



Red Hat Advanced Cluster Security for Kubernetes 4.0

運用

Red Hat Advanced Cluster Security for Kubernetes の運用

法律上の通知

Copyright © 2023 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

本書では、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 コンプライアンスの管理	8
2.1. コンプライアンスダッシュボードの表示	8
2.2. コンプライアンススキャンの実行	8
2.3. コンプライアンススキャン結果の表示	9
2.4. コンプライアンスステータスのフィルタリング	12
2.5. コンプライアンスレポートの生成	12
2.6. サポート対象のベンチマークバージョン	13
第3章 COMPLIANCE OPERATOR の使用	15
3.1. RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES で COMPLIANCE OPERATOR を使用する	15
第4章 セキュリティーリスクの評価	17
4.1. リスクビュー	17
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	17
4.3. リスクの詳細の表示	21
4.4. デプロイの詳細タブ	22
4.5. プロセス検出タブ	23
4.6. プロセスベースラインの使用	25
第5章 受付コントローラーの適用の使用	28
5.1. 受付コントローラーの適用について	28
5.2. 受付コントローラーの適用の有効化	29
5.3. 受付コントローラーの適用の回避	30
5.4. 受付コントローラーの適用の無効化	30
5.5. VALIDATINGWEBHOOKCONFIGURATION YAML ファイルの変更	32
第6章 セキュリティーポリシーの管理	34
6.1. デフォルトのセキュリティーポリシーの使用	34
6.2. 既存のセキュリティーポリシーの変更	35
6.3. ポリシーカテゴリーの作成と管理	36
6.4. カスタムポリシーの作成	37
6.5. セキュリティーポリシーの共有	55
第7章 デフォルトのセキュリティーポリシー	57
7.1. 重大度のセキュリティーポリシー	57
7.2. 重大度の高いセキュリティーポリシー	58
7.3. 重大度が中程度のセキュリティーポリシー	62
7.4. 重大度の低いセキュリティーポリシー	67
第8章 ネットワークポリシーの管理	71
8.1. ネットワークグラフ (2.0 プレビュー)	71
8.2. ネットワークグラフ (2.0 プレビュー) からのポリシーの生成について	78
8.3. ネットワークグラフ (2.0 プレビュー) でのネットワークポリシーの生成	79
8.4. ネットワークグラフ (1.0)	81
8.5. ポリシーの生成について	84
8.6. 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	88

8.7. ネットワークグラフ (2.0 プレビュー) のネットワークベースライニングについて	90
8.8. ネットワークベースラインの使用	91
第9章 クラスター設定の確認	94
9.1. CONFIGURATION MANAGEMENT ビューの使用	94
9.2. KUBERNETES ロールの設定ミスの特定	94
9.3. KUBERNETES シークレットの表示	96
9.4. ポリシー違反の検索	96
9.5. 失敗した CIS コントロールの検索	97
第10章 イメージの脆弱性の調査	98
10.1. イメージのスキャン	98
10.2. イメージの定期的なスキャン	102
10.3. アクティブではないイメージのスキャン	102
10.4. 脆弱性定義のフェッチ	103
10.5. 脆弱性スコアについて	103
10.6. 環境におけるイメージの表示	104
10.7. イメージの DOCKERFILE の表示	104
10.8. 脆弱性のあるコンテナイメージ層の特定	105
10.9. CVE を使用してコンポーネントを導入したイメージ内の DOCKERFILE 行を特定する	105
10.10. ベースイメージのオペレーティングシステムの特定	106
10.11. 言語固有の脆弱性スキャンの無効化	106
10.12. 関連情報	107
第11章 イメージの署名の確認	108
11.1. 署名統合の設定	108
11.2. ポリシーでの署名検証の使用	108
11.3. 署名の検証の実施	109
第12章 脆弱性の管理	110
12.1. 脆弱性管理	110
12.2. 一般的な脆弱性管理タスク	117
12.3. RHCOS ノードホストのスキャン	125
第13章 違反への対応	130
13.1. 違反ビュー	130
13.2. 違反の詳細の表示	130
第14章 デプロイメントコレクションの作成と使用	133
14.1. 前提条件	133
14.2. デプロイメントコレクションについて	133
14.3. デプロイメントコレクションへのアクセス	136
14.4. デプロイメントコレクションの作成	136
14.5. コレクションへのアクセススコープの移行	138
14.6. API を使用したコレクションの管理	139
第15章 検索およびフィルタリング	141
15.1. 検索構文	141
15.2. オートコンプリートの検索	142
15.3. グローバル検索の使用	142
15.4. ローカルページのフィルタリングの使用	143
15.5. 一般的な検索クエリー	143
15.6. 属性の検索	145
第16章 ユーザーアクセスの管理	150

16.1. RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES での RBAC の管理	150
16.2. PKI 認証の有効化	159
16.3. 認証プロバイダーを理解する	161
16.4. アイデンティティプロバイダーの設定	163
第17章 システムヘルスダッシュボードの使用	176
17.1. システムヘルスダッシュボードの詳細	176
17.2. RHACS ポータルを使用した診断バンドルの生成	177

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 コンポーネントについて説明します。

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	Destination
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
Images	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、および選択したクラスター内の1つ以上の名前空間を選択できます。クラスターまたは名前空間が選択されていない場合、ビューは自動的に **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

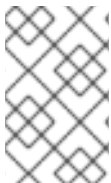
以下のセクションでは、ダッシュボードにある操作可能なウィジェットについて説明します。

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 件が一覧表示されます。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名前空間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、もしく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す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 (クラスターと名前空間) やリスク優先度スコアなどの追加情報が表示されます。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ポリシーの準拠に関する課題について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

コンプライアンス 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。

第2章 コンプライアンスの管理

Red Hat Advanced Cluster Security for Kubernetes を使用すると、コンテナ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、報告が可能です。以下のような業界標準に基づいて、追加設定なしでコンプライアンススキャンを実行できます。

- **Docker** および **Kubernetes** の **CIS Benchmarks** (インターネットセキュリティーセンター)
- **HIPAA** (Health Insurance Portability and Accountability Act)
- **NIST Special Publication 800-190** および **800-53** (米国国立標準技術研究所)
- **PCI DSS** (Payment Card Industry Data Security Standard)
- **OpenSCAP** (Open Security Content Automation Protocol): Compliance Operator がインストールされ、RHACS に結果を提供するように設定されている場合は、OpenShift Container Platform クラスターの RHACS で使用できます。

これらの標準に基づいて環境をスキャンすることで、以下が可能になります。

- 規制コンプライアンスに関するインフラストラクチャーを評価します。
- Docker Engine および Kubernetes オーケストレーターを強化します。
- 環境の全体的なセキュリティーポジションについて理解して管理します。
- クラスター、namespace、およびノードのコンプライアンスステータスの詳細ビューを取得します。

2.1. コンプライアンスダッシュボードの表示

コンプライアンスダッシュボードは、環境内のすべてのクラスター、namespace、およびノードにおけるコンプライアンス標準の概要ビューを提供します。

コンプライアンスダッシュボードにはチャートが含まれており、コンプライアンスに関する潜在的な問題を調査するためのオプションを提供します。単一クラスター、namespace、またはノードのコンプライアンススキャン結果に移動できます。さらに、コンテナ化された環境内のコンプライアンスの状態に関するレポートを生成できます。

手順

- RHACS ポータルで、ナビゲーションメニューから **Compliance** を選択します。



注記

Compliance 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する必要があります。

2.2. コンプライアンススキャンの実行

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お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がチェックされます。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 は環境のデータスナップショットを作成します。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、クラスターで実行している Sensor から

ホストベースのデータを収集します。その後、Central は各コレクター Pod で実行されているコンプライアンスコンテナからより多くのデータを収集します。コンプライアンスコンテナは、環境に関する以下のデータを収集します。

- Docker デーモン、Docker イメージ、および Docker コンテナの設定。
- Docker ネットワークに関する情報。
- Docker、Kubernetes、および OpenShift Container Platform のコマンドライン引数およびプロセス。
- 特定のファイルパスのパーミッション。
- コア Kubernetes および OpenShift Container Platform サービスの設定ファイル。

データ収集が完了すると、Central はデータに対するチェックを実行して結果を判別します。コンプライアンスダッシュボードから結果を表示し、結果に基づいてコンプライアンスレポートを生成することもできます。

注記

コンプライアンススキャンでは、以下のようになります。

- **コントロール** は、監査人が情報システムのコンプライアンスを評価する業界または規制コンプライアンス標準の項目 1 つを表します。Red Hat Advanced Cluster Security for Kubernetes は、1 つ以上のチェックを実行して、単一のコントロールへの準拠の感度をチェックします。
- **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。関連付けられたチェックのいずれかがコントロールに失敗した場合に、コントロールの状態がすべて **Fail** とマークされます。

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 **Compliance** を選択してコンプライアンスダッシュボードを開きます。
2. **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。

2.3. コンプライアンススキャン結果の表示

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は、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。ダッシュボードから直接コンプライアンス違反を表示し、詳細ビューをフィルタリングして、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ことができます。本セクションでは、コンプライアンススキャン結果を表示し、フィルターする方法を説明します。

ショートカットを使用して、クラスター、namespace、およびノード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の上部にあるショートカットを探します。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。

コンプライアンスのステータス

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。
該当なし	Red Hat Advanced Cluster Security for Kubernetes は、該当しないため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、Red Hat Advanced Cluster Security for Kubernetes は 合格 または 不合格 の判断を下すことができませんでした。
エラー	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。

2.3.1. クラスターのコンプライアンスステータスの表示

コンプライアンスダッシュボードから、すべてのクラスターまたは単一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、以下を実行します。
 - a. RHACS ポータルに移動し、ナビゲーションメニューから **Compliance** を選択してコンプライアンスダッシュボードを開きます。
 - b. コンプライアンスダッシュボードで **Clusters** をクリックします。
- 環境内の特定の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、以下を実行します。
 - a. RHACS ポータルに移動し、ナビゲーションメニューから **Compliance** を選択してコンプライアンスダッシュボードを開きます。
 - b. コンプライアンスダッシュボードで、**Passing standards by cluster** ウィジェットを探します。
 - c. このウィジェットで、クラスター名をクリックしてコンプライアンスのステータスを表示します。

2.3.2. namespace のコンプライアンスステータスの表示

コンプライアンスダッシュボードから、すべての namespace または単一の namespace のコンプライアンスステータスを表示できます。

手順

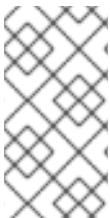
- 環境内の全 namespace のコンプライアンスステータスを表示するには、以下を実行します。
 1. RHACS ポータルに移動し、ナビゲーションメニューから **Compliance** を選択してコンプライアンスダッシュボードを開きます。
 2. コンプライアンスダッシュボードで **Namespaces** をクリックします。
- 環境内の特定の namespace のコンプライアンスステータスを表示するには、以下を実行します。
 1. RHACS ポータルに移動し、ナビゲーションメニューから **Compliance** を選択してコンプライアンスダッシュボードを開きます。
 2. **Namespaces** をクリックし、namespace の詳細ページを開きます。
 3. **Namespaces** テーブルから、namespace をクリックします。右側にサイドパネルが開きます。
 4. 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。

2.3.3. 特定の規格のコンプライアンスステータスの表示

Red Hat Advanced Cluster Security for Kubernetes は、NIST、PCI DSS、NIST、HIPAA、Kubernetes の CIS、Docker コンプライアンス標準の CIS をサポートしています。単一のコンプライアンス標準に関するコンプライアンス制御すべてを表示できます。

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 **Compliance** を選択してコンプライアンスダッシュボードを開きます。
2. コンプライアンスダッシュボードで、**Passing standards across clusters cluster**ウィジェットを探します。
3. このウィジェットで、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。



注記

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、関連情報の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](#)

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の表示

選択した標準の特定コントロールのコンプライアンスステータスを表示できます。

手順

1. RHACS ポータルに移動し、ナビゲーションメニューから **Compliance** を選択してコンプライアンスダッシュボードを開きます。
2. コンプライアンスダッシュボードで、**Passing standards by cluster** ウィジェットを探します。
3. このウィジェットで、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** テーブルから、コントロールをクリックします。右側に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。

2.4. コンプライアンスステータスのフィルタリング

Red Hat Advanced Cluster Security for Kubernetes 検索を使用すると、コンプライアンスダッシュボードからデータをさまざまな組み合わせで簡単にフィルタリングできます。クラスターのサブセット、業界標準、可否のコントロールに注意を向けるために、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範囲を絞り込むことができます。

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 **Compliance** を選択してコンプライアンスダッシュボードを開きます。
2. コンプライアンスダッシュボードで **Clusters**、**Namespaces** または **Nodes** のいずれかを選択して、詳細ページを開きます。
3. 検索バーにフィルター条件を入力してから **Enter** キーを押します。

2.5. コンプライアンスレポートの生成

Red Hat Advanced Cluster Security for Kubernetes を使用すると、レポートを生成して、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。

以下を生成できます。

- ビジネス要素にフォーカスし、PDF 形式のコンプライアンスステータスのチャートや要約を含む **エグゼクティブレポート**。
- 技術的な側面に重点が置かれ、CSV 形式の詳細情報が含まれる **エビデンスレポート**。

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 **Compliance** を選択してコンプライアンスダッシュボードを開きます。
2. コンプライアンスダッシュボードの右上にある **Export** をクリックします。
 - 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せん。

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。この証拠レポートには、コンプライアンス評価に関する詳細情報が含まれており、コンプライアンス監査人、DevOps エンジニア、セキュリティ担当者などの技術的ロールに合わせて調整されています。

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 (標準)	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。
オブジェクトタイプ	オブジェクトの Kubernetes エンティティタイプ。たとえば、 ノード 、 クラスター 、 DaemonSet 、 Deployment 、または StaticPod などです。
オブジェクト名	オブジェクトを一意に識別する Kubernetes システムによって生成された文字列であるオブジェクトの名前。例: gke-setup-dev21380-default-pool-8e086a77-1jfq
Control	コンプライアンス基準に記載されている管理番号。
コントロールの説明	制御が実行されるコンプライアンスチェックの説明。
State	コンプライアンスチェックの合否。たとえば、 Pass または Fail です。
エビデンス	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。
評価時間	コンプライアンススキャンを実行した時刻と日付。

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 は、以下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

第3章 COMPLIANCE OPERATOR の使用

3.1. RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES で COMPLIANCE OPERATOR を使用する

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 からの結果は、RHACS Compliance Dashboard で報告できます。

3.1.1. 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

以下の手順を実行して、Operator をインストールします。

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。

3.1.2. ScanSettingBinding オブジェクトの設定

openshift-compliance namespace で **ScanSettingBinding** オブジェクトを作成し、**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します。



注記

この例では **cis** プロファイルおよび **cis-node** プロファイルを使用しますが、OpenShift Container Platform は追加のプロファイルを提供します。詳細は、関連情報セクションの「コンプライアンスオペレーターについて」を参照してください。

手順

以下のオプションのいずれかを選択します。

- CLI を使用して、YAML ファイルとオブジェクトを作成します。以下に例を示します。
 - a. 次のテキストを使用して、**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
```

- Web コンソールを使用して、次の手順を実行してオブジェクトを作成します。
 - 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。

関連情報

- [Compliance Operator について](#)
- OpenShift Container Platform での [Compliance Operator スキャン](#)

オプション: RHACS のインストール後に Compliance Operator をインストールした場合は、次のオプションのいずれかを実行して、セキュアなクラスターで Sensor を再起動します。

- 以下のコマンドを実行します。

```
$ oc -n stackrox delete pod -lapp=sensor
```

- OpenShift Container Platform Web コンソールで、以下の手順を実行します。
 - a. アクティブなプロジェクトを **stackrox** に変更します。
 - b. **Workloads** → **Pods** に移動します。
 - c. 名前が **sensor-** で始まる Pod を見つけて、**Actions** → **Delete Pod** をクリックします。

検証

これらの手順を実行した後、RHACS でコンプライアンススキャンを実行し、**ocp4-cis** および **ocp4-cis-node** の結果が表示されることを確認します。詳細は、関連情報セクションのコンプライアンススキャンの実行を参照してください。

関連情報

- [RHACS でのコンプライアンススキャンの実行](#)

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展開が一覧表示され、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。一覧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一覧が表示されます。

- **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クス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	コンテナの CPU 制限
CPU Cores Request	コンテナの 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)	コンテナのメモリー制限
Memory Request (MB)	コンテナのメモリー要求
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	ボリ्यूムの宛先
Volume Name	ボリ्यूム名
Volume ReadOnly	書き込み可能なボリ्यूム
Volume Source	ボリ्यूムソース
Volume Type	ボリ्यू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. コンテナ設定セクション

コンテナ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。
- **関連情報**
 - **CPU Request (cores)** コンテナにより要求される CPU の数。

- **CPU Limit (cores):** コンテナが使用できる CPU の最大数。
- **Memory Request (MB):** コンテナによって要求されるメモリーサイズ。
- **Memory Limit (MB):** コンテナ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:** コンテナに特権がある場合に **true** を一覧表示します。

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。
- **引数:** バイナリーで渡された特定の引数。
- **Time:** 指定したコンテナでバイナリーが実行された最新の日時。
- **Pod ID:** コンテナが存在する 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、およびコンテナ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
- ポリシー違反
- コンテナの再起動
- コンテナの終了

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。
- 展開アイコンをクリックして、選択した Pod のコンテナ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

- コンテナが再起動すると、Red Hat Advanced Cluster Security for Kubernetes は以下ようになります。
 - Pod 内のコンテナごとに、アクティブではないコンテナインスタンス 最大 10 個のコンテナの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 **アプリケーション** および **サイドカー** が含まれる場合には、Red Hat Advanced Cluster Security for Kubernetes は最大 10 の **アプリケーション** インスタンスのアクティビティと、最大 10 の **サイドカー** インスタンスを保持します。
 - コンテナ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にプロセスベースラインを使用して、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、ベースラインに一覧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から最初のプロセスインジケ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 のコンテナに複数のコンテナがある場合には、Red Hat Advanced Cluster Security for Kubernetes は各コンテナ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ているものと、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一覧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。

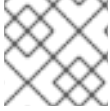
4.6.2. ベースラインへのプロセスの追加

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一覧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一覧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一覧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 受付コントローラーの適用の使用

Red Hat Advanced Cluster Security for Kubernetes は [Kubernetes 受付コントローラー](#) および [OpenShift Container Platform 受付プラグイン](#) と連携して、Kubernetes または OpenShift Container Platform がワークロード (例: デプロイメント、デーモンセットまたはジョブ) を作成する前にセキュリティポリシーを適用することができます。

Red Hat Advanced Cluster Security for Kubernetes 受付コントローラーは、Red Hat Advanced Cluster Security for Kubernetes で設定するポリシーに違反するワークロードを作成できないようにします。Red Hat Advanced Cluster Security for Kubernetes バージョン 3.0.41 以降、受付コントローラーを設定して、ポリシーに違反するワークロードに対する更新を防ぐことができます。

Red Hat Advanced Cluster Security for Kubernetes 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、Red Hat Advanced Cluster Security for Kubernetes 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 Red Hat Advanced Cluster Security for Kubernetes に送信します。Red Hat Advanced Cluster Security for Kubernetes は、設定されたセキュリティ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 で受付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. 受付コントローラーの適用について

受付コントローラーの適用を使用する予定の場合には、以下を考慮してください。

- **API レイテンシー:** 受付コントローラーの適用を使用すると、追加の API 検証が必要になるため、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、受付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。
- 受付コントローラーの適用を使用すると、以下が可能になります。
 - 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。
- 受付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 Red Hat Advanced Cluster Security for Kubernetes Sensor の間に接続性の問題がある場合には、受付コントローラーが失敗する場合があります。この問題を解決するには、受付コントローラーの適用の無効化セクションで説明されているように **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いる、受付コントローラーを有効にしている場合に、Red Hat Advanced Cluster Security for Kubernetes はポリシーに違反するデプロイをブロックしようとします。タイムアウトなど、コンプライアンス違反のデプロイメントが受付コントローラーによって拒否されない場合でも、Red Hat Advanced Cluster Security for Kubernetes は引き続き、レプリカをゼロにスケーリングするなど、他のデプロイ時の強制メカニズムを適用します。

5.2. 受付コントローラーの適用の有効化

Sensor をインストールする場合や、既存のクラスター設定を編集する場合に、**Clusters** ビューで受付コントローラーの適用を有効にすることができます。

手順

1. RHACS ポータルで、**Platform Configuration** → **Clusters** に移動します。
2. 一覧から既存のクラスターを選択するか、**+ New Cluster** を選択します。
3. クラスター設定パネルで、クラスターの詳細を入力します。
4. Red Hat は、受付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ことを計画している場合にのみ、**Configure Admission Controller Webhook to listen on creates** のトグルをオンにすることをお勧めします。
5. Red Hat は、受付コントローラーを使用して更新イベントを実行する予定の場合には **Configure Admission Controller Webhook to listen on updates** トグルをオンにすることを推奨します。
6. Red Hat は、受付コントローラーを使用して Pod 実行および Pod のポート転送イベントを強制することを予定している場合は、**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. 以下のオプションを設定します。
 - **Enforce on Object Creates**: このトグルでは、受付コントロールサービスの動作が制御されます。これを機能させるには、**Configure Admission Controller Webhook to listen on creates** トグルをオンにする必要があります。
 - **Enforce on Object Updates**: この切り替えにより、受付コントロールサービスの動作が制御されます。これを機能させるには、**Configure Admission Controller Webhook to listen on 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

既存クラスターの受付コントローラーを有効にする場合は、以下の変更を加えます。

- **Static Configuration** セクション。YAML ファイルをダウンロードし、Sensor を再デプロイする必要があります。
- **Dynamic Configuration** セクション。Red Hat Advanced Cluster Security for Kubernetes が自動的に Sensor を同期して変更を適用するため、ファイルおよびデプロイメントのダウンロードを省略できます。

10. **Finish** を選択します。

検証

- 新規クラスターを生成された YAML でプロビジョニングした後に、以下のコマンドを実行して受付コントローラーの適用が正しく設定されていることを確認します。

```
$ 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. 受付コントローラーの適用の回避

受付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。受付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、他のユーザーが受付コントローラーをバイパスした理由を理解できるように、問題トラッキングリンクまたはその他の参照をこのアノテーションの値として提供することを推奨します。

5.4. 受付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** ビューから受付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. 一覧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

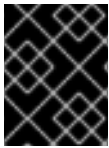
関連するポリシーで適用をオフにすることができ、受付コントローラーに対して適用を回避するように指示できます。

手順

1. RHACS ポータルで、**Platform Configuration** → **Policies** 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。
 - ポリシービューでスクロールダウンし、**Kubernetes Actions: Exec into Pod** ポリシーの横にある電源アイコンを選択して、そのポリシーを無効にします。
 - ポリシービューでスクロールダウンし、**Kubernetes Actions: Port Forward to Pod** ポリシーの横にある電源アイコンを選択して、そのポリシーを無効にします。
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** ビューから受付コントローラーの適用を無効にできます。



重要

Webhook をオフにして受付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. 一覧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** スクリプトを展開して実行します。

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

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



注記

Sensor 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

受付コントローラーを無効にすると、Red Hat Advanced Cluster Security for Kubernetes では **ValidatingWebhookConfiguration** は削除されません。ただし、違反の要求を確認する代わりに、すべての **AdmissionReview** 要求を受け入れます。

ValidatingWebhookConfiguration オブジェクトを削除するには、セキュアなクラスターで以下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

受付コントローラーを機能させるには、Sensor からの初期設定を機能させる必要があります。Kubernetes または OpenShift Container Platform はこの設定を保存し、すべての受付制御サービスレプリカが他のノードに再スケジュールされている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には、受付コントローラーは設定済みのデプロイ時のポリシーをすべて適用します。

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、受付コントローラーの適用は、収集された情報が不完全な場合でも、タイムアウトが期限切れになる前に収集されて利用可能な情報に基づいて引き続き機能します。
- 変更が受付コントロールサービスに伝播されないため、RHACS ポータルから受付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。

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
```

受付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場合に、Red Hat Advanced Cluster Security for Kubernetes は、受付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実施するように 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 ビューには、デフォルトのポリシーを一覧表示し、各ポリシーで以下のパラメーターが含まれます。

- **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一覧
- **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。

デフォルトのポリシーには事前設定されたパラメーターがあり、以下のようなカテゴリーに属します。

- 異常なアクティビティー
- 暗号通貨マイニング
- DevOps のベストプラクティス
- Kubernetes
- ネットワークツール
- パッケージ管理
- 権限
- セキュリティーのベストプラクティス
- システム変更
- 脆弱性管理

これらのカテゴリーを編集し、独自のカテゴリーを作成できます。

**注記**

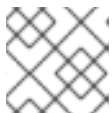
デフォルトのポリシーを削除したり、デフォルトポリシーのポリシー条件を編集したりすることはできません。

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** → **Policies** 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、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を展開します。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

次の方法を使用して、ポリシーカテゴリーを作成および管理できます。

- ポリシーを作成する場合、**ポリシーの詳細** セクションでポリシーカテゴリーを作成します。カテゴリー名は、ポリシーに関連付けられた文字列として保存され、コピーまたは削除できません。
- **Platform Configuration → Policy Management** に移動し、**Policy categories** タブをクリックして、ポリシーカテゴリーを作成、コピー、または削除します。ポリシーカテゴリーはデータベースに保存され、管理できます。このオプションは、Red Hat Advanced Cluster Security Cloud Service または RHACS (PostgreSQL データベース (テクノロジープレビュー) が有効になっている場合) でのみ使用できます。

6.3.1. ポリシー作成時のポリシーカテゴリーの作成

システムポリシービューから新しいポリシーカテゴリーを作成できます。

手順

1. RHACS ポータルで、**Platform Configuration → Policy Management** に移動します。
2. **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** セクションで、**Categories** フィールドに新しいカテゴリー名を入力し、**Create <category>** をクリックします。
5. **Review policy** セクションの見出しをクリックします。
6. **Save** 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作成

RHACS バージョン 3.74 は、Red Hat Advanced Cluster Security Cloud Service または PostgreSQL データベース (テクノロジープレビュー) が有効になっている場合、RHACS でポリシーカテゴリーを作成および管理する新しい方法を提供します。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。詳細については、RHACS ポータルの **Help → API reference** に移動してください。

重要

PostgreSQL のサポートはテクノロジープレビュー機能のみです。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 は実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、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。

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、既存のカテゴリのリストが表示され、カテゴリ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. カテゴリ名を入力し、**Create** をクリックします。

6.3.3. Policy categories タブを使用したポリシーカテゴリの変更

RHACS バージョン 3.74 は、Red Hat Advanced Cluster Security Cloud Service または PostgreSQL データベース (テクノロジープレビュー) が有効になっている場合、RHACS でポリシーカテゴリを作成および管理する新しい方法を提供します。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。この機能の使用方法については、次のセクションの 2 番目の手順を参照してください。

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

重要

PostgreSQL のサポートはテクノロジープレビュー機能のみです。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 は実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、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。

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、既存のカテゴリのリストが表示され、カテゴリ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ポリシーを作成できます。

手順

1. RHACS ポータルで、**Platform Configuration** → **Policies** 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、Red Hat Advanced Cluster Security for Kubernetes を通知プロバイダー (Webhook、Jira、PagerDuty、Splunk など) 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。
 - **Description** ボックスに、ポリシーの詳細を入力します。
 - **Rationale** ボックスにポリシーが存在する理由についての説明を入力します。
 - **Guidance** ボックス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、ポリシーの **Lifecycle stages** および **Event sources(Runtime lifecycle only)** を選択します。
 - **Build**、**Deploy**、または **Runtime**ポリシーを適用する **Lifecycle Stages** 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。
6. **Response method** には、以下のいずれかを選択します。
 - a. **Inform** (違反の一覧に違反を追加する)。
 - b. または **Inform and enforce** (アクションを適用する) を選択します。
 - 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。 **ON** (有効化) を選択してポリシーを適用して違反を報告し、 **OFF** (disable) を選択して違反を報告します。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**: Red Hat Advanced Cluster Security for Kubernetes は、イメージがポリシーの条件に一致すると、継続的インテグレーション (CI) ビルドに失敗します。
 - **Deploy**: Red Hat Advanced Cluster Security for Kubernetes は、ポリシーの条件に一致するデプロイの作成をブロックします。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、Red Hat Advanced Cluster Security for Kubernetes が非準拠の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。
 - **runtime**: Red Hat Advanced Cluster Security for Kubernetes は、ポリシーの条件に一致するすべての Pod を強制終了するか、Pod で実行されたアクションをブロックします。



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。

7. **Next** をクリックします。
8. **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。

9. **Next** をクリックします。

10. **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 to 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** を使用して、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するために、選択したエンティティに対象のポリシーが適用されないことを意味します。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。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 (**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。

11. **Next** をクリックします。

12. **Review policy** セクションで、ポリシー違反をプレビューします。

13. **Save** をクリックします。

関連情報

- [ポリシー条件](#)

6.4.2. リスクビューからのセキュリティポリシーの作成

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。
 - **Regular expressions** 列の **!** は、リストされているフィールドに正規表現のみを使用できることを示しています。
 - **AND、OR** 列の **!** 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ています。
- **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**。名前と値の両方で設定されます。
 - **Add Capabilities、Drop Capabilities、Days since image was created Days since image was last scanned** などの他の意味。



注記

セキュリティーポリシーの作成に論理演算子 **AND**、**OR**、**NOT** を使用するには、Red Hat Advanced Cluster Security for Kubernetes バージョン 3.0.45 以降が必要です。ただし、以前のバージョンでは、正規表現の列に記載されているフィールドに **正規表現** を使用できます。

属性	説明	RHACS バージョン	正規表現	NOT	AND, OR	フェーズ
Namespace	namespace の名前。	3.0.51 以降	✓	✓	✓	デプロイ

属性	説明	RHACS パー ジョン	正規表現	NOT	AND, OR	フェーズ
Image Registry	イメージレジストリーの名前。	すべて	✓	✓	✓	デプロイ
Image Remote	library/nginx など、レジストリー内のイメージのフルネーム。	すべて	✓	✓	✓	デプロイ
Image Tag	イメージの識別子。	すべて	✓	✓	✓	デプロイ
Days since image was created	イメージ作成日からの日数。	すべて	×	×	×	ビルド
Days since image was last scanned	最後のイメージスキャンからの日数。	すべて	×	×	×	ビルド
Dockerfile Line	命令と引数の両方を含む、Dockerfile の特定の行。	すべて	! 値のみ	×	✓	ビルド
Image is NOT Scanned	イメージにはスキャンデータは利用できません。	すべて	×	×	×	ビルド

属性	説明	RHACS パー ジョン	正規表現	NOT	AND, OR	フェーズ
CVSS	Common Vulnerability Scoring System は、スコアが > (指定された CVSS より大きい)、< (指定された CVSS より小さい)、または = (指定された CVSS と等しい) 脆弱性を持つイメージと照合するために使用します。	すべて	×	×	✓	ビルド
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。	すべて	✓	✓	✓	ビルド
CVE	Common Vulnerabilities and Exposures。特定の CVE 番号で使用。	すべて	✓	✓	✓	ビルド
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。	すべて	✓	×	✓	ビルド

属性	説明	RHACS バージョン	正規表現	NOT	AND, OR	フェーズ
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。	3.0.47 以降	✓	✓	✓	ビルド
Environment Variable	名前または値で環境変数を確認します。	すべて	!キーと値のみ	×	✓	デプロイ
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。	すべて	✓	×	✓	デプロイ

属性	説明	RHACS バージョン	正規表現	NOT	AND, OR	フェーズ
Disallowed Image Label	<p>使用されていない Docker イメージラベルの有無を確認します。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。Disallowed 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適用されます。</p>	3.0.40 以降	✓	×	✓	デプロイ

属性	説明	RHACS バージョン	正規表現	NOT	AND, OR	フェーズ
Required Image Label	必要な 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。 Required 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。	3.0.40 以降	✓	×	✓	デプロイ
Required Label	Kubernetes で必要なラベルが存在するかを確認します。	すべて	✓	×	✓	デプロイ
Required Annotation	Kubernetes に必要なアノテーションの有無を確認します。	すべて	✓	×	✓	デプロイ
Volume Name	ストレージの名前。	すべて	✓	✓	✓	デプロイ

属性	説明	RHACS パー ジョン	正規表現	NOT	AND, OR	フェーズ
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、 persist entVolume Claim また は hostPath です。	すべて	✓	✓	✓	デプロイ
Volume Destination	ボリューム がマウント されるパ ス。	すべて	✓	✓	✓	デプロイ
Volume Type	ボリューム の種別を設 定します。	すべて	✓	✓	✓	デプロイ
Writable Volume	書き込み可 能な状態で マウントさ れるボ リューム。	すべて	×	×	×	デプロイ
Protocol	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。	すべて	✓	✓	✓	デプロイ
Port	デプロイメ ントによっ て公開され るポート番 号。	すべて	×	✓	✓	デプロイ
Privileged	特権付きの 実行デプロ イメント。	すべて	×	×	×	デプロイ

属性	説明	RHACS バージョン	正規表現	NOT	AND, OR	フェーズ
Read-Only Root Filesystem	root ファイルシステムで読み取り専用として設定したコンテナ。	すべて	×	×	×	デプロイ
Drop Capabilities	コンテナからドロップする必要がある Linux 機能。たとえば、 CAP_SETPCAP または CAP_NET_RAW です。	すべて	×	×	✓	デプロイ
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、コンテナには追加できない Linux 機能。	すべて	×	×	✓	デプロイ
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。	すべて	✓	✓	✓	ランタイム
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。	すべて	✓	✓	✓	ランタイム

属性	説明	RHACS パー ジョン	正規表現	NOT	AND, OR	フェーズ
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。	すべて	✓	✓	✓	ランタイム
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。	すべて	×	✓	✓	ランタイム
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。	すべて	×	✓	✓	デプロイ
Service Account	サービスアカウントの名前	すべて	✓	✓	✓	デプロイ
Writable Host Mount	リソースが、書き込みパーミッションのあるホストにパスをマウントしている。	すべて	×	×	×	デプロイ
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、プロセス実行が一覧表示されていないデプロイメントを確認します。	すべて	×	×	×	ランタイム

属性	説明	RHACS パー ジョン	正規表現	NOT	AND, OR	フェーズ
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。	すべて	×	✓	×	デプロイ
Container Name	コンテナの名前。	3.0.52 以降	✓	✓	✓	デプロイ
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。	すべて	×	×	✓	デプロイ
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。	すべて	×	×	✓	デプロイ
Container Memory Request	特定のリソース用に予約されているメモリー量を確認します。	すべて	×	×	✓	デプロイ
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。	すべて	×	×	✓	デプロイ

属性	説明	RHACS バージョン	正規表現	NOT	AND, OR	フェーズ
Kubernetes Action	Pod Exec などの Kubernetes アクション の名前。	3.0.55 以降	×	×	!OR のみ	ランタイム
Kubernetes Resource	configmaps または secrets などのアクセスされた Kubernetes リソースの名前。	3.63 以降	×	×	!OR のみ	ランタイム
Kubernetes Resource Name	アクセスされた Kubernetes リソースの名前。	3.63 以降	✓	✓	!OR のみ	ランタイム
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。	3.63 以降	×	×	!OR のみ	ランタイム
Kubernetes User Name	リソースにアクセスしたユーザーの名前。	3.63 以降	✓	✓	!OR のみ	ランタイム
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。	3.63 以降	✓	×	!OR のみ	ランタイム

属性	説明	RHACS バージョン	正規表現	NOT	AND, OR	フェーズ
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 kubectrl	3.63 以降	✓	✓	!OR のみ	ランタイム
Source IP Address	ユーザーがリソースにアクセスした IP アドレス。	3.63 以降	✓	✓	!OR のみ	ランタイム
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。	3.63 以降	×	×	×	ランタイム
Runtime Class	デプロイメントの RuntimeClasses。	3.67 以降	✓	✓	✓	デプロイ
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。	3.68 以降	×	×	×	デプロイ
Liveness Probe	コンテナが liveness プロブを定義するかどうか。	3.69 以降	×	×	×	デプロイ

属性	説明	RHACS バージョン	正規表現	NOT	AND, OR	フェーズ
Readiness Probe	コンテナが readiness プローブを定義するかどうか。	3.69 以降	×	×	×	デプロイ
Replicas	デプロイメントレプリカの数。	3.69 以降	×	✓	✓	デプロイ
Privilege escalation	コンテナプロセスで親プロセスよりも多くの権限を取得できるように設定された場合に、アラートを出します。	3.70 以降	×	×	×	デプロイ
Ingress Network Policy	インGRESS Kubernetes ネットワークポリシーの有無を確認します。	3.70 以降	×	×	✓	デプロイ
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。	3.70 以降	×	×	✓	デプロイ

属性	説明	RHACS バージョン	正規表現	NOT	AND, OR	フェーズ
信頼できるイメージ署名によって検証されない	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。	3.70 以降	×	×	!OR のみ	デプロイ



注記

Red Hat Advanced Cluster Security for Kubernetes バージョン 3.0.44 以前を使用している場合に、**Policy criteria** セクションで指定するポリシー条件は AND が指定されています。これは、指定されたすべてのポリシー条件が一致する場合にのみ違反が発生することを意味します。

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

- 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一覧表示されます。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。
- 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。
- 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一覧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** → **Policies** 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** → **Policies** 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。

Red Hat Advanced Cluster Security for Kubernetes の各セキュリティポリシーには、一意の ID(UUID) と一意の名前があります。ポリシーをインポートすると、Red Hat Advanced Cluster Security for Kubernetes は以下のようにアップロードしたポリシーを処理します。

- 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に一致しない場合は、Red Hat Advanced Cluster Security for Kubernetes が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。Red Hat Advanced Cluster Security for Kubernetes は、インポートされたポリシーを新しい UID に保存します。
 - 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を確認します。ポリシー条件が一致する場合には、Red Hat Advanced Cluster Security for Kubernetes は既存のポリシーを維持し、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に、Red Hat Advanced Cluster Security for Kubernetes はエクスポートされたフィールドをすべて使用します。
- 別の Central インスタンスにインポートする場合には、Red Hat Advanced Cluster Security for Kubernetes はクラスタースコープ、クラスターの除外、通知などの特定のフィールドを省略します。Red Hat Advanced Cluster Security for Kubernetes は、メッセージにこれらの省略フィールドを表示します。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ポリシー

Red Hat Advanced Cluster Security for Kubernetes のデフォルトのセキュリティポリシーは、セキュリティの問題を特定し、環境内のセキュリティ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。	有効
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。	有効

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。	有効
ランタイム	特権コンテナで実行される iptables	特権 Pod が iptables を実行するときにアラートを出します。	有効

7.2. 重大度の高いセキュリティポリシー

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 CVSS ≥ 7	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。	無効
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。	有効

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。	有効
デプロイ	緊急デプロイメントのアノテーション	デプロイで admission.stackrox.io/break-glass:ticket-1234 などの緊急アノテーションを使用して、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場合にアラートを出します。	有効
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。	有効
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。	バージョン 3.72.0 以降ではデフォルトで無効
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	特権モードで実行されるコンテナ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。	有効
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。	無効
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。	有効
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。	有効

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	iptables の実行	誰かが iptables を実行したことを検出します。これは、コンテナ内のネットワーク状態を管理する非推奨の方法です。	有効
ランタイム	Kubernetes アクション: Exec into Pod	コンテナ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。	有効
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。	有効
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。	有効
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。	無効
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。	有効
ランタイム	nmap の実行	ランタイム中に誰かがコンテナ内で nmap プロセスを開始したときにアラートを出します。	有効
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。	有効
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。	無効

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	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。	有効
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。	有効
ランタイム	プロセスターゲットイング 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。	有効
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。	無効
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を検出します。	有効
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。	無効
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。	無効
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。	有効
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。	有効

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	不正なプロセス実行	Kubernetes デプロイメントのコンテナ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。	有効

7.3. 重大度が中程度のセキュリティポリシー

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	Docker CIS 4.4: セキュリティ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パッチを含め、イメージのコンテナをインスタンス化することが重要です。	無効
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。	有効
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が含まれている場合にアラートを出します。	有効
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	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が含まれている場合にアラートを出します。	無効

ライフサイクルステージ	名前	説明	ステータス
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	コンテナ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プロセスが、意図しない権限でコンテナを実行できる場合に発生する可能性があります。	有効
デプロイ	デプロイメントには、1つ以上のインGRESネットワークポリシーが必要	デプロイメントにインGRES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。	無効
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。	無効

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 5.1: 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システムです。	有効
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。	有効
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。	有効
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、コンテナボリュームを双方向、ホストからコンテナ、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。	有効

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 5.21: 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。	無効
デプロイ	Docker CIS 5.7: 特権ポートがコンテナ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。	有効
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、コンテナは別のネットワークスタック内に配置されず、コンテナのネットワークはコンテナ化されません。その結果、コンテナ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 名前空間は、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見えるホスト名とドメインを設定します。コンテナ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。	有効

ライフサイクルステージ	名前	説明	ステータス
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。	無効
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。	有効
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。	有効
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。	有効
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。	有効
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。	有効
デプロイ	特権付きコンテナー	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。	有効
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。	有効

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	Netcat の実行が検出されました	netcat がコンテナ内で実行されるタイミングを検出します。	有効
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。	有効
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。	有効
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。	有効

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。	有効
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。	無効
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。	有効
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。	無効

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	Docker CIS 4.1: コンテナのユーザーが作成されていることを確認する	コンテナが非 root ユーザーとして実行されていることを確認します。	有効
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。	有効
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。	有効
ビルドまたはデプロイ	Latest tag	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。	有効
ビルドまたはデプロイ	イメージの Red Hat Package Manager	デプロイに 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。	有効
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。	無効
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。	有効
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。	無効
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。	有効

ライフサイクルステージ	名前	説明	ステータス
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。	有効
デプロイ	必須のアノテーション: 電子メール	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。	無効
デプロイ	必要なアノテーション: 所有者/チーム	デプロイメントに所有者またはチームのアノテーションがない場合にアラートを出します。	無効
デプロイ	必要なラベル: 所有者/チーム	デプロイメントに所有者またはチームラベルがない場合にアラートを出します。	無効
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。	有効
ランタイム	chkconfig の実行	通常、コンテナでは使用されない ckconfig サービスマネージャーの使用を検出します。	有効
ランタイム	コンパイラツールの実行	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。	有効
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。	有効
ランタイム	シェル管理	シェルの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。	無効
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。	有効

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	systemd の実行	systemd サービスマネージャの使用状況を検出します。	有効

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提供します。

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



注記

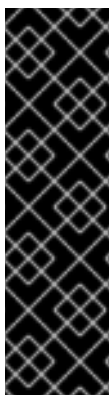
このドキュメントではネットワークグラフ (1.0) について説明します。これは RHACS 4.0 で非推奨となり、将来のリリースで削除される予定です。また、RHACS 3.74 および 4.0 でテクノロジープレビューとして提供されるネットワークグラフ (2.0 プレビュー) についても説明します。

8.1. ネットワークグラフ (2.0 プレビュー)

ネットワークグラフ (2.0 プレビュー) は RHACS 3.74 および 4.0 で提供され、テクノロジープレビュー機能です。

8.1.1. ネットワークグラフ (2.0 プレビュー) 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。



重要

ネットワークグラフ 2.0 はテクノロジープレビュー機能のみです。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 は実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、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。

RHACS は、それぞれのセキュアな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。

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。

外部エンティティ

これらは、クラスターの外部に接続されているエンティティを表します。詳細については、「ネットワークグラフ (2.0 プレビュー) の外部エンティティと接続」を参照してください。

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、選択した名前空間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。詳細については、「バージョン 1.0 または 2.0 プレビューの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。

ネットワークグラフ (2.0 プレビュー) のナビゲーションとユーザーインターフェイス

- グラフ内の項目をクリックすると、コンポーネントに関する追加情報を表示したり、ベースラインにネットワークフローを追加するなどのアクションを実行したりできます。
- 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。legend には、ネットワークグラフ上の namespace、デプロイ、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。

ネットワークグラフ (2.0 プレビュー) 内の外部エンティティと接続

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。

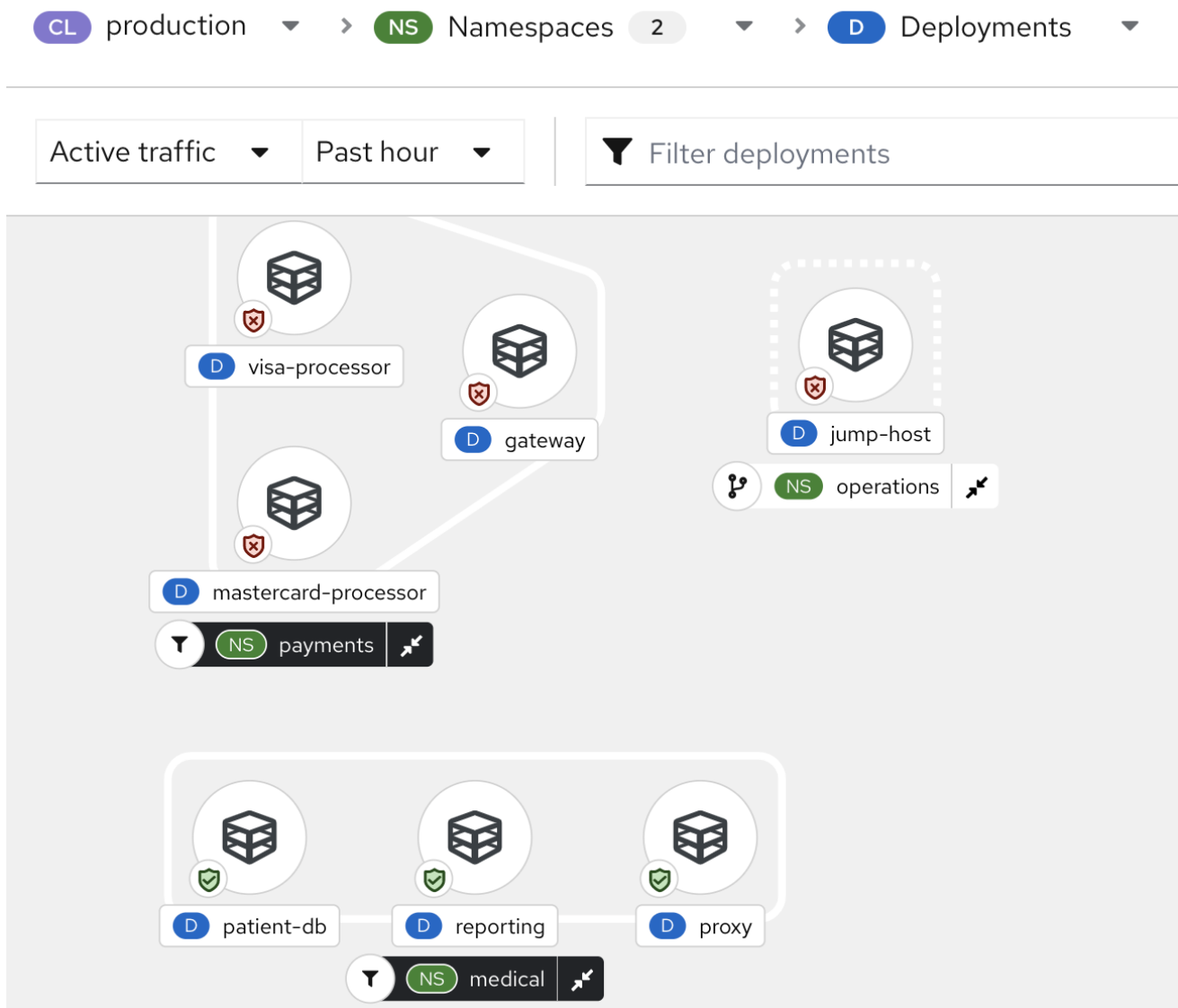
RHACS には、次のクラウドプロバイダーの IP 範囲が含まれています。

- 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、選択した名前空間でのデプロイメントがグラフに表示されます。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。

図8.1 ネットワークグラフの例

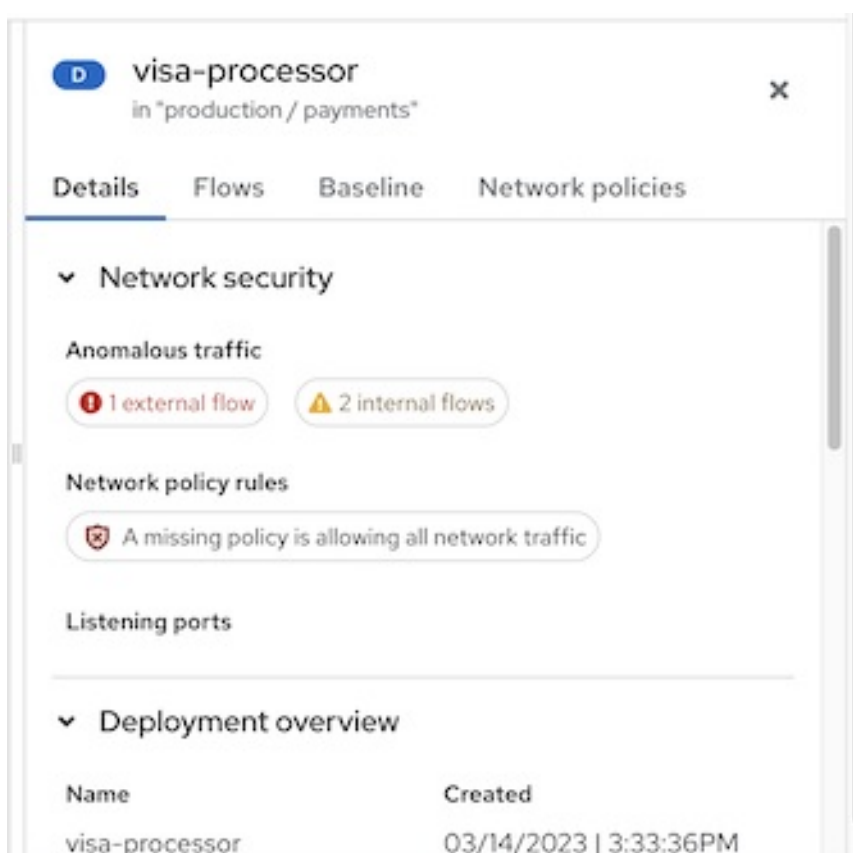


グラフ内の項目をクリックすると、折りたたみ可能なセクションを含む再配置されたサイドパネルに、その項目に関する情報が表示されます。次の項目をクリックできます。

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

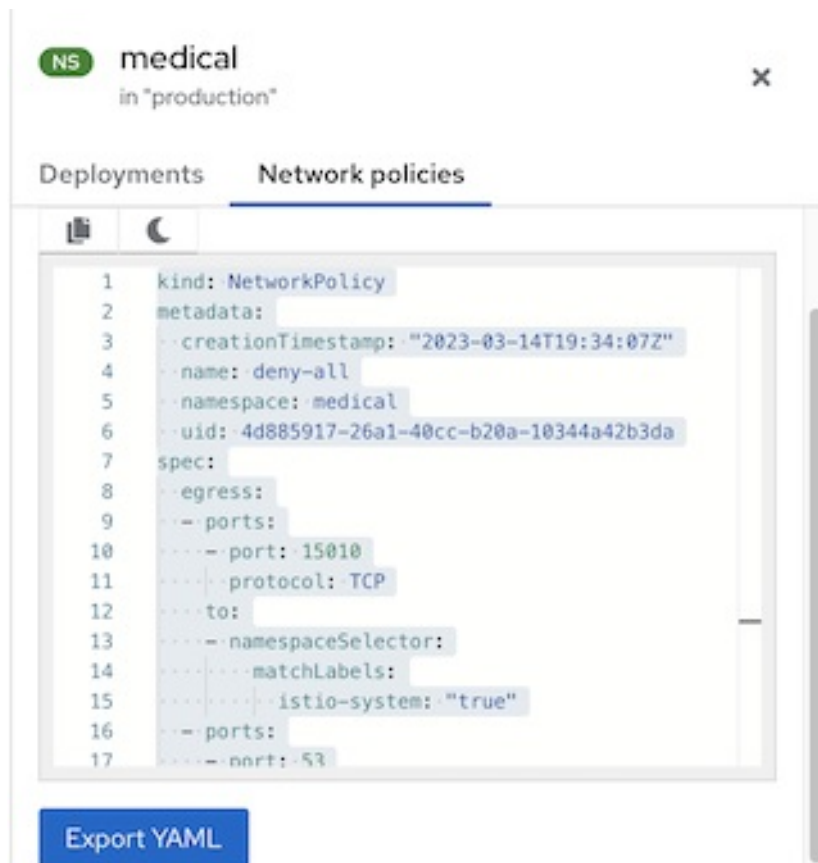
サイドパネルには、選択したグラフ内の項目に基づいた関連情報が表示されます。ヘッダー内の項目名の横にある **D** または **NS** ラベル (この例では postgres) は、それがデプロイメントであるか namespace であることを示します。次の例は、デプロイメントモードを示しています。

図8.2 デプロイメントの例のサイドパネル



namespace モードでは、サイドパネルに検索バーとデプロイメントのリストが含まれます。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。namespace モードでは、サイドパネルに **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



8.1.2. 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 (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
- コンテナ情報
- プロトコルとポート番号を含む、インGRESSおよびエGRESS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 (2.0 preview)**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. **Namespace** 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。

8.1.2.1. ネットワークグラフ (2.0 プレビュー) 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、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 (2.0 preview)** に移動し、ドロップダウンリストからクラスタを選択します。
2. **Namespace** リストをクリックし、検索フィールドを使用して namespace をつけるか、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** 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。

8.1.3. ネットワークグラフ (2.0 プレビュー) 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 (2.0 preview)** 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の上部のバー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。

8.1.4. ネットワークグラフ (2.0 プレビュー) 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。新しいネットワークポリシーのセットの影響をシミュレートするには、ネットワークポリシーシミュレーターを使用します。ネットワークポリシーシミュレーターを使用してポリシーを生成する方法については、「ネットワークグラフ (2.0 プレビュー) でのネットワークポリシーの生成」を参照してください。

手順

1. RHACS ポータルで、**Network Graph (2.0 preview)** 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、**Simulate network policy** を選択します。
4. オプション: **Generate and simulate network policies** をクリックして、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 ファイルを生成します。詳細については、「ネットワークグラフ (2.0 プレビュー) 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. **開く** 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。詳細については、「ネットワークグラフ (2.0 プレビュー) からのネットワークポリシーの生成」を参照してください。



警告

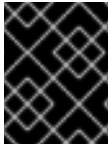
ネットワークポリシーを直接適用すると、アプリケーションの実行で問題が発生する可能性があります。実稼働環境のワークロードに適用する前に、常に開発環境またはテストクラスターでネットワークポリシーをダウンロードし、テストします。

8.2. ネットワークグラフ (2.0 プレビュー) 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。

8.3. ネットワークグラフ (2.0 プレビュー) 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からネットワークポリシーを生成できます。

生成されたポリシーは、現在選択されているクラスターに存在するすべてのデプロイメントに適用されます。また、ベースライン検出期間中に観察されたすべてのネットワークトラフィックも許可されます。

手順

1. RHACS ポータルで、**Network Graph (2.0 preview)**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、**Simulate network policy**を選択します。RHACS は、選択したクラスターに存在するすべてのデプロイメントのポリシーを生成します。
4. オプション: ポートとプロトコルを RHACS 生成ポリシーの範囲に含めない場合は、開いた情報パネルで **Exclude ports & protocols**を選択します。
5. **Generate and simulate network policies**を選択します。生成されたネットワークポリシー設定 YAML ファイルが同じパネルで開き、ネットワークグラフにポリシーの効果が表示されます。



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。

8.3.1. 生成されたポリシーをネットワークグラフ (2.0 プレビュー) に保存する

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、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。

8.3.2. 生成されたポリシーをネットワークグラフ (2.0 プレビュー) でテストする

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、クラスターに適用してテストできます。

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>.yaml" ❶
```

- ❶ 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>.yaml" ❶
```

- ❶ 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。

8.3.3. 生成されたポリシーをネットワークグラフ (2.0 プレビュー) に適用する

ネットワークグラフ (2.0 プレビュー) のネットワークグラフから生成されたネットワークポリシーを適用することはできません。自動化された手順の一部として Kubernetes ネットワークポリシーを適用します。

8.3.4. ネットワークグラフ (2.0 プレビュー) 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 (2.0 preview)** 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。

8.3.5. ネットワークグラフ (2.0 プレビュー) 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 -I \
'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。

8.4. ネットワークグラフ (1.0)

ネットワークグラフ (1.0) は RHACS 4.0 で非推奨となり、将来のリリースで削除される予定です。

8.4.1. ネットワークグラフ (1.0) について

ネットワークグラフは以下の項目を可視化および制御できます。

- Kubernetes ネットワークポリシーで定義済みの、許可されるネットワーク接続。
- namespace とデプロイメント間でアクティブな通信パス。

メニューバーで、情報を表示するクラスターと1つ以上の namespace を選択します。

ネットワークグラフでは、表示する接続のタイプを設定できます。**Flows** セクションで、次を選択します。

- **Active:** アクティブな接続のみを表示する。
- **Allowed:** 許可されるネットワーク接続のみを表示する。
- **All** をクリックすると、アクティブで許可されたネットワーク接続が表示されます。

ネットワークグラフで **Legend** をクリックすると、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。legend には、ネットワークグラフ上の namespace、デプロイ、および接続を表す記号の説明テキストが表示されます。

デプロイメントや namespace など、ネットワークグラフ内の項目をクリックすると、追加情報を表示するウィンドウが開きます。青いバーの矢印を選択すると、ウィンドウを展開したり折りたたんだりできます。

接続の上にマウスを置くと、アクティブな接続、ポート番号、使用中のプロトコルなどのネットワークフローに関する情報が表示されます。デプロイメントの上にマウスを置くと、インGRESSおよびエグレス接続、プロトコル、使用中のポート番号、およびデプロイメント間のネットワークトラフィックの方向に関する情報が表示されます。

許可されるネットワーク接続

RHACS は、それぞれのセキュアな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。

ネットワークグラフは、可能なネットワーク接続を点線として表示します。

実際のネットワークフロー

RHACS は、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は、確認されるネットワークフローを実線として表示します。

ネットワークベースライン

RHACS は既存のネットワークフローを検出し、ベースラインを作成します。

デプロイメントのネットワークベースラインを表示するには、ネットワークグラフでそのデプロイメントを選択します。**ネットワークフローの詳細パネル** には、異常なフローとベースラインフローの両方が表示されます。このパネルから、次のアクションを実行できます。

- **Mark as Anomalous** を選択して、ベースラインからネットワークフローを異常なものとしてマーク付けします。
- **Add to Baseline** を選択して、異常なフローからネットワークフローをベースラインに追加します。

RHACS が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すると、利用可能な更新の数を示す通知がネットワークグラフに表示されます。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。

外部エンティティおよび接続

ネットワークグラフは、マネージドクラスターと外部ソースの間のネットワーク接続を示します。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** ボックスと異なる CIDR アドレスブロックを指します。ただし、自動検出した 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
- Amazon Web Services (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 ブロックは毎日更新されます。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。

8.4.1.1. 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。Red Hat Advanced Cluster Security for Kubernetes 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。

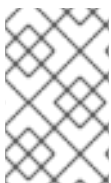
namespace 内のデプロイメントのネットワークポリシーやその他の関連詳細を表示するには、ネットワークグラフで namespace を選択します。

namespace 詳細パネルには、選択した namespace のすべてのデプロイメントが一覧表示されます。次に、詳細パネルでデプロイメントにマウスを移動し、右側に表示される **Navigate to deployment** アイコンを選択すると、デプロイメントの詳細が表示されます。

ネットワークグラフでデプロイメントを直接選択して、その詳細を表示することもできます。デプロイメントの詳細パネルには、**Network Flows** タブ、**Details** タブ、および **Network Policies** タブが含まれます。

各タブを選択して、関連情報を表示できます。

- **Network Flows** タブには、そのデプロイメントのイングレスおよびエグレス接続、プロトコル、および使用中のポート番号に関する情報が表示されます。
- **Details** タブには、オーケストレーターのパネルやアノテーションなど、サービスのデプロイ方法に関する情報が表示されます。
- **Network Policies** タブには、デプロイメントに適用されるすべてのネットワークポリシーに関する情報が表示されます。



注記

イングレスおよびエグレス接続、プロトコル、ポート番号、およびネットワークトラフィックの方向に関する情報を表示するには、Red Hat Advanced Cluster Security for Kubernetes バージョン 3.0.47 以降が必要です。

8.4.2. ネットワークグラフ (1.0) 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に自動検出された CIDR ブロックを表示するように設定したりできます。

手順

1. RHACS ポータルで、**Network Graph (1.0)** に移動し、**Configure CIDR Blocks** を選択します。
2. **Display auto-discovered CIDR blocks in Network Graph** オプションを切り替えて、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の上部のバー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。

3. **CIDR Block Name** と **CIDR Address** を追加して、カスタム CIDR アドレスを追加します。複数ののを追加するには、**Add** アイコンを選択します。
4. **設定の更新** をクリックして変更を保存します。

8.4.3. ネットワークグラフ (1.0) 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。新しいネットワークポリシーのセットの影響をシミュレートするには、ネットワークポリシーシミュレーターを使用します。

手順

1. RHACS ポータルで、**Network Graph (1.0)** に移動します。
2. namespace を 1 つ以上選択します。
3. ネットワークグラフのヘッダーで、**Network Policy Simulator** を選択します。
4. **Upload and simulate network policy YAML** を選択し、提案された 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。
5. 提案されたポリシーをチームと共有するには、**Share YAML** を選択します。
6. ポリシーを直接適用するには、**Apply Network Policies** を選択します。



警告

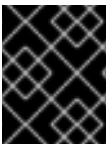
ネットワークポリシーを直接適用すると、アプリケーションの実行で問題が発生する可能性があります。実稼働環境のワークロードに適用する前に、常に開発環境またはテストクラスターでネットワークポリシーをダウンロードし、テストします。

8.5. 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。Red Hat Advanced Cluster Security for Kubernetes (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。

8.5.1. ネットワークグラフ (1.0) からのネットワークポリシーの生成

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。Red Hat Advanced Cluster Security for Kubernetes を使用すると、環境内で実際に観察されたネットワーク通信フローに基づいて、これらの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ビューからネットワークポリシーを生成できます。

生成されたポリシーは、ネットワークグラフに表示されるデプロイメントに適用され、選択した時間に確認されたすべてのネットワークトラフィックを許可します。

手順

1. RHACS ポータルで、**Network Graph** に移動します。
2. 正しいクラスター名がまだ選択されていない場合は、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. namespace を 1 つ以上選択します。
4. 一部のデプロイメントに対してのみポリシーを生成する場合は、**Add one or more deployment filters** フィールドを使用して、デプロイメントをフィルターする条件を追加します。フィルターを追加しない場合には、Red Hat Advanced Cluster Security for Kubernetes はクラスター内のすべてのデプロイメントのポリシーを生成します。
5. 上部のバーのメニューから適切な時間を選択します。選択した時間が短すぎる場合、定期的または低頻度のネットワーク通信が省略されます。
6. **Network Policy Simulator** を選択します。
7. 開いたパネルで、Red Hat Advanced Cluster Security for Kubernetes で生成されたポリシーでポートとプロトコルのスコープを設定しない場合は、**Exclude ports & protocols** を選択します。
8. **Generate and simulate network policies** を選択します。生成されたネットワークポリシー設定 YAML が同じパネルで開き、ネットワークグラフにポリシーの効果が表示されます。

8.5.2. 生成されたポリシーの保存

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、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。

8.5.3. 生成されたポリシーのテスト

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、クラスターに適用してテストできます。

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、以下のコマンドを使用します。

```
$ oc create -f "<generated_file>.yaml" ❶
```

- ❶ 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>.yaml" ❶
```

- ❶ 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。

8.5.4. 生成されたポリシーの適用

RHACS ポータルから生成されたネットワークポリシーを適用できます。

手順

- 生成されたポリシーを RHACS 内からクラスターに直接適用するには、**Apply Network Policies** 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。

8.5.5. 生成されたポリシーの削除

生成されたポリシーを直接適用して削除する場合は、**Network Policy Simulator** パネルの **Revert most recently applied YAML** アイコンを選択します。

手順

1. RHACS ポータルで、**Network Graph (1.0)** に移動します。
2. 正しいクラスター名がまだ選択されていない場合は、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. namespace を1つ以上選択します。
4. **Network Policy Simulator** を選択します。
5. **View active YAMLS** を選択します。
6. **Revert most recently applied YAML** アイコンを再度選択します。

8.5.6. ネットワークグラフ (1.0) 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、生成されたすべてのポリシーを削除できます。

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。

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](#)

8.6. 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

重要

ビルド時のネットワークポリシー生成は、テクノロジープレビュー機能のみです。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 は実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、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。

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GRESS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。

8.6.1. 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: 環境変数を使用](#)
 - [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 generate netpol -h
```

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

```
$ roxctl generate netpol <folder-path> 1
```

1 Kubernetes マニフェストがあるフォルダーのパスを指定します。

roxctl generate netpol 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。

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チェック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。このチェックを支援するために、RHACS を使用して、生成されたネットワークポリシーをシミュレートできます。



注記

Red Hat は、自動化を使用して、ワークロードのデプロイの一部としてクラスターにネットワークポリシーを適用することをお勧めします。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ことにより、GitOps アプローチに従うことができます。これにより、チームは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レビューする機会を得ることができます。

8.7. ネットワークグラフ (2.0 プレビュー) のネットワークベースライニングについて

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。

8.8. ネットワークベースラインの使用

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。



注記

- **ネットワークベースライン** 機能を使用するには、Red Hat Advanced Cluster Security for Kubernetes バージョン 3.0.54 以降を使用する必要があります。
- ベースライン違反のアラートを有効にするには、Red Hat Advanced Cluster Security for Kubernetes バージョン 3.0.56 以降を使用する必要があります。

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。Red Hat Advanced Cluster Security for Kubernetes はネットワークフローを検出すると、ベースラインを作成し、次のガイドラインに従って、検出されたすべてのネットワークフローをベースラインに追加します。

- RHACS は、新しいネットワークアクティビティを検出すると、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。

8.8.1. ネットワークグラフ (2.0 プレビュー) からネットワークベースラインを表示する

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. **Namespace** 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たら、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。

- 個別のエントリを選択し、 をクリックして、**Mark as anomalous** を選択します。

- 複数のエンティティを選択し、**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。

8.8.2. ネットワークグラフ (1.0) からネットワークベースラインを表示する

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. ネットワークグラフで、1つ以上の namespace を選択します。
2. デプロイメントを選択します。
Network Flow 詳細パネルには、異常なフローとベースラインフローの両方が表示されます。
3. 次のいずれかのアクションを実行します。
 - **Mark as Anomalous** を選択して、ベースラインからネットワークフローを異常なものとしてマーク付けします。
 - **Add to Baseline** を選択して、異常なフローからネットワークフローをベースラインに追加します。

8.8.3. ネットワークグラフ (2.0 プレビュー) からネットワークベースラインをダウンロードする

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 Web ポータルで、**Network Graph (2.0 preview)** に移動します。
2. **Namespace** 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たら、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。

8.8.4. ネットワークグラフ (2.0 プレビュー) 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. **Namespace** 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たら、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。

8.8.5. ネットワークグラフ (1.0) 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、違反をトリガーするように RHACS を設定できます。



注記

ベースライン違反のアラートを有効にするには、RHACS バージョン 3.0.56 以降が必要です。

手順

1. ネットワークグラフで、デプロイメントを選択します。
2. ネットワークフローの詳細パネルで **Baseline Settings** を選択します。
3. **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 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、名前空間、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とそれらの間の接続を視覚化するのに役立ちます。

9.1. 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。

9.2. 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。

9.2.1. Kubernetes ロールの割り当ての検索

Configuration Management ビューを使用して、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します。

手順

1. RHACS ポータルに移動し、左側のナビゲーションメニューから **Configuration Management** をクリックします。
2. **Configuration Management** ビューのヘッダーから **RBAC Visibility and Configuration** → **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MISSIONの詳細を表示します。

9.2.2. 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MISSIONを確認します。

手順

1. RHACS ポータルに移動し、左側のナビゲーションメニューから **Configuration Management** をクリックします。
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MISSIONなどの詳細を表示します。

9.2.3. 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 ポータルに移動し、左側のナビゲーションメニューから **Configuration Management** をクリックします。
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 ロールの一覧、付与するパーMISSION、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

ントに付与されていないロールが表示されます。

9.3. 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 ポータルに移動し、左側のナビゲーションメニューから **Configuration Management** をクリックします。
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 の一覧が表示されます。
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。

9.4. 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を表します。
- 最も外側のリングは、特定のカテゴリ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 ポータルに移動し、左側のナビゲーションメニューから **Configuration Management** をクリックします。
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一覧表示されます。
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。

9.5. 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一覧が表示されます。
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一覧表示されます。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10章 イメージの脆弱性の調査

Red Hat Advanced Cluster Security for Kubernetes を使用すると、イメージに対して脆弱性の有無を分析できます。スキャナーは、すべてのイメージレイヤーを分析し、CVE (Common Vulnerabilities and Exposures) 一覧と比較して、既知の脆弱性をチェックします。

スキャナーが脆弱性を見つけた場合は、以下を行います。

- 詳細に分析するために、[Vulnerability Management](#) ビューに表示します。
- リスクに応じて脆弱性をランク付けし、リスク評価のために RHACS ポータルでこれらの脆弱性をハイライトします。
- 有効な [セキュリティポリシー](#) と照合します。

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たコンポーネントを特定します。次のファイルを削除するように最終的なイメージを変更すると、インストールされているコンポーネントまたは脆弱性を特定できない場合があります。

コンポーネント	ファイル
パッケージマネージャー	<ul style="list-style-type: none"> ● <code>/etc/alpine-release</code> ● <code>/etc/apt/sources.list</code> ● <code>/etc/lsb-release</code> ● <code>/etc/os-release</code> または <code>/usr/lib/os-release</code> ● <code>/etc/oracle-release</code>、<code>/etc/centos-release</code>、<code>/etc/redhat-release</code>、または <code>/etc/system-release</code> ● 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	<ul style="list-style-type: none"> ● 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	<ul style="list-style-type: none"> ● <code>dotnet/shared/Microsoft.AspNetCore/</code> ● <code>dotnet/shared/Microsoft.NETCore.App/</code>

10.1. イメージのスキャン

Central はイメージスキャン要求を Scanner に送信します。これらの要求を受信すると、スキャナーは関連するレジストリーからイメージレイヤーをプルし、イメージを確認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識別します。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一覧と比較して、情報を Central に送信します。

Red Hat Advanced Cluster Security for Kubernetes は、別の脆弱性スキャナーと統合することもできます。

スキャナーは、以下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 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** 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10.1 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	<p>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。</p> <p>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については、イメージの脆弱性の検査 を参照してください。</p>
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。

Message	説明
Failed to retrieve metadata from the registry	<p>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。</p> <p>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。</p>
Image out of scope for Red Hat Vulnerability Scanner Certification	<p>Scanner がイメージをスキャンしたが、イメージは古く、Red Hat Scanner Certification の範囲内で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。</p> <div>  <div> <p>重要</p> <p>Red Hat コンテナイメージ 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。</p> </div> </div>

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。

- yum
- microdnf
- apt
- apk
- dpkg
- RPM

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。

- Java
- JavaScript
- Python
- Ruby

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) から、スキャナーは以下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

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。

ディストリビューション	バージョン
Alpine Linux	alpine:v3.2, alpine:v3.3, alpine:v3.4, alpine:v3.5, alpine:v3.6, alpine:v3.7, alpine:v3.8, alpine:v3.9, alpine:v3.10, alpine:v3.11, alpine:v3.12, alpine:v3.13, alpine:v3.14, alpine:v3.15, alpine:v3.16, alpine:edge
Amazon Linux	amzn:2018.03, amzn:2
CentOS	centos:6, centos:7, centos:8
Debian	debian:9, debian:10, debian:11, debian:unstable
Red Hat Enterprise Linux (RHEL)	rhel:6, rhel:7, rhel:8, rhel:9
Ubuntu	ubuntu:14.04, ubuntu:16.04, ubuntu:18.04, ubuntu:20.04, ubuntu:21.10, ubuntu:22.04



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。
- Scanner は、以下のイメージの脆弱性も特定します。ただし、脆弱性ソースはベンダーで更新されなくなりました。

ディストリビューション	バージョン
Debian	debian:8
Ubuntu	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

10.2. イメージの定期的なスキャン

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、アクティブではない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スキャンの結果をフェッチし、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。

10.3. アクティブではないイメージのスキャン

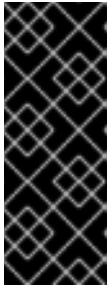
Red Hat Advanced Cluster Security for Kubernetes (RHACS) は、4 時間ごとにアクティブな (デプロイされた) イメージをすべてスキャンし、イメージスキャンの結果を更新して最新の脆弱性定義を反映します。

非アクティブな (デプロイメントされていない) 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 > Dashboard**に移動します。
2. **Dashboard** ビューヘッダーで、**IMAGES** を選択します。
3. **MANAGE WATCHES** をクリックして、監視されたイメージのスキャンを管理します。

4. **MANAGE WATCHED IMAGES** ダイアログで、スキャンを有効にする非アクティブなイメージの名前を入力します。
イメージ ID ではなく、イメージの名前を入力していることを確認してください。イメージ名は、レジストリーで始まり、タグで終わる完全修飾イメージ名です。例:
docker.io/vulnerables/cve-2017-7494:latest。
5. **ADD IMAGE** を選択します。その後、RHACS はイメージをスキャンして、エラーまたは成功のメッセージが表示されます。
6. (オプション) **REMOVE WATCH** をクリックして、ウォッチリストからイメージを削除します。



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。

7. **RETURN TO IMAGE LIST** を選択して、**IMAGES** ページを表示します。

10.4. 脆弱性定義のフェッチ

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、複数の Linux ディストリビューションと National Vulnerability Database を含むアップストリームソースからの脆弱性定義を組み合わせており、1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** 環境変数を設定して、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。



注記

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。

10.5. 脆弱性スコアについて

RHACS ポータルでは、脆弱性ごとに単一の Common Vulnerability Scoring System (CVSS) ベーススコアを表示します。Red Hat Advanced Cluster Security for Kubernetes では、以下の条件に基づいて CVSS スコアが表示されます。

- CVSS v3 スコアが利用可能な場合には、Red Hat Advanced Cluster Security for Kubernetes はスコアを表示し、**v3** をそのスコアと共に一覧表示します。例: **6.5(v3)**



注記

CVSS v3 スコアは、スキャナーバージョン 1.3.5 以降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には、Red Hat Advanced Cluster Security for Kubernetes は CVSS v2 スコアのみを表示します。(例: **6.5**)。

API を使用して CVSS スコアを取得できます。CVSS v3 情報が Common Vulnerabilities and Exposures (CVE) で利用可能な場合に、応答には CVSS v3 および CVSS v2 の両方の情報が含まれます。

CVE によっては、Red Hat Security Advisory (RHSA) CVSS スコアが、RHACS ポータルに表示される CVSS スコアとは異なる場合があります。この違いの原因として、1つの RHSA に複数の CVE が含まれており、Red Hat は脆弱性が他の Red Hat 製品にどのような影響を与えるかに基づいて異なるスコアを割り当てる可能性があることが挙げられます。

このような場合には、Red Hat Advanced Cluster Security for Kubernetes は以下を行います。

- National Vulnerability Database (NVD) から最も高い CVE を見つけ、RHSA の CVSS スコアとしてそのスコアを表示します。
- RHSA 内の各 CVE を、(NVD からの) 元の CVSS スコアを持つ個別の脆弱性として分類し、それぞれを表示して特定の CVE のポリシーを作成できるようにします。

10.5.1. 関連情報

- [アクティブではないイメージのスキャン](#)
- [roxctl CLI の使用を開始する](#)

10.6. 環境におけるイメージの表示

Red Hat Advanced Cluster Security for Kubernetes を使用すると、クラスター内のすべてのコンテナイメージの詳細を表示できます。

手順

1. RHACS ポータルに移動し、左側のナビゲーションメニューから **Vulnerability Management** をクリックします。
2. クラスター内のすべてのイメージの詳細を表示するには、**Vulnerability Management** ビューヘッダーで **Images** を選択します。

10.7. イメージの DOCKERFILE の表示

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、展開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展開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. 最もリスクの **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。

10.8. 脆弱性のあるコンテナイメージ層の特定

Vulnerability Management ビューを使用して、脆弱なコンポーネントと、そのコンポーネントが表示されるイメージ層を特定します。

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. 最もリスクの **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展開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。

10.9. CVE を使用してコンポーネントを導入したイメージ内の DOCKERFILE 行を特定する

CVE を持つコンポーネントを導入したイメージ内の特定の 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. 最もリスクの **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 details** ビューの **Image Findings** で、CVE が **Observed CVEs**、**Deferred CVEs**、および **False Positive CVEs** タブに一覧表示されます。
4. さらに調べたい CVE を見つけます。**Affected Components** 列で、<number> **Components** リンクをクリックして、CVE の影響を受けるコンポーネントのリストを表示します。このウィンドウでは、次のアクションを実行できます。
 - 特定のコンポーネントの横にある展開アイコンを選択して、CVE を導入したイメージの Dockerfile 行を表示します。CVE に対処するには、Dockerfile のこの行を変更する必要があります。たとえば、コンポーネントをアップグレードできます。
 - コンポーネントの名前をクリックして **Component Summary** ページに移動し、コンポーネントに関する詳細情報を表示します。

10.10. ベースイメージのオペレーティングシステムの特定

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. **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** として一覧表示します。

- 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。

10.11. 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 **kubecttl** 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。

10.12. 関連情報

- CVE (Common Vulnerabilities and Exposures) の詳細は、[Red Hat CVE Database](#) を参照してください。

第11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 受付コントローラーを使用してポリシーを適用し、不正な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 3.70 は、Cosign 署名および Cosign 公開鍵署名の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。
- 署名の検証には、1つ以上の Cosign 公開鍵との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、署名インテグレーションの公開鍵を変更または更新するたびに署名を検証します。

11.1. 署名統合の設定

イメージ署名の検証を実行する前に、最初に RHACS に Cosign 公開鍵を追加する必要があります。

前提条件

- PEM でエンコードされた Cosign 公開鍵がすでに存在する必要があります。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** セクションまで下方向にスクロールし、**Signature** 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** → **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。

11.2. 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. 一覧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](#)

11.3. 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 受付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。

詳細については、[受付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12章 脆弱性の管理

12.1. 脆弱性管理

環境内のセキュリティーの脆弱性は、サービス拒否、リモートコード実行、機密データへの不正アクセスなどの不正なアクションを実行するために攻撃者によって悪用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。

12.1.1. 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。

12.1.1.1. 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 방법으로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。

Red Hat Advanced Cluster Security for Kubernetes は、これらのアセットを以下の構造にグループ化します。

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

Red Hat Advanced Cluster Security for Kubernetes は、既知の脆弱性についてアセットをスキャンし、CVE (Common Vulnerabilities and Exposures) データを使用して既知の脆弱性の影響を評価します。

12.1.1.1.1. アプリケーション脆弱性の表示

Red Hat Advanced Cluster Security for Kubernetes で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** からエンティティを選択します。

12.1.1.1.2. イメージ脆弱性の表示

Red Hat Advanced Cluster Security for Kubernetes で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** ビューヘッダーで、**Images** を選択します。
3. イメージの一覧から、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。

関連情報

- [ローカルページのフィルタリングの使用](#)

12.1.1.1.3. インフラストラクチャーの脆弱性の表示

Red Hat Advanced Cluster Security for Kubernetes を使用して、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、**Application & Infrastructure** → **Cluster** の順に選択します。
3. クラスターの一覧から、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してアクションを実行することを優先します。

12.1.1.1.4. ノードの脆弱性の表示

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. ノードの一覧から、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。

12.1.1.2. 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。

12.1.1.3. 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。

12.1.1.4. 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

関連情報

- [誤検出または延期された CVE のレビュー](#)

12.1.1.4.1. 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** → **Dashboard** に移動します。
2. **Dashboard** ビューヘッダーで、**Images** を選択します。
3. イメージの一覧から、すでに評価したイメージを選択します。
4. **Image findings** セクションで CVE を選択します。
5. アクションを実行する CVE の影響を受けるコンポーネントを選択します。
6. CVE が修正されているコンポーネントのバージョンを確認し、イメージを更新します。

12.1.1.5. リスクの受け入れ


このセクションの手順に従って、Red Hat Advanced Cluster Security for Kubernetes のリスクを受け入れます。

前提条件

- **VulnerabilityManagementRequests** リソースの **書き込み** パーミッションが必要です。

軽減策の有無に関わらずリスクを受け入れるには、以下を実行します。

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、**Images** を選択します。
3. イメージの一覧から、すでに評価したイメージを選択します。
4. アクションを実行する CVE を一覧表示する行を見つけます。
5. 特定した CVE の右側にある  をクリックし、**Defer CVE** をクリックします。
6. CVE を延期する日時を選択します。
7. 選択したイメージタグの CVE またはこのイメージのすべてのタグを延期するかどうかを選択します。
8. 延期の理由を入力します。
9. **Request approval** をクリックします。CVE の右側にある青い情報アイコンを選択し、承認リンクをコピーして組織の延期承認者と共有します。


12.1.1.5.1. 脆弱性を誤検出としてのマーク付け

以下の手順では、脆弱性を誤検出としてマークします。

前提条件

- **VulnerabilityManagementRequests** リソースの **書き込み** パーミッションが必要です。

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、**Images** を選択します。
3. イメージの一覧から、すでに評価したイメージを選択します。
4. アクションを実行する CVE を一覧表示する行を見つけます。
5. 特定した CVE の右側にある  をクリックし、**Defer CVE** をクリックします。
6. CVE を延期する日時を選択します。
7. 選択したイメージタグの CVE またはこのイメージのすべてのタグを延期するかどうかを選択します。
8. 延期の理由を入力します。
9. **Request approval** をクリックします。
10. CVE の右側にある青い情報アイコンを選択し、承認リンクをコピーして組織の延期承認者と共有します。

12.1.1.5.2. 誤検出または延期された CVE のレビュー


以下の手順に従って、誤検出または遅延した CVE を確認します。

前提条件

- **VulnerabilityManagementApprovals** リソースの **書き込み** パーミッションが必要です。

誤検出または延期された CVE を確認できます。

手順

1. ブラウザーまたは RHACS ポータルで承認リンクを開きます。
2. **Vulnerability Management** → **Risk Acceptance** に移動して、CVE を検索します。
3. 脆弱性の範囲およびアクションを確認し、承認するかどうかを決定します。
4. CVE の右端にある  をクリックし、承認のリクエストを承認または拒否します。

12.1.1.6. 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電子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？

12.1.1.6.1. 脆弱性管理レポートのスケジュール

次の手順では、スケジュールされた脆弱性レポートを作成します。

手順

1. RHACS ポータルで、**Vulnerability Management** → **Reporting** に移動します。
 2. **レポートの作成** をクリックします。
 3. **Report name** フィールドにレポートの名前を入力します。
 4. **Repeat report...** ドロップダウンリストから、レポートの毎週または毎月の頻度を選択します。
 5. **On...** ドロップダウンリストから、レポートの曜日を選択します。
 6. オプション: レポートを説明するテキストを **Description** フィールドに入力します。
 7. **CVE fixability type** フィールドで、レポートに含める Common Vulnerabilities and Exposure (CVE) の修正可能性タイプを選択します。
 8. **Show vulnerabilities** ドロップダウンリストで、すべての脆弱性を表示するか、前回の成功したレポート以降に発見された脆弱性のみを表示するかを選択します。
 9. **CVE severities** ドロップダウンリストで、レポートに含める CVE の重大度を選択します。
 10. **Configure report scope** フィールドで、既存のコレクションを選択するか、**Create collection** をクリックして、新しいコレクションを作成します。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。コレクションの詳細については、「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」を参照してください。
-
- ##### 注記
- RHACS リリース 3.74 では、**レポートスコープ** がコレクションに置き換えられました。既存のレポートスコープはコレクションに移行されました。詳細については、「関連情報」セクションの「コレクションへのアクセススコープの移行」を参照してください。
11. レポートを電子メールで送信するには、既存の通知機能を選択するか、新しい電子メール通知機能を作成します。電子メール通知機能の作成の詳細については、「関連情報」セクションの「電子メールプラグインの設定」を参照してください。
 12. **Distribution list** フィールドにレポート受信者の電子メールアドレスを入力します。
 13. **Create** を選択して、レポートを作成およびスケジュールします。


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](#)

12.1.1.6.2. 脆弱性レポートの送信

以下の手順では、脆弱性レポートを送信します。


手順

1. RHACS ポータルで、**Vulnerability Management** → **Reporting** に移動します。
2. レポートの一覧から、レポートを選択します。
3. レポートの右側にある  を選択し、**Run report now** をクリックします。

12.1.1.6.3. 脆弱性レポートの編集

以下の手順では、脆弱性レポートを編集します。


手順

1. RHACS ポータルで、**Vulnerability Management** → **Reporting** に移動します。
2. レポートの一覧から、レポートを選択します。
3. レポートの右側にある  を選択し、**Edit** をクリックします。
4. 必要に応じてレポートを変更します。
5. **Save** をクリックします。

12.1.1.6.4. 脆弱性レポートの削除

以下の手順では、脆弱性レポートを削除します。

手順

1. RHACS ポータルで、**Vulnerability Management** → **Reporting** に移動します。
2. レポートの一覧から、レポートを選択します。
3. レポートの右側にある  を選択し、レポートの **削除** をクリックします。

12.2. 一般的な脆弱性管理タスク

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。以下は、**Vulnerability Management → Dashboard** ビューから実行できるいくつかの一般的なタスクです。

12.2.1. インフラストラクチャーに影響する重大な CVE の検索

Vulnerability Management ビューを使用して、プラットフォームに最適な CVE を特定します。

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. **Vulnerability Management** ビューヘッダーで CVE を選択します。
3. **CVE** ビューで、**Env Impact** 列ヘッダーを選択し、環境の影響に基づいて CVE を降順 (最も高いもの) に配置します。

12.2.2. 最も脆弱なイメージコンポーネントの検索

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. **Vulnerability Management** ビューヘッダーから、**Application & Infrastructure → Components** の順に選択します。
3. **Components** ビューで **CVEs** 列ヘッダーを選択し、CVE 数に基づいてコンポーネントを降順 (一番大きいもの) に配置します。

12.2.3. 脆弱性のあるコンテナイメージ層の特定

Vulnerability Management ビューを使用して、脆弱なコンポーネントと、そのコンポーネントが表示されるイメージ層を特定します。

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. 最もリスクの **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展開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。

12.2.4. 修正可能な CVE のみの詳細表示

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. **Vulnerability Management** ビューヘッダーから、**Filter CVEs → Fixable** の順に選択します。

12.2.5. ベースイメージのオペレーティングシステムの特定

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. **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** として一覧表示します。

- 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。

12.2.6. リスクの高いオブジェクトの特定

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 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、およびエンティティ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。

12.2.7. 最もリスクの高いイメージとコンポーネントの特定

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一覧表示されます。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。

12.2.8. イメージの Dockerfile の表示

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、展開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展開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. 最もリスクの **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。

12.2.9. ノードの脆弱性の識別の無効化

ノードの脆弱性の識別は、デフォルトで有効にされています。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. スキャナーの一覧から **StackRox** スキャナーを選択して詳細を表示します。
4. **Types** から **Node Scanner** オプションを削除します。
5. **Save** を選択します。

12.2.10. アクティブではないイメージのスキャン

Red Hat Advanced Cluster Security for Kubernetes (RHACS) は、4 時間ごとにアクティブな (デプロイされた) イメージをすべてスキャンし、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 (デプロイメントされていない) 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** > **Dashboard** に移動します。
2. **Dashboard** ビューヘッダーで、**IMAGES** を選択します。
3. **MANAGE WATCHES** をクリックして、監視されたイメージのスキャンを管理します。
4. **MANAGE WATCHED IMAGES** ダイアログで、スキャンを有効にする非アクティブなイメージの名前を入力します。

イメージ ID ではなく、イメージの名前を入力していることを確認してください。イメージ名は、レジストリーで始まり、タグで終わる完全修飾イメージ名です。例:
docker.io/vulnerables/cve-2017-7494:latest。

5. **ADD IMAGE** を選択します。その後、RHACS はイメージをスキャンして、エラーまたは成功のメッセージが表示されます。
6. (オプション) **REMOVE WATCH** をクリックして、ウォッチリストからイメージを削除します。



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。

7. **RETURN TO IMAGE LIST** を選択して、**IMAGES** ページを表示します。

12.2.11. 特定の CVE をブロックするポリシーの作成

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、**Add selected CVEs to Policy**(add アイコン) をクリックします。または、リストの CVE にマウスを移動して、右側の **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。

12.2.12. 最近検出された脆弱性の表示

Vulnerability Management ビューの **Recently Detected Vulnerabilities** ウィジェットには、スキャン時間と CVSS スコアに基づいて、スキャンイメージで最近検出された脆弱性の一覧が表示されます。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 の一覧を表示します。CVE の一覧をフィルタリングすることもできます。

12.2.13. 最も一般的な脆弱性の表示

Vulnerability Management ビューの **Mostly Common Vulnerabilities** ウィジェットには、CVSS スコアで配置されたデプロイメントやイメージの最大数に影響を与える脆弱性の一覧が表示されます。

- 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 の一覧を表示します。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。

12.2.14. 最も深刻なポリシー違反があるデプロイメントの特定

Vulnerability Management ビューの **Deployments with most severe policy violations** ウィジェットには、デプロイメントに影響する脆弱性の重大度の一覧が表示されます。

- 一覧の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、クラスターの名前、デプロイメントが存在する namespace、失敗したポリシーとその重大度の数が含まれます。
- 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。
- **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 の一覧を表示します。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。

12.2.15. Kubernetes および Istio の脆弱性の多くのクラスターの検索

Vulnerability Management ビューを使用して、環境内の Kubernetes および Istio の脆弱性の多くのクラスターを特定します。

Clusters with most K8S & Istio Vulnerabilities ウィジェットには、各クラスターの Kubernetes と Istio の脆弱性の数で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。

手順

1. 一覧からクラスターの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 Details** セクションには、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、名前空間、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一覧および修正可能な CVE の一覧が含まれます。
 - **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を選択して、詳細情報を表示できます。
2. 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一覧を表示します。

12.2.16. ノードの脆弱性の特定

Vulnerability Management ビューを使用して、ノードの脆弱性を特定できます。特定された脆弱性には、以下のような脆弱性が含まれます。

- コア Kubernetes コンポーネント。
- コンテナランタイム (Docker、CRI-O、runC、および containerd)。



注記

- Red Hat Advanced Cluster Security for Kubernetes は以下のオペレーティングシステムの脆弱性を特定できます。
 - Amazon Linux 2
 - CentOS
 - Debian
 - Garden Linux (Debian 11)
 - Red Hat Enterprise Linux CoreOS (RHCOS)
 - Red Hat Enterprise Linux (RHEL)
 - Ubuntu (AWS、Microsoft Azure、GCP、および GKE の特定のバージョン)

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 **Nodes** を選択すると、ノードに影響を与えるすべての CVE のリストが表示されます。
3. 一覧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。

12.3. 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](#)

12.3.1. 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 4.10 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。サポートされているマネージドおよびセルフマネージドの OpenShift Container Platform バージョンの詳細

細は、[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-rhel8:4.0.2",
  "imagePullPolicy": "IfNotPresent",
  "name": "node-inventory",
  "ports": [
    { "containerPort": 8444, "name": "grpc", "protocol": "TCP" },
    { "mountPath": "/host", "name": "host-root-ro", "readOnly": true },
    { "mountPath": "/tmp", "name": "tmp-volume", "mountPath": "/cache", "name": "cache-volume" }
  ]
}
```

12.3.2. 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 (Compliance コンテナと Node-inventory コンテナ) を作成します。Compliance コンテナは、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 コンテナと Node-inventory コンテナは、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。

12.3.3. 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。

12.3.4. 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12.1 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。

環境変数	説明
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。
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。

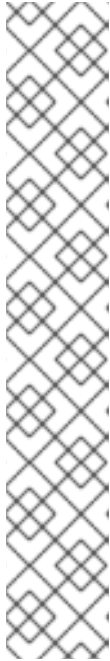
表12.2 コンプライアンス設定

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。

12.3.5. ノードの脆弱性の特定

Vulnerability Management ビューを使用して、ノードの脆弱性を特定できます。特定された脆弱性には、以下のような脆弱性が含まれます。

- コア Kubernetes コンポーネント。
- コンテナランタイム (Docker、CRI-O、runC、および containerd)。



注記

- Red Hat Advanced Cluster Security for Kubernetes は以下のオペレーティングシステムの脆弱性を特定できます。
 - Amazon Linux 2
 - CentOS
 - Debian
 - Garden Linux (Debian 11)
 - Red Hat Enterprise Linux CoreOS (RHCOS)
 - Red Hat Enterprise Linux (RHEL)
 - Ubuntu (AWS、Microsoft Azure、GCP、および GKE の特定のバージョン)

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 **Nodes** を選択すると、ノードに影響を与えるすべての CVE のリストが表示されます。
3. 一覧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。

第13章 違反への対応

Red Hat Advanced Cluster Security for Kubernetes を使用すると、ポリシー違反を表示し、違反の実際の原因にドリルダウンして、修正措置を講じることができます。

Red Hat Advanced Cluster Security for Kubernetes の組み込みポリシーは、脆弱性 (CVE)、DevOps のベストプラクティスの違反、リスクの高いビルドとデプロイのプラクティス、疑わしいランタイム動作など、さまざまなセキュリティの検出を識別します。デフォルトのすぐに使用可能なセキュリティポリシーを使用する場合でも、独自のカスタムポリシーを使用する場合でも、有効なポリシーが失敗すると、Red Hat Advanced Cluster Security for Kubernetes は違反を報告します。

13.1. 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。

検出された違反を確認するには、RHACS ポータルの左側のナビゲーションメニューから **Violations** を選択します。

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。

- **デプロイメント**: デプロイメントの名前。
- **クラスター**: クラスターの名前。
- **名前空間**: デプロイメントの名前空間。
- **ポリシー**: 違反したポリシーの名前。
- **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**: ポリシーカテゴリ。
- **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**)。
- **時間** - 違反が発生した日時。

他のビューと同様:

- 列見出しを選択して、違反を昇順または降順で並べ替えることができます。
- 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。

13.2. 違反の詳細の表示

違反ビューで **Violations** を選択すると、右側に **Violation Details** パネルが開きます。

Violation Details パネルには、複数のタブでグループ化された詳細情報が表示されます。

13.2.1. 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。

13.2.2. 施行タブ

Details パネルの **Enforcement** タブには、選択したポリシー違反に対応して実行された施行アクションのタイプの説明が表示されます。

13.2.3. 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

概要セクションには、次の情報がリストされています。

- **デプロイメント ID:** デプロイメントの英数字 ID。
- **ディプロイメント名:** デプロイメントの名前。
- **デプロイメントタイプ:** デプロイメントのタイプ。
- **クラスター:** コンテナがデプロイされているクラスターの名前。
- **レプリカ:** レプリケートされたデプロイメントの数。
- **名前空間:** デプロイされたクラスターの一意の識別子。
- **更新:** 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。

コンテナ設定セクション

コンテナ設定セクションには、以下の情報がリストされています。

- **イメージ名:** 選択したデプロイメントのイメージの名前。
- **リソース:**
 - **CPU 要求 (コア):** コンテナにより要求されるコアの数。
 - **メモリー要求 (MB):** コンテナによって要求されるメモリーサイズ。
- **ボリューム:**
 - **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。

- **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。

セキュリティコンテキストセクション

コンテナが特権コンテナとして実行されているかどうかを一覧表示します。

- **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

違反を含む namespace 内のすべてのネットワークポリシーをリスト表示します。

13.2.4. 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。

ポリシーの詳細セクション

ポリシーの詳細セクションには、次の情報が一覧表示されます。

- **Id**: ポリシーの数値識別子。
- **名前**: ポリシーの名前。
- **説明**: ポリシーアラートの内容の詳細な説明。
- **理論的根拠**: 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。
- **修正**: 違反を修正する方法に関する提案。
- **有効**: ポリシーが有効かどうかを示します。
- **カテゴリ**: ポリシーのポリシーカテゴリ。
- **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**)。
- **重大度** - 違反のリスクレベル。

ポリシー条件セクション

ポリシーのポリシー条件を一覧表示します。

第14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。詳細については、「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。

14.1. 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については、「関連情報」の「RHACS での RBAC の管理」を参照してください。

14.2. 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。

▼ Collection rules 3

Deployments with names matching

An exact value of ▼ reporting

A regex value of ▼ .*-db

in

Namespaces with labels matching exactly

kubernetes.io/metadata.name=medical

in


Clusters with names matching

An exact value of ▼ production

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。

14.3. 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。

14.4. 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。詳細については、「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。詳細については、「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。

14.4.1. 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。正規表現の詳細については、「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。正規表現の詳細については、「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. 名前空間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。

14.4.2. 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. 次のいずれかのアクションを実行します。
 - **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。

14.5. 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す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、レポート設定で使わ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。

14.6. 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。詳細については、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15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。

15.1. 検索構文

検索クエリーは、次の 2 つの部分で設定されています。

- 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** 名前空間で 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、名前空間 **frontend** または **backend** 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** は、**production** クラスターの **frontend** 名前空間で CVE-2018-11776 に違反するすべてのリソースを返します。

- 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、結果には名前空間 **stackrox** および **stix** から的一致が含まれます。

- **!** 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** 名前空間を除くすべての名前空間からの一致が結果に含まれます。

- 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。

15.2. 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。

15.3. 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
- Images
- Namespaces
- 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
- サービスアカウント
- 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。

- Red Hat Advanced Cluster Security for Kubernetes (RHACS) のバックエンドデータベースとしての PostgreSQL。
- Red Hat Advanced Cluster Security クラウドサービス (RHACS クラウドサービス)。

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一覧表示されます。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。

グローバル検索を行うには、RHACS ポータルで右上の **Search** を選択します。

15.4. 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。

15.5. 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

Query	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

Query	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

Query	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

Query	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

Query	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

Query	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

Query	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、適用された注釈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

Query	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

Query	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

デフォルトの名前空間にデプロイしているユーザーの検索

Query	例
Namespace:default Label:<key_value> または Namespace:default Annotation:<key_value>	Namespace:default Label:team=backend

15.6. 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 クラスターの一意的 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 を使します。
Component	ソフトウェア (daemon、docker)、オブジェクト (イメージ、コンテナ、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	展開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	コンテナイメージの実行時に使用するよう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、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の最大量。
Memory Request (MB)	特定のリソース用に予約されるメモリの最小量。
Namespace	namespace の名前。
Namespace ID	デプロイメントに含まれる名前空間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汚染のタイプ。
Taint Key	現在ノードに適用されている汚染のキー。
Taint Value	現在ノードに適用されている汚染の許容値。
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16章 ユーザーアクセスの管理

16.1. RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES での RBAC の管理

Red Hat Advanced Cluster Security for Kubernetes (RHACS) には、ロールを設定し、さまざまなユーザーに Red Hat Advanced Cluster Security for Kubernetes へのさまざまなレベルのアクセスを許可するのに使用できるロールベースのアクセス制御 (RBAC) が付属しています。

Red Hat Advanced Cluster Security for Kubernetes 3.63 には、特定の Red Hat Advanced Cluster Security for Kubernetes ユーザーまたはユーザーグループが Red Hat Advanced Cluster Security for Kubernetes と対話する方法、アクセスできるリソース、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出力に転送されます。

- **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 (**読み取り**) および変更 (**書き込み**) 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。

16.1.1. 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含まれています。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。
なし	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出力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。これには、セキュアなクラスターに Sensors を作成する権限セットが含まれます。
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知リクエストを承認するためのアクセスを提供できます。
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知を要求するためのアクセスを提供できます。
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。

16.1.1.1. 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。

16.1.1.2. 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](#)

16.1.1.3. 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一覧から、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。

この手順は、ユーザーまたはグループにロールを割り当てるための一連のステップを示しています。

6. **Value** に、選択したキーに基づいたユーザー ID、名前、電子メールアドレス、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。

16.1.2. 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。

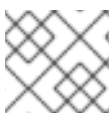
パーミッションセット	説明
Admin	すべてのリソースへの読み取りおよび書き込みアクセスを提供します。
Analyst	すべてのリソースに読み取り専用アクセスを提供します。
Continuous Integration	このアクセス許可セットは、CI (継続的インテグレーション) システムを対象としており、デプロイメントポリシーを適用するのに必要なアクセス許可が含まれています。
None	どのリソースにも読み取りおよび書き込み権限は許可されていません。
Sensor Creator	セキュアなクラスターでセンサーの作成に必要なリソースのパーミッションを提供します。

16.1.2.1. 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一覧が表示されます。



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。

16.1.2.2. 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**
 - **Policy**
 - **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。

16.1.3. 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。

16.1.3.1. 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、クラスターおよび名前空間の一覧、および選択したアクセススコープで許可されているものが表示されます。



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。

16.1.3.2. 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、一覧に表示されているクラスターおよび名前空間の一覧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
- そのクラスター内のすべての名前空間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。

16.1.4. 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	権限の読み取り	書き込み許可
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	<p>次の項目を表示します。</p> <ul style="list-style-type: none"> データ保持、セキュリティ通知、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 	<p>次の項目を編集します。</p> <ul style="list-style-type: none"> データ保持、セキュリティに関する通知、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) セキュアな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	既存のセキュアなクラスターを表示します。	新しいセキュアな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。
Deployment	セキュアな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

リソース	権限の読み取り	書き込み許可
DeploymentExtension	次の項目を表示します。 <ul style="list-style-type: none"> 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 リスク結果 	次の項目を変更します。 <ul style="list-style-type: none"> 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
Image	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。 <ul style="list-style-type: none"> 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 電子メール、Jira、Webhook などの通知システムの既存の統合 	次の項目を変更します。 <ul style="list-style-type: none"> 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	セキュアなクラスター内の Kubernetes RBAC のロールを表示します。	該当なし
K8sRoleBinding	セキュアなクラスター内の Kubernetes RBAC のロールバインディングを表示します。	該当なし
K8sSubject	セキュアなクラスター内の Kubernetes RBAC のユーザーとグループを表示します。	該当なし
Namespace	セキュアなクラスター内の既存の Kubernetes namespace を表示します。	該当なし
NetworkGraph	セキュアな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	セキュアな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。	セキュアな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。

リソース	権限の読み取り	書き込み許可
Node	セキュアなクラスター内の既存の Kubernetes ノードを表示します。	該当なし
ポリシー	既存のシステムポリシーを表示します。	システムポリシーを作成、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	セキュアなクラスター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	セキュアなクラスター内の Kubernetes サービスアカウントを一覧表示します。	該当なし

16.2. 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。

16.2.1. 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

セキュリティのため、セットアップの完了時に、**Minimum access role** を **None** に設定する事を Red Hat は推奨します。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** のル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。

16.2.2. 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>
```

16.2.3. 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。

16.2.4. 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。



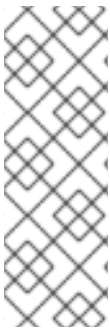
注記

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。

16.3. 認証プロバイダーを理解する

認証プロバイダーは、ユーザー ID のサードパーティーソース (ID プロバイダーや IDP など) に接続し、ユーザー ID を取得し、その ID 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](#)

16.3.1. 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。

16.3.1.1. 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

16.3.1.2. 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。

16.3.1.3. 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

16.3.1.4. 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

16.3.1.5. 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

16.3.1.6. 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

16.3.2. ルール

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。

16.3.3. 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。

16.3.4. 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。

16.4. アイデンティティプロバイダーの設定

16.4.1. 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。

16.4.1.1. 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

- Okta ポータルで、メニューバーから **Applications** を選択します。
- Add Application** をクリックし、**Create New App** を選択します。
- 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。
- Create** をクリックします。
- 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。
- Next** をクリックします。
- SAML Settings** ページで、次のフィールドに値を設定します。
 - シングルサインオン 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。
 - オーディエンス URI (SP エンティティ 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。
 - 属性ステートメント**
 - 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ています。
 - 名前:** 電子メール

- **フォーマット:** 指定なし
 - **値:** 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 **アプリの種類** 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。

16.4.1.2. SAML 2.0 アイデンティティ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 **最小アクセスルール** 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。
 - 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。

16.4.2. 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。

16.4.2.1. 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. ログの近くの左上隅に一覧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。

16.4.2.2. 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。

16.4.2.3. 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) の制限に合わせて設計されています。Red Hat は、初期の統合に対してのみ **Fragment** モードをサポートしており、以降の統合に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。詳細については、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。この設定は安全ではなく、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。

16.4.3. OpenShift Container Platform OAuth サーバーをアイデンティティ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。

16.4.3.1. OpenShift Container Platform OAuth サーバーをアイデンティティ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](#)を参照すること。



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

セキュリティのため、セットアップの完了時に、**Minimum access role** を **None** に設定する事を Red Hat は推奨します。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について説明します。

- **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 アイデンティティプロバイダーの設定](#)
- [信頼できる認証局の追加](#)

16.4.3.2. 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もつだけを設定します。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](#)

16.4.4. 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。

16.4.4.1. 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については、[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。

第17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。

17.1. 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、電子メール、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一覧表示されます。

- **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。

- **コレクターステータス** - 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。
- **センサステータス** - 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。
- **センサーアップグレード** - Central と比較すると、センサー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一覧表示されます。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** として一覧表示されます。

- 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。

17.2. 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には、**DebugLogs** リソースの **read** 権限が必要。

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。

17.2.1. 関連情報

- [診断バンドルの生成](#)

