレポートには、以下の情報が含まれています。

CSV フィールド	説明
Standard	CIS Kubernetes などのコンプライアンス標準。
Cluster	評価したクラスターの名前。
Namespace	デプロイメントが存在する namespace またはプロジェクトの名前。
Object Type	オブジェクトの Kubernetes エンティティータイプ。たとえば、**node**、**cluster**、**DaemonSet**、**Deployment**、**StaticPod** などです。
Object Name	オブジェクトの名前。これは、Kubernetes システムによって生成された、オブジェクトを一意に識別する文字列です。例: **gke-setup-dev21380-default-pool-8e086a77-1jfq**
Control	コンプライアンス標準に表示されるのと同じコントロールの番号。
Control Description	コントロールによって実行されるコンプライアンスチェックに関する説明。
State	コンプライアンスチェックの合否。
Evidence	特定のコンプライアンスチェックが失敗または合格した理由に関する説明。
Assessment Time	コンプライアンススキャンを実行した時刻と日付。

3.2.6. サポート対象のベンチマークバージョン

Red Hat Advanced Cluster Security for Kubernetes (RHACS) は、次の業界標準および規制フレームワークに対するコンプライアンスチェックをサポートしています。

ベンチマーク	サポート対象バージョン
Docker および Kubernetes の CIS Benchmarks (インターネットセキュリティーのセンター)	CIS Kubernetes v1.5.0 および CIS Docker v1.2.0
HIPAA (Health Insurance Portability and Accountability Act)	HIPAA 164
米国立標準技術研究所 (NIST)、	NIST Special Publication 800-190 and 800-53 Rev. 4
PCI DSS (Payment Card Industry Data Security Standard)	PCI DSS 3.2.1

3.3. コンプライアンススキャンのスケジュールとプロファイルコンプライアンスの評価 (テクノロジープレビュー)

重要

コンプライアンススキャンのスケジュールとプロファイルコンプライアンスの評価は、テクノロジープレビュー機能です。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat では、実稼働環境での使用を推奨していません。テクノロジープレビューの機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

Schedules ページで、運用上のニーズに合わせてコンプライアンススキャンスケジュールを作成および管理できます。同じクラスターで同じプロファイルをスキャンするスケジュールを、1 つだけ作成できます。

Coverage ページでスキャン結果を表示およびフィルタリングすることで、すべてのクラスターのコンプライアンスステータスを監視できます。

3.3.1. コンプライアンススキャンのカスタマイズと自動化

コンプライアンススキャンスケジュールを作成すると、運用要件に合わせてコンプライアンススキャンをカスタマイズおよび自動化できます。

注記

同じクラスターで同じプロファイルをスキャンするスケジュールを、1 つだけ作成できます。つまり、1 つのクラスターで同じプロファイルに対して複数のスキャンスケジュールを作成することはできません。

前提条件

- Compliance Operator がインストールされている。

Compliance Operator のインストール方法の詳細は、「Red Hat Advanced Cluster Security for Kubernetes での Compliance Operator の使用」を参照してください。

注記

- 現在、コンプライアンス機能と Compliance Operator は、インフラストラクチャーとプラットフォームのコンプライアンスのみを評価します。
- コンプライアンス機能を使用するには、Compliance Operator が実行されている必要があります。この機能は、Amazon Elastic Kubernetes Service (EKS) では **サポートされていません**。

手順

1. RHACS ポータルで、**Compliance → Schedules** をクリックします。
2. **Create scan schedule** をクリックします。
3. **Create scan schedule** ページで、次の情報を入力します。
   - **Name**: 各コンプライアンススキャンを識別するための名前を入力します。
   - **Description**: 各コンプライアンススキャンの理由を指定します。
   - **Schedule**: 必要なスケジュールに合わせてスキャンスケジュールを調整します。
     - **Frequency**: ドロップダウンリストから、スキャンを実行する頻度を選択します。次の値は、スキャンを実行する頻度に関連するものです。
       - **Daily**
       - **Weekly**
       - **Monthly**
     - **On day(s)**: リストから、スキャンを実行する曜日を 1 つ以上選択します。次の値は、スキャンを実行する曜日に関連するものです。
       - **Monday**
       - **Tuesday**
       - **Wednesday**
       - **Thursday**
       - **Friday**
       - **Saturday**
       - **Sunday**
       - **The first of the month**
       - **The middle of the month**

注記

これらの値は、スキャンの頻度を **Weekly** または **Monthly** に指定した場合にのみ適用されます。

- **Time**: **hh:mm** 形式でスキャンを実行する時刻の入力を開始します。表示されるリストから時刻を選択します。

4. **Next** をクリックします。
5. **Clusters** ページで、スキャンの対象とする 1 つ以上のクラスターを選択します。
6. **Next** をクリックします。
7. **Profiles** ページで、スキャンの対象とする 1 つ以上のプロファイルを選択します。
8. **Next** をクリックします。
9. オプション: 手動でトリガーするレポートのメール配信先を設定するには、次の手順を実行します。

注記

配信先は 1 つ以上追加できます。

a. **Add delivery destination** を展開します。

b. **Delivery destination** ページで、次の情報を入力します。

- **Email notifier**: ドロップダウンリストからメール通知を選択します。
  オプション: 新しいメール通知機能の統合を設定するには、次の手順を実行します。

i. **Select a notifier** ドロップダウンリストから、**Create email notifier** クリックします。

ii. **Create email notifier** ページで、次の情報を入力します。

- **Integration name**: メール通知設定の一意の名前を入力します。この名前は、この特定のメール通知設定を識別および管理するのに役立ちます。
- **Email server**: メールの送信に使用する SMTP サーバーのアドレスを指定します。
- **Username**: SMTP サーバーでの認証に必要なユーザー名を入力します。これは通常、メールの送信に使用されるメールアドレスです。
- **Password**: SMTP ユーザー名に関連付けられているパスワードを入力します。このパスワードは SMTP サーバーによる認証に使用されます。
- **From**: このメールアドレスは通常、メールの送信者を表し、受信者に表示されます。これは任意です。
- **Sender**: 送信者の名前を入力します。これは、**From** のメールアドレスと一緒に表示されます。この名前は、受信者がメールの送信者を識別するのに役立ちます。

- **Default recipient**: 特定の受信者が指定されていない場合に通知を届けるデフォルトのメールアドレスを入力します。これにより、メールが必ず誰かに届くようになります。

- **Annotation key for recipient** アノテーションキーを指定して、特定のデプロイメントまたは namespace に関連するポリシー違反について通知する受信者を定義します。これは任意です。

- オプション: SMTP サーバーが認証を必要としない場合は、**Enable unauthenticated SMTP** チェックボックスをオンにします。セキュリティー上の理由から、これは推奨されません。

- オプション: TLS 証明書の検証を無効にする場合は、**Disable TLS certificate validation (insecure)** チェックボックスをオンにします。セキュリティー上の理由から、これは推奨されません。

- オプション: **Use STARTTLS (requires TLS to be disabled)** フィールドで、ドロップダウンリストから SMTP サーバーへの接続を保護するための STARTTLS の種類を選択します。

重要

このオプションを使用するには、TLS 証明書の検証を無効にする必要があります。

次の値は、SMTP サーバーへの接続を保護するための STARTTLS のタイプに関連するものです。

  - **Disabled**
    データが暗号化されません。

  - **Plain**
    ユーザー名とパスワードを base64 でエンコードします。

  - **Login**
    セキュリティーを強化するために、ユーザー名とパスワードを別々の base64 エンコード文字列として送信します。

iii. **Save integration** をクリックします。

- **Distribution list** レポートを受信する受信者のメールアドレスを 1 つ以上コンマで区切って入力します。

- **Email template**: デフォルトのテンプレートが自動的に適用されます。
  オプション: 必要に応じてメールの件名と本文をカスタマイズするには、次の手順を実行します。

A. 鉛筆アイコンをクリックします。

B. **Edit email template** ページで、次の情報を入力します。

- **Email subject**: 希望するメールの件名を入力します。この件名は受信者の受信トレイに表示されます。メールの目的を明確に示すものにしてください。

- **Email body**: メールのテキストを作成します。これはメールのメインコンテンツです。テキスト、動的コンテンツ用のプレースホルダー、メッセージを効果的に伝えるために必要な書式設定を含めることができます。

C. **Apply** をクリックします。

10. **Next** をクリックします。
11. スキャン設定を確認し、**Save** をクリックします。

検証

1. RHACS ポータルで、**Compliance → Schedules** をクリックします。
2. 作成したコンプライアンススキャンを選択します。
3. **Clusters** セクションで、Operator のステータスが健全であることを確認します。
4. オプション: スキャンスケジュールを編集するには、次の手順を実行します。
   a. **Actions** ドロップダウンリストから、**Edit scan schedule** を選択します。
   b. 変更を加えます。
   c. **Save** をクリックします。
5. オプション: スキャンレポートを手動で送信するには、次の手順を実行します。

注記

メールの配信先を設定している場合にのみ、スキャンレポートを手動で送信できます。

- **Actions** ドロップダウンリストから、**Send report** を選択します。
  レポートの送信を要求したことを確認するメールが届きます。

関連情報

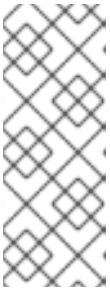
- Red Hat Advanced Cluster Security for Kubernetes での Compliance Operator の使用

3.3.2. クラスター全体のプロファイルコンプライアンスの評価

Coverage ページを表示すると、クラスター全体のノードとプラットフォームリソースのプロファイルコンプライアンスを評価できます。

前提条件

- Compliance Operator がインストールされている。
  Compliance Operator のインストール方法の詳細は、「Red Hat Advanced Cluster Security for Kubernetes での Compliance Operator の使用」を参照してください。



注記

- 現在、コンプライアンス機能と Compliance Operator は、インフラストラクチャーとプラットフォームのコンプライアンスのみを評価します。
- コンプライアンス機能を使用するには、Compliance Operator が実行されている必要があります。この機能は、Amazon Elastic Kubernetes Service (EKS) では **サポートされていません**。

- コンプライアンススキャンスケジュールを作成した。
コンプライアンススキャンスケジュールを作成する方法の詳細は、「コンプライアンススキャンのカスタマイズと自動化」を参照してください。

手順

- RHACS ポータルで、**Compliance → Coverage** をクリックします。

関連情報

- Red Hat Advanced Cluster Security for Kubernetes での Compliance Operator の使用
- コンプライアンススキャンのカスタマイズと自動化

3.3.3. Coverage ページの概要

Coverage ページを表示し、スケジュールにフィルターを適用すると、すべての結果がそれに応じてフィルタリングされます。このフィルターは、削除するまですべての Coverage ページに対して有効になります。1つのプロファイルに基づく結果をいつでも表示できます。

トグルグループを使用して、関連するベンチマークに基づいてグループ化されたプロファイルを選択できます。チェックの合計数と合格したチェックの数をもとに、準拠率を算出してください。

Checks ビューには、プロファイルチェックがリスト表示されます。これを使用して、コンプライアンスステータスを簡単に確認して把握できます。

プロファイルチェックの情報は、次のグループに分かれています。

- **Check**: プロファイルチェックの名前。
- **Controls**: 各チェックに関連するさまざまなコントロールを示します。
- **Fail status**: 失敗し、注意が必要なチェックを示します。
- **Pass status**: 正常に合格したチェックを示します。
- **Manual status**: 自動化できない組織または技術に関する知識がさらに必要なため、手動レビューが必要なチェックを示します。
- **Other status**: 警告や情報ステータスなど、合格または不合格以外のステータスのチェックを示します。
- **Compliance**: 全体的なコンプライアンスステータスを示します。環境が必要な標準を満たしていることを確認するのに役立ちます。

Clusters ビューには、クラスターがリスト表示されます。これを使用して、クラスターを効果的に監視および管理できます。

クラスター情報は次のグループに分かれています。

- **Cluster**: クラスターの名前。
- **Last scanned**: 個々のクラスターが最後にスキャンされた日時を示します。
- **Fail status**: スキャンが失敗し、注意が必要なクラスターを示します。
- **Pass status**: すべてのチェックに合格したクラスターを示します。
- **Manual status**: 自動化できない組織または技術に関する知識がさらに必要なため、手動レビューが必要なチェックを示します。
- **Other status**: 警告や情報アラートなど、合格または不合格以外のステータスのクラスターを表示します。
- **Compliance**: クラスターの全体的なコンプライアンスステータスを示します。クラスターが必要な標準を満たしていることを確認するのに役立ちます。

3.3.4. クラスターの健全性の監視および分析

プロファイルチェックのステータスを表示することで、クラスターの健全性を効率的に監視および分析できます。

重要

Compliance Operator がスキャン結果を返すまで待機してください。これには数分かかる場合があります。

手順

1. RHACS ポータルで、**Compliance → Coverage** をクリックします。
2. クラスターを選択して、個々のスキャンの詳細を表示します。
3. オプション: ステータスを表示するには、**Filter by keyword box** にプロファイルチェックの名前を入力します。
4. オプション: **Compliance status** ドロップダウンリストから、スキャンの詳細をフィルタリングするために使用する 1 つ以上のステータスを選択します。
   次の値は、スキャンの詳細をフィルタリングする方法に関連するものです。
   - **Pass**
   - **Fail**
   - **Error**
   - **Info**
   - **Manual**
   - **該当なし**
   - **Inconsistent**

3.3.5. コンプライアンススキャンステータスの概要

コンプライアンススキャンのステータスを理解することで、環境全体のセキュリティー体制を管理できます。

ステータス	説明
Fail	コンプライアンスチェックに失敗しました。
Pass	コンプライアンスチェックに合格しました。
Not Applicable	該当しないため、コンプライアンスチェックをスキップしました。
Info	コンプライアンスチェックでデータが収集されましたが、RHACS が合否を判断できませんでした。
Error	技術的な問題が原因でコンプライアンスチェックに失敗しました。
Manual	コンプライアンスを確保するには手動による介入が必要です。
Inconsistent	コンプライアンススキャンデータに一貫性がないため、綿密な検査と目標を絞った解決策が必要です。

第4章 セキュリティーリスクの評価

Red Hat Advanced Cluster Security for Kubernetes は環境全体にわたるリスクを評価し、セキュリティーリスクに合わせて実行中のデプロイメントをランク付けします。また、緊急の対応が必要な脆弱性、設定、およびランタイムアクティビティーに関する詳細も提供します。

4.1. リスクビュー

リスク ビューには、すべてのクラスターからのすべてのデプロイメントがリスト表示され、ポリシー違反、イメージコンテンツ、デプロイメント設定、およびその他の同様の要素に基づく多要素リスクメトリックで並べ替えられます。リストの上部にデプロイメントでは、最もリスクが高くなります。

Risk ビューには、各行に以下の属性を持つデプロイメントのリストが表示されます。

- **Name**: デプロイメントの名前。
- **Created**: デプロイメントの作成時間。
- **Cluster**: デプロイメントが実行されているクラスターの名前。
- **namespace**: デプロイメントが存在する namespace。
- **Priority**: 重大度およびリスクメトリックに基づく優先度のランク付け。

Risk ビューでは、以下を実行できます。

- 列見出しを選択して、違反を昇順または降順で並べ替えます。
- フィルターバーを使用して違反をフィルタリングします。
- フィルターされた条件に基づいて新しいポリシーを作成します。

デプロイメントのリスクに関する詳細を表示するには、**Risk** ビューでデプロイメントを選択します。

4.1.1. リスクビューの表示

Risk ビューですべてのリスクを分析し、修正措置を取ることができます。

手順

- RHACS ポータルに移動し、ナビゲーションメニューから **Risk** を選択します。

4.2. リスクビューからのセキュリティーポリシーの作成

リスク ビューで展開のリスクを評価しているときに、ローカルページフィルタリングを適用すると、使用しているフィルタリング条件をもとに新しいセキュリティーポリシーを作成できます。

手順

1. RHACS ポータルに移動し、ナビゲーションメニューから **Risk** を選択します。
2. ポリシーを作成するローカルページのフィルタリング条件を適用します。
3. **New Policy** を選択し、必須フィールドに入力して新規ポリシーを作成します。

4.2.1. Red Hat Advanced Cluster Security for Kubernetes がフィルタリング条件をポリシー条件に変換する方法について

使用するフィルター条件に基づいて、**リスク** ビューから新しいセキュリティーポリシーを作成する場合には、すべての条件が新しいポリシーに直接適用されるわけではありません。

- Red Hat Advanced Cluster Security for Kubernetes は、**Cluster**、**Namespace**、および **Deployment** フィルターを同等のポリシースコープに変換します。
    - **Risk** ビューのローカルページのフィルタリングでは、以下の方法を使用して検索用語を組み合わせます。
        - 同じカテゴリーの検索用語と **OR** 演算子を組み合わせます。たとえば、検索クエリーが **Cluster:A,B** の場合には、フィルターは **cluster A** または **cluster B**. のデプロイメントが返されます。
        - 異なるカテゴリーの検索用語と **AND** 演算子を組み合わせます。たとえば、検索クエリーが **Cluster:A+Namespace:Z** の場合には、フィルターは **クラスター A** および **namespace Z** のデプロイメントがマッチします。
    - 複数のスコープをポリシーに追加すると、このポリシーはすべてのスコープからの違反がマッチします。
        - たとえば、**(Cluster A OR Cluster B)AND(Namespace Z)** を検索すると、2 つのポリシースコープ **(Cluster=A AND Namespace=Z)** または **(Cluster=B AND Namespace=Z)** が結果として返されます。
- Red Hat Advanced Cluster Security for Kubernetes は、ポリシー条件に直接マップされないフィルターをドロップまたは変更し、ドロップされたフィルターを報告します。

次の表では、フィルタリング検索属性をポリシー条件にマップする方法を示します。

検索属性	ポリシー条件
Add Capabilities	Add Capabilities
Annotation	拒否されたアノテーション
CPU Cores Limit	コンテナーの CPU 制限
CPU Cores Request	コンテナーの CPU 要求
CVE	CVE
CVE Published On	✕ 廃止
CVE Snoozed	✕ 廃止
CVSS	CVSS
Cluster	⟳ スコープに変換

検索属性	ポリシー条件
Component	イメージコンポーネント (名前)
Component Version	イメージコンポーネント (バージョン)
Deployment	⟳ スコープに変換
Deployment Type	✕ 廃止
Dockerfile Instruction Keyword	Dockerfile 行 (キー)
Dockerfile Instruction Value	Dockerfile 行 (値)
Drop Capabilities	✕ 廃止
Environment Key	環境変数 (キー)
Environment Value	環境変数 (値)
Environment Variable Source	環境変数 (ソース)
Exposed Node Port	✕ 廃止
Exposing Service	✕ 廃止
Exposing Service Port	✕ 廃止
Exposure Level	ポートの公開
External Hostname	✕ 廃止
External IP	✕ 廃止
Image	✕ 廃止
Image Command	✕ 廃止
Image Created Time	イメージ作成からの日数
Image Entrypoint	✕ 廃止
Image Label	許可されていないイメージラベル
Image OS	イメージ OS
Image Pull Secret	✕ 廃止

検索属性	ポリシー条件
Image Registry	イメージレジストリー
Image Remote	イメージリモート
Image Scan Time	イメージが最後にスキャンされた後の日数
Image Tag	Image Tag
Image Top CVSS	✕ 廃止
Image User	✕ 廃止
Image Volumes	✕ 廃止
Label	⟳ スコープに変換
Max Exposure Level	✕ 廃止
Memory Limit (MB)	コンテナーのメモリー制限
Memory Request (MB)	コンテナーのメモリー要求
Namespace	⟳ スコープに変換
Namespace ID	✕ 廃止
Pod Label	✕ 廃止
Port	ポート
Port Protocol	プロトコル
Priority	✕ 廃止
Privileged	特権
Process Ancestor	プロセスの祖先
Process Arguments	プロセス引数
Process Name	プロセス名
Process Path	✕ 廃止
Process Tag	✕ 廃止

検索属性	ポリシー条件
Process UID	プロセス UID
Read Only Root Filesystem	読み取り専用ルートファイルシステム
Secret	× 廃止
Secret Path	× 廃止
Service Account	× 廃止
Service Account Permission Level	最小 RBAC パーミッションレベル
Toleration Key	× 廃止
Toleration Value	× 廃止
Volume Destination	ボリュームの宛先
Volume Name	ボリューム名
Volume ReadOnly	書き込み可能なボリューム
Volume Source	ボリュームソース
Volume Type	ボリュームタイプ

4.3. リスクの詳細の表示

Risk ビューでデプロイメントを選択すると、右側のパネルに **Risk Details** が表示されます。**Risk Details** パネルには、複数のタブにグループ化された詳細情報が表示されます。

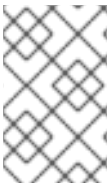
4.3.1. リスクインディケータータブ

Risk Details パネルの **Risk Indicators** タブには、検出されたリスクが説明されています。

Risk Indicators タブには以下のセクションが含まれます。

- **Policy Violations**: 選択したデプロイメントで違反しているポリシーの名前。
- **Suspicious Process Executions**: プロセスが実行されたさまざまなプロセス、引数、およびコンテナー名。
- **Image Vulnerabilities**: CVSS スコアをはじめとした合計 CVE を含むイメージ。
- **Service Configurations**: 読み取り/書き込み (RW) 機能、機能が廃止されているかどうか、特権付きコンテナーがあるかなど、多くの場合に問題が発生する可能性のある各種設定。
- **Service Reachability**: クラスター内外に公開されるコンテナーポート。

- **Components Useful for Attackers** 攻撃者がよく使用すると検出されたソフトウェアツール。
- **Number of Components in Image**: 各イメージにあるパッケージの数。
- **Image Freshness**: イメージ名と使用期間 (例: **285 days old**)
- **RBAC Configuration**: Kubernetes のロールベースアクセス制御 (RBAC) でのデプロイメントに付与されるパーミッションのレベル。



注記

Risk Indicators タブにすべてのセクションが表示されるわけではありません。Red Hat Advanced Cluster Security for Kubernetes は、選択したデプロイメントに影響のある関連セクションのみを表示します。

4.4. デプロイの詳細タブ

Deployment Risk パネルの **Deployment Details** タブのセクションには詳細情報が表示されるため、検出されたリスクに対処する方法について適切な決定を下すことができます。

4.4.1. 概要セクション

Overview セクションには、以下の詳細が表示されます。

- **Deployment ID**: デプロイメントの英数字 ID。
- **namespace**: デプロイメントが存在する Kubernetes または OpenShift Container Platform namespace。
- **updated**: デプロイメントが更新された日付のタイムスタンプ。
- **Deployment Type**: デプロイメントのタイプ (例: **Deployment** または **DaemonSet**)。
- **Replicas**: このデプロイメントにデプロイされた Pod の数。
- **Labels**: Kubernetes または OpenShift Container Platform アプリケーションに割り当てられるキー/値のラベル。
- **Cluster**: デプロイメントが実行されているクラスターの名前。
- **annotations**: デプロイメントの Kubernetes アノテーション。
- **Service Account**: Pod で実行されるプロセスのアイデンティティーを表します。プロセスがサービスアカウントを使用して認証されると、このプロセスは Kubernetes API サーバーに接続し、クラスターリソースにアクセスできます。Pod にサービスアカウントが割り当てられていない場合は、default のサービスアカウントを取得します。

4.4.2. コンテナー設定セクション

コンテナー設定セクションには、以下の詳細が表示されます。

- **Image Name**: デプロイされたイメージの名前。
- **リソース**
  - **CPU Request (cores)**: コンテナーにより要求される CPU の数。

- **CPU Limit (cores)**: コンテナーが使用できる CPU の最大数。
- **Memory Request (MB)**: コンテナーによって要求されるメモリーサイズ。
- **Memory Limit (MB)**: コンテナーが強制終了せずに使用できる最大メモリー量。

- **Mounts**
  - **Name**: マウントの名前。
  - **Source**: マウントのデータを取得するパス。
  - **Destination**: マウントのデータを送信する先のパス。
  - **type**: マウントのタイプ。
- **Secrets**: デプロイメントで使用される Kubernetes シークレットの名前、および X.509 証明書であるシークレット値の基本情報。

4.4.3. セキュリティーコンテキストセクション

Security Context セクションには、以下の詳細が表示されます。

- **Privileged**: コンテナーに特権がある場合に **true** をリスト表示します。

4.5. プロセス検出タブ

Process Discovery タブには、環境内の各コンテナーで実行されたすべてのバイナリーの包括的なリストが、デプロイメントごとに要約されて表示されます。

プロセス検出タブには、以下の詳細が表示されます。

- **Binary Name**: 実行されたバイナリーの名前。
- **Container**: プロセスが実行されるデプロイメントのコンテナー。
- **引数**: バイナリーで渡された特定の引数。
- **Time**: 指定したコンテナーでバイナリーが実行された最新の日時。
- **Pod ID**: コンテナーが存在する Pod の識別子。
- **UID**: プロセスが実行された Linux ユーザー ID。

フィルターバーに **Process Name:<name>** クエリーを使用して、特定のプロセスを検索します。

4.5.1. イベントタイムラインセクション

Process Discovery タブの **Event Timeline** セクションでは、選択したデプロイメントのイベントの概要が表示されます。ポリシー違反、プロセスアクティビティー、およびコンテナーの終了または再起動イベントの数が表示されます。

Event Timeline を選択して、詳細情報を表示できます。

Event Timeline モーダルボックスには、選択したデプロイメントのすべての Pod のイベントが表示されます。

タイムラインのイベントは、以下のように分類されます。

- プロセスアクティビティー
- ポリシー違反
- コンテナーの再起動
- コンテナーの終了

イベントは、タイムラインにアイコンとして表示されます。イベントの詳細を表示するには、マウスポインターをイベントアイコンの上に置きます。詳細はツールチップに表示されます。

- **Show Legend** をクリックして、イベントのタイプに対応するアイコンを確認します。
- **Export → Download PDF** または **Export → Download CSV** を選択して、イベントタイムライン情報をダウンロードします。
- **Show All** ドロップダウンメニューを選択して、タイムラインに表示するイベントタイプを絞り込みます。
- デプロイメントアイコンをクリックして、選択した Pod のコンテナーごとに個別にイベントを表示します。

タイムライン内のすべてのイベントは、下部のミニマップコントロールにも表示されます。ミニマップは、イベントのタイムラインに表示されるイベントの数を制御します。ミニマップで強調表示されている領域を変更して、タイムラインに表示されるイベントを変更できます。これには、ハイライトされた領域を左または右側 (または両方) から減らし、強調表示されている領域をドラッグします。

注記

- コンテナーが再起動すると、Red Hat Advanced Cluster Security for Kubernetes は以下のようになります。
  - Pod 内のコンテナーごとに、最大 10 個の非アクティブなコンテナーインスタンスのコンテナー終了および再起動イベントに関する情報を表示します。たとえば、Pod に 2 つのコンテナー **app** および **sidecar** が含まれると、Red Hat Advanced Cluster Security for Kubernetes は最大 10 の **app** インスタンスのアクティビティーと、最大 10 の **sidecar** インスタンスを保持します。
  - コンテナーの以前のインスタンスに関連付けられているプロセスアクティビティーは追跡しません。
- Red Hat Advanced Cluster Security for Kubernetes は、各 Pod のタプル (プロセス名、プロセス引数、UID) ごとの最新の実行のみを表示します。
- Red Hat Advanced Cluster Security for Kubernetes は、アクティブな Pod のイベントのみを表示します。
- Red Hat Advanced Cluster Security for Kubernetes は、Kubernetes およびコレクターが報告する時間に基づいて、報告されたタイムスタンプを調整します。Kubernetes タイムスタンプは 2 進法の精度を使用し、最も近い秒に時間を丸めます。ただし、コレクターはより正確なタイムスタンプを使用します。たとえば、Kubernetes がコンテナーの起動時間を **10:54:48** として報告し、コレクターは **10:54:47.5349823** で起動したコンテナーのプロセスを報告する場合には、Red Hat Advanced Cluster Security for Kubernetes はコンテナーの起動時間を **10:54:47.5349823** に調整します。

4.6. プロセスベースラインの使用

インフラストラクチャーセキュリティーにプロセスベースを使用して、リスクを最小限に抑えることができます。この方法では、Red Hat Advanced Cluster Security for Kubernetes はまず既存のプロセスを検出し、ベースラインを作成します。その後、デフォルトの deny-all モードで動作し、ベースラインにリスト表示されているプロセスのみを実行できます。

プロセスベースライン
Red Hat Advanced Cluster Security for Kubernetes をインストールすると、デフォルトのプロセスベースラインはありません。Red Hat Advanced Cluster Security for Kubernetes がデプロイメントを検出すると、デプロイメントの全コンテナータイプのプロセスベースラインが作成されます。次に、検出されたすべてのプロセスを独自のプロセスベースラインに追加します。

プロセスベースラインの状態
プロセス検出フェーズでは、すべてのベースラインがロック解除された状態になります。

ロック解除 の状態:

- Red Hat Advanced Cluster Security for Kubernetes が新しいプロセスを検出すると、そのプロセスをプロセスベースラインに追加します。
- プロセスはリスクとして表示されず、違反は発生しません。

Red Hat Advanced Cluster Security for Kubernetes がデプロイメントのコンテナーから最初のプロセスインジケーターを受け取ってから 1 時間後に、プロセス検出フェーズを終了します。この時点で、以下が行われます。

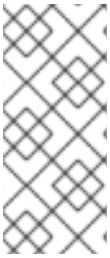
- Red Hat Advanced Cluster Security for Kubernetes は、プロセスのベースラインへのプロセスの追加を停止します。
- プロセスベースラインにない新しいプロセスはリスクとして表示されますが、違反はトリガーしません。

違反を生成するには、プロセスベースラインを手動でロックする必要があります。

ロック 状態:

- Red Hat Advanced Cluster Security for Kubernetes は、プロセスのベースラインへのプロセスの追加を停止します。
- プロセスベースラインにない新しいプロセスは違反をトリガーします。

ベースラインがロックされているかどうかに関係なく、ベースラインからいつでもプロセスを追加または削除できます。



注記

デプロイメントで、各 Pod のコンテナーに複数のコンテナーがある場合には、Red Hat Advanced Cluster Security for Kubernetes は各コンテナータイプごとにプロセスベースラインを作成します。ベースラインがロックされいるものと、ロック解除されているものがあるデプロイメントの場合には、そのデプロイメントのベースラインステータスは **Mixed** と表示されます。

4.6.1. プロセスベースラインの表示

Risk ビューからプロセスベースラインを表示できます。

手順

1. RHACS ポータルで、ナビゲーションメニューから **Risk** を選択します。
2. デフォルトの **Risk** ビューのデプロイメントリストからデプロイメントを選択します。デプロイメントの詳細が、右側のパネルで開きます。
3. **Deployment** details パネルで、 **Process Discovery** タブを選択します。
4. プロセスベースラインは **Spec Container Baselines** セクションに表示されます。

4.6.2. ベースラインへのプロセスの追加

ベースラインにプロセスを追加できます。

手順

1. RHACS ポータルで、ナビゲーションメニューから **Risk** を選択します。
2. デフォルトの **Risk** ビューのデプロイメントリストからデプロイメントを選択します。デプロイメントの詳細が、右側のパネルで開きます。
3. **Deployment** details パネルで、 **Process Discovery** タブを選択します。

4. **Running Processes** セクションで、プロセスベースラインに追加するプロセスの **Add** アイコンをクリックします。

注記

Add アイコンは、プロセスベースラインにないプロセスでのみ利用できます。

4.6.3. ベースラインからのプロセスの削除

ベースラインからプロセスを削除できます。

手順

1. RHACS ポータルで、ナビゲーションメニューから **Risk** を選択します。
2. デフォルトの **Risk** ビューのデプロイメントリストからデプロイメントを選択します。デプロイメントの詳細が、右側のパネルで開きます。
3. **Deployment** details パネルで、 **Process Discovery** タブを選択します。
4. **Spec Container baselines** セクションで、プロセスベースラインから削除するプロセスの **Remove** アイコンをクリックします。

4.6.4. プロセスベースラインのロックとロック解除

ベースラインを **ロック** して、ベースラインに記載されていない全プロセスの違反をトリガーし、ベースラインのロックを **解除** して違反をトリガーしないようにできます。

手順

1. RHACS ポータルで、ナビゲーションメニューから **Risk** を選択します。
2. デフォルトの **Risk** ビューのデプロイメントリストからデプロイメントを選択します。デプロイメントの詳細が、右側のパネルで開きます。
3. **Deployment** details パネルで、 **Process Discovery** タブを選択します。
4. **Spec Container baselines** セクションで、以下を実行します。
   - ベースラインにないプロセスの違反をトリガーするには、**Lock** アイコンをクリックします。
   - **Unlock** アイコンをクリックして、ベースラインにないプロセスの違反のトリガーを停止します。

第5章 アドミッションコントローラーの適用の使用

Red Hat Advanced Cluster Security for Kubernetes は、Kubernetes アドミッションコントローラー および OpenShift Container Platform アドミッションプラグイン と連携します。これにより、管理者は、Kubernetes または OpenShift Container Platform がワークロード (デプロイメント、デーモンセット、ジョブなど) を作成する前に、セキュリティーポリシーを適用できます。

RHACS アドミッションコントローラーは、RHACS で設定したポリシーに違反するワークロードをユーザーが作成することを防ぎます。RHACS バージョン 3.0.41 以降では、ポリシーに違反するワークロードの更新を防ぐようにアドミッションコントローラーを設定することもできます。

RHACS は **ValidatingAdmissionWebhook** コントローラーを使用して、プロビジョニングされるリソースが指定のセキュリティーポリシーに準拠していることを確認します。これに対応するために、RHACS により複数の Webhook ルールが含まれる **ValidatingWebhookConfiguration** が作成されます。

Kubernetes または OpenShift Container Platform API サーバーが Webhook ルールのいずれかに一致する要求を受信する場合には、API サーバーは **AdmissionReview** 要求を RHACS に送信します。RHACS は、設定されたセキュリティーポリシーに基づいて要求を受諾または拒否します。

注記

OpenShift Container Platform でアドミッションコントローラーの適用を使用するには、Red Hat Advanced Cluster Security for Kubernetes バージョン 3.0.49 以降が必要です。

5.1. アドミッションコントローラーの適用について

アドミッションコントローラーの適用を使用する場合は、次の点を考慮してください。

- **API レイテンシー**: アドミッションコントローラーの適用を使用すると、追加の API 検証要求が必要になるため、Kubernetes または OpenShift Container Platform API のレイテンシーが増加します。fabric8 などの数多くの標準 Kubernetes ライブラリーには、デフォルトで短時間の Kubernetes または OpenShift Container Platform API のタイムアウトが含まれています。また、使用しているカスタム自動化での API タイムアウトも検討してください。
- **イメージのスキャン**: クラスター設定パネルで **Contact Image Scanners** オプションを設定して、アドミッションコントローラーが要求の確認中にイメージをスキャンするかどうかを選択できます。
  - この設定を有効にすると、スキャンまたはイメージ署名の検証結果がまだ利用できない場合には、Red Hat Advanced Cluster Security for Kubernetes がイメージスキャナーに接続し、これにが原因でかなりの遅延が発生します。
  - この設定を無効にすると、キャッシュされたスキャンと署名の検証結果が利用可能な場合にのみ、適用するかどうかの意思決定に、イメージスキャンの条件が考慮されます。
- アドミッションコントローラーの適用は、以下に対して使用できます。
  - Pod の **securityContext** のオプション。
  - デプロイメント設定
  - イメージコンポーネントおよび脆弱性。
- アドミッションコントローラーの適用は、以下に対して使用することはできません。
  - プロセスなどのランタイム動作。

- ポートの公開に基づくポリシー。

- Kubernetes または OpenShift Container Platform API サーバーと RHACS Sensor の間に接続の問題がある場合、アドミッションコントローラーが失敗する可能性があります。この問題を解決するには、「アドミッションコントローラーの適用の無効化」セクションの説明に従って、**ValidatingWebhookConfiguration** オブジェクトを削除します。

- ポリシーに対してデプロイ時の適用を有効にして、アドミッションコントローラーを有効にすると、RHACS はポリシーに違反するデプロイメントのブロックを試行します。ポリシーに準拠しないデプロイメントがアドミッションコントローラーによって拒否されない場合 (たとえば、タイムアウトの場合) も、RHACS は、ゼロレプリカへのスケーリングなど、他のデプロイ時の適用メカニズムを適用します。

5.2. アドミッションコントローラーの適用の有効化

Sensor をインストールするとき、または既存のクラスター設定を編集するときに、**Clusters** ビューからアドミッションコントローラーの適用を有効にできます。

手順

1. RHACS ポータルで、**Platform Configuration → Clusters** に移動します。
2. リストから既存のクラスターを選択するか、**Secure a cluster → Legacy installation method** を選択して新しいクラスターを保護します。
3. 新しいクラスターを保護する場合は、クラスター設定パネルの **Static Configuration** セクションで、クラスターの詳細を入力します。
4. アドミッションコントローラーを使用してオブジェクト作成イベントを適用する予定の場合にのみ、作成時に **Configure Admission Controller Webhook to listen on Object Creates**トグルをオンにすることを推奨します。
5. アドミッションコントローラーを使用して更新イベントを適用する予定の場合、**Configure Admission Controller Webhook to listen on Object Updates** トグルをオンにすることを推奨します。
6. アドミッションコントローラーを使用して Pod の実行と Pod のポート転送イベントを適用する予定の場合にのみ、**Enable Admission Controller Webhook to listen on exec and port-forward events** トグルをオンにすることを推奨します。
7. **Dynamic Configuration** セクションで次のオプションを設定します。
   - **Enforce on Object Creates**: このトグルは、アドミッションコントロールサービスの動作を制御します。これを機能させるには、**Configure Admission Controller Webhook to listen on Object Creates** トグルをオンにする必要があります。
   - **Enforce on Object Updates**: このトグルは、アドミッションコントロールサービスの動作を制御します。これを機能させるには、**Configure Admission Controller Webhook to listen on Object Updates** トグルをオンにする必要があります。
8. **Next** を選択します。
9. **Download files** セクションで **Download YAML files and keys**を選択します。

注記

既存のクラスターに対してアドミッションコントローラーを有効にする場合は、次のガイダンスに従ってください。

- **Static Configuration** セクションで変更を加えた場合は、YAML ファイルをダウンロードして Sensor を再デプロイする必要があります。
- **Dynamic Configuration** セクション変更を加えた場合は、RHACS によって Sensor が自動的に同期され、変更が適用されるため、ファイルのダウンロードとデプロイをスキップできます。

10. **Finish** を選択します。

検証

- 生成された YAML を使用して新しいクラスターをプロビジョニングした後、次のコマンドを実行して、アドミッションコントローラーの適用が正しく設定されているかどうかを確認します。

```
$ oc get ValidatingWebhookConfiguration 1
```

1 Kubernetes を使用する場合は、**oc** の代わりに **kubectl** を入力します。

出力例

```
NAME       CREATED AT
stackrox   2019-09-24T06:07:34Z
```

5.3. アドミッションコントローラーの適用の回避

アドミッションコントローラーを回避するには、**admission.stackrox.io/break-glass** アノテーションを YAML 設定に追加します。アドミッションコントローラーを回避すると、デプロイメントの詳細を含むポリシー違反がトリガーされます。Red Hat は、アドミッションコントローラーを回避した理由を他のユーザーが理解できるように、問題トラッカーのリンクまたはその他の参照をこのアノテーションの値に指定することを推奨します。

5.4. アドミッションコントローラーの適用の無効化

Red Hat Advanced Cluster Security for Kubernetes (RHACS) ポータルの **Clusters** ビューからアドミッションコントローラーの適用を無効にできます。

手順

1. RHACS ポータルで、**Platform Configuration → Clusters** を選択します。
2. リストから既存のクラスターを選択します。
3. **Dynamic Configuration** セクションで、**Enforce on Object Creates** と **Enforce on Object Updates** のトグルをオフにします。
4. **Next** を選択します。

5. **Finish** を選択します。

5.4.1. 関連するポリシーの無効化

関連するポリシーの適用をオフにすると、アドミッションコントローラーに適用をスキップするように指示できます。

手順

1. RHACS ポータルで、**Platform Configuration** → **Policy Management** に移動します。
2. デフォルトのポリシーで適用を無効にします。
   - ポリシービューで、**Kubernetes Actions: Exec into Pod** ポリシーを見つけます。オーバーフローメニュー をクリックし、**Disable policy** を選択します。
   - ポリシービューで、**Kubernetes Actions: Port Forward to Pod** ポリシーを見つけます。オーバーフローメニュー をクリックし、**Disable policy** を選択します。
3. デフォルトの **Kubernetes Actions: Port Forward to Pod** および **Kubernetes Actions: Exec into Pod** の実行ポリシーの条件を使用して作成した他のカスタムポリシーの適用を無効にします。

5.4.2. Webhook の無効化

RHACS ポータルの **Clusters** ビューからアドミッションコントローラーの適用を無効にできます。

重要

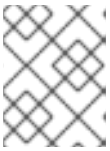
Webhook をオフにしてアドミッションコントローラーを無効にする場合は、Sensor バンドルを再デプロイする必要があります。

手順

1. RHACS ポータルで、**Platform Configuration** → **Clusters** に移動します。
2. リストから既存のクラスターを選択します。
3. **Static Configuration** セクションで、**Enable Admission Controller Webhook to listen on exec and port-forward events** トグルをオフにします。
4. **Next** を選択して、Sensor の設定を続行します。
5. **Download YAML file and keys**クリックします。
6. 監視対象クラスターにアクセスできるシステムから、**Sensor** スクリプトを抽出して実行します。

```
$ unzip -d sensor sensor-<cluster_name>.zip
```

```
$ ./sensor/sensor.sh
```



注記

センサーをデプロイするために必要な権限がないという警告が表示された場合は、画面の指示に従うか、クラスター管理者に連絡して支援を求めてください。

Sensor はデプロイされた後、Central に接続し、クラスター情報を提供します。

7. RHACS ポータルに戻り、デプロイメントが成功したかどうかを確認します。成功すると、セクション #2 の下に緑色のチェックマークが表示されます。緑色のチェックマークが表示されない場合は、次のコマンドを使用して問題を確認してください。
   - OpenShift Container Platform

     ```
     $ oc get pod -n stackrox -w
     ```

   - Kubernetes の場合:

     ```
     $ kubectl get pod -n stackrox -w
     ```

8. **Finish** を選択します。

注記

アドミッションコントローラーを無効にしても、RHACS は **ValidatingWebhookConfiguration** パラメーターを削除しません。要求について違反がチェックされずに、すべての **AdmissionReview** 要求が受け入れられます。

ValidatingWebhookConfiguration オブジェクトを削除するには、セキュアクラスターで次のコマンドを実行します。

- OpenShift Container Platform

  ```
  $ oc delete ValidatingWebhookConfiguration/stackrox
  ```

- Kubernetes の場合:

  ```
  $ kubectl delete ValidatingWebhookConfiguration/stackrox
  ```

5.5. VALIDATINGWEBHOOKCONFIGURATION YAML ファイルの変更

Red Hat Advanced Cluster Security for Kubernetes を使用すると、以下でセキュリティーポリシーを有効にできます。

- オブジェクトの作成
- オブジェクトの更新
- Pod の実行
- Pod ポート転送

Central または Sensor が利用できない場合
アドミッションコントローラーを機能させるには、Sensor からの初期設定が必要です。この設定は、Kubernetes または OpenShift Container Platform によって保存されるため、すべてのアドミッション

コントロールサービスのレプリカが他のノードに再スケジュールされた場合でもアクセスできます。この初期設定が存在する場合、アドミッションコントローラーは設定済みのすべてのデプロイ時ポリシーを適用します。

Sensor または Central が後に利用できなくなる場合:

- イメージスキャンを実行したり、キャッシュされたイメージスキャンに関する情報をクエリーしたりすることはできません。ただし、アドミッションコントローラーの適用は、収集された情報が不完全であっても、タイムアウト経過前に収集された利用可能な情報に基づいて引き続き機能します。
- RHACS ポータルからアドミッションコントローラーを無効にしたり、既存のポリシーの適用を変更したりすることはできません。変更がアドミッションコントロールサービスに伝播されないためです。

注記

受付コントロールの適用を無効にする必要がある場合は、以下のコマンドを実行して検証用の Webhook 設定を削除できます。

- OpenShift Container Platform

```
$ oc delete ValidatingWebhookConfiguration/stackrox
```

- Kubernetes の場合:

```
$ kubectl delete ValidatingWebhookConfiguration/stackrox
```

アドミッションコントローラーの信頼性の強化

Red Hat は、ワーカーノードではなく、コントロールプレーンで受付コントロールサービスをスケジュールすることを推奨します。デプロイメント YAML ファイルには、コントロールプレーンで実行するためのソフト設定が含まれていますが、これは適用されていません。

デフォルトでは、アドミッションコントロールサービスは 3 つのレプリカを実行します。信頼性を向上させるには、以下のコマンドを実行してレプリカを増やします。

```
$ oc -n stackrox scale deploy/admission-control --replicas=<number_of_replicas>
```
1

1 Kubernetes を使用する場合は、**oc** の代わりに **kubectl** を入力します。

roxctl CLI での使用

Sensor のデプロイメント YAML ファイルを生成する場合に、以下のオプションを使用できます。

- **--admission-controller-listen-on-updates**: このオプションを使用すると、Red Hat Advanced Cluster Security for Kubernetes は、Kubernetes または OpenShift Container Platform API サーバーから更新イベントを受信するように事前に設定された **ValidatingWebhookConfiguration** を使用して Sensor バンドルを生成します。
- **--admission-controller-enforce-on-updates**: このオプションを使用すると、Red Hat Advanced Cluster Security for Kubernetes は、アドミッションコントローラーがセキュリティーポリシーオブジェクトの更新も適用するように Central を設定します。

これらのオプションは両方とも任意で、デフォルトは **false** です。

第6章 セキュリティーポリシーの管理

Red Hat Advanced Cluster Security for Kubernetes では、追加設定なしのセキュリティーポリシーを使用して、コンテナー環境用にカスタムのマルチファクターポリシーを定義できます。これらのポリシーを設定すると、環境での高リスクサービスのデプロイメントを自動的に防ぎ、ランタイムのセキュリティーインシデントに対応できます。

6.1. デフォルトのセキュリティーポリシーの使用

Red Hat Advanced Cluster Security for Kubernetes には、セキュリティーの問題を特定して、お使いの環境でセキュリティーのベストプラクティスを実行できるように、幅広く対応する、デフォルトポリシーのセットが含まれています。

デフォルトのポリシーを表示するには、以下を実行します。

- RHACS ポータルで、**Platform Configuration → Policy Management** に移動します。

Policies ビューで、ポリシーを設定することもできます。

ポリシー情報は次のグループに分かれています。

- **Policy**: ポリシーの名前。
- **Description**: ポリシーのアラートの詳細な説明。
- **Status**: ポリシーの現在のステータス (**Enabled** または **Disabled** のいずれか)。
- **Notifiers**: ポリシーに設定された通知機能のリスト
- **Severity**: 必要な注意の程度について、クリティカル、高、中、低のいずれかのポリシーのランク付け。
- **Lifecycle**: このポリシーが適用されるコンテナーライフサイクル (ビルド、デプロイ、またはランタイム) のフェーズと、ポリシーが有効な場合に適用されるフェーズ。

Policy categories ビューには、カテゴリーがリスト表示されます。これを使用して、ポリシーのカテゴリーを管理できます。デフォルトでは、すべてのカテゴリーがリスト表示されます。必要に応じて、カテゴリー名を使用してカテゴリーをフィルタリングできます。

以下のカテゴリーがリスト表示されます。

- Anomalous Activity
- Cryptocurrency Mining
- DevOps Best Practices
- Docker CIS
- Kubernetes
- Kubernetes Events
- Network Tools
- Package Management

- Privileges
- Security Best Practices
- Supply Chain Security
- System Modification
- Vulnerability Management
- Zero Trust

注記

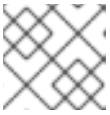
デフォルトのポリシーを削除したり、デフォルトポリシーのポリシー条件を編集したりすることはできません。

6.2. 既存のセキュリティーポリシーの変更

作成したポリシーと、Red Hat Advanced Cluster Security for Kubernetes が提供する既存のデフォルトポリシーを編集できます。

手順

1. RHACS ポータルで、**Platform Configuration → Policy Management** に移動します。
2. **Policies** ページから、編集するポリシーを選択します。
3. **Actions → Edit policy** を選択します。
4. **Policy details** を変更します。ポリシー名、重大度、カテゴリー、説明、理論的根拠、およびガイダンスを変更できます。**Attach notifiers** セクションの下にある利用可能な **Notifier** から選択して、通知機能をポリシーに割り当てることもできます。
5. **Next** をクリックします。
6. **Policy behavior** セクションで、ポリシーの **Lifecycle stages** および **Event sources** を選択します。
7. ポリシーの違反に対応する **Response method** を選択します。
8. **Next** をクリックします。
9. **Policy criteria** セクションで、**Drag out policy fields**セクションのカテゴリーをデプロイメントします。ドラッグアンドドロップポリシーフィールドを使用して、ポリシー条件の論理条件を指定します。

   注記

   デフォルトポリシーのポリシー条件は編集できません。

10. **Next** をクリックします。
11. **Policy scope** セクションで、**Restrict by scope**、**Exclude by scope**、および **Exclude images** 設定を変更します。

12. **Next** をクリックします。
13. **Review policy** セクションで、ポリシー違反をプレビューします。
14. **Save** をクリックします。

関連情報

- システムポリシービューからのセキュリティーポリシーの作成

6.3. ポリシーカテゴリーの作成と管理

6.3.1. Policy categories タブを使用したポリシーカテゴリーの作成

バージョン 3.74 以降の RHACS では、PostgreSQL データベースが有効になっている場合に、Red Hat Advanced Cluster Security Cloud Service または RHACS でポリシーカテゴリーを新しい方法で作成および管理できます。この機能を使用する場合、ポリシー作成以外のすべてのポリシーワークフローは変更されません。

PolicyCategoryService API オブジェクトを使用して、ポリシーカテゴリーを設定することもできます。詳細は、RHACS ポータルの **Help → API reference** に移動してください。

手順

1. RHACS ポータルで、**Platform Configuration → Policy Management** に移動します。
2. **Policy categories** タブをクリックします。このタブには、既存のカテゴリーのリストが表示され、カテゴリー名でリストをフィルタリングできます。**Show all categories** をクリックし、チェックボックスを選択して、表示されたリストからデフォルトまたはカスタムカテゴリーを削除することもできます。
3. **Create category** をクリックします。
4. カテゴリー名を入力し、**Create** をクリックします。

6.3.2. Policy categories タブを使用したポリシーカテゴリーの変更

バージョン 3.74 以降の RHACS では、PostgreSQL データベースが有効になっている場合に、Red Hat Advanced Cluster Security Cloud Service または RHACS でポリシーカテゴリーを新しい方法で作成および管理できます。この機能を使用する場合、ポリシー作成以外のすべてのポリシーワークフローは変更されません。

PolicyCategoryService API オブジェクトを使用して、ポリシーカテゴリーを設定することもできます。詳細は、RHACS ポータルの **Help → API reference** に移動してください。

手順

1. RHACS ポータルで、**Platform Configuration → Policy Management** に移動します。
2. **Policy categories** タブをクリックします。このタブには、既存のカテゴリーのリストが表示され、カテゴリー名でリストをフィルタリングできます。**Show all categories** をクリックし、チェックボックスを選択して、表示されたリストからデフォルトまたはカスタムカテゴリーを削除することもできます。

3. ポリシー名をクリックして、編集または削除します。デフォルトのポリシーカテゴリーは、選択、編集、または削除できません。

関連情報

- システムポリシービューからのセキュリティーポリシーの作成

6.4. カスタムポリシーの作成

デフォルトのポリシーを使用することに加えて、Red Hat Advanced Cluster Security for Kubernetes でカスタムポリシーを作成することもできます。

新しいポリシーを構築するには、既存のポリシーのクローンを作成するか、ゼロから新規ポリシーを作成します。

- RHACS ポータルの **Risk** ビューのフィルター条件をもとにポリシーを作成することもできます。
- また、ポリシー条件に論理演算子ではなく **AND**、**OR** および **NOT** を使用して高度なポリシーを作成することもできます。

6.4.1. システムポリシービューからのセキュリティーポリシーの作成

システムポリシービューから新しいセキュリティーポリシーを作成できます。

手順

1. RHACS ポータルで、**Platform Configuration** → **Policy Management** に移動します。
2. **Create policy** をクリックします。
3. **Policy details** セクションに、ポリシーに関する以下の情報を入力します。
   - ポリシーの **Name** を入力します。
   - オプション: **Attach notifiers** セクションの下にある利用可能な **Notifier** から選択して、通知機能をポリシーに割り当てることもできます。

注記

アラートを転送する前に、RHACS を Webhook、Jira、PagerDuty、Splunk などの通知プロバイダーと統合する必要があります。

   - このポリシーの **重大度** レベルを選択します (**Critical**、**High**、**Medium**、または **Low** のいずれか)。
   - このポリシーに適用するポリシーの **Categories** を選択します。カテゴリーの作成に関する詳細は、このドキュメントの後半で「ポリシーカテゴリーの作成および管理」を参照してください。
   - **Description** フィールドに、ポリシーの詳細を入力します。
   - **Rationale** フィールドにポリシーが存在する理由の説明を入力します。
   - **Guidance** フィールドでこのポリシーの違反を解決するための手順を入力します。

- オプション: **MITRE ATT&CK** セクションで、ポリシーに指定する tactics and the techniques を選択します。

  a. **Add tactic** をクリックし、ドロップダウンリストから調整を選択します。

  b. **Add technique** をクリックして、選択した戦略の手法を追加します。戦略には、複数の手法を指定できます。

4. **Next** をクリックします。

5. **Policy behavior** セクションで、次の手順を実行します。

   a. ポリシーが適用される **Lifecycle stages** (**Build**、**Deploy**、または **Runtime**) を選択します。複数のステージを選択できます。

      - ビルド時ポリシーは、CVE や Dockerfile 手順などのイメージフィールドに適用されます。

      - デプロイ時のポリシーにはすべてのビルドタイムポリシー条件を含めることができますが、特権モードで実行したり、Docker ソケットをマウントするなど、クラスター設定からのデータを含めることもできます。

      - ランタイムポリシーには、すべてのビルド時およびデプロイ時のポリシー条件を含めることができますが、ランタイム時のプロセス実行に関するデータを含めることもできます。

   b. オプション: **Runtime** lifecycle stage を選択した場合は、以下の **Event sources** のいずれかを選択します。

      - Deployment: イベントソースにプロセスとネットワークアクティビティー、Pod 実行、および Pod ポート転送が含まれている場合に、RHACS はポリシー違反をトリガーします。

      - イベントソースが Kubernetes 監査ログレコードと一致すると、RHACS はポリシー違反をトリガーします。

6. **Response method** には、次のいずれかのオプションを選択します。

   a. **Inform**: 違反の一覧に違反を追加する。

   b. **inform および enforce**: アクションを実施します。

7. オプション: **Inform and enforce** を選択した場合は、**Configure enforcement behavior** で、各ライフサイクルのトグルを使用してポリシーの適用動作を選択します。**Lifecycle stages** の設定時に選択したステージでのみ使用できます。適用の振る舞いは、ライフサイクルの各ステージで異なります。

   - **Build**: イメージがポリシーの基準と一致すると、RHACS は継続的インテグレーション (CI) ビルドに失敗します。

   - **Deploy**: **Deploy** 段階では、RHACS アドミッションコントローラーが設定され実行されている場合、RHACS はポリシーの条件に一致するデプロイメントの作成と更新をブロックします。

     - アドミッションコントローラーが適用されているクラスターでは、Kubernetes または OpenShift Container Platform サーバーがすべての非準拠のデプロイメントをブロックします。他のクラスターでは、RHACS は準拠していないデプロイメントを編集して、Pod がスケジュールされないようにします。

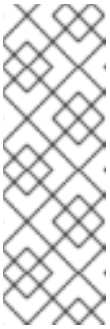
    - 既存のデプロイメントの場合、ポリシーの変更は、Kubernetes イベントが発生したときに、基準が次に検出されたときにのみ適用されます。適用の詳細は、「デプロイステージのセキュリティーポリシーの適用」を参照してください。

- **Runtime** - Pod のイベントがポリシーの基準と一致すると、RHACS はすべての Pod を削除します。

警告

ポリシーの適用は、実行中のアプリケーションまたは開発プロセスに影響を与える可能性があります。適用オプションを有効にする前に、すべての利害関係者に通知し、自動適用アクションに対応する方法を計画してください。

8. **Next** をクリックします。
9. **Policy Criteria** セクションで、ポリシーをトリガーする属性を設定します。
    a. **Policy Section** にポリシーフィールドをクリックしてドラッグし、基準を追加します。



注記

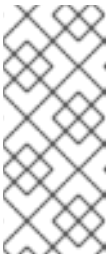
利用可能なポリシーフィールドは、ポリシーに選択したライフサイクルステージによって異なります。たとえば、**Kubernetes access policies** または **Networking** 下の基準は、ランタイムライフサイクルのポリシーを作成するときに利用できますが、ビルドライフサイクルのポリシーを作成する場合は利用できません。ポリシー条件の詳細 (条件に関する情報や、条件が使用可能なライフサイクルフェーズなど) については、「関連情報」セクションの「ポリシー条件」を参照してください。

    b. オプション: **Add condition** をクリックして、ポリシーをトリガーする追加の基準を含むポリシーセクションを追加します (たとえば、古いイメージに対してトリガーするには、**image tag** が **latest はないこと** や **image age** を設定し、イメージがビルドされてからの最小日数を指定できます)。
10. **Next** をクリックします。
11. **Policy scope** セクションで、以下を設定します。

- **Add inclusion scope** をクリックして、**Restrict by scope** を使用し、特定のクラスター、namespace、またはラベルだけに、このポリシーを有効にします。複数のスコープを追加したり、namespaces とラベルの RE2 Syntax で正規表現を使用したりすることもできます。

- **Add exclusion scope** をクリックして **Exclude by scope** を使用し、指定するデプロイメント、クラスター、namespace、およびラベルを除外します。ポリシーは、選択したエンティティーには適用されません。複数のスコープを追加したり、namespaces とラベルの RE2 Syntax で正規表現を使用したりすることもできます。ただし、デプロイメントの選択に正規表現を使用することはできません。

- **Excluded Images (Build Lifecycle only)** の場合は、違反をトリガーしないすべてのイメージを選択します。



注記

Excluded Images 設定は、**Build** ライフサイクルステージで継続的インテグレーションシステムでイメージを確認する場合にのみ適用されます。このポリシーを使用して、実行中のデプロイメント (**Deploy** ライフサイクルステージ) またはランタイムアクティビティー (**Runtime** ライフサイクルステージ) をチェックする場合、効果はありません。

12. **Next** をクリックします。
13. **Review policy** セクションで、ポリシー違反をプレビューします。
14. **Save** をクリックします。

6.4.1.1. デプロイステージのセキュリティーポリシーの実施

Red Hat Advanced Cluster Security for Kubernetes は、デプロイ時のポリシーに対して、アドミッションコントローラーによるハードな適用と RHACS Sensor によるソフトな適用という 2 つの形式のセキュリティーポリシー適用をサポートしています。アドミッションコントローラーは、ポリシーに違反するデプロイメントの作成または更新をブロックします。アドミッションコントローラーが無効または使用できない場合、Sensor はポリシーに違反するデプロイメントのレプリカを **0** にスケールダウンして強制を実行できます。

警告

ポリシーの適用は、実行中のアプリケーションまたは開発プロセスに影響を与える可能性があります。適用オプションを有効にする前に、すべての利害関係者に通知し、自動適用アクションに対応する方法を計画してください。

6.4.1.1.1. ハードエンフォースメント

ハードエンフォースメントは、RHACS アドミッションコントローラーによって実行されます。アドミッションコントローラーが適用されているクラスターでは、Kubernetes または OpenShift Container Platform サーバーがすべての非準拠のデプロイメントをブロックします。アドミッションコントローラーは、**CREATE** および **UPDATE** 操作をブロックします。デプロイ時の強制が有効に設定されたポリシーを満たす Pod の作成または更新リクエストはすべて失敗します。

注記

Kubernetes アドミッション Webhook は、**CREATE**、**UPDATE**、**DELETE**、または **CONNECT** 操作のみをサポートします。RHACS アドミッションコントローラーは、**CREATE** および **UPDATE** 操作のみをサポートします。**kubectl patch**、**kubectl set**、**kubectl scale** などの操作は、UPDATE 操作ではなく、PATCH 操作です。Kubernetes では PATCH 操作がサポートされていないため、RHACS は PATCH 操作の強制を実行できません。

ブロックを強制するには、RHACS でクラスターに対して次の設定を有効にする必要があります。

- **Enforce on Object Creates**: Dynamic Configuration セクションのこのトグルは、アドミッションコントロールサービスの動作を制御します。これを機能させるには、Static Configuration セクションの **Configure Admission Controller Webhook to listen on Object Creates** トグルをオンにする必要があります。
- **オブジェクトの更新時に強制**: Dynamic Configuration セクションのこのトグルは、アドミッションコントロールサービスの動作を制御します。これを機能させるには、Static Configuration セクションの **Configure Admission Controller Webhook to listen on Object Updates** トグルをオンにする必要があります。

Static Configuration 設定の項目を変更した場合に、その変更を有効にするにはセキュアクラスターを再デプロイする必要があります。

6.4.1.1.2. ソフトな適用

ソフトな適用は RHACS Sensor によって実行されます。このエンフォースメントにより、操作が開始しなくなります。ソフトな適用では、Sensor はレプリカを 0 にスケーリングし、Pod がスケジュールされるのを防ぎます。このエンフォースメントでは、クラスター内で準備ができていないデプロイメントが使用可能になります。

ソフトな適用が設定されていて、Sensor がダウンしている場合、RHACS は適用を実行できません。

6.4.1.1.3. namespace の除外

デフォルトでは、RHACS は、**stackrox**、**kube-system**、**istio-system** namespace などの特定の管理 namespace をエンフォースメントブロックから除外します。その理由は、RHACS が正しく機能するためには、これらの namespace 内の一部の項目をデプロイする必要があるためです。

6.4.1.1.4. 既存のデプロイメントへのエンフォースメント

既存のデプロイメントの場合、ポリシーの変更は、Kubernetes イベントが発生したときに、基準が次に検出されたときにのみ適用されます。ポリシーに変更を加えた場合は、**Policy Management** を選択し、**Reassess All** をクリックしてポリシーを再評価する必要があります。このアクションは、新しい受信 Kubernetes イベントがあるかどうかに関係なく、すべての既存のデプロイメントにデプロイポリシーを適用します。ポリシーに違反があった場合は、RHACS がエンフォースメントを実行します。

関連情報

- ポリシー条件
- アドミッションコントローラーの適用の使用

6.4.2. リスクビューからのセキュリティーポリシーの作成

リスク ビューで展開のリスクを評価しているときに、ローカルページフィルタリングを適用すると、使用しているフィルタリング条件をもとに新しいセキュリティーポリシーを作成できます。

手順

1. RHACS ポータルに移動し、ナビゲーションメニューから **Risk** を選択します。
2. ポリシーを作成するローカルページのフィルタリング条件を適用します。
3. **New Policy** を選択し、必須フィールドに入力して新規ポリシーを作成します。

関連情報

- ローカルページのフィルタリングの使用
- システムポリシービューからのセキュリティーポリシーの作成

6.4.3. ポリシー条件

Policy Criteria セクションで、ポリシーをトリガーするデータを設定できます。

以下の表に記載されている属性に基づいてポリシーを設定できます。

この表では、以下のようになります。

- **正規表現**、**AND**、**OR**、および **NOT** 列は、特定の属性とともに正規表現およびその他の論理演算子を使用できるかどうかを示します。
  - Regex の **!** (正規表現) は、リストされたフィールドに正規表現のみを使用できることを示します。
  - **AND** の **!**、または **OR** は、属性に前述の論理演算子のみを使用できることを示します。
  - **正規表現** の **×** / **NOT** / **AND**、**OR** 列は、属性がこれらのいずれもサポートしていないことを示します (正規表現、否定、論理演算子)。
- **RHACS バージョン** 列は、属性を使用する必要がある Red Hat Advanced Cluster Security for Kubernetes のバージョンを示します。
- 論理組み合わせ演算子の **AND** および **OR** は、以下の属性には使用できません。
  - ブール値: **true** および **false**
  - 最小値セマンティクス。たとえば、以下のようになります。
    - **最小 RBAC パーミッション**
    - **イメージ作成からの日数**
- **NOT** 論理演算子は、以下の属性に使用できません。
  - ブール値: **true** および **false**
  - **<**、**>**、**<=**、**>=** 演算子など、比較をすでに使用している数値。
  - 複数の値を指定できる複合条件。たとえば、以下のようになります。
    - **Dockerfile 行**。命令と引数の両方が含まれます。
    - **環境変数**。名前と値の両方で構成されます。
  - Add Capabilities、Drop Capabilities、Days since image was created Days since image was last scanned などの他の意味。

属性	説明	JSON 属性	許可される値	Regex、NOT、AND、OR	フェーズ
セクション: イメージレジストリー					

属性	説明	JSON 属性	許可される値	Regex、NOT、AND、OR	フェーズ
Image Registry	イメージレジストリーの名前。	Image Registry	String	正規表現、NOT、AND、OR	**Build**、**Deploy** **Runtime** (Runtime 条件で使用する場合)
Image Name	**library/nginx** など、レジストリー内のイメージのフルネーム。	Image Remote	String	正規表現、NOT、AND、OR	**Build**、**Deploy** **Runtime** (Runtime 条件で使用する場合)
Image Tag	イメージの識別子。	Image Tag	String	正規表現、NOT、AND、OR	**Build**、**Deploy** **Runtime** (Runtime 条件で使用する場合)
Image Signature	イメージの署名を検証するために使用できる署名統合のリスト。署名がないか、提供された署名統合の少なくとも 1 つによって署名が検証できないイメージに関するアラートを作成します。	Image Signature Verified By	すでに設定されているイメージ署名統合の有効な ID	! **OR** のみ	**Build**、**Deploy** **Runtime** (Runtime 条件で使用する場合)
セクション: イメージの内容					
The Common Vulnerabilities and Exposures (CVE) is fixable	この基準は、評価しているデプロイメント内のイメージに修正可能な CVE がある場合にのみ違反となります。	Fixable	ブール値	×	**Build**、**Deploy** **Runtime** (Runtime 条件で使用する場合)

属性	説明	JSON 属性	許可される値	Regex、NOT、AND、OR	フェーズ
Days Since CVE Was First Discovered In Image	この基準は、RHACS が特定のイメージ内で CVE を検出してから、指定された日数を超えた場合にのみ違反となります。	Days Since CVE Was First Discovered In Image	整数	×	**Build**、**Deploy** **Runtime** (Runtime 条件で使用する場合)
Days Since CVE Was First Discovered In System	この基準は、RHACS が監視するすべてのクラスター内にデプロイされた全イメージから CVE を検出してから、指定された日数を超えた場合にのみ違反となります。	Days Since CVE Was First Discovered In System	整数	×	**Build**、**Deploy** **Runtime** (Runtime 条件で使用する場合)
イメージの経過時間	イメージの作成日からの最小日数。	イメージの経過時間	整数	×	**Build**、**Deploy** **Runtime** (Runtime 条件で使用する場合)
イメージのスキャン期間	イメージが最後にスキャンされた後の最小日数。	イメージのスキャン期間	整数	×	**Build**、**Deploy** **Runtime** (Runtime 条件で使用する場合)
Image User	Dockerfile の USER ディレクティブと一致します。詳細は、https://docs.docker.com/engine/reference/builder/#user を参照してください。	Image User	String	正規表現、NOT、AND、OR	**Build**、**Deploy** **Runtime** (Runtime 条件で使用する場合)

属性	説明	JSON 属性	許可される値	Regex、NOT、AND、OR	フェーズ
Dockerfile Line	命令と引数の両方を含む、Dockerfile の特定の行。	Dockerfile Line	LABEL、RUN、CMD、EXPOSE、ENV、ADD、COPY、ENTRYPOINT、VOLUME、USER、WORKDIR、ONBUILD のいずれか	! 値 (AND、OR) の正規表現のみ	**Build**、**Deploy** **Runtime** (Runtime 条件で使用する場合)
イメージのスキャンステータス	イメージがスキャンされたかどうかを確認します。	スキャンされていないイメージ	ブール値	×	**Build**、**Deploy** **Runtime** (Runtime 条件で使用する場合)
Common Vulnerability Scoring System (CVSS)	CVSS: 指定の CVSS よりスコアが大きい (**>**)、小さい (**<**)、または等しい (**=**) 脆弱性を持つイメージを照合するために使用します。	CVSS	<、>、<=、>=、または何もなし (等しいことを意味します) – と – 10 進数 (オプションの小数値を含む数値)。 例: >=5 または 9.5	AND, OR	**Build**、**Deploy** **Runtime** (Runtime 条件で使用する場合)
重大度	CVSS またはベンダーに基づく脆弱性の重大度。Low、Moderate、Important、Critical のいずれかです。	重大度	<、>、⇐、>=、または何もなし (等しいことを意味します) – と – 次のうちのひとつ: UNKNOWN LOW MODERATE IMPORTANT CRITICAL 例: >=IMPORTANT、または CRITICAL	AND, OR	**Build**、**Deploy** **Runtime** (Runtime 条件で使用する場合)

属性	説明	JSON 属性	許可される値	Regex、NOT、AND、OR	フェーズ
Fixed By	イメージのフラグ付きの脆弱性を修正するパッケージのバージョン文字列。この基準は、CVE 基準などを使用して脆弱性を特定する他の基準に加えて使用される場合があります。	Fixed By	String	正規表現、NOT、AND、OR	**Build**、**Deploy** **Runtime** (Runtime 条件で使用する場合)
CVE	Common Vulnerabilities and Exposures。特定の CVE 番号で使用。	CVE	String	正規表現、NOT、AND、OR	**Build**、**Deploy** **Runtime** (Runtime 条件で使用する場合)
Image Component	イメージに存在する特定のソフトウェアコンポーネントの名前とバージョン番号。	Image Component	key=value value はオプションです。 値が見つからない場合は、"key=" の形式にする必要があります。	正規表現、AND、OR	**Build**、**Deploy** **Runtime** (Runtime 条件で使用する場合)
Image OS	イメージのベースオペレーティングシステムの名前およびバージョン番号。たとえば、**alpine:3.17.3** です。	Image OS	String	正規表現、NOT、AND、OR	**Build**、**Deploy** **Runtime** (Runtime 条件で使用する場合)

属性	説明	JSON 属性	許可される値	Regex、NOT、AND、OR	フェーズ
イメージラベルが必要	Docker イメージラベルが存在することを確認します。このポリシーは、デプロイメントのイメージに指定されたラベルがない場合にトリガーされます。キーおよび値フィールドの両方に正規表現を使用して、ラベルを照合できます。**Require Image Label** ポリシー条件は、Docker レジストリーと統合する場合にのみ機能します。Docker ラベルの詳細は、Docker のドキュメント (https://docs.docker.com/config/labels-custom-metadata/) を参照してください。	Required Image Label	key=value value はオプションです。 値が見つからない場合は、"key=" の形式にする必要があります。	正規表現、AND、OR	**Build**、**Deploy** **Runtime** (Runtime 条件で使用する場合)

属性	説明	JSON 属性	許可される値	Regex、NOT、AND、OR	フェーズ
許可されていないイメージラベル	特定の Docker イメージラベルが使用されていないことを確認します。このポリシーは、デプロイメントのイメージに指定されたラベルがある場合にトリガーされます。キーおよび値フィールドの両方に正規表現を使用して、ラベルを照合できます。'Disallow Image Label policy' 条件は、Docker レジストリーと統合する場合にのみ適用されます。Docker ラベルの詳細は、Docker のドキュメント (https://docs.docker.com/config/labels-custom-metadata/) を参照してください。	Disallowed Image Label	key=value value はオプションです。 値が見つからない場合は、"key=" の形式にする必要があります。	正規表現、AND、OR	**Build**、**Deploy** **Runtime** (Runtime 条件で使用する場合)
セクション: コンテナー設定					
Environment Variable	名前または値で環境変数を確認します。環境変数属性を含むポリシーを作成するときに、ポリシーを照合する環境変数のタイプを選	Environment Variable	RAW=key=value により、デプロイメント YAML で直接指定された環境変数を、特定のキーおよび値と照合します。キーのみを照合する	! キーと値の正規表現 (RAW を使用している場合) AND、OR	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)

属性	説明	JSON 属性	許可される値	Regex、NOT、AND、OR	フェーズ
	択できます。たとえば、デプロイメント YAML で raw 値を直接指定したり、config map、シークレット、フィールド、リソース要求や制限から値への参照を指定したりできます。デプロイメント YAML で直接指定された raw 値以外のタイプの場合、ポリシールールの対応する **value** 属性が無視されます。この場合、ポリシーの照合は、指定された環境変数のタイプの存在に基づいて評価されます。さらに、この条件では、raw 値以外のタイプの空でない **value** 属性を持つポリシーの作成が許可されません。		場合は、**value** 属性と省略できます。 環境変数が設定 YAML で定義されていない場合は、**SOURCE=KEY** という形式を使用できます。**SOURCE** は次のいずれかのオブジェクトです。 ● SECRET_KEY (SecretKeyRef) ● CONFIG_MAP_KEY (ConfigMapRef) ● FIELD (FieldRef) ● RESOURCE_FIELD (ResourceFieldRef) 上記のリストでは、API オブジェクトのラベルを指定してから、括弧内でユーザーインターフェイスのラベルを指定しています。		

属性	説明	JSON 属性	許可される値	Regex、NOT、AND、OR	フェーズ
Container CPU Request	特定のリソース用に予約されているコア数を確認します。	Container CPU Request	<、>、⇐、>=、または何もなし (等しいことを意味します) – と – 10 進数 (オプションの小数値を含む数値) 例: >=5 または 9.5	AND, OR	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)
Container CPU Limit	リソースが使用できるコアの最大数を確認します。	Container CPU Limit	(コンテナーの CPU 要求と同じ)	AND, OR	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)
Container Memory Request	要求される MB 数 (小数値を含む)。	Container Memory Request	(コンテナーの CPU 要求と同じ)	AND, OR	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)
Container Memory Limit	リソースが使用できるメモリーの最大量を確認します。	Container Memory Limit	(コンテナーの CPU 要求と同じ)	AND, OR	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)
Privileged container	デプロイメントが特権モードで設定されているかどうかを確認します。この条件は、各 Pod セキュリティーコンテキスト内の **privileged** フィールドの値のみを確認します。	Privileged Container	ブール値: 各 **PodSecurity Context** の **privileged** フィールドの値が **true** に設定されている場合、**true**	×	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)

属性	説明	JSON 属性	許可される値	Regex、NOT、AND、OR	フェーズ
Root filesystem writeability	デプロイメントが **readOnlyFilesystem** モードで設定されているかどうかを確認します。	Read-Only Root Filesystem	ブール値: 各 **PodSecurityContext** の **readOnlyRootFilesystem** フィールドの値が **true** に設定されている場合、**true**	×	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)
Seccomp Profile Type	デプロイメントに定義された **seccomp** プロファイルのタイプ。**seccomp** オプションが Pod レベルとコンテナーレベルの両方で指定されている場合、コンテナーオプションが Pod オプションをオーバーライドします。Security Context を参照してください。	Seccomp Profile Type	以下のいずれかになります。 UNCONFINED RUNTIME_DEFAULT LOCALHOST	×	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)
Privilege escalation	デプロイメントでコンテナープロセスが親プロセスよりも多くの権限を取得できる場合にアラートを出します。	Allow Privilege Escalation	ブール値	×	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)

属性	説明	JSON 属性	許可される値	Regex、NOT、AND、OR	フェーズ
Drop Capabilities	コンテナーからドロップする必要がある Linux 機能。指定された機能がドロップされない場合にアラートを発行します。たとえば、**SYS_ADMIN** および **SYS_BOOT** を使用して設定されており、デプロイメントでこれら 2 つの機能の **1つだけ** が削除されるか、**どちらも** 削除されない場合、アラートが発生します。	Drop Capabilities	以下のいずれかになります。 ALL AUDIT_CONTROL AUDIT_READ AUDIT_WRITE BLOCK_SUSPEND CHOWN DAC_OVERRIDE DAC_READ_SEARCH FOWNER FSETID IPC_LOCK IPC_OWNER KILL LEASE LINUX_IMMUTABLE MAC_ADMIN MAC_OVERRIDE MKNOD NET_ADMIN NET_BIND_SERVICE NET_BROADCAST NET_RAW SETGID SETFCAP SETPCAP SETUID SYS_ADMIN SYS_BOOT SYS_CHROOT SYS_MODULE SYS_NICE SYS_PACCT SYS_PTRACE SYS_RAWIO SYS_RESOURCE SYS_TIME SYS_TTY_CON	AND	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)

属性	説明	JSON 属性	許可される値	Regex、NOT、AND、OR	フェーズ
Add Capabilities	Raw パケットを送信したり、ファイルパーミッションをオーバーライドする機能など、コンテナーには追加できない Linux 機能。指定された機能が追加されたときにアラートを出します。たとえば、**NET_ADMIN** または **NET_RAW** で設定されており、デプロイメントマニフェスト YAML ファイルにこれら 2 つの機能のうち少なくとも 1 つが含まれている場合、アラートが発生します。	Add Capabilities	AUDIT_CONTROL AUDIT_READ AUDIT_WRITE BLOCK_SUSPEND CHOWN DAC_OVERRIDE DAC_READ_SEARCH FOWNER FSETID IPC_LOCK IPC_OWNER KILL LEASE LINUX_IMMUTABLE MAC_ADMIN MAC_OVERRIDE MKNOD NET_ADMIN NET_BIND_SERVICE NET_BROADCAST NET_RAW SETGID SETFCAP SETPCAP SETUID SYS_ADMIN SYS_BOOT SYS_CHROOT SYS_MODULE SYS_PACCT SYS_PTRACE SYS_RAWIO SYS_RESOURCE SYS_TIME SYS_TTY_CONFIG SYSLOG WAKE_ALARM		**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)
Container Name	コンテナーの名前。	Container Name	String	正規表現、NOT、AND、OR	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)

属性	説明	JSON 属性	許可される値	Regex、NOT、AND、OR	フェーズ
AppArmor プロファイル	コンテナーで使用されるアプリケーション Armor ("AppArmor") プロファイル。	AppArmor プロファイル	String	正規表現、NOT、AND、OR	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)
Liveness Probe	コンテナーが liveness プローブを定義するかどうか。	Liveness Probe	ブール値	×	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)
Readiness Probe	コンテナーが readiness プローブを定義するかどうか。	Readiness Probe	ブール値	×	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)
セクション: デプロイメントメタデータ					
Disallowed Annotation	指定された環境の Kubernetes リソースには存在できないアノテーション。	Disallowed Annotation	key=value value はオプションです。 値が見つからない場合は、"key=" の形式にする必要があります。	正規表現、AND、OR	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)
Required Label	Kubernetes で必要なラベルが存在するかどうかを確認します。	Required Label	key=value value はオプションです。 値が見つからない場合は、"key=" の形式にする必要があります。	正規表現、AND、OR	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)

属性	説明	JSON 属性	許可される値	Regex、NOT、AND、OR	フェーズ
Required Annotation	Kubernetes に必要なアノテーションの有無を確認します。	Required Annotation	key=value value はオプションです。 値が見つからない場合は、"key=" の形式にする必要があります。	正規表現、AND、OR	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)
Runtime Class	デプロイメントの **RuntimeClass**。	Runtime Class	String	正規表現、NOT、AND、OR	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)
ホストネットワーク	**HostNetwork** が有効になっているかどうかを確認します。これは、コンテナーが別のネットワークスタック内に配置されないことを意味します (たとえば、コンテナーのネットワークはコンテナー化されていません)。これは、コンテナーがホストのネットワークインターフェイスに完全にアクセスできることを意味します。	ホストネットワーク	ブール値	×	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)

属性	説明	JSON 属性	許可される値	Regex、NOT、AND、OR	フェーズ
Host PID	Process ID (PID) namespace がコンテナーとホスト間で分離されているかどうかを確認します。これにより、異なる PID namespace 内のプロセスが同じ PID を持つことができます。	Host PID	ブール値	×	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)
Host IPC	ホスト上の IPC (POSIX/SysV IPC) namespace (名前付き共有メモリーセグメント、セマフォ、メッセージキューを分離する namespace) が、コンテナーと共有されているかどうかを確認します。	Host IPC	ブール値	×	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)
Namespace	デプロイメントが属する namespace の名前。	Namespace	String	正規表現、NOT、AND、OR	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)

属性	説明	JSON 属性	許可される値	Regex、NOT、AND、OR	フェーズ
レプリカ	デプロイメントレプリカの数。**oc scale** を使用してデプロイメントのレプリカを 0 から特定の数値にスケーリングする場合は、デプロイメントがポリシーに違反すると、アドミッションコントローラーがこのアクションをブロックします。	レプリカ	<、>、⇐、>=、または何もなし (等しいことを意味します) –と– 10 進数 (オプションの小数値を含む数値)。 例: >=5 または 9.5	NOT、AND、OR	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)
セクション: ストレージ					
Volume Name	ストレージの名前。	Volume Name	String	正規表現、NOT、AND、OR	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)
Volume Source	ボリュームがプロビジョニングされるフォームを示します。たとえば、**persistentVolumeClaim** または **hostPath** です。	Volume Source	String	正規表現、NOT、AND、OR	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)
Volume Destination	ボリュームがマウントされるパス。	Volume Destination	String	正規表現、NOT、AND、OR	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)

属性	説明	JSON 属性	許可される値	Regex、NOT、AND、OR	フェーズ
Volume Type	ボリュームの種別を設定します。	Volume Type	String	正規表現、NOT、AND、OR	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)
マウントされたボリュームの書き込み可能性	書き込み可能な状態でマウントされるボリューム。	書き込み可能なマウント済みボリューム	ブール値	×	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)
マウントの伝播	コンテナーが **Bidirectional**、**Host to Container**、または **None** モードでボリュームをマウントしているかどうかを確認します。	マウントの伝播	以下のいずれかになります。 NONE HOSTTOCONTAINER BIDIRECTIONAL	NOT、AND、OR	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)
ホストマウントの書き込み可能性	リソースが、書き込み権限のあるホストにパスをマウントしている。	Writable Host Mount	ブール値	×	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)
セクション: ネットワーク					
プロトコル	公開されるポートによって使用される TCP や UDP などのプロトコル。	公開ポートプロトコル	String	正規表現、NOT、AND、OR	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)

属性	説明	JSON 属性	許可される値	Regex、NOT、AND、OR	フェーズ
ポート	デプロイメントによって公開されるポート番号。	公開されるポート	<、>、⇐、>=、または何もなし (等しいことを意味します) – と – 整数。 例: 1024 以上 22	NOT、AND、OR	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)
Exposed Node Port	デプロイメントによって外部に公開されるポート番号。	Exposed Node Port	(公開ポートと同じ)	NOT、AND、OR	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)
Port Exposure	ロードバランサーやノードポートなど、サービスの公開方法。	ポートの公開方法	以下のいずれかになります。 UNSET EXTERNAL NODE HOST INTERNAL ROUTE	NOT、AND、OR	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)
検出された予期しないネットワークフロー	検出されたネットワークトラフィックがデプロイメントのネットワークベースラインの一部であるかどうかを確認します。	検出された予期しないネットワークフロー	ブール値	×	**Runtime** ONLY - Network
Ingress Network Policy	イングレス Kubernetes ネットワークポリシーの有無を確認します。	Ingress Network Policy がある	ブール値	正規表現、AND、OR	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)

属性	説明	JSON 属性	許可される値	Regex、NOT、AND、OR	フェーズ
Egress Network Policy	エグレス Kubernetes ネットワークポリシーの有無を確認します。	Egress Network Policy がある	ブール値	正規表現、AND、OR	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)
セクション: プロセスアクティビティー					
プロセス名	デプロイメントで実行されるプロセスの名前。	プロセス名	String	正規表現、NOT、AND、OR	**ランタイム** のみ - プロセス
Process Ancestor	デプロイメントで実行されるプロセスの親プロセスの名前。	Process Ancestor	String	正規表現、NOT、AND、OR	**ランタイム** のみ - プロセス
Process Arguments	デプロイメントで実行されるプロセスのコマンド引数。	Process Arguments	String	正規表現、NOT、AND、OR	**ランタイム** のみ - プロセス
Process UID	デプロイメントで実行されるプロセスの UNIX ユーザー ID。	Process UID	整数	NOT、AND、OR	**ランタイム** のみ - プロセス
Unexpected Process Executed	デプロイメントにあるロックされたプロセスベースラインで、プロセス実行がリスト表示されていないデプロイメントを確認します。	Unexpected Process Executed	ブール値	×	**ランタイム** のみ - プロセス
セクション: Kubernetes アクセス					

属性	説明	JSON 属性	許可される値	Regex、NOT、AND、OR	フェーズ
Service Account	サービスアカウントの名前	Service Account	String	正規表現、NOT、AND、OR	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)
Automount Service Account Token	デプロイメント設定がサービスアカウントトークンを自動的にマウントするかどうかを確認します。	Automount Service Account Token	ブール値	×	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)
Minimum RBAC Permissions	デプロイメントの Kubernetes サービスアカウントに、指定のレベル以上 (**=** または **>**) の Kubernetes RBAC パーミッションレベルがあるかどうかを照合します。	Minimum RBAC Permissions	以下のいずれかになります。 DEFAULT ELEVATED_IN_NAMESPACE ELEVATED_CLUSTER_WIDE CLUSTER_ADMIN	NOT	**デプロイ**、**ランタイム** (ランタイムの条件とともに使用する場合)
セクション: Kubernetes イベント					
Kubernetes Action	**Pod Exec** などの Kubernetes アクションの名前。	Kubernetes Resource	以下のいずれかになります。 PODS_EXEC PODS_PORTFORWARD	! **OR** のみ	**Runtime** ONLY - Kubernetes Events
Kubernetes User Name	リソースにアクセスしたユーザーの名前。	Kubernetes User Name	ハイフン (-) とコロン (:) のみを含む英数字	正規表現、NOT, !**OR** のみ	**Runtime** ONLY - Kubernetes Events
Kubernetes User Group	リソースにアクセスしたユーザーが属するグループの名前。	Kubernetes User Groups	ハイフン (-) とコロン (:) のみを含む英数字	Regex, !**OR** のみ	**Runtime** ONLY - Kubernetes Events

属性	説明	JSON 属性	許可される値	Regex、NOT、AND、OR	フェーズ
Kubernetes Resource Type	アクセスされた Kubernetes リソースのタイプ。	Kubernetes Resource	以下のいずれかになります。 config map シークレット ClusterRoles ClusterRoleBindings NetworkPolicies SecurityContextConstraints EgressFirewalls	! **OR** のみ	**Runtime** ONLY - Audit Log
Kubernetes API Verb	**GET** や **POST** などのリソースへのアクセスに使用される Kubernetes API 動詞。	Kubernetes API Verb	以下のいずれかになります。 CREATE DELETE GET PATCH UPDATE	! **OR** のみ	**Runtime** ONLY - Audit Log
Kubernetes Resource Name	アクセスされた Kubernetes リソースの名前。	Kubernetes Resource Name	ハイフン (-) とコロン (:) のみを含む英数字	正規表現、NOT, !**OR** のみ	**Runtime** ONLY - Audit Log
User Agent	ユーザーがリソースへのアクセスに使用したユーザーエージェント。例: **oc**、または **kubectl**	User Agent	String	正規表現、NOT, !**OR** のみ	**Runtime** ONLY - Audit Log
Source IP Address	ユーザーがリソースにアクセスした IP アドレス。	Source IP Address	IPV4 または IPV6 アドレス	正規表現、NOT, !**OR** のみ	**Runtime** ONLY - Audit Log

属性	説明	JSON 属性	許可される値	Regex、NOT、AND、OR	フェーズ
Is Impersonated User	サービスアカウントまたは他のアカウントで権限を偽装ユーザーによって要求が行われたかどうかを確認します。	Is Impersonated User	ブール値	×	**Runtime** ONLY - Audit Log

6.4.3.1. ポリシー条件への論理条件の追加

ドラッグアンドドロップポリシーフィールドパネルを使用して、ポリシー条件に論理条件を指定できます。

前提条件

- Red Hat Advanced Cluster Security for Kubernetes バージョン 3.0.45 以降を使用している。

手順

1. **Policy Criteria** セクションで、**Add a new condition** を選択して、新しいポリシーセクションを追加します。
   - **Edit** アイコンをクリックして、ポリシーセクションの名前を変更できます。
   - **Drag out a policy field** セクションには、複数のカテゴリーで利用可能なポリシー条件がリスト表示されます。これらのカテゴリーを展開したり折りたたんだりして、ポリシー条件属性を表示できます。
2. policy セクションの **Drop a policy field** エリアに属性をドラッグします。
3. 選択する属性のタイプに応じて、選択した属性の条件を設定するオプションが異なります。以下に例を示します。
   - ブール値が **Read-Only Root Filesystem** の属性を選択すると、**READ-ONLY** オプションおよび **WRITABLE** オプションが表示されます。
   - **環境変数** が複合値の属性を選択すると、**Key**、**Value**、および **Value From** フィールドの値を入力するオプションと、利用可能なオプションの他の値を追加するアイコンが表示されます。
     a. 属性に複数の値を組み合わせるには、**Add** アイコンをクリックします。
     b. ポリシーセクションでリスト表示されている論理演算子 **AND** または **OR** をクリックして、**AND** 演算子と **OR** 演算子を切り替えることもできます。演算子間の切り替えは、ポリシーセクション内でのみ機能し、2 つの異なるポリシーセクション間では機能しません。
4. これらの手順を繰り返して、複数の **AND** および **OR** 条件を指定できます。追加した属性の条件を設定したら、**Next** をクリックしてポリシーの作成を続行します。

6.5. セキュリティーポリシーの共有

Red Hat Advanced Cluster Security for Kubernetes バージョン 3.0.44 以降、ポリシーをエクスポートおよびインポートして、異なる Central インスタンス間でセキュリティーポリシーを共有できます。これは、すべてのクラスターに同じ標準を有効にするのに役立ちます。ポリシーを共有するには、JSON ファイルとしてエクスポートして、別の Central インスタンスにインポートし直す必要があります。

注記

現在、RHACS ポータルを使用して、複数のセキュリティーポリシーを一度にエクスポートすることはできません。ただし、API を使用して複数のセキュリティーポリシーをエクスポートできます。RHACS ポータルで **Help → API reference** に移動し、API リファレンスを確認します。

6.5.1. セキュリティーポリシーのエクスポート

ポリシーをエクスポートすると、ポリシーの内容だけでなく、クラスターの範囲、クラスターの除外、および設定されたすべての通知も含まれます。

手順

1. RHACS ポータルで、**Platform Configuration → Policy Management** に移動します。
2. **Policies** ページから、編集するポリシーを選択します。
3. **Actions → Export policy to JSON** を選択します。

6.5.2. セキュリティーポリシーのインポート

RHACS ポータルの **システム** ポリシービューからセキュリティーポリシーをインポートできます。

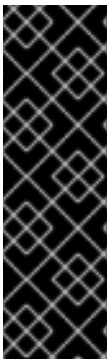
手順

1. RHACS ポータルで、**Platform Configuration → Policy Management** に移動します。
2. **Import policy** をクリックします。
3. **Import policy JSON** ダイアログで **Upload** をクリックし、アップロードする JSON ファイルを選択します。
4. **Begin import** をクリックします。

RHACS の各セキュリティーポリシーには、一意の ID (UID) と一意の名前があります。ポリシーをインポートすると、RHACS はアップロードされたポリシーを次のように処理します。

- インポートされたポリシーの UID と名前が既存のポリシーと一致しないと、RHACS は新しいポリシーを作成します。
- インポートされたポリシーで、既存のポリシーと同じ UID が設定されているが、別の名前の場合には、以下のいずれかを実行できます。
  - 両方のポリシーを保持する。RHACS はインポートされたポリシーを新しい UID で保存します。
  - 既存のポリシーをインポートされたポリシーに置き換える。

- インポートされたポリシーの名前が既存のポリシーと同じものの、UID が異なる場合は、以下のいずれかを実行できます。
  - インポートされたポリシーの新しい名前を指定して、両方のポリシーを保持する。
  - 既存のポリシーをインポートされたポリシーに置き換える。
- インポートされたポリシーの名前が既存のポリシーと同じ場合、Red Hat Advanced Cluster Security for Kubernetes はポリシー条件が既存のポリシーに一致するかどうかを確認します。ポリシー基準が一致する場合、RHACS は既存のポリシーを保持し、成功メッセージを表示します。ポリシー条件が一致しない場合は、以下のいずれかを実行できます。
  - インポートされたポリシーの新しい名前を指定して、両方のポリシーを保持する。
  - 既存のポリシーをインポートされたポリシーに置き換える。

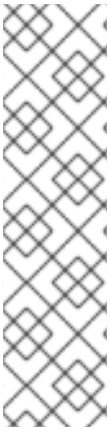


重要

- 同じ Central インスタンスにインポートする場合、RHACS はエクスポートされたすべてのフィールドを使用します。
- 別の Central インスタンスにインポートする場合、RHACS はクラスタースコープ、クラスター除外、通知などの特定のフィールドを省略します。RHACS は、省略されたフィールドをメッセージに表示します。これらのフィールドはインストールごとに異なりますが、フィールドを別の Central インスタンスに移行することはできません。

第7章 デフォルトのセキュリティーポリシー

Red Hat Advanced Cluster Security for Kubernetes のデフォルトのセキュリティーポリシーは、セキュリティーの問題を特定し、環境内のセキュリティーのベストプラクティスを確保するための幅広い範囲を提供します。これらのポリシーを設定することで、環境内でのリスクの高いサービスのデプロイを自動的に防止し、ランタイムのセキュリティーインシデントに対応できます。



注記

Red Hat Advanced Cluster Security for Kubernetes のポリシーの重大度レベルは、Red Hat Product Security が割り当てる重大度レベルとは異なります。

Red Hat Advanced Cluster Security for Kubernetes ポリシーの重大度レベルは Critical、High、Medium、および Low です。Red Hat Product Security の脆弱性の重大度レベルは、重大、重要、中程度、低度の影響となります。

ポリシーの重大度レベルと Red Hat Product Security の重大度レベルは関連していますが、これらを区別することが重要です。Red Hat Product Security の重大度レベルの詳細は、重大度のレベル を参照してください。

7.1. 重大度のセキュリティーポリシー

以下の表に、Red Hat Advanced Cluster Security for Kubernetes のデフォルトの重大度のセキュリティーポリシーを示します。ポリシーは、ライフサイクルステージごとに編成されています。

表7.1 重大度のセキュリティーポリシー

ライフサイクルステージ	名前	説明	ステータス
ビルドまたはデプロイ	Apache Struts: CVE-2017-5638	CVE-2017-5638 Apache Struts の脆弱性を含むイメージがデプロイメントに含まれている場合にアラートを出します。	Enabled
ビルドまたはデプロイ	Log4Shell: log4j リモートコード実行の脆弱性	CVE-2021-44228 および CVE-2021-45046 Log4Shell 脆弱性を含むイメージがデプロイメントに含まれている場合にアラートを出します。バージョン 2.0-beta9 から 2.15.0 (バージョン 2.12.2 を除く) の Apache Log4j Java ロギングライブラリーに欠陥が存在します。	Enabled

ライフサイクルステージ	名前	説明	ステータス
ビルドまたはデプロイ	Spring4Shell (Spring Framework Remote Code Execution) および Spring Cloud Function の脆弱性	Spring MVC に影響を与える CVE-2022-22965 脆弱性と、Spring Cloud に影響を与える CVE-2022-22963 脆弱性のいずれかを含むイメージがデプロイメントに含まれている場合にアラートを出します。バージョン 3.16、3.2.2、およびサポートされていない古いバージョンでは、Spring Cloud に欠陥が含まれています。バージョン 5.3.0 ~ 5.3.17、バージョン 5.2.0 ~ 5.2.19、およびサポートされていない古いバージョンの Spring Framework に欠陥があります。	Enabled
ランタイム	特権コンテナーで実行される iptables	特権 Pod が iptables を実行するときにアラートを出します。	Enabled

7.2. 重大度の高いセキュリティーポリシー

以下の表は、Red Hat Advanced Cluster Security for Kubernetes の重大度の高いデフォルトのセキュリティーポリシーを示しています。ポリシーは、ライフサイクルステージごとに編成されています。

表7.2 重大度の高いセキュリティーポリシー

ライフサイクルステージ	名前	説明	ステータス
ビルドまたはデプロイ	修正可能な 7 以上の Common Vulnerability Scoring System (CVSS)	修正可能な脆弱性を含むデプロイメントの CVSS が 7 以上の場合にアラートを出します。ただし、Red Hat は、CVSS スコアではなく、Common Vulnerabilities and Exposures (CVE) の重大度を使用してポリシーを作成することを推奨します。	Disabled

ライフサイクルステージ	名前	説明	ステータス
ビルドまたはデプロイ	修正可能な重大度が少なくとも「重要な影響」	修正可能な脆弱性を含むデプロイメントの重大度が「重要な影響」以上の場合にアラートを出します。	Enabled
ビルドまたはデプロイ	Rapid Reset: HTTP/2 プロトコルにおけるサービス拒否の脆弱性	HTTP/2 サーバーのサービス拒否 (DoS) 脆弱性の影響を受けやすいコンポーネントを含むイメージのデプロイメントに関するアラート。これは、HTTP/2 での多重化ストリームの処理における不具合に対処します。クライアントはリクエストを迅速に作成し、すぐにリセットできます。これにより、サーバー側の制限に達することを回避しながらサーバーに余分な作業が発生し、サービス拒否攻撃が発生する可能性があります。このポリシーを使用するには、ポリシーを複製し、有効にする前に **Fixable** ポリシー条件を追加することを検討してください。	Disabled
ビルドまたはデプロイ	イメージで公開されているセキュアシェル (ssh) ポート	一般に SSH アクセス用に予約されているポート 22 がデプロイで公開されたときにアラートを出します。	Enabled
デプロイ	緊急デプロイメントのアノテーション	デプロイメントで StackRox アドミッションコントローラーのチェックを回避するために "admission.stackrox.io/break-glass":"ticket-1234" などの緊急アノテーションが使用される場合にアラートを出します。	Enabled
デプロイ	環境変数に Secret が含まれています	デプロイメントに 'SECRET' を含む環境変数がある場合にアラートを出します。	Enabled

ライフサイクルステージ	名前	説明	ステータス
デプロイ	修正可能な CVSS >= 6 および特権	デプロイが特権モードで実行され、CVSS が 6 以上の修正可能な脆弱性がある場合にアラートを出します。ただし、Red Hat は、CVSS スコアではなく CVE の重大度を使用してポリシーを作成することを推奨します。	バージョン 3.72.0 以降ではデフォルトで Disabled
デプロイ	重要かつ重大な修正可能な CVE を含む特権コンテナー	特権モードで実行されるコンテナーに重要または重大な修正可能な脆弱性がある場合にアラートを出します。	Enabled
デプロイ	環境変数としてマウントされた Secret	環境変数としてマウントされた Kubernetes シークレットがデプロイメントに含まれている場合にアラートを出します。	Disabled
デプロイ	セキュアシェル (ssh) ポートの公開	一般に SSH アクセス用に予約されているポート 22 がデプロイで公開されたときにアラートを出します。	Enabled
ランタイム	暗号通貨マイニングプロセスの実行	暗号通貨マイニングプロセスを生成します。	Enabled
ランタイム	iptables の実行	誰かが iptables を実行したことを検出します。これは、コンテナー内のネットワーク状態を管理する非推奨の方法です。	Enabled
ランタイム	Kubernetes アクション: Exec into Pod	コンテナーでコマンドを実行する要求を Kubernetes API が受信したときにアラートを出します。	Enabled
ランタイム	Linux グループ追加の実行	誰かが addgroup または groupadd バイナリーを実行して Linux グループを追加したことを検出します。	Enabled

ライフサイクルステージ	名前	説明	ステータス
ランタイム	Linux ユーザー追加の実行	誰かが useradd または adduser バイナリーを実行して Linux ユーザーを追加したことを検出します。	Enabled
ランタイム	ログインバイナリー	誰かがログインを試みたことを示します。	Disabled
ランタイム	ネットワーク管理の実行	ネットワークの設定と管理を操作できるバイナリーファイルが誰かによって実行されたことを検出します。	Enabled
ランタイム	nmap の実行	ランタイム中に誰かがコンテナー内で nmap プロセスを開始したときにアラートを出します。	Enabled
ランタイム	OpenShift: Kubeadmin Secret へのアクセス	誰かが kubeadmin Secret にアクセスしたときにアラートを出します。	Enabled
ランタイム	パスワードバイナリー	誰かがパスワードを変更しようとしたことを示します。	Disabled
ランタイム	クラスター Kubelet エンドポイントを対象とするプロセス	healthz、kubelet API、または heapster エンドポイントの誤用を検出します。	Enabled
ランタイム	プロセスターゲットクラスター Kubernetes Docker Stats エンドポイント	Kubernetes docker stats エンドポイントの誤用を検出します。	Enabled
ランタイム	プロセスターゲティング Kubernetes サービスエンドポイント	Kubernetes サービス API エンドポイントの誤用を検出します。	Enabled
ランタイム	UID 0 のプロセス	デプロイメントに UID 0 で実行されるプロセスが含まれている場合にアラートを出します。	Disabled

ライフサイクルステージ	名前	説明	ステータス
ランタイム	セキュアシェルサーバー (sshd) の実行	SSH デーモンを実行するコンテナーを検出します。	Enabled
ランタイム	SetUID プロセス	エスカレートされた特権で特定のプログラムを実行できるようにする setuid バイナリーファイルを使用します。	Disabled
ランタイム	シャドウファイルの変更	誰かがシャドウファイルを変更しようとしたことを示します。	Disabled
ランタイム	Java アプリケーションによって生成されたシェル	bash、csh、sh、zsh などのシェルが Java アプリケーションのサブプロセスとして実行されるタイミングを検出します。	Enabled
ランタイム	不正なネットワークフロー	"異常な違反に関するアラート" 設定のベースラインから外れたネットワークフローに対して違反を生成します。	Enabled
ランタイム	不正なプロセス実行	Kubernetes デプロイメントのコンテナー仕様のロックされたプロセスベースラインによって明示的に許可されていないプロセス実行に対して違反を生成します。	Enabled

7.3. 重大度が中程度のセキュリティーポリシー

以下の表は、Red Hat Advanced Cluster Security for Kubernetes のデフォルトの重大度が中程度のセキュリティーポリシーを示しています。ポリシーは、ライフサイクルステージごとに編成されています。

表7.3 重大度が中程度のセキュリティーポリシー

ライフサイクルステージ	名前	説明	ステータス

ライフサイクルステージ	名前	説明	ステータス
Build	Docker CIS 4.4: セキュリティーパッチを含むイメージのスキャンと再構築の確認	イメージがスキャンされず、セキュリティーパッチを含むように再構築されていない場合に警告します。イメージを頻繁にスキャンして脆弱性を見つけ、イメージを再構築してセキュリティーパッチを含め、イメージのコンテナーをインスタンス化することが重要です。	Disabled
デプロイ	30 日間のスキャン期間	デプロイメントが 30 日間スキャンされていない場合にアラートを出します。	Enabled
デプロイ	CAP_SYS_ADMIN 機能が追加されました	デプロイに CAP_SYS_ADMIN でエスカレートしているコンテナーが含まれている場合にアラートを出します。	Enabled
デプロイ	読み書き可能なルートファイルシステムを使用するコンテナー	デプロイに読み取り/書き込みルートファイルシステムを持つコンテナーが含まれている場合にアラートを出します。	Disabled
デプロイ	権限のエスカレーションが許可されたコンテナー	コンテナーが意図しない権限で実行され、セキュリティーリスクが発生している可能性がある場合にアラートを出します。この状況は、親プロセスよりも多くの権限を持つコンテナープロセスが、意図しない権限でコンテナーを実行できる場合に発生する可能性があります。	Enabled
デプロイ	デプロイメントには、1 つ以上のイングレスネットワークポリシーが必要	デプロイメントにイングレスネットワークポリシーが欠落している場合にアラートします。	Disabled

ライフサイクルステージ	名前	説明	ステータス
デプロイ	外部に公開されたエンドポイントを使用したデプロイメント	何らかの方法で外部に公開されているサービスがデプロイメントに含まれているかどうかを検出します。クラスター外に公開されるサービスを使用するデプロイメントは、クラスター外から到達できるため、侵入を試みるリスクが高くなります。このポリシーは、クラスター外にサービス公開する必要があるか検証できるように、アラートを提供します。サービスがクラスター内の通信のみに必要な場合は、サービスタイプ ClusterIP を使用します。	Disabled
デプロイ	Docker CIS 5.1: 該当する場合は、AppArmor プロファイルが有効になっていることを確認します	AppArmor プロファイルと呼ばれるセキュリティーポリシーを適用することで、AppArmor を使用して Linux オペレーティングシステムとアプリケーションを保護します。AppArmor は、Debian や Ubuntu などの一部の Linux ディストリビューションでデフォルトで利用できる Linux アプリケーションセキュリティーシステムです。	Enabled
デプロイ	Docker CIS 5.15: ホストのプロセス namespace が共有されていないことを確認する	コンテナーとホストの間にプロセスレベルの分離を作成します。プロセス ID (PID) namespace はプロセス ID 空間を分離します。つまり、異なる PID namespace のプロセスが同じ PID を持つことができます。	Enabled

ライフサイクルステージ	名前	説明	ステータス
デプロイ	Docker CIS 5.16: ホストの IPC namespace が共有されていないことを確認する	ホスト上の IPC namespace がコンテナーと共有されている場合にアラートを出します。IPC (POSIX/SysV IPC) namespace は、名前付き共有メモリーセグメント、セマフォ、およびメッセージキューを分離します。	Enabled
デプロイ	Docker CIS 5.19: マウント伝播モードが有効になっていないことを確認する	マウント伝搬モードが有効になっている場合にアラートを出します。マウント伝達モードが有効になっている場合、コンテナーボリュームを双方向、ホストからコンテナー、およびなしモードでマウントできます。明示的に必要な場合を除き、双方向マウント伝搬モードを使用しないでください。	Enabled
デプロイ	Docker CIS 5.21: デフォルトの seccomp プロファイルが無効になっていないことを確認する	seccomp プロファイルが無効になったときに警告します。seccomp プロファイルは、許可リストを使用して一般的なシステムコールを許可し、その他すべてをブロックします。	Disabled
デプロイ	Docker CIS 5.7: 特権ポートがコンテナー内にマップされていないことを確認する	特権ポートがコンテナー内でマップされたときにアラートを出します。1024 未満の TCP/IP ポート番号は特権ポートです。セキュリティー上の理由から、通常のユーザーとプロセスはそれらを使用できませんが、コンテナーはそれらのポートを特権ポートにマップする場合があります。	Enabled

ライフサイクルステージ	名前	説明	ステータス
デプロイ	Docker CIS 5.9 および 5.20: ホストのネットワーク namespace が共有されていないことを確認する	ホストのネットワーク namespace が共有されている場合に警告します。HostNetwork が有効な場合、コンテナーは別のネットワークスタック内に配置されず、コンテナーのネットワークはコンテナー化されません。その結果、コンテナーはホストのネットワークインターフェイスに完全にアクセスでき、共有 UTS namespace が有効になります。UTS namespace は、ホスト名と NIS ドメイン名を分離し、その namespace で実行中のプロセスから認識されるホスト名とドメインを設定します。コンテナー内で実行されるプロセスは通常、ホスト名またはドメイン名を知る必要がないため、UTS namespace をホストと共有しないでください。	Enabled
デプロイ	スキャンなしのイメージ	デプロイメントにスキャンされていないイメージが含まれている場合にアラートを出します。	Disabled
ランタイム	Kubernetes アクション: Pod へのポート転送	Kubernetes API がポート転送リクエストを受信したときにアラートを出します。	Enabled
デプロイ	コンテナーランタイムソケットのマウント	デプロイでコンテナーランタイムソケットにボリュームマウントがある場合にアラートを出します。	Enabled
デプロイ	重要なホストディレクトリーのマウント	デプロイメントが機密性の高いホストディレクトリーをマウントするときにアラートを出します。	Enabled

ライフサイクルステージ	名前	説明	ステータス
デプロイ	リソース要求または制限が指定されていません	リソースの要求と制限がないコンテナーがデプロイメントに含まれている場合にアラートを出します。	Enabled
デプロイ	自動的にマウントされる Pod サービスアカウントトークン	アプリケーションが Kubernetes API との対話を必要とする Pod のみにデフォルトサービスアカウントトークンのマウントを最小限に抑えることで、Pod のデフォルトサービスアカウントトークンが侵害されないように保護します。	Enabled
デプロイ	Privileged Container	デプロイメントに特権モードで実行されるコンテナーが含まれている場合にアラートを出します。	Enabled
ランタイム	crontab の実行	crontab スケジュールジョブエディターの使用を検出します。	Enabled
ランタイム	Netcat の実行が検出されました	netcat がコンテナー内で実行されるタイミングを検出します。	Enabled
ランタイム	OpenShift: Advanced Cluster Security Central Admin Secret へのアクセス	誰かが Red Hat Advanced Cluster Security Central Secret にアクセスしたときにアラートを出します。	Enabled
ランタイム	OpenShift: なりすましユーザーがアクセスする Kubernetes Secret	誰かがユーザーになりすましてクラスター内の Secret にアクセスしたときにアラートを出します。	Enabled
ランタイム	リモートファイルコピーバイナリー実行	デプロイメントでリモートファイルコピーツールが実行されたときにアラートを出します。	Enabled

7.4. 重大度の低いセキュリティーポリシー

以下の表は、重要度が低い Red Hat Advanced Cluster Security for Kubernetes のデフォルトのセキュリティーポリシーを示しています。ポリシーは、ライフサイクルステージごとに編成されています。

表7.4 重大度の低いセキュリティーポリシー

ライフサイクルステージ	名前	説明	ステータス
ビルドまたはデプロイ	イメージの 90 日間経過	デプロイメントが 90 日間更新されていない場合にアラートを出します。	Enabled
ビルドまたはデプロイ	COPY の代わりに使用される ADD コマンド	デプロイメントで ADD コマンドが使用されたときにアラートを出します。	Disabled
ビルドまたはデプロイ	イメージ内の Alpine Linux Package Manager (apk)	デプロイに Alpine Linux パッケージマネージャー (apk) が含まれている場合にアラートを出します。	Enabled
ビルドまたはデプロイ	イメージの curl	デプロイメントに curl が含まれている場合にアラートを出します。	Disabled
ビルドまたはデプロイ	Docker CIS 4.1: コンテナーのユーザーが作成されていることを確認する	コンテナーが非 root ユーザーとして実行されていることを確認します。	Enabled
ビルドまたはデプロイ	Docker CIS 4.7: 更新指示に関するアラート	Dockerfile で更新命令が単独で使用されないようにします。	Enabled
ビルドまたはデプロイ	CMD で指定された安全でない	デプロイでコマンドに 'insecure' が使用されている場合にアラートを出します。	Enabled
ビルドまたはデプロイ	latest タグ	'latest' タグを使用するイメージがデプロイメントに含まれている場合にアラートを出します。	Enabled
ビルドまたはデプロイ	イメージの Red Hat Package Manager	デプロイメントに Red Hat、Fedora、または CentOS パッケージ管理システムのコンポーネントが含まれている場合にアラートを出します。	Enabled

ライフサイクルステージ	名前	説明	ステータス
ビルドまたはデプロイ	Required Image Label	指定されたラベルがないイメージがデプロイメントに含まれている場合にアラートを出します。	Disabled
ビルドまたはデプロイ	Ubuntu パッケージマネージャーの実行	Ubuntu パッケージ管理システムの使用状況を検出します。	Enabled
ビルドまたはデプロイ	イメージの Ubuntu パッケージマネージャー	デプロイメントのイメージに Debian または Ubuntu パッケージ管理システムのコンポーネントが含まれている場合にアラートを出します。	Enabled
ビルドまたはデプロイ	イメージ内の Wget	デプロイメントに wget が含まれている場合にアラートを出します。	Disabled
デプロイ	すべての機能を削除する	デプロイメントですべての機能が削除されない場合に警告します。	Disabled
デプロイ	Orchestrator Secrets ボリュームの不適切な使用	デプロイメントで 'VOLUME /run/secrets' を含む Dockerfile が使用されている場合にアラートを出します。	Enabled
デプロイ	Kubernetes ダッシュボードがデプロイされました	Kubernetes ダッシュボードサービスが検出されたときにアラートを出します。	Enabled
デプロイ	必須のアノテーション: Email	デプロイメントに 'email' アノテーションが欠落している場合にアラートを出します。	Disabled
デプロイ	必要なアノテーション: Owner/Team	デプロイメントに 'owner' または 'team' アノテーションがない場合にアラートを出します。	Disabled

ライフサイクルステージ	名前	説明	ステータス
デプロイ	必要なラベル: Owner/Team	デプロイメントに 'owner' または 'team' ベルがない場合にアラートを出します。	Disabled
ランタイム	Alpine Linux パッケージマネージャーの実行	実行時に Alpine Linux パッケージマネージャー (apk) が実行されたときにアラートを出します。	Enabled
ランタイム	chkconfig の実行	通常、コンテナーでは使用されない ckconfig サービスマネージャーの使用を検出します。	Enabled
ランタイム	コンパイラーツールの実行	ソフトウェアをコンパイルするバイナリーファイルが実行時に実行されると警告します。	Enabled
ランタイム	Red Hat Package Manager の実行	実行時に Red Hat、Fedora、または CentOS パッケージマネージャープログラムが実行されたときにアラートを出します。	Enabled
ランタイム	シェル管理	シェルを追加または削除するコマンドが実行されたときに警告します。	Disabled
ランタイム	systemctl の実行	systemctl サービスマネージャーの使用状況を検出します。	Enabled
ランタイム	systemd の実行	systemd サービスマネージャーの使用状況を検出します。	Enabled

第8章 ネットワークポリシーの管理

Kubernetes ネットワークポリシー は、Pod のグループを相互およびその他のネットワークエンドポイントと通信できるようにする仕様です。これらのネットワークポリシーは YAML ファイルとして設定されます。これらのファイルだけを見ると、適用されたネットワークポリシーが目的のネットワークトポロジーを実現しているかどうかを特定するのが難しいことがよくあります。

Red Hat Advanced Cluster Security for Kubernetes (RHACS) は、定義されたすべてのネットワークポリシーをオーケストレーターから収集し、これらのポリシーを使いやすくするツールを備えています。

ネットワークポリシーの適用をサポートするために、RHACS は次のツールを提供します。

- ネットワークグラフ
- ネットワークポリシージェネレーター
- ネットワークポリシーシミュレーター
- ビルド時のネットワークポリシージェネレーター

8.1. ネットワークグラフ

8.1.1. ネットワークグラフについて

ネットワークグラフは、環境内のデプロイメント、ネットワークフロー、およびネットワークポリシーに関する高レベルの詳細情報を提供します。

RHACS は、それぞれのセキュアクラスター内のすべてのネットワークポリシーを処理して、どのデプロイメントが相互に通信できるか、またどのデプロイメントが外部ネットワークに到達できるかを示します。また、実行中のデプロイメントを監視し、デプロイメント間のトラフィックを追跡します。ネットワークグラフでは次の項目を表示できます。

内部エンティティー

これらは、RFC 1918 で定義されているプライベートアドレス空間に属する IP アドレスとデプロイメント間の接続を表します。詳細は、「内部エンティティーが関係する接続」を参照してください。

外部エンティティー

これらは、RFC 1918 で定義されているプライベートアドレス空間に属さない IP アドレスとデプロイメント間の接続を表します。詳細は、「ネットワークグラフの外部エンティティーおよび接続」を参照してください。

ネットワークコンポーネント

上部のメニューから、選択したクラスター (**CL** ラベルで示される) のグラフに表示する namespace (**NS** ラベルで示される) とデプロイメント (**D** ラベルで示される) を選択できます。ドロップダウンリストを使用し、Common Vulnerabilities and Exposures (CVE)、ラベル、イメージなどのフィルタリングの条件を選択することで、デプロイメントをさらにフィルタリングできます。

ネットワークフロー

グラフには次のいずれかのフローを選択できます。

アクティブなトラフィック

このデフォルトオプションを選択すると、選択した namespace または特定のデプロイメントに焦点を当てた、観測されたトラフィックが表示されます。情報を表示する期間を選択できます。

非アクティブなフロー

このオプションを選択すると、ネットワークポリシーで許可されている潜在的なフローが表示され、より厳密な分離を実現するために必要な欠落しているネットワークポリシーを特定するのに役立ちます。情報を表示する期間を選択できます。

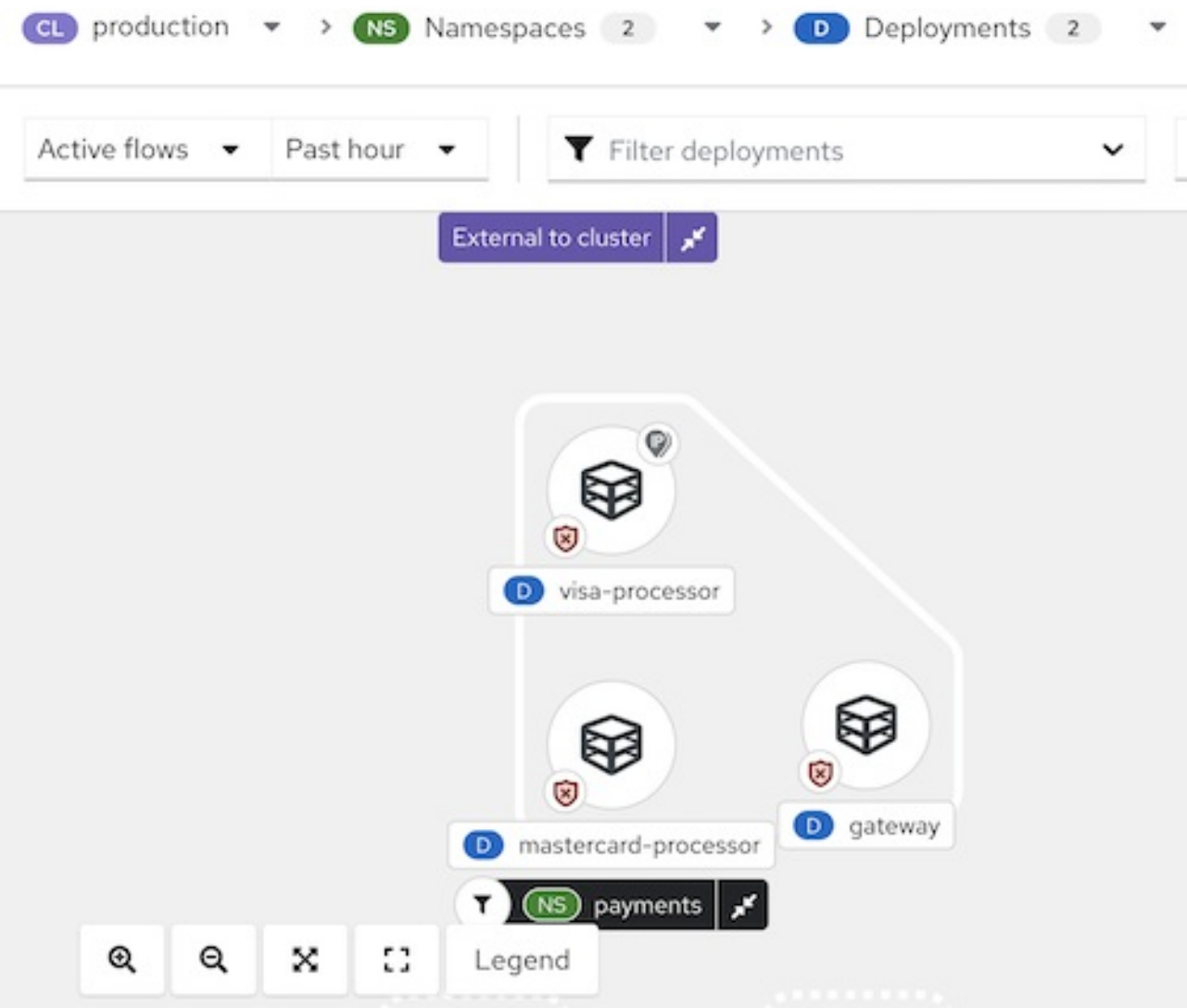
ネットワークポリシー

選択したコンポーネントの既存のポリシーを表示したり、ポリシーのないコンポーネントを表示したりできます。ネットワークグラフビューからネットワークポリシーをシミュレーションすることもできます。詳細は、"ネットワークグラフからのネットワークポリシーのシミュレーション" を参照してください。

8.1.1.1. ネットワークグラフの表示、ナビゲーション、およびユーザーインターフェイス

ネットワークグラフ (下図参照) を使用すると、各項目をクリックし、その項目に関する追加情報を確認できます。グラフ内では、ベースラインにネットワークフローを追加するなどの操作を実行することもできます。

図8.1 ネットワークグラフの例



ネットワークグラフを使用する際は、次のヒントが役立ちます。

- 凡例を開くと、使用されているシンボルとその意味に関する情報が表示されます。凡例には、ネットワークグラフ上の namespace、デプロイメント、および接続を表す記号の説明テキストが表示されます。

- ドロップダウンリストから追加の表示オプションを選択すると、ネットワークポリシーステータスバッジ、アクティブな外部トラフィックバッジ、エッジ接続のポートおよびプロトコルラベルなどのアイコンをグラフに表示するかどうかを制御します。

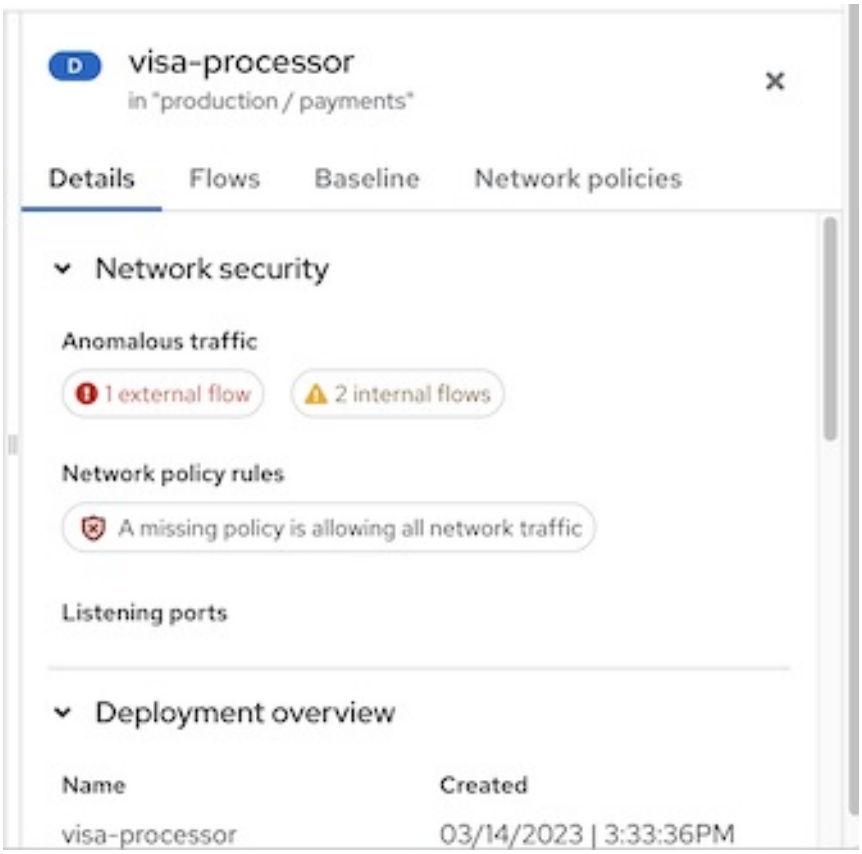
- RHACS は、ノードの参加または離脱など、ネットワークトラフィックの変化を検出します。変更が検出されると、ネットワークグラフに利用可能な更新の数を示す通知が表示されます。集中力が中断されないように、グラフは自動的には更新されません。通知をクリックしてグラフを更新します。

グラフ内の項目をクリックすると、折りたたみ可能なセクションを含む再配置されたサイドパネルに、その項目に関する情報が表示されます。次の項目をクリックできます。

- デプロイメント
- namespace
- 外部エンティティー
- CIDR ブロック
- 外部グループ

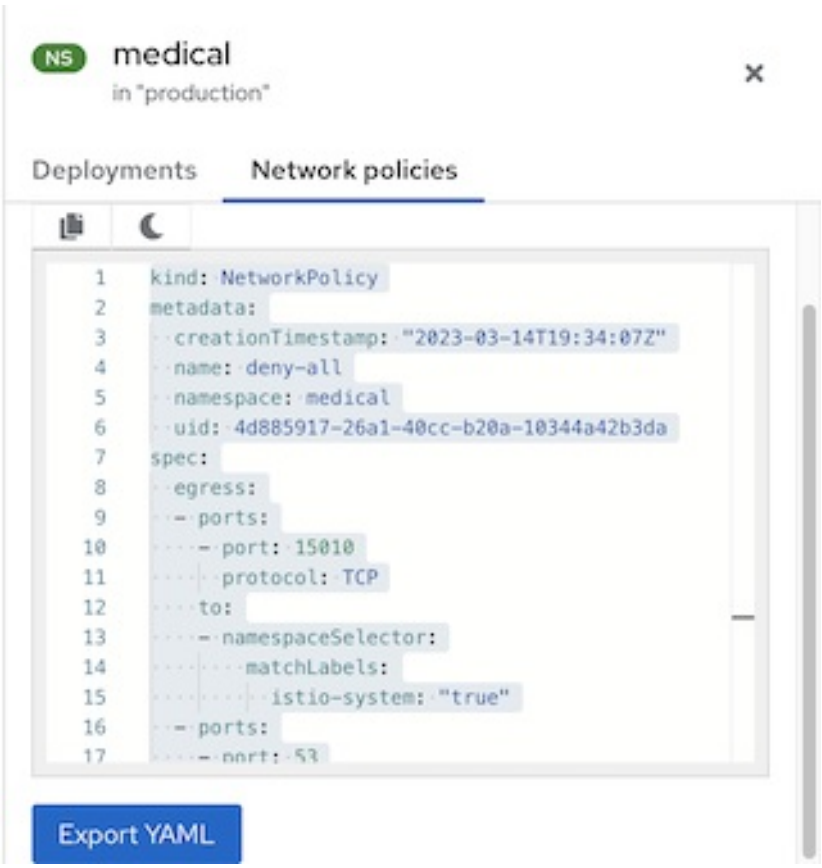
サイドパネルには、選択したグラフ内の項目に基づいた関連情報が表示されます。ヘッダー内の項目名の横にある **D** または **NS** ラベル (この例では "visa-processor") は、それがデプロイメントであるか namespace であるかを示します。以下の例は、デプロイメントのサイドパネルを示しています。

図8.2 デプロイメントの例のサイドパネル



namespace を表示すると、サイドパネルに検索バーとデプロイメントのリストが表示されます。デプロイメントをクリックして、その情報を表示できます。サイドパネルには **Network policies** タブも含まれます。このタブから、次の例に示すように、その namespace で定義されているネットワークポリシーを表示、クリップボードにコピー、またはエクスポートできます。

図8.3 namespace の例のサイドパネル



8.1.1.2. ネットワークグラフの外部エンティティーおよび接続

ネットワークグラフビューには、マネージドクラスターと外部ソース間のネットワーク接続が表示されます。さらに、RHACS は、Google Cloud、AWS、Microsoft Azure、Oracle Cloud、Cloudflare などのパブリッククラスレスドメイン間ルーティング (CIDR) アドレスブロックを自動的に検出して強調表示します。この情報を使用すると、アクティブな外部接続のあるデプロイメントを特定し、ネットワークの外部から不正な接続を行っているかどうかを判断できます。

デフォルトでは、外部接続は、ネットワークグラフ内の共通の **External Entities** アイコンと異なる CIDR アドレスブロックを指します。ただし、**Manage CIDR blocks** をクリックし、**Auto-discovered CIDR blocks** を選択解除すると、自動検出された CIDR ブロックを表示しないように選択できます。

RHACS には、次のクラウドプロバイダーの IP 範囲が含まれています。

- Google Cloud
- AWS
- Microsoft Azure
- Oracle Cloud
- Cloudflare

RHACS は、クラウドプロバイダーの IP 範囲を 7 日ごとに取得して更新し、CIDR ブロックを毎日更新します。オフラインモードを使用している場合は、新しいサポートパッケージをインストールしてこれらの範囲を更新できます。

次の図は、ネットワークグラフの例を示しています。この例では、ユーザーが選択したオプションに基づいて、選択した namespace 内のデプロイメントがグラフに表示されています。トラフィックフロー

は、デプロイメントなどの項目をクリックするまで表示されません。グラフでは赤いバッジを使用して、ポリシーが欠落しているため、すべてのネットワークトラフィックが許可されているデプロイメントを示します。

8.1.1.3. 内部エンティティーに関連する接続

ネットワークグラフは、既知のデプロイメントまたは CIDR ブロックに属さないエンティティーへのアクティブな接続を持つデプロイメントを識別するのに役立ちます。このような接続の一部は、クラスターの外部に到達することなく、クラスターのプライベートネットワーク内で確立されます。ネットワークグラフは、そのような接続を **内部エンティティー** への接続または内部エンティティーからの接続として表します。

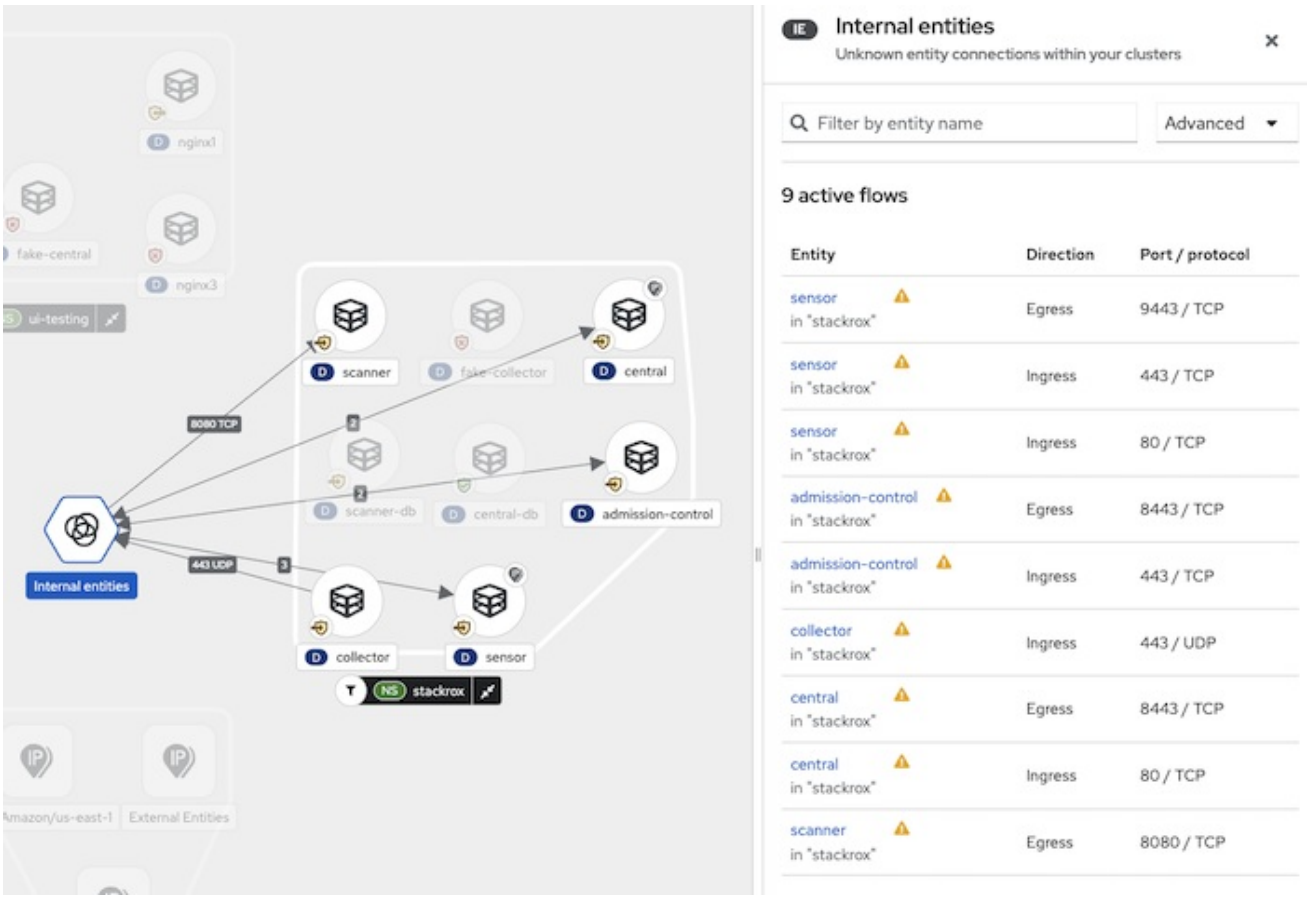
内部エンティティーとの接続は、RFC 1918 で定義されているように、プライベートアドレス空間に属する IP アドレスとデプロイメントの間の接続を表します。場合によっては、接続に関係する一方または両方のデプロイメントを Sensor が識別できないことがあります。その場合、システムが IP アドレスを分析し、接続が内部か外部かを判断します。

次の場合、接続が内部エンティティーに関係するものとして分類されることがあります。

- 接続を開始する側 (クライアント) が接続を試みている間に、IP アドレスが変更されるか、接続を受け入れるデプロイメント (サーバー) が削除された。
- オーケストレーター API と通信するデプロイメント。
- Calico などのネットワーク CNI プラグインを使用して通信するデプロイメント。
- Sensor の再起動により、過去のデプロイメントへの IP アドレスのマッピングがリセットされた。たとえば、Sensor が過去のエンティティーの IP アドレスまたは既存のエンティティーの過去の IP アドレスを認識しない場合などです。
- オーケストレーターによって管理されていないエンティティー (場合によっては、**クラスター外部エンティティー** と見なされるもの) が関係するが、RFC 1918 で定義されているプライベートアドレス空間の IP アドレスを使用している接続。

内部エンティティーは、次の図に示すアイコンで示されます。**Internal entities** をクリックすると、これらのエンティティーのフローが表示されます。

図8.4 内部エンティティーの例



8.1.2. アクセス制御およびパーミッション

ネットワークグラフを表示するには、ユーザーは少なくとも **Network Graph Viewer**のデフォルト権限セットに付与された権限を持っている必要があります。

以下のパーミッションが **Network Graph Viewer**パーミッションセットに付与されます。

- **Deployment** の読み取り
- **NetworkGraph** の読み取り
- **NetworkPolicy** の読み取り

詳細は、「関連情報」セクションの「システム権限セット」を参照してください。

関連情報

- システム権限セット

8.1.3. デプロイメント情報の表示

ネットワークグラフは、RHACS が検出したデプロイメント、namespace、接続の視覚的なマップを提供します。グラフ内のデプロイメントをクリックすると、次の詳細を含むデプロイメントに関する情報を表示できます。

- ネットワークセキュリティー (フローの数、既存または欠落しているネットワークポリシールール、リスニングポートなど)

- ラベルとアノテーション
- ポート設定
- コンテナー情報
- プロトコルとポート番号を含む、イングレスおよびエグレス接続の異常なフローとベースラインフロー
- ネットワークポリシー

手順

namespace 内のデプロイメントの詳細を表示するには:

1. RHACS ポータルで、**Network Graph** に移動し、ドロップダウンリストからクラスターを選択します。
2. **Namespaces** リストをクリックし、検索フィールドを使用して namespace を見つけるか、個々の namespace を選択します。
3. **Deployments** リストをクリックし、検索フィールドを使用してデプロイメントを見つけるか、ネットワークグラフに表示する個々のデプロイメントを選択します。
4. ネットワークグラフでデプロイメントをクリックして情報パネルを表示します。
5. **Details**、**Flows**、**Baseline**、または **Network policies** タブをクリックして、対応する情報を表示します。

8.1.4. ネットワークグラフでのネットワークポリシーの表示

ネットワークポリシーでは、Pod のグループ間および他のネットワークのエンドポイントとの間で許可される通信を指定します。Kubernetes **NetworkPolicy** リソースはラベルを使用して Pod を選択し、選択した Pod との間で許可されるトラフィックを指定するルールを定義します。RHACS は、すべての Kubernetes クラスター、namespace、デプロイメント、および Pod のネットワークポリシー情報を検出し、ネットワークグラフに表示します。

手順

1. RHACS ポータルで、**Network Graph** に移動し、ドロップダウンリストからクラスターを選択します。
2. **Namespaces** リストをクリックして個々の namespace を選択するか、検索フィールドを使用して namespace を検索します。
3. **Deployments** 一覧をクリックして個別のデプロイメントを選択するか、検索フィールドを使用してデプロイメントを特定します。
4. ネットワークグラフでデプロイメントをクリックして情報パネルを表示します。
5. **Details** タブの **Network security** セクションで、次の情報を示すネットワークポリシールールに関する概要メッセージを表示できます。
   - イングレスまたはエグレストラフィックを規制するポリシーがネットワークに存在する場合

- ネットワークにポリシーがないため、すべてのイングレスまたはエグレストラフィックが許可されている場合

6. ネットワークポリシーの YAML ファイルを表示するには、ポリシールールをクリックするか、**Network policies** タブをクリックします。

8.1.5. ネットワークグラフでの CIDR ブロックの設定

カスタム CIDR ブロックを指定したり、ネットワークグラフで自動検出された CIDR ブロックの表示を設定したりできます。

手順

1. RHACS ポータルで **Network Graph** に移動し、**Manage CIDR Blocks** を選択します。次のアクションを実行できます。

   - **Auto-discovered CIDR blocks** を切り替えて、ネットワークグラフで自動検出された CIDR ブロックを非表示にします。

     注記

     自動検出された CIDR ブロックを非表示にすると、ネットワークグラフで選択したクラスターだけでなく、すべてのクラスターに対して自動検出された CIDR ブロックが非表示になります。

   - 次の手順を実行して、カスタム CIDR ブロックをグラフに追加します。

     a. フィールドに CIDR 名と CIDR アドレスを入力します。追加の CIDR ブロックを追加するには、**Add CIDR block** をクリックし、各ブロックの情報を入力します。

     b. **設定の更新** をクリックして変更を保存します。

8.2. ネットワークグラフを使用したネットワークポリシーの生成およびシミュレート

8.2.1. ネットワークグラフからのポリシーの生成について

Kubernetes ネットワークポリシーは、受信ネットワークトラフィックを受信する Pod と、送信トラフィックを送信する Pod を制御します。ネットワークポリシーを使用して Pod へのトラフィックを有効にし、無効にすることで、ネットワークの攻撃エリアを制限できます。

これらのネットワークポリシーは YAML 設定ファイルです。通常、ネットワークフローに関するインサイトを得て、手動でこれらのファイルを作成するのは困難です。RHACS を使用して、これらのファイルを生成できます。ネットワークポリシーを自動的に生成する場合、RHACS は次のガイドラインに従います。

- RHACS は、namespace 内のデプロイメントごとに単一のネットワークポリシーを生成します。ポリシーの Pod セレクターは、デプロイメントの Pod セレクターです。
  - デプロイメントにすでにネットワークポリシーがある場合、RHACS は新しいポリシーを生成したり、既存のポリシーを削除したりしません。
    生成されたポリシーは、トラフィックを既存のデプロイメントに制限するだけです。

- 後で作成するデプロイメントには、新しいネットワークポリシーを作成または生成しないかぎり、制限はありません。
- 新しいデプロイメントでネットワークポリシーを使用してデプロイメントに接続する必要がある場合は、ネットワークポリシーを編集してアクセスを許可する必要があります。

- 各ポリシーにはデプロイメント名と同じ名前が付けられ、その後に **stackrox-generated-** が付けられます。たとえば、生成されたネットワークポリシーのデプロイメント **depABC** のポリシー名は **stackrox-generated-depABC** です。生成されたすべてのポリシーには、識別ラベルもあります。

- RHACS は、次の条件のいずれかが満たされる場合に、任意の IP アドレスからのトラフィックを許可する単一のルールを生成します。
  - デプロイメントに、選択した時間内にクラスターの外部からの受信接続がある場合
  - デプロイメントがノードポートまたはロードバランサーサービスを通じて公開される場合

- RHACS は、受信接続が存在するデプロイメントごとに 1 つの **ingress** ルールを生成します。
  - デプロイメントが同じ namespace にある場合には、このルールは他のデプロイメントの Pod セレクターラベルを使用します。
  - デプロイメントが異なる namespace にある場合には、このルールは namespace セレクターを使用します。これを可能にするために、RHACS はラベル **namespace.metadata.stackrox.io/name** を各 namespace に自動的に追加します。

重要

スタンドアロン Pod にラベルがない場合には、生成されたポリシーは Pod の全体的な namespace からのトラフィックを許可します。

8.2.2. ネットワークグラフでのネットワークポリシーの生成

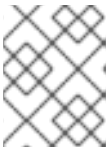
RHACS を使用すると、環境内で実際に監視されたネットワーク通信フローに基づいてネットワークポリシーを自動的に生成できます。

ネットワークグラフで選択したクラスター、namespace、デプロイメントに基づいてポリシーを生成できます。ポリシーは、現在の Network Graph スコープに含まれるすべてのデプロイメントに対して生成されます。たとえば、現在のスコープには、クラスター全体、クラスターと namespace、選択した namespace にある個別に選別したデプロイメントが含まれます。また、クラスター、namespace、およびデプロイメントの選択を任意に組み合わせて、**Filter deployments** フィールドのフィルターの 1 つを適用して、スコープをさらに縮小することもできます。たとえば、特定の CVE の影響を受ける特定のクラスターおよび namespace 内のデプロイメントに範囲を絞り込むことができます。ポリシーは、ベースライン検出期間中に確認されたトラフィックから生成されます。

1. RHACS ポータルで、**Network Graph** に移動します。
2. クラスターを選択し、1 つ以上の namespace を選択します。
3. オプション: 個別のデプロイメントを選択して、生成されるポリシーを対象のデプロイメントのみに制限します。**フィルターデプロイメント** 機能を使用して、スコープをさらに絞り込むこともできます。
4. ネットワークグラフのヘッダーで、**Network policy generator** を選択します。

5. オプション: 開いた情報パネルで、**Exclude ports & protocols**を選択して、ベースラインからネットワークポリシーを生成するときにポート/プロトコルの制限を削除します。
例として、**nginx3** デプロイメントは **nginx4** へのポート 80 接続を作成し、これは **nginx4** のベースラインの一部として含まれています。ポリシーが生成され、このチェックボックスが選択されていない場合 (デフォルトの動作)、生成されたポリシーは、**nginx3** から **nginx4** への接続で許可するポートを 80 のみに制限します。このオプションを選択してポリシーが生成された場合、生成されたポリシーは **nginx3** から **nginx4** までの接続内の全ポートを許可します。

6. **Generate and simulate network policies**をクリックします。RHACS は、選択したスコープのポリシーを生成します。このスコープは、**Generate network policies** パネルの上部に表示されます。



注記

スコープのデプロイメント情報をクリックすると、含まれるデプロイメントのリストが表示されます。

7. オプション: 生成されたネットワークポリシー設定 YAML ファイルをクリップボードにコピーするか、パネルのダウンロードアイコンをクリックしてダウンロードします。

8. オプション: 生成されたネットワークポリシーを既存のネットワークポリシーと比較するには、**Compare** をクリックします。既存のネットワークポリシーと生成されたネットワークポリシーの YAML ファイルは、サイドバイサイドビューで表示されます。

注記

既存の ingress ポリシーが含まれる namespace や、**stackrox** や **acs** などの特定の保護された namespace でのデプロイメントなど、一部の項目には生成されたポリシーがありません。

9. オプション: **Actions** メニューをクリックして、次のアクティビティーを実行します。

   - YAML ファイルを通知機能と共有する: YAML ファイルを、設定したシステム通知機能の 1 つ (Slack、ServiceNow、汎用 Webhook を使用するアプリケーションなど) に送信します。これらの通知機能は、**Platform Configuration → Integrations** に移動して設定します。詳細は、「関連情報」セクションのドキュメントを参照してください。

   - アクティブなトラフィックからルールを再構築する: 表示される生成されたポリシーを更新します。

   - ルールを以前に適用した YAML に戻す: シミュレートされたポリシーを削除し、最後のネットワークポリシーに戻します。

8.2.3. 生成されたポリシーのネットワークグラフでの保存

生成されたネットワークポリシーを RHACS からダウンロードして保存できます。このオプションを使用してポリシーをダウンロードし、Git などのバージョン管理システムにポリシーをコミットできるようにします。

手順

- ネットワークポリシーを生成した後、**Network Policy Simulator** パネルで **Download YAML** アイコンをクリックします。

8.2.4. 生成されたポリシーのネットワークグラフでのテスト

RHACS が生成するネットワークポリシーをダウンロードした後、CLI または自動デプロイメント手順を使用してクラスターにポリシーを適用してテストできます。生成されたネットワークポリシーをネットワークグラフに直接適用することはできません。

手順

1. 保存した YAML ファイルを使用してポリシーを作成するには、次のコマンドを実行します。

```
$ oc create -f "<generated_file>.yml" 1
```

1 Kubernetes を使用する場合は、**oc** の代わりに **kubectl** を入力します。

2. 生成されたポリシーで問題が発生する場合は、以下のコマンドを実行してそのポリシーを削除できます。

```
$ oc delete -f "<generated_file>.yml" 1
```

1 Kubernetes を使用する場合は、**oc** の代わりに **kubectl** を入力します。

警告

ネットワークポリシーを直接適用すると、アプリケーションの実行で問題が発生する可能性があります。実稼働環境のワークロードに適用する前に、常に開発環境またはテストクラスターでネットワークポリシーをダウンロードし、テストします。

8.2.5. ネットワークグラフで以前に適用されたポリシーに戻す

ポリシーを削除して、以前に適用したポリシーに戻すことができます。

手順

1. RHACS ポータルで、**Network Graph** に移動します。
2. 上部のバーのメニューからクラスター名を選択します。
3. 1 つ以上の namespace とデプロイメントを選択します。
4. **Simulate network policy** を選択します。
5. **View active YAMLS** を選択します。
6. **Actions** メニューから、**Revert rules to previously applied YAML** を選択します。

警告

ネットワークポリシーを直接適用すると、アプリケーションの実行で問題が発生する可能性があります。実稼働環境のワークロードに適用する前に、常に開発環境またはテストクラスターでネットワークポリシーをダウンロードし、テストします。

8.2.6. ネットワークグラフで自動生成されたすべてのポリシーの削除

RHACS を使用して作成したクラスターから、自動生成されたポリシーをすべて削除できます。

手順

- 以下のコマンドを実行します。

```
$ oc get ns -o jsonpath='{.items[*].metadata.name}' | \
xargs -n 1 oc delete networkpolicies -l \
'network-policy-generator.stackrox.io/generated=true' -n 1
```

1 Kubernetes を使用する場合は、**oc** の代わりに **kubectl** を入力します。

8.2.7. ネットワークグラフからのネットワークポリシーのシミュレーション

現在のネットワークポリシーでは、不要なネットワーク通信が許可される可能性があります。ネットワークポリシージェネレーターを使用して、一連のデプロイメントに対して計算されたベースラインへの ingress トラフィックを制限するネットワークポリシーを作成できます。

注記

ネットワークグラフには、生成されたポリシーが視覚化されて表示されません。生成されるポリシーは ingress トラフィックのみを対象とし、egress トラフィックを制限するポリシーは生成されません。

手順

1. RHACS ポータルで、**Network Graph** に移動します。
2. クラスターを選択し、1 つ以上の namespace を選択します。
3. ネットワークグラフのヘッダーで、**Network policy generator** を選択します。
4. オプション: シミュレーションで使用するネットワークポリシーを含む YAML ファイルを生成するには、**Generate and simulate network policies** をクリックします。詳細は、「ネットワークグラフでのネットワークポリシーの生成」を参照してください。
5. シミュレーションで使用するネットワークポリシーの YAML ファイルをアップロードします。ネットワークグラフビューには、提案されたネットワークポリシーが何を達成するかが表示されます。以下の手順を実行します。

a. **Upload YAML** をクリックし、ファイルを選択します。

b. **Open** をクリックします。システムは、アップロードされたポリシーの処理ステータスを示すメッセージを表示します。

6. 現在のネットワークポリシーに対応するアクティブな YAML ファイルを表示するには、**View active YAMLS** タブをクリックし、ドロップダウンリストからポリシーを選択します。次のアクションを実行することもできます。

- 適切なボタンをクリックして、表示された YAML ファイルをコピーまたはダウンロードします。
- **Actions** メニューを使用して、アクティブなトラフィックからルールを再構築するか、以前に適用された YAML にルールを戻します。詳細は、「ネットワークグラフでのネットワークポリシーの生成」を参照してください。

関連情報

- オフラインモードでのカーネルサポートパッケージの更新
- Generic Webhook を使用した統合

8.3. ネットワークグラフでのネットワークベース化について

RHACS では、ネットワークベースライニングを使用することでリスクを最小限に抑えることができます。インフラストラクチャーをセキュアに保つためのプロアクティブなアプローチです。RHACS は、まず既存のネットワークフローを検出してベースラインを作成し、次にこのベースラインの外にあるネットワークフローを異常として扱います。

RHACS をインストールする場合、デフォルトのネットワークベースラインはありません。RHACS はネットワークフローを検出すると、次のガイドラインに従ってベースラインを作成し、検出されたすべてのネットワークフローをそれに追加します。

- RHACS は、新しいネットワークアクティビティーを検出すると、そのネットワークフローをネットワークベースラインに追加します。
- ネットワークフローは、異常なフローとして表示されず、違反は発生しません。

検出フェーズの後、次のアクションが発生します。

- RHACS は、ネットワークベースラインへのネットワークフローの追加を停止します。
- ネットワークベースラインにない新しいネットワークフローは異常なフローとして表示されますが、違反はトリガーされません。

8.3.1. ネットワークグラフからのネットワークベースライン表示

ネットワークグラフビューからネットワークベースラインを表示できます。

手順

1. **Namespaces** リストをクリックし、検索フィールドを使用して namespace を見つけるか、個々の namespace を選択します。
2. **Deployments** リストをクリックし、検索フィールドを使用してデプロイメントを見つけるか、ネットワークグラフに表示する個々のデプロイメントを選択します。

3. ネットワークグラフでデプロイメントをクリックして情報パネルを表示します。
4. **Baseline** タブを選択します。**filter by entity name**フィールドを使用して、表示されるフローをさらに制限します。
5. オプション: 次のいずれかのアクションを実行して、ベースラインフローを異常としてマークできます。

   - 個々のエンティティーを選択します。オーバーフローメニュー  をクリックし、**Mark as anomalous** を選択します。
   - 複数のエンティティーを選択し、**Bulk actions** をクリックして、**Mark as anomalous** を選択します。

6. オプション: ポートとプロトコルを除外するには、ボックスをオンにします。
7. オプション: ベースラインをネットワークポリシー YAML ファイルとして保存するには、**Download baseline as network policy**をクリックします。

8.3.2. ネットワークグラフからのネットワークベースラインのダウンロード

ネットワークグラフビューからネットワークベースラインを YAML ファイルとしてダウンロードできます。

手順

1. RHACS ポータルで、**Network Graph** に移動します。
2. **Namespaces** リストをクリックし、検索フィールドを使用して namespace を見つけるか、個々の namespace を選択します。
3. **Deployments** リストをクリックし、検索フィールドを使用してデプロイメントを見つけるか、ネットワークグラフに表示する個々のデプロイメントを選択します。
4. ネットワークグラフでデプロイメントをクリックして情報パネルを表示します。
5. **Baseline** タブには、ベースラインフローがリストされます。**filter by entity name**フィールドを使用して、フローのリストをさらに制限します。
6. オプション: ポートとプロトコルを除外するには、ボックスをオンにします。
7. **Download baseline as network policy**をクリックします。

8.3.3. ネットワークベースライン時間枠の設定

ROX_NETWORK_BASELINE_OBSERVATION_PERIOD および **ROX_BASELINE_GENERATION_DURATION** 環境変数を使用して、観測期間とネットワークベースラインの生成期間を設定できます。

手順

1. 次のコマンドを実行して、**ROX_NETWORK_BASELINE_OBSERVATION_PERIOD** 環境変数を設定します。

```
$ oc -n stackrox set env deploy/central \ 1
  ROX_NETWORK_BASELINE_OBSERVATION_PERIOD=<value> 2
```

1. Kubernetes を使用する場合は、**oc** の代わりに **kubectl** を入力します。
2. 値は時間単位である必要があります (例: **300ms**、**-1.5h**、または **2h45m**)。有効な時間単位は、**ns**、**us** または **µs**、**ms**、**s**、**m**、**h** です。

2. 次のコマンドを実行して、**ROX_BASELINE_GENERATION_DURATION** 環境変数を設定します。

```
$ oc -n stackrox set env deploy/central \ 1
  ROX_BASELINE_GENERATION_DURATION=<value> 2
```

1. Kubernetes を使用する場合は、**oc** の代わりに **kubectl** を入力します。
2. 値は時間単位である必要があります (例: **300ms**、**-1.5h**、または **2h45m**)。有効な時間単位は、**ns**、**us** または **µs**、**ms**、**s**、**m**、**h** です。

8.3.4. ネットワークグラフでのベースライン違反に関するアラートの有効化

異常なネットワークフローを検出し、ベースラインにないトラフィックの違反をトリガーするように RHACS を設定できます。これは、ネットワークポリシーでトラフィックをブロックする前に、ネットワークに不要なトラフィックが含まれているかどうかを判断するのに役立ちます。

手順

1. **Namespaces** リストをクリックし、検索フィールドを使用して namespace を見つけるか、個々の namespace を選択します。
2. **Deployments** リストをクリックし、検索フィールドを使用してデプロイメントを見つけるか、ネットワークグラフに表示する個々のデプロイメントを選択します。
3. ネットワークグラフでデプロイメントをクリックして情報パネルを表示します。
4. **Baseline** タブでは、ベースラインフローを表示できます。**filter by entity name**フィールドを使用して、表示されるフローをさらに制限します。
5. **Alert on baseline violations** オプションを切り替えます。
   - **Alert on baseline violations** オプションを切り替えると、異常なネットワークフローによって違反がトリガーされます。
   - **Alert on baseline violations** オプションを再度切り替えると、異常なネットワークフローの違反の受信を停止できます。

第9章 ビルド時のネットワークポリシーツール

ビルド時のネットワークポリシーツールを使用すると、**roxctl** CLI を使用した開発および運用ワークフローでの Kubernetes ネットワークポリシーの作成と検証を自動化できます。これらのツールは、プロジェクトのワークロードおよびネットワークポリシーマニフェストなど、指定されたファイルディレクトリーで動作し、RHACS 認証を必要としません。

表9.1 ネットワークポリシーツール

コマンド	説明
roxctl netpol generate	指定されたディレクトリー内のプロジェクトの YAML マニフェストを分析することにより、Kubernetes ネットワークポリシーを生成します。詳細は、ビルド時のネットワークポリシージェネレーターの使用 を参照してください。
roxctl netpol connectivity map	ワークロードと Kubernetes ネットワークポリシーマニフェストを調べて、プロジェクトディレクトリー内のワークロード間で許可されている接続をリスト表示します。出力は、さまざまなテキスト形式またはグラフィカルな **.dot** 形式で生成できます。詳細は、roxctl netpol connectivity map コマンドを使用した接続マッピング を参照してください。
roxctl netpol connectivity diff	2 つのプロジェクトバージョン間で許可される接続にバリエーションのリストを作成します。これは、ワークロードと各バージョンのディレクトリー内の Kubernetes ネットワークポリシーマニフェストによって決まります。この機能は、ソースコード (構文) に対して **diff** を実行した場合には特定できない、意味上の違いを示します。詳細は、プロジェクトバージョン間での許可される接続の違いの確認 を参照してください。

9.1. ビルド時のネットワークポリシージェネレーターの使用

ビルド時のネットワークポリシージェネレーターは、アプリケーション YAML マニフェストに基づいて Kubernetes ネットワークポリシーを自動的に生成できます。これを使用して、クラスターにアプリケーションをデプロイする前に、継続的インテグレーション/継続的デプロイメント (CI/CD) パイプラインの一部としてネットワークポリシーを開発できます。

Red Hat は、NP-Guard プロジェクト の開発者と協力してこの機能を開発しました。まず、ビルド時のネットワークポリシージェネレーターは、ローカルフォルダー内の Kubernetes マニフェストを分析します。これには、サービスマニフェスト、config map、および **Pod**、**Deployment**、**ReplicaSet**、**Job**、**DaemonSet**、**StatefulSet** などのワークロードマニフェストが含まれます。次に、必要な接続を検出し、Pod の分離を実現するための Kubernetes ネットワークポリシーを作成します。これらのポリシーでは、必要なイングレスおよびエグレストラフィックをそれ以上も以下も許可しません。

9.1.1. ビルド時のネットワークポリシーの生成

ビルド時のネットワークポリシージェネレーターは、**roxctl** CLI に含まれています。ビルド時のネットワークポリシー生成機能の場合、**roxctl** CLI は RHACS Central と通信する必要がないため、任意の開発環境で使用できます。

前提条件

1. ビルド時のネットワークポリシージェネレーターは、コマンドの実行時に指定したディレクトリーを再帰的にスキャンします。したがって、コマンドを実行する前に、サービスマニフェスト、config map、ワークロードマニフェスト (**Pod**、**Deployment**、**ReplicaSet**、**Job**、**DaemonSet**、**StatefulSet** など) が、指定されたディレクトリーに YAML ファイルとしてすでに存在している必要があります。

2. **kubectl apply -f** コマンドを使用して、これらの YAML ファイルをそのまま適用できることを確認します。ビルド時のネットワークポリシージェネレーターは、Helm スタイルのテンプレートを使用するファイルでは機能しません。

3. サービスネットワークアドレスがハードコーディングされていないことを確認します。サービスに接続する必要があるすべてのワークロードは、サービスネットワークアドレスを変数として指定する必要があります。この変数は、ワークロードのリソース環境変数を使用するか、config map で指定できます。

   - 例 1: 環境変数の使用
   - 例 2: config map の使用
   - 例 3: config map の使用

4. サービスネットワークアドレスは、次の公式の正規表現パターンに一致する必要があります。

   ```
   (http(s)?://)?<svc>(.<ns>(.svc.cluster.local)?)?(:<portNum>)? (1)
   ```

   (1) このパターンでは、

   - <svc> はサービス名
   - <ns> はサービスを定義した namespace
   - <portNum> は公開されたサービスのポート番号

   以下は、パターンに一致するいくつかの例です。

   - **wordpress-mysql:3306**
   - **redis-follower.redis.svc.cluster.local:6379**
   - **redis-leader.redis**
   - **http://rating-service.**

手順

1. help コマンドを実行して、ビルド時のネットワークポリシー生成機能が使用可能であることを確認します。

   ```
   $ roxctl netpol generate -h
   ```

2. **netpol generate** コマンドを使用してポリシーを生成します。

```
$ roxctl netpol generate <folder_path> [flags] 1
```

1 フォルダーへのパスを指定します。フォルダーには、分析用の YAML リソースを含むサブフォルダーを含めることができます。このコマンドは、サブフォルダーツリー全体をスキャンします。オプションで、パラメーターを指定してコマンドの動作を変更することもできます。

オプションのパラメーターの詳細は、roxctl netpol generate コマンドオプション を参照してください。

次のステップ

- ポリシーを生成した後、関連するネットワークアドレスが YAML ファイルで期待どおりに指定されていない場合に備えて、ポリシーの完全性と正確性を検査する必要があります。
- 最も重要なことは、必要な接続が分離ポリシーによってブロックされていないことを確認することです。この検査には、**roxctl netpol 接続マップ** ツールを使用できます。

注記

自動化を使用してワークロードデプロイメントの一部としてネットワークポリシーをクラスターに適用すると、時間を短縮し、正確さを確保できます。プルリクエストを使用して生成されたポリシーを送信すると、GitOps アプローチに使用でき、ポリシーをパイプラインの一部としてデプロイする前にチームはポリシーを確認する機会を得ることができます。

9.1.2. roxctl netpol generate コマンドオプション

roxctl netpol generate コマンドは、次のオプションをサポートしています。

オプション	説明
-h、--help	**netpol** コマンドのヘルプテキストを表示します。
-d, --output-dir <dir>	生成されたポリシーをターゲットフォルダーに保存します。ポリシーごとに 1 つのファイルです。
-f, --output-file <filename>	生成されたポリシーを保存して単一の YAML ファイルにマージします。
--fail	最初に発生したエラーで失敗します。デフォルト値は **false** です。
--remove	出力パスがすでに存在する場合は削除します。
--strict	警告をエラーとして扱います。デフォルト値は **false** です。

9.2. ROXCTL NETPOL CONNECTIVITY MAP コマンドを使用した接続マッピング

接続マッピングレポートを使用すると、Kubernetes マニフェストで定義されたネットワークポリシーに基づいた、さまざまなワークロード間で許可される接続に関する情報を取得できます。設定したネットワークポリシーに従って、Kubernetes 環境内のさまざまなワークロードがどのように相互通信を許可されるかを視覚化して理解できます。

接続マッピング情報を取得するには、**roxctl netpol connectivity map** コマンドに、Kubernetes ワークロードとネットワークポリシーマニフェストを含むディレクトリーパスを指定する必要があります。この出力では、分析された Kubernetes リソース内の接続の詳細が示されます。

9.2.1. Kubernetes マニフェストディレクトリーからの接続マッピング情報の取得

手順

- 次のコマンドを実行して、接続マッピング情報を取得します。

```
$ roxctl netpol connectivity map <folder_path> [flags) 1
```

1 フォルダーへのパスを指定します。たとえば、分析用の YAML リソースとネットワークポリシーを含むサブフォルダーなども指定できます (例: **netpol-analysis-example-minimal/**)。このコマンドは、サブフォルダーツリー全体をスキャンします。オプションで、パラメーターを指定してコマンドの動作を変更することもできます。

オプションのパラメーターの詳細は、roxctl netpol connectivity map コマンドオプション を参照してください。

例9.1 出力例

src	dst	conn
0.0.0.0-255.255.255.255	default/frontend[Deployment]	TCP 8080
default/frontend[Deployment]	0.0.0.0-255.255.255.255	UDP 53
default/frontend[Deployment]	default/backend[Deployment]	TCP 9090

出力には、許可された接続回線のリストを含む表が表示されます。各接続線は、送信元 (**src**)、宛先 (**dst**)、および許可された接続属性 (**conn**) の 3 つの部分で構成されます。

src を送信元エンドポイント、**dst** を宛先エンドポイント、**conn** を許容される接続属性として解釈できます。エンドポイントの形式は、**namespace/name[Kind]** (例: **default/backend[Deployment]**) です。

9.2.2. 接続マップの出力形式および視覚化

txt、**md**、**csv**、**json**、**dot** などのさまざまな出力形式を使用できます。**dot** 形式は、出力を接続グラフとして視覚化するのに最適です。これは、Graphviz ツール などのグラフ視覚化ソフトウェアや VSCode の拡張機能 を使用して表示できます。Graphviz がローカルでインストールされているか、オ

ンラインビューアーを介してインストールされているかに関係なく、Graphviz を使用して **dot** 出力を **svg**、**jpeg**、または **png** などの形式に変換できます。

9.2.3. Graphviz を使用した dot 出力からの SVG グラフの生成

以下の手順に従って、**dot** 出力から **svg** 形式のグラフを作成します。

前提条件

- Graphviz がローカルシステムにインストールされている。

手順

- 以下のコマンドを実行して **svg** 形式のグラフを作成します。

```
$ dot -Tsvg connlist_output.dot > connlist_output_graph.svg
```

以下は、ドットの出力と Graphviz によって生成された結果グラフの例です。

  - 例 1: dot output
  - 例 2: Graph generated by Graphviz

9.2.4. roxctl netpol connectivity map コマンドオプション

roxctl netpol connectivity map コマンドは、次のオプションをサポートしています。

オプション	説明
--fail	最初に発生したエラーで失敗します。デフォルト値は **false** です。
--focus-workload string	出力内の指定されたワークロード名の接続に注目します。
-h、--help	**roxctl netpol connectivity map** コマンドのヘルプテキストを表示します。
-f, --output-file string	出力された接続リストを特定のファイルに保存します。
-o, --output-format string	出力形式を設定します。サポートされる形式は **txt**、**json**、**md**、**dot**、および **csv** です。デフォルト値は **txt** です。
--remove	出力パスがすでに存在する場合は削除します。デフォルト値は **false** です。
--save-to-file	接続リストの出力をデフォルトのファイルに保存します。デフォルト値は **false** です。

オプション	説明
--strict	警告をエラーとして扱います。デフォルト値は **false** です。

9.3. プロジェクトバージョン間での許可される接続の違いの確認

このコマンドは、2 つのプロジェクトバージョン間で許可される接続の違いを理解するのに役立ちます。各バージョンのディレクトリーにあるワークロードと Kubernetes ネットワークポリシーマニフェストを分析し、相違点をテキスト形式で表現します。

接続差異レポートは、**text**、**md**、**dot**、**csv** などのさまざまな出力形式で表示できます。

9.3.1. roxctl netpol connectivity diff コマンドを使用した接続の差異レポートの生成

接続差異レポートを作成する場合、**roxctl netpol connectivity diff** コマンドを実行する際にネットワークポリシーなどの Kubernetes マニフェストがそれぞれに含まれている、2 つのフォルダー **dir1** と **dir2** が必要です。

手順

- 次のコマンドを実行して、指定したディレクトリー内の Kubernetes マニフェスト間の接続の違いを確認します。

```
$ roxctl netpol connectivity diff --dir1=<folder_path_1> --dir2=<folder_path_2> [flags] (1)
```

(1) フォルダーへのパスを指定します。分析用の YAML リソースとネットワークポリシーなど、サブフォルダーを含めることができます。このコマンドは、両方のディレクトリーのサブフォルダーツリー全体をスキャンします。たとえば、**<folder_path_1>** は **netpol-analysis-example-minimal/**、**<folder_path_2>** は **netpol-diff-example-minimal/** に置き換えます。オプションで、パラメーターを指定してコマンドの動作を変更することもできます。

オプションのパラメーターの詳細は、roxctl netpol connectivity diff コマンドオプション を参照してください。

注記

このコマンドは、**kubectl apply -f** を使用して許容できるすべての YAML ファイルを考慮し、これらが **roxctl netpol connectivity diff** コマンドの有効な入力になります。

例9.2 出力例

diff-type	比較元	比較先	dir 1	dir 2	workloads-diff-info

diff-type	比較元	比較先	dir 1	dir 2	workloads-diff-info
changed	default/frontend[Deployment]	default/backend[Deployment]	TCP 9090	TCP 9090,UDP 53	
added	0.0.0.0-255.255.255.255	default/backend[Deployment]	接続なし	TCP 9090	

意味的な差異レポートは、**dir1** で許可されている接続と比較して、**dir2** で変更、追加、または削除された接続の概要を示します。出力を確認すると、各行に使用可能な接続が表示されています。それぞれの接続は、**dir1** と比較すると、**dir2** で追加、削除または変更されています。

以下は、**roxctl netpol connectivity diff** コマンドによってさまざまな形式で生成される出力例です。

- 例 1: text フォーマット
- 例 2: md フォーマット
- 例 3: ドットフォーマットから生成された svg グラフ
- 例 4: csv フォーマット

該当する場合、**workloads-diff-info** は、追加または削除された接続の追加または削除されたワークロードに関する情報を追加で提供します。

たとえば、ワークロード **B** が削除されたためにワークロード **A** からワークロード **B** への接続が削除された場合、**workloads-diff-info** はワークロード **B** が削除されたことを示します。ただし、ネットワークポリシーの変更のみが原因でそのような接続が削除され、ワークロード **A** も **B** も削除されなかった場合、**workloads-diff-info** は空になります。

9.3.2. roxctl netpol connectivity diff コマンドオプション

roxctl netpol connectivity diff コマンドは、次のオプションをサポートしています。

オプション	説明
--dir1 string	入力リソースの最初のディレクトリーパス。これは必須のオプションです。
--dir2 string	最初のディレクトリーパスと比較される入力リソースの 2 番目のディレクトリーパス。これは必須のオプションです。
--fail	最初に発生したエラーで失敗します。デフォルト値は **false** です。

オプション	説明
-h、--help	**roxctl netpol connectivity diff** コマンドのヘルプテキストを表示します。
-f, --output-file string	接続の差分出力を特定のファイルに保存します。
-o, --output-format string	出力形式を設定します。サポートされている形式は、**txt**、**md**、**dot**、および **csv** です。デフォルト値は **txt** です。
--remove	出力パスがすでに存在する場合は削除します。デフォルト値は **false** です。
--save-to-file	接続の違いの出力をデフォルトのファイルに保存します。デフォルト値は **false** です。
--strict	警告をエラーとして扱います。デフォルト値は **false** です。

9.3.3. 構文上の差異出力と意味上の差異出力の区別

次の例では、**dir1** は **netpol-analysis-example-minimal/**、**dir2** は **netpol-diff-example-minimal/** に置き換えます。ネットワークポリシー **backend-netpol** で加えられた小さな変更が、ディレクトリー間の差異となっています。

dir1 のポリシーの例:

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  creationTimestamp: null
  name: backend-netpol
spec:
  ingress:
  - from:
    - podSelector:
        matchLabels:
          app: frontend
    ports:
    - port: 9090
      protocol: TCP
  podSelector:
    matchLabels:
      app: backendservice
  policyTypes:
  - Ingress
  - Egress
status: {}
```

dir2 の変更は、ports 属性の前に **-** が追加されており、これにより差分出力が生成されます。

9.3.3.1. 構文的な違いの出力

手順

- 次のコマンドを実行して、指定した 2 つのディレクトリーにある **netpols.yaml** ファイルの内容を比較します。

```
$ diff netpol-diff-example-minimal/netpols.yaml netpol-analysis-example-minimal/netpols.yaml
```

出力例

```
12c12
<   - ports:
---
>     ports:
```

9.3.3.2. 意味上の差異の出力

手順

- 次のコマンドを実行して、指定した 2 つのディレクトリー内の Kubernetes マニフェストとネットワークポリシー間の接続の違いを分析します。

```
$ roxctl netpol connectivity diff --dir1=roxctl/netpol/connectivity/diff/testdata/netpol-analysis-example-minimal/ --dir2=roxctl/netpol/connectivity/diff/testdata/netpol-diff-example-minimal
```

出力例

```
Connectivity diff:
diff-type: changed, source: default/frontend[Deployment], destination: default/backend[Deployment], dir1:  TCP 9090, dir2: TCP 9090,UDP 53
diff-type: added, source: 0.0.0.0-255.255.255.255, destination: default/backend[Deployment], dir1:  No Connections, dir2: TCP 9090
```

第10章 リスニングエンドポイントの監査

Red Hat Advanced Cluster Security for Kubernetes (RHACS) は、セキュアクラスター内のポートでリッスンしているプロセスを監査し、このデータをデプロイメント、namespace、クラスターごとにフィルターする機能を備えています。

以下の方法を使用して、リッスンしているプロセスおよびポートに関する情報を表示できます。

- RHACS Web ポータルで、**Network → Listening Endpoints** に移動します。
- API の **ListeningEndpointsService** オブジェクトに接続します。API の詳細は、RHACS Web ポータルの **Help → API reference** に移動してください。

このページには、デプロイメント別のプロセスのリストが表示され、リスト上の各プロセスについて次の情報が表示されます。

- デプロイメント名
- クラスター
- Namespace
- 数、またはデプロイメント内のポートでリッスンしているプロセスの数

フィルターフィールドを使用し、個々のデプロイメント、namespace、およびクラスターを入力することで、ページに表示される情報をさらにフィルターできます。

リストの上部にあるデプロイメントアイコンをクリックして、リストされているすべてのデプロイメントの全セクションをデプロイメントするか、単一のデプロイメント行のデプロイメントアイコンをクリックして、そのデプロイメントに関する追加情報を表示します。以下の情報が含まれています。

- Exec file path: プロセスの場所
- PID: プロセスのシステム ID
- Port: プロセスがリッスンしているポート
- Protocol: プロセスが使用しているプロトコル
- Pod ID: プロセスが含まれる Pod の名前
- コンテナー名: リスニング中のプロセスが配置されているコンテナーの名前

デプロイメント名をクリックすると、RHACS Web ポータルの **リスク** ページが表示され、ポリシー違反や追加のデプロイメント詳細などのリスク指標を含む、デプロイメントに関する情報を表示できます。

第11章 クラスター設定の確認

Configuration Management ビューを使用し、クラスター内のさまざまなエンティティー間の相関を理解してクラスター設定を効率的に管理する方法を説明します。

すべての OpenShift Container Platform クラスターには、クラスター全体に分散された異なるエンティティーが多数含まれているため、利用可能な情報を理解して操作することがより困難になります。

Red Hat Advanced Cluster Security for Kubernetes (RHACS) は、1つのページでこれらの分散エンティティーをすべて組み合わせて、効率的な設定管理が実現できます。すべてのクラスター、namespace、ノード、デプロイメント、イメージ、シークレット、ユーザー、グループ、サービスアカウント、およびロールに関する情報を1つの **Configuration Management** ビューにまとめ、さまざまなエンティティーとそれらの間の接続を視覚化するのに役立ちます。

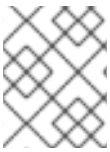
11.1. CONFIGURATION MANAGEMENT ビューの使用

Configuration Management ビューを開くには、ナビゲーションメニューから **Configuration Management** を選択します。**Dashboard** と同様に、便利なウィジェットが表示されます。

これらのウィジェットは対話形式で、以下の情報が表示されます。

- 重大度別のセキュリティーポリシー違反
- CIS (Center for Information Security) Docker および Kubernetes ベンチマーク制御の状態
- ほとんどのクラスターで管理者権限を持つユーザー
- クラスターで最も広く使用されているシークレット

Configuration Management ビューのヘッダーには、クラスター内のポリシーおよび CIS コントロールの数が表示されます。



注記

ポリシー数とポリシーリストビューには、デプロイメントライフサイクルフェーズのポリシーのみが含まれます。

ヘッダーには、エンティティー間の切り替えを可能にするドロップダウンメニューが含まれます。たとえば、以下を行うことができます。

- **Policies** をクリックしてすべてのポリシーと重大度を表示するか、**CIS Controls** を選択してすべてのコントロールに関する詳細情報を表示します。
- **Application and Infrastructure** をクリックし、クラスター、namespace、ノード、デプロイメント、イメージ、およびシークレットを選択して詳細情報を表示します。
- **RBAC Visibility and Configuration** をクリックし、ユーザーおよびグループ、サービスアカウント、およびロールを選択して詳細情報を表示します。

11.2. KUBERNETES ロールの設定ミスの特定

設定管理 ビューを使用して、**cluster-admin** ロールが付与されているユーザー、グループ、サービスアカウント、または誰にも付与されていないロールなど、潜在的な設定ミスを特定できます。

11.2.1. Kubernetes のロールとその割り当てを見つける

特定のユーザーおよびグループに割り当てられている Kubernetes ロールに関する情報を取得するには、**Configuration Management** ビューを使用します。

手順

1. RHACS ポータルで、**Configuration Management** をクリックします。
2. **Configuration Management** ビューのヘッダーから **Role-Based Access Control → Users and Groups** を選択します。**ユーザーとグループ** ビューには、Kubernetes のユーザーとグループのリスト、割り当てられたロール、およびそれぞれに対して **cluster-admin** ロールが有効になっているかどうかが表示されます。
3. ユーザーまたはグループを選択して、関連付けられたクラスターおよび namespace パーミッションの詳細を表示します。

11.2.2. サービスアカウントおよびそのパーミッションの検索

Configuration Management ビューを使用して、サービスアカウントが使用されている場所とそのパーミッションを確認します。

手順

1. RHACS ポータルで、**Configuration Management** に移動します。
2. **Configuration Management** ビューのヘッダーから **RBAC Visibility and Configuration → Service Accounts** を選択します。**サービスアカウント** ビューには、クラスター全体の Kubernetes サービスアカウントのリスト、割り当てられたロール、**cluster-admin** ロールが有効になっているかどうか、およびそれらを使用するデプロイが表示されます。
3. 行または下線付きのリンクを選択して、選択したサービスアカウントに付与されているクラスターと namespace のパーミッションなどの詳細を表示します。

11.2.3. 未使用の Kubernetes ロールの検索

Configuration Management ビューを使用して、Kubernetes ロールの詳細情報を取得し、未使用のロールを検索します。

手順

1. RHACS ポータルで、**Configuration Management** に移動します。
2. **Configuration Management** ビューのヘッダーから**RBAC Visibility and Configuration → Roles** を選択します。**Roles** ビューには、クラスター全体の Kubernetes ロールのリスト、付与するパーミッション、およびそれらが使用される場所が表示されます。
3. ロールに関する詳細を表示するには、行またはインラインのリンクを選択します。
4. ユーザー、グループ、またはサービスアカウントに付与されていないロールを検索するには、**Users & Groups**列ヘッダーを選択します。次に、**Shift** キーを押しながら **サービスアカウント** 列ヘッダーを選択します。このリストには、ユーザー、グループ、またはサービスアカウントに付与されていないロールが表示されます。

11.3. KUBERNETES シークレットの表示

環境で使用する Kubernetes Secret を表示し、それらのシークレットを使用してデプロイメントを特定します。

手順

1. RHACS ポータルで、**Configuration Management** に移動します。
2. **Secrets Most Used Across Deployments**ウィジェットで **View All** を選択します。**Secrets** ビューには、Kubernetes Secret のリストが表示されます。
3. 詳細を表示する行を選択します。

使用できる情報を使用して、シークレットが不要なデプロイメントで使用されているかどうかを特定します。

11.4. ポリシー違反の検索

Configuration Management ビューの **Policy Violations by Severity**ウィジェットは、サンバーストチャートでポリシー違反を表示します。チャートの各レベルは、1 つのリングまたは円で表されます。

- 最も内側の円は、違反の総数を表します。
- 次のリングは、**低**、**中**、**高**、**重大** のポリシーカテゴリーを表します。
- 最も外側のリングは、特定のカテゴリーの個々のポリシーを表します。

Configuration Management ビューには、**ライフサイクルステージ** が **Deploy** に設定されているポリシーに関する情報のみが表示されます。これには、ランタイムの動作に対応するポリシーや、**Build** ステージの評価用に設定されたポリシーは含まれません。

手順

1. RHACS ポータルで、**Configuration Management** に移動します。
2. **Policy Violations by Severity**で、サンバーストチャートにマウスを移動して、ポリシー違反の詳細を表示します。
3. 優先度の高いポリシー違反に関する詳細情報を表示するには、**高と評価された n** を選択します.**n** は数値です。**Policies** ビューには、選択したカテゴリーでフィルタリングされたポリシー違反の一覧が表示されます。
4. ポリシーの説明、修復、違反によるデプロイメントなどの詳細情報を表示します。詳細はパネルに表示されます。
5. 情報パネルの **Policy Findings** セクションには、このような違反が発生したデプロイメントがリスト表示されます。
6. **Policy Findings** セクションでデプロイメントを選択し、Kubernetes ラベル、アノテーション、サービスアカウントなどの関連する詳細を表示します。

詳細情報を使用して、違反の修復を計画することができます。

11.5. 失敗した CIS コントロールの検索

Configuration Management ビューの **Policy Violations** サンバーストチャートと同様に、**CIS controls** ウィジェットは、障害のある Center for Information Security (CIS) 制御に関する情報を表示します。

チャートの各レベルは、1 つのリングまたは円で表されます。

- 最も内側の円は、失敗したコントロールの割合を表します。
- 次のリングは、制御カテゴリーを表します。
- 外部リングは、特定のカテゴリー内の個々のコントロールを表します。

手順

1. **CIS controls** ウィジェットのヘッダーから **CIS Docker v1.2.0** を選択します。これを使用して、CIS Docker コントロールと Kubernetes コントロールを切り替えます。
2. サンバーストチャートにカーソルを合わせ、失敗した制御の詳細を表示します。
3. **n controls failing** を選択します。**n** は数値で、失敗した制御に関する詳細情報を表示します。**Controls** ビューには、コンプライアンス状態に基づいてフィルターされる失敗した制御のリストが表示されます。
4. コントロールに失敗した制御の説明やノードなど、詳細情報を表示する行を選択します。
5. 情報パネルの **Control Findings** セクションには、コントロールが失敗するノードがリスト表示されます。Kubernetes ラベル、アノテーション、その他のメタデータなど、詳細を表示する行を選択します。

詳細情報を使用して、ノード、業界標準、または障害のある制御にフォーカスできます。コンテナー化されたインフラストラクチャーのコンプライアンスステータスの評価、確認、およびレポートを実行することもできます。

第12章 イメージの脆弱性の調査

Red Hat Advanced Cluster Security for Kubernetes では、RHACS のスキャナーを使用してイメージの脆弱性を分析したり、サポートされている別のスキャナーを使用するように 統合を設定 したりできます。

RHACS のスキャナーは、各イメージレイヤーを分析してパッケージを検索し、さまざまなソースから入力された脆弱性データベースと比較することで既知の脆弱性と照合します。ソースには、使用するスキャナーに応じて、National Vulnerability Database (NVD)、Open Source Vulnerabilities (OSV) データベース、オペレーティングシステムの脆弱性フィードなどが含まれます。

注記

RHACS Scanner V4 は、こちらのライセンス に基づいて OSV.dev で入手可能な OSV データベースを使用します。

RHACS には、StackRox Scanner と Scanner V4 の 2 つのスキャナーが含まれています。

StackRox Scanner は、Clair v2 オープンソーススキャナーのフォークから生まれたものであり、デフォルトのスキャナーです。RHACS バージョン 4.4 では、ClairCore 上に構築された、追加のイメージスキャン機能を提供する Scanner V4 が導入されました。

注記

このドキュメントでは、StackRox Scanner と Scanner V4 の 2 つのスキャナーによって提供される複合スキャン機能を指すために、"RHACS スキャナー" または "Scanner" という用語を使用します。特定のスキャナーの機能に言及する場合、特定のスキャナーの名前を使用します。

RHACS スキャナーは脆弱性を検出すると、次のアクションを実行します。

- **Vulnerability Management** ビューに脆弱性を表示し、詳細な分析を行えるようにします。
- 脆弱性をリスクに応じてランク付けし、RHACS ポータルで強調表示して、リスク評価を行えるようにします。
- 有効な セキュリティーポリシー と照合します。

RHACS スキャナーは、イメージを検査し、イメージ内のファイルに基づいてインストールされているコンポーネントを特定します。最終的なイメージが変更されて次のファイルが削除された場合、インストールされているコンポーネントや脆弱性を特定できない可能性があります。

コンポーネント	ファイル

コンポーネント	ファイル
パッケージマネージャー	• **/etc/alpine-release** • **/etc/apt/sources.list** • **/etc/lsb-release** • **/etc/os-release** または **/usr/lib/os-release** • **/etc/oracle-release**、**/etc/centos-release**、**/etc/redhat-release**、または **/etc/system-release** • その他の同様のシステムファイル。
言語レベルの依存関係	• JavaScript の **package.json**。 • Python の場合は **dist-info** または **egg-info** です。 • Java Archive(JAR)for Java Archive(JAR) の **MANIFEST.MF**。
アプリケーションレベルの依存関係	• **dotnet/shared/Microsoft.AspNetCore.App/** • **dotnet/shared/Microsoft.NETCore.App/**

12.1. RHACS SCANNER V4 について

RHACS は独自のスキャナーを備えています。統合を設定して、別の脆弱性スキャナーとともに RHACS を使用することもできます。

バージョン 4.4 以降は、ClairCore 上に構築された Scanner V4 が、言語およびオペレーティングシステム固有のイメージコンポーネントのスキャンを提供します。RHACS バージョン 4.4 では、一部のスキャン機能を提供するために、今後のリリースでその機能が実装されるまで、StackRox Scanner も使用されます。

関連情報

- Operator を使用して OpenShift Container Platform 用の RHACS をインストールするための Scanner V4 の設定
- Helm を使用して OpenShift Container Platform 用の RHACS をインストールするための Scanner V4 の設定
- Helm を使用して RHACS for Kubernetes をインストールするための Scanner V4 の設定

12.2. イメージのスキャン

RHACS バージョン 4.4 は、StackRox Scanner と Scanner V4 の 2 つのスキャナーを提供します。どちらのスキャナーも、ネットワークに接続されたセキュアクラスター内のイメージを検査できます。Operator を使用してデプロイされた Red Hat OpenShift 環境の場合、またはスキャン委譲が使用されている場合、セキュアクラスターのスキャンがデフォルトで有効になります。詳細は、「イメージスキャン委譲へのアクセス」を参照してください。

StackRox Scanner を使用する場合、RHACS は次のアクションを実行します。

- Central がイメージスキャンの要求を StackRox Scanner に送信します。
- StackRox Scanner は、この要求を受信すると、関連するレジストリーからイメージレイヤーを取得し、イメージをチェックして、各レイヤーにインストールされているパッケージを特定します。次に、特定されたパッケージとプログラミング言語固有の依存関係を脆弱性リストと比較し、情報を Central に送り返します。
- StackRox Scanner は、次の領域の脆弱性を特定します。
    - ベースイメージのオペレーティングシステム
    - パッケージマネージャーによりインストールされるパッケージ
    - プログラミング言語固有の依存関係
    - プログラミングランタイムとフレームワーク

Scanner V4 を使用する場合、RHACS は次のアクションを実行します。

- Central が、特定のイメージをダウンロードしてインデックス作成 (分析) するように Scanner V4 Indexer に要求します。
- Scanner V4 Indexer は、レジストリーからイメージのメタデータを取得してイメージのレイヤーを確認し、以前にインデックス作成されていない各レイヤーをダウンロードします。
- Scanner V4 Indexer は、インデックス作成プロセスを支援するマッピングファイルを Central に要求します。Scanner V4 Indexer はインデックスレポートを作成します。
- Central は、特定のイメージを既知の脆弱性と照合するように Scanner V4 Matcher に要求します。このプロセスにより、最終的なスキャン結果、つまり脆弱性レポートが生成されます。Scanner V4 Matcher は、Central から最新の脆弱性を要求します。
- Scanner V4 Matcher は、イメージのインデックス作成の結果、つまりインデックスレポートを Scanner V4 Indexer に要求します。次に、レポートを使用して関連する脆弱性を特定します。この対話は、イメージのインデックスが Central クラスターで作成された場合にのみ発生します。この対話は、セキュアクラスターでインデックス作成されたイメージの脆弱性を Scanner V4 が照合する場合には発生しません。
- Indexer は、イメージレイヤーのダウンロードとインデックス作成が 1 回だけ行われるように、インデックス作成の結果に関連するデータを Scanner V4 DB に保存します。これにより、不必要なネットワークトラフィックやその他のリソースの使用が防止されます。
- セキュアクラスターのスキャンが有効になっている場合、Sensor は Scanner V4 にイメージのインデックス作成を要求します。Scanner V4 Indexer は、Central が同じ namespace に存在しない限り、インデックス作成プロセスを支援するマッピングファイルを Sensor に要求します。その場合は、代わりに Central と通信します。

12.2.1. Scanner の一般的な警告メッセージの理解と対処

Red Hat Advanced Cluster Security for Kubernetes (RHACS) でイメージをスキャンすると、**CVE DATA MAY BE INACCURATE** という警告メッセージが表示される場合があります。イメージ内のオペレーティングシステムまたはその他のパッケージに関する完全な情報を取得できない場合、Scanner はこのメッセージを表示します。

以下の表は、一般的な Scanner の警告メッセージを示しています。

表12.1 警告メッセージ

Message	説明
Unable to retrieve the OS CVE data, only Language CVE data is available	Scanner がイメージのベースオペレーティングシステムを正式にサポートしていないことを示します。したがって、オペレーティングシステムレベルのパッケージの CVE データを取得できません。
Stale OS CVE data	イメージのベースオペレーティングシステムのサポートが終了したことを示します。これは、脆弱性データが古くなっていることを意味します。たとえば、Debian 8 および 9 です。 イメージ内のコンポーネントを識別するために必要なファイルの詳細は、イメージの脆弱性の検査 を参照してください。
Failed to get the base OS information	Scanner がイメージをスキャンしたが、イメージに使用されたベースオペレーティングシステムを特定できなかったことを示します。
Failed to retrieve metadata from the registry	ネットワーク上でターゲットレジストリーに到達できないことを示します。原因は、ファイアウォールが **docker.io** をブロックしているか、認証の問題がアクセスを妨げている可能性があります。 根本原因を分析するには、プライベートレジストリーまたはリポジトリー用に特別なレジストリー統合を作成し、RHACS Central の Pod ログを取得します。これを行う方法については、イメージレジストリーとの統合 を参照してください。

Message	説明
Image out of scope for Red Hat Vulnerability Scanner Certification	Scanner がイメージをスキャンしたが、イメージが古く、Red Hat Scanner Certification の範囲内にないことを示します。詳細は、Partner Guide for Red Hat Vulnerability Scanner Certification を参照してください。  **重要** Red Hat コンテナーイメージ を使用している場合は、2020 年 6 月以降のベースイメージの使用を検討してください。

12.2.2. サポート対象のパッケージ形式

スキャナーは、以下のパッケージ形式を使用するイメージの脆弱性の有無を確認できます。

- apt
- apk
- dpkg
- rpm

12.2.3. サポート対象のプログラミング言語

Scanner は、次のプログラミング言語の依存関係の脆弱性をチェックできます。

- Go (Scanner V4 のみ)
  - バイナリー: バイナリーのビルドに使用された標準ライブラリーのバージョンが分析されます。バイナリーがモジュールサポート (go.mod) を使用してビルドされている場合、依存関係も分析されます。
- Java
  - JAR
  - WAR
  - EAR
- JavaScript
  - Node.js
  - npm package.json
- Python
  - egg および wheel 形式

- Ruby
  - gem

12.2.4. サポート対象のランタイムおよびフレームワーク

Red Hat Advanced Cluster Security for Kubernetes 3.0.50 (Scanner バージョン 2.5.0) 以降の StackRox Scanner は、次の開発者プラットフォームの脆弱性を特定します。

- .NET Core
- ASP.NET Core

これらは Scanner V4 ではサポートされていません。

12.2.5. サポート対象オペレーティングシステム

このセクションにリストされているサポート対象のプラットフォームは、Scanner で脆弱性が特定されるディストリビューションで、Red Hat Advanced Cluster Security for Kubernetes をインストールできるサポート対象のプラットフォームとは異なります。

Scanner は、以下の Linux ディストリビューションを含むイメージの脆弱性を特定します。使用される脆弱性データベースの詳細は、「RHACS アーキテクチャー」の「脆弱性ソース」を参照してください。

ディストリビューション	バージョン
Alpine Linux	**alpine:3.2**[1]、**alpine:3.3**、**alpine:3.4**、**alpine:3.5**、**alpine:3.6**、**alpine:3.7**、**alpine:3.8**、**alpine:3.9**、**alpine:3.10**、**alpine:3.11**、**alpine:3.12**、**alpine:3.13**、**alpine:3.14**、**alpine:3.15**、**alpine:3.16**、**alpine:3.17**、**alpine:3.18**、**alpine:3.19**、**alpine:3.20**、**alpine:edge**
Amazon Linux	**amzn:2018.03**, **amzn:2**, **amzn:2023**[2]
CentOS	**centos:6**[1], **centos:7**[1], **centos:8**[1]
Debian	**debian:10**、**debian:11**、**debian:12**、**debian:unstable**[1]、**distroless**
Oracle Linux	**ol:5**[2]、**ol:6**[2]、**ol:7**[2]、**ol:8**[2]、**ol:9**[2]
Photon OS	**photon:1.0**[2]、**photon:2.0**[2]、**photon:3.0**[2]
Red Hat Enterprise Linux (RHEL)	**rhel:6**[3], **rhel:7**[3], **rhel:8**[3], **rhel:9**[3]
SUSE	**sles:11**[2]、**sles:12**[2]、**sles:15**[2]、**opensuse-leap:15.0**[2]、**opensuse-leap:15.1**[2]

ディストリビューション	バージョン
Ubuntu	**ubuntu:14.04、ubuntu:16.04、ubuntu:18.04、ubuntu:20.04,ubuntu:22.04、ubuntu:23.10、ubuntu:24.04** 以下の脆弱性ソースはベンダーによって更新されていません。**ubuntu:12.04、ubuntu:12.10、ubuntu:13.04、ubuntu:14.10、ubuntu:15.04、ubuntu::15.10、ubuntu::16.10、ubuntu:17.04、ubuntu:17.10、ubuntu:18.10、ubuntu:19.04、ubuntu:19.10、ubuntu:20.10、ubuntu:21.04、ubuntu:21.10、ubuntu:22.10、ubuntu:23.04、debian:8**[1]、**debian:9**[1]、**debian:10**[1]

1. StackRox Scanner でのみサポートされます。
2. Scanner V4 でのみサポートされます。
3. 2020 年 6 月より古いイメージは、Scanner V4 ではサポートされていません。

注記

- Fedora は脆弱性データベースを管理していないため、Scanner は Fedora オペレーティングシステムをサポートしていません。ただし、Scanner は Fedora ベースのイメージで言語固有の脆弱性を検出します。

関連情報

- 脆弱性ソース
- イメージ脆弱性スキャナーとの統合

12.3. イメージスキャン委譲へのアクセス

場合によっては、セキュアクラスターからのみアクセスできる隔離されたコンテナーイメージレジストリーを使用することがあります。イメージスキャン委譲機能を使用すると、セキュアクラスター内の任意のレジストリーからイメージをスキャンできます。

12.3.1. イメージスキャン委譲を利用したイメージスキャンの強化

現在、デフォルトで、Central Services Scanner は、OpenShift Container Platform 統合レジストリーからのイメージを除き、セキュアクラスター内で確認されたイメージに対してインデックス作成 (コンポーネントの識別) と脆弱性照合 (脆弱性データによるコンポーネントの強化) の両方を実行します。

OpenShift Container Platform 統合レジストリーからのイメージの場合、セキュアクラスターにインストールされた Scanner-slim がインデックス付けを実行し、Central Services Scanner が脆弱性の照合を実行します。

イメージスキャン委譲機能は、スキャン機能を拡張するものであり、すべてのレジストリーのイメージにインデックスを作成し、脆弱性照合のためにイメージを Central に送信することを Scanner-slim に許

可します。この機能を使用するには、Scanner-slim がセキュアクラスターにインストールされていることを確認してください。Scanner-slim が存在しない場合、スキャン要求が Central に直接送信されます。

12.3.2. イメージスキャン委譲の設定

新しいレジストリー委譲設定で、イメージスキャンの委譲元のレジストリーを指定します。Sensor が監視するイメージの場合、この設定では、レジストリーなし、すべてのレジストリー、または特定のレジストリーからのスキャンを委譲できます。**roxctl** CLI、Jenkins プラグイン、または API を使用してスキャンの委譲を有効にするには、宛先クラスターとソースレジストリーも指定する必要があります。

前提条件

- イメージをスキャンするには、Scanner-slim をセキュアクラスターにインストールする必要があります。

注記

Scanner-slim の有効化は、OpenShift Container Platform および Kubernetes セキュアクラスターでサポートされています。

手順

1. RHACS ポータルで、**Platform Configuration → Clusters** に移動します。
2. **Clusters** ビューヘッダーで、**Manage delegated scanning** をクリックします。
3. **Delegated Image Scanning** ページで、以下の情報を提供します。
   - **Delegate scanning for**: 次のオプションのいずれかを選択して、イメージ委譲の範囲を選択します。
     - None: デフォルトオプション。このオプションは、OpenShift Container Platform 統合レジストリーからのイメージを除き、セキュアクラスターによってイメージがスキャンされないことを指定します。
     - All registries: このオプションは、すべてのイメージが保護されたクラスターによってスキャンされることを示します。
     - Specified registries: このオプションは、レジストリーリストに基づいて、保護されたクラスターによってスキャンされるイメージを指定します。
   - **Select default cluster to delegate to** ドロップダウンリストから、コマンドラインインターフェイス (CLI) および API からのスキャンリクエストを処理するデフォルトクラスターの名前を選択します。これはオプションであり、必要に応じて **None** を選択できます。
   - オプション: **Add registry** をクリックし、ソースレジストリーと宛先クラスターの詳細を指定します。スキャンリクエストが CLI および API から送信されていない場合は、宛先クラスターを **None** として選択できます。必要に応じて、複数のソースレジストリーおよび宛先クラスターを追加できます。
4. **Save** をクリックします。

イメージ統合は Central と Sensor の間で同期されるようになり、Sensor は各 namespace からプルシークレットをキャプチャーします。次に、Sensor はこれらの認証情報を使用してイメージレジストリーに対して認証します。

12.3.3. セキュアクラスターへの Scanner-slim のインストールと設定

12.3.3.1. Operator の使用

RHACS Operator は、OpenShift Container Platform 統合レジストリーと、必要に応じて他のレジストリー内のイメージをスキャンするために、各セキュアクラスターに Scanner-slim バージョンをインストールします。

詳細は、Operator を使用したセキュアクラスターへの RHACS のインストール を参照してください。

12.3.3.2. Helm の使用

セキュアクラスターサービスの Helm チャート (**secured-cluster-services**) は、各セキュアクラスターに Scanner-slim バージョンをインストールします。Kubernetes では、セキュアクラスターサービスに、オプションのコンポーネントとして Scanner-slim が含まれています。一方、OpenShift Container Platform では、OpenShift Container Platform 統合レジストリーと、必要に応じて他のレジストリー内のイメージをスキャンするために、RHACS によって各セキュアクラスターに Scanner-slim バージョンがインストールされます。

- OpenShift Container Platform でのインストールについては、カスタマイズせずに secured-cluster-services Helm チャートをインストールする を参照してください。
- Amazon Elastic Kubernetes Service (Amazon EKS)、Google Kubernetes Engine (Google GKE)、Microsoft Azure Kubernetes Service (Microsoft AKS) などの OpenShift Container Platform 以外でのインストールについては、カスタマイズせずに secured-cluster-services Helm チャートをインストールする を参照してください。

12.3.3.3. インストール後の検証

手順

- セキュアクラスターのステータスで、Scanner が存在し、健全であることを確認します。
  a. RHACS ポータルで、**Platform Configuration → Clusters** に移動します。
  b. **Clusters** ビューで、クラスターを選択して詳細を表示します。
  c. **Health Status** カードに、**Scanner** が存在し、**Healthy** としてマークされていることを確認します。

12.3.3.4. イメージスキャンの使用

roxctl CLI、Jenkins、および API を使用して、クラスター固有の OpenShift Container Platform 統合イメージレジストリーに保存されているイメージをスキャンできます。スキャン委譲の設定で適切なクラスターを指定することも、**roxctl** CLI、Jenkins、および API で利用可能なクラスターパラメーターを使用することもできます。

roxctl CLI を使用してイメージをスキャンする方法の詳細は、roxctl CLI を使用したイメージスキャン を参照してください。

12.4. スキャンの設定

アクティブなイメージと非アクティブなイメージの自動スキャンなど、スキャンの設定を指定できます。

12.4.1. アクティブなイメージの自動スキャン

Red Hat Advanced Cluster Security for Kubernetes はアクティブなイメージを定期的にスキャンし、イメージスキャン結果を更新して最新の脆弱性定義を反映します。アクティブなイメージは、お使いの環境にデプロイしたイメージです。

注記

Red Hat Advanced Cluster Security for Kubernetes 3.0.57 から、イメージの **ウォッチ** 設定を指定して、非アクティブなイメージの自動スキャンを有効にできます。

Central は、Scanner またはその他の統合イメージスキャナーからすべてのアクティブなイメージのスキャン結果を取得し、その結果を 4 時間ごとに更新します。

roxctl CLI を使用して、オンデマンドでイメージスキャンの結果を確認することもできます。

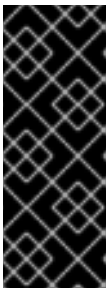
12.4.2. 非アクティブなイメージのスキャン

Red Hat Advanced Cluster Security for Kubernetes (RHACS) は、すべてのアクティブな (デプロイされた) イメージを 4 時間ごとにスキャンし、イメージスキャンの結果を更新して最新の脆弱性定義を反映します。

非アクティブな (デプロイされていない) イメージを自動的にスキャンするように RHACS を設定することもできます。

手順

1. RHACS ポータルで、**Vulnerability Management → Workload CVEs** をクリックします。
2. **Manage watched images** をクリックします。
3. **Image name** フィールドに、レジストリーで始まりイメージタグで終わる完全修飾イメージ名を入力します (例: **docker.io/library/nginx:latest**)。
4. **Add image to watch list**をクリックします。
5. オプション: 監視対象のイメージを削除するには、**Manage watched images** ウィンドウでイメージを見つけて、**Remove watch** をクリックします。



重要

RHACS ポータルで、**Platform Configuration → System Configuration** をクリックして、データ保持設定を表示します。

ウォッチリストから削除されたイメージに関連するすべてのデータは、**System Configuration** ページに記載されている日数の間 RHACS ポータルに表示され続け、その期間が終了した後にのみ削除されます。

6. **Close** をクリックして、**Workload CVEs** ページに戻ります。

関連情報

- 非アクティブなイメージのスキャン
- roxctl CLI のインストール

12.5. 脆弱性について

RHACS は、複数の脆弱性フィードから脆弱性定義と更新を取得します。これらのフィードには、NVD などの一般的な性質のものと、Alpine、Debian、Ubuntu などのディストリビューション固有のものがあります。検出された脆弱性の表示と対処の詳細は、脆弱性の管理 を参照してください。

12.5.1. 脆弱性定義の取得

オンラインモードでは、Central が 1 つのフィードから 5 分ごとに脆弱性定義を取得します。このフィードは、アップストリームのソースからの脆弱性定義を組み合わせたものであり、3 時間ごとに更新されます。

- フィードのアドレスは **https://definitions.stackrox.io** です。
- **ROX_SCANNER_VULN_UPDATE_INTERVAL** 環境変数を設定することで、Central および StackRox Scanner のデフォルトのクエリー頻度を変更できます。

```
$ oc -n stackrox set env deploy/central ROX_SCANNER_VULN_UPDATE_INTERVAL=<value> 1
```

1 Kubernetes を使用する場合は、**oc** の代わりに **kubectl** を入力します。

次のガイドラインに注意してください。

- StackRox スキャナーの設定マップには、スキャナーの更新頻度を設定するための **updater.interval** パラメーターがまだありますが、**fetchFromCentral** パラメーターは含まれなくなりました。
- この環境変数の設定は、Scanner V4 ではサポートされていません。

RHACS が使用する脆弱性ソースの詳細は、「Red Hat Advanced Cluster Security for Kubernetes アーキテクチャー」の「脆弱性ソース」を参照してください。

関連情報

- 脆弱性ソース

12.5.2. ダッシュボードの脆弱性スコアについて

Red Hat Advanced Cluster Security for Kubernetes ポータルの脆弱性管理ダッシュボードには、脆弱性ごとに 1 つの Common Vulnerability Scoring System (CVSS) ベーススコアが表示されます。RHACS は、次の基準に基づいて CVSS スコアを表示します。

- CVSS v3 スコアが利用可能な場合、スコアが **v3** とともに表示されます。例: **6.5(v3)**

注記

CVSS v3 スコアは、StackRox Scanner バージョン 1.3.5 以降または Scanner V4 を使用している場合にのみ利用できます。

- CVSS v3 スコアが利用できない場合、CVSS v2 スコアのみが表示されることがあります。(例: **6.5**)。

API を使用して CVSS スコアを取得できます。脆弱性に関して CVSS v3 情報が利用可能な場合、応答に CVSS v3 と CVSS v2 の両方の情報が含まれることがあります。

Red Hat セキュリティーアドバイザリー (RHSA) の場合、CVSS スコアは、関連するすべての CVE の中で最も高い CVSS スコアに設定されます。1 つの RHSA に複数の CVE が含まれることがあります。Red Hat は、脆弱性が他の Red Hat 製品に与える影響に基づいて異なるスコアを割り当てることがあります。

12.6. 言語固有の脆弱性スキャンの無効化

スキャナーは、デフォルトでプログラミング言語固有の依存関係の脆弱性を特定します。言語固有の依存関係スキャンを無効にすることができます。

手順

- 言語固有の脆弱性スキャンを無効にするには、以下のコマンドを実行します。

```
$ oc -n stackrox set env deploy/scanner \ 1
  ROX_LANGUAGE_VULNS=false 2
```

1. Kubernetes を使用する場合は、**oc** の代わりに **kubectl** を入力します。
2. Red Hat Advanced Cluster Security for Kubernetes バージョン 3.0.47 以前を使用している場合は、環境変数名 **ROX_LANGUAGE_VULNS** を、**LANGUAGE_VULNS** に置き換えます。

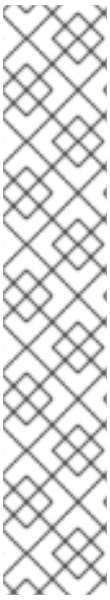
12.7. 関連情報

- Red Hat CVE データベース

第13章 イメージの署名の確認

Red Hat Advanced Cluster Security for Kubernetes (RHACS) を使用して、事前に設定されたキーに対してイメージ署名を検証することで、クラスター内のコンテナーイメージの整合性を確保できます。

署名されていないイメージや署名が確認されていないイメージをブロックするポリシーを作成できます。RHACS アドミッションコントローラーを使用してポリシーを適用し、無許可のデプロイメントの作成を停止することもできます。



注記

- RHACS は、Cosign 署名と Cosign 公開鍵/証明書検証のみをサポートします。Cosign の詳細は、Cosign overview を参照してください。
- Cosign 署名検証の場合、RHACS は透明性ログ Rekor との通信をサポートしていません。
- 署名検証には、少なくとも 1 つの Cosign 検証方法を使用して署名統合を設定する必要があります。
- すべてのデプロイおよび監視されたイメージに対して以下を実行します。
  - RHACS は署名を 4 時間ごとに取得および検証します。
  - RHACS は、署名統合検証データを変更または更新するたびに署名を検証します。

13.1. 署名統合の設定

イメージ署名の検証を実行する前に、まず RHACS で署名統合を作成する必要があります。

署名統合は複数の検証方法で設定できます。次の検証方法がサポートされています。

- Cosign 公開鍵
- Cosign 証明書

13.1.1. Cosign 公開鍵の設定

前提条件

- PEM でエンコードされた Cosign 公開鍵がすでに存在している必要があります。Cosign の詳細は、Cosign overview を参照してください。

手順

1. RHACS ポータルで、**Platform Configuration** → **Integrations** を選択します。
2. **Signature Integrations** までスクロールし、**Signature** をクリックします。
3. **New integration** をクリックします。
4. **Integration name** の名前を入力します。
5. **Cosign public Keys** → **Add a new public key** をクリックします。

6. **Public key** 名を入力します。
7. **Public key value** フィールドに、PEM でエンコードされた公開鍵を入力します。
8. (オプション)**Add a new public key** をクリックして詳細を入力すると、複数のキーを追加できます。
9. **Save** をクリックします。

13.1.2. Cosign 証明書の設定

前提条件

- 証明書のアイデンティティーと発行者がある。オプションで、PEM でエンコードされた証明書とチェーンがある。Cosign 証明書の詳細は、Cosign 証明書の検証 を参照してください。

手順

1. RHACS ポータルで、**Platform Configuration** → **Integrations** を選択します。
2. **Signature Integrations** までスクロールし、**Signature** をクリックします。
3. **New integration** をクリックします。
4. **Integration name** の名前を入力します。
5. **Cosign certificates** → **Add a new certificate verification** をクリックします。
6. **Certificate OIDC Issuer** を入力します。RE2 構文 ではオプションで正規表現を使用できます。
7. **Certificate identity** を入力します。RE2 構文 ではオプションで正規表現を使用できます。
8. (オプション) 証明書を検証するために **Certificate Chain PEM encoded** を入力します。チェーンが提供されていない場合、証明書は Fulcio ルートに対して検証されます。
9. (オプション) 署名を検証するために **PEM でエンコードされた証明書** を入力します。
10. (オプション) **Add a new certificate verification** をクリックして詳細を入力すると、複数の証明書検証を追加できます。
11. **Save** をクリックします。

13.2. ポリシーでの署名検証の使用

カスタムセキュリティーポリシーの作成時に、**Trusted image signers** ポリシー条件を使用してイメージ署名を検証できます。

前提条件

- 最低でも 1 つ以上の Cosign 公開鍵で署名統合を設定している。

手順

1. ポリシーの作成または編集時には、**Policy criteria** セクションのポリシーフィールドドロップ領域に、**Not verified by trusted image signers**ポリシー条件をドラッグします。

2. **Select** をクリックします。

3. リストから信頼されるイメージ署名を選択し、**Save** をクリックします。

関連情報

- システムポリシービューからのセキュリティーポリシーの作成
- ポリシー条件

13.3. 署名の検証の実施

ユーザーが署名されていないイメージを使用できないように、RHACS アドミッションコントローラーを使用して署名検証を有効にできます。最初に、クラスター設定で **Contact Image Scanners** 機能を有効にする必要があります。次に、セキュリティーポリシーを作成して署名の検証を強制する間に、**Inform and enforce** オプションを使用できます。

詳細は、アドミッションコントローラーの適用の有効化 を参照してください。

関連情報

- システムポリシービューからのセキュリティーポリシーの作成

第14章 脆弱性の管理

14.1. 脆弱性管理の概要

環境内のセキュリティーの脆弱性が攻撃者によって悪用されると、サービス拒否攻撃の実行、リモートコードの実行、機密データへの不正アクセスなどの不正なアクションが実行される可能性があります。したがって、脆弱性の管理は、Kubernetes セキュリティープログラムを成功させるための基本的なステップです。

14.1.1. 脆弱性管理プロセス

脆弱性管理は、脆弱性を特定して修復する継続的なプロセスです。Red Hat Advanced Cluster Security for Kubernetes は、脆弱性管理プロセスを容易にするのに役立ちます。

脆弱性管理プログラムには、多くの場合、以下の重要なタスクが含まれます。

- アセット評価の実行
- 脆弱性の優先順位付け
- 露出の評価
- 措置の実行
- 継続的なアセットの再評価

Red Hat Advanced Cluster Security for Kubernetes は、組織が OpenShift Container Platform および Kubernetes クラスターで継続的な評価を実行するのに役立ちます。これにより、組織は、環境内の脆弱性に優先順位を付けて対処するために必要なコンテキスト情報をより効果的に提供できます。

14.1.1.1. アセット評価の実行

組織のアセットの評価を実行するには、以下のアクションが含まれます。

- 環境内のアセットの特定
- これらのアセットをスキャンして、既知の脆弱性を特定する
- 環境内の脆弱性について、影響を受ける利害関係者に報告する

Red Hat Advanced Cluster Security for Kubernetes を Kubernetes または OpenShift Container Platform クラスターにインストールすると、最初にクラスター内で実行されているアセットが集約され、それらのアセットを識別できるようになります。RHACS を使用すると、OpenShift Container Platform および Kubernetes クラスターで継続的な評価を実行できます。RHACS は、環境内の脆弱性に優先順位を付けて、より効果的に対処するためのコンテキスト情報を提供します。

RHACS を使用した脆弱性管理プロセスで監視する必要がある重要なアセットには、次のものがあります。

- **コンポーネント**: コンポーネントは、イメージの一部として使用したり、ノードで実行したりできるソフトウェアパッケージです。コンポーネントは、脆弱性が存在する最低レベルです。したがって、特定の方法でソフトウェアコンポーネントをアップグレード、変更、または削除して脆弱性を修正する必要があります。

- **イメージ**: コードの実行可能な部分を実行するための環境を作成するソフトウェアコンポーネントおよびコードのコレクション。イメージでは、コンポーネントをアップグレードして脆弱性を修正できます。

- **ノード**: OpenShift または Kubernetes および OpenShift Container Platform または Kubernetes サービスを設定するコンポーネントを使用してアプリケーションを管理し、実行するために使用されるサーバー。

RHACS はこれらのアセットを次の構造にグループ化します。

- **Deployment**: 1 つまたは複数のイメージに基づくコンテナーで Pod を実行できる Kubernetes のアプリケーションの定義。

- **namespace**: アプリケーションをサポートおよび分離するデプロイメントなどのリソースのグループ。

- **クラスター**: OpenShift または Kubernetes を使用してアプリケーションを実行するために使用されるノードのグループ。

RHACS は、アセットをスキャンして既知の脆弱性を検出し、Common Vulnerabilities and Exposures (CVE) データを使用して既知の脆弱性の影響を評価します。

14.1.1.2. 脆弱性の優先順位付け

次の質問に答えて、アクションと調査のために環境の脆弱性に優先順位を付けます。

- 影響を受けるアセットは、組織にとってどの程度重要ですか？

- 脆弱性の重大度がどの程度の場合に、調査の必要がありますか？

- 脆弱性は、影響を受けるソフトウェアコンポーネントのパッチで修正できますか？

- 脆弱性の存在は、組織のセキュリティーポリシーのいずれかに違反していますか？

これらの質問への回答は、セキュリティーおよび開発チームが脆弱性の露出を測定する必要があるかどうかを判断します。

Red Hat Advanced Cluster Security for Kubernetes では、アプリケーションやコンポーネントの脆弱性を優先順位付けする手段を提供します。

14.1.1.3. 露出の評価

脆弱性の露出を評価するには、以下の質問に回答してください。

- アプリケーションは脆弱性の影響を受けますか？

- 脆弱性は他の要因によって軽減されていますか？

- この脆弱性の悪用につながる可能性のある既知の脅威はありますか？

- 脆弱性のあるソフトウェアパッケージを使用していますか？

- 特定の脆弱性およびソフトウェアパッケージに時間を割くことに価値はありますか？

評価に基づいて、以下のアクションを実行します。

- 脆弱性が公開されていない、脆弱性が使いの環境に適用されないと判断した場合は、脆弱

- 脆弱性が公開されていないか、脆弱性がお使いの環境に適用されないと判断した場合は、脆弱性を誤検出としてマークすることを検討してください。
- リスクにさらされた場合は、そのリスクの修正、軽減、または受け入れることを希望するかを検討してください。
- 攻撃対象領域を減らすためにソフトウェアパッケージを削除または変更するかどうかを検討してください。

14.1.1.4. 措置の実行

脆弱性に対するアクションを実行することを決定したら、次のいずれかのアクションを実行できます。

- 脆弱性を修正する
- リスクを軽減して受け入れる
- リスクを受け入れる
- 脆弱性を誤検出としてマークする

以下のアクションのいずれかを実行すると、脆弱性を修復できます。

- ソフトウェアパッケージを削除する
- ソフトウェアパッケージを脆弱性のないバージョンに更新する

14.2. 脆弱性の確認と対処

一般的な脆弱性管理タスクには、脆弱性の特定と優先順位付け、脆弱性の修復、および新しい脅威の監視が含まれます。

14.2.1. 脆弱性の表示

これまで、RHACS では、システムで検出された脆弱性を脆弱性管理ダッシュボードに表示していました。このダッシュボードは RHACS 4.5 で非推奨となり、今後のリリースで削除される予定です。ダッシュボードの詳細は、脆弱性管理ダッシュボードの使用 を参照してください。

Vulnerability Management → **Workload CVEs** ページに、システム内のクラスターで実行されているアプリケーションの脆弱性に関する情報が表示されます。イメージとデプロイメント全体の脆弱性情報を表示できます。**Workload CVEs** ページには、脆弱性のあるイメージとデプロイメントを表示する機能や、イメージ、デプロイメント、namespace、クラスター、CVE、コンポーネント、コンポーネントソースでフィルタリングする機能など、詳細なフィルタリング機能があります。

14.2.2. ワークロードの CVE の表示

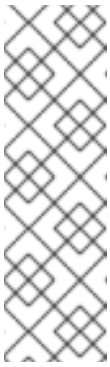
Vulnerability Management → **Workload CVEs** ページに、システム内のクラスターで実行されているアプリケーションの脆弱性に関する情報が表示されます。イメージとデプロイメント全体の脆弱性情報を表示できます。**Workload CVEs** ページには、脆弱性のあるイメージとデプロイメントを表示する機能や、イメージ、デプロイメント、namespace、クラスター、CVE、コンポーネント、コンポーネントソースでフィルタリングする機能など、ダッシュボードよりも詳細なフィルタリング機能があります。

手順

1. イメージの CVE をすべて表示するには、**View image vulnerabilities** リストから **Image vulnerabilities** を選択します。

2. **View image vulnerabilities** リストから、イメージの表示方法を選択します。以下のオプションがあります。

   - **Image vulnerabilities**: RHACS によって CVE が検出されたイメージとデプロイメントを表示します。
   - **Images without vulnerabilities**: 以下の条件を 1 つ以上満たすイメージを表示します。
     - CVE がないイメージ
     - CVE の検出漏れにつながるスキャナーエラーが報告されたイメージ



注記

このリストには、実際に脆弱性を含むイメージが誤って表示される場合があります。たとえば、スキャナーがイメージをスキャンでき、そのことが RHACS に認識されているにもかかわらず、スキャンが正常に完了しなかった場合、脆弱性は検出されません。このような状況が発生するのは、RHACS のスキャナーでサポートされていないオペレーティングシステムがイメージに含まれている場合です。イメージリスト内のイメージにマウスを移動するか、イメージ名をクリックして詳細を表示すると、スキャンエラーが表示されます。

3. CVE をエンティティー別にフィルタリングするには、適切なフィルターと属性を選択します。複数のエンティティーと属性を選択するには、右矢印アイコンをクリックして別の条件を追加します。必要に応じて、テキストなどの適切な情報を入力するか、日付またはオブジェクトを選択します。

   フィルターのエンティティーと属性を次の表に示します。

表14.1 CVE のフィルタリング

エンティティー	属性
Image	• **Name**: イメージの名前。 • **Operating system**: イメージのオペレーティングシステム。 • **Tag**: イメージのタグ。 • **Label**: イメージのラベル。 • **Registry**: イメージが配置されているレジストリー。

エンティティー	属性

CVE	• **Name**: CVE の名前。
	• **Discovered time**: RHACS が CVE を検出した日付。
	• **CVSS**: CVE の重大度。重大度は、次のオプションから選択できます。
	  ◦ **is greater than**
	  ◦ **is greater than or equal to**
	  ◦ **is equal to**
	  ◦ **is less than or equal to**
	  ◦ **is less than**

Image Component	• **Name**: イメージコンポーネントの名前 (例: **activerecord-sql-server-adapter**)。
	• **Source**:
	  ◦ OS
	  ◦ Python
	  ◦ Java
	  ◦ Ruby
	  ◦ Node.js
	  ◦ Go
	  ◦ Dotnet Core Runtime
	  ◦ Infrastructure
	• **Version**: イメージコンポーネントのバージョン (例: **3.4.21**)。これを使用すると、たとえばコンポーネント名と組み合わせて、特定のバージョンのコンポーネントを検索できます。

Deployment	• **Name**: デプロイメントの名前。 • **Label**: デプロイメントのラベル。 • **Annotation**: デプロイメントのアノテーション。

エンティティー	属性
Namespace	• **Name**: namespace の名前。 • **Label**: namespace のラベル。 • **Annotation**: namespace のアノテーション。
Cluster	• **Name**: クラスターの名前。 • **Label**: クラスターのラベル。 • **Type**: クラスターのタイプ (例: OCP)。 • **Platform type**: プラットフォームタイプ (例: OpenShift 4 クラスター)。

4. 次のオプションを選択して、結果のリストを絞り込むことができます。

   - **Prioritize by namespace view**: リスクの優先度に従って並べ替えられた namespace のリストを表示します。このビューを使用すると、最も重要な領域をすばやく特定して対処できます。このビューで、テーブル行の **<number> deployments** をクリックすると、選択した namespace のデプロイメント、イメージ、および CVE のみを表示するフィルターが適用された状態で、ワークロードの CVE のリストビューに戻ります。

   - **Default filters**: **Workload CVEs** ページにアクセスしたときに自動的に適用される CVE 重大度と CVE ステータスのフィルターを選択できます。これらのフィルターは、このページにのみ適用され、RHACS Web ポータルの別のセクションまたはブックマークされた URL からページにアクセスしたときに適用されます。フィルターはブラウザーのローカルストレージに保存されます。

   - **CVE severity**: 1 つ以上のレベルを選択できます。

   - **CVE status**: **Fixable** または **Not fixable** を選択できます。

注記

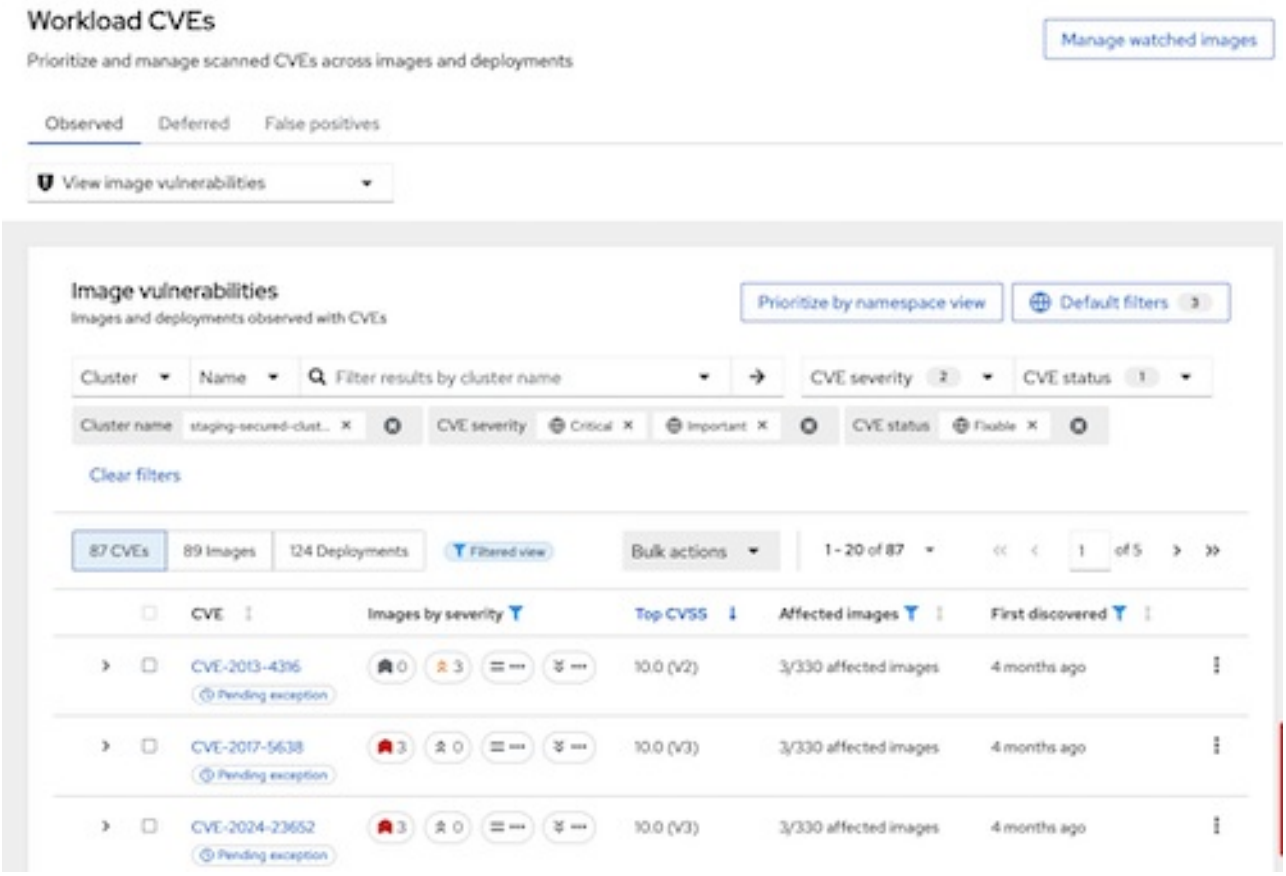
Filtered view アイコンは、表示された結果が選択した条件に基づいてフィルターされたことを示します。**Clear filters** をクリックしてすべてのフィルターを削除することも、個々のフィルターをクリックして削除することもできます。

結果のリストで、CVE、イメージ名、またはデプロイメント名をクリックすると、項目に関する詳細情報が表示されます。たとえば、項目タイプに応じて、次の情報を表示できます。

- CVE が修正可能かどうか
- イメージがアクティブかどうか
- CVE を含むイメージの Dockerfile 行
- Red Hat の CVE およびその他の CVE データベースに関する情報への外部リンク

検索例

次の図は、**staging-secured-cluster** というクラスターの検索条件の例を示しています。これは、そのクラスターについて、重大度が重大および重要で、ステータスが修正可能である CVE を表示するための条件です。



14.2.3. ノードの CVE の表示

RHACS を使用すると、ノード内の脆弱性を特定できます。特定される脆弱性は次のとおりです。

- Kubernetes コアコンポーネントの脆弱性
- Docker、CRI-O、runC、containerd などのコンテナーランタイムの脆弱性

RHACS がスキャンできるオペレーティングシステムの詳細は、「サポート対象オペレーティングシステム」を参照してください。

手順

1. RHACS ポータルで、**Vulnerability Management → Node CVEs** をクリックします。
2. データを表示するために、次のいずれかのタスクを実行します。
   - すべてのノードに影響するすべての CVE のリストを表示するには、**<number> CVEs** を選択します。
   - CVE を含むノードのリストを表示するには、**<number> Nodes** を選択します。
3. オプション: CVE をエンティティー別にフィルタリングするには、適切なフィルターと属性を選択します。フィルタリング条件をさらに追加するには、次の手順に従います。

a. リストからエンティティーまたは属性を選択します。

b. 必要に応じて、テキストなどの適切な情報を入力するか、日付またはオブジェクトを選択します。

c. 右矢印アイコンをクリックします。

d. オプション: 追加のエンティティーと属性を選択し、右矢印アイコンをクリックして追加します。フィルターのエンティティーと属性を次の表に示します。

表14.2 CVE のフィルタリング

エンティティー	属性
Node	• **Name**: ノードの名前。 • **Operating system**: ノードのオペレーティングシステム (例: Red Hat Enterprise Linux (RHEL))。 • **Label**: ノードのラベル。 • **Annotation**: ノードのアノテーション。 • **Scan time**: ノードのスキャン日。

CVE	• **Name**: CVE の名前。
	• **Discovered time**: RHACS が CVE を検出した日付。
	• **CVSS**: CVE の重大度。重大度は、次のオプションから選択できます。
	  ◦ **is greater than**
	  ◦ **is greater than or equal to**
	  ◦ **is equal to**
	  ◦ **is less than or equal to**
	  ◦ **is less than**

Node Component	• **Name**: コンポーネントの名前。 • **Version**: コンポーネントのバージョン (例: **4.15.0-2024**)。これを使用すると、たとえばコンポーネント名と組み合わせて、特定のバージョンのコンポーネントを検索できます。

エンティティー	属性
Cluster	• **Name**: クラスターの名前。 • **Label**: クラスターのラベル。 • **Type**: クラスターのタイプ (例: OCP)。 • **Platform type**: プラットフォームのタイプ (例: OpenShift 4 クラスター)。

4. オプション: 結果のリストを絞り込むには、次のいずれかのタスクを実行します。
   - **CVE severity** をクリックし、1 つ以上のレベルを選択します。
   - **CVE status** をクリックし、**Fixable** または **Not fixable** を選択します。
5. オプション: ノードの詳細と、そのノードの CVSS スコアと修正可能な CVE に基づく CVE 情報を表示するには、ノードのリストでノード名をクリックします。

14.2.3.1. ノードの脆弱性の特定を無効にする

ノード内の脆弱性の特定はデフォルトで有効になっています。これは RHACS ポータルから無効にできます。

手順

1. RHACS ポータルで、**Platform Configuration** → **Integrations** に移動します。
2. **Image Integrations** セクションで **StackRox Scanner** を選択します。
3. スキャナーのリストから **StackRox** スキャナーを選択して詳細を表示します。
4. **Edit** をクリックします。
5. イメージスキャナーのみを使用し、ノードスキャナーを使用しない場合は、**Image Scanner** をクリックします。
6. **Save** をクリックします。

関連情報

- サポート対象オペレーティングシステム

14.2.4. プラットフォームの CVE の表示

Platform CVEs ページには、システム内のクラスターの脆弱性に関する情報が表示されます。

手順

1. **Vulnerability Management** → **Platform CVEs** をクリックします。

2. 適切なフィルターと属性を選択することで、CVE をエンティティー別にフィルタリングできます。右矢印アイコンをクリックして別の条件を追加することで、複数のエンティティーと属性を選択できます。必要に応じて、テキストなどの適切な情報を入力するか、日付またはオブジェクトを選択します。フィルターのエンティティーと属性を次の表に示します。

表14.3 CVE のフィルタリング

エンティティー	属性
クラスター	• **Name**: クラスターの名前。 • **Label**: クラスターのラベル。 • **Type**: クラスターのタイプ (例: OCP)。 • **Platform type**: プラットフォームタイプ (例: OpenShift 4 クラスター)。

CVE	• **Name**: CVE の名前。
	• **Discovered time**: RHACS が CVE を検出した日付。
	• **CVSS**: CVE の重大度。重大度は、次のオプションから選択できます。
	  ○ **is greater than**
	  ○ **is greater than or equal to**
	  ○ **is equal to**
	  ○ **is less than or equal to**
	  ○ **is less than**
	• **Type**: CVE のタイプ:
	  ○ Kubernetes CVE
	  ○ Istio CVE
	  ○ OpenShift CVE

3. CVE ステータスでフィルタリングするには、**CVE status** をクリックし、**Fixable** または **Not fixable** を選択します。

注記

Filtered view アイコンは、表示された結果が選択した条件に基づいてフィルターされたことを示します。**Clear filters** をクリックしてすべてのフィルターを削除することも、個々のフィルターをクリックして削除することもできます。

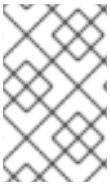
結果のリストで CVE をクリックすると、その項目の詳細情報が表示されます。たとえば、次の情報が入力されている場合は、その情報を表示できます。

- CVE のドキュメント
- Red Hat の CVE およびその他の CVE データベースに関する情報への外部リンク
- CVE が修正可能か修正不可能か
- 影響を受けるクラスターのリスト

14.2.5. CVE の除外

ノードとプラットフォームの CVE をスヌーズし、ノード、プラットフォーム、およびイメージの CVE を延期または誤検出としてマークすることで、RHACS で CVE を除外または無視できます。CVE が誤検出であることがわかっている場合、または CVE を軽減するための手順をすでに実行している場合は、CVE を除外することを推奨します。スヌーズされた CVE は脆弱性レポートに表示されず、ポリシー違反をトリガーすることもありません。

CVE をスヌーズして、指定した期間グローバルに無視することができます。CVE をスヌーズするのに承認は必要ありません。



注記

ノードおよびプラットフォーム CVE をスヌーズするには、**ROX_VULN_MGMT_LEGACY_SNOOZE** 環境変数を **true** に設定する必要があります。

CVE を延期したり誤検出としてマークしたりすることは、例外管理ワークフローを通じて行われます。このワークフローでは、保留中、承認済み、拒否済みの延期要求および誤検出要求を表示できます。CVE 例外の範囲を、1 つのイメージ、1 つのイメージのすべてのタグ、またはすべてのイメージに対してグローバルに設定できます。

要求を承認または拒否する場合は、コメントを追加する必要があります。CVE は、例外要求が承認されるまで監視対象ステータスのままになります。別のユーザーによって拒否された保留中の延期要求は、レポート、ポリシー違反、およびシステム内の他の場所で引き続き表示されますが、**Vulnerability Management → Workload CVEs** にアクセスすると、CVE の横に **Pending exception** ラベルが表示されます。

延期または誤検出の例外が承認されると、次の影響があります。

- **Vulnerability Management → Workflow CVEs** の **Observed** タブから CVE が削除され、**Deferred** または **False positive** タブに移動する
- CVE によって CVE に関連するポリシー違反がトリガーされなくなる
- 自動生成された脆弱性レポートに CVE が表示されなくなる

14.2.5.1. プラットフォームとノードの CVE のスヌーズ

インフラストラクチャーに関連しないプラットフォームおよびノードの CVE をスヌーズできます。CVE は、スヌーズを解除するまで、1 日、1 週間、2 週間、1 カ月間、または無期限にスヌーズできます。CVE のスヌーズは直ちに有効になり、追加の承認手順は必要ありません。

注記

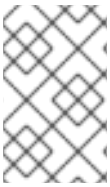
CVE をスヌーズする機能は、デフォルトでは Web ポータルまたは API で有効になっていません。CVE をスヌーズする機能を有効にするには、ランタイム環境変数 **ROX_VULN_MGMT_LEGACY_SNOOZE** を **true** に設定してください。

手順

1. RHACS ポータルで、次のいずれかのタスクを実行します。
   - プラットフォームの CVE を表示するには、**Vulnerability Management → Platform CVEs** をクリックします。
   - ノードの CVE を表示するには、**Vulnerability Management → Node CVEs** をクリックします。
2. 1 つ以上の CVE を選択します。
3. CVE をスヌーズするための適切な方法を選択します。
   - 1 つの CVE を選択した場合は、オーバーフローメニュー をクリックし、**Snooze CVE** を選択します。
   - 複数の CVE を選択した場合は、**Bulk actions → Snooze CVEs** をクリックします。
4. スヌーズする期間を選択します。
5. **Snooze CVEs** をクリックします。
   CVE のスヌーズを要求したことを確認するメッセージが表示されます。

14.2.5.2. プラットフォームとノードの CVE のスヌーズ解除

以前にスヌーズしたプラットフォームおよびノードの CVE のスヌーズを解除できます。



注記

CVE をスヌーズする機能は、デフォルトでは Web ポータルまたは API で有効になっていません。CVE をスヌーズする機能を有効にするには、ランタイム環境変数 **ROX_VULN_MGMT_LEGACY_SNOOZE** を **true** に設定してください。

手順

1. RHACS ポータルで、次のいずれかのタスクを実行します。
   - プラットフォームの CVE のリストを表示するには、**Vulnerability Management → Platform CVEs** をクリックします。
   - ノードの CVE のリストを表示するには、**Vulnerability Management → Node CVEs** をクリックします。
2. スヌーズされた CVE のリストを表示するために、ヘッダービューで **Show snoozed CVEs** をクリックします。
3. スヌーズされた CVE のリストから CVE を 1 つ以上選択します。

4. CVE のスヌーズを解除するには、適切な方法を選択します。

    - 1 つの CVE を選択した場合は、オーバーフローメニュー をクリックし、**Unsnooze CVE** を選択します。
    - 複数の CVE を選択した場合は、**Bulk actions → Unsnooze CVEs** をクリックします。

5. もう一度 **Unsnooze CVEs** をクリックします。
   CVE のスヌーズ解除を要求したことを確認するメッセージが表示されます。

14.2.5.3. スヌーズされた CVE の表示

スヌーズされたプラットフォームおよびノード CVE のリストを表示できます。

注記

CVE をスヌーズする機能は、デフォルトでは Web ポータルまたは API で有効になっていません。CVE をスヌーズする機能を有効にするには、ランタイム環境変数 **ROX_VULN_MGMT_LEGACY_SNOOZE** を **true** に設定してください。

手順

1. RHACS ポータルで、次のいずれかのタスクを実行します。
    - プラットフォームの CVE のリストを表示するには、**Vulnerability Management → Platform CVEs** をクリックします。
    - ノードの CVE のリストを表示するには、**Vulnerability Management → Node CVEs** をクリックします。
2. **Show snoozed CVEs** をクリックし、リストを表示します。

14.2.5.4. 脆弱性をグローバルに誤検出としてマークする

グローバルに、つまりすべてのイメージを対象に脆弱性を誤検出としてマークすることで、脆弱性の例外を作成できます。例外管理ワークフローで、脆弱性を誤検出としてマークする要求の承認を受ける必要があります。

前提条件

- **VulnerabilityManagementRequests** リソースに対する **write** 権限がある。

手順

1. RHACS ポータルで、**Vulnerability Management → Workload CVEs** をクリックします。
2. CVE をマークするための適切な方法を選択します。
    - 1 つの CVE をマークする場合は、次の手順を実行します。
        a. 操作を実行する CVE が含まれている行を見つけます。

b. 特定した CVE のオーバーフローメニュー  をクリックし**Mark as false positive** を選択します。

- 複数の CVE をマークする場合は、次の手順を実行します。

  a. 各 CVE を選択します。

  b. **Bulk actions** ドロップダウンリストから、**Mark as false positives** を選択します。

3. 例外を要求する理由を入力します。
4. オプション: 例外要求に含まれる CVE を確認するには、**CVE selections** をクリックします。
5. **Submit request** をクリックします。
   例外を要求したことを確認するメッセージが表示されます。
6. オプション: 承認リンクをコピーして組織の例外承認者と共有するには、コピーアイコンをクリックします。
7. **Close** をクリックします。

14.2.5.5. イメージまたはイメージタグの脆弱性を誤検出としてマークする

脆弱性の例外を作成する場合、1 つのイメージを対象に、またはイメージに関連付けられているすべてのタグを対象に、脆弱性を誤検出としてマークすることができます。例外管理ワークフローで、脆弱性を誤検出としてマークする要求の承認を受ける必要があります。

前提条件

- **VulnerabilityManagementRequests** リソースに対する **write** 権限がある。

手順

1. RHACS ポータルで、**Vulnerability Management → Workload CVEs** をクリックします。
2. イメージのリストを表示するために、**<number> Images** をクリックします。
3. 誤検出としてマークするイメージがリストされている行を見つけて、イメージ名をクリックします。
4. CVE をマークするための適切な方法を選択します。

- 1 つの CVE をマークする場合は、次の手順を実行します。

  a. 操作を実行する CVE が含まれている行を見つけます。

  b. 特定した CVE のオーバーフローメニュー  をクリックし**Mark as false positive** を選択します。

- 複数の CVE をマークする場合は、次の手順を実行します。

  a. 各 CVE を選択します。

  b. **Bulk actions** ドロップダウンリストから、**Mark as false positives** を選択します。

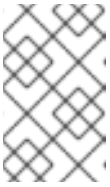
5. 範囲を選択します。イメージに関連付けられているすべてのタグを選択するか、イメージのみを選択できます。
6. 例外を要求する理由を入力します。
7. オプション: 例外要求に含まれる CVE を確認するには、**CVE selections** をクリックします。
8. **Submit request** をクリックします。
   例外を要求したことを確認するメッセージが表示されます。
9. オプション: 承認リンクをコピーして組織の例外承認者と共有するには、コピーアイコンをクリックします。
10. **Close** をクリックします。

14.2.5.6. 延期された CVE と誤検出された CVE の表示

Workload CVEs ページを使用すると、延期された CVE や誤検出としてマークされた CVE を表示できます。

手順

1. 延期された CVE または誤検出としてマークされた CVE (承認者によって承認された例外を含む) を表示するには、**Vulnerability Management → Workload CVEs** をクリックします。次のいずれかの操作を実行します。
   - 延期された CVE を表示するには、**Deferred** タブをクリックします。
   - 誤検出としてマークされた CVE を表示するには、**False positives** タブをクリックします。



注記

延期された CVE または誤検出の CVE を承認、拒否、または変更するには、**Vulnerability Management → Exception Management** をクリックします。

2. オプション: 延期または誤検出に関する追加情報を表示するには、**Request details** 列の **View** をクリックします。**Exception Management** ページが表示されます。

14.2.5.7. CVE の延期

軽減策の有無にかかわらずリスクを受け入れ、CVE を延期することができます。例外管理ワークフローで延期要求の承認を受ける必要があります。

前提条件

- **VulnerabilityManagementRequests** リソースに対する **write** 権限がある。

手順

1. RHACS ポータルで、**Vulnerability Management → Workload CVEs** をクリックします。
2. CVE を延期するための適切な方法を選択します。
   - 1 つの CVE を延期する場合は、次の手順を実行します。

a. 誤検出としてマークする CVE を含む行を見つけます。

b. 特定した CVE のオーバーフローメニュー  をクリックし、**Defer CVE** をクリックします。

- 複数の CVE を延期する場合は、次の手順を実行します。

a. 各 CVE を選択します。

b. **Bulk actions → Defer CVEs** をクリックします。

3. 延期期間を選択します。
4. 例外を要求する理由を入力します。
5. オプション: 例外メニューに含まれる CVE を確認するには、**CVE selections** クリックします。
6. **Submit request** をクリックします。
延期を要求したことを確認するメールが届きます。
7. オプション: 承認リンクをコピーして組織の例外承認者と共有するには、コピーアイコンをクリックします。
8. **Close** をクリックします。

14.2.5.7.1. 脆弱性例外の有効期限の設定

脆弱性の管理例外に使用できる期間を設定できます。このオプションは、ユーザーが CVE の延期を要求した場合に利用できます。

前提条件

- **VulnerabilityManagementRequests** リソースに対する **write** 権限がある。

手順

1. RHACS ポータルで、**Platform Configuration → Exception Configuration** に移動します。
2. CVE の延期を要求するときにユーザーが選択できる有効期限を設定できます。期間を有効にすると、ユーザーが利用できるようになります。無効にすると、ユーザーインターフェイスから期間が削除されます。

14.2.5.8. CVE を延期または誤検出としてマークするための例外要求の確認と管理

CVE を延期および誤検出としてマークするための例外要求を確認、更新、承認、または拒否できます。

前提条件

- **VulnerabilityManagementRequests** リソースに対する **write** 権限がある。

手順

1. 保留中のリクエストのリストを表示するために、次のいずれかのタスクを実行します。

- 承認リンクをブラウザーに貼り付けます。
- **Vulnerability Management → Exception Management** をクリックし、**Pending requests** タブで要求の名前をクリックします。

2. 脆弱性の範囲を確認し、承認するかどうかを決定します。
3. 保留中の要求を管理するための適切なオプションを選択します。
   - 要求を拒否し、CVE を監視対象状態に戻す場合は、**Deny request** をクリックします。拒否の理由を入力し、**Deny** をクリックします。
   - 要求を承認する場合は、**Approve request** をクリックします。承認の理由を入力し、**Approve** をクリックします。
4. 作成した要求をキャンセルし、CVE を監視対象ステータスに戻すには、**Cancel request** をクリックします。キャンセルできるのは、自分が作成した要求だけです。
5. 作成した要求の延期期間または理由を更新するには、**Update request** をクリックします。更新できるのは、自分が作成した要求だけです。
   変更を加えたら、**Submit request** をクリックします。

   要求を送信したことを確認するメールが届きます。

14.2.6. CVE が含まれるコンポーネントを取り込んだイメージ内の Dockerfile 行を特定する

CVE が含まれるコンポーネントを取り込んだイメージ内の Dockerfile 行を特定できます。

手順

問題のある行を表示するには、以下を行います。

1. RHACS ポータルで、**Vulnerability Management → Workload CVEs** をクリックします。
2. タブをクリックすると、CVE の種類が表示されます。次のタブがあります。
   - **Observed**
   - **Deferred**
   - **False positives**
3. CVE のリストで CVE 名をクリックすると、CVE の詳細を含むページが開きます。**Affected components** 列に、CVE が含まれるコンポーネントがリスト表示されます。
4. CVE を展開すると、そのコンポーネントを取り込んだ Dockerfile 行などの追加情報が表示されます。

14.2.7. 新しいコンポーネントバージョンの検索

以下の手順では、アップグレード先のコンポーネントのバージョンを見つけます。

手順

1. RHACS ポータルで、**Vulnerability Management → Workload CVEs** をクリックします。

2. **<number> Images** をクリックしてイメージを選択します。

3. 追加情報を表示するために、CVE を見つけて展開アイコンをクリックします。
追加情報には、CVE が含まれるコンポーネントと、CVE が修正されたバージョン (修正可能な場合) が含まれています。

4. イメージを新しいバージョンに更新します。

14.2.8. API を使用したワークロードの脆弱性のエクスポート

API を使用して、Red Hat Advanced Cluster Security for Kubernetes のワークロードの脆弱性をエクスポートできます。

これらの例では、ワークロードはデプロイメントとそれに関連付けられたイメージで構成されます。エクスポートでは、**/v1/export/vuln-mgmt/workloads** ストリーミング API が使用されます。デプロイメントとイメージを組み合わせてエクスポートできます。**images** ペイロードには完全な脆弱性情報が含まれています。出力はストリーミングされ、次のスキーマを持ちます。

```
{"result": {"deployment": {...}, "images": [...]}}
...
{"result": {"deployment": {...}, "images": [...]}}
```

次の例では、これらの環境変数が設定されていることを前提としています。

- **ROX_API_TOKEN**: **Deployment** および **Image** リソースの **view** 権限を持つ API トークン
- **ROX_ENDPOINT**: Central の API が利用できるエンドポイント
- すべてのワークロードをエクスポートするには、次のコマンドを入力します。

```
$ curl -H "Authorization: Bearer $ROX_API_TOKEN" $ROX_ENDPOINT/v1/export/vuln-mgmt/workloads
```

- クエリータイムアウトを 60 秒にしてすべてのワークロードをエクスポートするには、次のコマンドを入力します。

```
$ curl -H "Authorization: Bearer $ROX_API_TOKEN" $ROX_ENDPOINT/v1/export/vuln-mgmt/workloads?timeout=60
```

- クエリー **Deployment:app Namespace:default** に一致するすべてのワークロードをエクスポートするには、次のコマンドを入力します。

```
$ curl -H "Authorization: Bearer $ROX_API_TOKEN" $ROX_ENDPOINT/v1/export/vuln-mgmt/workloads?query=Deployment%3Aapp%2BNamespace%3Adefault
```

関連情報

- 検索およびフィルタリング

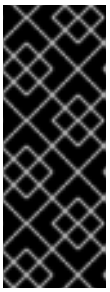
14.2.8.1. 非アクティブなイメージのスキャン

Red Hat Advanced Cluster Security for Kubernetes (RHACS) は、すべてのアクティブな (デプロイされた) イメージを 4 時間ごとにスキャンし、イメージスキャンの結果を更新して最新の脆弱性定義を反映します。

非アクティブな (デプロイされていない) イメージを自動的にスキャンするように RHACS を設定することもできます。

手順

1. RHACS ポータルで、**Vulnerability Management → Workload CVEs** をクリックします。
2. **Manage watched images** をクリックします。
3. **Image name** フィールドに、レジストリーで始まりイメージタグで終わる完全修飾イメージ名を入力します (例: **docker.io/library/nginx:latest**)。
4. **Add image to watch list**をクリックします。
5. オプション: 監視対象のイメージを削除するには、**Manage watched images** ウィンドウでイメージを見つけて、**Remove watch** をクリックします。



重要

RHACS ポータルで、**Platform Configuration → System Configuration** をクリックして、データ保持設定を表示します。

ウォッチリストから削除されたイメージに関連するすべてのデータは、**System Configuration** ページに記載されている日数の間 RHACS ポータルに表示され続け、その期間が終了した後にのみ削除されます。

6. **Close** をクリックして、**Workload CVEs** ページに戻ります。

14.3. 脆弱性レポート

RHACS Web ポータルの **Vulnerability Management → Vulnerability Reporting** メニューから、オンデマンドのイメージ脆弱性レポートを作成してダウンロードできます。このレポートには、イメージおよびデプロイメント内の Common Vulnerabilities and Exposures (RHACS でワークロード CVE と呼ばれるもの) の包括的なリストが含まれています。

このレポートを監査人や社内関係者と共有するには、RHACS でメールをスケジュールするか、レポートをダウンロードして他の方法で共有します。

14.3.1. チームへの脆弱性の報告

組織は脆弱性を絶えず再評価して報告する必要があるため、脆弱性管理プロセスを支援するために主要な利害関係者へのコミュニケーションをスケジュールすることが役立つと考える組織もあります。

Red Hat Advanced Cluster Security for Kubernetes を使用して、このように繰り返し発生するメールによるコミュニケーションスケジュールを作成できます。これらのコミュニケーションは、主要な利害関係者が必要とする最も関連性の高い情報に限定する必要があります。

これらの連絡を送信するには、次の質問を考慮する必要があります。

- 利害関係者とコミュニケーションをとるときに最も影響を与えるスケジュールは何ですか？
- 誰が対象者となりますか？
- レポートで特定の重大度の脆弱性のみを送信する必要がありますか？

- レポートで修正可能な脆弱性のみを送信する必要がありますか？

14.3.2. 脆弱性管理レポート設定の作成

RHACS は、脆弱性管理レポート設定を作成するプロセスを順をおって説明します。この設定により、スケジュールされた時間に実行されるレポートジョブまたはオンデマンドで実行されるレポートジョブに含まれる情報が決まります。

手順

1. RHACS ポータルで、**Vulnerability Management → Vulnerability Reporting** をクリックします。
2. **Create report** をクリックします。
3. **Report name** フィールドにレポート設定の名前を入力します。
4. オプション: **Report description** フィールドにレポート設定を説明するテキストを入力します。
5. **CVE severity** フィールドで、レポート設定に含める Common Vulnerabilities and Exposures (CVE) の重大度を選択します。
6. **CVE status** を選択します。**Fixable**、**Unfixable**、またはその両方を選択できます。
7. **Image type** フィールドで、デプロイされたイメージ、監視されたイメージ、またはその両方からの CVE を含めるかを選択します。
8. **CVEs discovered since** フィールドで、レポート設定に CVE を含める期間を選択します。
9. **Configure collection included** フィールドで、少なくとも 1 つのコレクションを設定する必要があります。次のいずれかの操作を実行します。
   - 追加する既存のコレクションを選択します。コレクション情報を表示したり、コレクションを編集したり、コレクション結果のプレビューを表示するには、**View** をクリックします。コレクションを表示するときに、フィールドにテキストを入力すると、そのテキスト文字列に一致するコレクションが検索されます。
   - **Create collection** をクリックして新しいコレクションを作成します。

   注記

   コレクションの詳細は、「関連情報」セクションの「デプロイメントコレクションの作成および使用」を参照してください。
10. **次へ** をクリックして配信先を設定し、必要に応じて配信スケジュールを設定します。

14.3.2.1. 配信先およびスケジューリングの設定

前のページで最後にスケジュールされたレポート以降に検出された CVE を含めるオプションを選択した場合を除き、脆弱性レポートの宛先と配信スケジュールの設定は、任意です。このオプションを選択した場合は、脆弱性レポートの宛先と配信スケジュールを設定する必要があります。

手順

1. 配信先を設定するには、**Configure delivery destinations** セクションで、配信先を追加し、レポートのスケジュールを設定できます。

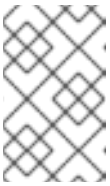
2. レポートをメールで送信するには、少なくとも 1 つのメール通知機能を設定する必要があります。レポートをメールで送信するには、既存の通知機能を選択するか、新しいメール通知機能を作成します。メール通知機能の作成の詳細は、「関連情報」セクションの「メールプラグインの設定」を参照してください。
通知機能を選択すると、通知機能で **Default recipients** として設定されたメールアドレスが **Distribution list** フィールドに表示されます。メールアドレスは、コンマで区切って追加できます。

3. デフォルトのメールテンプレートが自動的に適用されます。このデフォルトのテンプレートを編集するには、以下の手順を実行します。

   a. 編集アイコンをクリックし、カスタマイズした件名とメール本文を **Edit** タブに入力します。

   b. **Preview** タブをクリックして、提案されたテンプレートを表示します。

   c. **Apply** をクリックして、テンプレートへの変更を保存します。



注記

特定のレポートのレポートジョブを確認すると、レポートの作成時にデフォルトのテンプレートが使用されたかカスタマイズされたテンプレートが使用されたかを確認できます。

4. **Configure schedule** セクションで、レポートの頻度と曜日を選択します。

5. **Next** をクリックして脆弱性レポートの設定を確認し、作成を完了します。

14.3.2.2. レポート設定の確認および作成

脆弱性レポートを作成する前に、その設定の詳細を確認できます。

手順

1. **Review and create** セクションでは、レポート設定パラメーター、配信先、メール配信を選択した場合に使用されるメールテンプレート、配信スケジュール、レポート形式を確認できます。変更を加えるには、**戻る** をクリックして前のセクションに戻り、変更するフィールドを編集します。

2. **Create** をクリックしてレポート設定を作成し、保存します。

14.3.3. 脆弱性レポートのパーミッション

レポートを作成、表示、およびダウンロードする機能は、ユーザーアカウントのアクセス制御設定またはロールおよび権限セットによって異なります。

たとえば、ユーザーアカウントにアクセス権限があるデータのレポートのみを表示、作成、ダウンロードできます。さらに、以下の制限が適用されます。

- ダウンロードできるのは、自分が生成したレポートのみで、他のユーザーが生成したレポートをダウンロードできません。

- レポート権限は、ユーザーアカウントのアクセス設定に応じて制限されます。アカウントのアクセス設定が変更された場合、古いレポートには変更が反映されません。たとえば、新しい権限が与えられ、その権限で現在許可されている脆弱性データを表示したい場合は、新しい脆弱性レポートを作成する必要があります。

14.3.4. 脆弱性レポート設定の編集

既存の脆弱性レポート設定は、レポート設定のリストから編集するか、最初に個別のレポート設定を選択して編集できます。

手順

1. RHACS Web ポータルで、**Vulnerability Management → Vulnerability Reporting** をクリックします。
2. 既存の脆弱性レポート設定を編集するには、次のいずれかの操作を実行します。
   - レポート設定のリストで編集するレポート設定を見つけます。オーバーフローメニュー  をクリックし、**Edit report** を選択します。
   - レポート設定のリストでレポート設定名をクリックします。次に、**Actions** をクリックし、**Edit report** を選択します。
3. レポート設定に変更を加えて保存します。

14.3.5. 脆弱性レポートのダウンロード

オンデマンドの脆弱性レポートを生成し、ダウンロードできます。

注記

ダウンロードできるのは、自分が生成したレポートのみで、他のユーザーが生成したレポートをダウンロードできません。

手順

1. RHACS Web ポータルで、**Vulnerability Management → Vulnerability Reporting** をクリックします。
2. レポート設定のリストで、ダウンロード可能なレポートの作成に使用するレポート設定を見つけます。
3. 次のいずれかの方法を使用して、脆弱性レポートを生成します。
   - 一覧からレポートを生成するには、以下を実行します。
     a. オーバーフローメニュー  をクリックし、**Generate download** を選択します。**アクティブなジョブのステータス** 列には、レポート作成のステータスが表示されます。**Processing** のステータスが消えたら、レポートをダウンロードできます。
   - レポートウィンドウからレポートを生成するには、次の手順を実行します。

a. レポート設定名をクリックして、設定の詳細ウィンドウを開きます。

b. **Actions** をクリックして、**Generate download** を選択します。

4. レポートをダウンロードするには、レポート設定の一覧を表示する場合に、レポート設定名をクリックして開きます。

5. ヘッダーのメニューから **All report jobs** をクリックします。

6. レポートが完了したら、**Status** 列の **Ready for download** リンクをクリックします。レポートは **.csv** 形式で、ダウンロードするために **.zip** ファイルに圧縮されます。

14.3.6. 脆弱性レポートをオンデマンドで送信する

スケジュールされた送信時刻を待たずに、脆弱性レポートをすぐに送信できます。

手順

1. RHACS Web ポータルで、**Vulnerability Management → Vulnerability Reporting** をクリックします。

2. レポート設定のリストで、送信するレポートのレポート設定を見つけます。

3. オーバーフローメニュー をクリックし、**Send report now** を選択します。

14.3.7. 脆弱性レポート設定のクローン作成

脆弱性レポート設定の複製を作成することで、そのコピーを作成できます。これは、異なるデプロイメントまたは namespace での脆弱性をレポートするなど、軽微な変更を加えてレポート設定を再利用する場合に便利です。

手順

1. RHACS Web ポータルで、**Vulnerability Management → Vulnerability Reporting** をクリックします。

2. レポート設定のリストで、複製するレポート設定を見つけます。

3. **Clone report** をクリックします。

4. レポートのパラメーターと配信先に必要な変更を加えます。

5. **Create** をクリックします。

14.3.8. 脆弱性レポート設定の削除

レポート設定を削除すると、その設定と、この設定を使用して以前に実行されたレポートがすべて削除されます。

手順

1. RHACS Web ポータルで、**Vulnerability Management → Vulnerability Reporting** をクリックします。

2. レポートのリストで、削除するレポート設定を見つけます。
3. オーバーフローメニュー をクリックし、**Delete report** を選択します。

14.3.9. 脆弱性管理レポートのジョブ保持設定

脆弱性レポートジョブリクエストの有効期限を決定する設定や、レポートジョブのその他の保持設定を指定できます。

注記

これらの設定は、次の脆弱性レポートジョブには影響しません。

- **WAITING** または **PREPARING** 状態のジョブ (未完了のジョブ)
- 最後に成功したスケジュールされたレポートジョブ
- 最後に成功したオンデマンドのメール送信レポートジョブ
- 最後に成功したダウンロード可能なレポートジョブ
- 手動削除またはダウンロード可能なレポートのプルーニング設定によってレポートファイルが削除されていない、ダウンロード可能なレポートジョブ

手順

1. RHACS Web ポータルで、**Platform Configuration → System Configuration** に移動します。脆弱性レポートジョブに対して以下の設定を行うことができます。
   - **Vulnerability report run history retention** 実行された脆弱性レポートジョブの記録が保存される日数。この設定は、レポート設定が選択されている場合に、**Vulnerability Management → Vulnerability Reporting** の **All report jobs** タブにレポートジョブを表示する日数を制御します。除外日以降のレポート履歴はすべて削除されます。ただし、次のジョブは除きます。
     - 未完了のジョブ。
     - 準備されたダウンロード可能なレポートがシステムにまだ存在するジョブ。
     - 各ジョブタイプ (スケジュールされたメール、オンデマンドメール、またはダウンロード) の最後に成功したレポートジョブ。これにより、ユーザーは各タイプの最後に実行されたジョブに関する情報を確実に得ることができます。
   - **Prepared downloadable vulnerability reports retention days** レポート設定が選択されている場合に、準備が完了しているオンデマンドのダウンロード可能な脆弱性レポートジョブが、**Vulnerability Management → Vulnerability Reporting** の **All report jobs** タブでダウンロードできる日数。
   - **Prepared downloadable vulnerability reports limit** 準備されたダウンロード可能な脆弱性レポートジョブに割り当てられるスペースの制限 (MB 単位)。制限に達すると、ダウンロードキュー内の最も古いレポートジョブが削除されます。
2. これらの値を変更するには、**Edit** をクリックして変更を加え、**Save** をクリックします。

14.3.10. 関連情報

- デプロイメントコレクションの作成と使用
- コレクションへのアクセススコープの移行
- メールプラグインの設定

14.4. 脆弱性管理ダッシュボードの使用 (非推奨)

これまで、RHACS では、システムで検出された脆弱性を脆弱性管理ダッシュボードに表示していました。このダッシュボードを使用すると、イメージ、ノード、プラットフォームごとに脆弱性を表示できます。クラスター、namespace、デプロイメント、ノードコンポーネント、イメージコンポーネントごとに脆弱性を表示することもできます。このダッシュボードは RHACS 4.5 で非推奨となり、今後のリリースで削除される予定です。

重要

脆弱性に関する追加情報の表示、脆弱性の延期、脆弱性を誤検出としてマークする操作など、脆弱性に対する操作を実行するには、**Vulnerability Management → Workload CVEs** をクリックします。CVE を延期する要求と誤検出としてマークする要求を確認するには、**Vulnerability Management → Exception Management** をクリックします。

14.4.1. ダッシュボードを使用してアプリケーションの脆弱性を表示する

ダッシュボードを使用して、Red Hat Advanced Cluster Security for Kubernetes のアプリケーションの脆弱性を表示できます。

手順

1. RHACS ポータルで、**Vulnerability Management → Dashboard** に移動します。
2. **Dashboard** ビューヘッダーで、**Application & Infrastructure → Namespaces** または **Deployments** を選択します。
3. リストから、確認する **Namespace** または **Deployment** を検索し、選択します。
4. アプリケーションの詳細を取得するには、右側の **Related entities** からエンティティーを選択します。

14.4.2. ダッシュボードを使用してイメージの脆弱性を表示する

ダッシュボードを使用して、Red Hat Advanced Cluster Security for Kubernetes のイメージの脆弱性を表示できます。

手順

1. RHACS ポータルで、**Vulnerability Management → Dashboard** に移動します。
2. **Dashboard** ビューのヘッダーで、**<number> Images** を選択します。
3. イメージのリストから、調査するイメージを選択します。次のいずれかの手順を実行して、リストをフィルタリングすることもできます。
    a. 検索バーに **Image** と入力して、**Image** 属性を選択します。

b. 検索バーにイメージ名を入力します。

4. イメージの詳細ビューで、リストされている CVE を確認し、影響を受けるコンポーネントに対処するためのアクションを優先的に実行します。

5. 右側の **Related entities** から **Components** を選択し、選択したイメージの影響を受けるすべてのコンポーネントに関する詳細情報を取得します。または、特定の CVE の影響を受けるコンポーネントを見つけるには、**Image findings** セクションの **Affected components** 列から **Components** を選択します。

14.4.3. ダッシュボードを使用してクラスターの脆弱性を表示する

Red Hat Advanced Cluster Security for Kubernetes を使用すると、クラスター内の脆弱性を表示できます。

手順

1. RHACS ポータルで、**Vulnerability Management** → **Dashboard** に移動します。
2. **Dashboard** ビューのヘッダーで、**Application & Infrastructure** → **Clusters** を選択します。
3. クラスターのリストから、調査するクラスターを選択します。
4. クラスターの脆弱性を確認し、クラスター上の影響を受けるノードに対するアクションの優先順位を決定します。

14.4.4. ダッシュボードを使用してノードの脆弱性を表示する

Red Hat Advanced Cluster Security for Kubernetes を使用して、特定ノードで脆弱性を表示できます。

手順

1. RHACS ポータルで、**Vulnerability Management** → **Dashboard** に移動します。
2. **Dashboard** ビューヘッダーで、**Nodes** を選択します。
3. ノードのリストから、調査するノードを選択します。
4. 選択したノードの脆弱性を確認し、アクションの実行に優先順位を付けます。
5. 影響を受けるコンポーネントに関する詳細情報を取得するには、右側の **Related entities** から **Components** を選択します。

14.4.5. ダッシュボードを使用して最も脆弱なイメージコンポーネントを特定する

Vulnerability Management ビューを使用して、脆弱なイメージコンポーネントを特定します。

手順

1. RHACS ポータルにアクセスし、ナビゲーションメニューから **Vulnerability Management** → **Dashboard** をクリックします。
2. **Vulnerability Management** ビューのヘッダーから、**Application & Infrastructure** → **Image Components** を選択します。

3. **Image Components** ビューで、**Image CVEs** 列ヘッダーを選択して、CVE 数に基づいてコンポーネントを降順 (最も多いものから順) に並べます。

14.4.6. ダッシュボードを使用して修正可能な CVE の詳細のみを表示する

Vulnerability Management ビューを使用して、修正可能な CVE をフィルタリングして表示します。

手順

1. RHACS ポータルで、**Vulnerability Management → Dashboard** に移動します。
2. **Vulnerability Management** ビューのヘッダーの **Filter CVEs** で、**Fixable** をクリックします。

14.4.7. ダッシュボードを使用してベースイメージのオペレーティングシステムを特定する

Vulnerability Management ビューを使用して、ベースイメージのオペレーティングシステムを特定します。

手順

1. RHACS ポータルにアクセスし、ナビゲーションメニューから **Vulnerability Management → Dashboard** をクリックします。
2. **Vulnerability Management** ビューヘッダーから **Images** を選択します。
3. **Image OS** 列の下に、すべてのイメージのベースオペレーティングシステム (OS) および OS バージョンを表示します。
4. イメージを選択して、その詳細を表示します。ベースオペレーティングシステムは、**Image Summary → Details and Metadata** セクションでも利用できます。

注記

Red Hat Advanced Cluster Security for Kubernetes は、以下のいずれかの場合に、**Image OS** を **unknown** としてリスト表示します。

- オペレーティングシステム情報が利用できない場合、または
- 使用中のイメージスキャナーでこの情報が提供されない場合。

Docker Trusted Registry、Google Container Registry、および Anchore では、この情報を提供されません。

14.4.8. ダッシュボードを使用して最もリスクの高いオブジェクトを特定する

Vulnerability Management ビューを使用して、環境内の主要なリスクオブジェクトを特定します。**Top Risky** ウィジェットは、環境内のトップリスクのイメージ、デプロイメント、クラスター、および namespace に関する情報を表示します。このリスクは、脆弱性の数と CVSS スコアに基づいて決定されます。

手順

1. RHACS ポータルにアクセスし、ナビゲーションメニューから **Vulnerability Management → Dashboard** をクリックします。

2. **Top Risky** ウィジェットヘッダーを選択して、リスクイメージ、デプロイメント、クラスター、および namespace の中から選択します。
グラフの小さな円は、選択したオブジェクト (イメージ、デプロイメント、クラスター、namespace) を表します。円にマウスをかざし、その円が表すオブジェクトの概要を確認します。円を選択して、選択したオブジェクト、その関連エンティティー、およびエンティティー間の接続に関する詳細情報を表示します。

   たとえば、**Top Risky Deployments by CVE Count and CVSS score**を表示する場合には、グラフの各円はデプロイメントを表します。

   - デプロイメントにカーソルを合わせると、デプロイメントの概要が表示されます。これには、デプロイメント名、クラスターと namespace の名前、重大度、リスクの優先度、CVSS、および CVE カウント (修正可能を含む) が含まれます。

   - デプロイメントを選択すると、選択したデプロイメントの **Deployment** ビューが開きます。**Deployment** ビューには、デプロイメントの詳細情報が表示され、そのデプロイメントのポリシー違反、共通脆弱性、CVE、およびリスクイメージに関する情報が含まれます。

3. ウィジェットヘッダーで **View All** を選択して、選択したタイプのオブジェクトをすべて表示します。たとえば、**Top Risky Deployments by CVE Count and CVSS score**を選択した場合には、**View All** を選択して、インフラストラクチャー内のすべてのデプロイメントに関する詳細情報を表示できます。

14.4.9. ダッシュボードを使用して最もリスクの高いイメージとコンポーネントを特定する

Top Risky と同様に、**Top Riskiest** ウィジェットには、最もリスクの高いイメージとコンポーネントの名前がリスト表示されます。このウィジェットには、リストされたイメージ内の CVE の総数と修正可能な CVE の数も含まれています。

手順

1. RHACS ポータルに移動し、ナビゲーションメニューから **Vulnerability Management** をクリックします。

2. **Top Riskiest Images** ウィジェットヘッダーを選択して、リスクイメージとコンポーネントを選択します。**Top Riskiest Images** を表示する場合は、以下を実行します。

   - リスト内のイメージにカーソルを合わせると、イメージの概要が表示されます。これには、イメージ名、スキャン時間、CVE の数、重大度 (クリティカル、高、中、低) が含まれます。

   - イメージを選択すると、選択したイメージの **Image** ビューが開きます。**Image** ビューには、イメージの詳細が表示され、CVSS スコア別の CVE、最もリスクの高いコンポーネント、修正可能な CVE、およびイメージの Dockerfile に関する情報が含まれます。

3. ウィジェットヘッダーで **View All** を選択して、選択したタイプのオブジェクトをすべて表示します。たとえば、**Top Riskiest Components**を選択した場合は、**View All** を選んでインフラストラクチャー内のすべてのコンポーネントに関する詳細情報を表示できます。

14.4.10. ダッシュボードを使用してイメージの Dockerfile を表示する

Vulnerability Management ビューを使用して、イメージの脆弱性の根本的な原因を検索します。Dockerfile を表示して、Dockerfile 内のどのコマンドが脆弱性を導入したか、およびその単一のコマンドに関連付けられているすべてのコンポーネントを正確に見つけることができます。

Dockerfile セクションには、次の情報が表示されます。

- Dockerfile のすべてのレイヤー
- 各レイヤーの命令とその値
- 各レイヤーに含まれるコンポーネント
- 各レイヤーのコンポーネントの CVE 数

特定のレイヤーで導入されたコンポーネントがある場合は、デプロイメントアイコンを選択してコンポーネントの概要を表示できます。これらのコンポーネントに CVE がある場合は、個別のコンポーネントのデプロイメントアイコンを選択して、そのコンポーネントに影響を与える CVE の詳細を取得できます。

手順

1. RHACS ポータルで、**Vulnerability Management → Dashboard** に移動します。
2. **Top Riskiest Images** ウィジェットからイメージを選択するか、ダッシュボードの上部にある **Images** ボタンをクリックしてイメージを選択します。
3. **Image** の詳細ビューで、**Dockerfile** の横にある展開アイコンを選択して、手順、値、作成日、およびコンポーネントの概要を表示します。
4. 詳細情報を表示するには、個別のコンポーネントの展開アイコンを選択します。

14.4.11. ダッシュボードを使用して脆弱性をもたらすコンテナーイメージレイヤーを特定する

Vulnerability Management ダッシュボードを使用すると、脆弱なコンポーネントとそのコンポーネントが現れるイメージレイヤーを特定できます。

手順

1. RHACS ポータルにアクセスし、ナビゲーションメニューから **Vulnerability Management → Dashboard** をクリックします。
2. **Top Riskiest Images** ウィジェットからイメージを選択するか、ダッシュボードの上部にある **Images** ボタンをクリックしてイメージを選択します。
3. **Image** の詳細ビューで、**Dockerfile** の横にある展開アイコンを選択して、イメージコンポーネントの概要を表示します。
4. 特定のコンポーネントのデプロイメントアイコンを選択して、選択したコンポーネントに影響する CVE の詳細を取得します。

14.4.12. ダッシュボードを使用して最近検出された脆弱性を表示する

Vulnerability Management → Dashboard ビューの **Recently Detected Vulnerabilities** ウィジェットには、スキャンしたイメージで最近検出された脆弱性のリストが、スキャン時間と CVSS スコアに基づいて表示されます。また、CVE の影響を受けるイメージの数と、お使いの環境への影響 (パーセンテージ) に関する情報も含まれます。

- リスト内の CVE にカーソルを合わせると、CVE の概要が表示されます。これには、スキャン時間、CVSS スコア、説明、影響、および CVSSv2 と v3 のどちらを使用してスコアリングされたかが含まれます。

- CVE を選択すると、選択した **CVE** の詳細ビューが開きます。**CVE** の詳細ビューには、表示される CVE およびコンポーネント、イメージ、デプロイメントおよびデプロイメントの詳細が表示されます。

- **Recently Detected Vulnerabilities** ウィジェットヘッダーで **View All** を選択し、インフラストラクチャー内のすべての CVE のリストを表示します。CVE の一覧をフィルタリングすることもできます。

14.4.13. ダッシュボードを使用して最も一般的な脆弱性を表示する

Vulnerability Management → Dashboard ビューの **Most Common Vulnerabilities** ウィジェットには、最も多くのデプロイメントとイメージに影響を与える脆弱性のリストが、CVSS スコア順に表示されます。

- リスト内の CVE にカーソルを合わせると、CVE の概要が表示されます。これには、スキャン時間、CVSS スコア、説明、影響、および CVSSv2 と v3 のどちらを使用してスコアリングされたかが含まれます。

- CVE を選択すると、選択した **CVE** の詳細ビューが開きます。**CVE** の詳細ビューには、表示される CVE およびコンポーネント、イメージ、デプロイメントおよびデプロイメントの詳細が表示されます。

- **Most Common Vulnerabilities** ウィジェットヘッダーで **View All** を選択し、インフラストラクチャー内のすべての CVE のリストを表示します。CVE の一覧をフィルタリングすることもできます。CVE を CSV ファイルとしてエクスポートするには、**Export → Download CVES as CSV** の順に選択します。

14.4.14. ダッシュボードを使用して Kubernetes と Istio の脆弱性が最も多いクラスターを特定する

脆弱性管理ダッシュボードを使用すると、環境内で Kubernetes、Red Hat OpenShift、および Istio の脆弱性 (非推奨) が最も多いクラスターを特定できます。

手順

1. RHACS ポータルで、**Vulnerability Management→ Dashboard** をクリックします。**Clusters with most orchestrator and Istio vulnerabilities** ウィジェットには、各クラスター内の Kubernetes、Red Hat OpenShift、および Istio (非推奨) の脆弱性の数によってランク付けされたクラスターのリストが表示されます。リストの一番上にあるクラスターは、脆弱性の数が最も多いクラスターです。

2. リストからクラスターの 1 つをクリックして、クラスターの詳細を表示します。**Cluster** ビューには以下が含まれます。

   - **Cluster Summary** セクション。クラスターの詳細とメタデータ、最もリスクの高いオブジェクト (デプロイメント、namespace、およびイメージ)、最近検出された脆弱性、最もリスクの高いイメージ、および最も重大なポリシー違反のあるデプロイメントが表示されます。

   - **Cluster Findings** セクション。これには、失敗したポリシーのリストおよび修正可能な CVE のリストが含まれます。

- **Related Entities** セクション。クラスターに含まれる namespace、デプロイメント、ポリシー、イメージ、コンポーネント、CVE の数が表示されます。これらのエンティティーを選択して、詳細情報を表示できます。

3. ウィジェットヘッダーの **View All** をクリックして、すべてのクラスターのリストを表示します。

14.4.15. ダッシュボードを使用してノードの脆弱性を特定する

Vulnerability Management ビューを使用して、ノードの脆弱性を特定できます。特定される脆弱性には、Docker、CRI-O、runC、containerd などの Kubernetes コアコンポーネントとコンテナーランタイムの脆弱性も含まれます。RHACS がスキャンできるオペレーティングシステムの詳細は、「サポート対象オペレーティングシステム」を参照してください。

手順

1. RHACS ポータルで、**Vulnerability Management** → **Dashboard** に移動します。
2. ヘッダーの **Nodes** を選択し、ノードに影響するすべての CVE のリストを表示します。
3. リストからノードを選択し、そのノードに影響するすべての CVE の詳細を表示します。
   a. ノードを選択すると、選択したノードの **Node** の詳細パネルが開きます。**Node** ビューには、ノードの詳細が表示され、CVSS スコア別の CVE およびそのノードの修正可能な CVE に関する情報が含まれます。
   b. 選択したノードのすべての CVE のリストを表示するには、**CVEs by CVSS score** で、**View All** を選択します。CVE の一覧をフィルタリングすることもできます。
   c. 修正可能な CVE を CSV ファイルとしてエクスポートするには、**Node Findings** セクションで **Export as CSV** を選択します。

関連情報

- サポート対象オペレーティングシステム

14.4.16. ダッシュボードを使用して特定の CVE をブロックするポリシーを作成する

Vulnerability Management ビューから、新しいポリシーを作成したり、既存のポリシーに特定の CVE を追加したりすることができます。

手順

1. **Vulnerability Management** ビューヘッダーから **CVE** をクリックします。
2. 1 つ以上の CVE のチェックボックスを選択してから、**Add selected CVEs to Policy**(**add** アイコン) をクリックするか、リスト内の CVE の上にマウスを移動して **Add** アイコンを選択します。
3. **Policy Name** の場合:
   - 既存のポリシーに CVE を追加するには、ドロップダウンリストから既存のポリシーを選択します。
   - 新規ポリシーを作成するには、新規ポリシーの名前を入力し、**Create <policy_name>** を選択します。

4. **Severity** の値を選択します (**Critical**、**High**、**Medium**、または **Low** のいずれか)。
5. ポリシーを適用する **Lifecycle Stage** を、**Build** または **Deploy** から選択します。また、ライフサイクルステージの両方を選択することもできます。
6. **Description** ボックスに、ポリシーの詳細を入力します。
7. ポリシーを作成して後で有効にする場合は、**Enable Policy** トグルをオフにします。**Enable Policy** トグルはデフォルトでオンになっています。
8. このポリシーに含まれる CVE を確認してください。
9. **Save Policy** をクリックします。

14.5. RHCOS ノードホストのスキャン

OpenShift Container Platform の場合、コントロールプレーンとしてサポートされるオペレーティングシステムは、Red Hat Enterprise Linux CoreOS (RHCOS) のみです。一方、ノードホストの場合、OpenShift Container Platform は RHCOS と Red Hat Enterprise Linux (RHEL) の両方をサポートします。Red Hat Advanced Cluster Security for Kubernetes (RHACS) を使用すると、RHCOS ノードの脆弱性をスキャンし、潜在的なセキュリティー脅威を検出できます。

RHACS は、RHCOS インストールの一部としてノードホストにインストールされた RHCOS RPM をスキャンして、既知の脆弱性がないか調べます。

まず、RHACS は RHCOS コンポーネントを分析して検出します。次に、RHEL および OpenShift 4.X Open Vulnerability and Assessment Language (OVAL) v2 セキュリティーデータストリームを使用して、特定されたコンポーネントの脆弱性を照合します。

注記

- **roxctl** CLI を使用して RHACS をインストールした場合は、RHCOS ノードのスキャン機能を手動で有効にする必要があります。OpenShift Container Platform で Helm または Operator インストール方法を使用する場合、この機能はデフォルトで有効になります。

関連情報

- RHEL Versions Utilized by RHEL CoreOS and OCP

14.5.1. RHCOS ノードスキャンの有効化

OpenShift Container Platform を使用する場合は、Red Hat Advanced Cluster Security for Kubernetes (RHACS) を使用して、Red Hat Enterprise Linux CoreOS (RHCOS) ノードの脆弱性スキャンを有効にできます。

前提条件

- Secured クラスターの RHCOS ノードホストをスキャンするには、OpenShift Container Platform 4.11 以降に Secured クラスターをインストールしておく必要があります。サポートされるプラットフォームおよびアーキテクチャーの詳細は、Red Hat Advanced Cluster Security for Kubernetes Support Matrix を参照してください。RHACS のライフサイクルのサポート情報は、Red Hat Advanced Cluster Security for Kubernetes サポートポリシー を参照してください。

手順

1. 次のコマンドのいずれかを実行して、コンプライアンスコンテナーを更新します。

   - メトリクスが無効になっているデフォルトのコンプライアンスコンテナーの場合は、次のコマンドを実行します。

     ```
     $ oc -n stackrox patch daemonset/collector -p '{"spec":{"template":{"spec":{"containers":
     [{"name":"compliance","env":[{"name":"ROX_METRICS_PORT","value":"disabled"},
     {"name":"ROX_NODE_SCANNING_ENDPOINT","value":"127.0.0.1:8444"},
     {"name":"ROX_NODE_SCANNING_INTERVAL","value":"4h"},
     {"name":"ROX_NODE_SCANNING_INTERVAL_DEVIATION","value":"24m"},
     {"name":"ROX_NODE_SCANNING_MAX_INITIAL_WAIT","value":"5m"},
     {"name":"ROX_RHCOS_NODE_SCANNING","value":"true"},
     {"name":"ROX_CALL_NODE_INVENTORY_ENABLED","value":"true"}]}]}}}'
     ```

   - Prometheus メトリクスが有効になっているコンプライアンスコンテナーの場合は、次のコマンドを実行します。

     ```
     $ oc -n stackrox patch daemonset/collector -p '{"spec":{"template":{"spec":{"containers":
     [{"name":"compliance","env":[{"name":"ROX_METRICS_PORT","value":":9091"},
     {"name":"ROX_NODE_SCANNING_ENDPOINT","value":"127.0.0.1:8444"},
     {"name":"ROX_NODE_SCANNING_INTERVAL","value":"4h"},
     {"name":"ROX_NODE_SCANNING_INTERVAL_DEVIATION","value":"24m"},
     {"name":"ROX_NODE_SCANNING_MAX_INITIAL_WAIT","value":"5m"},
     {"name":"ROX_RHCOS_NODE_SCANNING","value":"true"},
     {"name":"ROX_CALL_NODE_INVENTORY_ENABLED","value":"true"}]}]}}}'
     ```

2. 次の手順を実行して、Collector DaemonSet (DS) を更新します。

   a. 次のコマンドを実行して、新しいボリュームマウントを Collector DS に追加します。

      ```
      $ oc -n stackrox patch daemonset/collector -p '{"spec":{"template":{"spec":{"volumes":
      [{"name":"tmp-volume","emptyDir":{}},{"name":"cache-volume","emptyDir":
      {"sizeLimit":"200Mi"}}]}}}}'
      ```

   b. 次のコマンドを実行して、新しい **NodeScanner** コンテナーを追加します。

      ```
      $ oc -n stackrox patch daemonset/collector -p '{"spec":{"template":{"spec":{"containers":
      [{"command":["/scanner","--nodeinventory","--config=",""],"env":
      [{"name":"ROX_NODE_NAME","valueFrom":{"fieldRef":
      {"apiVersion":"v1","fieldPath":"spec.nodeName"}}},
      {"name":"ROX_CLAIR_V4_SCANNING","value":"true"},
      {"name":"ROX_COMPLIANCE_OPERATOR_INTEGRATION","value":"true"},
      {"name":"ROX_CSV_EXPORT","value":"false"},
      {"name":"ROX_DECLARATIVE_CONFIGURATION","value":"false"},
      {"name":"ROX_INTEGRATIONS_AS_CONFIG","value":"false"},
      {"name":"ROX_NETPOL_FIELDS","value":"true"},
      {"name":"ROX_NETWORK_DETECTION_BASELINE_SIMULATION","value":"true"},
      {"name":"ROX_NETWORK_GRAPH_PATTERNFLY","value":"true"},
      {"name":"ROX_NODE_SCANNING_CACHE_TIME","value":"3h36m"},
      {"name":"ROX_NODE_SCANNING_INITIAL_BACKOFF","value":"30s"},
      {"name":"ROX_NODE_SCANNING_MAX_BACKOFF","value":"5m"},
      {"name":"ROX_PROCESSES_LISTENING_ON_PORT","value":"false"},
      {"name":"ROX_QUAY_ROBOT_ACCOUNTS","value":"true"},
      ```

{"name":"ROX_ROXCTL_NETPOL_GENERATE","value":"true"},
{"name":"ROX_SOURCED_AUTOGENERATED_INTEGRATIONS","value":"false"},
{"name":"ROX_SYSLOG_EXTRA_FIELDS","value":"true"},
{"name":"ROX_SYSTEM_HEALTH_PF","value":"false"},
{"name":"ROX_VULN_MGMT_WORKLOAD_CVES","value":"false"}],"image":"registry.redhat.io/advanced-cluster-security/rhacs-scanner-slim-rhel8:4.5.1","imagePullPolicy":"IfNotPresent","name":"node-inventory","ports":[{"containerPort":8444,"name":"grpc","protocol":"TCP"}],"volumeMounts":[{"mountPath":"/host","name":"host-root-ro","readOnly":true},
{"mountPath":"/tmp/","name":"tmp-volume"},{"mountPath":"/cache","name":"cache-volume"}]}]}}}}'

14.5.2. 分析と検出

RHACS を OpenShift Container Platform とともに使用すると、RHACS は分析と検出用に 2 つの調整コンテナー (Compliance コンテナーと Node-inventory コンテナー) を作成します。Compliance コンテナーは、以前の RHACS バージョンの一部としてすでに組み込まれていました。ただし、Node-inventory コンテナーは RHACS 4.0 で新しく追加されたもので、OpenShift Container Platform クラスターノードでのみ機能します。

起動時に、Compliance コンテナーと Node-inventory コンテナーは、5 分以内に Red Hat Enterprise Linux CoreOS (RHCOS) ソフトウェアコンポーネントの最初のインベントリースキャンを開始します。次に、Node-inventory コンテナーはノードのファイルシステムをスキャンして、インストールされている RPM パッケージを特定し、RHCOS ソフトウェアコンポーネントについてレポートします。その後、インベントリースキャンが定期的な間隔 (通常は 4 時間ごと) で行われます。Compliance コンテナーの ROX_NODE_SCANNING_INTERVAL 環境変数を設定することで、デフォルトの間隔をカスタマイズできます。

14.5.3. 脆弱性の照合

Central や Scanner などの Central サービスは、脆弱性の照合を実行します。Scanner は、Red Hat の Open Vulnerability and Assessment Language (OVAL) v2 セキュリティーデータストリームを使用して、Red Hat Enterprise Linux CoreOS (RHCOS) ソフトウェアコンポーネントの脆弱性を照合します。

以前のバージョンとは異なり、RHACS 4.0 では、カーネルとコンテナーのランタイムのバージョンを見つけるために Kubernetes ノードのメタデータを使用しなくなりました。代わりに、インストールされている RHCOS RPM を使用してその情報を評価します。

14.5.4. 関連する環境変数

次の環境変数を使用して、RHACS での RHCOS ノードのスキャンを設定できます。

表14.4 Node-inventory 設定

環境変数	説明
ROX_NODE_SCANNING_CACHE_TIME	キャッシュされたインベントリーが古いとみなされるまでの時間。デフォルトは **ROX_NODE_SCANNING_INTERVAL** の 90%、つまり **3h36m** です。
ROX_NODE_SCANNING_INITIAL_BACKOFF	バックオフファイルが見つかった場合にノードスキャンが遅延する最初の時間 (秒)。デフォルト値は **30s** です。

環境変数	説明
ROX_NODE_SCANNING_MAX_BACKOFF	バックオフの上限。デフォルト値は 5m で、これは Kubernetes 再起動ポリシー安定性タイマーの 50% です。

表14.5 コンプライアンス設定

環境変数	説明
ROX_NODE_SCANNING_INTERVAL	ノードスキャン間の間隔期間の基本値。デフォルト値は **4h** です。
ROX_NODE_SCANNING_INTERVAL_DEVIATION	ノードスキャンの継続時間は、基本間隔時間と異なる場合があります。ただし、最大値は **ROX_NODE_SCANNING_INTERVAL** によって制限されます。
ROX_NODE_SCANNING_MAX_INITIAL_WAIT	最初のノードスキャンまでの最大待機時間。ランダムに生成されます。この値を **0** に設定すると、初期ノードスキャンの待機時間を無効にすることができます。デフォルト値は **5m** です。

14.5.5. ダッシュボードを使用してノードの脆弱性を特定する

Vulnerability Management ビューを使用して、ノードの脆弱性を特定できます。特定される脆弱性には、Docker、CRI-O、runC、containerd などの Kubernetes コアコンポーネントとコンテナーランタイムの脆弱性も含まれます。RHACS がスキャンできるオペレーティングシステムの詳細は、「サポート対象オペレーティングシステム」を参照してください。

手順

1. RHACS ポータルで、**Vulnerability Management → Dashboard** に移動します。
2. ヘッダーの **Nodes** を選択し、ノードに影響するすべての CVE のリストを表示します。
3. リストからノードを選択し、そのノードに影響するすべての CVE の詳細を表示します。
   a. ノードを選択すると、選択したノードの **Node** の詳細パネルが開きます。**Node** ビューには、ノードの詳細が表示され、CVSS スコア別の CVE およびそのノードの修正可能な CVE に関する情報が含まれます。
   b. 選択したノードのすべての CVE のリストを表示するには、**CVEs by CVSS score** で、**View All** を選択します。CVE の一覧をフィルタリングすることもできます。
   c. 修正可能な CVE を CSV ファイルとしてエクスポートするには、**Node Findings** セクションで **Export as CSV** を選択します。

14.5.6. ノードの CVE の表示

RHACS を使用すると、ノード内の脆弱性を特定できます。特定される脆弱性は次のとおりです。

- Kubernetes コアコンポーネントの脆弱性
- Docker、CRI-O、runC、containerd などのコンテナーランタイムの脆弱性

RHACS がスキャンできるオペレーティングシステムの詳細は、「サポート対象オペレーティングシステム」を参照してください。

手順

1. RHACS ポータルで、**Vulnerability Management → Node CVEs** をクリックします。
2. データを表示するために、次のいずれかのタスクを実行します。
   - すべてのノードに影響するすべての CVE のリストを表示するには、**<number> CVEs** を選択します。
   - CVE を含むノードのリストを表示するには、**<number> Nodes** を選択します。
3. オプション: CVE をエンティティー別にフィルタリングするには、適切なフィルターと属性を選択します。フィルタリング条件をさらに追加するには、次の手順に従います。
   a. リストからエンティティーまたは属性を選択します。
   b. 必要に応じて、テキストなどの適切な情報を入力するか、日付またはオブジェクトを選択します。
   c. 右矢印アイコンをクリックします。
   d. オプション: 追加のエンティティーと属性を選択し、右矢印アイコンをクリックして追加します。フィルターのエンティティーと属性を次の表に示します。

表14.6 CVE のフィルタリング

エンティティー	属性
ノード	• **Name**: ノードの名前。 • **Operating system**: ノードのオペレーティングシステム (例: Red Hat Enterprise Linux (RHEL))。 • **Label**: ノードのラベル。 • **Annotation**: ノードのアノテーション。 • **Scan time**: ノードのスキャン日。

エンティティー	属性

CVE	• **Name**: CVE の名前。
	• **Discovered time**: RHACS が CVE を検出した日付。
	• **CVSS**: CVE の重大度。重大度は、次のオプションから選択できます。
	  ◦ **is greater than**
	  ◦ **is greater than or equal to**
	  ◦ **is equal to**
	  ◦ **is less than or equal to**
	  ◦ **is less than**

Node Component	• **Name**: コンポーネントの名前。 • **Version**: コンポーネントのバージョン (例: **4.15.0-2024**)。これを使用すると、たとえばコンポーネント名と組み合わせて、特定のバージョンのコンポーネントを検索できます。
クラスター	• **Name**: クラスターの名前。 • **Label**: クラスターのラベル。 • **Type**: クラスターのタイプ (例: OCP)。 • **Platform type**: プラットフォームのタイプ (例: OpenShift 4 クラスター)。

4. オプション: 結果のリストを絞り込むには、次のいずれかのタスクを実行します。
   - **CVE severity** をクリックし、1 つ以上のレベルを選択します。
   - **CVE status** をクリックし、**Fixable** または **Not fixable** を選択します。
5. オプション: ノードの詳細と、そのノードの CVSS スコアと修正可能な CVE に基づく CVE 情報を表示するには、ノードのリストでノード名をクリックします。

第15章 違反への対応

Red Hat Advanced Cluster Security for Kubernetes (RHACS) を使用すると、ポリシー違反を表示し、違反の実際の原因をドリルダウンして、是正措置を講じることができます。

RHACS の組み込みポリシーは、脆弱性 (CVE)、DevOps ベストプラクティスの違反、高リスクのビルドおよびデプロイメントプラクティス、不審な実行時の動作など、さまざまなセキュリティーの検出結果を特定します。カスタマイズなしに使用できるデフォルトのセキュリティーポリシーを使用するか、独自のカスタムポリシーを使用するかに関係なく、有効なポリシーが失敗すると、RHACS は違反を報告します。

15.1. 違反ビュー

Violations ビューですべての違反を分析し、修正措置を講じることができます。

RHACS ポータルで、**Violations** に移動して、検出された違反を確認します。

Violations ビューには、各行に次の属性を持つ違反のリストが表示されます。

- **ポリシー**: 違反したポリシーの名前。
- **Entity**: 違反が発生したエンティティー。
- **Type** : デプロイメント、namespace、クラスターなどのエンティティーのタイプ。
- **実施済み**: 違反が発生したときにポリシーが実施されたかどうかを示します。
- **重大度**: 重大度を **Low**、**Medium**、**High**、または **Critical** で示します。
- **カテゴリー**: ポリシーカテゴリー。ポリシーカテゴリーは、**Policy categories** タブの **Platform Configuration → Policy Management** にリストされます。
- **ライフサイクル**: ポリシーが適用されるライフサイクルステージ (**Build**、**Deploy**、または **Runtime**)。
- **Time** - 違反が発生した日時。

他のビューと同様に、以下のアクションを実行できます。

- 列見出しを選択して、違反を昇順または降順で並べ替えます。
- フィルターバーを使用して違反をフィルタリングします。詳細は、検索とフィルタリングセクションを参照してください。
- 違反の詳細を表示するには、**Violations** ビューで違反を選択します。

15.1.1. 違反を解決済みとしてマークする

ランタイム違反のあるポリシーが削除された場合、違反の試みは **Violations** ページから削除されません。違反を解決済みとしてマークすることで、手動で違反を削除できます。

手順

1. **Violations** を選択し、違反リストから違反を見つけます。

2. オーバーフローメニュー  をクリックし、次のいずれかのオプションを選択します。

    - **解決してプロセスベースラインに追加**: 違反を解決し、関連するプロセスをプロセスベースラインに追加します。プロセスを再度実行すると、新たな違反が表示されます。
    - Mark as resolved: 違反を解決します。

15.2. 違反の詳細の表示

Violations ビューで違反を選択すると、違反に関する詳細情報が含まれるウィンドウが開きます。複数のタブにグループ化された詳細情報が表示されます。

15.2.1. 違反タブ

Violation Details パネルの Violation タブには、ポリシーにどのように違反したかの説明が表示されます。ポリシーがデプロイフェーズ属性を対象としている場合は、違反名など、ポリシーに違反した特定の値を表示できます。ポリシーが実行時アクティビティーを対象としている場合は、引数やポリシーを作成した祖先プロセスなど、ポリシーに違反したプロセスに関する詳細情報を表示できます。

15.2.2. デプロイメントタブ

Details パネルの Deployment タブには、違反が適用されるデプロイメントの詳細が表示されます。

概要セクション
Deployment overview セクションには、以下の情報が一覧表示されます。

- **デプロイメント ID**: デプロイメントの英数字 ID。
- Deployment name: デプロイメントの名前。
- Deployment type: デプロイメントのタイプ。
- Cluster: コンテナーがデプロイされているクラスターの名前。
- Namespace: デプロイされたクラスターの一意の識別子。
- Replicas: レプリケートされたデプロイメントの数。
- Created: デプロイメントが作成された日時。
- Updated: デプロイメントが更新された日時。
- **ラベル**: 選択したデプロイメントに適用されるラベル。
- **アノテーション**: 選択したデプロイメントに適用されるアノテーション。
- **サービスアカウント**: 選択したデプロイメントのサービスアカウントの名前。

コンテナー設定セクション
Container configuration セクションには、以下の情報がリストされています。

- **コンテナー**: コンテナーごとに、以下の情報を提供します。
    - **イメージ名**: 選択したデプロイメントのイメージの名前。名前をクリックすると、イメージに関する詳細情報が表示されます。

- **Resources**: このセクションでは、次のフィールドの情報を提供します。
  - **CPU 要求 (コア)**: コンテナーにより要求されるコアの数。
  - **CPU 制限 (コア)** : コンテナーが要求できるコアの最大数。
  - **メモリー要求 (MB)**: コンテナーによって要求されるメモリーサイズ。
  - **メモリー制限 (MB)** : コンテナーが要求できる最大メモリー。
- **ボリューム**: コンテナーにマウントされているボリューム (存在する場合)。
- **シークレット**: 選択したデプロイメントに関連付けられているシークレット。シークレットごとに、次のフィールドの情報を提供します。
  - **名前**: シークレットの名前。
  - **コンテナーパス**: シークレットが保存される場所。
- **名前**: サービスがマウントされる場所の名前。
- **ソース**: データソースパス。
- **宛先**: データが保存されるパス。
- **タイプ**: ボリュームのタイプ。

ポート設定セクション
Port configuration セクションには、次のフィールドを含む、デプロイメント内のポートに関する情報が表示されます。

- ports: デプロイメントによって公開されるすべてのポート、およびこのデプロイメントとポートに関連付けられたすべての Kubernetes サービス (存在する場合)。ポートごとに、次のフィールドがリストされます。
  - containerPort: デプロイメントによって公開されるポート番号。
  - protocol: ポートで使用されるプロトコル (TCP や UDP など)。
  - exposure: ロードバランサーやノードポートなど、サービスの公開方法。
  - exposureInfo: このセクションでは、次のフィールドの情報を提供します。
    - level: サービスが内部でポートを公開するか、外部に公開するかどうかを示します。
    - serviceName : Kubernetes サービスの名前。
    - serviceID: RHACS に保存されている Kubernetes サービスの ID。
    - serviceClusterIp: **クラスター内** の別のデプロイメントまたはサービスがこのサービスにアクセスするために使用できる IP アドレス。これは外部 IP アドレスではありません。
    - servicePort: サービスによって使用されるポート。
    - nodePort: 外部トラフィックがノードに入ってくるノード上のポート。

- **externallps**: クラスターの外部からサービスに外部アクセスするために使用できる IP アドレス (存在する場合)。このフィールドは内部サービスでは利用できません。

セキュリティーコンテキストセクション
Security context セクションには、コンテナーが特権コンテナーとして実行されているかどうかがリストされます。

- **特権**:
  - **特権がある** 場合は **true**。
  - **特権がない** 場合は **false**。

ネットワークポリシーセクション
Network policy セクションには、namespace と、違反を含む namespace 内のすべてのネットワークポリシーがリストされます。ネットワークポリシー名をクリックして、ネットワークポリシーの完全な YAML ファイルを表示します。

15.2.3. ポリシータブ

Details パネルの **Policy** タブには、違反の原因となったポリシーの詳細が表示されます。

ポリシーの概要セクション
Policy overview セクションには、次の情報が一覧表示されます。

- **Severity**: 必要な注意の量に関するポリシーのランク付け (critical、high、medium、または low)。
- **Categories**: ポリシーのポリシーカテゴリー。ポリシーカテゴリーは、**Policy categories** タブの **Platform Configuration → Policy Management** にリストされます。
- **Type**: ポリシーがユーザー生成 (ユーザーによって作成されたポリシー) であるか、システムポリシー (デフォルトで RHACS に組み込まれているポリシー) であるか。
- **Description**: ポリシーアラートの内容の詳細な説明。
- **Rationale**: ポリシーの確立の背後にある理由と、それが重要である理由に関する情報。
- **Guidance**: 違反に対処する方法に関する提案。
- **MITRE ATT&CK**: このポリシーに適用される MITRE tactics and techniques があるかどうかを示します。

ポリシーの動作
Policy behavior セクションでは、次の情報を提供します。

- **ライフサイクルステージ**: ポリシーが属するライフサイクルステージ (**Build**、**Deploy**、または **Runtime**)。
- **Event source**: このフィールドは、ライフサイクルステージが **Runtime** の場合にのみ適用されます。次のいずれかです。
  - **Deployment**: イベントソースにプロセスおよびネットワークアクティビティー、Pod 実行、Pod ポート転送が含まれる場合、RHACS はポリシー違反をトリガーします。
  - **Audit logs**: イベントソースが Kubernetes 監査ログレコードと一致すると、RHACS はポリシー違反をトリガーします。

- **Response**: 応答は次のいずれかになります。
  - **Inform**: ポリシー違反では、違反リストに違反が生成されます。
  - **Inform and enforce**: 違反が適用されます。
- **Enforcement**: 応答が **Inform and enforce** に設定されている場合、次のステージに設定されている適用タイプがリストされます。
  - **Build**: イメージがポリシーの基準と一致すると、RHACS は継続的インテグレーション (CI) ビルドに失敗します。
  - **Deploy**: **Deploy** 段階では、RHACS アドミッションコントローラーが設定され実行されている場合、RHACS はポリシーの条件に一致するデプロイメントの作成と更新をブロックします。
    - アドミッションコントローラーが適用されているクラスターでは、Kubernetes または OpenShift Container Platform サーバーがすべての非準拠のデプロイメントをブロックします。他のクラスターでは、RHACS は準拠していないデプロイメントを編集して、Pod がスケジュールされないようにします。
    - 既存のデプロイメントの場合、ポリシーの変更は、Kubernetes イベントが発生したときに、基準が次に検出されたときにのみ適用されます。適用の詳細は、「デプロイステージのセキュリティーポリシーの適用」を参照してください。
  - **Runtime**: Pod 内のイベントがポリシーの基準に一致すると、RHACS はすべての Pod を削除します。

ポリシー条件セクション
Policy criteria セクションには、ポリシーのポリシー条件が一覧表示されます。

15.2.3.1. デプロイステージのセキュリティーポリシーの実施

Red Hat Advanced Cluster Security for Kubernetes は、デプロイ時のポリシーに対して、アドミッションコントローラーによるハードな適用と RHACS Sensor によるソフトな適用という 2 つの形式のセキュリティーポリシー適用をサポートしています。アドミッションコントローラーは、ポリシーに違反するデプロイメントの作成または更新をブロックします。アドミッションコントローラーが無効または使用できない場合、Sensor はポリシーに違反するデプロイメントのレプリカを **0** にスケールダウンして強制を実行できます。

警告

ポリシーの適用は、実行中のアプリケーションまたは開発プロセスに影響を与える可能性があります。適用オプションを有効にする前に、すべての利害関係者に通知し、自動適用アクションに対応する方法を計画してください。

15.2.3.1.1. ハードエンフォースメント

ハードエンフォースメントは、RHACS アドミッションコントローラーによって実行されます。アドミッションコントローラーが適用されているクラスターでは、Kubernetes または OpenShift Container Platform サーバーがすべての非準拠のデプロイメントをブロックします。アドミッションコントロー

ラーは、**CREATE** および **UPDATE** 操作をブロックします。デプロイ時の強制が有効に設定されたポリシーを満たす Pod の作成または更新リクエストはすべて失敗します。

注記

Kubernetes アドミッション Webhook は、**CREATE**、**UPDATE**、**DELETE**、または **CONNECT** 操作のみをサポートします。RHACS アドミッションコントローラーは、**CREATE** および **UPDATE** 操作のみをサポートします。**kubectl patch**、**kubectl set**、**kubectl scale** などの操作は、UPDATE 操作ではなく、PATCH 操作です。Kubernetes では PATCH 操作がサポートされていないため、RHACS は PATCH 操作の強制を実行できません。

ブロックを強制するには、RHACS でクラスターに対して次の設定を有効にする必要があります。

- **Enforce on Object Creates**: **Dynamic Configuration** セクションのこのトグルは、アドミッションコントロールサービスの動作を制御します。これを機能させるには、**Static Configuration** セクションの **Configure Admission Controller Webhook to listen on Object Creates** トグルをオンにする必要があります。
- **オブジェクトの更新時に強制**: **Dynamic Configuration** セクションのこのトグルは、アドミッションコントロールサービスの動作を制御します。これを機能させるには、**Static Configuration** セクションの **Configure Admission Controller Webhook to listen on Object Updates** トグルをオンにする必要があります。

Static Configuration 設定の項目を変更した場合に、その変更を有効にするにはセキュアクラスターを再デプロイする必要があります。

15.2.3.1.2. ソフトな適用

ソフトな適用は RHACS Sensor によって実行されます。このエンフォースメントにより、操作が開始しなくなります。ソフトな適用では、Sensor はレプリカを 0 にスケーリングし、Pod がスケジュールされるのを防ぎます。このエンフォースメントでは、クラスター内で準備ができていないデプロイメントが使用可能になります。

ソフトな適用が設定されていて、Sensor がダウンしている場合、RHACS は適用を実行できません。

15.2.3.1.3. namespace の除外

デフォルトでは、RHACS は、**stackrox**、**kube-system**、**istio-system** namespace などの特定の管理 namespace をエンフォースメントブロックから除外します。その理由は、RHACS が正しく機能するためには、これらの namespace 内の一部の項目をデプロイする必要があるためです。

15.2.3.1.4. 既存のデプロイメントへのエンフォースメント

既存のデプロイメントの場合、ポリシーの変更は、Kubernetes イベントが発生したときに、基準が次に検出されたときにのみ適用されます。ポリシーに変更を加えた場合は、**Policy Management** を選択し、**Reassess All** をクリックしてポリシーを再評価する必要があります。このアクションは、新しい受信 Kubernetes イベントがあるかどうかに関係なく、すべての既存のデプロイメントにデプロイポリシーを適用します。ポリシーに違反があった場合は、RHACS がエンフォースメントを実行します。

関連情報

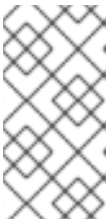
- アドミッションコントローラーの適用の使用

第16章 デプロイメントコレクションの作成と使用

RHACS のコレクションを使用して、マッチングパターンを使用してリソースのグループを定義し、名前を付けることができます。その後、これらのコレクションを使用するように、システムプロセスを設定できます。

現在、コレクションは次の条件下でのみ利用可能です。

- コレクションはデプロイメントでのみ使用できます。
- コレクションは、脆弱性レポートでのみ使用できます。詳細は、関連情報セクションの「脆弱性レポート」を参照してください。
- デプロイメントコレクションは、PostgreSQL データベースを使用している RHACS 顧客のみが利用できます。



注記

デフォルトでは、RHACS Cloud Service は PostgreSQL データベースを使用し、RHACS リリース 4.0 以降をインストールするときにもデフォルトで使用されます。3.74 より前のリリースを使用している RHACS のお客様は、Red Hat の支援を受けて PostgreSQL データベースに移行できます。

16.1. 前提条件

コレクション機能を使用するには、ユーザーアカウントに次の権限が必要です。

- **WorkflowAdministration**: コレクションを表示するには、**読み取り** アクセス権限が必要であり、コレクションを追加、変更、または削除するには、**書き込み** アクセス権限が必要です。
- **Deployment**: 設定されたルールがデプロイメントとどのように一致するかを理解するには、**読み取りアクセス** または **読み取りおよび書き込みアクセス** が必要です。

これらの権限は、**Admin** システムロールに含まれています。ロールとパーミッションの詳細は、「関連情報」の「RHACS での RBAC の管理」を参照してください。

16.2. デプロイメントコレクションについて

デプロイメントコレクションは、PostgreSQL データベースを使用する RHACS のお客様のみが利用できます。デフォルトでは、RHACS Cloud Service は PostgreSQL データベースを使用し、RHACS リリース 4.0 以降をインストールするときにもデフォルトで使用されます。3.74 より前のリリースを使用している RHACS のお客様は、Red Hat の支援を受けて PostgreSQL データベースに移行できます。

RHACS コレクションは、ユーザー定義の名前付き参照です。選択ルールを使用して、論理グループを定義します。これらのルールは、デプロイメント、namespace、またはクラスターの名前またはラベルに一致する可能性があります。完全一致または正規表現を使用してルールを指定できます。コレクションは実行時に解決され、コレクションの定義時には存在しないオブジェクトを参照できます。コレクションは、他のコレクションを使用して構築し、複雑な階層を記述することができます。

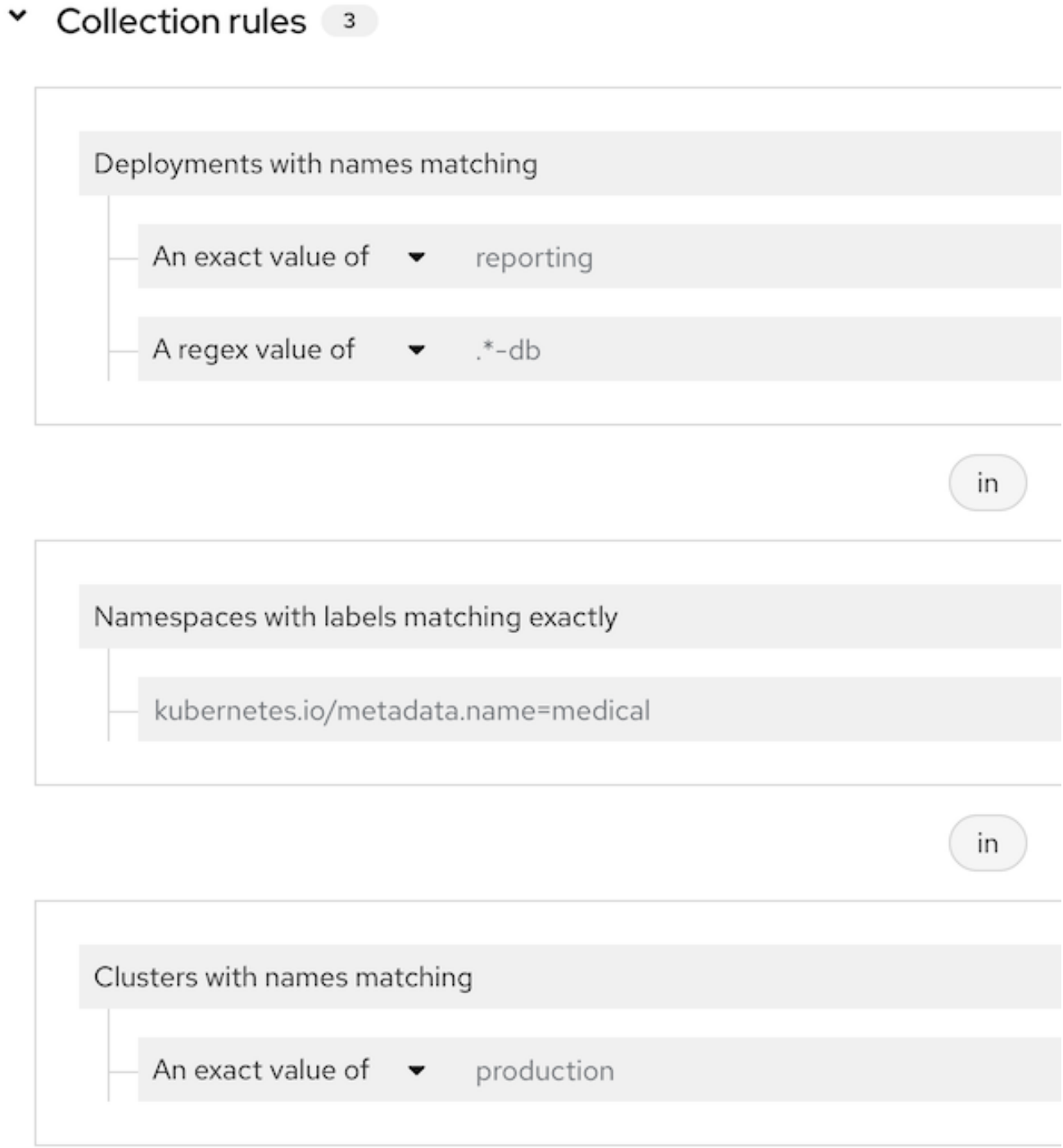
コレクションは、動的インフラストラクチャーがどのように編成されているかを説明する言語を提供し、包含および除外スコープなどの RHACS プロパティーのクローン作成および繰り返し編集の必要性を排除します。

コレクションを使用して、次のようなシステム内のデプロイメントのグループを識別できます。

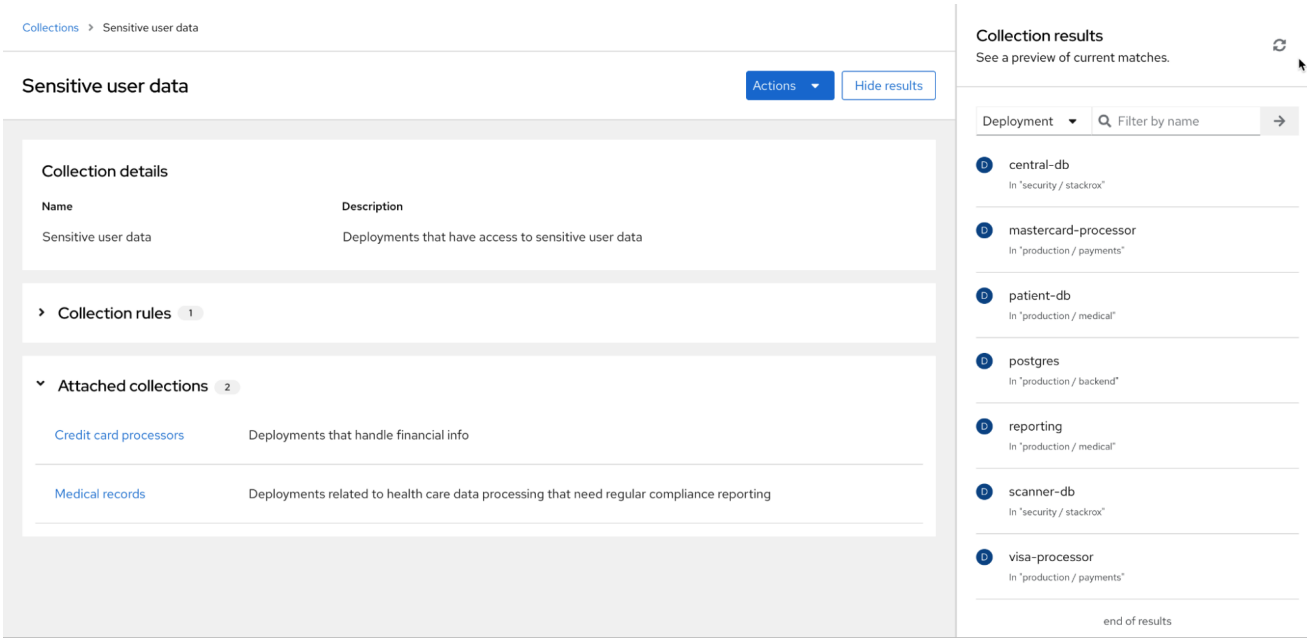
- 特定の開発チームが所有するインフラストラクチャー領域
- 開発クラスターまたは実稼働クラスターで実行するときに異なるポリシーの例外を必要とするアプリケーション
- 共通のデプロイメントラベルで定義された、複数の namespace にまたがる分散アプリケーション
- 実稼働環境またはテスト環境全体

コレクションは、RHACS ポータルを使用して、作成および管理できます。コレクションエディターは、デプロイメント、namespace、およびクラスターレベルで選択ルールを適用するのに役立ちます。正規表現を含む単純なルールと複雑なルールを使用できます。

次の図に示すように、1 つ以上のデプロイメント、namespace、またはクラスターを選択して、コレクションを定義できます。この図は、reporting という名前のデプロイメントを含むコレクション、または名前に **db** を含むコレクションを示しています。コレクションには、**kubernetes.io/metadata.name=medical** という特定のラベルが付いた namespace 内の名前に一致するデプロイメント、および **production** という名前のクラスター内のデプロイメントが含まれます。



コレクションエディターは、他のコレクションをアタッチまたはネストすることにより、複雑な階層を記述するのにも役立ちます。エディターにはリアルタイムプレビューサイドパネルがあり、設定したルールとの一致結果を表示することで、適用しているルールを理解するのに役立ちます。次の図は、一連のコレクションルール (表示されていません) を使用した「Sensitive User Data」という名前のコレクションからの結果の例を示しています。「Sensitive User Data」コレクションには、「Credit card processors」と「Medical records」という 2 つのコレクションが添付されており、これらのコレクションには、それぞれ独自のコレクションルールがあります。サイドパネルに表示される結果には、3 つのコレクションすべてに設定されたルールに一致するアイテムが含まれています。



16.3. デプロイメントコレクションへのアクセス

コレクションを使用するには、**Platform Configuration** → **Collections** をクリックします。このページには、現在設定されているコレクションのリストが表示されます。次のアクションを実行できます。

- **Search by name** フィールドにテキストを入力してコレクションを検索し、→ を押します。
- コレクションリスト内のコレクションをクリックして、コレクションを読み取り専用モードで表示します。
- 既存のコレクションの ⋮ をクリックして、編集、複製、または削除します。

> **注記**
>
> RHACS でアクティブに使用されているコレクションは削除できません。

- **Create collection** をクリックして、新しいデプロイメントコレクションを作成します。

16.4. デプロイメントコレクションの作成

コレクションを作成する場合は、コレクションに名前を付けて、コレクションのルールを定義する必要があります。

手順

1. Collections ページで、**Create collection** をクリックします。
2. コレクションの名前と説明を入力します。
3. **Collection rules** セクションで、次のアクションの 1 つ以上を実行する必要があります。
   - コレクションのルールを定義します。詳細は、「コレクションルールの作成」セクションを参照してください。

- 既存のコレクションをコレクションにアタッチします。詳細は、「アタッチされたコレクションの追加」セクションを参照してください。

4. ルールの設定またはアタッチされたコレクションの選択の結果は、**Collection results** ライブプレビューパネルで確認できます。このパネルを非表示にするには、**Hide results** をクリックします。

5. **Save** をクリックします。

16.4.1. コレクションルールの作成

コレクションを作成する場合は、1つ以上のルールを設定するか、作成する新しいコレクションに別のコレクションをアタッチする必要があります。

注記

現在、コレクションはデプロイメントでのみ使用できます。

コレクションに含めるリソースを選択するルールを設定します。プレビューパネルを使用して、設定したコレクションルールの結果を確認します。ルールは任意の順序で設定できます。

手順

1. **Deployments** セクションで、ドロップダウンリストから次のいずれかのオプションを選択します。

   - **All deployments**: コレクション内のすべてのデプロイメントが含まれます。このオプションを選択した場合は、namespace またはクラスターを使用するか、別のコレクションをアタッチして、コレクションをフィルタリングする必要があります。

   - **Deployments with names matching**: このオプションをクリックして、名前で選択し、次のいずれかのオプションをクリックします。

     - **An exact value of** を選択し、デプロイメントの正確な名前を入力します。

     - 正規表現を使用して、デプロイメントを検索するには、**A regex value of** を選択します。このオプションは、デプロイメントの正確な名前がわからない場合に役立ちます。正規表現は、パターンを定義する文字、数字、および記号の文字列です。RHACS は、このパターンを使用して、文字または文字グループを照合し、結果を返します。正規表現の詳細は、「関連情報」セクションの「Regular-Expressions.info」を参照してください。

   - **Deployments with labels matching exactly**: このオプションをクリックして、入力したテキストと正確に一致するラベルが付いたデプロイメントを選択します。ラベルは、**key=value** 形式の有効な Kubernetes ラベルにする必要があります。

2. オプション: 追加の包含条件に一致する名前またはラベルが付いたデプロイメントをさらに追加するには、**OR** をクリックして、別の正確な値または正規表現の値を設定します。

次の例は、医療アプリケーションのコレクションを設定する手順を示しています。この例では、コレクションに **reporting** デプロイメント、つまり **patient-db** というデータベースを含め、**key = kubernetes.io/metadata.name** および **value = medical** というラベルが付いた namespace を選択します。この例では、次の手順を実行します。

1. **Collection rules** で、**Deployments with names matching** を選択します。

2. **An exact value of** をクリックし、**reporting** と入力します。

3. **OR** をクリックします。

4. **A regex value of** をクリックし、**.*-db** と入力し、環境内で名前が **db** で終わるすべてのデプロイメントを選択します。**regex value** オプションは、パターンマッチングに正規表現を使用します。正規表現の詳細は、関連情報セクションの「Regular-Expressions.info」を参照してください。右側のパネルには、含めたくないデータベースが表示される場合があります。追加のフィルターを使用して、これらのデータベースを除外できます。以下に例を示します。

   a. **Namespaces with labels matching exactly** をクリックし、**kubernetes.io/metadata.name=medical** と入力して、namespace ラベルでフィルタリングして、**medical** というラベルが付いた namespace にデプロイメントのみを含めます。

   b. namespace の名前がわかっている場合は、**Namespaces with names matching** をクリックし、名前を入力します。

16.4.2. アタッチされたコレクションの追加

デプロイメントに基づいて、小さなコレクションを作成する場合は、コレクションをグループ化して、他のコレクションに追加すると、便利です。これらの小さなコレクションを再利用および結合し、より大きな階層コレクションにすることができます。作成中のコレクションにコレクションを追加するには:

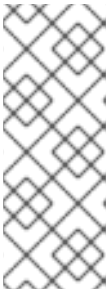
1. 次のいずれかの操作を実行します。

   - **Filter by name** フィールドにテキストを入力し、→ を押して、一致する結果を表示します。

   - **Available collections** リストからコレクションの名前をクリックして、コレクションの名前とルール、およびそのコレクションに一致するデプロイメントなど、コレクションに関する情報を表示します。

2. コレクション情報を表示したら、ウィンドウを閉じて、**Attached collections** ページに戻ります。

3. **+Attach** をクリックします。**Attached collections** セクションには、アタッチしたコレクションが一覧表示されます。



注記

アタッチされたコレクションを追加すると、アタッチされたコレクションには、設定された選択ルールに基づく結果が含まれます。たとえば、アタッチされたコレクションに、親コレクションで使用されるルールによって除外されるリソースが含まれている場合は、アタッチされたコレクションのルールにより、それらのアイテムは引き続き親コレクションに追加されます。アタッチされたコレクションは、**OR** 演算子を使用して、元のコレクションを拡張します。

4. **Save** をクリックします。

16.5. コレクションへのアクセススコープの移行

RHACS での **rocksdb** から PostgreSQL へのデータベースの変更は、リリース 3.74 以降テクノロジープレビューとして提供され、リリース 4.0 で一般提供されます。データベースが **rocksdb** から PostgreSQL に移行されると、脆弱性レポートで使用される既存のアクセススコープがコレクションに

移行されます。**Vulnerability Management → Reporting** に移動し、レポート情報を表示すると、移行によって既存のレポートが正しく設定されたことを確認できます。

移行プロセスでは、レポート設定で使用されたアクセススコープのコレクションオブジェクトが作成されます。RHACS は、アクセススコープの複雑さに応じて、1 つのアクセススコープに対して 2 つ以上のコレクションを生成します。特定のアクセススコープに対して生成されるコレクションには、次の種類があります。

- 組み込みコレクション: 元のアクセススコープの正確な選択ロジックを模倣するために、RHACS は 1 つ以上のコレクションを生成します。このコレクションでは、デプロイメントが一致すると、元のアクセススコープと同じクラスターと namespace が選択されます。コレクション名の形式は、**System-generated embedded collection number for the scope** であり、**number** は、0 から始まる番号です。

注記

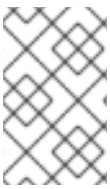
これらの埋め込みコレクションには、アタッチされたコレクションはありません。クラスターと namespace の選択ルールはありますが、元のアクセススコープがデプロイメントをフィルタリングしなかったため、デプロイメントルールはありません。

- アクセススコープのルートコレクション: このコレクションは、レポート設定に追加されます。コレクション名の形式は、**System-generated root collection for the scope** です。このコレクションはルールを定義しませんが、1 つ以上の埋め込みコレクションをアタッチします。これらの埋め込みコレクションを組み合わせると、元のアクセススコープと同じクラスターと namespace が選択されます。

クラスターまたは namespace のラベルセレクターを定義するアクセススコープの場合、RHACS は、キーと値の間に 'IN' 演算子があるスコープのみを移行できます。RHACS ポータルで作成されたラベルセレクターを含むアクセススコープでは、デフォルトで 'IN' 演算子が使用されていました。'NOT_IN'、'EXISTS'、および 'NOT_EXISTS' 演算子を使用したスコープの移行はサポートされていません。アクセススコープのコレクションを作成できない場合は、移行中にログメッセージが作成されます。ログメッセージの形式は次のとおりです。

> Failed to create collections for scope _scope-name_: Unsupported operator NOT_IN in scope's label selectors. Only operator 'IN' is supported.
> The scope is attached to the following report configurations: [list of report configs]; Please manually create an equivalent collection and edit the listed report configurations to use this collection. Note that reports will not function correctly until a collection is attached.

Vulnerability Management → Reporting でレポートをクリックして、レポート情報ページを表示することもできます。このページには、レポートにコレクションをアタッチする必要がある場合のメッセージが含まれています。



注記

移行中は、元のアクセススコープは削除されません。脆弱性管理レポートのフィルタリングにのみ使用するアクセススコープを作成した場合は、アクセススコープを手動で削除できます。

16.6. API を使用したコレクションの管理

CollectionService API オブジェクトを使用して、コレクションを設定できます。たとえば、**CollectionService_DryRunCollection** を使用して、RHACS ポータルのライブプレビューパネルに

相当する結果のリストを返すことができます。詳細は、RHACS ポータルの **Help → API reference** に移動してください。

関連情報

- RHACS での RBAC の管理
- 脆弱性レポート
- 正規表現の使用: Regular-Expressions.info

第17章 検索およびフィルタリング

クラスターを保護するには、リソースを即座に見つける機能が重要です。Red Hat Advanced Cluster Security for Kubernetes 検索機能を使用して、関連するリソースをより迅速に検索します。たとえば、これを使用して、新しく公開された CVE に公開されているデプロイメントを検索したり、外部ネットワークに公開されているすべてのデプロイメントを検索したりできます。

17.1. 検索構文

検索クエリーは、次の 2 つの部分で構成されています。

- 検索するリソースタイプを識別する属性。
- 一致するリソースを見つける検索用語。

たとえば、**visa-processor** のデプロイメントですべての違反を見つけるには、検索クエリーは **Deployment:visa-processor** です。この検索クエリーでは、**Deployment** が属性であり、**visa-processor** が検索語です。

注記

検索語を使用する前に、属性を選択する必要があります。ただし、**Risk** ビューや **Violations** ビューなどの一部のビューでは、Red Hat Advanced Cluster Security for Kubernetes は、入力した検索語に基づいて関連する属性を自動的に適用します。

- クエリーでは複数の属性を使用できます。複数の属性を使用する場合、結果にはすべての属性に一致するアイテムのみが含まれます。

  例

  Namespace:frontend CVE:CVE-2018-11776 を検索すると、**frontend** namespace の CVE-2018-11776 に違反するリソースのみが返されます。

- 各属性で複数の検索語を使用できます。複数の検索語を使用すると、結果には、いずれかの検索語に一致するすべてのアイテムが含まれます。

  例

  検索クエリー **Namespace: frontend backend** を使用すると、**frontend** または **backend** namespace から一致する結果が返されます。

- 複数の属性と検索語のペアを組み合わせることができます。

  例

  検索クエリー **Cluster:production Namespace:frontend CVE:CVE-2018-11776** では、**production** クラスターの **frontend** namespace の CVE-2018-11776 に違反するすべてのリソースが返されます。

- 検索語は単語の一部にすることができます。その場合、Red Hat Advanced Cluster Security for Kubernetes は一致するすべての結果を返します。

  例

  Deployment:def を検索すると、結果には **def** で始まるすべてのデプロイメントが含まれます。

- 特定の用語を明示的に検索するには、引用符で囲まれた検索用語を使用します。

  例

  Deployment:"def" を検索すると、結果にはデプロイメント **def** のみが含まれます。

- 検索語の前に **r/** を使用して、正規表現を使用することもできます。

  例

  Namespace:r/st.*x を検索すると、**stackrox** および **stix** namespace からの一致が結果に表示されます。

- **!** を使用すると、結果に表示したくない検索用語を示します。

  例

  Namespace:!stackrox を検索すると、**stackrox** namespace を除くすべての namespace からの一致が結果に表示されます。

- 比較演算子 **>**、**<**、**=**、**>=**、または **<=** を使用して、特定の値または値の範囲を一致させます。

  例

  CVSS:>=6 を検索すると、結果には、Common Vulnerability Scoring System (CVSS) スコアが 6 以上のすべての脆弱性が含まれます。

17.2. オートコンプリートの検索

クエリーを入力すると、Red Hat Advanced Cluster Security for Kubernetes は属性および検索語に関連する提案を自動的に表示します。

17.3. グローバル検索の使用

グローバル検索を使用すると、環境内のすべてのリソースを検索できます。検索クエリーで使用するリソースタイプに基づいて、結果は次のカテゴリーにグループ化されます。

- すべての結果 (すべてのカテゴリーで一致する結果を一覧表示)
- クラスター
- デプロイメント
- イメージ
- namespace
- ノード
- ポリシー
- ポリシーカテゴリー [1]
- ロール
- ロールバインディング

- シークレット
- Service accounts
- ユーザーおよびグループ
- Violations

1. **Policy categories** オプションは、次を使用する場合にのみ使用できます。
    - PostgreSQL を Red Hat Advanced Cluster Security for Kubernetes (RHACS) のバックエンドデータベースとして使用する場合
    - Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service)

これらのカテゴリーは、RHACS ポータルのグローバル検索ページに表としてリスト表示されます。カテゴリー名をクリックすると、選択したカテゴリーに属する結果を識別できます。

グローバル検索を実行するには、RHACS ポータルで **Search** を選択します。

17.4. ローカルページのフィルタリングの使用

RHACS ポータルのすべてのビュー内からローカルページのフィルタリングを使用できます。ローカルページフィルタリングはグローバル検索と同様に機能しますが、関連する属性のみが使用可能です。検索バーを選択して、特定のビューで使用可能なすべての属性を表示できます。

17.5. 一般的な検索クエリー

Red Hat Advanced Cluster Security for Kubernetes で実行できる一般的な検索クエリーを次に示します。

特定の CVE の影響を受けるデプロイメントの検索

クエリー	例
CVE:<CVE_number>	**CVE:CVE-2018-11776**

特権のある実行中のデプロイメントの検索

クエリー	例
Privileged:<true_or_false>	**Privileged:true**

外部ネットワークにさらされているデプロイメントの検索

クエリー	例
Exposure Level:<level>	**Exposure Level:External**

特定のプロセスを実行しているデプロイメントの検索

クエリー	例
Process Name:<process_name>	**Process Name:bash**

深刻であるが修正可能な脆弱性があるデプロイメントの検索

クエリー	例
CVSS:<expression_and_score>	**CVSS:>=6 Fixable:.***

環境変数を介して公開されたパスワードを使用するデプロイメントの検索

クエリー	例
Environment Key:<query>	**Environment Key:r/.*pass.***

特定のソフトウェアコンポーネントが含まれている実行中のデプロイメントの検索

クエリー	例
Component:<component_name>	**Component:libgpg-error** または **Component:sudo**

ユーザーまたはグループの検索

Kubernetes の ラベルおよびセレクター、ならびに アノテーション を使用して、メタデータをデプロイメントにアタッチします。次に、適用されたアノテーションおよびラベルに基づいてクエリーを実行し、個人またはグループを識別できます。

特定のデプロイメントを所有しているユーザーの検索

クエリー	例
Deployment:<deployment_name> Label:<key_value> または **Deployment:<deployment_name> Annotation:<key_value>**	**Deployment:app-server Label:team=backend**

パブリックレジストリーからイメージをデプロイしているユーザーの検索

クエリー	例
Image Registry:<registry_name> Label:<key_value> または **Image Registry:<registry_name> Annotation:<key_value>**	**Image Registry:docker.io Label:team=backend**

デフォルトの namespace にデプロイしているユーザーの検索

クエリー	例
Namespace:default Label:<key_value> または **Namespace:default Annotation:<key_value>**	**Namespace:default Label:team=backend**

17.6. 属性の検索

以下は、Red Hat Advanced Cluster Security for Kubernetes での検索およびフィルタリング中に使用できる検索属性のリストです。

属性	説明
Add Capabilities	コンテナーに追加の Linux 機能を提供します。たとえば、ファイルを変更したり、ネットワーク操作を実行したりする機能です。
Annotation	オーケストレーターオブジェクトに添付された任意の非識別メタデータ。
CPU Cores Limit	リソースが使用できるコアの最大数。
CPU Cores Request	特定のリソース用に予約されるコアの最小数。
CVE	Common Vulnerabilities and Exposures。特定の CVE 番号で使用。
CVSS	一般的な脆弱性スコアリングシステム。CVSS スコアより大なり (>)、より小なり (<)、または等号 (=) 記号で使用します。
Category	ポリシーカテゴリーには、DevOps のベストプラクティス、セキュリティーのベストプラクティス、特権、脆弱性管理、複数、および作成したカスタムポリシーカテゴリーが含まれます。
Cert Expiration	証明書の有効期限。
Cluster	Kubernetes または OpenShift Container Platform クラスターの名前。
Cluster ID	Kubernetes または OpenShift Container Platform クラスターの一意の ID。
Cluster Role	クラスター全体のロールを検索する場合は **true** を使用し、namespace スコープのロールを検索する場合は **false** を使用します。
Component	ソフトウェア (daemond、docker)、オブジェクト (イメージ、コンテナー、サービス)、レジストリー (Docker イメージのリポジトリー)。
Component Count	イメージ内のコンポーネントの数。
Component version	ソフトウェア、オブジェクト、またはレジストリーのバージョン。
Created Time	シークレットオブジェクトが作成された日時。

属性	説明
Deployment	デプロイメントの名前。
Deployment Type	デプロイのベースとなる Kubernetes コントローラーのタイプ。
説明	デプロイメントの説明。
Dockerfile Instruction Keyword	イメージ内の Dockerfile 命令のキーワード。
Dockerfile Instruction Value	イメージ内の Dockerfile 命令の値。
Drop Capabilities	コンテナーから削除された Linux 機能。たとえば、**CAP_SETUID** または **CAP_NET_RAW** です。
Enforcement	デプロイメントに割り当てられた強制のタイプ。たとえば、**None**、**Scale to Zero Replicas**、**Add an Unsatisfiable Node Constraint** などです。
Environment Key	コンテナーの環境をさらに識別および整理するためのメタデータである、ラベルのキー値文字列のキー部分。
Environment Value	コンテナーの環境をさらに識別および整理するためのメタデータであるラベルキー値文字列の値部分。
Exposed Node Port	公開されたノードポートのポート番号。
Exposing Service	公開されたサービスの名前。
Exposing Service Port	公開されたサービスのポート番号。
Exposure Level	**external**、**node** など、デプロイメントポートの公開のタイプ。
External Hostname	デプロイメントの外部ポート公開のホスト名。
External IP	デプロイメントの外部ポート公開の IP アドレス。
Fixable CVE Count	イメージ上の修正可能な CVE の数。
Fixed By	イメージのフラグ付きの脆弱性を修正するパッケージのバージョン文字列。
Image	イメージの名前。
Image Command	イメージで指定されているコマンド。
Image Created Time	イメージが作成された日時。

属性	説明
Image Entrypoint	イメージで指定されているエントリーポイントコマンド。
Image Pull Secret	デプロイメントで指定されている、イメージをプルするときに使用するシークレットの名前。
Image Pull Secret Registry	イメージプルシークレットのレジストリーの名前。
Image Registry	イメージレジストリーの名前。
Image Remote	リモートアクセス可能なイメージの表示。
Image Scan Time	イメージが最後にスキャンされた日時。
Image Tag	イメージの識別子。
Image Users	コンテナーイメージの実行時に使用するように設定されているユーザーまたはグループの名前。
Image Volumes	コンテナーイメージで設定されたボリュームの名前。
Inactive Deployment	非アクティブなデプロイメントを検索するには **true** を使用し、アクティブなデプロイメントを検索するには **false** を使用します。
Label	イメージ、コンテナー、デーモン、ボリューム、ネットワーク、およびその他のリソースをさらに識別および整理するためのメタデータである、ラベルのキー値文字列のキー部分。
Lifecycle Stage	このポリシーが設定されている、またはアラートがトリガーされたライフサイクルステージのタイプ。
Max Exposure Level	デプロイメントの場合は、特定のすべてのポート/サービスのネットワーク公開の最大レベル。
Memory Limit (MB)	リソースが使用できるメモリーの最大量。
Memory Request (MB)	特定のリソース用に予約されるメモリーの最小量。
Namespace	namespace の名前。
Namespace ID	デプロイメントに含まれる namespace オブジェクトの一意の ID。
Node	ノードの名前。
Node ID	ノードの一意の ID。

属性	説明
Pod Label	個別の Pod に添付された単一の識別メタデータ。
Policy	セキュリティーポリシーの名前。
Port	デプロイメントによって公開されるポート番号。
Port Protocol	公開されたポートで使用される TCP や UDP などの IP プロトコル。
Priority	デプロイメントのリスク優先度。**Risks** ビューでのみ使用可能)
Privileged	特権のある稼働中のデプロイメントを検索するには **true** を使用し、それ以外の場合は **false** を使用します。
Process Ancestor	デプロイメント内のプロセスインジケーターの親プロセスの名前。
Process Arguments	デプロイメント内のプロセスインジケーターのコマンド引数。
プロセス名	デプロイメント内のプロセスインジケーターのプロセスの名前。
Process Path	デプロイメントのプロセスインジケーターのコンテナー内のバイナリーへのパス。
Process UID	デプロイメントのプロセスインジケーターの Unix ユーザー ID。
Read Only Root Filesystem	**true** を使用して、読み取り専用として設定されたルートファイルシステムで実行しているコンテナーを検索します。
Role	Kubernetes RBAC ロールの名前。
Role Binding	Kubernetes RBAC ロールバインディングの名前。
Role ID	Kubernetes RBAC ロールバインディングがバインドされているロール ID。
Secret	機密情報を保持する秘密オブジェクトの名前。
Secret Path	ファイルシステム内のシークレットオブジェクトへのパス。
Secret Type	シークレットのタイプ (証明書や RSA 公開鍵など)。
Service Account	サービスアカウントまたはデプロイメントのサービスアカウント名。
Severity	違反の重要度の表示: Critical、High、Medium、Low。
Subject	Kubernetes RBAC でのサブジェクトの名前。

属性	説明
Subject Kind	**SERVICE_ACCOUNT**、**USER**、**GROUP** などの Kubernetes RBAC のサブジェクトのタイプ。
Taint Effect	現在ノードに適用されているテイントのタイプ。
Taint Key	現在ノードに適用されているテイントのキー。
Taint Value	現在ノードに適用されているテイントの許容値。
Toleration Key	デプロイメントに適用される許容範囲のキー。
Toleration Value	デプロイメントに適用される許容値の値。
Violation	ポリシーで指定された条件が満たされない場合に **Violations** ページに表示される通知。
Violation State	解決された違反を検索するのに使用します。
Violation Time	違反が最初に発生した日時。
Volume Destination	データボリュームのマウントパス。
Volume Name	ストレージの名前。
Volume ReadOnly	**true** を使用して、読み取り専用としてマウントされているボリュームを検索します。
Volume Source	ボリュームがプロビジョニングされる形式を示します (例: **persistentVolumeClaim** または **hostPath**)。
Volume Type	ボリュームの種別を設定します。

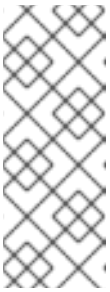
第18章 ユーザーアクセスの管理

18.1. RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES での RBAC の管理

Red Hat Advanced Cluster Security for Kubernetes (RHACS) には、ロールベースのアクセス制御 (RBAC) が組み込まれています。RBAC を使用すると、ロールを設定し、Red Hat Advanced Cluster Security for Kubernetes へのさまざまなレベルのアクセス権をさまざまなユーザーに付与できます。

RHACS にはバージョン 3.63 以降、特定の RHACS ユーザーまたはユーザーグループが RHACS と対話する方法、アクセス可能なリソース、実行できるアクションを定義するきめ細かい特定の権限のセットを設定できるスコープ付きアクセス制御機能が含まれています。

- **ロール** は、権限セットとアクセススコープの集まりです。ルールを指定することにより、ユーザーおよびグループにロールを割り当てることができます。これらのルールは、認証プロバイダーを設定するときに設定できます。Red Hat Advanced Cluster Security for Kubernetes には 2 つのタイプのロールがあります。
    - Red Hat によって作成され、変更できないシステムロール。
    - Red Hat Advanced Cluster Security for Kubernetes 管理者がいつでも作成および変更できるカスタムロール。



注記

- ユーザーに複数のロールを割り当てると、割り当てられたロールの組み合わせた権限にアクセスできます。
- カスタムロールにユーザーが割り当てられていて、そのロールを削除すると、関連付けられているすべてのユーザーが、設定した最小アクセスロールに転送されます。

- **アクセス許可セット** は、特定のリソースに対してロールが実行できるアクションを定義する権限のセットです。**リソース** は、Red Hat Advanced Cluster Security for Kubernetes の機能であり、表示 (**read**) および変更 (**write**) 権限を設定できます。Red Hat Advanced Cluster Security for Kubernetes には、次の 2 種類の権限セットがあります。
    - Red Hat によって作成され、変更できないシステム権限セット。
    - Red Hat Advanced Cluster Security for Kubernetes 管理者がいつでも作成および変更できるカスタム権限セット。
- **アクセススコープ** は、ユーザーがアクセスできる Kubernetes および OpenShift Container Platform リソースのセットです。たとえば、ユーザーが特定のプロジェクトの Pod に関する情報にのみアクセスできるようにするアクセススコープを定義できます。Red Hat Advanced Cluster Security for Kubernetes には、次の 2 種類のアクセススコープがあります。
    - Red Hat により作成され、変更できないシステムアクセススコープ。
    - Red Hat Advanced Cluster Security for Kubernetes 管理者がいつでも作成および変更できるカスタムアクセススコープ。

18.1.1. システムロール

Red Hat Advanced Cluster Security for Kubernetes (RHACS) には、ルールの作成時にユーザーに適用できるデフォルトのシステムロールがいくつか用意されています。必要に応じて、カスタムロールを作成することもできます。

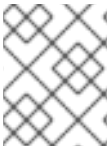
システムロール	説明
Admin	このロールは管理者を対象としています。これを使用して、すべてのリソースへの読み取りおよび書き込みアクセスを提供します。
Analyst	このロールは、変更を加えることはできないが、すべてを表示できるユーザーを対象としています。これを使用して、すべてのリソースに読み取り専用アクセスを提供します。
Continuous Integration	このロールは、CI (継続的インテグレーション) システムを対象としており、デプロイメントポリシーを適用するのに必要なアクセス許可セットが含まれています。
None	このロールには、リソースへの読み取りおよび書き込みアクセス権がありません。このロールを、すべてのユーザーの最小アクセスロールとして設定できます。
Sensor Creator	RHACS はこのロールを使用して、新しいクラスターのセットアップを自動化します。このロールには、セキュアクラスターに Sensor を作成する権限セットが含まれています。
Scope Manager	このロールには、アクセススコープの作成および変更に必要な最小限の権限が含まれます。
Vulnerability Management Approver	このロールを使用すると、脆弱性の延期または誤検出の要求を承認するためのアクセス権を付与できます。
Vulnerability Management Requester	このロールを使用すると、脆弱性の延期または誤検出を要求するためのアクセス権を付与できます。
Vulnerability Report Creator	このロールを使用すると、スケジュールされた脆弱性レポートの脆弱性レポート設定を作成および管理できます。

18.1.1.1. システムロールの権限セットおよびアクセス範囲の表示

デフォルトのシステムロールの権限セットおよびアクセス範囲を表示できます。

手順

1. RHACS ポータルで、**Platform Configuration → Access control** に移動します。
2. **Roles** を選択します。
3. ロールの 1 つをクリックして、その詳細を表示します。詳細ページには、選択されたロールの権限セットおよびアクセス範囲が表示されます。



注記

デフォルトのシステムロールの権限セットおよびアクセス範囲を変更することはできません。

18.1.1.2. カスタムロールの作成

アクセス制御 ビューから新しいロールを作成できます。

前提条件

- カスタムロールを作成、変更、および削除するには、**Admin** ロール、または **AuthProvider** および **Role** リソースの読み取りおよび書き込み権限を持つロールが必要です。
- ロールを作成する前に、カスタムロールの権限セットおよびアクセススコープを作成する必要があります。

手順

1. RHACS ポータルで、**Platform Configuration** → **Access Control** に移動します。
2. **Roles** を選択します。
3. **Create role** をクリックします。
4. 新しいロールの **Name** および **Description** を入力します。
5. ロールの **権限セット** を選択します。
6. ロールの **アクセススコープ** を選択します。
7. **Save** をクリックします。

関連情報

- カスタム権限セットの作成
- カスタムアクセススコープの作成

18.1.1.3. ユーザーまたはグループへのロールの割り当て

RHACS ポータルを使用して、ユーザーまたはグループにロールを割り当てることができます。

手順

1. RHACS ポータルで、**Platform Configuration** → **Access Control** に移動します。
2. 認証プロバイダーのリストから、認証プロバイダーを選択します。
3. **Edit minimum role and rules**をクリックします。
4. **Rules** セクションで、**Add new rule** をクリックします。
5. **Key** で、**userid**、**name**、**email**、または **group** から 1 つ選択します。

6. **Value** に、選択したキーに基づいたユーザー ID、名前、メールアドレス、またはグループの値を入力します。
7. **Role** ドロップダウンメニューをクリックして、割り当てるロールを選択します。
8. **Save** をクリックします。

ユーザーまたはグループごとにこれらの手順を繰り返し、異なるロールを割り当てることができます。

18.1.2. システム権限セット

Red Hat Advanced Cluster Security for Kubernetes には、ロールに適用できるデフォルトのシステム権限セットがいくつか含まれています。必要に応じて、カスタム権限セットを作成することもできます。

パーミッションセット	説明
Admin	すべてのリソースへの読み取りおよび書き込みアクセスを提供します。
Analyst	すべてのリソースに読み取り専用アクセスを提供します。
Continuous Integration	このアクセス許可セットは、CI (継続的インテグレーション) システムを対象としており、デプロイメントポリシーを適用するのに必要なアクセス許可が含まれています。
Network Graph Viewer	ネットワークグラフを表示するための最小限の権限を提供します。
None	どのリソースにも読み取りおよび書き込み権限は許可されていません。
Sensor Creator	セキュアクラスターに Sensor を作成するのに必要なリソースの権限を提供します。

18.1.2.1. システム権限セットの権限の表示

RHACS ポータルで設定されたシステム権限の権限を表示できます。

手順

1. RHACS ポータルで、**Platform Configuration → Access control** に移動します。
2. **Permission sets** を選択します。
3. 権限セットの 1 つをクリックして、その詳細を表示します。詳細ページには、選択した権限セットに対するリソースおよびその権限のリストが表示されます。

注記

システム権限セットの権限を変更することはできません。

18.1.2.2. カスタム権限セットの作成

Access Control ビューから新しいアクセス許可セットを作成できます。

前提条件

- **管理者** ロール、または権限セットを作成、変更、および削除するには、**AuthProvider** リソースおよび **Role** リソースの読み取りおよび書き込み権限を持つ権限セットを持つロールが必要です。

手順

1. RHACS ポータルで、**Platform Configuration** → **Access Control** に移動します。
2. **Permission sets** を選択します。
3. **Create permission set** をクリックします。
4. 新しい権限セットの **Name** および **Description** を入力します。
5. リソースごとに、**Access level** 列で、**No access**、**Read access**、または、**Read and Write access** のいずれかのアクセス許可を選択します。

警告

- ユーザーに権限セットを設定する場合は、次のリソースに読み取り専用の権限を付与する必要があります。
  - **Alert**
  - **Cluster**
  - **Deployment**
  - **Image**
  - **NetworkPolicy**
  - **NetworkGraph**
  - **WorkflowAdministration**
  - **Secret**
- これらの権限は、新しい権限セットを作成するときに事前に選択されています。
- これらの権限を付与しない場合、ユーザーは RHACS ポータルでページを表示する際に問題が発生します。

6. **Save** をクリックします。

18.1.3. システムアクセススコープ

Red Hat Advanced Cluster Security for Kubernetes には、ロールに適用できるデフォルトのシステムアクセススコープがいくつか含まれています。必要に応じて、カスタムアクセススコープを作成することもできます。

アクセススコープ	説明
Unrestricted	Red Hat Advanced Cluster Security for Kubernetes が監視するすべてのクラスターと namespace へのアクセスを提供します。
Deny All	Kubernetes および OpenShift Container Platform リソースへのアクセスを提供しません。

18.1.3.1. システムアクセススコープの詳細の表示

RHACS ポータルで、アクセススコープで許可されているまたは許可されていない Kubernetes および OpenShift Container Platform リソースを表示できます。

手順

1. RHACS ポータルで、**Platform Configuration → Access control** に移動します。
2. **Access scopes** を選択します。
3. アクセススコープの 1 つをクリックして、その詳細を表示します。詳細ページには、クラスターと namespace のリストと、選択したアクセススコープで許可されているクラスターと namespace が表示されます。

注記

システムアクセススコープに許可されているリソースを変更することはできません。

18.1.3.2. カスタムアクセススコープの作成

アクセス制御 ビューから新しいアクセススコープを作成できます。

前提条件

- **管理者** ロール、または権限セットを作成、変更、および削除するには、**AuthProvider** リソースおよび **Role** リソースの読み取りおよび書き込み権限を持つ権限セットを持つロールが必要です。

手順

1. RHACS ポータルで、**Platform Configuration → Access control** に移動します。
2. **Access scopes** を選択します。
3. **Create access scope** をクリックします。
4. 新しいアクセススコープの **名前** と **説明** を入力します。
5. **Allowed resources** セクションの下で、以下を行います。

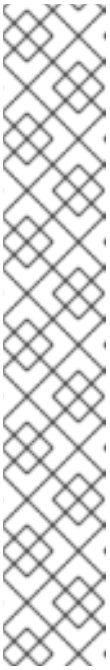
- **Cluster filter** および **Namespace filter** フィールドを使用して、リストに表示されるクラスターと namespace のリストをフィルタリングします。
- **Cluster name** を展開して、そのクラスター内の namespace の一覧を表示します。
- クラスター内のすべての namespace へのアクセスを許可するには、**Manual selection** 列のスイッチを切り替えます。

注記

特定のクラスターへのアクセスにより、ユーザーはクラスターのスコープ内の次のリソースにアクセスできます。

- OpenShift Container Platform または Kubernetes クラスターのメタデータおよびセキュリティー情報
- 許可されたクラスターのコンプライアンス情報
- ノードのメタデータおよびセキュリティー情報
- 該当するクラスター内のすべての namespace とそれに関連するセキュリティー情報へのアクセス

- namespace へのアクセスを許可するには、namespace の **Manual selection** 列でスイッチを切り替えます。



注記

特定の namespace にアクセスすると、namespace のスコープ内で次の情報にアクセスできます。

- デプロイメントに関するアラートおよび違反
- イメージの脆弱性データ
- デプロイメントメタデータおよびセキュリティー情報
- ロールおよびユーザー情報
- デプロイメントのネットワークグラフ、ポリシー、およびベースライン情報
- プロセス情報およびプロセスベースライン設定
- 各デプロイメントの優先リスク情報

6. ラベルに基づいてクラスターおよび namespace へのアクセスを許可する場合は、**Label selection rules** セクションの **Add label selector** をクリックします。次に、**Add rule** をクリックして、ラベルセレクターの **キー** と **値** のペアを指定します。クラスターおよび namespace のラベルを指定できます。
7. **Save** をクリックします。

18.1.4. リソース定義

Red Hat Advanced Cluster Security for Kubernetes には、複数のリソースが含まれています。次の表に、リソースと、ユーザーが **read** または **write** 権限で実行できるアクションを示します。

リソース	読み取り権限	書き込み権限
アクセス	認証プロバイダーが提供する認証プロバイダーに関するメタデータなど、ユーザーメタデータを Red Hat Advanced Cluster Security for Kubernetes ロールおよび Red Hat Advanced Cluster Security for Kubernetes インスタンスにアクセスしたユーザーと照合する Single Sign-On (SSO) およびロールベースのアクセス制御 (RBAC) ルールの設定を表示します。	SSO 設定および設定された RBAC ルールを作成、変更、または削除します。
管理	次の項目を表示します。 ● データ保持、セキュリティー通知、その他の関連設定のオプション ● Red Hat Advanced Cluster Security for Kubernetes コンポーネントの現在のログの詳細レベル ● アップロードされたプローブファイルのマニフェストコンテンツ ● 既存のイメージスキャナーの統合 ● 自動アップグレードのステータス ● Red Hat Advanced Cluster Security for Kubernetes のサービス間認証に関するメタデータ ● スキャナバンドル (ダウンロード) の内容	次の項目を編集します。 ● データ保持、セキュリティーに関する通知、および関連する設定 ● ログレベル ● Central でのサポートパッケージ (アップロード) ● イメージスキャナの統合 (作成/変更/削除) ● セキュアクラスターの自動アップグレード (有効化/無効化) ● サービス間認証認証情報 (取り消し/再発行)
アラート	既存のポリシー違反を表示します。	ポリシー違反を解決または編集します。
CVE	**内部でのみ使用**	**内部でのみ使用**
Cluster	既存のセキュアクラスターを表示します。	新しいセキュアクラスターを追加し、既存のクラスターを変更または削除します。
コンプライアンス	コンプライアンスの基準と結果、最近のコンプライアンスの実行と関連する完了ステータスを表示します。	コンプライアンスの実行をトリガーします。

リソース	読み取り権限	書き込み権限
Deployment	セキュアクラスター内のデプロイメント (ワークロード) を表示します。	該当なし
DeploymentExtension	次の項目を表示します。 ● プロセスベースライン ● デプロイメントにおけるプロセスアクティビティー ● リスク結果	次の項目を変更します。 ● プロセスのベースライン (プロセスの追加または削除)
Detection	イメージまたはデプロイメント YAML のビルド時ポリシーを確認します。	該当なし
イメージ	イメージ、そのコンポーネント、およびそれらの脆弱性を表示します。	該当なし
インテグレーション	次の項目を表示します。 ● 既存の API トークン ● Amazon Web Services (AWS) S3 などの自動バックアップシステムとの既存の統合 ● 既存のイメージレジストリーの統合 ● メール、Jira、Webhook などの通知システムの既存のインテグレーション	次の項目を変更します。 ● API トークン (新しいトークンの作成または既存のトークンの取り消し) ● バックアップ統合の設定 ● イメージレジストリーの統合 (作成/編集/削除) ● 通知の統合 (作成/編集/削除)
K8sRole	セキュアクラスター内の Kubernetes RBAC のロールを表示します。	該当なし
K8sRoleBinding	セキュアクラスター内の Kubernetes RBAC のロールバインディングを表示します。	該当なし
K8sSubject	セキュアクラスター内の Kubernetes RBAC のユーザーとグループを表示します。	該当なし
Namespace	セキュアクラスター内の既存の Kubernetes namespace を表示します。	該当なし

リソース	読み取り権限	書き込み権限
NetworkGraph	セキュアクラスター内のアクティブで許可されたネットワーク接続を表示します。	該当なし
NetworkPolicy	セキュアクラスター内の既存のネットワークポリシーを表示し、変更をシミュレートします。	セキュアクラスターにネットワークポリシーの変更を適用します。
Node	セキュアクラスター内の既存の Kubernetes ノードを表示します。	該当なし
WorkflowAdministration	すべてのリソースコレクションを表示します。	リソースコレクションを追加、変更、または削除します。
ロール	既存の Red Hat Advanced Cluster Security for Kubernetes RBAC ロールおよびその権限を表示します。	ロールおよびその権限を追加、変更、または削除します。
Secret	セキュアクラスター内のシークレットに関するメタデータを表示します。	該当なし
ServiceAccount	セキュアクラスター内の Kubernetes サービスアカウントをリスト表示します。	該当なし

18.1.5. 認証および承認リソースの宣言型設定

認証プロバイダー、ロール、パーミッションセット、アクセススコープなどの認証および承認リソースに宣言型設定を使用できます。宣言型設定の使用方法は、「関連情報」の「宣言型設定の使用」を参照してください。

関連情報

- 宣言型設定の使用

18.2. PKI 認証の有効化

認証にエンタープライズ認証局 (CA) を使用する場合は、Red Hat Advanced Cluster Security for Kubernetes (RHACS) を設定して、ユーザーの個人証明書を使用してユーザーを認証できます。

PKI 認証を設定した後、ユーザーおよび API クライアントは個人証明書を使用してログインできます。証明書を持たないユーザーは、API トークン、ローカル管理者パスワード、または他の認証プロバイダーを含む他の認証オプションを引き続き使用できます。PKI 認証は、Web UI、gRPC、および REST API と同じポート番号で使用できます。

PKI 認証を設定する場合、デフォルトでは、Red Hat Advanced Cluster Security for Kubernetes は、PKI、Web UI、gRPC、その他のシングルサインオン (SSO) プロバイダー、および REST API に同じポートを使用します。YAML 設定ファイルを使用してエンドポイントを設定および公開することにより、PKI 認証用に別のポートを設定することもできます。

18.2.1. RHACS ポータルを使用した PKI 認証の設定

RHACS ポータルを使用して、公開鍵インフラストラクチャー (PKI) 認証を設定できます。

手順

1. RHACS ポータルで、**Platform Configuration → Access Control** に移動します。
2. **Create Auth Provider** をクリックし、ドロップダウンリストから **User Certificates** を選択します。
3. **Name** フィールドに、この認証プロバイダーの名前を指定します。
4. **CA certificate(s) (PEM)** フィールドに、ルート CA 証明書を PEM 形式で貼り付けます。
5. PKI 認証を使用して RHACS にアクセスするユーザーに **Minimum access role** を割り当てます。ユーザーは、RHACS にログインするために、このロールに付与された権限、またはより高い権限を持つロールを持っている必要があります。

   ヒント

   セキュリティーのために、セットアップを実行する際には、最初に **Minimum access role** を **None** に設定することを推奨します。後で、**Access Control** ページに戻って、ID プロバイダーのユーザーメタデータに基づいて、より調整されたアクセスルールを設定できます。

6. RHACS にアクセスするユーザーとグループのアクセスルールを追加するには、**Rules** セクションで **Add new rule** をクリックします。たとえば、**administrator** と呼ばれるユーザーに **Admin** のロールを与える場合は、次のキーと値のペアを使用してアクセスルールを作成できます。

キー	値
名前	管理者 (administrator)
Role	Admin

7. **Save** をクリックします。

18.2.2. roxctl CLI を使用した PKI 認証の設定

roxctl CLI を使用して PKI 認証を設定できます。

手順

- 以下のコマンドを実行します。

```
$ roxctl -e <hostname>:<port_number> central userpki create -c <ca_certificate_file> -r
<default_role_name> <provider_name>
```

18.2.3. 認証キーおよび証明書の更新

RHACS ポータルを使用して、認証キーおよび証明書を更新できます。

手順

1. 新しい認証プロバイダーを作成します。
2. 古い認証プロバイダーから新しい認証プロバイダーにロールマッピングをコピーします。
3. 古いルート CA キーを使用して、古い認証プロバイダーの名前を変更または削除します。

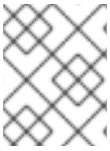
18.2.4. クライアント証明書を使用したログイン

PKI 認証を設定すると、RHACS ポータルのログインページに証明書プロンプトが表示されます。プロンプトは、設定されたルート CA により信頼されているクライアント証明書がユーザーのシステムにインストールされている場合にのみ表示されます。

このセクションで説明されている手順に従って、クライアント証明書を使用してログインします。

手順

1. RHACS ポータルを開きます。
2. ブラウザーのプロンプトで証明書を選択します。
3. ログインページで、認証プロバイダー名オプションを選択し、証明書を使用してログインします。証明書を使用してログインしない場合は、管理者パスワードまたは別のログイン方法を使用してログインすることもできます。



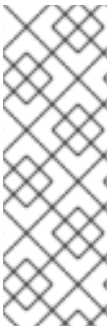
注記

クライアント証明書を使用して RHACS ポータルにログインすると、ブラウザーを再起動しない限り、別の証明書でログインすることができません。

18.3. 認証プロバイダーについて

認証プロバイダーは、ユーザーアイデンティティーのサードパーティーソース (アイデンティティープロバイダー (IDP) など) に接続し、ユーザーアイデンティティーを取得します。そして、そのアイデンティティーに基づいてトークンを発行し、そのトークンを Red Hat Advanced Cluster Security for Kubernetes (RHACS) に返します。このトークンにより、RHACS はユーザーを承認できるようになります。RHACS は、ユーザーインターフェイスおよび API 呼び出し内でトークンを使用します。

RHACS をインストールした後、ユーザーを認証するように IDP を設定する必要があります。



注記

IDP として OpenID Connect (OIDC) を使用している場合、RHACS は、ユーザー ID トークンまたは **UserInfo** エンドポイント応答から **groups**、**email**、**userid**、**name** などの特定のクレームの値を検査するマッピングルールに依存してユーザーを承認します。これらの詳細が存在しない場合、マッピングは成功せず、ユーザーは必要なリソースにアクセスできません。したがって、マッピングを成功させるには、IDP からのユーザーを承認するために必要なクレーム (**groups** など) が IDP の認証応答に含まれていることを確認する必要があります。

関連情報

- Okta Identity Cloud を SAML 2.0 プロバイダーとして設定

- Google Workspace を OIDC ID プロバイダーとして設定する
- OpenShift Container Platform OAuth サーバーをアイデンティティープロバイダーとして設定
- SSO 設定を使用して Azure AD を RHACS に接続する

18.3.1. クレームマッピング

クレームは、アイデンティティプロバイダーが発行するトークン内にユーザーに関するデータを含めます。

クレームマッピングを使用すると、RHACS が IDP から受け取るクレーム属性を RHACS 発行トークンの別の属性にカスタマイズするかどうかを指定できます。クレームマッピングを使用しない場合、RHACS は RHACS 発行のトークンにクレーム属性を含めません。

たとえば、クレームマッピングを使用して、ユーザー ID の **roles** から RHACS 発行のトークンの **groups** にマッピングできます。

RHACS は、認証プロバイダーごとに異なるデフォルトのクレームマッピングを使用します。

18.3.1.1. OIDC のデフォルトのクレームマッピング

次のリストは、デフォルトの OIDC クレームマッピングを示しています。

- **sub** から **userid** に
- **name** から **name** に
- **email** から **email** に
- **groups** から **groups** に

18.3.1.2. Auth0 のデフォルトのクレームマッピング

Auth0 のデフォルトのクレームマッピングは、OIDC のデフォルトのクレームマッピングと同じです。

18.3.1.3. SAML 2.0 のデフォルトのクレームマッピング

次のリストは、SAML 2.0 のデフォルトのクレームマッピングに適用されます。

- **Subject.NameID** は **userid** にマッピングされる
- 応答からのすべての SAML **AttributeStatement.Attribute** は、その名前にマッピングされる

18.3.1.4. Google IAP のデフォルトのクレームマッピング

次のリストは、Google IAP のデフォルトのクレームマッピングを示しています。

- **sub** から **userid** に
- **email** から **email** に
- **hd** から **hd** に
- **google.access_levels** から **access_levels** に

18.3.1.5. ユーザー証明書のデフォルトのクレームマッピング

ユーザー証明書は、サードパーティーの IDP と通信する代わりに、ユーザーが使用する証明書からユーザー情報を取得するため、他のすべての認証プロバイダーとは異なります。

ユーザー証明書のデフォルトのクレームマッピングには次のものが含まれます。

- **CertFingerprint** から **userid** に
- **Subject → Common Name** から **name** に
- **EmailAddresses** から **email** に
- **Subject → Organizational Unit** から **groups** に

18.3.1.6. OpenShift Auth のデフォルトのクレームマッピング

次のリストは、OpenShift Auth のデフォルトのクレームマッピングを示しています。

- **groups** から **groups** に
- **uid** から **userid** に
- **name** から **name** に

18.3.2. Rules

ユーザーを承認するために、RHACS は、ユーザー ID から **groups**、**email**、**userid**、**name** などの特定のクレームの値を検査するマッピングルールに依存します。ルールを使用すると、特定の値を持つ属性を持つユーザーを特定のロールにマッピングできます。例として、ルールには次の内容を含めることができます。key は **email**、**value** は **john@redhat.com**、**role** は **Admin** です。

クレームが欠落している場合、マッピングは成功せず、ユーザーは必要なリソースにアクセスできません。したがって、マッピングを成功させるには、IDP からの認証応答に、ユーザー (**groups** など) を承認するために必要なクレームが含まれていることを確認する必要があります。

18.3.3. 最小アクセスロール

RHACS は、特定の認証プロバイダーが発行した RHACS トークンを使用して、すべての呼び出し元に最小限のアクセスロールを割り当てます。最小アクセスロールは、デフォルトでは **None** に設定されています。

たとえば、**Analyst** という最小アクセスロールを持つ認証プロバイダーがあるとします。その場合、このプロバイダーを使用してログインするすべてのユーザーには、**Analyst** ロールが割り当てられます。

18.3.4. 必須の属性

必須の属性は、ユーザー ID に特定の値の属性があるかどうかに基づいて、RHACS トークンの発行を制限できます。

たとえば、キー **is_internal** の属性の属性値が **true** である場合にのみトークンを発行するように RHACS を設定できます。**is_internal** 属性が **false** に設定されているか、設定されていないユーザーはトークンを取得しません。

18.4. アイデンティティープロバイダーの設定

18.4.1. Okta Identity Cloud を SAML 2.0 プロバイダーとして設定

Okta は、Red Hat Advanced Cluster Security for Kubernetes (RHACS) のシングルサインオン (SSO) プロバイダーとして使用できます。

18.4.1.1. Okta アプリの作成

Okta を Red Hat Advanced Cluster Security for Kubernetes の SAML 2.0 プロバイダーとして使用する前に、Okta アプリを作成する必要があります。

警告

Okta の **Developer Console** はカスタム SAML 2.0 アプリケーションの作成をサポートしていません。**Developer Console** を使用している場合は、最初に **Admin Console** (**Classic UI**) に切り替える必要があります。切り替えるには、ページの左上にある **Developer Console** をクリックして、**Classic UI** を選択します。

前提条件

- Okta ポータルの管理者権限を持つアカウントが必要です。

手順

1. Okta ポータルで、メニューバーから **Applications** を選択します。
2. **Add Application** をクリックし、**Create New App** を選択します。
3. **Create a New Application Integration** ダイアログボックスで、プラットフォームを **Web** のままにし、ユーザーにサインインするプロトコルに **SAML 2.0** を選択します。
4. **Create** をクリックします。
5. **General Settings** ページで、**App name** フィールドにアプリの名前を入力します。
6. **Next** をクリックします。
7. **SAML Settings** ページで、次のフィールドに値を設定します。
    a. **Single Sign-On URL**
        - **https://<RHACS_portal_hostname>/sso/providers/saml/acs** を指定します。
        - **Use this for Recipient URL and Destination URL**オプションをオンのままにします。
        - RHACS ポータルにさまざまな URL でアクセスできる場合は、**Allow this app to request other SSO URLs** オプションをオンにして、指定した形式を使用して代替 URL を追加することでそれらを追加できます。
    b. **Audience URI (SP Entity ID)**
        - 値を **RHACS** または任意の別の値に設定します。

- 選択した値を覚えておいてください。Red Hat Advanced Cluster Security for Kubernetes を設定するときに、この値が必要になります。

c. **Attribute Statements**

- 少なくとも 1 つの属性ステートメントを追加する必要があります。
- Red Hat は、email 属性の使用を推奨します。
  - **Name:** email
  - **Format:** 指定なし
  - **Value:** user.email

8. 続行する前に、少なくとも 1 つの **Attribute Statement** が設定されていることを確認してください。
9. **Next** をクリックします。
10. **Feedback** ページで、該当するオプションを選択します。
11. 適切な **App type** を選択します。
12. **Finish** をクリックします。

設定が完了すると、新しいアプリの **サインオン** 設定ページにリダイレクトされます。黄色のボックスには、Red Hat Advanced Cluster Security for Kubernetes を設定するのに必要な情報へのリンクが含まれています。

アプリを作成したら、Okta ユーザーをこのアプリケーションに割り当てます。**Assignments** タブに移動し、Red Hat Advanced Cluster Security for Kubernetes にアクセスできる個々のユーザーまたはグループのセットを割り当てます。たとえば、グループ **Everyone** を割り当てて、組織内のすべてのユーザーが Red Hat Advanced Cluster Security for Kubernetes にアクセスできるようにします。

18.4.1.2. SAML 2.0 アイデンティティープロバイダーの設定

このセクションの手順を使用して、Security Assertion Markup Language (SAML) 2.0 ID プロバイダーを Red Hat Advanced Cluster Security for Kubernetes (RHACS) と統合します。

前提条件

- RHACS で ID プロバイダーを設定する権限が必要です。
- Okta ID プロバイダーの場合、RHACS 用に設定された Okta アプリが必要です。

手順

1. RHACS ポータルで、**Platform Configuration → Access Control** に移動します。
2. **Create auth provider** をクリックし、ドロップダウンリストから **SAML 2.0** を選択します。
3. **Name** フィールドに、この認証プロバイダーを識別する名前を入力します。たとえば、**Okta** や **Google** などです。統合名は、ユーザーが適切なサインインオプションを選択できるように、ログインページに表示されます。

4. ServiceProvider issuer フィールドに、Okta で **Audience URI** または **SP Entity ID** として使用している値、または他のプロバイダーで同様の値を入力します。

5. Configuration のタイプを選択します。

   - Option 1: Dynamic Configuration: このオプションを選択した場合は、IdP Metadata URL を入力するか、ID プロバイダーコンソールから利用可能な Identity Provider metadata の URL を入力します。設定値は URL から取得します。

   - Option 2: Static Configuration: Okta コンソールの View Setup Instructions リンクから必要な静的フィールドをコピーするか、他のプロバイダーの場合は同様の場所にコピーします。

     - **IdP 発行者**
     - **IdP SSO URL**
     - **名前 ID 形式**
     - **IdP 証明書 (PEM)**

6. SAML を使用して RHACS にアクセスするユーザーに **最小アクセスロール** を割り当てます。

   ヒント

   セットアップの完了時に、**最小アクセスルール** を **管理者** に設定します。後で、Access Control ページに戻って、ID プロバイダーのユーザーメタデータに基づいて、より調整されたアクセスルールを設定できます。

7. Save をクリックします。

重要

SAML ID プロバイダーの認証応答が次の条件を満たしている場合:

- **NotValidAfter** アサーションを含み、ユーザーセッションは **NotValidAfter** フィールドで指定された時間が経過するまで有効なままです。ユーザーセッションの有効期限が切れた後、ユーザーは再認証する必要があります。
- **NotValidAfter** アサーションを含まない: ユーザーセッションは 30 日間有効なままであり、その後ユーザーは再認証する必要があります。

検証

1. RHACS ポータルで、Platform Configuration → Access Control に移動します。
2. Auth Providers タブを選択します。
3. 設定を確認する認証プロバイダーをクリックします。
4. Auth Provider セクションのヘッダーから Test login を選択します。新しいブラウザータブで、Test login ページが開きます。
5. 認証情報を使用してログインします。

   - 正常にログインした場合、RHACS は、システムへのログインに使用した資格情報に対して ID プロバイダーが送信した **User ID** と **User Attributes** を表示します。

IDプロバイダーが送信した **User ID** と **User Attributes** を表示します。

- ログイン試行が失敗した場合、RHACS は ID プロバイダーの応答を処理できなかった理由を説明するメッセージを表示します。

6. **Test login** ブラウザータブを閉じます。

注記

応答が認証の成功を示している場合でも、ID プロバイダーからのユーザーメタデータに基づいて追加のアクセスルールを作成しないといけない場合があります。

18.4.2. Google Workspace を OIDC ID プロバイダーとして設定する

Google Workspace は、Red Hat Advanced Cluster Security for Kubernetes のシングルサインオン (SSO) プロバイダーとして使用できます。

18.4.2.1. GCP プロジェクトの OAuth 2.0 認証情報の設定

Google Workspace を Red Hat Advanced Cluster Security for Kubernetes の ID プロバイダーとして設定するには、最初に GCP プロジェクトの OAuth 2.0 認証情報を設定する必要があります。

前提条件

- 新しいプロジェクトを作成するには、組織の Google Workspace アカウントへの管理者レベルのアクセス権、または既存のプロジェクトの OAuth 2.0 認証情報を作成および設定するためのパーミッションが必要です。Red Hat は、Red Hat Advanced Cluster Security for Kubernetes へのアクセスを管理する新しいプロジェクトを作成することを推奨します。

手順

1. 新しい Google Cloud Platform (GCP) プロジェクトを作成します。プロジェクトの作成および管理 に関する Google ドキュメントのトピックをご覧ください。
2. プロジェクトを作成したら、Google API コンソールで **Credentials** ページを開きます。
3. ロゴの近くの左上隅にリスト表示されているプロジェクト名を確認して、正しいプロジェクトを使用していることを確認します。
4. 新しい認証情報を作成するには、**Create Credentials→ OAuth client ID** に移動します。
5. **Application type** で **Web application** を選択します。
6. **Name** ボックスに、アプリケーションの名前 (**RHACS** など) を入力します。
7. **Authorized redirect URIs** ボックスに、**https://<stackrox_hostname>:<port_number>/sso/providers/oidc/callback** と入力します。
   - **<stackrox_hostname>** を、Central インスタンスを公開するホスト名に置き換えます。
   - **<port_number>** を、Central を公開するポート番号に置き換えます。標準の HTTPS ポート **443** を使用している場合は、ポート番号を省略できます。
8. **Create** をクリックします。これにより、アプリケーションと認証情報が作成され、認証情報ページにリダイレクトされます。

9. 情報ボックスが開き、新しく作成されたアプリケーションの詳細が表示されます。情報ボックスを閉じます。

10. **.apps.googleusercontent.com** で終わる **クライアント ID** をコピーして保存します。このクライアント ID は、Google API コンソールを使用して確認できます。

11. 左側のナビゲーションメニューから **OAuth consent screen** を選択します。

注記

OAuth 同意画面の設定は、前の手順で作成したアプリケーションだけでなく、GCP プロジェクト全体で有効です。このプロジェクトですでに OAuth 同意画面が設定されていて、Red Hat Advanced Cluster Security for Kubernetes ログインに別の設定を適用する場合は、新しい GCP プロジェクトを作成します。

12. OAuth 同意画面ページで、以下を行います。

    a. **Application type** に **Internal** を選択します。**Public** を選択すると、Google アカウントを持っている人なら誰でもログインできます。

    b. わかりやすい **アプリケーション名** を入力します。この名前は、ユーザーがサインインするときに同意画面に表示されます。たとえば、**RHACS** または **<organization_name> SSO for Red Hat Advanced Cluster Security for Kubernetes** を使用します。

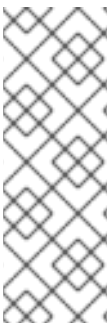
    c. **Scopes for Google APIs** に、**email**、**profile**、**openid** スコープのみがリストされていることを確認します。シングルサインオンには、これらのスコープのみが必要です。追加のスコープを付与すると、機密データが公開されるリスクが高まります。

18.4.2.2. クライアントシークレットの指定

Red Hat Advanced Cluster Security for Kubernetes バージョン 3.0.39 以降は、クライアントシークレットを指定するときに OAuth 2.0 認証コード付与 認証フローをサポートします。この認証フローを使用すると、Red Hat Advanced Cluster Security for Kubernetes は更新トークンを使用して、OIDC ID プロバイダーで設定されたトークンの有効期限を超えてユーザーがログインし続けるようにします。

ユーザーがログアウトすると、Red Hat Advanced Cluster Security for Kubernetes はクライアント側から更新トークンを削除します。さらに、ID プロバイダー API が更新トークンの失効をサポートしている場合、Red Hat Advanced Cluster Security for Kubernetes は、更新トークンを失効させる要求も ID プロバイダーに送信します。

OIDC ID プロバイダーと統合するように Red Hat Advanced Cluster Security for Kubernetes を設定するときに、クライアントシークレットを指定できます。



注記

- **フラグメント コールバックモード** で **クライアントシークレット** を使用することはできません。
- 既存の認証プロバイダーの設定を編集することはできません。
- **クライアントシークレット** を使用する場合は、Red Hat Advanced Cluster Security for Kubernetes で新しい OIDC 統合を作成する必要があります。

Red Hat は、Red Hat Advanced Cluster Security for Kubernetes を OIDC ID プロバイダーに接続するときに、クライアントシークレットを使用することを推奨します。**クライアントシークレット** を使用しない場合は、**Do not use Client Secret (not recommended)**オプションを選択する必要があります。

18.4.2.3. OIDC ID プロバイダーの設定

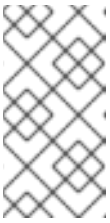
OpenID Connect (OIDC) ID プロバイダーを使用するように Red Hat Advanced Cluster Security for Kubernetes (RHACS) を設定できます。

前提条件

- Google Workspace などの ID プロバイダーでアプリケーションを設定している。
- RHACS で ID プロバイダーを設定する権限が必要です。

手順

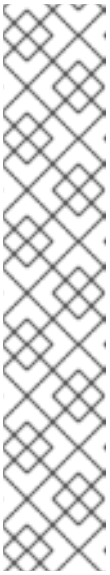
1. RHACS ポータルで、**Platform Configuration → Access Control** に移動します。
2. **Create auth provider** をクリックし、ドロップダウンリストから **OpenID Connect** を選択します。
3. 以下のフィールドに情報を入力します。
   - **Name**: 認証プロバイダーを識別する名前。たとえば、**Google Workspace** です。統合名は、ユーザーが適切なサインインオプションを選択できるように、ログインページに表示されます。
   - **Callback mode**: ID プロバイダーが別のモードを必要としない限り、デフォルト値である **Auto-select (recommended)** を選択します。

     注記

     Fragment モードは、シングルページアプリケーション (SPA) の制限を考慮して設計されています。Red Hat は、初期の統合の **Fragment** モードのみをサポートしています。最近の統合でこのモードを使用することは推奨していません。

   - **Issuer**: ID プロバイダーのルート URL。たとえば、Google Workspace の場合は **https://accounts.google.com** です。詳細は、ID プロバイダーのドキュメントを参照してください。



注記

RHACS バージョン 3.0.49 以降を使用している場合は、**Issuer** に対して次のアクションを実行できます。

- ルート URL の前に **https+insecure://** を付けて、TLS 検証を飛ばします。この設定はセキュアでなく、Red Hat は推奨していません。テスト目的でのみ使用してください。
- ルート URL とともに **?key1=value1&key2=value2** などのクエリー文字列を指定します。RHACS は、入力したとおりに **Issuer** の値を認証エンドポイントに追加します。これを使用して、プロバイダーのログイン画面をカスタマイズできます。たとえば、**hd** パラメーター を使用して Google Workspace のログイン画面を特定のホストドメインに最適化したり、**pfidpadapterid** パラメーターを使用して **PingFederate** で認証方法を事前に選択したりできます。

- **クライアント ID**: 設定されたプロジェクトの OIDC クライアント ID。
- **Client Secret**: ID プロバイダー (IdP) から提供されたクライアントシークレットを入力します。推奨されていないクライアントシークレットを使用していない場合は、**Do not use Client Secret** を選択します。

4. 選択した ID プロバイダーを使用して RHACS にアクセスするユーザーに **最小アクセスロール** を割り当てます。

ヒント

セットアップの完了時に、**最小アクセスルール** を **管理者** に設定します。後で、**Access Control** ページに戻って、ID プロバイダーのユーザーメタデータに基づいて、より調整されたアクセスルールを設定できます。

5. RHACS にアクセスするユーザーとグループのアクセスルールを追加するには、**Rules** セクションで **Add new rule** をクリックします。たとえば、**administrator** と呼ばれるユーザーに **Admin** のロールを与える場合は、次のキーと値のペアを使用してアクセスルールを作成できます。

キー	値
名前	管理者 (administrator)
Role	Admin

6. **Save** をクリックします。

検証

1. RHACS ポータルで、**Platform Configuration → Access Control** に移動します。
2. **Auth providers** タブを選択します。
3. 設定を確認する認証プロバイダーを選択します。
4. **Auth Provider** セクションのヘッダーから **Test login** を選択します。新しいブラウザータブで、**Test login** ページが開きます。

5. クレデンシャルを使用してログインします。
   - 正常にログインした場合、RHACS は、システムへのログインに使用した資格情報に対して ID プロバイダーが送信した **User ID** と **User Attributes** を表示します。
   - ログイン試行が失敗した場合、RHACS は ID プロバイダーの応答を処理できなかった理由を説明するメッセージを表示します。
6. **Test Login** ブラウザータブを閉じます。

18.4.3. OpenShift Container Platform OAuth サーバーをアイデンティティープロバイダーとして設定

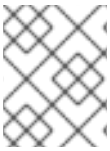
OpenShift Container Platform には、Red Hat Advanced Cluster Security for Kubernetes (RHACS) の認証プロバイダーとして使用できる組み込みの OAuth サーバーが含まれています。

18.4.3.1. OpenShift Container Platform OAuth サーバーをアイデンティティープロバイダーとして設定

組み込みの OpenShift Container Platform OAuth サーバーを RHACS の ID プロバイダーとして統合するには、このセクションの手順を使用します。

前提条件

- RHACS で ID プロバイダーを設定するには、**AuthProvider** 権限が必要である。
- ID プロバイダーを介して OpenShift Container Platform OAuth サーバーでユーザーおよびグループをすでに設定しておく必要がある。ID プロバイダーの要件は、ID プロバイダーの設定の概要 を参照してください。



注記

以下の手順では、OpenShift Container Platform OAuth サーバー用に **central** という名前のメインルートを 1 つだけ設定します。

手順

1. RHACS ポータルで、**Platform Configuration → Access Control** に移動します。
2. **Create auth provider** をクリックし、ドロップダウンリストから **OpenShift Auth** を選択します。
3. **Name** フィールドに認証プロバイダーの名前を入力します。
4. 選択した ID プロバイダーを使用して RHACS にアクセスするユーザーに **Minimum access role** を割り当てます。ユーザーは、RHACS にログインするために、このロールに付与された権限、またはより高い権限を持つロールを持っている必要があります。

ヒント

セキュリティーのために、セットアップを実行する際には、最初に **Minimum access role** を **None** に設定することを推奨します。後で、**Access Control** ページに戻って、ID プロバイダーのユーザーメタデータに基づいて、より調整されたアクセスルールを設定できます。

5. オプション: RHACS にアクセスするユーザーとグループのアクセスルールを追加するには、**Rules** セクションで **Add new rule** をクリックし、ルール情報を入力して **Save** をクリックします。アクセスを設定するには、ユーザーまたはグループの属性が必要です。

ヒント

グループは通常、チームまたはアクセス許可セットに関連付けられており、ユーザーよりも頻繁に変更する必要がないため、グループマッピングはより堅牢です。

OpenShift Container Platform でユーザー情報を取得するには、以下のいずれかの方法を使用できます。

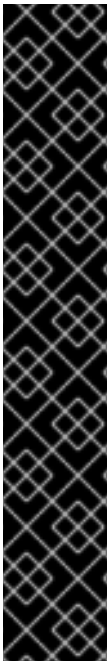
- **User Management** → **Users** → <username> → **YAML** をクリックします。
- **k8s/cluster/user.openshift.io~v1~User/<username>/yaml** ファイルにアクセスし、**name**、**uid** (RHACS の **userid**)、および **groups** の値を書き留めます。
- **OpenShift Container Platform API リファレンス** で説明されているように、OpenShift Container Platform API を使用します。

次の設定例では、次の属性を持つ **Admin** ロールのルールを設定する方法を説明します。

- **name**: **administrator**
- **groups**: **["system:authenticated", "system:authenticated:oauth", "myAdministratorsGroup"]**
- **uid**: **12345-00aa-1234-123b-123fcdef1234**

次のいずれかの手順を使用して、この管理者ロールのルールを追加できます。

- 名前のルールを設定するには、**Key** ドロップダウンリストから **name** を選択し、**Value** フィールドに **administrator** と入力して、**Role** で **Administrator** を選択します。
- グループのルールを設定するには、**Key** ドロップダウンリストから **groups** を選択し、**Value** フィールドに **myAdministratorsGroup** と入力して、**Role** で **Admin** を選択します。
- ユーザー名のルールを設定するには、**Key** ドロップダウンリストから **userid** を選択し、**Value** フィールドに **12345-00aa-1234-123b-123fcdef1234** を入力して、**Role** で **Admin** を選択します。



重要

- OpenShift Container Platform OAuth サーバーにカスタム TLS 証明書を使用する場合は、CA のルート証明書を信頼されたルート CA として Red Hat Advanced Cluster Security for Kubernetes に追加する必要があります。そうしないと、Central は OpenShift Container Platform OAuth サーバーに接続できません。
- **roxctl** CLI を使用して Red Hat Advanced Cluster Security for Kubernetes をインストールするときに OpenShift Container Platform OAuth サーバー統合を有効にするには、Central で **ROX_ENABLE_OPENSHIFT_AUTH** 環境変数を **true** に設定します。

```
$ oc -n stackrox set env deploy/central
ROX_ENABLE_OPENSHIFT_AUTH=true
```

- アクセスルールの場合、OpenShift Container Platform OAuth サーバーはキー **Email** を返しません。

関連情報

- LDAP アイデンティティープロバイダーの設定
- 信頼できる認証局の追加

18.4.3.2. OpenShift Container Platform OAuth サーバーの追加ルートの作成

Red Hat Advanced Cluster Security for Kubernetes ポータルを使用して OpenShift Container Platform OAuth サーバーを ID プロバイダーとして設定すると、RHACS は OAuth サーバーのルートを 1 つだけ設定します。ただし、Central カスタムリソースで注釈として指定することにより、追加のルートを作成できます。

前提条件

- サービスアカウントを OpenShift Container Platform OAuth サーバーの OAuth クライアントとして 設定しておく必要がある。

手順

- RHACS Operator を使用して RHACS をインストールした場合:
  1. Central カスタムリソースのパッチを含む **CENTRAL_ADDITIONAL_ROUTES** 環境変数を作成します。

```
$ CENTRAL_ADDITIONAL_ROUTES='
spec:
  central:
    exposure:
      loadBalancer:
        enabled: false
        port: 443
      nodePort:
        enabled: false
      route:
        enabled: true
```

```
    persistence:
      persistentVolumeClaim:
        claimName: stackrox-db
  customize:
    annotations:
      serviceaccounts.openshift.io/oauth-redirecturi.main: sso/providers/openshift/callback
(1)
      serviceaccounts.openshift.io/oauth-redirectreference.main: "
{\"kind\":\"OAuthRedirectReference\",\"apiVersion\":\"v1\",\"reference\":
{\"kind\":\"Route\",\"name\":\"central\"}}" (2)
      serviceaccounts.openshift.io/oauth-redirecturi.second:
sso/providers/openshift/callback (3)
      serviceaccounts.openshift.io/oauth-redirectreference.second: "
{\"kind\":\"OAuthRedirectReference\",\"apiVersion\":\"v1\",\"reference\":
{\"kind\":\"Route\",\"name\":\"second-central\"}}" (4)
'
```

(1) メインルートを設定するためのリダイレクト URI。

(2) メインルートのリダイレクト URI 参照。

(3) 2 番目のルートを設定するためのリダイレクト。

(4) 2 番目のルートのリダイレクト参照。

2. **CENTRAL_ADDITIONAL_ROUTES** パッチを Central カスタムリソースに適用します。

```
$ oc patch centrals.platform.stackrox.io \
  -n <namespace> \ (1)
  <custom-resource> \ (2)
  --patch "$CENTRAL_ADDITIONAL_ROUTES" \
  --type=merge
```

(1) **<namespace>** を、Central カスタムリソースを含むプロジェクトの名前に置き換えます。

(2) **<custom-resource>** を Central カスタムリソースの名前に置き換えます。

- または、Helm を使用して RHACS をインストールした場合:

1. 次のアノテーションを **values-public.yaml** ファイルに追加します。

```
customize:
  central:
    annotations:
      serviceaccounts.openshift.io/oauth-redirecturi.main: sso/providers/openshift/callback
(1)
      serviceaccounts.openshift.io/oauth-redirectreference.main: "
{\"kind\":\"OAuthRedirectReference\",\"apiVersion\":\"v1\",\"reference\":
{\"kind\":\"Route\",\"name\":\"central\"}}" (2)
      serviceaccounts.openshift.io/oauth-redirecturi.second:
sso/providers/openshift/callback (3)
```

```
      serviceaccounts.openshift.io/oauth-redirectreference.second: "
{\"kind\":\"OAuthRedirectReference\",\"apiVersion\":\"v1\",\"reference\":
{\"kind\":\"Route\",\"name\":\"second-central\"}}" 4
```

1. メインルートを設定するためのリダイレクト。
2. メインルートのリダイレクトリファレンス。
3. 2 番目のルートを設定するためのリダイレクト。
4. 2 番目のルートのリダイレクト参照。

2. **helm upgrade** を使用して、Central カスタムリソースにカスタムアノテーションを適用します。

```
$ helm upgrade -n stackrox \
  stackrox-central-services rhacs/central-services \
  -f <path_to_values_public.yaml> 1
```

1. **-f** オプションを使用して、**values-public.yaml** 設定ファイルのパスを指定します。

関連情報

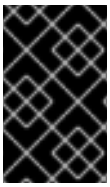
- OAuth クライアントとしてのサービスアカウント
- OAuth クライアントとしてのサービスアカウントの URI のリダイレクト

18.4.4. SSO 設定を使用して Azure AD を RHACS に接続する

サインオン (SSO) 設定を使用して Azure Active Directory (AD) を RHACS に接続するには、特定のクレーム (トークンに対する **group** クレームなど) を追加し、ユーザー、グループ、またはその両方をエンタープライズアプリケーションに割り当てる必要があります。

18.4.4.1. SSO 設定を使用した SAML アプリケーションのトークンへのグループクレームの追加

トークンに **group** クレームを含めるように Azure AD でのアプリケーション登録を設定します。手順については、SSO 設定を使用して SAML アプリケーションのトークンにグループクレームを追加する を参照してください。



重要

最新バージョンの Azure AD を使用していることを確認してください。Azure AD を最新バージョンにアップグレードする方法の詳細は、Azure AD Connect: 以前のバージョンから最新バージョンへのアップグレード を参照してください。

18.5. 管理者ユーザーの削除

Red Hat Advanced Cluster Security for Kubernetes (RHACS) は、インストールプロセス中に、ユーザー名とパスワードを使用してログインできる管理者アカウント **admin** を作成します。パスワードは、明示的に上書きされない限り動的に生成され、RHACS インスタンスごとに一意になります。

実稼働環境では、認証プロバイダーを作成し、**admin** ユーザーを削除することを強く推奨します。

18.5.1. インストール後に管理者ユーザーを削除

認証プロバイダーが正常に作成されたら、**admin** ユーザーを削除することを強く推奨します。

admin ユーザーの削除は、RHACS ポータルのインストール方法によって異なります。

手順

次のいずれかの手順を実行します。

- Operator インストールの場合、**Central** カスタムリソースで **central.adminPasswordGenerationDisabled** を **true** に設定します。
- Helm のインストールの場合:
  1. **Central** Helm 設定で、**central.adminPassword.generate** を **false** に設定します。
  2. 手順に従って設定を変更してください。詳細は、「デプロイメント後の設定オプションの変更」を参照してください。
- **roxctl** インストールの場合:
  1. マニフェストを生成するときに、**Disable password generation** を **false** に設定します。
  2. 変更を適用するには、**roxctl** を使用して Central をインストールする手順に従います。詳細は、「roxctl CLI を使用した Central のインストール」を参照してください。

関連情報

- central-services Helm チャートをデプロイした後の設定オプションの変更 (OpenShift Container Platform)
- central-services Helm チャートをデプロイした後の設定オプションの変更 (Kubernetes)
- roxctl CLI を使用して Central をインストールする

設定変更を適用した後、**admin** ユーザーとしてログインできなくなります。

注記

設定の変更を元に戻すことで、フォールバックとして **admin** ユーザーを再度追加できます。**admin** ユーザーを再度有効にすると、新しいパスワードが生成されます。

18.6. 短期間のアクセス権の設定

Red Hat Advanced Cluster Security for Kubernetes (RHACS) は、ユーザーインターフェイスおよび API 呼び出しへの短期間のアクセス権を設定する機能を備えています。

これを設定するには、OpenID Connect (OIDC) ID トークンを RHACS 発行のトークンと交換します。

これは、有効期間の長い API トークンよりも短期間のアクセス権が望ましい継続的インテグレーション (CI) を使用する場合に特に推奨されます。

次の手順では、ユーザーインターフェイスと API 呼び出しへの短期間のアクセス権を設定する方法に関するワークフローを概説します。

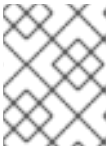
1. RHACS が発行した有効期間が短いトークンを交換するために、OIDC ID トークン発行者を信頼するように RHACS を設定します。
2. API を呼び出して、OIDC ID トークンを有効期間の短い RHACS 発行のトークンと交換します。

18.6.1. OIDC ID トークン発行者の短期アクセスを設定する

OpenID Connect (OIDC) ID トークン発行者に対する短期間のアクセス権の設定を開始します。

手順

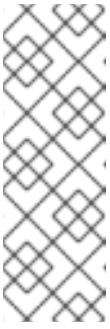
1. RHACS ポータルで、**Platform Configuration → Integrations** に移動します。
2. **Authentication Tokens** カテゴリーまでスクロールし、**Machine access configuration** をクリックします。
3. **Create configuration** をクリックします。
4. **configuration type** を選択し、次のいずれかを選択します。
   - 任意の OIDC ID トークン発行者を使用する場合は、**Generic** を選択します。
   - GitHub Actions から RHACS にアクセスする予定の場合は、**GitHub Actions** を選択します。
5. OIDC ID トークンの発行者を入力します。
6. この設定に基づいて発行するトークンの **token lifetime** を入力します。



注記

token lifetime の形式は **XhYmZs** です。24 時間より長く設定することはできません。

7. ルールを設定に追加します。
   - **Key** は、使用する OIDC トークンのクレームです。
   - **Value** は、期待される OIDC トークンのクレームの値です。
   - **Role** は、OIDC トークンのクレームと値が存在する場合にトークンに割り当てるロールです。



注記

ルールは、クレームの値に基づいてロールを割り当てる認証プロバイダールールに似ています。

一般的なルールとして、Red Hat はルール内で一意で不変のクレームを使用することを推奨します。通常は、OIDC ID トークン内で **sub** クレームを使用することを推奨します。OIDC トークンのクレームの詳細は、標準 OIDC クレームのリスト を参照してください。

8. **Save** をクリックします。

18.6.2. ID トークンの交換

前提条件

- 有効な OpenID Connect (OIDC) トークンがある。
- アクセスする RHACS インスタンスの **Machine access configuration** を追加した。

手順

1. POST リクエストの JSON データを準備します。

```
{
   "idToken": "<id_token>"
}
```

2. POST リクエストを API **/v1/auth/m2m/exchange** に送信します。
3. API レスポンスを待ちます。

```
{
   "accessToken": "<access_token>"
}
```

4. 返されたアクセストークンを使用して、RHACS インスタンスにアクセスします。

注記

GitHub Actions を使用する場合は、stackrox/central-login GitHub Action を使用できます。

18.7. マルチテナンシーについて

Red Hat Advanced Cluster Security for Kubernetes には、Central インスタンス内にマルチテナンシーを実装する方法が用意されています。

マルチテナンシーは、RHACS 内のロールベースアクセス制御 (RBAC) とアクセススコープを使用して実装できます。

18.7.1. リソーススコープについて

RHACS には、RBAC 内で使用されるリソースが組み込まれています。各リソースには、権限が関連付けられているだけでなく、スコープが指定されています。

RHACS では、リソースに次のタイプのスコープが指定されています。

- グローバルスコープ。リソースはクラスターにも namespace にも割り当てられません。
- クラススコープ。リソースは特定のクラスターに割り当てられます。
- namespace スコープ。リソースは特定の namespace に割り当てられます。

カスタムのアクセススコープを作成するときは、リソースのスコープが重要です。カスタムのアクセススコープは、RHACS 内でマルチテナンシーを実現するために使用します。

アクセススコープでのスコープ指定には、クラススコープか namespace スコープのリソースのみが適用されます。グローバルスコープのリソースは、アクセススコープによってスコープ指定されません。したがって、RHACS 内のマルチテナンシーは、クラスターまたは namespace によってスコープ指定されたリソースに対してのみ実現できます。

18.7.2. namespace ごとのマルチテナンシーの例

RHACS 内のマルチテナンシーの一般的な例としては、ユーザーを特定の namespace に関連付け、特定の namespace へのアクセスのみを許可することが挙げられます。

次の例では、カスタム権限セット、アクセススコープ、およびロールを組み合わせています。このロールが割り当てられたユーザーまたはグループは、自分に対するスコープが指定された特定の namespace またはクラスター内のデプロイメントに関する CVE 情報、違反、および情報のみを表示できます。

手順

1. RHACS ポータルで、**Platform Configuration → Access Control** を選択します。
2. **Permission Sets** を選択します。
3. **Create permission set** をクリックします。
4. 権限セットの **Name** と **Description** を入力します。
5. 次のリソースとアクセスレベルを選択し、**Save** をクリックします。
   - Alert の **READ**
   - Deployment の **READ**
   - DeploymentExtension の **READ**
   - Image の **READ**
   - K8sRole の **READ**
   - K8sRoleBinding の **READ**
   - K8sSubject の **READ**
   - NetworkGraph の **READ**
   - NetworkPolicy の **READ**
   - Secret の **READ**
   - ServiceAccount の **READ**
6. **Access Scopes** を選択します。
7. **Create access scope** をクリックします。
8. アクセススコープの **Name** と **Description** を入力します。

9. **Allowed resources** セクションで、スコープ指定に使用する namespace を選択し、**Save** をクリックします。
10. **Roles** を選択します。
11. **Create role** をクリックします。
12. ロールの **Name** と **Description** を入力します。
13. 以前に作成したロールの **Permission Set** と **Access scope** を選択し、**Save** をクリックします。
14. 必要なユーザーまたはグループにロールを割り当てます。ユーザーまたはグループへのロールの割り当て を参照してください。

注記

このサンプルロールが割り当てられたユーザーの RHACS ダッシュボードのオプションは、管理者が使用できるオプションと比べて、最小限に抑えられています。このユーザーには、関連するページのみが表示されます。

18.7.3. 制限事項

グローバルスコープ を持つリソースでは、RHACS 内でマルチテナンシーを実現できません。

次のリソースはグローバルスコープを持ちます。

- Access
- Administration
- Detection
- Integration
- VulnerabilityManagementApprovals
- VulnerabilityManagementRequests
- WatchedImage
- WorkflowAdministration

これらのリソースは、RHACS Central インスタンス内のすべてのユーザー間で共有され、スコープを指定することはできません。

関連情報

- カスタム権限セットの作成
- カスタムアクセススコープの作成
- カスタムロールの作成

第19章 システムヘルスダッシュボードの使用

Red Hat Advanced Cluster Security for Kubernetes システムヘルスダッシュボードは、Red Hat Advanced Cluster Security for Kubernetes コンポーネントのヘルス関連情報を表示する単一のインターフェイスを提供します。

注記

システムヘルスダッシュボードは、Red Hat Advanced Cluster Security for Kubernetes 3.0.53 以降でのみ使用できます。

19.1. システムヘルスダッシュボードの詳細

ヘルスダッシュボードにアクセスするには、以下を行います。

- RHACS ポータルで、**Platform Configuration → System Health** に移動します。

ヘルスダッシュボードは、次のグループに情報を整理します。

- **クラスターヘルス** - Red Hat Advanced Cluster Security for Kubernetes クラスターの全体的な状態を表示します。
- **脆弱性の定義** - 脆弱性の定義の最終更新時刻を表示します。
- **イメージの統合** - 統合したすべてのレジストリーの状態を表示します。
- **通知機能の統合** - 統合した通知機能 (Slack、メール、Jira、またはその他の同様の統合) の状態を表示します。
- **バックアップ統合** - 統合したバックアッププロバイダーの状態を表示します。

ダッシュボードには、さまざまなコンポーネントの次の状態がリスト表示されます。

- **Healthy** - コンポーネントは機能しています。
- **Degraded** - コンポーネントが一部正常ではありません。この状態は、クラスターが機能していることを意味しますが、一部のコンポーネントは正常ではなく、注意が必要です。
- **Unhealthy** - このコンポーネントは正常ではなく、早急な対応が必要です。
- **Uninitialized** - コンポーネントが、ヘルス評価について Central に報告していません。初期化されていない状態には注意が必要な場合がありますが、多くの場合、コンポーネントは数分後または統合が使用されたときにヘルスステータスを報告します。

クラスターヘルスセクション

Cluster Overview には、Red Hat Advanced Cluster Security for Kubernetes クラスターの状態に関する情報が表示されます。以下に関するヘルス状態を報告します。

- **Collector ステータス** - Red Hat Advanced Cluster Security for Kubernetes が使用する Collector Pod が正常であると報告しているかどうかを示します。
- **Sensor ステータス** - Red Hat Advanced Cluster Security for Kubernetes が使用する Sensor Pod が正常であると報告しているかどうかを示します。
- **Sensor アップグレード** - Central と比較すると、Sensor が正しいバージョンを実行しているかどうかを示します。

- **認証情報の有効期限** - Red Hat Advanced Cluster Security for Kubernetes の認証情報が有効期限に近づいているかどうかを示します。

注記

クラスターが **Uninitialized** 状態の場合は、チェックインするまで、Red Hat Advanced Cluster Security for Kubernetes により保護されているクラスターの数について報告されません。

脆弱性の定義セクション
Vulnerabilities Definition セクションには、脆弱性の定義が最後に更新された時刻と、定義が最新であるかどうかが表示されます。

統合セクション
Image Integrations、**Notifier Integrations**、および **Backup Integrations** の 3 つの統合セクションがあります。**Cluster Health** セクションと同様に、このセクションには、統合が正常ではない場合にその数がリスト表示されます。それ以外の場合は、すべての統合が正常であると報告されます。

注記

Integrations セクションでは、次の条件のいずれかが満たされた場合に、正常な統合が **0** としてリスト表示されます。

- Red Hat Advanced Cluster Security for Kubernetes をサードパーティーのツールと統合していません。
- 一部のツールと統合しましたが、統合が無効になっているか、ポリシー違反を設定していません。

19.2. 製品の使用状況データの表示

RHACS は、RHACS Sensor から収集されたメトリックに基づいて、保護された Kubernetes ノードと保護されたクラスターの CPU ユニットの数に関する製品使用状況データを提供します。この情報は、レポート用の RHACS 消費データの概算を出すのに便利です。

Kubernetes で CPU ユニットを定義する方法の詳細は、CPU resource units を参照してください。

注記

OpenShift Container Platform は独自の使用状況レポートを提供します。この情報は、セルフマネージド Kubernetes システムでの使用を目的としています。

RHACS は、Web ポータルと API で次の使用状況データを提供します。

- Currently secured CPU units: 最新のメトリック収集時点で、RHACS でセキュリティー保護されたクラスターで使用されている Kubernetes CPU ユニットの数。
- Currently secured node count: 最新のメトリクスコレクション時点で、RHACS でセキュリティー保護された Kubernetes ノードの数。
- Maximum secured CPU units: RHACS セキュアクラスターで使用される CPU ユニットの最大数。時間ごとに測定され、**開始日** と **終了日** で定義された期間について集計されます。

- Maximum secured node count: RHACS によって保護された Kubernetes ノードの最大数。時間単位で測定され、**開始日** と **終了日** で定義された期間で集計されます。
- CPU units observation date: 最大確保 CPU ユニット数のデータを収集した日付。
- Node count observation date: 最大確保ノード数データを収集した日。

Sensor は 5 分ごとにデータを収集するため、現在のデータが表示されるまでに少し時間がかかる場合があります。履歴データを表示するには、**開始日** と **終了日** を設定し、データファイルをダウンロードする必要があります。日付範囲には包括的な値が含まれ、タイムゾーンによって異なります。

表示される最大値は、要求された期間における 1 時間ごとの最大値に基づいて計算されます。1 時間ごとの最大値は CSV 形式でダウンロードできます。

注記

表示されるデータは Red Hat に送信されず、Prometheus メトリックとしても表示されません。

手順

1. RHACS ポータルで、**Platform Configuration → System Health** に移動します。
2. **Show product usage** をクリックします。
3. **Start date** と **End date** フィールドで、データを表示する日付を選択します。この範囲は包括的であり、タイムゾーンによって異なります。
4. オプション: 詳細データをダウンロードするには、**Download CSV** をクリックします。

このデータは、**ProductUsageService** API オブジェクトを使用して取得することもできます。詳細は、RHACS ポータルの **Help → API reference** に移動してください。

19.3. RHACS ポータルを使用した診断バンドルの生成

RHACS ポータルのシステムヘルスダッシュボードを使用して、診断バンドルを生成できます。

前提条件

- 診断バンドルを生成するために、**DebugLogs** リソースの **read** 権限がある。

手順

1. RHACS ポータルで、**Platform Configuration → System Health** を選択します。
2. **System Health** ビューヘッダーで、**Generate Diagnostic Bundle** をクリックします。
3. **Filter by clusters** ドロップダウンメニューで、診断データを生成するクラスターを選択します。
4. **Filter by starting time** で、診断データを含める日付および時刻 (UTC 形式) を指定します。
5. **Download Diagnostic Bundle** をクリックします。

19.3.1. 関連情報

- 診断バンドルの生成

第20章 管理イベントページの使用

Red Hat Advanced Cluster Security for Kubernetes (RHACS) を使用すると、管理イベント情報を単一のインターフェイスで表示できます。このインターフェイスを使用すると、重要なイベントの詳細を理解して解釈するのに役立ちます。

20.1. 異なるドメインのイベントログにアクセスする

管理イベントページを表示すると、さまざまなドメインのさまざまなイベントログにアクセスできます。

手順

- RHACS プラットフォームで、**Platform Configuration → Administration Events** に移動します。

20.2. 管理イベントページの概要

管理イベントページは、次のグループに情報を編成します。

- **ドメイン**: イベントが発生した RHACS 内の特定のエリアまたはドメインごとにイベントを分類します。この分類は、イベントのコンテキストを整理して理解するのに役立ちます。
  以下のドメインが含まれます。
  - **Authentication**
  - **General**
  - **Image Scanning**
  - **Integrations**
- **Resource type**: 関連するリソースまたはコンポーネントタイプに基づいてイベントを指定します。
  次のリソースタイプが含まれます。
  - **API Token**
  - **Cluster**
  - **Image**
  - **Node**
  - **Notifier**
- **Level**: イベントの重大度または重要性を示します。
  以下のレベルが含まれます。
  - **Error**
  - **Warning**
  - **正常に接続できる場合**

- **Info**
- **Unknown**

- **Event last occurred at**: イベントが発生した時点のタイムスタンプと日付に関する情報を提供します。これは、問題を診断し、一連のアクションやインシデントを理解するために不可欠なイベントのタイミングを追跡するのに役立ちます。

- **Count** : 特定のイベントが発生した回数を示します。この数値は、問題の頻度を評価するのに役立ちます。複数回発生したイベントは、修正する必要がある、継続する問題を示しています。

各イベントにより、エラーを修正するために必要なことを示唆します。

20.3. 特定のドメインのイベントに関する情報を取得する

管理イベントの詳細を表示すると、その特定のドメインのイベントに関する詳細情報が得られます。これにより、イベントのコンテキストと詳細をより深く理解できます。

手順

- **Administration Events** ページで、ドメインをクリックして詳細を表示します。

20.4. 管理イベントの詳細の概要

管理イベントでは、エラーまたはイベントを説明するログ情報を提供します。

ログには次の情報が提供されます。

- イベントの背景
- エラーを修正するための手順

管理イベントページでは、情報が次のグループに分類されます。

- **Resource type**: 関連するリソースまたはコンポーネントタイプに基づいてイベントを指定します。
  次のリソースタイプが含まれます。
  - **API Token**
  - **Cluster**
  - **Image**
  - **Node**
  - **Notifier**

- **Resource name**: イベントが参照するリソースまたはコンポーネントの名前を指定します。これは、イベントが発生したドメイン内の特定のインスタンスを識別します。

- **event type**: イベントのソースを指定します。現在、Central は、ログステートメントから作成された管理イベントに対応するログイベントを生成します。

- **イベント ID**: 各イベントに割り当てられる英数字で構成された一意の識別子。イベント ID は、時間の経過に伴うイベントの識別、追跡、管理に役立ちます。

- **作成日**: イベントが最初に作成または記録されたときのタイムスタンプと日付を示します。
- **Last occurred at**: イベントが最後に発生したときのタイムスタンプと日付を指定します。これによりイベントのタイミングが追跡され、再発する問題の診断と修正に重要となる可能性があります。
- **Count** : 特定のイベントが発生した回数を示します。この数値は、問題の頻度を評価するのに役立ちます。複数回発生したイベントは、修正する必要がある、継続する問題を示しています。

20.5. 管理イベントの有効期限の設定

日数を指定することで、管理イベントの有効期限を制御できます。これは、イベントを管理し、必要な期間にわたって情報を保持するために重要です。

注記

デフォルトでは、管理イベントは 4 日間保持されます。これらのイベントの保存期間は、作成時間ではなく、最後に発生した時間によって決まります。つまり、イベントが期限切れになり、最後に発生した時間が指定された保存期間を超えた場合にのみ削除されます。

手順

1. RHACS ポータルで、**Platform Configuration → System Configuration** に移動します。管理イベントに対して、以下の設定を指定できます。
   - **Administration events retention days**: 管理イベントを保持する日数。
2. この値を変更するには、**Edit** をクリックして変更を行い、**Save** をクリックします。