Red Hat Advanced Cluster Security for Kubernetes 4.4

RHACS Cloud Service

RHACS Cloud Service について

Last Updated: 2024-08-30

Red Hat Advanced Cluster Security for Kubernetes 4.4 RHACS Cloud Service

RHACS Cloud Service について

法律上の通知

概要

RHACS Cloud Service を理解するためのガイダンス。

目次

第1章 RHACS CLOUD SERVICE のサービス説明

1.1. RHACS の概要

Red Hat Advanced Cluster Security for Kubernetes (RHACS) は、エンタープライズ対応の Kubernetes ネイティブコンテナーセキュリティーソリューションです。クラウドネイティブアプリケーションの構築、デプロイ、実行をよりセキュアに行うのに役立ちます。

Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service) は、Kubernetes ネイティブのセキュリティーをサービスとして提供します。Red Hat は RHACS Cloud Service を使用して Central サービスを維持、アップグレード、管理します。

Central サービスには、ユーザーインターフェイス (UI)、データストレージ、RHACS アプリケーションプログラミングインターフェイス (API)、およびイメージスキャン機能が含まれます。Central サービスは Red Hat Hybrid Cloud Console を通じてデプロイします。新しい ACS インスタンスを作成すると、RHACS 用の個別のコントロールプレーンが作成されます。

RHACS Cloud Service を使用すると、Central インスタンスと通信する自己管理型クラスターを保護できます。保護対象のクラスターは、セキュアクラスターと呼ばれ、Red Hat ではなくお客様が管理します。セキュアクラスターサービスには、オプションの脆弱性スキャンサービス、アドミッションコントロールサービス、実行時の監視とコンプライアンスに使用されるデータ収集サービスが含まれます。セキュアクラスターサービスは、保護対象の OpenShift または Kubernetes クラスターにインストールします。

1.2. アーキテクチャー

RHACS Cloud Service は、eu-west-1 と us-east-1 の 2 つのリージョンにわたって Amazon Web Services (AWS) でホストされており、クラウドプロバイダーが提供するネットワークアクセスポイントを使用します。RHACS Cloud Service の各テナントは、高可用性の Egress プロキシーを使用し、3 つのアベイラビリティーゾーンに分散されます。RHACS Cloud Service のシステムアーキテクチャーとコンポーネントの詳細は、Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service) アーキテクチャー を参照してください。

1.3. 請求書

お客様は、Amazon Web Services (AWS) マーケットプレイスで RHACS Cloud Service サブスクリプションを購入できます。サービスコストは、保護対象のコア、またはセキュアクラスターに属するノードの vCPU ごとに時間単位で課金されます。

> **例1.1 サブスクリプションコストの例**
>
> 8 つの vCPU を持つ 5 つの同一ノードを備えた 2 つのセキュアクラスター (Amazon EC2 m7g.2xlarge など) への接続を確立した場合、保護対象のコアの合計数は 80 (2 x 5 x 8 = 80) になります。

1.4. セキュリティーおよびコンプライアンス

Central インスタンス内のすべての RHACS Cloud Service データは、転送中および保存時に暗号化されます。データは、定期的にスケジュールされたバックアップとともに、完全なレプリケーションと高可用性を備えたセキュアなストレージに保存されます。RHACS Cloud Service は、最適なパフォーマンスとデータ常駐要件を満たす機能を保証するクラウドデータセンターを通じて利用できます。

1.4.1. 情報セキュリティーのガイドライン、ロール、責任

Red Hat の情報セキュリティーガイドラインは、NIST サイバーセキュリティーフレームワーク に準拠しており、経営幹部によって承認されています。Red Hat は、世界各地に分散した認定情報セキュリティー専門家の専任チームを維持しています。次のリソースを参照してください。

- FIRST: RH-ISIRT チーム
- TF-CSIRT: RH-ISIRT チーム

Red Hat は、お客様とそのビジネスを保護するために厳格な社内ポリシーと実践を実施しています。これらのポリシーと実践は機密です。さらに、当社は、データプライバシーに関連するものを含め、適用されるすべての法律および規制を遵守します。

Red Hat の情報セキュリティーのロールと責任は、第三者によって管理されません。

Red Hat は、全従業員の業務、企業のエンドポイントデバイス、認証および認可の実践を管理する企業情報セキュリティー管理システム (ISMS) の ISO 27001 認証を維持しています。当社では、Red Hat が採用しているすべてのインフラストラクチャー、製品、サービス、テクノロジーに Red Hat Enterprise Security Standard (ESS) を実装することで、標準化されたアプローチを採用しています。ESS のコピーはリクエストに応じて提供されます。

RHACS Cloud Service は、Amazon Web Services (AWS) でホストされている OpenShift Dedicated のインスタンス上で実行されます。OpenShift Dedicated は、ISO 27001、ISO 27017、ISO 27018、PCI DSS、SOC 2 Type 2、および HIPAA に準拠しています。情報セキュリティーを管理するために、強力なプロセスとセキュリティー制御が業界標準に準拠しています。

RHACS Cloud Service は、OpenShift Dedicated 用に定義されたものと同じセキュリティー原則、ガイドライン、プロセス、および制御に従います。これらの認定は、当社のサービスプラットフォーム、関連する運用、および管理プラクティスがコアセキュリティー要件とどのように一致しているかを実証します。当社は、ビルドパイプラインのセキュリティーを含む、NIST が定義する堅牢なセキュアソフトウェア開発フレームワーク (SSDF) プラクティスに従うことで、これらの要件の多くを満たしています。SSDF コントロールの実装は、すべての製品とサービスに対して、Secure Software Management Lifecycle (SSML) を通じて実装されます。

Red Hat の実績のある経験豊富なグローバルの Site Reliability Engineering (SRE) チームは 24 時間 365 日対応しており、RHACS Cloud Service のホストされたコンポーネントに関連するクラスターのライフサイクル、インフラストラクチャー設定、スケーリング、メンテナンス、セキュリティーパッチ適用、インシデント対応を積極的に管理します。Red Hat SRE チームは、RHACS Cloud Service コントロールプレーンの HA、稼働時間、バックアップ、復元、セキュリティーの管理を担当します。RHACS Cloud Service には、99.95% の可用性 SLA と、電話またはチャットによる 24 時間 365 日の RH SRE サポートが付属しています。

ポリシーの実装、脆弱性管理、OpenShift Container Platform 環境内でのセキュアクラスターコンポーネントの導入など、製品の使用についてはお客様の責任となります。Red Hat SRE チームは、次のような前述のコンプライアンスフレームワークに沿って、テナントデータを含むコントロールプレーンを管理します。

- すべての Red Hat SRE は、バックプレーンを介してデータプレーンクラスターにアクセスし、クラスターへの監査されたアクセスを可能にします。
- Red Hat SRE は、Red Hat レジストリーからのイメージのみをデプロイします。Red Hat レジストリーに投稿されたすべてのコンテンツは、厳格なチェックを受けます。これらのイメージは、セルフ管理のお客様が利用できるイメージと同じです。

- 各テナントには独自の mTLS CA があり、転送中のデータを暗号化してマルチテナント分離を可能にします。追加の分離は、SELinux 制御の namespace とネットワークポリシーにより提供されます。
- 各テナントには独自の RDS データベースインスタンスがあります。

すべての Red Hat SRE と開発者は、厳格なセキュア開発ライフサイクルのトレーニングを受けます。

詳細は、以下を参照してください。

- Red Hat Site Reliability Engineering (SRE) サービス
- Red Hat OpenShift Dedicated
- An Overview of Red Hat's Secure Development Lifecycle (SDL) practices

1.4.2. 脆弱性管理プログラム

Red Hat は、ビルドプロセス中に製品の脆弱性をスキャンし、専任の製品セキュリティーチームが新たに発見された脆弱性を追跡および評価します。Red Hat Information Security は、実行中の環境の脆弱性を定期的にスキャンします。

認定済みの影響度が重大および重要のセキュリティーエラータアドバイザリー (RHSA) と、優先度が緊急で、弊社が優先度高と判断したバグ修正エラータアドバイザリー (RHBA) が利用可能になると、リリースされます。その他の利用可能な修正プログラムと認定パッチはすべて、定期的な更新を通じてリリースされます。重大度が重大または重要の不具合の影響を受けるすべての RHACS Cloud Service ソフトウェアは、修正プログラムが利用可能になるとすぐに更新されます。重大または優先度の高い問題の修復に関する詳細は、Understanding Red Hat's Product Security Incident Response Plan を参照してください。

1.4.3. セキュリティー試験と監査

RHACS Cloud Service は現在、外部のセキュリティー認証やアテステーションを取得していません。

Red Hat 情報リスクおよびセキュリティーチームは、情報セキュリティー管理システム (ISMS) の ISO 27001:2013 認証を取得しました。

1.4.4. システム相互運用性セキュリティー

RHACS Cloud Service は、レジストリー、CI システム、通知システム、ServiceNow や Jira などのワークフローシステム、セキュリティー情報およびイベント管理 (SIEM) プラットフォームとの統合をサポートしています。サポートされている統合の詳細は、インテグレーション ドキュメントを参照してください。カスタムインテグレーションは、API または汎用 Webhook を使用して実装できます。

RHACS Cloud Service は、顧客のサイトと Red Hat 間のすべてのインフライトトラフィックの認証とエンドツーエンドの暗号化の両方に証明書ベースのアーキテクチャー (mTLS) を使用します。VPN は必要ありません。IP 許可リストはサポートされていません。データ転送は mTLS を使用して暗号化されます。セキュア FTP を含むファイル転送はサポートされていません。

1.4.5. 悪意のあるコードの防止

RHACS Cloud Service は Red Hat Enterprise Linux CoreOS (RHCOS) にデプロイされます。RHCOS のユーザー空間は読み取り専用です。さらに、すべての RHACS Cloud Service インスタンスは、実行時に RHACS によって監視されます。Red Hat は、Windows および Mac プラットフォーム向けに、集中管理およびログ記録される、市販のエンタープライズグレードのウイルス対策ソリューションを使用し

ています。Linux ベースのプラットフォーム上のウイルス対策ソリューションは、追加の脆弱性をもたらす可能性があるため、Red Hat のストラテジーには含まれていません。代わりに、プラットフォームを保護するために、組み込みツール (SELinux など) を強化して利用します。

Red Hat は、個々のエンドポイントセキュリティーに SentinelOne と osquery を使用しており、ベンダーから更新が利用可能になるとすぐに更新が行われます。

すべてのサードパーティーの JavaScript ライブラリーがダウンロードされ、ビルドイメージに組み込まれ、公開前に脆弱性がスキャンされます。

1.4.6. システム開発ライフサイクルセキュリティー

Red Hat は安全な開発ライフサイクルのプラクティスに従います。Red Hat 製品のセキュリティープラクティスは、可能な限り、Open Web Application Security Project (OWASP) および ISO12207:2017 に準拠しています。Red Hat は、OWASP プロジェクトの推奨事項とその他の安全なソフトウェア開発プラクティスをカバーし、製品の全体的なセキュリティー体制を強化します。OWASP プロジェクトは選択された CWE の脆弱性に基づいて構築されるため、OWASP プロジェクト分析は Red Hat の自動スキャン、セキュリティーテスト、および脅威モデルに含まれています。Red Hat は、製品の弱点を監視し、問題が悪用されて脆弱性になる前に対処します。

詳細は、以下を参照してください。

- Red Hat Software Development Life Cycle practices
- Security by design: Security principles and threat modeling

アプリケーションは定期的にスキャンされ、製品のコンテナースキャン結果は公開されます。たとえば、Red Hat Ecosystem Catalog サイトでは、**rhacs-main** などのコンポーネントイメージを選択し、**Security** タブをクリックして、ヘルスインデックスとセキュリティー更新のステータスを確認できます。

Red Hat のポリシーの一環として、サポート終了となる依存サードパーティーコンポーネントに対してサポートポリシーとメンテナンスプランが発行されます。

1.4.7. Software Bill of Materials

Red Hat は、コア Red Hat 製品向けの SBOM (ソフトウェアの Bill of Material) ファイルを公開しました。SBOM は、ライセンスと来歴情報を含むソフトウェアコンポーネントと依存関係の、機械可読の包括的なインベントリー (マニフェスト) です。SBOM ファイルは、ソフトウェアアプリケーションとライブラリーのセットに含まれる内容の調達と監査のレビューを確立するのに役立ちます。SBOM は、Vulnerability Exploitability eXchange (VEX) と組み合わせることで、組織が脆弱性リスク評価プロセスに対処するのに役立ちます。これらを組み合わせることで、潜在的なリスクが存在する可能性のある場所 (脆弱なアーティファクトが含まれている場所、およびこのアーティファクトとコンポーネントまたは製品との相関関係) と、既知の脆弱性またはエクスプロイトに対する現在のステータスに関する情報が提供されます。

Red Hat は他のベンダーと協力して、Common Security Advisory Framework (CSAF)-VEX ファイルと相関関係のある有用な SBOM を公開するための特定の要件を定義し、コンシューマーとパートナーにこのデータの使用方法を通知することに取り組んでいます。現時点では、RHACS Cloud Service の SBOM を含む Red Hat が公開している SBOM ファイルは、顧客テスト用のベータ版とみなされており、https://access.redhat.com/security/data/sbom/beta/spdx/ から入手できます。

Red Hat のセキュリティーデータの詳細は、Red Hat セキュリティーデータの将来 を参照してください。

1.4.8. データセンターとプロバイダー

Red Hat は、サブスクリプションサポートサービスの提供に次のサードパーティープロバイダーを使用しています。

- Flexential は、Red Hat カスタマーポータルのデータベースをサポートするために使用される主要なデータセンターである Raleigh Data Center をホストしています。
- Digital Realty は、Red Hat カスタマーポータルのデータベースをサポートするセカンダリーバックアップデータセンターである Phoenix Data Center をホストしています。
- Salesforce は、顧客チケットシステムの背後にあるエンジンを提供します。
- AWS はデータセンターインフラストラクチャーの容量を増強するために使用され、その一部は Red Hat カスタマーポータルアプリケーションのサポートに使用されます。
- Akamai は、Web アプリケーションファイアウォールをホストし、DDoS 保護を提供するために使用されます。
- Iron Mountain は、機密資料の破壊を処理するために使用されます。

1.5. ACCESS CONTROL

ユーザーアカウントは、ロールベースのアクセス制御 (RBAC) を使用して管理されます。詳細は、Managing RBAC in Red Hat Advanced Cluster Security for Kubernetes を参照してください。Red Hat Site Reliability Engineer (SRE) は Central インスタンスにアクセスできます。アクセスは OpenShift RBAC によって制御されます。認証情報は終了時に即座に取り消されます。

1.5.1. 認証プロバイダー

Red Hat Hybrid Cloud Console を使用して Central インスタンスを作成すると、クラスター管理者の認証がプロセスの一部として設定されます。お客様は、統合ソリューションの一部として Central インスタンスへのすべてのアクセスを管理する必要があります。利用可能な認証方法の詳細は、認証プロバイダーについて を参照してください。

RHACS Cloud Service のデフォルトの ID プロバイダーは、Red Hat Single Sign-On (SSO) です。認可ルールは、RHACS Cloud Service を作成したユーザーと、Red Hat SSO で組織管理者としてマークされているユーザーに管理者アクセスを提供するように設定されています。RHACS Cloud Service では、**admin** ログインはデフォルトで無効になっており、SRE によって一時的にのみ有効にできます。Red Hat SSO を使用した認証の詳細は、ACS コンソールへのデフォルトのアクセス を参照してください。

1.5.2. パスワード管理

Red Hat のパスワードポリシーでは、複雑なパスワードの使用が求められます。パスワードには少なくとも 14 文字と、次の文字クラスのうち少なくとも 3 つが含まれている必要があります。

- 10 進数 (0-9)
- 大文字 (A-Z)
- 小文字 (a-z)
- 句読点、スペース、その他の文字

ほとんどのシステムでは 2 要素認証が必要です。

Red Hat は、NIST ガイドライン に従ってパスワードのベストプラクティスに従います。

1.5.3. リモートアクセス

リモートサポートとトラブルシューティングへのアクセスは、次のガイドラインの実装を通じて厳密に制御されます。

- VPN アクセスのための強力な 2 要素認証
- 管理ネットワークと運営ネットワークが分離されたネットワークでは、踏み台ホストを介した追加の認証が必要です。
- すべてのアクセスと管理は暗号化されたセッションを介して実行されます

当社のカスタマーサポートチームは、トラブルシューティングのためのリモートアクセスソリューションとして Bomgar を提供しています。Bomgar セッションは任意で、お客様が開始する必要があり、監視および制御できます。

情報漏洩を防ぐため、ログはセキュリティー情報およびイベント管理 (SIEM) アプリケーションである Splunk を通じて SRE に送信されます。

1.5.4. 規制コンプライアンス

最新のコンプライアンス情報については、OpenShift Dedicated のプロセスとセキュリティーについてを参照してください。

1.6. データ保護

Red Hat は、ログ記録、アクセス制御、暗号化などのさまざまな方法を使用してデータ保護を提供します。

1.6.1. データ保存メディアの保護

Red Hat は、盗難や破壊のリスクから Red Hat のデータとクライアントデータを保護するために、以下の方法を採用しています。

- アクセスロギング
- 自動アカウント終了手続き
- 最小権限の原則の適用

データは、可能で実用的な場合は NIST ガイドラインと Federal Information Processing Standards (FIPS) に従った強力なデータ暗号化を使用して、転送中および保存時に暗号化されます。これにはバックアップシステムも含まれます。

RHACS Cloud Service は、AWS が管理するキー管理サービス (KMS) キーを使用して、Amazon Relational Database Service (RDS) データベース内の保存データを暗号化します。アプリケーションとデータベース間のすべてのデータ、およびシステム間のデータ交換は、転送中に暗号化されます。

1.6.1.1. データの保持と破棄

個人データを含む記録は、法律で義務付けられているとおりに保持されます。法律で義務付けられていない、または合理的なビジネス上の必要性がない記録は安全に削除されます。軍事グレードのツールを使用した安全なデータ破棄要件が操作手順に含まれています。さらに、スタッフは安全な文書破棄施設

を利用できます。

1.6.1.2. 暗号化

Red Hat は、AWS によって毎年ローテーションされる AWS 管理キー を使用します。キーの使用については、AWS KMS キー管理 を参照してください。RDS の詳細は、 Amazon RDS セキュリティー を参照してください。

1.6.1.3. マルチテナンシー

RHACS Cloud Service は、OpenShift Container Platform 上の namespace ごとにテナントを分離します。SELinux は追加の分離を提供します。各顧客には固有の RDS インスタンスがあります。

1.6.1.4. データの所有者

顧客データは、パブリックインターネット上では利用できない暗号化された RDS データベースに保存されます。Site Reliability Engineer (SRE) のみがアクセスでき、アクセスは監査されます。

すべての RHACS Cloud Service システムには、Red Hat 外部 SSO が統合されています。認可ルールは、Cloud Service を作成したユーザーと、Red Hat SSO で組織管理者としてマークされているユーザーに管理者アクセスを提供するように設定されています。RHACS Cloud Service では、管理者ログインはデフォルトで無効になっており、SRE によって一時的にのみ有効にできます。

Red Hat は、RHACS Cloud Service に接続されているセキュアクラスターの数と機能の使用状況に関する情報を収集します。アプリケーションによって生成され、RDS データベースに保存されるメタデータは、顧客が所有します。Red Hat は、トラブルシューティングの目的で、顧客の許可を得た場合にのみデータにアクセスします。Red Hat のアクセスには、監査された権限昇格が必要です。

契約終了時に、Red Hat はリクエストに応じて安全なディスク消去を実行できます。ただし、メディアを物理的に破壊することはできません (AWS などのクラウドプロバイダーはこのオプションを提供していません)。

侵害が発生した場合にデータを保護するために、次のアクションを実行できます。

- クラスター管理ページを使用して、すべてのセキュアクラスターを RHACS Cloud Service から直ちに切断します。
- Access Control ページを使用して、RHACS Cloud Service へのアクセスを直ちに無効にします。
- RHACS インスタンスをすぐに削除すると、RDS インスタンスも削除されます。

AWS RDS (データストア) 固有のアクセス変更は、RHACS Cloud Service SRE エンジニアによって実装されます。

1.7. メトリクスとロギング

1.7.1. サービスメトリクス

サービスメトリクスは内部専用です。Red Hat は、合意されたレベルでサービスを提供および維持します。サービスメトリクスにアクセスできるのは、許可された Red Hat 担当者のみです。詳細は、PRODUCT APPENDIX 4 RED HAT ONLINE SERVICES を参照してください。

1.7.2. カスタマーメトリクス

コアの使用量と容量のメトリクスを Subscription Watch または サブスクリプションページ から利用できます。

1.7.3. サービスロギング

Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service) のコンポーネントのシステムログは、すべて内部専用であり、Red Hat の担当者のみが利用できます。Red Hat は、コンポーネントログへのユーザーアクセスを提供しません。詳細は、PRODUCT APPENDIX 4 RED HAT ONLINE SERVICES を参照してください。

1.8. 更新およびアップグレード

Red Hat は、サービスに影響を与える更新やアップグレードを行う前に、お客様に通知する商業的に合理的な努力を行っています。Central インスタンスに対するサービス更新の必要性とそのタイミングに関する決定は、Red Hat が単独で責任を負います。

Central サービスの更新が発生するタイミングを制御できません。詳細は、PRODUCT APPENDIX 4 RED HAT ONLINE SERVICES を参照してください。Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service) のバージョンへのアップグレードは、サービス更新の一部とみなされます。アップグレードは顧客にとって透過的であり、更新サイトへの接続は必要ありません。

RHACS Cloud Service との互換性を維持するために必要な RHACS Secured Cluster サービスのアップグレードを適時に実施するのはお客様の責任です。

Red Hat では、RHACS Cloud Service に接続されている Secured Cluster の自動アップグレードを有効にすることを推奨しています。

アップグレードバージョンの詳細は、Red Hat Advanced Cluster Security for Kubernetes サポートマトリックス を参照してください。

1.9. 利用可能

可用性と災害回避は、セキュリティープラットフォームにとって非常に重要な側面です。Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service) は、障害に対する多数の保護を複数のレベルで提供します。クラウドプロバイダーの障害の可能性を考慮して、Red Hat はマルチアベイラビリティーゾーンを確立しました。

1.9.1. バックアップおよび障害復旧

RHACS Cloud Service 障害復旧ストラテジーには、データベースのバックアップとカスタマイズが含まれます。これは、Central データベースに保存されているカスタマーデータにも当てはまります。復旧時間はアプライアンスの数とデータベースのサイズによって異なりますが、アプライアンスはクラスター化して分散できるため、適切なアーキテクチャー計画を立てることで事前に RTO を短縮できます。

すべてのスナップショットは、適切なクラウドプロバイダーのスナップショット API を使用して作成され、暗号化されてからセキュアなオブジェクトストレージ (Amazon Web Services (AWS) の場合は S3 バケット) にアップロードされます。

- Red Hat は目標復旧ポイント (RPO) と目標復旧時間 (RTO) を約束しません。詳細は、PRODUCT APPENDIX 4 RED HAT ONLINE SERVICES を参照してください。
- Site Reliability Engineering は、予防措置としてのみバックアップを実行します。バックアップはクラスターと同じリージョンに保存されます。

- Kubernetes のベストプラクティスに沿ったワークロードとともに、マルチアベイラビリティーゾーンのセキュアクラスターをデプロイして、リージョン内で高可用性を確保することを推奨します。

障害復旧計画は少なくとも毎年実行されます。組織全体で BC ライフサイクルが一貫して遵守されるように、Business Continuity Management の標準とガイドラインが整備されています。このポリシーには、少なくとも年に 1 回、または機能計画の大幅な変更時にテストを実行するという要件が含まれています。計画の実行または有効化の後にレビューセッションを実施する必要があり、必要に応じて計画が更新されます。

Red Hat には発電機バックアップシステムがあります。Red Hat の IT 運用システムは、冗長性が確実に機能するように定期的にテストが行われる Tier 3 データセンター機能でホストされています。コンプライアンスを検証するために毎年監査が行われます。

1.10. RHACS CLOUD SERVICE のサポートを受ける

このドキュメントで説明されている手順、または RHACS Cloud Service 全般で問題が発生した場合は、Red Hat カスタマーポータル にアクセスしてください。

カスタマーポータルから、次のアクションを実行できます。

- Red Hat 製品に関するアーティクルおよびソリューションを対象とした Red Hat ナレッジベースの検索またはブラウズ。
- Red Hat サポートに対するサポートケースの送信。
- その他の製品ドキュメントへのアクセス。

クラスターの問題を特定するには、RHACS Cloud Service の Insights を使用できます。Insights により、問題の詳細と、利用可能な場合は問題の解決方法に関する情報が提供されます。

1.11. サービスの削除

Red Hat Hybrid Cloud Console からデフォルトの削除操作を使用して RHACS Cloud Service を削除できます。RHACS Cloud Service Central インスタンスを削除すると、すべての RHACS コンポーネントが自動的に削除されます。削除すると、元に戻すことはできません。

1.12. 価格

サブスクリプション料金の詳細は、PRODUCT APPENDIX 4 RED HAT ONLINE SERVICES を参照してください。

1.13. サービスレベルアグリーメント

Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service) に対して提供されるサービスレベルアグリーメント (SLA) の詳細は、PRODUCT APPENDIX 4 RED HAT ONLINE SERVICES を参照してください。

第2章 RED HAT ADVANCED CLUSTER SECURITY CLOUD SERVICE の責任の概要

このドキュメントでは、RHACS Cloud Service マネージドサービスに関する Red Hat とお客様の責任を説明します。

2.1. RHACS CLOUD SERVICE の責任の共有

Red Hat は RHACS Cloud Service サービス (**Central サービス** とも呼ばれます) を管理しますが、お客様にも一定の責任があります。

リソースまたはアクション	Red Hat の責任	お客様の責任
ホストされたコンポーネント (Central コンポーネントとも呼ばれる)	• プラットフォームモニタリング • ソフトウェアの更新 • 高可用性 • バックアップおよび復元 • セキュリティー • インフラストラクチャー設定 • スケーリング • メンテナンス • 脆弱性管理	• アクセスとアイデンティティーの承認
セキュアクラスター (オンプレミスまたはクラウド)		• ソフトウェアの更新 • バックアップおよび復元 • セキュリティー • インフラストラクチャー設定 • スケーリング • メンテナンス • アクセスとアイデンティティーの承認 • 脆弱性管理

第3章 RED HAT ADVANCED CLUSTER SECURITY CLOUD SERVICE のアーキテクチャー

Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service) のアーキテクチャーと概念を説明します。

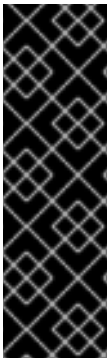
3.1. RED HAT ADVANCED CLUSTER SECURITY CLOUD SERVICE のアーキテクチャーの概要

Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service) は、Red Hat 管理の Software-as-a-Service (SaaS) プラットフォームです。Kubernetes および OpenShift Container Platform のクラスターとアプリケーションを、ビルド、デプロイ、ランタイムのライフサイクル全体にわたって保護できます。

RHACS Cloud Service には、Center for Internet Security (CIS) ベンチマークや National Institute of Standards Technology (NIST) ガイドラインなどの業界標準に基づいた、多くの組み込みの DevOps 強制制御とセキュリティーに重点を置いたベストプラクティスが含まれています。また、既存の DevOps ツールおよびワークフローと統合して、セキュリティーとコンプライアンスを向上させることもできます。

RHACS Cloud Service アーキテクチャー

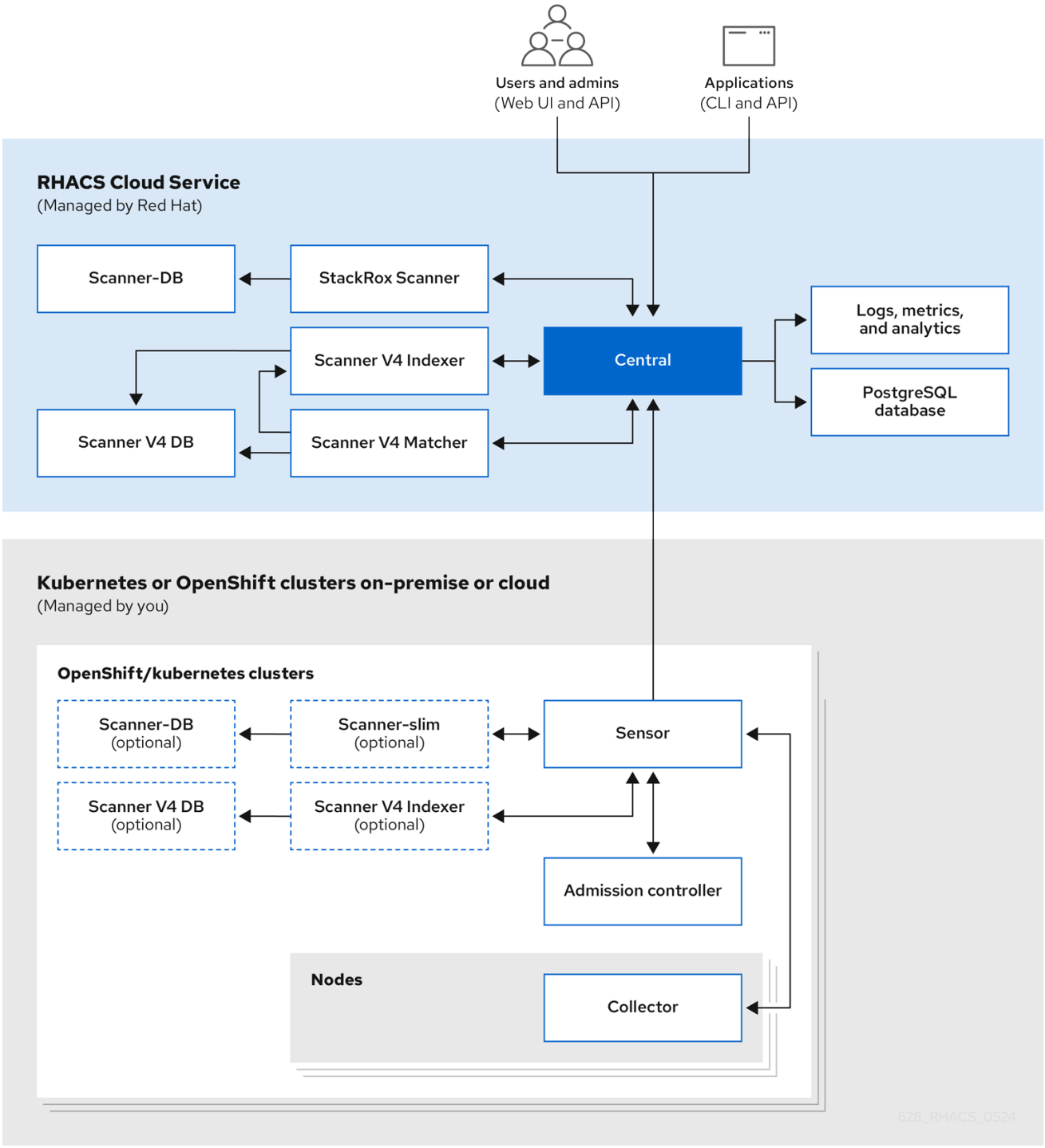
次の図は、StackRox Scanner と、バージョン 4.4 のテクノロジープレビューである Scanner V4 を使用したアーキテクチャーを示しています。Scanner V4 のインストールは任意ですが、インストールするとさらなる利点が得られます。



重要

Scanner V4 はテクノロジープレビューのみの機能です。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat では、実稼働環境での使用を推奨していません。テクノロジープレビューの機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

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



Central サービスには、ユーザーインターフェイス (UI)、データストレージ、RHACS アプリケーションプログラミングインターフェイス (API)、およびイメージスキャン機能が含まれます。Central サービスは Red Hat Hybrid Cloud Console を通じてデプロイします。新しい ACS インスタンスを作成すると、RHACS 用の個別のコントロールプレーンが作成されます。

RHACS Cloud Service を使用すると、Central インスタンスと通信する自己管理型クラスターを保護できます。保護対象のクラスターは、セキュアクラスターと呼ばれ、Red Hat ではなくお客様が管理します。セキュアクラスターサービスには、オプションの脆弱性スキャンサービス、アドミッションコントロールサービス、実行時の監視とコンプライアンスに使用されるデータ収集サービスが含まれます。セキュアクラスターサービスは、保護対象の OpenShift または Kubernetes クラスターにインストールします。

3.2. CENTRAL

Central は、Red Hat によって管理される、RHACS Cloud Service のコントロールプレーンです。このサービスには次のコンポーネントが含まれています。

- **Central**: RHACS アプリケーション管理インターフェイスおよびサービスです。API 対話とユーザーインターフェイス (RHACS ポータル) アクセスを処理します。
- **Central DB**: Central DB は RHACS のデータベースで、すべてのデータ永続性を処理します。現在、PostgreSQL 13 をベースにしています。
- **Scanner V4** (テクノロジープレビュー): バージョン 4.4 以降、RHACS にはコンテナーイメージをスキャンするための Scanner V4 脆弱性スキャナーが含まれています。Scanner V4 は、Clair スキャナーにも利用されている ClairCore 上に構築されています。Scanner V4 には、スキャンに使用される Indexer、Matcher、および Scanner V4 DB コンポーネントが含まれています。
- **StackRox Scanner**: StackRox Scanner は、RHACS のデフォルトのスキャナーです。StackRox Scanner は、Clair v2 オープンソーススキャナーのフォークから生まれました。
- **Scanner-DB**: このデータベースには、StackRox Scanner のデータが含まれています。

RHACS のスキャナーは、各イメージレイヤーを分析してベースオペレーティングシステムを特定し、プログラミング言語パッケージと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ってインストールされたパッケージを識別します。スキャナーは、さまざまな脆弱性ソースからの既知の脆弱性とスキャン結果を照合します。さらに、StackRox Scanner が、ノードのオペレーティングシステムとプラットフォームの脆弱性を特定します。これらの機能は、今後のリリースで Scanner V4 に追加される予定です。

3.2.1. 脆弱性ソース

RHACS は次の脆弱性ソースを使用します。

- Alpine Security Database
- Amazon Linux Security Center で追跡されるデータ
- Debian Security Tracker
- Oracle OVAL
- Photon OVAL
- Red Hat OVAL
- Red Hat CVE Map : これは、Red Hat Container Catalog に表示されるイメージに使用されます。
- SUSE OVAL
- Ubuntu OVAL
- OSV: Go、Java、Node.js (JavaScript)、Python、Ruby などの言語関連の脆弱性に使用されます。このソースは、脆弱性の CVE 番号ではなく GitHub Security Advisory (GHSA) ID を提供する場合があります。

注記

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

- NVD: ベンダーが情報を提供していない場合に情報のギャップを埋めるなど、さまざまな目的で使用されます。たとえば、Alpine は、詳細、CVSS スコア、重大度、公開日を提供していません。

注記

この製品は、NVD API を使用していますが、NVD による承認や認定を受けていません。

- StackRox: アップストリームの StackRox プロジェクトは、他のソースからのデータのフォーマットやデータの欠如が原因で発見されていない可能性のある一連の脆弱性を管理しています。

Scanner V4 Indexer は次のソースを使用します。

- repository-to-cpe.json: RPM リポジトリーを関連する CPE にマッピングします。これは、RHEL ベースのイメージの脆弱性を照合するために必要です。
- container-name-repos-map.json: コンテナー名と、コンテナーの配布先のリポジトリーを照合します。

3.3. セキュアクラスターサービス

保護対象の各クラスターには、セキュアクラスターサービスを RHACS Cloud Service を使用してインストールします。セキュアクラスターサービスには、次のコンポーネントが含まれています。

- **Sensor**: クラスターの分析と監視を行うサービスです。OpenShift Container Platform または Kubernetes API および Collector イベントをリッスンして、クラスターの現在の状態を報告します。RHACS Cloud Service ポリシーに基づき、デプロイタイムおよびランタイムの違反もトリガーします。さらに、ネットワークポリシーの適用、RHACS Cloud Service ポリシーの再処理の開始、Admission コントローラーとの対話など、すべてのクラスターの対話も担当します。
- **Admission コントローラー**: ユーザーが RHACS Cloud Service のセキュリティーポリシーに違反するワークロードを作成するのを防ぎます。
- **Collector**: クラスターノード上のコンテナーアクティビティーを分析および監視します。コンテナーのランタイムとネットワークアクティビティーの情報を収集し、収集したデータを Sensor に送信します。
- **StackRox Scanner** および **Scanner V4** (テクノロジープレビュー): Kubernetes では、セキュアクラスターサービスに、オプションのコンポーネントとして Scanner-slim が含まれています。一方、OpenShift Container Platform では、OpenShift Container Platform 統合レジストリーと、必要に応じて他のレジストリー内のイメージをスキャンするために、RHACS Cloud Service によって各セキュアクラスターに Scanner-slim バージョンがインストールされます。

重要

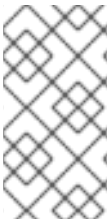
Scanner V4 はテクノロジープレビューのみの機能です。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat では、実稼働環境での使用を推奨していません。テクノロジープレビューの機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

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

- **Scanner V4 Indexer**: Scanner V4 Indexer は、以前はイメージ分析と呼ばれていたイメージのインデックス作成を実行します。Indexer は、イメージとレジストリーの認証情報を指定されると、レジストリーからイメージを取得します。ベースオペレーティングシステムを検索し、システムが存在する場合はそのパッケージを検索します。指定されたイメージの結果を含むインデックスレポートを保存および出力します。

- **Scanner V4 DB**: このコンポーネントは、Scanner V4 が有効な場合にインストールされます。このデータベースには、インデックスレポートを含む Scanner V4 の情報が格納されます。最適なパフォーマンスを得るには、Scanner V4 DB 用に永続ボリューム要求 (PVC) を設定してください。

- **Scanner-DB**: このデータベースには、StackRox Scanner のデータが含まれています。



注記

secure-cluster-services が **central-services** と同じクラスターにインストールされ、同じ namespace にインストールされている場合、**secure-cluster-services** は Scanner V4 コンポーネントをデプロイしません。代わりに、**central-services** に Scanner V4 のデプロイメントがすでに含まれているとみなします。

関連情報

- 外部コンポーネント

3.4. データアクセスと権限

Red Hat は、セキュアクラスターサービスをインストールするクラスターにアクセスできません。また、RHACS Cloud Service には、セキュアクラスターにアクセスするための権限は必要ありません。たとえば、新しい IAM ポリシー、アクセスロール、または API トークンを作成する必要はありません。

ただし、RHACS Cloud Service は、セキュアクラスターサービスから送信されるデータを保存します。すべてのデータは RHACS Cloud Service 内で暗号化されます。RHACS Cloud Service プラットフォーム内のデータの暗号化は、データの機密性と整合性を確保するのに役立ちます。

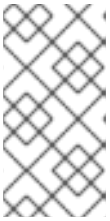
セキュアクラスターサービスをクラスターにインストールすると、データが生成され、RHACS Cloud Service に送信されます。このデータは RHACS Cloud Service プラットフォーム内で安全に保管され、認可された SRE チームメンバーとシステムのみがこのデータにアクセスできます。RHACS Cloud Service は、このデータを使用してクラスターとアプリケーションのセキュリティーとコンプライアンスを監視し、デプロイメントの最適化に役立つ貴重な見解と分析を提供します。

第4章 RHACS CLOUD SERVICE の概要

Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service) は、Red Hat OpenShift および Kubernetes クラスターにセキュリティーサービスを提供します。セキュアクラスターでサポートされるプラットフォームの詳細は、Red Hat Advanced Cluster Security for Kubernetes Support Matrix を参照してください。

前提条件

- Red Hat Hybrid Cloud Console から **Advanced Cluster Security** メニューオプションにアクセスできることを確認する。



注記

RHACS Cloud Service コンソールにアクセスするには、Red Hat Single Sign-On (SSO) 認証情報が必要です。別の ID プロバイダーが設定されている場合は、その認証情報が必要です。ACS コンソールへのデフォルトのアクセス を参照してください。

4.1. インストール手順の概要

次のセクションでは、インストール手順の概要と関連ドキュメントへのリンクを示します。

4.1.1. Red Hat OpenShift クラスターの保護

Operator を使用して Red Hat OpenShift クラスターを保護するには、次の手順を実行します。

1. セキュリティー保護するクラスターが 要件 を満たしていることを確認します。
2. Red Hat Hybrid Cloud Console で、**ACS インスタンス を作成** します。
3. 保護対象の各 Red Hat OpenShift クラスターで、**stackrox** という名前のプロジェクトを作成します。このプロジェクトに、RHACS Cloud Service のセキュアクラスターのリソースを格納します。
4. ACS コンソールで、init バンドルを作成 します。init バンドルには、RHACS Cloud Service のセキュアクラスターと ACS コンソール間の通信を可能にするシークレットが含まれています。
5. 各 Red Hat OpenShift クラスターで、init バンドルを使用し、リソースを作成して、init バンドルを適用 します。
6. 各 Red Hat OpenShift クラスターで、RHACS Operator をインストール します。
7. 各 Red Hat OpenShift クラスターで、Operator を使用して セキュアクラスターのリソースを **stackrox** プロジェクトにインストール します。
8. セキュアクラスターが ACS インスタンスと通信できることを確認して、インストールを検証します。

Helm チャートまたは **roxctl** CLI を使用して Red Hat OpenShift クラスターを保護するには、次の手順を実行します。

1. セキュリティー保護するクラスターが 要件 を満たしていることを確認します。
2. Red Hat Hybrid Cloud Console で、**ACS インスタンス を作成** します。

3. 保護対象の各 Red Hat OpenShift クラスターで、**stackrox** という名前のプロジェクトを作成します。このプロジェクトに、RHACS Cloud Service のセキュアクラスターのリソースを格納します。

4. ACS コンソールで、init バンドルを作成 します。init バンドルには、RHACS Cloud Service のセキュアクラスターと ACS コンソール間の通信を可能にするシークレットが含まれています。

5. 各 Red Hat OpenShift クラスターで、init バンドルを使用し、リソースを作成して、init バンドルを適用 します。

6. 各 Red Hat OpenShift クラスターで、Helm チャート または **roxctl** CLI を使用して、セキュアクラスターのリソースを **stackrox** プロジェクトにインストールします。

7. セキュアクラスターが ACS インスタンスと通信できることを確認して、インストールを検証します。

4.1.2. Kubernetes クラスターの保護

Kubernetes クラスターを保護するには、次の手順を実行します。

1. セキュリティー保護するクラスターが 要件 を満たしていることを確認します。

2. Red Hat Hybrid Cloud Console で、**ACS インスタンス** を作成 します。

3. ACS コンソールで、init バンドルを作成 します。init バンドルには、RHACS Cloud Service のセキュアクラスターと ACS コンソール間の通信を可能にするシークレットが含まれています。

4. 各 Kubernetes クラスターで、init バンドルを使用し、リソースを作成して、init バンドルを適用 します。

5. 各 Kubernetes クラスターで、Helm チャートまたは **roxctl** CLI を使用して、セキュアクラスターのリソースをインストール します。

6. セキュアクラスターが ACS インスタンスと通信できることを確認して、インストールを検証します。

4.2. ACS コンソールへのデフォルトのアクセス

デフォルトでは、ユーザーが使用できる認証メカニズムは、Red Hat Single Sign-On (SSO) を使用した認証です。Red Hat SSO 認証プロバイダーを削除または変更することはできません。ただし、最小アクセスロールを変更してルールを追加したり、別の ID プロバイダーを追加したりすることはできます。

注記

ACS での認証プロバイダーの動作の詳細は、認証プロバイダーについて を参照してください。

sso.redhat.com の専用 OIDC クライアントが ACS コンソールごとに作成されます。すべての OIDC クライアントは同じ **sso.redhat.com** レルムを共有します。**sso.redhat.com** によって発行されたトークンからのクレームは、次のように ACS 発行のトークンにマッピングされます。

- **realm_access.roles** から **groups** に
- **org_id** から **rh_org_id** に

- **is_org_admin** から **rh_is_org_admin** に
- **sub** から **userid** に

組み込みの Red Hat SSO 認証プロバイダーには、必須属性 **rh_org_id** があります。この属性は、RHACS Cloud Service インスタンスを作成したユーザーのアカウントに割り当てられた組織 ID に設定されているものです。これは、ユーザーが属している組織アカウントの ID です。これは、ユーザーが所属し、所有されている "テナント" と考えることができます。同じ組織アカウントを持つユーザーのみが、Red Hat SSO 認証プロバイダーを使用して ACS コンソールにアクセスできます。

注記

ACS コンソールへのアクセスをさらに制御するには、Red Hat SSO 認証プロバイダーを利用するのではなく、別のアイデンティティープロバイダーを設定してください。詳細は、認証プロバイダーについて を参照してください。他の認証プロバイダーをログインページの最初の認証オプションとして設定するには、その名前を辞書順で **Red Hat SSO** より小さくする必要があります。

最小アクセスロールは **None** に設定されます。このフィールドに別の値を割り当てると、同じ組織アカウントを持つすべてのユーザーが RHACS Cloud Service インスタンスにアクセスできるようになります。

組み込みの Red Hat SSO 認証プロバイダーで設定されるその他のルールには、次のものがあります。

- **userid** を **Admin** にマッピングするルール
- 組織の管理者を **Admin** にマッピングするルール

さらにルールを追加して、同じ組織アカウントを持つ他のユーザーに ACS コンソールへのアクセスを許可できます。たとえば、**email** をキーとして使用できます。

第5章 RED HAT ADVANCED CLUSTER SECURITY CLOUD SERVICE のデフォルトのリソース要件

5.1. RHACS CLOUD SERVICE の一般的な要件

Red Hat Advanced Cluster Security Cloud Service をインストールする前に、システムがいくつかの要件を満たしている必要があります。

警告

RHACS Cloud Service を以下の場所にインストールしないでください。

- Amazon Elastic File System (Amazon EFS)。代わりに、デフォルトの **gp2** ボリュームタイプで Amazon Elastic Block Store (Amazon EBS) を使用してください。
- Streaming SIMD Extensions (SSE) 4.2 命令セットを備えていない古い CPU。たとえば、**Sandy Bridge** より古い Intel プロセッサー、および **Bulldozer** より古い AMD プロセッサー。これらのプロセッサーは 2011 年にリリースされました。

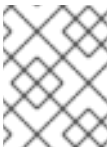
RHACS Cloud Service をインストールするには、次のいずれかのシステムが必要です。

- OpenShift Container Platform バージョン 4.11 以降、および Red Hat Enterprise Linux CoreOS (RHCOS) または Red Hat Enterprise Linux (RHEL) のサポートされているオペレーティングシステムを搭載したクラスターノード。
- サポートされているマネージド Kubernetes プラットフォーム、および Amazon Linux、CentOS、Google の Container-Optimized OS、Red Hat Enterprise Linux CoreOS (RHCOS)、Debian、Red Hat Enterprise Linux (RHEL)、または Ubuntu のサポートされているオペレーティングシステムを搭載したクラスターノード。
  サポートされるプラットフォームおよびアーキテクチャーの詳細は、Red Hat Advanced Cluster Security for Kubernetes Support Matrix を参照してください。

クラスターノードには、次の最小要件と推奨事項が適用されます。

アーキテクチャー

サポートされているアーキテクチャーは、**amd64**、**ppc64le**、または **s390x** です。



注記

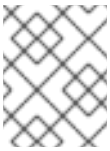
セキュアクラスターサービスは、IBM Power (**ppc64le**)、IBM Z (**s390x**)、および IBM® LinuxONE (**s390x**) クラスターでサポートされています。

プロセッサー

3 つの CPU コアが必要です。

メモリー

6 GiB の RAM が必要です。



注記

各コンポーネントのデフォルトのメモリー要件と CPU 要件を確認し、ノードサイズがそれらをサポートできることを確認してください。

ストレージ

RHACS Cloud Service の場合、永続ボリューム要求 (PVC) は必要ありません。ただし、スキャナー V4 が有効になっているセキュアクラスターを使用している場合は、PVC が強く推奨されます。最高のパフォーマンスを得るには、ソリッドステートドライブ (SSD) を使用してください。ただし、SSD を使用できない場合は、別のタイプのストレージを使用できます。

重要

RHACS Cloud Service では Ceph FS ストレージを使用しないでください。Red Hat は、RHACS Cloud Service に RBD ブロックモード PVC を使用することを推奨しています。

Helm チャートを使用して RHACS Cloud Service をインストールする場合は、次の要件を満たす必要があります。

- Helm チャートを使用して RHACS Cloud Service をインストールまたは設定する場合は、Helm コマンドラインインターフェイス (CLI) v3.2 以降が必要です。**helm version** コマンドを使用して、インストールされている Helm のバージョンを確認します。
- Red Hat コンテナーレジストリーにアクセスできる必要があります。**registry.redhat.io** からイメージをダウンロードする方法は、Red Hat コンテナーレジストリーの認証 を参照してください。

5.2. セキュアクラスターサービス

セキュアクラスターサービスには、次のコンポーネントが含まれています。

- Sensor
- Admission コントローラー
- Collector

5.2.1. Sensor

Sensor は、Kubernetes および OpenShift Container Platform クラスターをモニターします。これらのサービスは現在、単一のデプロイメントでデプロイされ、Kubernetes API とのインタラクションを処理し、Collector と連携しています。

メモリーと CPU の要件
次の表に、セキュアクラスターに Sensor をインストールして実行するために必要なメモリーとストレージの最小値を示します。

Sensor	CPU	メモリー
要求	2 コア	4 GiB
制限	4 コア	8 GiB

5.2.2. Admission コントローラー

Admission コントローラーは、ユーザーが設定したポリシーに違反するワークロードを作成するのを防ぎます。

メモリーと CPU の要件
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。次の表に、各レプリカのリクエストと制限を示します。

Admission コントローラー	CPU	メモリー
要求	0.05 コア	100 MiB
制限	0.5 コア	500 MiB

5.2.3. Collector

Collector は、セキュアクラスター内の各ノードのランタイムアクティビティーを監視します。Sensor に接続してこの情報をレポートします。コレクター Pod には 3 つのコンテナーがあります。最初のコンテナーはコレクターで、ノード上のランタイムアクティビティーを実際に監視して報告します。他の 2 つはコンプライアンスと node-inventory です。

コレクション要件
CORE_BPF 収集方法を使用するには、ベースカーネルが BTF をサポートし、BTF ファイルが Collector で使用できる必要があります。通常、カーネルのバージョンは 5.8 (RHEL ノードの場合は 4.18) 以降である必要があり、**CONFIG_DEBUG_INFO_BTF** 設定オプションを設定する必要があります。

Collector は、次の一覧に示されている標準の場所で BTF ファイルを検索します。

例5.1 BTF ファイルの場所

```
/sys/kernel/btf/vmlinux
/boot/vmlinux-<kernel-version>
/lib/modules/<kernel-version>/vmlinux-<kernel-version>
/lib/modules/<kernel-version>/build/vmlinux
/usr/lib/modules/<kernel-version>/kernel/vmlinux
/usr/lib/debug/boot/vmlinux-<kernel-version>
/usr/lib/debug/boot/vmlinux-<kernel-version>.debug
/usr/lib/debug/lib/modules/<kernel-version>/vmlinux
```

これらのファイルのいずれかが存在する場合は、カーネルに BTF サポートがあり、**CORE_BPF** が設定可能である可能性があります。

メモリーと CPU の要件
デフォルトでは、Collector サービスは 3 つのレプリカを実行します。次の表に、各レプリカの要求と制限、および Collector レプリカの合計を示します。

Collector コンテナー

タイプ	CPU	メモリー
要求	0.06 コア	320 MiB
制限	0.9 コア	1000 MiB

Compliance コンテナー

タイプ	CPU	メモリー
要求	0.01 コア	10 MiB
制限	1 コア	2000 MiB

Node-inventory コンテナー

タイプ	CPU	メモリー
要求	0.01 コア	10 MiB
制限	1 コア	500 MiB

Collector レプリカ要件の合計

タイプ	CPU	メモリー
要求	0.07 コア	340 MiB
制限	2.75 コア	3500 MiB

5.2.4. Scanner V4 (テクノロジープレビュー)

Scanner V4 は任意です。Scanner V4 がセキュアクラスターにインストールされている場合は、次の要件が適用されます。

重要

Scanner V4 はテクノロジープレビューのみの機能です。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat では、実稼働環境での使用を推奨していません。テクノロジープレビューの機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

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

この表の要件は、デフォルトである 2 レプリカに基づいています。

Scanner V4 Indexer	CPU	メモリー
要求	2 コア	3000 MiB
制限	4 コア	6 GiB

Scanner V4 は、データを保存するために Scanner V4 DB を必要とします。次の表に、Scanner V4 DB をインストールして実行するために必要なメモリーとストレージの最小値を示します。Scanner V4 DB の場合は、PVC の使用を強く推奨します。使用すると、最適なパフォーマンスが実現するためです。PVC は 10 GiB である必要があります。

Scanner V4 DB	CPU	メモリー
要求	0.2 コア	3 GiB
制限	2 コア	4 GiB

第6章 RED HAT ADVANCED CLUSTER SECURITY CLOUD SERVICE の推奨リソース要件

推奨されるリソースのガイドラインは、指定された数の namespace にわたって次のオブジェクトを作成する集中的なテストを実行することによって作成されました。

- 10 のデプロイメント、スリープ状態の 3 つの Pod レプリカ、4 つのシークレット、4 つの config map のマウント
- 10 のサービス。それぞれが以前のデプロイメントの 1 つの TCP/8080 および TCP/8443 ポートを指します。
- 以前のサービスの最初を指す 1 つのルート
- 2048 個のランダムな文字列文字を含む 10 個のシークレット
- 2048 個のランダムな文字列文字を含む 10 個の config map

結果の分析中に、使用されるリソースの増加の主な要因としてデプロイメントの数が特定されました。デプロイメントの数は、必要なリソースの見積もりに使用されました。

関連情報

- デフォルトのリソース要件

6.1. セキュアクラスターサービス

セキュアクラスターサービスには、次のコンポーネントが含まれています。

- Sensor
- Admission コントローラー
- Collector

注記

このページには Collector コンポーネントは含まれていません。必要なリソース要件は、デフォルトのリソース要件ページにリストされています。

6.1.1. Sensor

Sensor は、Kubernetes および OpenShift Container Platform クラスターをモニターします。これらのサービスは現在、単一のデプロイメントでデプロイされ、Kubernetes API とのインタラクションを処理し、Collector と連携しています。

メモリーと CPU の要件
次の表に、セキュアクラスターで Sensor を実行するために必要なメモリーと CPU の最小値を示します。

デプロイメント	デプロイメントごとの Pod	CPU	メモリー
< 25,000	3	2 コア	8 GiB
< 50,000	3	2 コア	16 GiB

6.1.2. Admission コントローラー

Admission コントローラーは、ユーザーが設定したポリシーに違反するワークロードを作成するのを防ぎます。

メモリーと CPU の要件
次の表に、セキュアクラスターでアドミッションコントローラーを実行するために必要なメモリーと CPU の最小値を示します。

デプロイメント	デプロイメントごとの Pod	CPU	メモリー
< 25,000	3	0.5 コア	600 MiB
< 50,000	3	0.5 コア	1200 MiB

第7章 RED HAT OPENSHIFT のセキュアクラスターを使用した RHACS CLOUD SERVICE のセットアップ

7.1. RED HAT CLOUD での RHACS CLOUD インスタンスの作成

Red Hat Hybrid Cloud Console でインスタンスを選択して、Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service) にアクセスします。**ACS インスタンス** には、Red Hat がお客様の代わりに設定および管理する RHACS Cloud Service 管理インターフェイスとサービスが含まれています。管理インターフェイスは、セキュアクラスターに接続します。セキュアクラスターには、脆弱性をスキャンして情報を収集するサービスが含まれています。1 つのインスタンスが多くのクラスターに接続して監視できます。

7.1.1. コンソールでのインスタンスの作成

Red Hat Hybrid Cloud Console で、セキュアクラスターに接続するための **ACS インスタンス** を作成します。

手順

ACS インスタンス を作成するには:

1. Red Hat Hybrid Cloud Console にログインします。
2. ナビゲーションメニューから、**Advanced Cluster Security → ACS Instances** を選択します。
3. **ACS インスタンスの作成** を選択し、表示されたフィールドに情報を入力するか、ドロップダウンリストから適切なオプションを選択します。
   - **Name**: **ACS インスタンス** の名前を入力します。**ACS インスタンス** には、"Central" とも呼ばれる RHACS Central コンポーネントが含まれています。このコンポーネントには、RHACS Cloud Service 管理インターフェイスと、Red Hat によって設定および管理されるサービスが含まれています。お客様は、Central と通信するセキュアクラスターを管理します。多くのセキュアクラスターを 1 つのインスタンスに接続できます。
   - **クラウドプロバイダー**: Central が配置されているクラウドプロバイダー。**AWS** を選択します。
   - **クラウドリージョン**: Central が配置されているクラウドプロバイダーのリージョン。次のいずれかのリージョンを選択します。
     - 米国東部、バージニア北部
     - ヨーロッパ、アイルランド
   - **アベイラビリティーゾーン**: デフォルト値 (**Multi**) を使用します。
4. **Create instance** をクリックします。

7.1.2. 次のステップ

- 保護対象の各 Red Hat OpenShift クラスターで、**stackrox** という名前のプロジェクトを作成します。このプロジェクトに、RHACS Cloud Service のセキュアクラスターのリソースを格納します。

7.2. RED HAT OPENSHIFT のセキュアクラスターでのプロジェクトの作成

保護対象の各 Red Hat OpenShift クラスターにプロジェクトを作成します。次に、このプロジェクトを使用して、Operator チャートまたは Helm チャートを使用して RHACS Cloud Service リソースをインストールします。

7.2.1. クラスター上にプロジェクトを作成する

手順

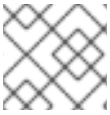
- OpenShift Container Platform クラスターで、**Home → Projects** に移動し、RHACS Cloud Service のプロジェクトを作成します。プロジェクトの **Name** として **stackrox** を使用します。

7.2.2. 次のステップ

- ACS コンソールで、init バンドルを作成 します。init バンドルには、RHACS Cloud Service のセキュアクラスターと ACS コンソール間の通信を可能にするシークレットが含まれています。

7.3. セキュアクラスター用の INIT バンドルを生成する

SecuredCluster リソースをクラスターにインストールする前に、init バンドルを作成する必要があります。**SecuredCluster** がインストールおよび設定されているクラスターは、このバンドルを使用して Central で認証します。RHACS ポータルまたは **roxctl** CLI を使用して、init バンドルを作成できます。次に、それを使用してリソースを作成することにより、init バンドルを適用します。



注記

init バンドルを作成するには、**Admin** ユーザーロールが必要です。

7.3.1. init バンドルの生成

7.3.1.1. RHACS ポータルを使用した init バンドルの生成

RHACS ポータルを使用して、シークレットを含む init バンドルを作成できます。

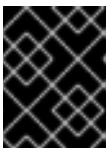
注記

init バンドルを作成するには、**Admin** ユーザーロールが必要です。

手順

1. 「Operator 方式を使用した Central インストールの検証」の説明に従って、RHACS ポータルのアドレスを見つけます。
2. RHACS ポータルにログインします。
3. セキュアクラスターがない場合は、**Platform Configuration → Clusters** ページが表示されます。
4. **Create init bundle**をクリックします。
5. クラスター init バンドルの名前を入力します。

6. プラットフォームを選択します。
7. セキュアクラスターに使用するインストール方法 (**Operator** または **Helm chart**) を選択します。
8. **Download** をクリックし、init バンドルを生成してダウンロードします。init バンドルは YAML ファイル形式で作成されます。同じインストール方法を使用する場合は、すべてのセキュアクラスターに対して 1 つの init バンドルとそれに対応する YAML ファイルを使用できます。



重要

このバンドルにはシークレットが含まれているため、セキュアに保管してください。

9. init バンドルを使用して適用し、セキュアクラスター上にリソースを作成します。
10. 各クラスターにセキュアクラスターサービスをインストールします。

7.3.1.2. roxctl CLI を使用した init バンドルの生成

roxctl CLI を使用して、シークレットを含む init バンドルを作成できます。

注記

init バンドルを作成するには、**Admin** ユーザーロールが必要です。

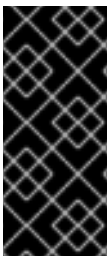
前提条件

- **ROX_API_TOKEN** および **ROX_CENTRAL_ADDRESS** 環境変数が設定されている。
  a. 次のコマンドを実行して **ROX_API_TOKEN** を設定します。

```
$ export ROX_API_TOKEN=<api_token>
```

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

```
$ export ROX_CENTRAL_ADDRESS=<address>:<port_number>
```



重要

RHACS Cloud Service では、Central アドレスを必要とする **roxctl** コマンドを使用する場合は、Red Hat Hybrid Cloud Console の **インスタンスの詳細** セクションに表示される **Central インスタンスのアドレス** を使用します。たとえば、**acs-data-ABCD12345.acs.rhcloud.com** の代わりに **acs-ABCD12345.acs.rhcloud.com** を使用します。

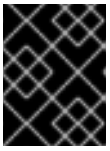
手順

- Helm インストールのシークレットを含むクラスター初期化バンドルを生成するには、次のコマンドを実行します。

```
$ roxctl -e "$ROX_CENTRAL_ADDRESS" \
  central init-bundles generate <cluster_init_bundle_name> \
  --output cluster_init_bundle.yaml
```

- Operator インストール用のシークレットを含むクラスター初期化バンドルを生成するには、次のコマンドを実行します。

```
$ roxctl -e "$ROX_CENTRAL_ADDRESS" \
  central init-bundles generate <cluster_init_bundle_name> \
  --output-secrets cluster_init_bundle.yaml
```



重要

このバンドルにはシークレットが含まれているため、安全に保管してください。同じバンドルを使用して、複数のセキュアクラスターを設定できます。

7.3.2. 次のステップ

- init バンドルを使用したリソースの作成

7.4. セキュアクラスター用の INIT バンドルを適用する

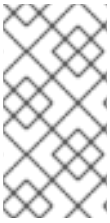
init バンドルを使用してリソースを作成し、それを適用します。

注記

init バンドルを適用するには、**Admin** ユーザーロールが必要です。

7.4.1. セキュアクラスターに init バンドルを適用する

セキュアクラスターを設定する前に、init バンドルを使用してそれを適用し、セキュアクラスター上に必要なリソースを作成する必要があります。init バンドルを適用すると、セキュアクラスター上のサービスが RHACS Cloud Service と通信できるようになります。



注記

Helm チャートを使用してインストールする場合は、この手順を実行しないでください。Helm を使用してインストールを完了してください。関連情報セクションの「Helm チャートを使用したセキュアクラスターへの RHACS のインストール」を参照してください。

前提条件

- シークレットを含む init バンドルを生成している必要があります。
- セキュアクラスターサービスをインストールするクラスター上に、**stackrox** プロジェクトまたは namespace を作成した。プロジェクトとして **stackrox** を使用することは必須ではありませんが、使用すると、クラスターのスキャン時に RHACS プロセスの脆弱性が報告されなくなります。

手順

リソースを作成するには、次の手順のいずれか 1 つだけを実行します。

- OpenShift Container Platform Web コンソールを使用してリソースを作成する: OpenShift Container Platform Web コンソールで、**stackrox** namespace に移動します。上部のメニューで + をクリックして、**Import YAML** ページを開きます。init バンドルファイルをドラッグするか、その内容をコピーしてエディターに貼り付け、**Create** をクリックします。コマンドが完了すると、**collector-tls**、**sensor-tls**、admission-control-tls の各リソースが作成されたことが画面に表示されます。

- Red Hat OpenShift CLI を使用してリソースを作成する: Red Hat OpenShift CLI を使用して、次のコマンドを実行してリソースを作成します。

```
$ oc create -f <init_bundle>.yaml \ 1
  -n <stackrox> 2
```

1 シークレットを含む init バンドルのファイル名を指定します。

2 Central サービスがインストールされているプロジェクトの名前を指定します。

検証

- 新しい証明書を取得するには、センサーを再起動します。
Sensor を再起動する方法の詳細は、「関連情報」セクションの「Sensor コンテナーの再起動」を参照してください。

7.4.2. 次のステップ

- 各 Red Hat OpenShift クラスターで、RHACS Operator をインストール します。

- 監視するすべてのクラスターに RHACS のセキュアクラスターサービスをインストールします。

7.4.3. 関連情報

- Sensor コンテナーの再起動

7.5. OPERATOR のインストール

RHACS Operator をセキュアクラスターにインストールします。

7.5.1. RHACS Cloud Service 用の RHACS Operator のインストール

OpenShift Container Platform で提供される OperatorHub を使用するのが、RHACS Operator をインストールする最も簡単な方法です。

前提条件

- Operator インストールパーミッションを持つアカウントを使用して OpenShift Container Platform クラスターにアクセスできる。

- OpenShift Container Platform 4.11 以降を使用している。サポートされるプラットフォームおよびアーキテクチャーの詳細は、Red Hat Advanced Cluster Security for Kubernetes Support Matrix を参照してください。

手順

1. Web コンソールで、**Operators → OperatorHub** ページに移動します。
2. Red Hat Advanced Cluster Security for Kubernetes が表示されない場合は、**Filter by keyword** ボックスに **Advanced Cluster Security** と入力して、Red Hat Advanced Cluster Security for Kubernetes Operator を検索します。
3. 詳細ページを表示するには、**Red Hat Advanced Cluster Security for Kubernetes Operator**を選択します。
4. Operator に関する情報を読み、**Install** をクリックします。
5. **Install Operator** ページで以下を行います。
   - **Installation mode** のデフォルト値を **All namespaces on the cluster**として保持します。
   - **Installed namespace** フィールドで、Operator をインストールする特定の namespace を選択します。Red Hat Advanced Cluster Security for Kubernetes Operator を **rhacs-operator** namespace にインストールします。
   - **Update approval** には、自動更新または手動更新を選択します。
     自動更新を選択した場合、Operator の新しいバージョンが利用可能になると、Operator Lifecycle Manager (OLM) が Operator の実行中のインスタンスを自動的にアップグレードします。

     手動更新を選択した場合、新しいバージョンの Operator が利用可能になると、OLM は更新リクエストを作成します。クラスター管理者は、更新リクエストを手動で承認して、Operator を最新バージョンに更新する必要があります。

     Red Hat は、RHACS Cloud Service で Operator の自動アップグレードを有効にすることを推奨します。詳細は、Red Hat Advanced Cluster Security for Kubernetes Support Matrix を参照してください。
6. **Install** をクリックします。

検証

- インストールが完了したら、**Operators → Installed Operators** に移動して、Red Hat Advanced Cluster Security for Kubernetes Operator が **Succeeded** ステータスとともにリスト表示されていることを確認します。

7.5.2. 次のステップ

- 各 Red Hat OpenShift クラスターで、セキュアクラスターのリソースを **stackrox** プロジェクトにインストール します。

7.6. RHACS CLOUD SERVICE からのセキュアクラスターリソースのインストール

Operator チャートまたは Helm チャートを使用して、セキュアクラスターに RHACS Cloud Service をインストールできます。**roxctl** CLI を使用してインストールすることもできますが、この方法を使用する必要がある特定のインストールニーズがないかぎり、この方法は使用しないでください。

前提条件

- Red Hat OpenShift クラスターを作成し、そこに Operator をインストールした。
- RHACS Cloud Service の ACS コンソールで、init バンドルを作成してダウンロードした。
- **oc create** コマンドを使用して init バンドルを適用した。
- インストール中に、アドレスとポート番号を含む **Central API Endpoint** をメモした。この情報を表示するには、クラウドコンソールのナビゲーションメニューから **Advanced Cluster Security → ACS Instances** を選択し、作成した ACS インスタンスをクリックします。

7.6.1. Operator を使用したセキュアクラスターへの RHACS のインストール

7.6.1.1. セキュアクラスターサービスのインストール

Operator を使用してクラスターにセキュアクラスターサービスをインストールできます。これにより、**SecuredCluster** カスタムリソースが作成されます。セキュアクラスターサービスは、監視する環境内のすべてのクラスターにインストールする必要があります。

前提条件

- OpenShift Container Platform を使用している場合は、バージョン 4.11 以降をインストールした。
- 保護対象のクラスター (セキュアクラスターと呼ばれます) に RHACS Operator をインストールした。
- init バンドルを生成し、クラスターに適用した。

手順

1. セキュアクラスターの OpenShift Container Platform Web コンソールで、**Operators → Installed Operators** ページに移動します。
2. RHACS Operator をクリックします。
3. **Operator details** ページの central ナビゲーションメニューから **Secured Cluster** をクリックします。
4. **Create SecuredCluster** をクリックします。
5. **Configure via** フィールドで次のいずれかのオプションを選択します。
   - **Form view**: 画面上のフィールドを使用してセキュアクラスターを設定する場合、および他のフィールドを変更する必要がない場合は、このオプションを使用します。
   - **YAML view**: このビューは、YAML ファイルを使用してセキュアクラスターをセットアップするために使用します。YAML ファイルがウィンドウに表示され、その中のフィールドを編集できます。このオプションを選択した場合、ファイルの編集が終了したら、**Create** をクリックします。
6. **Form view** を使用している場合は、デフォルトの名前を受け入れるか編集して、新しいプロジェクト名を入力します。デフォルト値は **stackrox-secured-cluster-services** です。
7. オプション: クラスターのラベルを追加します。
8. **SecuredCluster** カスタムリソースの一意の名前を入力します。

9. **Central Endpoint** には、Central インスタンスのアドレスとポート番号を入力します。たとえば、Central が **https://central.example.com** で利用できる場合は、central エンドポイントを **central.example.com:443** として指定します。

    - RHACS Cloud Service の場合、アドレスとポート番号を含む **Central API Endpoint** を使用します。この情報を表示するには、クラウドコンソールのナビゲーションメニューから **Advanced Cluster Security → ACS Instances** を選択し、作成した ACS インスタンスをクリックします。

    - Central がインストールされている同じクラスターにセキュアクラスターサービスをインストールする場合に **のみ**、デフォルト値 **central.stackrox.svc:443** を使用します。

    - 複数のクラスターを設定する場合は、デフォルト値を使用しないでください。代わりに、各クラスターの **Central Endpoint** 値を設定するときにホスト名を使用します。

10. 残りのフィールドについては、デフォルト値を受け入れるか、必要に応じてカスタム値を設定します。たとえば、カスタム証明書または信頼されていない CA を使用している場合は、TLS の設定が必要になる場合があります。詳細は、「Operator を使用した RHACS のセキュアクラスターサービスオプションの設定」を参照してください。

11. **Create** をクリックします。

12. 少し待った後、**SecuredClusters** ページに **stackrox-secured-cluster-services** のステータスが表示されます。次のような状態が表示される場合があります。

    - **Conditions: Deployed, Initialized**: セキュアクラスターサービスがインストールされており、セキュアクラスターが Central と通信しています。

    - **Conditions: Initialized, Irreconcilable**: セキュアクラスターが Central と通信していません。RHACS Web ポータルで作成した init バンドルがセキュアクラスターに適用されていることを確認してください。

次のステップ

1. 追加のセキュアクラスター設定を設定します (オプション)。
2. インストールの検証

7.6.2. Helm チャートを使用したセキュアクラスターへの RHACS Cloud Service のインストール

Helm チャートをカスタマイズせずに使用するか、デフォルト値を使用するか、設定パラメーターをカスタマイズして、セキュアクラスターに RHACS をインストールできます。

最初に、Helm チャートリポジトリーを追加していることを確認します。

7.6.2.1. Helm チャートリポジトリーの追加

手順

- RHACS チャートリポジトリーを追加します。

```
$ helm repo add rhacs https://mirror.openshift.com/pub/rhacs/charts/
```

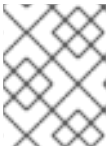
Red Hat Advanced Cluster Security for Kubernetes の Helm リポジトリーには、次のようなさまざまなコンポーネントをインストールするための Helm チャートが含まれています。

- 集中型コンポーネント (Central および Scanner) をインストールするための Central サービス Helm チャート (**central-services**)。

注記

集中型コンポーネントは 1 回だけデプロイします。同じインストールを使用して複数の別のクラスターを監視できます。

- クラスターおよびノードごとのコンポーネント (Sensor、Admission Controller、Collector、および Scanner-slim) をインストールするためのセキュアクラスターサービスの Helm チャート (**secured-cluster-services**)。



注記

モニターする各クラスターにクラスターごとのコンポーネントをデプロイし、モニターするすべてのノードにノードごとのコンポーネントをデプロイします。

検証

- 次のコマンドを実行して、追加されたチャートリポジトリーを確認します。

```
$ helm search repo -l rhacs/
```

7.6.2.2. カスタマイズせずに Helm チャートを使用してセキュアクラスターに RHACS Cloud Service をインストールする

7.6.2.2.1. カスタマイズせずに secured-cluster-services Helm チャートをインストールする

次の手順に従って、**secured-cluster-services** Helm チャートをインストールし、クラスターおよびノードごとのコンポーネント (Sensor、Admission コントローラー、Collector、および Scanner-slim) をデプロイします。

前提条件

- クラスターの RHACS init バンドルを生成しておく必要があります。
- Red Hat コンテナーレジストリーへのアクセス権と、認証用のプルシークレットが必要です。**registry.redhat.io** からイメージをダウンロードする方法は、Red Hat コンテナーレジストリーの認証 を参照してください。
- Central サービスを公開するアドレスとポート番号が必要です。

手順

- Kubernetes ベースのクラスターで次のコマンドを実行します。

```
$ helm install -n stackrox --create-namespace \
  stackrox-secured-cluster-services rhacs/secured-cluster-services \
  -f <path_to_cluster_init_bundle.yaml> \ 1
  -f <path_to_pull_secret.yaml> \ 2
```

```
    --set clusterName=<name_of_the_secured_cluster> \
    --set centralEndpoint=<endpoint_of_central_service> 3
    --set imagePullSecrets.username=<your redhat.com username> \4
    --set imagePullSecrets.password=<your redhat.com password>5
```

1. **-f** オプションを使用して、init バンドルのパスを指定します。
2. -f オプションを使用して、Red Hat コンテナーレジストリー認証用のプルシークレットのパスを指定します。
3. Central のアドレスとポート番号を指定します。例: **acs.domain.com:443**
4. Red Hat コンテナーレジストリー認証のプルシークレットのユーザー名を含めます。
5. Red Hat コンテナーレジストリー認証のプルシークレットのパスワードを含めます。

- OpenShift Container Platform クラスターで以下のコマンドを実行します。

```
$ helm install -n stackrox --create-namespace \
    stackrox-secured-cluster-services rhacs/secured-cluster-services \
    -f <path_to_cluster_init_bundle.yaml> \1
    -f <path_to_pull_secret.yaml> \2
    --set clusterName=<name_of_the_secured_cluster> \
    --set centralEndpoint=<endpoint_of_central_service> 3
    --set scanner.disable=false 4
```

1. **-f** オプションを使用して、init バンドルのパスを指定します。
2. -f オプションを使用して、Red Hat コンテナーレジストリー認証用のプルシークレットのパスを指定します。
3. Central のアドレスとポート番号を指定します。例: **acs.domain.com:443**
4. **scanner.disable** パラメーターの値を **false** に設定します。これは、インストール中に Scanner-slim が有効になることを意味します。Kubernetes では、セキュアクラスターサービスに、オプションのコンポーネントとして Scanner-slim が含まれています。

関連情報

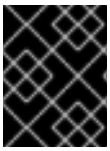
- セキュアクラスター用の init バンドルを生成する
- セキュアクラスター用の init バンドルを適用する

7.6.2.3. カスタマイズした secured-cluster-services Helm チャートの設定

helm install および **helm upgrade** コマンドで Helm チャート設定パラメーターを使用できます。これらのパラメーターは、**--set** オプションを使用するか、YAML 設定ファイルを作成することで指定します。

以下のファイルを作成して、Red Hat Advanced Cluster Security for Kubernetes をインストールするための Helm チャートを設定します。

- パブリック設定ファイル **values-public.yaml**: このファイルを使用して、機密性の低いすべての設定オプションを保存します。
- プライベート設定ファイル **values-private.yaml**: このファイルを使用して、機密性の高いすべての設定オプションを保存します。このファイルは安全に保管してください。



重要

secured-cluster-services Helm チャートを使用する場合、チャートの一部である **values.yaml** ファイルを変更しないでください。

7.6.2.3.1. 設定パラメーター

パラメーター	説明
clusterName	クラスターの名前です。
centralEndpoint	Central エンドポイントのアドレス (ポート番号を含む)。gRPC に対応していないロードバランサーを使用している場合は、エンドポイントアドレスの前に **wss://** を付けて、WebSocket プロトコルを使用します。複数のクラスターを設定する場合は、アドレスにホスト名を使用します (例: **central.example.com:443**)。
sensor.endpoint	ポート番号を含む Sensor エンドポイントのアドレスです。
sensor.imagePullPolicy	Sensor コンテナーのイメージプルポリシーです。
sensor.serviceTLS.cert	Sensor が使用する内部サービス間の TLS 証明書です。
sensor.serviceTLS.key	Sensor が使用する内部サービス間 TLS 証明書キーです。
sensor.resources.requests.memory	Sensor コンテナーのメモリーリクエストです。このパラメーターを使用して、デフォルト値をオーバーライドします。
sensor.resources.requests.cpu	Sensor コンテナーの CPU リクエストです。このパラメーターを使用して、デフォルト値をオーバーライドします。
sensor.resources.limits.memory	Sensor コンテナーのメモリー制限です。このパラメーターを使用して、デフォルト値をオーバーライドします。
sensor.resources.limits.cpu	Sensor コンテナーの CPU 制限です。このパラメーターを使用して、デフォルト値をオーバーライドします。

パラメーター	説明
sensor.nodeSelector	ノードセレクターのラベルを **label-key: label-value** の形式で指定して、指定したラベルを持つノードでのみ Sensor をスケジュールするように強制します。
sensor.tolerations	ノードセレクターが taint されたノードを選択する場合は、このパラメーターを使用して、Sensor の taint toleration キー、値、および effect を指定します。このパラメーターは、主にインフラストラクチャーノードに使用されます。
image.main.name	**main** イメージの名前です。
image.collector.name	Collector イメージの名前です。
image.main.registry	main イメージに使用しているレジストリーのアドレスです。
image.collector.registry	Collector イメージに使用しているレジストリーのアドレスです。
image.scanner.registry	Scanner イメージに使用しているレジストリーのアドレスです。
image.scannerDb.registry	Scanner DB イメージに使用しているレジストリーのアドレスです。
image.scannerV4.registry	Scanner V4 イメージに使用しているレジストリーのアドレスです。
image.scannerV4DB.registry	Scanner V4 DB イメージに使用しているレジストリーのアドレスです。
image.main.pullPolicy	**main** イメージのイメージプルポリシーです。
image.collector.pullPolicy	Collector イメージのイメージプルポリシーです。
image.main.tag	使用する **main** イメージのタグです。
image.collector.tag	使用する **collector** イメージのタグです。
collector.collectionMethod	**CORE_BPF**、**EBPF** (非推奨)、**NO_COLLECTION** のいずれか。
collector.imagePullPolicy	Collector コンテナーのイメージプルポリシーです。

パラメーター	説明
collector.complianceImagePullPolicy	Compliance コンテナーのイメージプルポリシーです。
collector.disableTaintTolerations	**false** を指定すると、許容値が Collector に適用され、Collector Pod は taint のあるすべてのノードにスケジュールできます。**true** として指定すると、許容値は適用されず、Collector Pod は taint のあるノードにスケジュールされません。
collector.resources.requests.memory	Collector コンテナーのメモリーリクエストです。このパラメーターを使用して、デフォルト値をオーバーライドします。
collector.resources.requests.cpu	Collector コンテナーの CPU リクエストです。このパラメーターを使用して、デフォルト値をオーバーライドします。
collector.resources.limits.memory	Collector コンテナーのメモリー制限です。このパラメーターを使用して、デフォルト値をオーバーライドします。
collector.resources.limits.cpu	Collector コンテナーの CPU 制限です。このパラメーターを使用して、デフォルト値をオーバーライドします。
collector.complianceResources.requests.memory	Compliance コンテナーのメモリーリクエストです。このパラメーターを使用して、デフォルト値をオーバーライドします。
collector.complianceResources.requests.cpu	Compliance の CPU リクエストです。このパラメーターを使用して、デフォルト値をオーバーライドします。
collector.complianceResources.limits.memory	Compliance コンテナーのメモリー制限です。このパラメーターを使用して、デフォルト値をオーバーライドします。
collector.complianceResources.limits.cpu	Compliance コンテナーの CPU 制限です。このパラメーターを使用して、デフォルト値をオーバーライドします。
collector.serviceTLS.cert	Collector が使用する内部サービス間 TLS 証明書です。
collector.serviceTLS.key	Collector が使用する内部サービス間 TLS 証明書キーです。

パラメーター	説明
admissionControl.listenOnCreates	この設定は、ワークロード作成イベントの **AdmissionReview** リクエストで Red Hat Advanced Cluster Security for Kubernetes に接続するように Kubernetes を設定するかどうかを制御します。
admissionControl.listenOnUpdates	このパラメーターを **false** に設定すると、Red Hat Advanced Cluster Security for Kubernetes は、Kubernetes API サーバーがオブジェクト更新イベントを送信しないようにする **ValidatingWebhookConfiguration** を作成します。オブジェクトの更新ボリュームは通常、オブジェクトが作成するボリュームよりも多いため、これを **false** のままにしておくと、アドミッションコントロールサービスのロードが制限され、アドミッションコントロールサービスが誤動作する可能性が低くなります。
admissionControl.listenOnEvents	この設定は、Kubernetes **exec** および **portforward** イベントの **AdmissionReview** リクエストで Red Hat Advanced Cluster Security for Kubernetes に接続するようにクラスターを設定するかどうかを制御します。RHACS は、OpenShift Container Platform 3.11 ではこの機能をサポートしていません。
admissionControl.dynamic.enforceOnCreates	この設定は、Red Hat Advanced Cluster Security for Kubernetes がポリシーを評価するかどうかを制御します。無効にすると、すべての AdmissionReview リクエストが自動的に承認されます。
admissionControl.dynamic.enforceOnUpdates	この設定は、アドミッションコントロールサービスの動作を制御します。これを機能させるには、**listenOnUpdates** を **true** として指定する必要があります。
admissionControl.dynamic.scanInline	このオプションを **true** に設定すると、アドミッションコントロールサービスは、アドミッションデシジョンを行う前にイメージスキャンをリクエストします。イメージスキャンには数秒かかるため、このオプションを有効にするのは、クラスターで使用されるすべてのイメージがデプロイ前にスキャンされることを確認できる場合のみです (たとえば、イメージビルド中の CI 統合によって)。このオプションは、RHACS ポータルの **Contact image scanners** オプションに対応します。
admissionControl.dynamic.disableBypass	Admission コントローラーのバイパスを無効にするには、**true** に設定します。

パラメーター	説明
admissionControl.dynamic.timeout	アドミッションレビューリクエストを評価する間、Red Hat Advanced Cluster Security for Kubernetes が待機する最大時間 (秒単位) です。これを使用して、イメージスキャンを有効にするときにリクエストのタイムアウトを設定します。イメージスキャンの実行時間が指定した時間より長い場合、Red Hat Advanced Cluster Security for Kubernetes はリクエストを受け入れます。
admissionControl.resources.requests.memory	Admission Control コンテナーのメモリーリクエストです。このパラメーターを使用して、デフォルト値をオーバーライドします。
admissionControl.resources.requests.cpu	Admission Control コンテナーの CPU リクエストです。このパラメーターを使用して、デフォルト値をオーバーライドします。
admissionControl.resources.limits.memory	Admission Control コンテナーのメモリー制限です。このパラメーターを使用して、デフォルト値をオーバーライドします。
admissionControl.resources.limits.cpu	Admission Control コンテナーの CPU 制限です。このパラメーターを使用して、デフォルト値をオーバーライドします。
admissionControl.nodeSelector	ノードセレクターのラベルを **label-key: label-value** の形式で指定して、指定したラベルを持つノードでのみ Admission Control をスケジュールするように強制します。
admissionControl.tolerations	ノードセレクターが taint されたノードを選択する場合は、このパラメーターを使用して、アドミッションコントロールの taint toleration キー、値、および effect を指定します。このパラメーターは、主にインフラストラクチャーノードに使用されます。
admissionControl.serviceTLS.cert	Admission Control が使用する内部サービス間 TLS 証明書です。
admissionControl.serviceTLS.key	Admission Control が使用する内部サービス間 TLS 証明書キーです。
registryOverride	このパラメーターを使用して、デフォルトの **docker.io** レジストリーをオーバーライドします。他のレジストリーを使用している場合は、レジストリーの名前を指定してください。

パラメーター	説明
collector.disableTaintTolerations	**false** を指定すると、許容値が Collector に適用され、Collector Pod は taint のあるすべてのノードにスケジュールできます。**true** として指定した場合、許容値は適用されず、Collector Pod は taint のあるノードにスケジュールされません。
createUpgraderServiceAccount	**sensor-upgrader** アカウントを作成するには、**true** を指定します。デフォルトでは、Red Hat Advanced Cluster Security for Kubernetes は、各セキュアクラスターに **sensor-upgrader** という名前のサービスアカウントを作成します。このアカウントは高い権限を持ちますが、アップグレードの時のみ使用されます。このアカウントを作成しない場合、Sensor に十分な権限がない場合は、将来のアップグレードを手動で完了する必要があります。
createSecrets	**false** を指定すると、Sensor、Collector、および Admission コントローラーのオーケストレーターシークレットの作成がスキップされます。
collector.slimMode	Collector のデプロイに slim Collector イメージを使用する場合は、**true** を指定します。EBPF 収集方法で slim Collector イメージを使用するには、対応する eBPF プローブを Central で提供する必要があります。Red Hat Advanced Cluster Security for Kubernetes をオフラインモードで実行している場合、slim Collector を機能させるために、stackrox.io からカーネルサポートパッケージをダウンロードして Central にアップロードする必要があります。それ以外の場合は、Central が https://collector-modules.stackrox.io/ でホストされているオンラインプローブリポジトリーにアクセスできることを確認する必要があります。
sensor.resources	Sensor のリソース仕様です。
admissionControl.resources	Admission コントローラーのリソース仕様です。
collector.resources	Collector のリソース仕様です。
collector.complianceResources	Collector の Compliance コンテナーのリソース仕様です。
exposeMonitoring	このオプションを **true** に設定すると、Red Hat Advanced Cluster Security for Kubernetes がポート番号 9090 で Sensor、Collector、および Admission コントローラーの Prometheus メトリクスエンドポイントを公開します。

パラメーター	説明
auditLogs.disableCollection	このオプションを **true** に設定すると、Red Hat Advanced Cluster Security for Kubernetes が、設定マップとシークレットへのアクセスと変更を検出するために使用される監査ログ検出機能を無効にします。
scanner.disable	このオプションを **false** に設定すると、Red Hat Advanced Cluster Security for Kubernetes がセキュアクラスターに Scanner-slim と Scanner DB をデプロイし、OpenShift Container Registry 上のイメージをスキャンできるようにします。Scanner-slim の有効化は、OpenShift Container Platform および Kubernetes セキュアクラスターでサポートされています。デフォルトは **true** です。
scanner.dbTolerations	ノードセレクターが taint されたノードを選択する場合は、このパラメーターを使用して、Scanner DB の taint toleration キー、値、および effect を指定します。
scanner.replicas	Collector の Compliance コンテナーのリソース仕様です。
scanner.logLevel	このパラメーターを設定すると、Scanner のログレベルを変更できます。このオプションは、トラブルシューティングの目的でのみ使用してください。
scanner.autoscaling.disable	このオプションを **true** に設定すると、Red Hat Advanced Cluster Security for Kubernetes が Scanner デプロイメントでの自動スケーリングを無効にします。
scanner.autoscaling.minReplicas	自動スケーリングのレプリカの最小数です。デフォルトは 2 です。
scanner.autoscaling.maxReplicas	自動スケーリングのレプリカの最大数です。デフォルトは 5 です。
scanner.nodeSelector	ノードセレクターのラベルを **label-key: label-value** の形式で指定して、指定したラベルを持つノードでのみ Scanner をスケジュールするように強制します。
scanner.tolerations	ノードセレクターが taint されたノードを選択する場合は、このパラメーターを使用して、Scanner の taint toleration キー、値、および effect を指定します。

パラメーター	説明
scanner.dbNodeSelector	ノードセレクターのラベルを **label-key: label-value** の形式で指定して、指定したラベルを持つノードでのみ Scanner DB をスケジュールするように強制します。
scanner.dbTolerations	ノードセレクターが taint されたノードを選択する場合は、このパラメーターを使用して、Scanner DB の taint toleration キー、値、および effect を指定します。
scanner.resources.requests.memory	Scanner コンテナーのメモリー要求。このパラメーターを使用して、デフォルト値をオーバーライドします。
scanner.resources.requests.cpu	Scanner コンテナーの CPU 要求。このパラメーターを使用して、デフォルト値をオーバーライドします。
scanner.resources.limits.memory	Scanner コンテナーのメモリー制限。このパラメーターを使用して、デフォルト値をオーバーライドします。
scanner.resources.limits.cpu	Scanner コンテナーの CPU 制限。このパラメーターを使用して、デフォルト値をオーバーライドします。
scanner.dbResources.requests.memory	Scanner DB コンテナーのメモリー要求。このパラメーターを使用して、デフォルト値をオーバーライドします。
scanner.dbResources.requests.cpu	Scanner DB コンテナーの CPU 要求。このパラメーターを使用して、デフォルト値をオーバーライドします。
scanner.dbResources.limits.memory	Scanner DB コンテナーのメモリー制限。このパラメーターを使用して、デフォルト値をオーバーライドします。
scanner.dbResources.limits.cpu	Scanner DB コンテナーの CPU 制限。このパラメーターを使用して、デフォルト値をオーバーライドします。
monitoring.openshift.enabled	このオプションを **false** に設定すると、Red Hat Advanced Cluster Security for Kubernetes が Red Hat OpenShift モニタリングをセットアップしません。Red Hat OpenShift 4 では、デフォルトで **true** に設定されます。

7.6.2.3.1.1. 環境変数

Sensor および Admission コントローラーの環境変数は、次の形式で指定できます。

```
customize:
  envVars:
    ENV_VAR1: "value1"
    ENV_VAR2: "value2"
```

customize 設定を使用すると、この Helm チャートによって作成されたすべてのオブジェクトのカスタム Kubernetes メタデータ (ラベルとアノテーション) と、ワークロードの追加の Pod ラベル、Pod アノテーション、コンテナー環境変数を指定できます。

より一般的なスコープ (たとえば、すべてのオブジェクト) で定義されたメタデータを、より狭いスコープ (たとえば、Sensor デプロイメントのみ) で定義されたメタデータでオーバーライドできるという意味で、設定は階層的です。

7.6.2.3.2. カスタマイズした secured-cluster-services Helm チャートのインストール

values-public.yaml ファイルと **values-private.yaml** ファイルを設定したら、**secured-cluster-services** Helm チャートをインストールして、次のクラスターおよびノードごとのコンポーネントをデプロイします。

- Sensor
- Admission コントローラー
- Collector
- Scanner: StackRox Scanner がインストールされている場合、必要に応じてセキュアクラスターにデプロイする
- Scanner DB: StackRox Scanner がインストールされている場合、必要に応じてセキュアクラスターにデプロイする
- Scanner V4 Indexer および Scanner V4 DB: Scanner V4 がインストールされている場合、必要に応じてセキュアクラスターにデプロイする

重要

Scanner V4 はテクノロジープレビューのみの機能です。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat では、実稼働環境での使用を推奨していません。テクノロジープレビューの機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

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

前提条件

- クラスターの RHACS init バンドルを生成しておく必要があります。
- Red Hat コンテナーレジストリーへのアクセス権と、認証用のプルシークレットが必要です。**registry.redhat.io** からイメージをダウンロードする方法は、Red Hat コンテナーレジストリーの認証 を参照してください。

- Central サービスを公開するアドレスとポート番号が必要です。

手順

- 以下のコマンドを実行します。

```
$ helm install -n stackrox \
  --create-namespace stackrox-secured-cluster-services rhacs/secured-cluster-services \
  -f <name_of_cluster_init_bundle.yaml> \
  -f <path_to_values_public.yaml> -f <path_to_values_private.yaml> \ 1
  --set imagePullSecrets.username=<username> \ 2
  --set imagePullSecrets.password=<password> 3
```

1. **-f** オプションを使用して、YAML 設定ファイルのパスを指定します。
2. Red Hat コンテナーレジストリー認証のプルシークレットのユーザー名を含めます。
3. Red Hat コンテナーレジストリー認証のプルシークレットのパスワードを含めます。

注記

継続的インテグレーション (CI) システムを使用して **secured-cluster-services** Helm チャートをデプロイするには、init バンドル YAML ファイルを環境変数として **helm install** コマンドに渡します。

```
$ helm install ... -f <(echo "$INIT_BUNDLE_YAML_SECRET") 1
```

1. base64 でエンコードされた変数を使用している場合は、代わりに **helm install ... -f <(echo "$INIT_BUNDLE_YAML_SECRET" | base64 --decode)** コマンドを使用してください。

関連情報

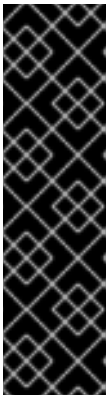
- セキュアクラスター用の init バンドルを生成する
- セキュアクラスター用の init バンドルを適用する

7.6.2.4. secured-cluster-services Helm チャートをデプロイした後の設定オプションの変更

secured-cluster-services Helm チャートをデプロイした後、設定オプションを変更できます。

helm upgrade コマンドを使用して変更を加える場合は、次のガイドラインと要件が適用されます。

- **--set** または **--set-file** パラメーターを使用して設定値を指定することもできます。ただし、これらのオプションは保存されないため、変更を加えるたびにすべてのオプションを手動で再度指定する必要があります。
- 変更の内容によっては (たとえば Scanner V4 などの新しいコンポーネントを有効にした場合は)、コンポーネントに対して新しい証明書を発行する必要があります。したがって、これらの変更を行う場合は CA を指定する必要があります。



重要

Scanner V4 はテクノロジープレビューのみの機能です。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat では、実稼働環境での使用を推奨していません。テクノロジープレビューの機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

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

- CA が初期インストール中に Helm チャートによって生成された場合は、自動的に生成された該当する値をクラスターから取得し、**helm upgrade** コマンドで指定する必要があります。**central-services** Helm チャートのインストール後の注記に、自動生成された値を取得するためのコマンドが含まれています。
- CA が Helm チャートの外部で生成されたものであり、**central-services** チャートのインストール時にその CA を指定した場合は、**helm upgrade** コマンドを使用するときに、たとえば **helm upgrade** コマンドで **--reuse-values** フラグを使用して、その操作を再度実行する必要があります。

手順

1. **values-public.yaml** および **values-private.yaml** 設定ファイルを新しい値で更新します。
2. **helm upgrade** コマンドを実行し、**-f** オプションを使用して設定ファイルを指定します。

```
$ helm upgrade -n stackrox \
  stackrox-secured-cluster-services rhacs/secured-cluster-services \
  --reuse-values \ 1
  -f <path_to_values_public.yaml> \
  -f <path_to_values_private.yaml>
```

1 **values_public.yaml** ファイルと **values_private.yaml** ファイルに含まれていない値を変更した場合は、**--reuse-values** パラメーターを含めます。

7.6.3. roxctl CLI を使用したセキュアクラスターへの RHACS のインストール

CLI を使用してセキュアクラスターに RHACS をインストールするには、次の手順を実行します。

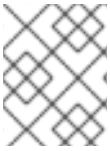
1. **roxctl** CLI をインストールします。
2. Sensor を取り付けます。

7.6.3.1. roxctl CLI のインストール

最初にバイナリーをダウンロードする必要があります。**roxctl** は、Linux、Windows、または macOS にインストールできます。

7.6.3.1.1. Linux への roxctl CLI のインストール

次の手順を使用して、Linux に **roxctl** CLI バイナリーをインストールできます。



注記

Linux 用の **roxctl** CLI は、**amd64**、**ppcl64le**、および **s390x** アーキテクチャーで使用できます。

手順

1. ターゲットのオペレーティングシステムの **roxctl** アーキテクチャーを確認します。

```
$ arch="$(uname -m | sed "s/x86_64//")"; arch="${arch:+-$arch}"
```

2. **roxctl** CLI をダウンロードします。

```
$ curl -L -f -o roxctl
"https://mirror.openshift.com/pub/rhacs/assets/4.4.4/bin/Linux/roxctl${arch}"
```

3. **roxctl** バイナリーを実行可能にします。

```
$ chmod +x roxctl
```

4. **PATH** 上にあるディレクトリーに **roxctl** バイナリーを配置します。
PATH を確認するには、以下のコマンドを実行します。

```
$ echo $PATH
```

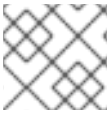
検証

- インストールした **roxctl** のバージョンを確認します。

```
$ roxctl version
```

7.6.3.1.2. macOS への roxctl CLI のインストール

次の手順を使用して、**roxctl** CLI バイナリーを macOS にインストールできます。



注記

macOS 用の **roxctl** CLI は、**amd64** アーキテクチャーで利用できます。

手順

1. **roxctl** CLI をダウンロードします。

```
$ curl -L -f -o roxctl
"https://mirror.openshift.com/pub/rhacs/assets/4.4.4/bin/Darwin/roxctl${arch}"
```

2. バイナリーからすべての拡張属性を削除します。

```
$ xattr -c roxctl
```

3. **roxctl** バイナリーを実行可能にします。

```
$ chmod +x roxctl
```

4. **PATH** 上にあるディレクトリーに **roxctl** バイナリーを配置します。
PATH を確認するには、以下のコマンドを実行します。

```
$ echo $PATH
```

検証

- インストールした **roxctl** のバージョンを確認します。

```
$ roxctl version
```

7.6.3.1.3. Windows への roxctl CLI のインストール

次の手順を使用して、**roxctl** CLI バイナリーを Windows にインストールできます。

注記

Windows 用の **roxctl** CLI は、**amd64** アーキテクチャーで使用できます。

手順

- **roxctl** CLI をダウンロードします。

```
$ curl -f -O https://mirror.openshift.com/pub/rhacs/assets/4.4.4/bin/Windows/roxctl.exe
```

検証

- インストールした **roxctl** のバージョンを確認します。

```
$ roxctl version
```

7.6.3.2. Sensor のインストール

クラスターをモニターするには、Sensor をデプロイする必要があります。モニターする各クラスターに Sensor をデプロイする必要があります。このインストール方法は、マニフェストインストール方法とも呼ばれます。

マニフェストインストール方法を使用してインストールを実行するには、次の手順の **いずれか 1 つだけ** を実行します。

- RHACS Web ポータルを使用してクラスターバンドルをダウンロードし、Sensor スクリプトを展開して実行します。
- **roxctl** CLI を使用して、OpenShift Container Platform クラスターに必要な Sensor 設定を生成し、それを Central インスタンスに関連付けます。

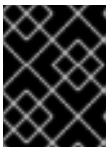
前提条件

- Central サービスがすでにインストールされている。または、Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service) で **ACS インスタンス** を選択すると、Central サービスにアクセスできます。

7.6.3.2.1. Web ポータルを使用したマニフェストインストール方法

手順

1. セキュアクラスターの RHACS ポータルで、**Platform Configuration → Clusters** に移動します。
2. **Secure a cluster → Legacy installation method** を選択します。
3. クラスターの名前を指定します。
4. Sensor をデプロイする場所に基づいて、フィールドに適切な値を入力します。
   - 同じクラスターに Sensor をデプロイする場合は、すべてのフィールドのデフォルト値を受け入れます。
   - 別のクラスターにデプロイする場合は、**central.stackrox.svc:443** を、他のクラスターからアクセス可能なロードバランサー、ノードポート、またはポート番号を含む他のアドレスに置き換えます。
   - HAProxy、AWS Application Load Balancer (ALB)、AWS Elastic Load Balancing (ELB) などの非 gRPC 対応のロードバランサーを使用している場合は、WebSocket Secure (**wss**) プロトコルを使用してください。**wss** を使用するには:
     - アドレスの前に **wss://** を付けます。
     - アドレスの後にポート番号を追加します (例 **wss://stackrox-central.example.com:443**)。
5. **Next** をクリックして、Sensor のセットアップを続行します。
6. **Download YAML File and Keys**をクリックして、クラスターバンドル (zip アーカイブ) をダウンロードします。



重要

クラスターバンドルの zip アーカイブには、クラスターごとに固有の設定とキーが含まれています。同じファイルを別のクラスターで再利用しないでください。

7. 監視対象のクラスターにアクセスできるシステムで、クラスターバンドルから **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
```

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。

7.6.3.2.2. roxctl CLI を使用したマニフェストインストール

手順

1. 以下のコマンドを実行して、OpenShift Container Platform クラスターに必要な Sensor 設定を生成し、Central インスタンスに関連付けます。

```
$ roxctl sensor generate openshift --openshift-version <ocp_version> --name
<cluster_name> --central "$ROX_ENDPOINT" 1
```

1 **--openshift-version** オプションでは、クラスターの主要な OpenShift Container Platform バージョン番号を指定します。たとえば、OpenShift Container Platform バージョン **3.x** の場合は **3** を指定し、OpenShift Container Platform バージョン **4.x** の場合は **4** を指定します。

2. 監視対象のクラスターにアクセスできるシステムで、クラスターバンドルから **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
```

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。

検証

1. 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。成功した場合、**Platform Configuration** → **Clusters** でクラスターのリストを表示すると、クラスターのステータスに緑色のチェックマークと **Healthy** ステータスが表示されます。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 で、次のコマンドを入力します。

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

   - Kubernetes で、次のコマンドを入力します。

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

2. **Finish** をクリックしてウィンドウを閉じます。

インストール後、Sensor はセキュリティー情報の RHACS へのレポートを開始し、RHACS ポータルダッシュボードは、Sensor をインストールしたクラスターからのデプロイメント、イメージ、およびポリシー違反を表示し始めます。

7.6.4. 次のステップ

- セキュアクラスターが ACS インスタンスと通信できることを確認して、インストールを検証します。

7.7. RHACS CLOUD SERVICE でセキュアクラスターサービスのプロキシーを設定する

セキュアクラスターと指定されたプロキシーサーバー間の接続を確立するには、Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service) 環境内でセキュアクラスターサービスのプロキシーを設定する必要があります。これにより、信頼性の高いデータ収集と送信が保証されます。

7.7.1. SecuredCluster CR での環境変数の指定

Egress プロキシーを設定するには、クラスター全体の Red Hat OpenShift プロキシーを使用するか、SecuredCluster カスタムリソース (CR) 設定ファイル内に **HTTP_PROXY**、**HTTPS_PROXY**、および **NO_PROXY** 環境変数を指定します。これにより、プロキシーを適切に使用し、指定ドメイン内の内部要求をバイパスします。

プロキシー設定は、実行中のすべてのサービス (Sensor、Collector、Admission Controller、Scanner) に適用されます。

手順

- SecuredCluster CR 設定ファイルのカスタマイズ仕様で、**HTTP_PROXY**、**HTTPS_PROXY**、および **NO_PROXY** 環境変数を指定します。
  以下に例を示します。

```
# proxy collector
customize:
  envVars:
    - name: HTTP_PROXY
      value: http://egress-proxy.stackrox.svc:xxxx (1)
    - name: HTTPS_PROXY
      value: http://egress-proxy.stackrox.svc:xxxx (2)
    - name: NO_PROXY
      value: .stackrox.svc (3)
```

(1) 変数 **HTTP_PROXY** は **http://egress-proxy.stackrox.svc:xxxx** の値に設定されます。これは、HTTP 接続に使用されるプロキシーサーバーです。

(2) 変数 **HTTPS_PROXY** は、**http://egress-proxy.stackrox.svc:xxxx** 値に設定されます。これは、HTTPS 接続に使用されるプロキシーサーバーです。

(3) 変数 **NO _PROXY** は **.stackrox.svc** に設定されます。この変数は、プロキシーサーバー経由でアクセスすべきではないホスト名または IP アドレスを定義するために使用されます。

7.8. セキュアクラスターのインストールの検証

RHACS Cloud Service をインストールした後、いくつかの手順を実行して、インストールが成功したことを確認できます。

インストールを検証するには、Red Hat Hybrid Cloud Console から ACS コンソールにアクセスします。ダッシュボードには、ノード、デプロイメント、イメージ、および違反に関する情報とともに、RHACS Cloud Service がモニタリングしているクラスターの数が表示されます。

ACS コンソールにデータが表示されない場合:

- 1 つ以上のセキュアクラスターが RHACS Cloud Service インスタンスに接続されていることを確認します。詳細は、RHACS Cloud Service からの保護されたクラスターリソースのインストール を参照してください。
- Sensor Pod のログを調べて、RHACS Cloud Service インスタンスへの接続が成功していることを確認します。
- Red Hat OpenShift クラスターで、**Platform Configuration → Clusters** に移動して、コンポーネントが正常であることを確認し、追加の動作情報を表示します。
- ローカルクラスターの Operator で **SecuredCluster** API の値を調べて、**Central API エンドポイント** が正しく入力されていることを確認します。この値は、Red Hat Hybrid Cloud Console の **ACS インスタンス** の詳細に表示される値と同じである必要があります。

第8章 KUBERNETES で保護されたクラスターを使用した RHACS CLOUD SERVICE のセットアップ

8.1. KUBERNETES クラスター用の RHACS CLOUD SERVICE インスタンスを作成する

Red Hat Hybrid Cloud Console でインスタンスを選択して、Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service) にアクセスします。**ACS インスタンス** には、Red Hat がお客様の代わりに設定および管理する RHACS Cloud Service 管理インターフェイスとサービスが含まれています。管理インターフェイスは、セキュアクラスターに接続します。セキュアクラスターには、脆弱性をスキャンして情報を収集するサービスが含まれています。1 つのインスタンスが多くのクラスターに接続して監視できます。

8.1.1. コンソールでのインスタンスの作成

Red Hat Hybrid Cloud Console で、セキュアクラスターに接続するための **ACS インスタンス** を作成します。

手順

ACS インスタンス を作成するには:

1. Red Hat Hybrid Cloud Console にログインします。
2. ナビゲーションメニューから、**Advanced Cluster Security → ACS Instances** を選択します。
3. **ACS インスタンスの作成** を選択し、表示されたフィールドに情報を入力するか、ドロップダウンリストから適切なオプションを選択します。
   - **Name**: **ACS インスタンス** の名前を入力します。**ACS インスタンス** には、"Central" とも呼ばれる RHACS Central コンポーネントが含まれています。このコンポーネントには、RHACS Cloud Service 管理インターフェイスと、Red Hat によって設定および管理されるサービスが含まれています。お客様は、Central と通信するセキュアクラスターを管理します。多くのセキュアクラスターを 1 つのインスタンスに接続できます。
   - **クラウドプロバイダー**: Central が配置されているクラウドプロバイダー。**AWS** を選択します。
   - **クラウドリージョン**: Central が配置されているクラウドプロバイダーのリージョン。次のいずれかのリージョンを選択します。
     - 米国東部、バージニア北部
     - ヨーロッパ、アイルランド
   - **アベイラビリティーゾーン**: デフォルト値 (**Multi**) を使用します。
4. **Create instance** をクリックします。

8.1.2. 次のステップ

- 保護対象の各 Kubernetes クラスターで、Helm チャートまたは **roxctl** CLI を使用して、セキュアクラスターのリソースをインストール します。

8.2. KUBERNETES のセキュアクラスター用の INIT バンドルを生成する

SecuredCluster リソースをクラスターにインストールする前に、init バンドルを作成する必要があります。**SecuredCluster** がインストールおよび設定されているクラスターは、このバンドルを使用して ACS コンソールで認証します。RHACS ポータルまたは **roxctl** CLI を使用して、init バンドルを作成できます。次に、それを使用してリソースを作成することにより、init バンドルを適用します。

8.2.1. RHACS ポータルを使用した init バンドルの生成

RHACS ポータルを使用して、シークレットを含む init バンドルを作成できます。

注記

init バンドルを作成するには、**Admin** ユーザーロールが必要です。

手順

1. 「Operator 方式を使用した Central インストールの検証」の説明に従って、RHACS ポータルのアドレスを見つけます。
2. RHACS ポータルにログインします。
3. セキュアクラスターがない場合は、**Platform Configuration** → **Clusters** ページが表示されます。
4. **Create init bundle**をクリックします。
5. クラスター init バンドルの名前を入力します。
6. プラットフォームを選択します。
7. セキュアクラスターに使用するインストール方法 (**Operator** または **Helm chart**) を選択します。
8. **Download** をクリックし、init バンドルを生成してダウンロードします。init バンドルは YAML ファイル形式で作成されます。同じインストール方法を使用する場合は、すべてのセキュアクラスターに対して 1 つの init バンドルとそれに対応する YAML ファイルを使用できます。

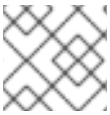
重要

このバンドルにはシークレットが含まれているため、セキュアに保管してください。

9. init バンドルを使用して適用し、セキュアクラスター上にリソースを作成します。
10. 各クラスターにセキュアクラスターサービスをインストールします。

8.2.2. roxctl CLI を使用した init バンドルの生成

roxctl CLI を使用して、シークレットを含む init バンドルを作成できます。



注記

init バンドルを作成するには、**Admin** ユーザーロールが必要です。

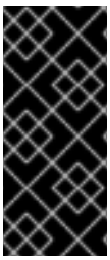
前提条件

- **ROX_API_TOKEN** および **ROX_CENTRAL_ADDRESS** 環境変数が設定されている。
  a. 次のコマンドを実行して **ROX_API_TOKEN** を設定します。

```
$ export ROX_API_TOKEN=<api_token>
```

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

```
$ export ROX_CENTRAL_ADDRESS=<address>:<port_number>
```



重要

RHACS Cloud Service では、Central アドレスを必要とする **roxctl** コマンドを使用する場合は、Red Hat Hybrid Cloud Console の **インスタンスの詳細** セクションに表示される **Central インスタンスのアドレス** を使用します。たとえば、**acs-data-ABCD12345.acs.rhcloud.com** の代わりに **acs-ABCD12345.acs.rhcloud.com** を使用します。

手順

- Helm インストールのシークレットを含むクラスター初期化バンドルを生成するには、次のコマンドを実行します。

```
$ roxctl -e "$ROX_CENTRAL_ADDRESS" \
  central init-bundles generate <cluster_init_bundle_name> \
  --output cluster_init_bundle.yaml
```

- Operator インストール用のシークレットを含むクラスター初期化バンドルを生成するには、次のコマンドを実行します。

```
$ roxctl -e "$ROX_CENTRAL_ADDRESS" \
  central init-bundles generate <cluster_init_bundle_name> \
  --output-secrets cluster_init_bundle.yaml
```

重要

このバンドルにはシークレットが含まれているため、安全に保管してください。同じバンドルを使用して、複数のセキュアクラスターを設定できます。

8.2.3. 次のステップ

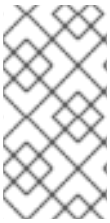
- init バンドルを使用したリソースの作成

8.3. KUBERNETES のセキュアクラスター用の INIT バンドルを適用する

init バンドルを使用してリソースを作成し、それを適用します。

8.3.1. セキュアクラスターに init バンドルを適用する

セキュアクラスターを設定する前に、init バンドルを使用してそれを適用し、セキュアクラスター上に必要なリソースを作成する必要があります。init バンドルを適用すると、セキュアクラスター上のサービスが RHACS Cloud Service と通信できるようになります。



注記

Helm チャートを使用してインストールする場合は、この手順を実行しないでください。Helm を使用してインストールを完了してください。関連情報セクションの「Helm チャートを使用したセキュアクラスターへの RHACS のインストール」を参照してください。

前提条件

- シークレットを含む init バンドルを生成している必要があります。
- セキュアクラスターサービスをインストールするクラスター上に、**stackrox** プロジェクトまたは namespace を作成した。プロジェクトとして **stackrox** を使用することは必須ではありませんが、使用すると、クラスターのスキャン時に RHACS プロセスの脆弱性が報告されなくなります。

手順

- **kubectl** CLI を使用して、次のコマンドを実行してリソースを作成します。

```
$ kubectl create namespace stackrox 1
$ kubectl create -f <init_bundle>.yaml \ 2
  -n <stackrox> 3
```

1. セキュアクラスターのリソースをインストールするプロジェクトを作成します。この例では **stackrox** を使用します。
2. シークレットを含む init バンドルのファイル名を指定します。
3. 作成したプロジェクト名を指定します。この例では **stackrox** を使用します。

検証

- 新しい証明書を取得するには、センサーを再起動します。
  Sensor を再起動する方法の詳細は、「関連情報」セクションの「Sensor コンテナーの再起動」を参照してください。

8.3.2. 次のステップ

- 監視するすべてのクラスターに RHACS のセキュアクラスターサービスをインストールします。

8.3.3. 関連情報

- Sensor コンテナーの再起動

8.4. RHACS CLOUD SERVICE から KUBERNETES クラスターに安全なクラスターサービスをインストールする

次のいずれかの方法を使用して、セキュアクラスターに RHACS Cloud Service をインストールできます。

- Helm チャートを使用する
- **roxctl** CLI を使用する (この方法を使用する必要がある特定のインストールが必要でない限り、この方法は使用しないでください)

8.4.1. Helm チャートを使用したセキュアクラスターへの RHACS Cloud Service のインストール

カスタマイズなしの Helm チャート、デフォルト値の Helm チャート、または設定パラメーターをカスタマイズした Helm チャートを使用して、セキュアクラスターにに RHACS をインストールできます。

最初に、Helm チャートリポジトリーを追加していることを確認します。

8.4.1.1. Helm チャートリポジトリーの追加

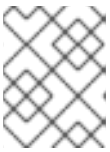
手順

- RHACS チャートリポジトリーを追加します。

```
$ helm repo add rhacs https://mirror.openshift.com/pub/rhacs/charts/
```

Red Hat Advanced Cluster Security for Kubernetes の Helm リポジトリーには、次のようなさまざまなコンポーネントをインストールするための Helm チャートが含まれています。

- クラスターおよびノードごとのコンポーネント (Sensor、Admission Controller、Collector、および Scanner-slim) をインストールするためのセキュアクラスターサービスの Helm チャート (**secured-cluster-services**)。



注記

モニターする各クラスターにクラスターごとのコンポーネントをデプロイし、モニターするすべてのノードにノードごとのコンポーネントをデプロイします。

検証

- 次のコマンドを実行して、追加されたチャートリポジトリーを確認します。

```
$ helm search repo -l rhacs/
```

8.4.1.2. カスタマイズせずに Helm チャートを使用してセキュアクラスターに RHACS Cloud Service をインストールする

8.4.1.2.1. カスタマイズせずに secured-cluster-services Helm チャートをインストールする

次の手順に従って、**secured-cluster-services** Helm チャートをインストールし、クラスターおよびノードごとのコンポーネント (Sensor、Admission コントローラー、Collector、および Scanner-slim) をデプロイします。

前提条件

- クラスターの RHACS init バンドルを生成しておく必要があります。
- Red Hat コンテナーレジストリーへのアクセス権と、認証用のプルシークレットが必要です。**registry.redhat.io** からイメージをダウンロードする方法は、Red Hat コンテナーレジストリーの認証 を参照してください。
- アドレスとポート番号を含む **Central API Endpoint** が必要です。この情報を表示するには、クラウドコンソールのナビゲーションメニューから **Advanced Cluster Security → ACS Instances** を選択し、作成した ACS インスタンスをクリックします。

手順

- Kubernetes ベースのクラスターで次のコマンドを実行します。

```
$ helm install -n stackrox --create-namespace \
  stackrox-secured-cluster-services rhacs/secured-cluster-services \
  -f <path_to_cluster_init_bundle.yaml> \ 1
  -f <path_to_pull_secret.yaml> \ 2
  --set clusterName=<name_of_the_secured_cluster> \
  --set centralEndpoint=<endpoint_of_central_service> 3
  --set imagePullSecrets.username=<your redhat.com username> \ 4
  --set imagePullSecrets.password=<your redhat.com password> 5
```

1. **-f** オプションを使用して、init バンドルのパスを指定します。
2. -f オプションを使用して、Red Hat コンテナーレジストリー認証用のプルシークレットのパスを指定します。
3. アドレスとポート番号を含む Central API エンドポイントを入力します。Advanced Cluster Security → ACS Instances を選択し、作成した ACS インスタンスをクリックすると、Red Hat Hybrid Cloud Console コンソールでこの情報を再度表示できます。
4. Red Hat コンテナーレジストリー認証のプルシークレットのユーザー名を含めます。
5. Red Hat コンテナーレジストリー認証のプルシークレットのパスワードを含めます。

8.4.1.3. カスタマイズした secured-cluster-services Helm チャートの設定

このセクションでは、**helm install** および **helm upgrade** コマンドで使用できる Helm チャート設定パラメーターを説明します。これらのパラメーターは、**--set** オプションを使用するか、YAML 設定ファイルを作成することで指定できます。

以下のファイルを作成して、Red Hat Advanced Cluster Security for Kubernetes をインストールするための Helm チャートを設定します。

- パブリック設定ファイル **values-public.yaml**: このファイルを使用して、機密性の低いすべての設定オプションを保存します。
- プライベート設定ファイル **values-private.yaml**: このファイルを使用して、機密性の高いすべての設定オプションを保存します。このファイルは安全に保管してください。

重要

secured-cluster-services Helm チャートを使用している間は、チャートの一部である **values.yaml** ファイルを変更しないでください。

8.4.1.3.1. 設定パラメーター

パラメーター	説明
clusterName	クラスターの名前です。
centralEndpoint	Central エンドポイントのアドレス (ポート番号を含む)。gRPC に対応していないロードバランサーを使用している場合は、エンドポイントアドレスの前に **wss://** を付けて、WebSocket プロトコルを使用します。複数のクラスターを設定する場合は、アドレスにホスト名を使用します (例: **central.example.com:443**)。
sensor.endpoint	ポート番号を含む Sensor エンドポイントのアドレスです。
sensor.imagePullPolicy	Sensor コンテナーのイメージプルポリシーです。
sensor.serviceTLS.cert	Sensor が使用する内部サービス間の TLS 証明書です。
sensor.serviceTLS.key	Sensor が使用する内部サービス間 TLS 証明書キーです。
sensor.resources.requests.memory	Sensor コンテナーのメモリーリクエストです。このパラメーターを使用して、デフォルト値をオーバーライドします。
sensor.resources.requests.cpu	Sensor コンテナーの CPU リクエストです。このパラメーターを使用して、デフォルト値をオーバーライドします。
sensor.resources.limits.memory	Sensor コンテナーのメモリー制限です。このパラメーターを使用して、デフォルト値をオーバーライドします。
sensor.resources.limits.cpu	Sensor コンテナーの CPU 制限です。このパラメーターを使用して、デフォルト値をオーバーライドします。
sensor.nodeSelector	ノードセレクターのラベルを **label-key: label-value** の形式で指定して、指定したラベルを持つノードでのみ Sensor をスケジュールするように強制します。

パラメーター	説明
sensor.tolerations	ノードセレクターが taint されたノードを選択する場合は、このパラメーターを使用して、Sensor の taint toleration キー、値、および effect を指定します。このパラメーターは、主にインフラストラクチャーノードに使用されます。
image.main.name	**main** イメージの名前です。
image.collector.name	Collector イメージの名前です。
image.main.registry	main イメージに使用しているレジストリーのアドレスです。
image.collector.registry	Collector イメージに使用しているレジストリーのアドレスです。
image.scanner.registry	Scanner イメージに使用しているレジストリーのアドレスです。
image.scannerDb.registry	Scanner DB イメージに使用しているレジストリーのアドレスです。
image.scannerV4.registry	Scanner V4 イメージに使用しているレジストリーのアドレスです。
image.scannerV4DB.registry	Scanner V4 DB イメージに使用しているレジストリーのアドレスです。
image.main.pullPolicy	**main** イメージのイメージプルポリシーです。
image.collector.pullPolicy	Collector イメージのイメージプルポリシーです。
image.main.tag	使用する **main** イメージのタグです。
image.collector.tag	使用する **collector** イメージのタグです。
collector.collectionMethod	**CORE_BPF**、**EBPF** (非推奨)、**NO_COLLECTION** のいずれか。
collector.imagePullPolicy	Collector コンテナーのイメージプルポリシーです。
collector.complianceImagePullPolicy	Compliance コンテナーのイメージプルポリシーです。

パラメーター	説明
collector.disableTaintTolerations	**false** を指定すると、許容値が Collector に適用され、Collector Pod は taint のあるすべてのノードにスケジュールできます。**true** として指定すると、許容値は適用されず、Collector Pod は taint のあるノードにスケジュールされません。
collector.resources.requests.memory	Collector コンテナーのメモリーリクエストです。このパラメーターを使用して、デフォルト値をオーバーライドします。
collector.resources.requests.cpu	Collector コンテナーの CPU リクエストです。このパラメーターを使用して、デフォルト値をオーバーライドします。
collector.resources.limits.memory	Collector コンテナーのメモリー制限です。このパラメーターを使用して、デフォルト値をオーバーライドします。
collector.resources.limits.cpu	Collector コンテナーの CPU 制限です。このパラメーターを使用して、デフォルト値をオーバーライドします。
collector.complianceResources.requests.memory	Compliance コンテナーのメモリーリクエストです。このパラメーターを使用して、デフォルト値をオーバーライドします。
collector.complianceResources.requests.cpu	Compliance の CPU リクエストです。このパラメーターを使用して、デフォルト値をオーバーライドします。
collector.complianceResources.limits.memory	Compliance コンテナーのメモリー制限です。このパラメーターを使用して、デフォルト値をオーバーライドします。
collector.complianceResources.limits.cpu	Compliance コンテナーの CPU 制限です。このパラメーターを使用して、デフォルト値をオーバーライドします。
collector.serviceTLS.cert	Collector が使用する内部サービス間 TLS 証明書です。
collector.serviceTLS.key	Collector が使用する内部サービス間 TLS 証明書キーです。

パラメーター	説明
admissionControl.listenOnCreates	この設定は、ワークロード作成イベントの **AdmissionReview** リクエストで Red Hat Advanced Cluster Security for Kubernetes に接続するように Kubernetes を設定するかどうかを制御します。
admissionControl.listenOnUpdates	このパラメーターを **false** に設定すると、Red Hat Advanced Cluster Security for Kubernetes は、Kubernetes API サーバーがオブジェクト更新イベントを送信しないようにする **ValidatingWebhookConfiguration** を作成します。オブジェクトの更新ボリュームは通常、オブジェクトが作成するボリュームよりも多いため、これを **false** のままにしておくと、アドミッションコントロールサービスのロードが制限され、アドミッションコントロールサービスが誤動作する可能性が低くなります。
admissionControl.listenOnEvents	この設定は、Kubernetes **exec** および **portforward** イベントの **AdmissionReview** リクエストで Red Hat Advanced Cluster Security for Kubernetes に接続するようにクラスターを設定するかどうかを制御します。RHACS は、OpenShift Container Platform 3.11 ではこの機能をサポートしていません。
admissionControl.dynamic.enforceOnCreates	この設定は、Red Hat Advanced Cluster Security for Kubernetes がポリシーを評価するかどうかを制御します。無効にすると、すべての AdmissionReview リクエストが自動的に承認されます。
admissionControl.dynamic.enforceOnUpdates	この設定は、アドミッションコントロールサービスの動作を制御します。これを機能させるには、**listenOnUpdates** を **true** として指定する必要があります。
admissionControl.dynamic.scanInline	このオプションを **true** に設定すると、アドミッションコントロールサービスは、アドミッションデシジョンを行う前にイメージスキャンをリクエストします。イメージスキャンには数秒かかるため、このオプションを有効にするのは、クラスターで使用されるすべてのイメージがデプロイ前にスキャンされることを確認できる場合のみです (たとえば、イメージビルド中の CI 統合によって)。このオプションは、RHACS ポータルの **Contact image scanners** オプションに対応します。
admissionControl.dynamic.disableBypass	Admission コントローラーのバイパスを無効にするには、**true** に設定します。

パラメーター	説明
admissionControl.dynamic.timeout	アドミッションレビューリクエストを評価する間、Red Hat Advanced Cluster Security for Kubernetes が待機する最大時間 (秒単位) です。これを使用して、イメージスキャンを有効にするときにリクエストのタイムアウトを設定します。イメージスキャンの実行時間が指定した時間より長い場合、Red Hat Advanced Cluster Security for Kubernetes はリクエストを受け入れます。
admissionControl.resources.requests.memory	Admission Control コンテナーのメモリーリクエストです。このパラメーターを使用して、デフォルト値をオーバーライドします。
admissionControl.resources.requests.cpu	Admission Control コンテナーの CPU リクエストです。このパラメーターを使用して、デフォルト値をオーバーライドします。
admissionControl.resources.limits.memory	Admission Control コンテナーのメモリー制限です。このパラメーターを使用して、デフォルト値をオーバーライドします。
admissionControl.resources.limits.cpu	Admission Control コンテナーの CPU 制限です。このパラメーターを使用して、デフォルト値をオーバーライドします。
admissionControl.nodeSelector	ノードセレクターのラベルを **label-key: label-value** の形式で指定して、指定したラベルを持つノードでのみ Admission Control をスケジュールするように強制します。
admissionControl.tolerations	ノードセレクターが taint されたノードを選択する場合は、このパラメーターを使用して、アドミッションコントロールの taint toleration キー、値、および effect を指定します。このパラメーターは、主にインフラストラクチャーノードに使用されます。
admissionControl.serviceTLS.cert	Admission Control が使用する内部サービス間 TLS 証明書です。
admissionControl.serviceTLS.key	Admission Control が使用する内部サービス間 TLS 証明書キーです。
registryOverride	このパラメーターを使用して、デフォルトの **docker.io** レジストリーをオーバーライドします。他のレジストリーを使用している場合は、レジストリーの名前を指定してください。

パラメーター	説明
collector.disableTaintTolerations	**false** を指定すると、許容値が Collector に適用され、Collector Pod は taint のあるすべてのノードにスケジュールできます。**true** として指定した場合、許容値は適用されず、Collector Pod は taint のあるノードにスケジュールされません。
createUpgraderServiceAccount	**sensor-upgrader** アカウントを作成するには、**true** を指定します。デフォルトでは、Red Hat Advanced Cluster Security for Kubernetes は、各セキュアクラスターに **sensor-upgrader** という名前のサービスアカウントを作成します。このアカウントは高い権限を持ちますが、アップグレードの時のみ使用されます。このアカウントを作成しない場合、Sensor に十分な権限がない場合は、将来のアップグレードを手動で完了する必要があります。
createSecrets	**false** を指定すると、Sensor、Collector、および Admission コントローラーのオーケストレーターシークレットの作成がスキップされます。
collector.slimMode	Collector のデプロイに slim Collector イメージを使用する場合は、**true** を指定します。EBPF 収集方法で slim Collector イメージを使用するには、対応する eBPF プローブを Central で提供する必要があります。Red Hat Advanced Cluster Security for Kubernetes をオフラインモードで実行している場合、slim Collector を機能させるために、stackrox.io からカーネルサポートパッケージをダウンロードして Central にアップロードする必要があります。それ以外の場合は、Central が https://collector-modules.stackrox.io/ でホストされているオンラインプローブリポジトリーにアクセスできることを確認する必要があります。
sensor.resources	Sensor のリソース仕様です。
admissionControl.resources	Admission コントローラーのリソース仕様です。
collector.resources	Collector のリソース仕様です。
collector.complianceResources	Collector の Compliance コンテナーのリソース仕様です。
exposeMonitoring	このオプションを **true** に設定すると、Red Hat Advanced Cluster Security for Kubernetes がポート番号 9090 で Sensor、Collector、および Admission コントローラーの Prometheus メトリクスエンドポイントを公開します。

パラメーター	説明
auditLogs.disableCollection	このオプションを **true** に設定すると、Red Hat Advanced Cluster Security for Kubernetes が、設定マップとシークレットへのアクセスと変更を検出するために使用される監査ログ検出機能を無効にします。
scanner.disable	このオプションを **false** に設定すると、Red Hat Advanced Cluster Security for Kubernetes がセキュアクラスターに Scanner-slim と Scanner DB をデプロイし、OpenShift Container Registry 上のイメージをスキャンできるようにします。Scanner-slim の有効化は、OpenShift Container Platform および Kubernetes セキュアクラスターでサポートされています。デフォルトは **true** です。
scanner.dbTolerations	ノードセレクターが taint されたノードを選択する場合は、このパラメーターを使用して、Scanner DB の taint toleration キー、値、および effect を指定します。
scanner.replicas	Collector の Compliance コンテナーのリソース仕様です。
scanner.logLevel	このパラメーターを設定すると、Scanner のログレベルを変更できます。このオプションは、トラブルシューティングの目的でのみ使用してください。
scanner.autoscaling.disable	このオプションを **true** に設定すると、Red Hat Advanced Cluster Security for Kubernetes が Scanner デプロイメントでの自動スケーリングを無効にします。
scanner.autoscaling.minReplicas	自動スケーリングのレプリカの最小数です。デフォルトは 2 です。
scanner.autoscaling.maxReplicas	自動スケーリングのレプリカの最大数です。デフォルトは 5 です。
scanner.nodeSelector	ノードセレクターのラベルを **label-key: label-value** の形式で指定して、指定したラベルを持つノードでのみ Scanner をスケジュールするように強制します。
scanner.tolerations	ノードセレクターが taint されたノードを選択する場合は、このパラメーターを使用して、Scanner の taint toleration キー、値、および effect を指定します。

パラメーター	説明
scanner.dbNodeSelector	ノードセレクターのラベルを **label-key: label-value** の形式で指定して、指定したラベルを持つノードでのみ Scanner DB をスケジュールするように強制します。
scanner.dbTolerations	ノードセレクターが taint されたノードを選択する場合は、このパラメーターを使用して、Scanner DB の taint toleration キー、値、および effect を指定します。
scanner.resources.requests.memory	Scanner コンテナーのメモリー要求。このパラメーターを使用して、デフォルト値をオーバーライドします。
scanner.resources.requests.cpu	Scanner コンテナーの CPU 要求。このパラメーターを使用して、デフォルト値をオーバーライドします。
scanner.resources.limits.memory	Scanner コンテナーのメモリー制限。このパラメーターを使用して、デフォルト値をオーバーライドします。
scanner.resources.limits.cpu	Scanner コンテナーの CPU 制限。このパラメーターを使用して、デフォルト値をオーバーライドします。
scanner.dbResources.requests.memory	Scanner DB コンテナーのメモリー要求。このパラメーターを使用して、デフォルト値をオーバーライドします。
scanner.dbResources.requests.cpu	Scanner DB コンテナーの CPU 要求。このパラメーターを使用して、デフォルト値をオーバーライドします。
scanner.dbResources.limits.memory	Scanner DB コンテナーのメモリー制限。このパラメーターを使用して、デフォルト値をオーバーライドします。
scanner.dbResources.limits.cpu	Scanner DB コンテナーの CPU 制限。このパラメーターを使用して、デフォルト値をオーバーライドします。
monitoring.openshift.enabled	このオプションを **false** に設定すると、Red Hat Advanced Cluster Security for Kubernetes が Red Hat OpenShift モニタリングをセットアップしません。Red Hat OpenShift 4 では、デフォルトで **true** に設定されます。

8.4.1.3.1.1. 環境変数

Sensor および Admission コントローラーの環境変数は、次の形式で指定できます。

```
customize:
  envVars:
    ENV_VAR1: "value1"
    ENV_VAR2: "value2"
```

customize 設定を使用すると、この Helm チャートによって作成されたすべてのオブジェクトのカスタム Kubernetes メタデータ (ラベルとアノテーション) と、ワークロードの追加の Pod ラベル、Pod アノテーション、コンテナー環境変数を指定できます。

より一般的なスコープ (たとえば、すべてのオブジェクト) で定義されたメタデータを、より狭いスコープ (たとえば、Sensor デプロイメントのみ) で定義されたメタデータでオーバーライドできるという意味で、設定は階層的です。

8.4.1.3.2. カスタマイズした secured-cluster-services Helm チャートのインストール

values-public.yaml ファイルと **values-private.yaml** ファイルを設定したら、**secured-cluster-services** Helm チャートをインストールして、次のクラスターおよびノードごとのコンポーネントをデプロイします。

- Sensor
- Admission コントローラー
- Collector
- Scanner: StackRox Scanner がインストールされている場合、必要に応じてセキュアクラスターにデプロイする
- Scanner DB: StackRox Scanner がインストールされている場合、必要に応じてセキュアクラスターにデプロイする
- Scanner V4 Indexer および Scanner V4 DB: Scanner V4 がインストールされている場合、必要に応じてセキュアクラスターにデプロイする

重要

Scanner V4 はテクノロジープレビューのみの機能です。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat では、実稼働環境での使用を推奨していません。テクノロジープレビューの機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

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

前提条件

- クラスターの RHACS init バンドルを生成しておく必要があります。
- Red Hat コンテナーレジストリーへのアクセス権と、認証用のプルシークレットが必要です。**registry.redhat.io** からイメージをダウンロードする方法は、Red Hat コンテナーレジストリーの認証 を参照してください。

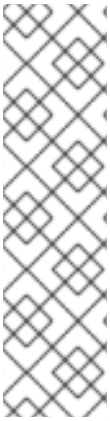
- アドレスとポート番号を含む **Central API Endpoint** が必要です。この情報を表示するには、クラウドコンソールのナビゲーションメニューから **Advanced Cluster Security → ACS Instances** を選択し、作成した ACS インスタンスをクリックします。

手順

- 以下のコマンドを実行します。

```
$ helm install -n stackrox \
  --create-namespace stackrox-secured-cluster-services rhacs/secured-cluster-services \
  -f <name_of_cluster_init_bundle.yaml> \
  -f <path_to_values_public.yaml> -f <path_to_values_private.yaml> \ 1
  --set imagePullSecrets.username=<username> \ 2
  --set imagePullSecrets.password=<password> 3
```

1. **-f** オプションを使用して、YAML 設定ファイルのパスを指定します。
2. Red Hat コンテナーレジストリー認証のプルシークレットのユーザー名を含めます。
3. Red Hat コンテナーレジストリー認証のプルシークレットのパスワードを含めます。



注記

継続的インテグレーション (CI) システムを使用して **secured-cluster-services** Helm チャートをデプロイするには、init バンドル YAML ファイルを環境変数として **helm install** コマンドに渡します。

```
$ helm install ... -f <(echo "$INIT_BUNDLE_YAML_SECRET") 1
```

1. base64 でエンコードされた変数を使用している場合は、代わりに **helm install ... -f <(echo "$INIT_BUNDLE_YAML_SECRET" | base64 --decode)** コマンドを使用してください。

8.4.1.4. secured-cluster-services Helm チャートをデプロイした後の設定オプションの変更

secured-cluster-services Helm チャートをデプロイした後、設定オプションを変更できます。

helm upgrade コマンドを使用して変更を加える場合は、次のガイドラインと要件が適用されます。

- **--set** または **--set-file** パラメーターを使用して設定値を指定することもできます。ただし、これらのオプションは保存されないため、変更を加えるたびにすべてのオプションを手動で再度指定する必要があります。
- 変更の内容によっては (たとえば Scanner V4 などの新しいコンポーネントを有効にした場合は)、コンポーネントに対して新しい証明書を発行する必要があります。したがって、これらの変更を行う場合は CA を指定する必要があります。

重要

Scanner V4 はテクノロジープレビューのみの機能です。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat では、実稼働環境での使用を推奨していません。テクノロジープレビューの機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

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

- CA が初期インストール中に Helm チャートによって生成された場合は、自動的に生成された該当する値をクラスターから取得し、**helm upgrade** コマンドで指定する必要があります。**central-services** Helm チャートのインストール後の注記に、自動生成された値を取得するためのコマンドが含まれています。
- CA が Helm チャートの外部で生成されたものであり、**central-services** チャートのインストール時にその CA を指定した場合は、**helm upgrade** コマンドを使用するときに、たとえば **helm upgrade** コマンドで **--reuse-values** フラグを使用して、その操作を再度実行する必要があります。

手順

1. **values-public.yaml** および **values-private.yaml** 設定ファイルを新しい値で更新します。
2. **helm upgrade** コマンドを実行し、**-f** オプションを使用して設定ファイルを指定します。

```
$ helm upgrade -n stackrox \
  stackrox-secured-cluster-services rhacs/secured-cluster-services \
  --reuse-values \ 1
  -f <path_to_values_public.yaml> \
  -f <path_to_values_private.yaml>
```

1 **values_public.yaml** ファイルと **values_private.yaml** ファイルに含まれていない値を変更した場合は、**--reuse-values** パラメーターを含めます。

8.4.2. roxctl CLI を使用したセキュアクラスターへの RHACS のインストール

CLI を使用してセキュアクラスターに RHACS をインストールするには、次の手順を実行します。

1. **roxctl** CLI をインストールします。
2. Sensor を取り付けます。

8.4.2.1. roxctl CLI のインストール

最初にバイナリーをダウンロードする必要があります。**roxctl** は、Linux、Windows、または macOS にインストールできます。

8.4.2.1.1. Linux への roxctl CLI のインストール

次の手順を使用して、Linux に **roxctl** CLI バイナリーをインストールできます。

注記

Linux 用の **roxctl** CLI は、**amd64**、**ppcl64le**、および **s390x** アーキテクチャーで使用できます。

手順

1. ターゲットのオペレーティングシステムの **roxctl** アーキテクチャーを確認します。

```
$ arch="$(uname -m | sed "s/x86_64//")"; arch="${arch:+-$arch}"
```

2. **roxctl** CLI をダウンロードします。

```
$ curl -L -f -o roxctl
"https://mirror.openshift.com/pub/rhacs/assets/4.4.4/bin/Linux/roxctl${arch}"
```

3. **roxctl** バイナリーを実行可能にします。

```
$ chmod +x roxctl
```

4. **PATH** 上にあるディレクトリーに **roxctl** バイナリーを配置します。
PATH を確認するには、以下のコマンドを実行します。

```
$ echo $PATH
```

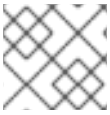
検証

- インストールした **roxctl** のバージョンを確認します。

```
$ roxctl version
```

8.4.2.1.2. macOS への roxctl CLI のインストール

次の手順を使用して、**roxctl** CLI バイナリーを macOS にインストールできます。



注記

macOS 用の **roxctl** CLI は、**amd64** アーキテクチャーで利用できます。

手順

1. **roxctl** CLI をダウンロードします。

```
$ curl -L -f -o roxctl
"https://mirror.openshift.com/pub/rhacs/assets/4.4.4/bin/Darwin/roxctl${arch}"
```

2. バイナリーからすべての拡張属性を削除します。

```
$ xattr -c roxctl
```

3. **roxctl** バイナリーを実行可能にします。

```
$ chmod +x roxctl
```

4. **PATH** 上にあるディレクトリーに **roxctl** バイナリーを配置します。
PATH を確認するには、以下のコマンドを実行します。

```
$ echo $PATH
```

検証

- インストールした **roxctl** のバージョンを確認します。

```
$ roxctl version
```

8.4.2.1.3. Windows への roxctl CLI のインストール

次の手順を使用して、**roxctl** CLI バイナリーを Windows にインストールできます。

注記

Windows 用の **roxctl** CLI は、**amd64** アーキテクチャーで使用できます。

手順

- **roxctl** CLI をダウンロードします。

```
$ curl -f -O https://mirror.openshift.com/pub/rhacs/assets/4.4.4/bin/Windows/roxctl.exe
```

検証

- インストールした **roxctl** のバージョンを確認します。

```
$ roxctl version
```

8.4.2.2. Sensor のインストール

クラスターをモニターするには、Sensor をデプロイする必要があります。モニターする各クラスターに Sensor をデプロイする必要があります。このインストール方法は、マニフェストインストール方法とも呼ばれます。

マニフェストインストール方法を使用してインストールを実行するには、次の手順の **いずれか 1 つだけ** を実行します。

- RHACS Web ポータルを使用してクラスターバンドルをダウンロードし、Sensor スクリプトを展開して実行します。
- **roxctl** CLI を使用して、OpenShift Container Platform クラスターに必要な Sensor 設定を生成し、それを Central インスタンスに関連付けます。

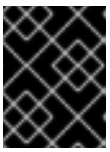
前提条件

- Central サービスがすでにインストールされている。または、Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service) で **ACS インスタンス** を選択すると、Central サービスにアクセスできます。

8.4.2.2.1. Web ポータルを使用したマニフェストインストール方法

手順

1. セキュアクラスターの RHACS ポータルで、**Platform Configuration → Clusters** に移動します。
2. **Secure a cluster → Legacy installation method** を選択します。
3. クラスターの名前を指定します。
4. Sensor をデプロイする場所に基づいて、フィールドに適切な値を入力します。
   - アドレスとポート番号を含む Central API エンドポイントを入力します。**Advanced Cluster Security → ACS Instances** を選択し、作成した ACS インスタンスをクリックすると、Red Hat Hybrid Cloud Console でこの情報を再度表示できます。
5. **Next** をクリックして、Sensor のセットアップを続行します。
6. **Download YAML File and Keys**をクリックして、クラスターバンドル (zip アーカイブ) をダウンロードします。



重要

クラスターバンドルの zip アーカイブには、クラスターごとに固有の設定とキーが含まれています。同じファイルを別のクラスターで再利用しないでください。

7. 監視対象のクラスターにアクセスできるシステムで、クラスターバンドルから **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
```

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。

8.4.2.2.2. roxctl CLI を使用したマニフェストインストール

手順

1. 以下のコマンドを実行して、OpenShift Container Platform クラスターに必要な Sensor 設定を生成し、Central インスタンスに関連付けます。

```
$ roxctl sensor generate openshift --openshift-version <ocp_version> --name
<cluster_name> --central "$ROX_ENDPOINT" 1
```

1 **--openshift-version** オプションでは、クラスターの主要な OpenShift Container Platform バージョン番号を指定します。たとえば、OpenShift Container Platform バージョン **3.x** の場合は **3** を指定し、OpenShift Container Platform バージョン **4.x** の場合は **4** を指定します。

2. 監視対象のクラスターにアクセスできるシステムで、クラスターバンドルから **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
```

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。

検証

1. 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。成功した場合、**Platform Configuration → Clusters** でクラスターのリストを表示すると、クラスターのステータスに緑色のチェックマークと **Healthy** ステータスが表示されます。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。

   - Kubernetes で、次のコマンドを入力します。

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

2. **Finish** をクリックしてウィンドウを閉じます。

インストール後、Sensor はセキュリティー情報の RHACS へのレポートを開始し、RHACS ポータルダッシュボードは、Sensor をインストールしたクラスターからのデプロイメント、イメージ、およびポリシー違反を表示し始めます。

8.5. セキュアクラスターのインストールの検証

RHACS Cloud Service をインストールした後、いくつかの手順を実行して、インストールが成功したことを確認できます。

インストールを検証するには、Red Hat Hybrid Cloud Console から ACS コンソールにアクセスします。ダッシュボードには、ノード、デプロイメント、イメージ、および違反に関する情報とともに、RHACS Cloud Service がモニタリングしているクラスターの数が表示されます。

ACS コンソールにデータが表示されない場合:

- 1 つ以上のセキュアクラスターが RHACS Cloud Service インスタンスに接続されていることを確認します。詳細は、Helm チャート または **roxctl** CLI を使用して、インストール手順を参照してください。

- Sensor Pod のログを調べて、RHACS Cloud Service インスタンスへの接続が成功していることを確認します。

- ローカルクラスターの Operator で **SecuredCluster** API の値を調べて、**Central API エンドポイント** が正しく入力されていることを確認します。この値は、Red Hat Hybrid Cloud Console の **ACS インスタンス** の詳細に表示される値と同じである必要があります。

第9章 RHACS CLOUD SERVICE のアップグレード

9.1. OPERATOR を使用した RHACS CLOUD SERVICE でのセキュアクラスターのアップグレード

Red Hat は、Central サービスを含む管理対象のコンポーネントに対して定期的なサービス更新を提供します。このサービス更新には、Red Hat Advanced Cluster Security Cloud Service の新しいバージョンへのアップグレードも含まれます。

RHACS Cloud Service との互換性を確保するには、セキュアクラスター上の RHACS のバージョンを定期的にアップグレードする必要があります。

9.1.1. アップグレードへの準備

Red Hat Advanced Cluster Security for Kubernetes (RHACS) のバージョンをアップグレードする前に、次の手順を実行します。

- アップグレードするクラスターに **SecuredCluster** カスタムリソース (CR) が含まれている場合は、収集方法を **CORE_BPF** に変更します。詳細は、「収集方法の変更」を参照してください。

9.1.1.1. 収集方法の変更

アップグレードするクラスターに **SecuredCluster** CR が含まれていて、4.1 以降からアップグレードする場合は、アップグレードする前にノードごとの収集設定が **CORE_BPF** に設定されていることを確認する必要があります。それ以外の場合は、収集方法を **EBPF** に設定します。収集方法を **EBPF** に設定するには、アップグレード後に **forceCollection** パラメーターを **true** に設定し、収集方法が **EBPF** であることを確認する必要があります。

手順

1. OpenShift Container Platform Web コンソールで、RHACS Operator ページに移動します。
2. 上部のナビゲーションメニューで **Secured Cluster** を選択します。
3. インスタンス名 (例: **stackrox-secured-cluster-services)** をクリックします。
4. 設定を変更するには、次のいずれかの方法を使用します。
   - **Form view** の **Per Node Settings** → **Collector Settings** → **Collection** で、**CORE_BPF** を選択します。
   - **YAML** をクリックして YAML エディターを開き、**spec.perNode.collector.collection** 属性を見つけます。値が **KernelModule** の場合は、**CORE_BPF** に変更します。

     注記

     EBPF は、4.1 より前のバージョンからアップグレードする場合、または EBPF を使用する特別な理由がある場合にのみ使用してください。
5. **Save** をクリックします。

関連情報

- インストール済み Operator の更新

9.1.2. セキュアクラスターの Operator アップグレードのロールバック

Operator のアップグレードをロールバックするには、CLI または OpenShift Container Platform Web コンソールのいずれかを使用できます。

注記

セキュアクラスターでは、Operator のアップグレードをロールバックする必要があるのは、セキュアクラスターに問題がある場合など、まれなケースのみです。

9.1.2.1. CLI を使用した Operator アップグレードのロールバック

CLI コマンドを使用して Operator バージョンをロールバックできます。

手順

1. 次のコマンドを実行して、OLM サブスクリプションを削除します。
   - OpenShift Container Platform の場合、以下のコマンドを実行します。

     ```
     $ oc -n rhacs-operator delete subscription rhacs-operator
     ```

   - Kubernetes の場合、次のコマンドを実行します。

     ```
     $ kubectl -n rhacs-operator delete subscription rhacs-operator
     ```

2. 次のコマンドを実行して、クラスターサービスバージョン (CSV) を削除します。
   - OpenShift Container Platform の場合、以下のコマンドを実行します。

     ```
     $ oc -n rhacs-operator delete csv -l operators.coreos.com/rhacs-operator.rhacs-operator
     ```

   - Kubernetes の場合、次のコマンドを実行します。

     ```
     $ kubectl -n rhacs-operator delete csv -l operators.coreos.com/rhacs-operator.rhacs-operator
     ```

3. ロールバックされたチャネルに最新バージョンの Operator をインストールします。

9.1.2.2. Web コンソールを使用した Operator アップグレードのロールバック

OpenShift Container Platform Web コンソールを使用して Operator バージョンをロールバックできます。

前提条件

- **cluster-admin** パーミッションを持つアカウントを使用して OpenShift Container Platform クラスター Web コンソールにアクセスできる。

手順

1. **Operators → Installed Operators** ページに移動します。
2. RHACS Operator をクリックします。
3. **Operator Details** ページで、**Actions** リストから **Uninstall Operator** を選択します。この操作を実行すると、Operator は実行を停止し、更新を受信しなくなります。
4. ロールバックされたチャネルに最新バージョンの Operator をインストールします。

関連情報

- Operator Lifecycle Manager ワークフロー
- 保留中の Operator 更新の手動による承認

9.1.3. Operator アップグレードに関する問題のトラブルシューティング

RHACS Operator のアップグレード関連の問題を調査して解決するには、次の手順に従ってください。

9.1.3.1. Central クラスターまたはセキュアクラスターのデプロイに失敗する

RHACS Operator が次の状態にある場合は、カスタムリソースの状態をチェックして問題を見つける必要があります。

- Operator がセキュアクラスターをデプロイできない場合
- Operator が CR の変更を実際のリソースに適用できない場合
- セキュアクラスターの場合は、次のコマンドを実行して状態を確認します。

```
$ oc -n rhacs-operator describe securedclusters.platform.stackrox.io
```
(1)

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

状態の出力から設定エラーを特定できます。

出力例

```
 Conditions:
   Last Transition Time:  2023-04-19T10:49:57Z
   Status:                False
   Type:                  Deployed
   Last Transition Time:  2023-04-19T10:49:57Z
   Status:                True
   Type:                  Initialized
   Last Transition Time:  2023-04-19T10:59:10Z
   Message:               Deployment.apps "central" is invalid:
spec.template.spec.containers[0].resources.requests: Invalid value: "50": must be less than or equal
to cpu limit
   Reason:                ReconcileError
   Status:                True
   Type:                  Irreconcilable
   Last Transition Time:  2023-04-19T10:49:57Z
   Message:               No proxy configuration is desired
```

```
    Reason:              NoProxyConfig
    Status:              False
    Type:                ProxyConfigFailed
    Last Transition Time:  2023-04-19T10:49:57Z
    Message:               Deployment.apps "central" is invalid:
spec.template.spec.containers[0].resources.requests: Invalid value: "50": must be less than or equal
to cpu limit
    Reason:              InstallError
    Status:              True
    Type:                ReleaseFailed
```

さらに、RHACS Pod のログを表示して、問題に関する詳細情報を見つけることができます。次のコマンドを実行してログを表示します。

```
oc -n rhacs-operator logs deploy/rhacs-operator-controller-manager manager
```
(1)

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

9.2. HELM チャートを使用した RHACS CLOUD SERVICE でのセキュアクラスターのアップグレード

Helm チャートを使用して、RHACS Cloud Service でセキュアクラスターをアップグレードできます。

Helm チャートを使用して RHACS セキュアクラスターをインストールした場合は、Helm チャートを更新し、**helm upgrade** コマンドを実行することで、RHACS の最新バージョンにアップグレードできます。

9.2.1. Helm チャートリポジトリーの更新

Red Hat Advanced Cluster Security for Kubernetes の新しいバージョンにアップグレードする前に、常に Helm チャートを更新する必要があります。

前提条件

- Red Hat Advanced Cluster Security for Kubernetes の Helm チャートリポジトリーをすでに追加している。
- Helm バージョン 3.8.3 以降を使用している。

手順

- Red Hat Advanced Cluster Security for Kubernetes チャートリポジトリーを追加します。

```
$ helm repo update
```

検証

- 次のコマンドを実行して、追加されたチャートリポジトリーを確認します。

```
$ helm search repo -l rhacs/
```

9.2.2. Helm アップグレードコマンドの実行

helm upgrade コマンドを使用して、Red Hat Advanced Cluster Security for Kubernetes (RHACS) を更新できます。

前提条件

- Red Hat Advanced Cluster Security for Kubernetes (RHACS) のインストールに使用した **values-private.yaml** 設定ファイルにアクセスできる。アクセスできない場合は、この手順のコマンドを続行する前に、ルート証明書を含む **values-private.yaml** 設定ファイルを生成する必要があります。

手順

- helm upgrade コマンドを実行し、**-f** オプションを使用して設定ファイルを指定します。

```
$ helm upgrade -n stackrox stackrox-secured-cluster-services \
  rhacs/secured-cluster-services --version <current-rhacs-version> \ 1
  -f values-private.yaml
```

1 YAML 設定ファイルのパスを指定するには、-f オプションを使用します。

9.2.3. 関連情報

- Helm チャートを使用したセキュアクラスターへの RHACS Cloud Service のインストール

9.3. ROXCTL CLI を使用した RHACS CLOUD SERVICE でのセキュアクラスターの手動アップグレード

roxctl CLI を使用して、RHACS Cloud Service でセキュアクラスターをアップグレードできます。

重要

roxctl CLI を使用してセキュアクラスターをインストールした場合にのみ、セキュアクラスターを手動でアップグレードする必要があります。

9.3.1. roxctl CLI のアップグレード

roxctl CLI を最新バージョンにアップグレードするには、現在のバージョンの **roxctl** CLI をアンインストールしてから、最新バージョンの **roxctl** CLI をインストールする必要があります。

9.3.1.1. roxctl CLI のアンインストール

次の手順を使用して、Linux 上の **roxctl** CLI バイナリーをアンインストールできます。

手順

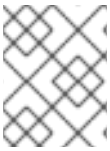
- **roxctl** バイナリーを見つけて削除します。

```
$ ROXPATH=$(which roxctl) && rm -f $ROXPATH 1
```

1 環境によっては、**roxctl** バイナリーを削除するために管理者権限が必要になる場合があります。

9.3.1.2. Linux への roxctl CLI のインストール

次の手順を使用して、Linux に **roxctl** CLI バイナリーをインストールできます。



注記

Linux 用の **roxctl** CLI は、**amd64**、**ppcl64le**、および **s390x** アーキテクチャーで使用できます。

手順

1. ターゲットのオペレーティングシステムの **roxctl** アーキテクチャーを確認します。

```
$ arch="$(uname -m | sed "s/x86_64//")"; arch="${arch:+-$arch}"
```

2. **roxctl** CLI をダウンロードします。

```
$ curl -L -f -o roxctl
"https://mirror.openshift.com/pub/rhacs/assets/4.4.4/bin/Linux/roxctl${arch}"
```

3. **roxctl** バイナリーを実行可能にします。

```
$ chmod +x roxctl
```

4. **PATH** 上にあるディレクトリーに **roxctl** バイナリーを配置します。
PATH を確認するには、以下のコマンドを実行します。

```
$ echo $PATH
```

検証

- インストールした **roxctl** のバージョンを確認します。

```
$ roxctl version
```

9.3.1.3. macOS への roxctl CLI のインストール

次の手順を使用して、**roxctl** CLI バイナリーを macOS にインストールできます。

注記

macOS 用の **roxctl** CLI は、**amd64** アーキテクチャーで利用できます。

手順

1. **roxctl** CLI をダウンロードします。

```
$ curl -L -f -o roxctl
"https://mirror.openshift.com/pub/rhacs/assets/4.4.4/bin/Darwin/roxctl${arch}"
```

2. バイナリーからすべての拡張属性を削除します。

```
$ xattr -c roxctl
```

3. **roxctl** バイナリーを実行可能にします。

```
$ chmod +x roxctl
```

4. **PATH** 上にあるディレクトリーに **roxctl** バイナリーを配置します。
PATH を確認するには、以下のコマンドを実行します。

```
$ echo $PATH
```

検証

- インストールした **roxctl** のバージョンを確認します。

```
$ roxctl version
```

9.3.1.4. Windows への roxctl CLI のインストール

次の手順を使用して、**roxctl** CLI バイナリーを Windows にインストールできます。

注記

Windows 用の **roxctl** CLI は、**amd64** アーキテクチャーで使用できます。

手順

- **roxctl** CLI をダウンロードします。

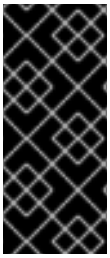
```
$ curl -f -O https://mirror.openshift.com/pub/rhacs/assets/4.4.4/bin/Windows/roxctl.exe
```

検証

- インストールした **roxctl** のバージョンを確認します。

```
$ roxctl version
```

9.3.2. すべてのセキュアクラスターの手動アップグレード



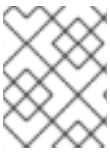
重要

最適な機能を実現するには、RHACS Cloud Service が実行しているバージョンと同じ RHACS バージョンをセキュアクラスターに使用してください。自動アップグレードを使用している場合は、自動アップグレードを使用してすべてのセキュアクラスターを更新します。自動アップグレードを使用していない場合は、すべてのセキュアクラスターでこのセクションの手順を完了してください。

Sensor、Collector、および Admission コントローラーを実行している各セキュアクラスターの手動アップグレードを完了するには、次の手順に従ってください。

9.3.2.1. その他のイメージの更新

自動アップグレードを使用しない場合は、各セキュアクラスターの Sensor、Collector、Compliance イメージを更新する必要があります。



注記

Kubernetes を使用している場合は、この手順にリストされているコマンドで **oc** の代わりに **kubectl** を使用してください。

手順

1. Sensor イメージを更新します。

```
$ oc -n stackrox set image deploy/sensor sensor=registry.redhat.io/advanced-cluster-security/rhacs-main-rhel8:4.4.4 (1)
```

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

2. Compliance イメージを更新します。

```
$ oc -n stackrox set image ds/collector compliance=registry.redhat.io/advanced-cluster-security/rhacs-main-rhel8:4.4.4 (1)
```

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

3. Collector イメージを更新します。

```
$ oc -n stackrox set image ds/collector collector=registry.redhat.io/advanced-cluster-security/rhacs-collector-rhel8:4.4.4 (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

コレクタースリムイメージを使用している場合は、代わりに次のコマンドを実行します。

```
$ oc -n stackrox set image ds/collector collector=registry.redhat.io/advanced-cluster-security/rhacs-collector-slim-rhel8:{rhacs-version}
```

4. アドミッションコントロールイメージを更新します。

```
$ oc -n stackrox set image deploy/admission-control admission-control=registry.redhat.io/advanced-cluster-security/rhacs-main-rhel8:4.4.4
```

重要

roxctl CLI を使用して Red Hat OpenShift に RHACS をインストールした場合は、Security Context Constraints (SCC) を移行する必要があります。

詳細は、「関連情報」セクションの「手動アップグレード中の SCC の移行」を参照してください。

関連情報

- roxctl CLI を使用した認証

9.3.2.2. 手動アップグレード中の SCC の移行

roxctl CLI を使用して手動アップグレード中に Security Context Constraints (SCC) を移行すると、Red Hat OpenShift SCC を使用するように Red Hat Advanced Cluster Security for Kubernetes (RHACS) サービスを移行して、Central クラスターとすべてのセキュアクラスター全体で互換性と最適なセキュリティー設定を確保できます。

手順

1. すべてのセキュアクラスターにデプロイされているすべての RHACS サービスをリスト表示します。

```
$ oc -n stackrox describe pods | grep 'openshift.io/scc\|^Name:'
```

出力例

```
Name:      admission-control-6f4dcc6b4c-2phwd
           openshift.io/scc: stackrox-admission-control
#...
Name:      central-575487bfcb-sjdx8
           openshift.io/scc: stackrox-central
Name:      central-db-7c7885bb-6bgbd
           openshift.io/scc: stackrox-central-db
Name:      collector-56nkr
           openshift.io/scc: stackrox-collector
#...
Name:      scanner-68fc55b599-f2wm6
           openshift.io/scc: stackrox-scanner
```

```
Name:      scanner-68fc55b599-fztlh
#...
Name:      sensor-84545f86b7-xgdwf
           openshift.io/scc: stackrox-sensor
#...
```

この例では、各 Pod に独自のカスタム SCC があり、**openshift.io/scc** フィールドで指定されていることがわかります。

2. RHACS のカスタム SCC の代わりに Red Hat OpenShift SCC を使用するには、必要なロールとロールバインディングを追加します。

3. すべてのセキュアクラスターで Red Hat OpenShift SCC を使用するために必要なロールとロールバインディングを追加するには、次の手順を実行します。

   a. 次の内容を使用して、ロールリソースとロールバインディングリソースを定義する **upgrade-scs.yaml** という名前のファイルを作成します。

例9.1 サンプル YAML ファイル

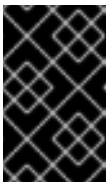
```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role (1)
metadata:
  annotations:
    email: support@stackrox.com
    owner: stackrox
  labels:
    app.kubernetes.io/component: collector
    app.kubernetes.io/instance: stackrox-secured-cluster-services
    app.kubernetes.io/name: stackrox
    app.kubernetes.io/part-of: stackrox-secured-cluster-services
    app.kubernetes.io/version: 4.4.0
    auto-upgrade.stackrox.io/component: sensor
  name: use-privileged-scc (2)
  namespace: stackrox (3)
rules: (4)
- apiGroups:
  - security.openshift.io
  resourceNames:
  - privileged
  resources:
  - securitycontextconstraints
  verbs:
  - use
- - -
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding (5)
metadata:
  annotations:
    email: support@stackrox.com
    owner: stackrox
  labels:
    app.kubernetes.io/component: collector
    app.kubernetes.io/instance: stackrox-secured-cluster-services
    app.kubernetes.io/name: stackrox
    app.kubernetes.io/part-of: stackrox-secured-cluster-services
```

```
    app.kubernetes.io/version: 4.4.0
    auto-upgrade.stackrox.io/component: sensor
  name: collector-use-scc 6
  namespace: stackrox
roleRef: 7
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: use-privileged-scc
subjects: 8
- kind: ServiceAccount
  name: collector
  namespace: stackrox
- - -
```

1. Kubernetes リソースのタイプ。この例では **Role** です。
2. ロールリソースの名前。
3. ロールの作成先の namespace。
4. ロールリソースによって付与される権限を記述します。
5. Kubernetes リソースのタイプ。この例では **RoleBinding** です。
6. ロールバインディングリソースの名前。
7. 同じ namespace 内のバインドするロールを指定します。
8. ロールにバインドするサブジェクトを指定します。

b. 次のコマンドを実行して、**upgrade-scs.yaml** ファイルで指定したロールリソースとロールバインディングリソースを作成します。

```
$ oc -n stackrox create -f ./update-scs.yaml
```



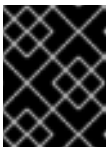
重要

upgrade-scs.yaml ファイルで指定したロールとロールバインディングを作成するには、各セキュアクラスターでこのコマンドを実行する必要があります。

4. RHACS に固有の SCC を削除します。

a. すべてのセキュアクラスターに固有の SCC を削除するには、次のコマンドを実行します。

```
$ oc delete scc/stackrox-admission-control scc/stackrox-collector scc/stackrox-sensor
```



重要

各セキュアクラスターに固有の SCC を削除するには、各セキュアクラスターでこのコマンドを実行する必要があります。

検証

- 次のコマンドを実行して、すべての Pod が正しい SCC を使用していることを確認します。

```
$ oc -n stackrox describe pods | grep 'openshift.io/scc\|^Name:'
```

出力を次の表と比較してください。

コンポーネント	以前のカスタム SCC	Red Hat OpenShift 4 の新しい SCC
Central	**stackrox-central**	**nonroot-v2**
Central-db	**stackrox-central-db**	**nonroot-v2**
Scanner	**stackrox-scanner**	**nonroot-v2**
Scanner-db	**stackrox-scanner**	**nonroot-v2**
Admission Controller	**stackrox-admission-control**	**restricted-v2**
Collector	**stackrox-collector**	**privileged**
Sensor	**stackrox-sensor**	**restricted-v2**

9.3.2.2.1. Sensor デプロイメントの GOMEMLIMIT 環境変数の編集

バージョン 4.4 にアップグレードするには、**GOMEMLIMIT** 環境変数を **ROX_MEMLIMIT** 環境変数に手動で置き換える必要があります。この変数はデプロイメントごとに編集する必要があります。

手順

1. 次のコマンドを実行して、Sensor デプロイメントの変数を編集します。

```
$ oc -n stackrox edit deploy/sensor (1)
```

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

2. **GOMEMLIMIT** 変数を **ROX_MEMLIMIT** に置き換えます。
3. ファイルを保存します。

9.3.2.2.2. Collector デプロイメントの GOMEMLIMIT 環境変数の編集

バージョン 4.4 にアップグレードするには、**GOMEMLIMIT** 環境変数を **ROX_MEMLIMIT** 環境変数に手動で置き換える必要があります。この変数はデプロイメントごとに編集する必要があります。

手順

1. Collector デプロイメントの変数を編集するには、次のコマンドを実行します。

```
$ oc -n stackrox edit deploy/collector 1
```

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

2. **GOMEMLIMIT** 変数を **ROX_MEMLIMIT** に置き換えます。
3. ファイルを保存します。

9.3.2.2.3. Admission Controller デプロイメントの GOMEMLIMIT 環境変数の編集

バージョン 4.4 にアップグレードするには、**GOMEMLIMIT** 環境変数を **ROX_MEMLIMIT** 環境変数に手動で置き換える必要があります。この変数はデプロイメントごとに編集する必要があります。

手順

1. 次のコマンドを実行して、Admission Controller デプロイメントの変数を編集します。

```
$ oc -n stackrox edit deploy/admission-control 1
```

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

2. **GOMEMLIMIT** 変数を **ROX_MEMLIMIT** に置き換えます。
3. ファイルを保存します。

9.3.2.2.4. セキュアクラスターのアップグレードの確認

セキュアクラスターをアップグレードしたら、更新された Pod が機能していることを確認します。

手順

- 新しい Pod がデプロイされていることを確認します。

```
$ oc get deploy,ds -n stackrox -o wide 1
```

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

```
$ oc get pod -n stackrox --watch 1
```

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

9.3.3. RHCOS ノードスキャンの有効化

OpenShift Container Platform を使用する場合は、Red Hat Advanced Cluster Security for Kubernetes (RHACS) を使用して、Red Hat Enterprise Linux CoreOS (RHCOS) ノードの脆弱性スキャンを有効にできます。

前提条件

- Secured クラスターの RHCOS ノードホストをスキャンするには、OpenShift Container Platform 4.11 以降に Secured クラスターをインストールしておく必要があります。サポートされるプラットフォームおよびアーキテクチャーの詳細は、Red Hat Advanced Cluster Security for Kubernetes Support Matrix を参照してください。RHACS のライフサイクルのサポート情報は、Red Hat Advanced Cluster Security for Kubernetes サポートポリシー を参照してください。

手順

1. 次のコマンドのいずれかを実行して、コンプライアンスコンテナーを更新します。
   - メトリクスが無効になっているデフォルトのコンプライアンスコンテナーの場合は、次のコマンドを実行します。

```
$ oc -n stackrox patch daemonset/collector -p '{"spec":{"template":{"spec":{"containers":
[{"name":"compliance","env":[{"name":"ROX_METRICS_PORT","value":"disabled"},
{"name":"ROX_NODE_SCANNING_ENDPOINT","value":"127.0.0.1:8444"},
{"name":"ROX_NODE_SCANNING_INTERVAL","value":"4h"},
{"name":"ROX_NODE_SCANNING_INTERVAL_DEVIATION","value":"24m"},
{"name":"ROX_NODE_SCANNING_MAX_INITIAL_WAIT","value":"5m"},
{"name":"ROX_RHCOS_NODE_SCANNING","value":"true"},
{"name":"ROX_CALL_NODE_INVENTORY_ENABLED","value":"true"}]}]}}}}'
```

   - Prometheus メトリクスが有効になっているコンプライアンスコンテナーの場合は、次のコマンドを実行します。

```
$ oc -n stackrox patch daemonset/collector -p '{"spec":{"template":{"spec":{"containers":
[{"name":"compliance","env":[{"name":"ROX_METRICS_PORT","value":":9091"},
{"name":"ROX_NODE_SCANNING_ENDPOINT","value":"127.0.0.1:8444"},
{"name":"ROX_NODE_SCANNING_INTERVAL","value":"4h"},
{"name":"ROX_NODE_SCANNING_INTERVAL_DEVIATION","value":"24m"},
{"name":"ROX_NODE_SCANNING_MAX_INITIAL_WAIT","value":"5m"},
{"name":"ROX_RHCOS_NODE_SCANNING","value":"true"},
{"name":"ROX_CALL_NODE_INVENTORY_ENABLED","value":"true"}]}]}}}}'
```

2. 次の手順を実行して、Collector DaemonSet (DS) を更新します。
   a. 次のコマンドを実行して、新しいボリュームマウントを Collector DS に追加します。

```
$ oc -n stackrox patch daemonset/collector -p '{"spec":{"template":{"spec":{"volumes":
[{"name":"tmp-volume","emptyDir":{}},{"name":"cache-volume","emptyDir":
{"sizeLimit":"200Mi"}}]}}}}'
```

   b. 次のコマンドを実行して、新しい **NodeScanner** コンテナーを追加します。

```
$ oc -n stackrox patch daemonset/collector -p '{"spec":{"template":{"spec":{"containers":
[{"command":["/scanner","--nodeinventory","--config=",""],"env":
[{"name":"ROX_NODE_NAME","valueFrom":{"fieldRef":
{"apiVersion":"v1","fieldPath":"spec.nodeName"}}},
{"name":"ROX_CLAIR_V4_SCANNING","value":"true"},
{"name":"ROX_COMPLIANCE_OPERATOR_INTEGRATION","value":"true"},
{"name":"ROX_CSV_EXPORT","value":"false"},
{"name":"ROX_DECLARATIVE_CONFIGURATION","value":"false"},
```

{"name":"ROX_INTEGRATIONS_AS_CONFIG","value":"false"},
{"name":"ROX_NETPOL_FIELDS","value":"true"},
{"name":"ROX_NETWORK_DETECTION_BASELINE_SIMULATION","value":"true"},
{"name":"ROX_NETWORK_GRAPH_PATTERNFLY","value":"true"},
{"name":"ROX_NODE_SCANNING_CACHE_TIME","value":"3h36m"},
{"name":"ROX_NODE_SCANNING_INITIAL_BACKOFF","value":"30s"},
{"name":"ROX_NODE_SCANNING_MAX_BACKOFF","value":"5m"},
{"name":"ROX_PROCESSES_LISTENING_ON_PORT","value":"false"},
{"name":"ROX_QUAY_ROBOT_ACCOUNTS","value":"true"},
{"name":"ROX_ROXCTL_NETPOL_GENERATE","value":"true"},
{"name":"ROX_SOURCED_AUTOGENERATED_INTEGRATIONS","value":"false"},
{"name":"ROX_SYSLOG_EXTRA_FIELDS","value":"true"},
{"name":"ROX_SYSTEM_HEALTH_PF","value":"false"},
{"name":"ROX_VULN_MGMT_WORKLOAD_CVES","value":"false"}],"image":"registry.redhat.io/advanced-cluster-security/rhacs-scanner-slim-rhel8:4.4.4","imagePullPolicy":"IfNotPresent","name":"node-inventory","ports":[{"containerPort":8444,"name":"grpc","protocol":"TCP"}],"volumeMounts":[{"mountPath":"/host","name":"host-root-ro","readOnly":true},
{"mountPath":"/tmp/","name":"tmp-volume"},{"mountPath":"/cache","name":"cache-volume"}]}]}}}}'

関連情報

- RHCOS ノードホストのスキャン