Red Hat Advanced Cluster Security for Kubernetes 3.70

設定

Red Hat Advanced Cluster Security for Kubernetes の設定

Last Updated: 2022-12-11

Red Hat Advanced Cluster Security for Kubernetes 3.70 設定

Red Hat Advanced Cluster Security for Kubernetes の設定

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

法律上の通知

概要

本書では、証明書の設定、自動アップグレード、プロキシー設定など、一般的な設定タスクを実行する方法を説明します。また、モニタリングおよびロギングの有効化に関する情報も含まれます。

目次

第1章 カスタム証明書の追加

Red Hat Advanced Cluster Security for Kubernetes でカスタム TLS 証明書を使用する方法を学びます。証明書を設定した後、ユーザーと API クライアントは、セントラルに接続するときに証明書のセキュリティー警告をバイパスする必要はありません。

1.1. カスタムセキュリティー証明書の追加

インストール中、または既存の Red Hat Advanced Cluster Security for Kubernetes デプロイメントにセキュリティー証明書を適用できます。

1.1.1. カスタム証明書を追加するための前提条件

前提条件

- PEM でエンコードされた秘密鍵と証明書ファイルがすでに存在する必要がある。
- 証明書ファイルは、人間が読める形式のブロックで開始および終了する必要がある。以下に例を示します。

```
-----BEGIN CERTIFICATE-----
MIICLDCCAdKgAwIBAgIBADAKBggqhkjOPQQDAjB9MQswCQYDVQQGEwJCRTEPMA0G
...
l4wOuDwKQa+upc8GftXE2C//4mKANBC6It01gUaTIpo=
-----END CERTIFICATE-----
```

- 証明書ファイルには、単一の (リーフ) 証明書または証明書チェーンのいずれかを含めることができる。

> **警告**
>
> - 証明書が信頼されたルートによって直接署名されていない場合は、中間証明書を含む完全な証明書チェーンを提供する必要がある。
> - チェーン内のすべての証明書は、リーフ証明書がチェーンの最初でルート証明書がチェーンの最後になるように順序付けられている必要がある。

- グローバルに信頼されていないカスタム証明書を使用している場合は、カスタム証明書を信頼するようにセンサーを設定する必要もある。

1.1.2. 新規インストール中のカスタム証明書の追加

手順

- Helm を使用して Red Hat Advanced Cluster Security for Kubernetes をインストールする場合:
  1. カスタム証明書とそのキーを **values-private.yaml** ファイルに追加します。

```
central:
  # Configure a default TLS certificate (public cert + private key) for central
  defaultTLS:
    cert: |
      -----BEGIN CERTIFICATE-----

EXAMPLE!MIIMIICLDCCAdKgAwIBAgIBADAKBggqhkjOPQQDAjB9MQswCQYDVQQGE
wJCRTEPMA0G
      ...
      -----END CERTIFICATE-----
    key: |
      -----BEGIN EC PRIVATE KEY-----
      EXAMPLE!MHcl4wOuDwKQa+upc8GftXE2C//4mKANBC6It01gUaTIpo=
      ...
      -----END EC PRIVATE KEY-----
```

2. インストール中に設定ファイルを提供します。

```
$ helm install -n stackrox --create-namespace stackrox-central-services rhacs/central-
services -f values-private.yaml
```

- **roxctl** CLI を使用して Red Hat Advanced Cluster Security for Kubernetes をインストールする場合は、インストーラーの実行時に証明書とキーファイルを提供します。
  - 非対話型インストーラーの場合は、**--default-tls-cert** および **--default-tls-key** オプションを使用します。

```
$ roxctl central generate --default-tls-cert "cert.pem" --default-tls-key "key.pem"
```

  - 対話型インストーラーの場合、プロンプトの回答を入力するときに証明書とキーファイルを提供します。

```
...
Enter PEM cert bundle file (optional): <cert.pem>
Enter PEM private key file (optional): <key.pem>
Enter administrator password (default: autogenerated):
Enter orchestrator (k8s, openshift): openshift
...
```

1.1.3. 既存のインスタンスのカスタム証明書の追加

手順

- Helm を使用して Red Hat Advanced Cluster Security for Kubernetes をインストールした場合:
  1. カスタム証明書とそのキーを **values-private.yaml** ファイルに追加します。

```
central:
  # Configure a default TLS certificate (public cert + private key) for central
  defaultTLS:
    cert: |
      -----BEGIN CERTIFICATE-----

EXAMPLE!MIIMIICLDCCAdKgAwIBAgIBADAKBggqhkjOPQQDAjB9MQswCQYDVQQGE
```

```
wJCRTEPMA0G
    ...
    -----END CERTIFICATE-----
  key: |
    -----BEGIN EC PRIVATE KEY-----
    EXAMPLE!MHcl4wOuDwKQa+upc8GftXE2C//4mKANBC6It01gUaTIpo=
    ...
    -----END EC PRIVATE KEY-----
```

2. **helm upgrade** コマンドを使用して、更新された設定ファイルを提供します。

```
$ helm upgrade -n stackrox --create-namespace stackrox-central-services rhacs/central-services -f values-private.yaml
```

- **roxctl** CLI を使用して Red Hat Advanced Cluster Security for Kubernetes をインストールした場合。
  - PEM でエンコードされたキーと証明書ファイルから TLS シークレットを作成して適用します。

```
$ oc -n stackrox create secret tls central-default-tls-cert \
  --cert <server_cert.pem> \
  --key <server_key.pem> \
  --dry-run -o yaml | oc apply -f -
```

このコマンドを実行すると、セントラルは Pod を再起動しなくても、新しいキーと証明書を自動的に適用します。変更が反映されるまでに最大 1 分かかる場合があります。

1.1.4. 既存のインスタンスのカスタム証明書の更新

セントラルのカスタム証明書を使用する場合は、次の手順を実行して証明書を更新できます。

手順

1. 既存のカスタム証明書のシークレットを削除します。

```
$ oc delete secret central-default-tls-cert
```

2. 新規シークレットを作成します。

```
$ oc -n stackrox create secret tls central-default-tls-cert \
  --cert <server_cert.pem> \
  --key <server_key.pem> \
  --dry-run -o yaml | oc apply -f -
```

3. セントラルコンテナーを再起動します。

1.1.4.1. セントラルコンテナーの再起動

セントラルコンテナーを強制終了するか、セントラル Pod を削除することで、セントラルコンテナーを再起動できます。

手順

- 次のコマンドを実行して、セントラルコンテナーを強制終了します。

注記

OpenShift Container Platform が変更を伝播し、セントラルコンテナーを再始動するまで、少なくとも 1 分間待機する必要があります。

```
$ oc -n stackrox exec deploy/central -c central -- kill 1
```

- または、次のコマンドを実行して セントラル Pod を削除します。

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

1.2. カスタム証明書を信頼するようにセンサーを設定する

グローバルに信頼されていないカスタム証明書を使用している場合は、カスタム証明書を信頼するようにセンサーを設定する必要があります。そうしないと、エラーが発生する可能性があります。特定のタイプのエラーは、設定と使用する証明書によって異なる場合があります。通常、これは **x509 validation** 関連のエラーです。

注記

グローバルに信頼できる証明書を使用している場合は、カスタム証明書を信頼するようにセンサーを設定する必要はありません。

1.2.1. センサーバーンドルのダウンロード

センサーバーンドルには、センサーをインストールするために必要な設定ファイルとスクリプトが含まれています。センサーバーンドルは RHACS ポータルからダウンロードできます。

手順

1. RHACS ポータルに移動します。
2. **Platform Configuration → Clusters** に移動します。
3. **New Cluster** をクリックして、クラスターの名前を指定します。
4. 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。そうでない場合は、別のクラスターにデプロイする場合、アドレス **central.stackrox.svc:443** を、インストールを予定している別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、またはその他のアドレス (ポート番号を含む) に置き換えます。

注記

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:

1. アドレスの前に **wss://** を付けます。そして、
2. 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** をクリックして先に進みます。
6. **Download YAML File and Keys** クリックします。

1.2.2. 新しいセンサーをデプロイするときにカスタム証明書を信頼するようにセンサーを設定する

前提条件

- センサーバーンドルをダウンロードした。

手順

- **sensor.sh** スクリプトを使用している場合:
  1. センサーバーンドルを unzip します。

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

  2. **sensor.sh** スクリプトを実行します。

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

     センサー (**./sensor/sensor.sh**) スクリプトを実行すると、証明書が自動的に適用されます。また、**sensor.sh** スクリプトを実行する前に、**sensor/additional-cas/** ディレクトリーに追加のカスタム証明書を配置することもできます。

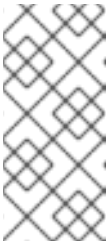
- **sensor.sh** スクリプトを使用していない場合:
  1. センサーバーンドルを unzip します。

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

  2. 以下のコマンドを実行してシークレットを作成します。

     ```
     $ ./sensor/ca-setup-sensor.sh -d sensor/additional-cas/ (1)
     ```

     (1) **-d** オプションを使用して、カスタム証明書を含むディレクトリーを指定します。



注記

secret already exists というエラーメッセージが表示された場合は、**-u** オプションを指定してスクリプトを再実行します。

```
$ ./sensor/ca-setup-sensor.sh -d sensor/additional-cas/ -u
```

3. YAML ファイルを使用してセンサーのデプロイを続行します。

1.2.3. カスタム証明書を信頼するように既存のセンサーを設定する

前提条件

- センサーバーンドルをダウンロードした。

手順

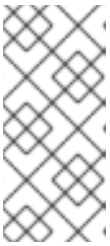
1. センサーバーンドルを unzip します。

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

2. 以下のコマンドを実行してシークレットを作成します。

```
$ ./sensor/ca-setup-sensor.sh -d sensor/additional-cas/ 1
```

1 **-d** オプションを使用して、カスタム証明書を含むディレクトリーを指定します。



注記

secret already exists というエラーメッセージが表示された場合は、**-u** オプションを指定してスクリプトを再実行します。

```
$ ./sensor/ca-setup-sensor.sh -d sensor/additional-cas/ -u
```

3. YAML ファイルを使用してセンサーのデプロイを続行します。

既存のセンサーに証明書を追加した場合は、センサーコンテナーを再起動する必要があります。

1.2.3.1. センサーコンテナーの再起動

コンテナーを強制終了するか、センサー Pod を削除することで、センサーコンテナーを再起動できます。

手順

- 次のコマンドを実行して、センサーコンテナーを強制終了します。

注記

OpenShift Container Platform または Kubernetes が変更を伝播し、センサーコンテナーを再起動するまで、少なくとも 1 分間待機する必要があります。

- OpenShift Container Platform

```
$ oc -n stackrox deploy/sensor -c sensor -- kill 1
```

- Kubernetes の場合:

```
$ kubectl -n stackrox deploy/sensor -c sensor -- kill 1
```

- または、次のコマンドを実行してセンサー Pod を削除します。
  - OpenShift Container Platform

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

  - Kubernetes の場合:

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

第2章 信頼できる認証局の追加

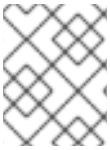
カスタム信頼済み証明書を Red Hat Cluster Security for Kubernetes に追加する方法を学びます。

ネットワークでエンタープライズ認証局 (CA) または自己署名証明書を使用している場合は、CA のルート証明書を信頼されたルート CA として Red Hat Advanced Cluster Security for Kubernetes に追加する必要があります。

信頼できるルート CA を追加すると、次のことが可能になります。

- セントラルと Scanner は、他のツールと統合するときにリモートサーバーを信頼します。
- セントラルに使用するカスタム証明書を信頼するセンサー。

インストール中または既存のデプロイメントに CA を追加できます。



注記

まず、セントラルをデプロイしたクラスターで信頼できる CA を設定してから、変更をスキャナーとセンサーに伝達する必要があります。

2.1. 追加の CA の設定

カスタム CA を追加するには:

手順

1. **ca-setup.sh** スクリプトをダウンロードします。

注記

- 新規インストールを行う場合は、**ca-setup.sh** スクリプトが **central-bundle/central/scripts/ca-setup.sh** の **scripts** ディレクトリーにあります。
- OpenShift Container Platform クラスターにログインしたのと同じターミナルで **ca-setup.sh** スクリプトを実行する必要があります。

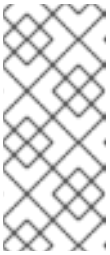
2. **ca-setup.sh** スクリプトを実行可能にします。

```
$ chmod +x ca-setup.sh
```

3. 以下を追加します:

   a. 単一の証明書。**-f** (ファイル) オプションを使用します。

```
$ ./ca-setup.sh -f <certificate>
```

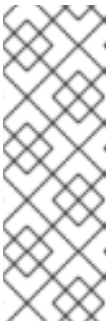


注記

- PEM でエンコードされた証明書ファイル (拡張子は任意) を使用する必要があります。
- **-u** (更新) オプションを **-f** オプションと一緒に使用して、以前に追加された証明書を更新することもできます。

b. 一度に複数の証明書を作成し、ディレクトリー内のすべての証明書を移動してから、**-d** (ディレクトリー) オプションを使用します。

```
$ ./ca-setup.sh -d <directory_name>
```



注記

- エクステンションが **.crt** の PEM エンコード証明書ファイルを使用する必要があります。
- 各ファイルには、1 つの証明書のみが含まれている必要があります。
- **-u** (更新) オプションを **-d** オプションと一緒に使用して、以前に追加された証明書を更新することもできます。

2.2. 変更の伝播

信頼できる CA を設定した後、Red Hat Advanced Cluster Security for Kubernetes サービスにそれらを信頼させる必要があります。

- インストール後に信頼できる CA を設定した場合は、セントラルを再起動する必要があります。
- さらに、イメージレジストリーと統合するための証明書も追加する場合は、セントラルとスキャナーの両方を再起動する必要があります。

2.2.1. セントラルコンテナーの再起動

セントラルコンテナーを強制終了するか、セントラル Pod を削除することで、セントラルコンテナーを再起動できます。

手順

- 次のコマンドを実行して、セントラルコンテナーを強制終了します。

注記

OpenShift Container Platform が変更を伝播し、セントラルコンテナーを再始動するまで、少なくとも 1 分間待機する必要があります。

```
$ oc -n stackrox exec deploy/central -c central -- kill 1
```

- または、次のコマンドを実行して セントラル Pod を削除します。

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

2.2.2. スキャナーコンテナーの再起動

Pod を削除すると、スキャナーコンテナーを再起動できます。

手順

- 次のコマンドを実行して、スキャナー Pod を削除します。
  - OpenShift Container Platform

```
$ oc delete pod -n stackrox -l app=scanner
```

  - Kubernetes の場合:

```
$ kubectl delete pod -n stackrox -l app=scanner
```

重要

信頼済み証明書を追加し、セントラルを設定すると、CA は、作成する新しいセンサーデプロイメントバンドルに含まれます。

- セントラルへの接続中に既存のセンサーが問題を報告した場合は、センサーデプロイメント YAML ファイルを生成し、既存のクラスターを更新する必要があります。
- **sensor.sh** スクリプトを使用して新しいセンサーをデプロイする場合、**sensor.sh** スクリプトを実行する前に、以下のコマンドを実行してください。

```
$ ./ca-setup-sensor.sh -d ./additional-cas/
```

- Helm を使用して新しいセンサーをデプロイする場合は、追加のスクリプトを実行する必要はありません。

第3章 内部証明書の再発行

Red Hat Advanced Cluster Security for Kubernetes の各コンポーネントは、X.509 証明書を使用して他のコンポーネントに対して自身を認証します。これらの証明書には有効期限があり、有効期限が切れる前に再発行する必要があります。証明書の有効期限は、RHACS ポータルの **Platform Configuration → Clusters** ビューで確認できます。

3.1. セントラルの内部証明書の再発行

セントラルは、他の Red Hat Advanced Cluster Security for Kubernetes サービスと通信するときに、ビルトインのサーバー証明書を認証に使用します。この証明書は、セントラルインストールに固有のものです。セントラル証明書の有効期限が近づくと、RHACS ポータルに情報バナーが表示されます。

注記

情報バナーは、証明書の有効期限の 15 日前にのみ表示されます。

前提条件

- 証明書を再発行するには、**ServiceIdentity** リソースの **write** 権限が必要である。

手順

1. バナーのリンクをクリックして、証明書とキー値を含む新しい OpenShift Container Platform シークレットを含む YAML 設定ファイルをダウンロードします。
2. セントラルをインストールしたクラスターに新しい YAML 設定ファイルを適用します。

```
$ oc apply -f <secret_file.yaml>
```

3. セントラルを再起動して、変更を適用します。

3.1.1. セントラルコンテナーの再起動

セントラルコンテナーを強制終了するか、セントラル Pod を削除することで、セントラルコンテナーを再起動できます。

手順

- 次のコマンドを実行して、セントラルコンテナーを強制終了します。

注記

OpenShift Container Platform が変更を伝播し、セントラルコンテナーを再始動するまで、少なくとも 1 分間待機する必要があります。

```
$ oc -n stackrox exec deploy/central -c central -- kill 1
```

- または、次のコマンドを実行して セントラル Pod を削除します。

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

3.2. スキャナーの内部証明書の再発行

スキャナーには、セントラルとの通信に使用する証明書が組み込まれています。

スキャナー証明書の有効期限が近づくと、RHACS ポータルに情報バナーが表示されます。

注記

情報バナーは、証明書の有効期限の 15 日前にのみ表示されます。

前提条件

- 証明書を再発行するには、**ServiceIdentity** リソースの **write** 権限が必要である。

手順

1. バナーのリンクをクリックして、証明書とキー値を含む新しい OpenShift Container Platform シークレットを含む YAML 設定ファイルをダウンロードします。
2. スキャナーをインストールしたクラスターに新しい YAML 設定ファイルを適用します。

```
$ oc apply -f <secret_file.yaml>
```

3. スキャナーを再起動して変更を適用します。

3.2.1. スキャナーおよびスキャナー DB コンテナーの再起動

Pod を削除することで、スキャナーとスキャナー DB コンテナーを再起動できます。

手順

- スキャナーおよびスキャナー DB Pod を削除するには、次のコマンドを実行します。
  - OpenShift Container Platform

```
$ oc delete pod -n stackrox -l app=scanner; oc -n stackrox delete pod -l app=scanner-db
```

  - Kubernetes の場合:

```
$ kubectl delete pod -n stackrox -l app=scanner; kubectl -n stackrox delete pod -l
app=scanner-db
```

3.3. センサー、コレクター、およびアドミッションコントローラーの内部証明書の再発行

センサー、コレクター、およびアドミッションコントローラーは、証明書を使用して相互に通信し、セントラルと通信します。

証明書を置き換えるには、以下のいずれかの方法を使用します。

- セキュアなクラスターで init バンドルを作成し、ダウンロードしてインストールします。

- 自動アップグレード機能を使用します。自動アップグレードは、**roxctl** CLI を使用する静的マニフェストのデプロイメントでのみ利用できます。

3.3.1. init バンドルを使用したセキュリティー保護されたクラスターの内部証明書の再発行

セキュリティー保護されたクラスターには、Collector、Sensor、および Admission Control コンポーネントが含まれます。これらのコンポーネントは、他の Red Hat Advanced Cluster Security for Kubernetes コンポーネントとの通信時に、認証に組み込みサーバー証明書を使用します。

セントラル証明書の有効期限が近づくと、RHACS ポータルに情報バナーが表示されます。

注記

情報バナーは、証明書の有効期限の 15 日前にのみ表示されます。

前提条件

- 証明書を再発行するには、**ServiceIdentity** リソースの **write** 権限が必要である。

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。複数のセキュリティー保護されたクラスターで同じバンドルを使用できます。

手順

- RHACS ポータルを使用して init バンドルを生成するには、以下を実行します。
  a. **Platform Configuration** → **Clusters** を選択します。
  b. **Manage Tokens** をクリックします。
  c. **Authentication Tokens** セクションに移動し、**Cluster Init Bundle** をクリックします。
  d. **Generate bundle** をクリックします。
  e. クラスター初期化バンドルの名前を入力し、**Generate** をクリックします。
  f. 生成されたバンドルをダウンロードするには、**Download Kubernetes secrets file** をクリックします。
- **roxctl** CLI を使用して init バンドルを生成するには、以下のコマンドを実行します。

```
$ roxctl -e <endpoint> -p <admin_password> central init-bundle generate <bundle_name> --output-secrets init-bundle.yaml
```

次のステップ

- セキュリティー保護されたクラスターごとに必要なリソースを作成するには、以下のコマンドを実行します。

```
$ oc -n stackrox apply -f <init-bundle.yaml>
```

関連情報

- init バンドルを使用したリソースの作成

3.3.2. 自動アップグレードを使用して、セキュリティー保護されたクラスターの内部証明書を再発行する

自動アップグレードを使用して、センサー、コレクター、およびアドミッションコントローラーの内部証明書を再発行できます。

注記

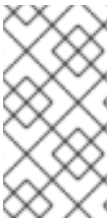
自動アップグレードは、**roxctl** CLI を使用する静的マニフェストベースのデプロイメントにのみ適用されます。**インストール** の章の roxctl CLI を使用したインストールの項の Central のインストールを参照してください。

前提条件

- すべてのクラスターに対して自動アップグレードを有効にしておく必要がある。
- 証明書を再発行するには、**ServiceIdentity** リソースの **write** 権限が必要である。

手順

1. RHACS ポータルで、**Platform Configuration** → **Clusters** に移動します。
2. **Clusters** ビューで、**Cluster** を選択して詳細を表示します。
3. クラスターの詳細パネルから、**自動アップグレードを使用して認証情報を適用する** リンクを選択します。



注記

自動アップグレードを適用すると、Red Hat Advanced Cluster Security for Kubernetes は選択されたクラスターに新しい資格情報を作成します。ただし、通知は引き続き表示されます。サービスの再起動後、各 Red Hat Advanced Cluster Security for Kubernetes サービスが新しい資格情報の使用を開始すると、通知は消えます。

関連情報

- Central のインストール

第4章 セキュリティー通知の追加

Red Hat Advanced Cluster Security for Kubernetes を使用すると、ユーザーがログインしたときに表示されるセキュリティー通知を追加できます。RHACS ポータルの上部または下部に組織全体のメッセージまたは免責事項を設定することもできます。

このメッセージは、企業ポリシーのリマンダーとして機能し、適切なポリシーを従業員に通知することができます。または、法的な理由でこれらのメッセージを表示して、アクションが監査されていることをユーザーに警告することもできます。

4.1. カスタムログインメッセージの追加

ログイン前の警告メッセージの表示は、悪意のあるユーザーまたは十分な情報を与えられていないユーザーに、アクションの結果について警告します。

前提条件

- ログインメッセージの設定オプションを表示するには、**read** 権限を持つ **Config** ロールが必要である。
- ログインメッセージを変更、有効化、または無効化するには、**write** 権限を持つ **Config** ロールが必要である。

手順

1. RHACS ポータルで、**Platform Configuration → System Configuration** に移動します。
2. **System Configuration** ビューのヘッダーで、**Edit** をクリックします。
3. **Login Configuration** セクションにログインメッセージを入力します。
4. ログインメッセージを有効にするには、**Login Configuration** セクションのトグルをオンにします。
5. **Save** をクリックします。

4.2. カスタムヘッダーとフッターの追加

カスタムテキストをヘッダーとフッターに配置し、テキストとその背景色を設定できます。

前提条件

- カスタムヘッダーとフッターの設定オプションを表示するには、**read** 権限を持つ **Config** ロールが必要です。
- カスタムヘッダーとフッターを変更、有効化、または無効化するには、**write** 権限を持つ **Config** ロールが必要です。

手順

1. RHACS ポータルで、**Platform Configuration → System Configuration** に移動します。
2. **System Configuration** ビューのヘッダーで、**Edit** をクリックします。

3. **Header Configuration** セクションと **Footer Configuration** セクションで、ヘッダーとフッターのテキストを入力します。
4. ヘッダーとフッターの **Text Color**、**Size**、**Background Color** をカスタマイズします。
5. ヘッダーを有効にするには、**Header Configuration** セクションでトグルをオンにします。
6. フッターを有効にするには、**Footer Configuration** セクションでトグルをオンにします。
7. **Save** をクリックします。

第5章 オフラインモードの有効化

オフラインモードを有効にすることで、インターネットに接続されていないクラスターに対して Red Hat Advanced Cluster Security for Kubernetes を使用できます。オフラインモードでは、Red Hat Advanced Cluster Security for Kubernetes コンポーネントはインターネット上のアドレスまたはホストに接続しません。

注記

Red Hat Advanced Cluster Security for Kubernetes は、ユーザーが指定したホスト名、IP アドレス、またはその他のリソースがインターネット上にあるかどうかを判断しません。たとえば、インターネット上でホストされている Docker レジストリーと統合しようとしても、Red Hat Advanced Cluster Security for Kubernetes はこのリクエストをブロックしません。

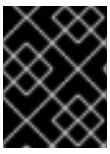
Red Hat Advanced Cluster Security for Kubernetes をオフラインモードでデプロイして操作するには:

1. RHACS イメージをダウンロードして、クラスターにインストールします。OpenShift Container Platform を使用している場合は、Operator Lifecycle Manager (OLM) および OperatorHub を使用して、インターネットに接続されているワークステーションにイメージをダウンロードできます。次に、ワークステーションは、セキュリティー保護されたクラスターにも接続されているミラーレジストリーにイメージをプッシュします。他のプラットフォームの場合は、Skopeo や Docker などのプログラムを使用して、リモートレジストリーからイメージをプルし、イメージの直接ダウンロード で説明されているように、このイメージを独自のプライベートレジストリーにプッシュできます。

2. インストール中にオフラインモードを有効にします。

3. (オプション) 新しい定義ファイルをアップロードして、スキャナーの脆弱性リストを定期的に更新します。

4. (オプション) 必要に応じて、新しいカーネルサポートパッケージをアップロードして、より多くのカーネルバージョンでランタイムコレクションのサポートを追加します。



重要

オフラインモードを有効にできるのはインストール中のみで、アップグレード中は有効にできません。

5.1. オフラインで使用するためのイメージのダウンロード

5.1.1. イメージを直接ダウンロードする

Red Hat Advanced Cluster Security for Kubernetes イメージを手動でプル、再タグ付け、およびレジストリーにプッシュできます。現在のバージョンのイメージバンドルに含まれているイメージは次のとおりです。

- **registry.redhat.io/advanced-cluster-security/rhacs-main-rhel8:3.70.2**
- **registry.redhat.io/advanced-cluster-security/rhacs-scanner-rhel8:3.70.2**
- **registry.redhat.io/advanced-cluster-security/rhacs-scanner-db-rhel8:3.70.2**
- **registry.redhat.io/advanced-cluster-security/rhacs-collector-rhel8:3.70.2**

- **registry.redhat.io/advanced-cluster-security/rhacs-collector-slim-rhel8:3.70.2**

5.1.1.1. イメージのタグの付け直し

Docker コマンドラインインターフェイスを使用して、イメージをダウンロードしてタグを付け直すことができます。

重要

イメージにタグを付け直すときは、イメージの名前とタグを維持する必要があります。たとえば、以下を使用します:

```
$ docker tag registry.redhat.io/advanced-cluster-security/rhacs-main-rhel8:3.70.2
<your_registry>/rhacs-main-rhel8:3.70.2
```

そして、次の例のようにタグを付け直さないでください。

```
$ docker tag registry.redhat.io/advanced-cluster-security/rhacs-main-rhel8:3.70.2
<your_registry>/other-name:latest
```

手順

1. レジストリーにログインします。

```
$ docker login registry.redhat.io
```

2. イメージをプルします:

```
$ docker pull <image>
```

3. イメージにタグを付け直します。

```
$ docker tag <image> <new_image>
```

4. 更新されたイメージをレジストリーにプッシュします。

```
$ docker push <new_image>
```

5.2. インストール中のオフラインモードの有効化

Red Hat Advanced Cluster Security for Kubernetes のインストール中にオフラインモードを有効にできます。

5.2.1. Helm 設定を使用したオフラインモードの有効化

Helm チャートを使用して、Kubernetes 用の Red Hat Advanced Cluster Security をインストールするときに、インストール中にオフラインモードを有効にできます。

手順

1. セントラルサービスの Helm チャートをインストールするときは、**values-public.yaml** 設定ファイルで **env.offlineMode** 環境変数の値を **true** に設定します。

2. secured-cluster-services Helm チャートをインストールするときは、**values-public.yaml** 設定ファイルで **config.offlineMode** パラメーターの値を **true** に設定します。

5.2.1.1. 関連情報

- central-services Helm チャートの設定
- secure-cluster-services Helm チャートの設定

5.2.2. roxctl CLI を使用したオフラインモードの有効化

roxctl CLI を使用して、Red Hat Advanced Cluster Security for Kubernetes をインストールするときにオフラインモードを有効にできます。

手順

1. インターネットに接続されたデフォルトのレジストリー (**registry.redhat.io**) 以外のレジストリーを使用している場合は、**image to use** プロンプトに応答するときに、Red Hat Advanced Cluster Security for Kubernetes イメージをプッシュした場所を指定します。

```
Enter main image to use (if unset, the default will be used): <your_registry>/rhacs-main-rhel8:3.70.2
```

注記

デフォルトのイメージは、プロンプト **Enter default container images settings:** に対する回答によって異なります。デフォルトのオプションの **rhacs** を入力した場合、デフォルトのイメージは **registry.redhat.io/advanced-cluster-security/rhacs-main-rhel8:3.70.2** になります。

```
Enter Scanner DB image to use (if unset, the default will be used): <your_registry>/rhacs-scanner-db-rhel8:3.70.2
```

```
Enter Scanner image to use (if unset, the default will be used): <your_registry>/rhacs-scanner-rhel8:3.70.2
```

2. オフラインモードを有効にするには、**Enter whether to run StackRox in offline mode** プロンプトに答えるときに **true** を入力します。

```
Enter whether to run StackRox in offline mode, which avoids reaching out to the internet (default: "false"): true
```

3. 後で、RHACS ポータルの **Platform Configuration → Clusters** ビューでセンサーをリモートクラスターに追加する場合は、**Collector Image Repository** フィールドにコレクターのイメージ名を指定する必要があります。

5.3. オフラインモードでのスキャナー定義の更新

スキャナーには、ローカルの脆弱性定義データベースが含まれています。Red Hat Advanced Cluster Security for Kubernetes が通常モード (インターネットに接続されている) で実行されている場合、スキャナーはインターネットから新しい脆弱性定義を取得し、そのデータベースを更新します。

ただし、Red Hat Advanced Cluster Security for Kubernetes をオフラインモードで使用している場合は、スキャナー定義をセントラルにアップロードして手動で更新する必要があります。

Red Hat Advanced Cluster Security for Kubernetes がオフラインモードで実行されている場合、スキャナーはセントラルからの新しい定義をチェックします。新しい定義が利用可能な場合、スキャナーはセントラルから新しい定義をダウンロードし、それらをデフォルトとしてマークしてから、更新された定義を使用してイメージをスキャンします。

オフラインモードで定義を更新するには:

1. 定義をダウンロードします。
2. 定義をセントラルにアップロードします。

5.3.1. スキャナー定義のダウンロード

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は、スキャナーが使用する脆弱性定義データベースをダウンロードしてからセントラルにアップロードできます。

前提条件

- スキャナー定義をダウンロードするには、インターネットにアクセスできるシステムが必要である。

手順

- https://install.stackrox.io/scanner/scanner-vuln-updates.zip に移動して、定義をダウンロードします。

5.3.2. セントラルへの定義のアップロード

スキャナー定義をセントラルにアップロードするには、API トークンまたは管理者パスワードのいずれかを使用できます。Red Hat は、各トークンに特定のアクセス制御権限が割り当てられているため、実稼働環境で認証トークンを使用することが推奨されます。

5.3.2.1. API トークンを使用してセントラルに定義をアップロードする

API トークンを使用して、スキャナーが使用する脆弱性定義データベースをセントラルにアップロードできます。

前提条件

- 管理者ロールを持つ API トークンがある。
- **roxctl** コマンドラインインターフェイス (CLI) をインストールしておく必要がある。

手順

1. **ROX_API_TOKEN** および **ROX_CENTRAL_ADDRESS** 環境変数を設定します。

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

-

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

2. 次のコマンドを実行して、定義ファイルをアップロードします。

```
$ roxctl scanner upload-db \
  -e "$ROX_CENTRAL_ADDRESS" \
  --scanner-db-file=<compressed_scanner_definitions.zip>
```

5.3.2.1.1. 関連情報

- **roxctl** CLI を使用した認証

5.3.2.2. 管理者パスワードを使用してセントラルに定義をアップロードする

Red Hat Advanced Cluster Security for Kubernetes 管理者パスワードを使用して、スキャナーが使用する脆弱性定義データベースをセントラルにアップロードできます。

前提条件

- 管理者パスワードが必要である。
- **roxctl** コマンドラインインターフェイス (CLI) をインストールしておく必要がある。

手順

1. **ROX_CENTRAL_ADDRESS** 環境変数を設定します。

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

2. 次のコマンドを実行して、定義ファイルをアップロードします。

```
$ roxctl scanner upload-db \
  -p <your_administrator_password> \
  -e "$ROX_CENTRAL_ADDRESS" \
  --scanner-db-file=<compressed_scanner_definitions.zip>
```

5.4. オフラインモードでのカーネルサポートパッケージの更新

コレクターは、セキュリティー保護されたクラスター内の各ノードの実行時アクティビティーを監視します。アクティビティーをモニターするには、コレクターにプローブが必要です。これらのプローブは、ホストにインストールされている Linux カーネルバージョンに固有のカーネルモジュールまたは eBPF プログラムです。コレクターイメージには、一連のビルトインプローブが含まれています。

Red Hat Advanced Cluster Security for Kubernetes が通常モード (インターネットに接続されている) で実行されている場合、必要なプローブがビルトインされていない場合、コレクターは新しいプローブを自動的にダウンロードします。

オフラインモードでは、最近サポートされたすべての Linux カーネルバージョンのプローブを含むパッケージを手動でダウンロードして、セントラルにアップロードできます。次に、コレクターはこれらのプローブをセントラルからダウンロードします。

コレクターは、次の順序で新しいプローブをチェックします。以下をチェックします:

1. 既存のコレクターイメージ。
2. カーネルサポートパッケージ (セントラルにアップロードした場合)。
3. インターネット上で利用可能な Red Hat 操作のサーバー。コレクターは、セントラルのネットワーク接続を使用して、プローブをチェックおよびダウンロードします。

コレクターがチェック後に新しいプローブを取得しない場合、**CrashLoopBackoff** イベントを報告します。

ネットワーク設定によってアウトバウンドトラフィックが制限されている場合は、最近サポートされたすべての Linux カーネルバージョンのプローブを含むパッケージを手動でダウンロードして、セントラルにアップロードできます。次に、コレクターはこれらのプローブをセントラルからダウンロードするため、インターネットへのアウトバウンドアクセスを回避できます。

5.4.1. カーネルサポートパッケージのダウンロード

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は、最近サポートされたすべての Linux カーネルバージョンのプローブを含むパッケージをダウンロードして、セントラルにアップロードできます。

手順

- https://install.stackrox.io/collector/support-packages/index.html から利用可能なサポートパッケージを表示およびダウンロードします。カーネルサポートパッケージリストは、Red Hat Advanced Cluster Security for Kubernetes バージョンに基づいてサポートパッケージを分類します。

5.4.2. カーネルサポートパッケージのセントラルへのアップロード

最近サポートされたすべての Linux カーネルバージョンのプローブを含むカーネルサポートパッケージをセントラルにアップロードできます。

前提条件

- 管理者ロールを持つ API トークンがある。
- **roxctl** コマンドラインインターフェイス (CLI) をインストールしておく必要がある。

手順

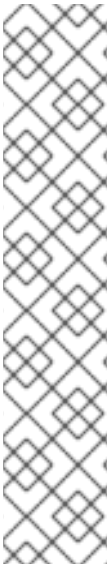
1. **ROX_API_TOKEN** および **ROX_CENTRAL_ADDRESS** 環境変数を設定します。

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

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

2. 次のコマンドを実行して、カーネルサポートパッケージをアップロードします。

```
$ roxctl collector support-packages upload <package_file> \
  -e "$ROX_CENTRAL_ADDRESS"
```



注記

- 以前にセントラルにアップロードされたコンテンツを含む新しいサポートパッケージをアップロードすると、新しいファイルのみがアップロードされます。
- セントラルに存在するものと同じ名前で内容が異なるファイルを含む新しいサポートパッケージをアップロードすると、**roxctl** は警告メッセージを表示し、ファイルを上書きしません。
  - **--overwrite** オプションを upload コマンドとともに使用して、ファイルを上書きできます。
- 必要なプローブを含むサポートパッケージをアップロードすると、セントラルはこのプローブをダウンロードするための (インターネットへの) アウトバウンドリクエストを行いません。セントラルは、サポートパッケージのプローブを使用します。

第6章 アラートデータの保持を有効にする

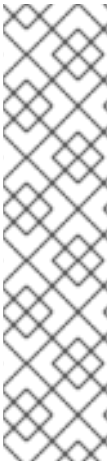
Red Hat Advanced Cluster Security for Kubernetes アラートの保持期間を設定する方法を学びます。

Red Hat Advanced Cluster Security for Kubernetes を使用すると、履歴アラートを保存する時間を設定できます。次に、Red Hat Advanced Cluster Security for Kubernetes は、指定された時間が経過すると古いアラートを削除します。

不要になったアラートを自動的に削除することで、ストレージコストを節約できます。

保存期間を設定できるアラートには、次のものがあります。

- 未解決 (アクティブ) と解決済みの両方のランタイムアラート。
- 現在のデプロイメントに適用されない古いデプロイ時アラート。



注記

- データ保持設定はデフォルトで有効になっています。これらの設定は、インストール後に変更できます。
- Red Hat Advanced Cluster Security for Kubernetes をアップグレードする場合、以前に有効にしていない限り、データ保持設定は適用されません。
- RHACS ポータルまたは API を使用して、アラート保持の設定を行うことができます。
- 削除プロセスは 1 時間ごとに実行されます。現在、これを変更することはできません。

6.1. アラートデータ保持の設定

RHACS ポータルを使用することで、アラート保持の設定を行うことができます。

前提条件

- データの保持を設定するには、**read** および **write** 権限を持つ **Config** ロールが必要である。

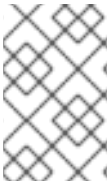
手順

1. RHACS ポータルで、**Platform Configuration → System Configuration** に移動します。
2. **System Configuration** ビューのヘッダーで、**Edit** をクリックします。
3. **Data Retention Configuration** セクションで、各タイプのデータの日数を更新します。
   - **すべてのランタイム違反**
   - **解決されたデプロイフェーズ違反**
   - **削除されたデプロイメントのランタイム違反**
   - **デプロイされなくなったイメージ**

注記

あるタイプのデータを永久に保存するには、保存期間を **0** 日に設定します。

4. **Save** をクリックします。



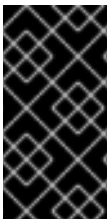
注記

Red Hat Advanced Cluster Security for Kubernetes API を使用してアラートデータの保持を設定するには、API リファレンスドキュメントの **ConfigService** グループにある **PutConfig** と関連する API を確認してください。

第7章 HTTP を介した RHACS ポータルの公開

暗号化されていない HTTP サーバーを有効にして、ingress コントローラー、Layer 7 ロードバランサー、Istio、またはその他のソリューションを介して RHACS ポータルを公開します。

暗号化されていない HTTP バックエンドを優先するイングレスコントローラー、Istio、または Layer 7 ロードバランサーを使用する場合、HTTP を介して RHACS ポータルを公開するように Red Hat Advanced Cluster Security for Kubernetes を設定できます。これを行うと、RHACS ポータルがプレーンテキストのバックエンドで利用できるようになります。



重要

HTTP 経由で RHACS ポータルを公開するには、ingress コントローラー、Layer 7 ロードバランサー、または Istio を使用して外部トラフィックを HTTPS で暗号化する必要があります。プレーン HTTP を使用して RHACS ポータルを外部クライアントに直接公開することは安全ではありません。

インストール中または既存のデプロイメントで、HTTP を介して RHACS ポータルを公開できます。

7.1. 前提条件

- HTTP エンドポイントを指定するには、**<endpoints_spec>** を使用する必要があります。これは、**<type>@<addr>:<port>** という形式のシングルエンドポイント仕様のコンマ区切りリストです。
    - **type** は **grpc** または **http** です。type として **http** を使用すると、ほとんどのユースケースで機能します。高度なユースケースでは、**grpc** を使用するか、その値を省略できます。**type** の値を省略すると、プロキシーで 2 つのエンドポイントを設定できます。1 つは gRPC 用で、もう 1 つは HTTP 用です。これらのエンドポイントは、いずれもセントラルで公開されている同じ HTTP ポートを指しています。しかし、ほとんどのプロキシーは、gRPC と HTTP の両方のトラフィックを同じ外部ポートで伝送することをサポートしていません。
    - **addr** は、セントラルを公開する IP アドレスです。これを省略するか、ポート転送を使用してのみアクセスできる HTTP エンドポイントが必要な場合は **localhost** または **127.0.0.1** を使用できます。
    - **port** は、セントラルを公開するポートです。
    - 以下は、いくつかの有効な **<endpoints_spec>** 値です。
        - **8080**
        - **http@8080**
        - **:8081**
        - **grpc@:8081**
        - **localhost:8080**
        - **http@localhost:8080**
        - **http@8080,grpc@8081**
        - **8080, grpc@:8081, http@0.0.0.0:8082**

7.2. インストール中に HTTP を介して RHACS ポータルを公開する

roxctl CLI を使用して Red Hat Advanced Cluster Security for Kubernetes をインストールする場合は、**roxctl central generate interactive** コマンドで **--plaintext-endpoints** オプションを使用して、インストール中に HTTP サーバーを有効にします。

手順

- 次のコマンドを実行して、対話型インストールプロセス中に HTTP エンドポイントを指定します。

```
$ roxctl central generate interactive \
  --plaintext-endpoints=<endpoints_spec> 1
```

1 **<type>@<addr>:<port>** の形式のエンドポイント仕様です。詳細については、前提条件セクションを参照してください。

7.3. 既存のデプロイメント用の HTTP での RHACS ポータル公開

既存の Red Hat Cluster Security for Kubernetes デプロイメントで HTTP サーバーを有効にできます。

手順

1. パッチを作成し、**ROX_PLAINTEXT_ENDPOINTS** 環境変数を定義します。

```
$ CENTRAL_PLAINTEXT_PATCH='
spec:
  template:
    spec:
      containers:
      - name: central
        env:
        - name: ROX_PLAINTEXT_ENDPOINTS
          value: <endpoints_spec> 1
'
```

1 **<type>@<addr>:<port>** の形式のエンドポイント仕様です。詳細については、前提条件セクションを参照してください。

2. **ROX_PLAINTEXT_ENDPOINTS** 環境変数をセントラルデプロイメントに追加します。

```
$ oc -n stackrox patch deploy/central -p "$CENTRAL_PLAINTEXT_PATCH"
```

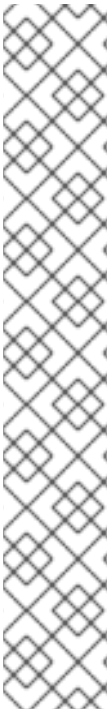
第8章 セキュリティー保護されたクラスターの自動アップグレードの設定

セキュリティー保護された各クラスターのアップグレードプロセスを自動化し、RHACS ポータルからアップグレードステータスを表示できます。

自動アップグレードでは、セキュリティー保護された各クラスターをアップグレードする手動タスクを自動化することで、最新の状態を維持しやすくなります。

自動アップグレードでは、セントラルをアップグレードした後、セキュリティー保護されたすべてのクラスターのセンサー、コレクター、およびコンプライアンスサービスは、自動的に最新バージョンにアップグレードされます。

Red Hat Advanced Cluster Security for Kubernetes を使用すると、RHACS ポータル内からすべてのセキュリティー保護されたクラスターを集中管理することもできます。新しい **Clusters** ビューには、セキュリティー保護されたすべてのクラスター、すべてのクラスターのセンサーバージョン、およびアップグレードステータスメッセージに関する情報が表示されます。このビューを使用して、セキュリティー保護されたクラスターを選択的にアップグレードしたり、設定を変更したりすることもできます。



注記

- 自動アップグレード機能はデフォルトで有効になっています。
- プライベートイメージレジストリーを使用している場合は、最初にセンサーイメージとコレクターイメージをプライベートレジストリーにプッシュする必要があります。
- センサーは、デフォルトの RBAC 権限で実行する必要があります。
- 自動アップグレードでは、クラスターで実行されている Red Hat Advanced Cluster Security for Kubernetes サービスに適用したパッチは保持されません。ただし、Red Hat Advanced Cluster Security for Kubernetes オブジェクトに追加したすべてのラベルとアノテーションは保持されます。
- デフォルトでは、Red Hat Advanced Cluster Security for Kubernetes は、セキュアな各クラスターに **sensor-upgrader** と呼ばれる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。このアカウントを削除すると、センサーに十分な権限がないため、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。

8.1. 自動アップグレードの有効化

すべてのセキュリティー保護されたクラスターの自動アップグレードを有効にして、それらのクラスターのコレクターとコンプライアンスサービスを最新バージョンに自動的にアップグレードできます。

手順

1. RHACS ポータルで、**Platform Configuration → Clusters** に移動します。
2. **Automatically upgrade secured clusters** トグルを有効にします。

注記

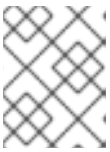
新規インストールの場合、**Automatically upgrade secured clusters**トグルはデフォルトで有効になっています。

8.2. 自動アップグレードを無効にする

セキュリティー保護されたクラスターのアップグレードを手動で管理する場合は、自動アップグレードを無効にすることができます。

手順

1. RHACS ポータルで、**Platform Configuration → Clusters** に移動します。
2. **Automatically upgrade secured clusters**トグルを無効にします。



注記

新規インストールの場合、**Automatically upgrade secured clusters**トグルはデフォルトで有効になっています。

8.3. 自動アップグレードステータス

Clusters ビューには、すべてのクラスターとそのアップグレードステータスが一覧表示されます。

アップグレードステータス	説明
セントラルバージョンで最新	セキュリティー保護されたクラスターは、セントラルと同じバージョンを実行しています。
アップグレード可能	センサーとコレクターの新しいバージョンが利用可能です。
アップグレードに失敗。アップグレードを再試行。	以前の自動アップグレードは失敗しました。
手動アップグレードが必要	センサーとコレクターのバージョンは、バージョン 2.5.29.0 よりも古いバージョンです。セキュリティー保護された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。
プリフライトチェックが完了	アップグレードが進行中です。自動アップグレードを実行する前に、アップグレードインストーラーはプリフライトチェックを実行します。プリフライトのチェック中に、インストーラーは特定の条件が満たされているかどうかを確認してから、アップグレードプロセスのみを開始します。

8.4. 自動アップグレードの失敗

場合によっては、Red Hat Advanced Cluster Security for Kubernetes の自動アップグレードがインストールに失敗することがあります。アップグレードが失敗すると、セキュリティー保護されたクラスターのステータスメッセージは次のように変わります。**Upgrade failed.Retry upgrade**。失敗に関する

詳細情報を表示し、アップグレードが失敗した理由を理解するには、**Clusters** ビューでセキュリティー保護されたクラスター行を確認できます。

失敗の一般的な理由は次のとおりです。

- イメージが欠落しているか、スケジュールできないため、センサーアップグレーダーのデプロイメントが実行されなかった可能性があります。
- RBAC 権限が不十分であるか、クラスターの状態が認識できないために、プリフライトチェックが失敗した可能性があります。これは、Red Hat Advanced Cluster Security for Kubernetesy のサービス設定を編集した場合、または **auto-upgrade.stackrox.io/component** ラベルが欠落している場合に発生する可能性があります。
- アップグレードの実行中にエラーが発生する可能性があります。これが発生した場合、アップグレードインストーラーは自動的にアップグレードのロールバックを試みます。

注記

場合によっては、ロールバックも失敗する可能性があります。このような場合は、クラスターログを表示して問題を特定するか、サポートにお問合せください。

アップグレードの失敗の根本原因を特定して修正したら、**Retry Upgrade** オプションを使用して、セキュリティー保護されたクラスターをアップグレードできます。

8.5. RHACS ポータルからセキュリティー保護されたクラスターを手動でアップグレードする

自動アップグレードを有効にしたくない場合は、**Clusters** ビューを使用して、セキュリティー保護されたクラスターのアップグレードを管理できます。

セキュリティー保護されたクラスターのアップグレードを手動でトリガーするには:

手順

1. RHACS ポータルで、**Platform Configuration → Clusters** に移動します。
2. アップグレードするクラスターの **Upgrade status** 列で、**Upgrade available** オプションを選択します。
3. 複数のクラスターを一度にアップグレードするには、更新するクラスターの **Cluster** 列のチェックボックスを選択します。
4. **アップグレード** をクリックします。

第9章 外部ネットワークアクセス用のプロキシーの設定

ネットワーク設定でプロキシー経由のアウトバウンドトラフィックが制限されている場合は、Red Hat Advanced Cluster Security for Kubernetes でプロキシー設定を設定して、プロキシー経由でトラフィックをルーティングできます。

Red Hat Advanced Cluster Security for Kubernetes でプロキシーを使用する場合:

- セントラルおよびスキャナーからのすべての出力 HTTP、HTTPS、およびその他の TCP トラフィックは、プロキシーを通過します。
- セントラルとスキャナー間のトラフィックはプロキシーを通過しません。
- プロキシー設定は、他の Red Hat Advanced Cluster Security for Kubernetes コンポーネントには影響しません。
- オフラインモードを使用しておらず、セキュリティー保護されたクラスターで実行されているコレクターが、実行時に追加のカーネルモジュールまたは eBPF プローブをダウンロードする必要がある場合:
  - コレクターはセンサーに連絡してダウンロードを試みます。
  - 次に、センサーはこのリクエストをセントラルに転送します。
  - セントラルはプロキシーを使用して、**https://collector-modules.stackrox.io** でモジュールまたはプローブを見つけます。

9.1. 既存のデプロイメントでのプロキシーの設定

既存のデプロイメントでプロキシーを設定するには、**proxy-config** シークレットを YAML ファイルとしてエクスポートし、そのファイルのプロキシー設定を更新して、シークレットとしてアップロードする必要があります。

手順

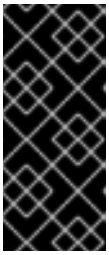
1. 既存のシークレットを YAML ファイルとして保存します。

```
$ oc -n stackrox get secret proxy-config \
  -o go-template='{{index .data "config.yaml" | \
  base64decode}}{{"\n"}}' > /tmp/proxy-config.yaml
```

2. インストール中にプロキシーを設定するセクションで指定されているように、YAML 設定ファイルで変更するフィールドを編集します。

3. 変更を保存した後、次のコマンドを実行してシークレットを置き換えます。

```
$ oc -n stackrox create secret generic proxy-config \
  --from-file=config.yaml=/tmp/proxy-config.yaml -o yaml --dry-run | \
  oc label -f - --local -o yaml app.kubernetes.io/name=stackrox | \
  oc apply -f -
```



重要

- OpenShift Container Platform が変更をセントラルとスキャナーに伝播するまで少なくとも 1 分待つ必要があります。
- プロキシー設定を変更した後に発信接続に問題が発生した場合は、セントラル Pod とスキャナー Pod を再起動する必要があります。

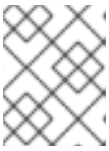
9.2. インストール中にプロキシーを設定する

roxctl コマンドラインインターフェイス (CLI) または Helm を使用して Red Hat Advanced Cluster Security for Kubernetes をインストールする場合、インストール中にプロキシー設定を指定できます。

roxctl central generate コマンドを使用してインストーラーを実行すると、インストーラーはご使用の環境のシークレットとデプロイメント設定ファイルを生成します。生成された設定シークレット (YAML) ファイルを編集して、プロキシーを設定できます。現在、**roxctl** CLI を使用してプロキシーを設定することはできません。設定は Kubernetes シークレットに保存され、セントラルとスキャナーの両方で共有されます。

手順

1. デプロイメントバンドルディレクトリーから設定ファイル **central/proxy-config-secret.yaml** を開きます。



注記

Helm を使用している場合、設定ファイルは **central/templates/proxy-config-secret.yaml** にあります。

2. 設定ファイルで変更するフィールドを編集します。

```
apiVersion: v1
kind: Secret
metadata:
  namespace: stackrox
  name: proxy-config
type: Opaque
stringData:
  config.yaml: |- (1)
    # # NOTE: Both central and scanner should be restarted if this secret is changed.
    # # While it is possible that some components will pick up the new proxy configuration
    # # without a restart, it cannot be guaranteed that this will apply to every possible
    # # integration etc.
    # url: http://proxy.name:port (2)
    # username: username (3)
    # password: password (4)
    # # If the following value is set to true, the proxy wil NOT be excluded for the default hosts:
    # # - *.stackrox, *.stackrox.svc
    # # - localhost, localhost.localdomain, 127.0.0.0/8, ::1
    # # - *.local
    # omitDefaultExcludes: false
    # excludes:  # hostnames (may include * components) for which you do not (5)
    # # want to use a proxy, like in-cluster repositories.
    # - some.domain
```

```
    # # The following configuration sections allow specifying a different proxy to be used for
HTTP(S) connections.
    # # If they are omitted, the above configuration is used for HTTP(S) connections as well as
TCP connections.
    # # If only the `http` section is given, it will be used for HTTPS connections as well.
    # # Note: in most cases, a single, global proxy configuration is sufficient.
    # http:
    #   url: http://http-proxy.name:port 6
    #   username: username 7
    #   password: password 8
    # https:
    #   url: http://https-proxy.name:port 9
    #   username: username 10
    #   password: password 11
```

3 4 7 8 10 11 **username** と **password** の追加は、最初と **http** と **https** セクションの両方で任意に行うことができます。

2 6 9 **url** オプションは、次の URL スキームをサポートします。

- HTTP プロキシーの場合は **http:**//。
- TLS が有効化された HTTP プロキシーの場合は **https:**//。
- SOCKS5 プロキシーの場合は **socks5:**//。

5 **excludes** リストには、DNS 名 (* ワイルドカードの有無にかかわらず)、IP アドレス、または CIDR 表記の IP ブロック (たとえば、**10.0.0.0/8**) を含めることができます。このリストの値は、プロトコルに関係なく、すべての出力接続に適用されます。

1 **stringData** セクションの **|-** 行は、設定データの開始を示します。

注記

- 最初にファイルを開くと、すべての値がコメントアウトされます (行の先頭にある **#** 記号を使用)。二重ハッシュ記号で始まる行 **# #** には、設定キーの説明が含まれています。
- フィールドを編集するときは、**config.yaml: |-** 行に対して 2 つのスペースのインデントレベルを維持していることを確認してください。

3. 設定ファイルを編集した後、通常のインストールを続行できます。更新された設定は、提供されたアドレスとポート番号で実行されているプロキシーを使用するように Red Hat Advanced Cluster Security for Kubernetes に指示します。

第10章 診断バンドルの生成

診断バンドルを生成し、そのデータを送信して、サポートチームが Red Hat Advanced Cluster Security for Kubernetes コンポーネントのステータスと正常性に関する洞察を提供できるようにすることができます。

Red Hat は、Red Hat Advanced Cluster Security for Kubernetes の問題の調査中に、診断バンドルの送信をリクエストする場合があります。診断バンドルを生成し、送信する前にそのデータを検査できます。

注記

診断バンドルは暗号化されておらず、環境内のクラスターの数に応じて、バンドルサイズは 100 KB から 1 MB の間です。このデータを Red Hat に転送するには、常に暗号化されたチャネルを使用してください。

10.1. 診断バンドルデータ

診断バンドルを生成すると、次のデータが含まれます。

- セントラルヒーププロファイル。
- システムログ: すべての Red Hat Advanced Cluster Security for Kubernetes コンポーネントのログ (過去 20 分間) と、最近クラッシュしたコンポーネントのログ (クラッシュの最大 20 分前)。システムログは、環境のサイズによって異なります。大規模なデプロイメントの場合、データには、再起動回数が多いなど、重大なエラーのみが発生したコンポーネントのログファイルが含まれます。
- Red Hat Advanced Cluster Security for Kubernetes コンポーネントの YAML 定義: このデータには Kubernetes シークレットは含まれていません。
- OpenShift Container Platform または Kubernetes イベント: **stackrox** 名前空間内のオブジェクトに関連するイベントの詳細。
- オンライン Telemetry データには、次のものが含まれます。
    - ストレージ情報: データベースのサイズと、接続されたボリュームで使用可能な空き領域の量に関する詳細。
    - Red Hat Advanced Cluster Security for Kubernetes コンポーネントの可用性情報: Red Hat Advanced Cluster Security for Kubernetes コンポーネントのバージョン、それらのメモリー使用量、および報告されたエラーに関する詳細。
    - 粒度の細かい使用状況に関する統計: API エンドポイント呼び出しカウントと報告されたエラーステータスに関する詳細。API リクエストで送信される実際のデータは含まれません。
    - ノード情報: セキュリティー保護された各クラスターのノードに関する詳細。これには、カーネルとオペレーティングシステムのバージョン、リソースのプレッシャー、および taint が含まれます。
    - 環境情報: Kubernetes または OpenShift Container Platform のバージョン、Istio バージョン (該当する場合)、クラウドプロバイダーのタイプ、およびその他の同様の情報を含む、セキュリティー保護された各クラスターに関する詳細。

10.2. RHACS ポータルを使用した診断バンドルの生成

RHACS ポータルのシステムヘルスダッシュボードを使用して、診断バンドルを生成できます。

前提条件

- 診断バンドルを生成するには、**DebugLogs** リソースの **read** 権限が必要。

手順

1. RHACS ポータルで、**Platform Configuration → System Health** を選択します。
2. **System Health** ビューヘッダーで、**Generate Diagnostic Bundle** をクリックします。
3. **Filter by clusters** ドロップダウンメニューで、診断データを生成するクラスターを選択します。
4. **Filter by starting time** で、診断データを含める日付および時刻 (UTC 形式) を指定します。
5. **Download Diagnostic Bundle** をクリックします。

10.3. ROXCTL CLI を使用した診断バンドルの生成

roxctl CLI を使用して診断バンドルを生成できます。

前提条件

- 診断バンドルを生成するには、**DebugLogs** リソースの **read** 権限が必要。

手順

- 次のコマンドを実行して、診断バンドルを生成します。

```
$ roxctl central debug download-diagnostics
```

第11章 エンドポイントの設定

YAML 設定ファイルを使用して、Red Hat Advanced Cluster Security for Kubernetes (RHACS) のエンドポイントを設定する方法を学習します。

YAML 設定ファイルを使用して、公開されたエンドポイントを設定できます。この設定ファイルを使用して、Red Hat Advanced Cluster Security for Kubernetes の 1 つ以上のエンドポイントを定義し、各エンドポイントの TLS 設定をカスタマイズしたり、特定のエンドポイントの TLS を無効にしたりできます。また、クライアント認証が必要かどうか、およびどのクライアント証明書を受け入れるかを定義することもできます。

11.1. カスタム YAML 設定

Red Hat Advanced Cluster Security for Kubernetes は、YAML 設定を **ConfigMap** として使用し、設定の変更と管理を容易にします。

カスタム YAML 設定ファイルを使用する場合、エンドポイントごとに以下を設定できます。

- **HTTP**、**gRPC** など、またはその両方を使用するプロトコル。
- TLS を有効または無効にします。
- サーバー証明書を指定します。
- クライアント認証を信頼するクライアント認証局 (CA)。
- クライアント認証局 (**mTLS**) が必要かどうかを指定します。

設定ファイルを使用して、インストール中または Red Hat Advanced Cluster Security for Kubernetes の既存のインスタンスでエンドポイントを指定できます。ただし、デフォルトのポート **8443** 以外の追加のポートを公開する場合は、それらの追加のポートでのトラフィックを許可するネットワークポリシーを作成する必要があります。

以下は、Red Hat Advanced Cluster Security for Kubernetes のサンプル **endpoints.yaml** 設定ファイルです。

```
# Sample endpoints.yaml configuration for Central.
#
# # CAREFUL: If the following line is uncommented, do not expose the default endpoint on port 8443 by default.
# #         This will break normal operation.
# disableDefault: true # if true, do not serve on :8443 (1)
endpoints: (2)
  # Serve plaintext HTTP only on port 8080
  - listen: ":8080" (3)
    # Backend protocols, possible values are 'http' and 'grpc'. If unset or empty, assume both.
    protocols: (4)
      - http
    tls: (5)
      # Disable TLS. If this is not specified, assume TLS is enabled.
      disable: true (6)
  # Serve HTTP and  gRPC for sensors only on port 8444
  - listen: ":8444" (7)
    tls: (8)
```

```
    # Which TLS certificates to serve, possible values are 'service' (For  service certificates that Red Hat Advanced Cluster Security for Kubernetes generates)
    # and 'default' (user-configured default TLS certificate). If unset or empty, assume both.
    serverCerts: 9
      - default
      - service
    # Client authentication settings.
    clientAuth: 10
      # Enforce TLS client authentication. If unset, do not enforce, only request certificates
      # opportunistically.
      required: true 11
      # Which TLS client CAs to serve, possible values are 'service' (CA for service
      # certificates that Red Hat Advanced Cluster Security for Kubernetes generates) and 'user' (CAs for PKI auth providers). If unset or empty, assume both.
      certAuthorities: 12
      # if not set, assume ["user", "service"]
        - service
```

1 **true** を使用して、デフォルトのポート番号 **8443** での公開を無効にします。デフォルト値は **false** です。**true** に変更すると、既存の機能が破損する可能性があります。

2 セントラルを公開するための追加のエンドポイントのリスト。

3 7 リッスンするアドレスとポート番号。**endpoints** を使用している場合は、この値を指定する必要があります。形式 **port**、**:port**、または **address:port** を使用して、値を指定できます。以下に例を示します。

- **8080** または **:8080** - すべてのインターフェイスのポート **8080** でリッスンします。
- **0.0.0.0:8080** - すべての IPv4 (IPv6 ではない) インターフェイスのポート **8080** でリッスンします。
- **127.0.0.1:8080** - ローカルループバックデバイスのポート **8080** でのみリッスンします。

4 指定されたエンドポイントに使用するプロトコル。使用できる値は **http** と **grpc** です。値を指定しない場合、セントラルは指定されたポートで HTTP トラフィックと gRPC トラフィックの両方をリッスンします。RHACS ポータル専用のエンドポイントを公開する場合は、**http** を使用します。ただし、これらのクライアントは gRPC と HTTP の両方を必要とするため、サービス間通信または **roxctl** CLI にエンドポイントを使用することはできません。エンドポイントで HTTP プロトコルと Red Hat プロトコルの両方を有効にするために、このキーの値を指定しないことをお勧めします。エンドポイントを Red Hat Advanced Cluster Security for Kubernetes サービスのみに制限する場合は、**clientAuth** オプションを使用します。

5 8 これを使用して、エンドポイントの TLS 設定を指定します。値を指定しない場合、Red Hat Advanced Cluster Security for Kubernetes は、以下のすべてのネストされたキーのデフォルト設定で TLS を有効にします。

6 指定したエンドポイントで TLS を無効にするには、**true** を使用します。デフォルト値は **false** です。**true** に設定すると、**serverCerts** と **clientAuth** の値を指定できなくなります。

9 サーバー TLS 証明書を設定するソースのリストを指定します。**serverCerts** リストは順序に依存します。つまり、一致する SNI (Server Name Indication) がない場合、リストの最初の項目がセントラルがデフォルトで使用する証明書を決定します。これを使用して複数の証明書を指定でき、セントラルは SNI に基づいて適切な証明書を自動的に選択します。設定可能な値は以下のとおりです。

- **default**: 設定済みのカスタム TLS 証明書が存在する場合はそれを使用します。
- **service**: Red Hat Advanced Cluster Security for Kubernetes が生成する内部サービス証明書を使用します。

10 これを使用して、TLS が有効なエンドポイントのクライアント証明書認証の動作を設定します。

11 **true** を使用して、有効なクライアント証明書を持つクライアントのみを許可します。デフォルト値は **false** です。**true** を **service** の **certAuthorities** 設定と組み合わせて使用すると、Red Hat Advanced Cluster Security for Kubernetes サービスのみがこのエンドポイントに接続できるようになります。

12 クライアント証明書を検証するための CA のリスト。デフォルト値は **["service", "user"]** です。**certAuthorities** リストは順序に依存しません。つまり、このリスト内のアイテムの位置は重要ではありません。また、空のリスト **[]** として設定すると、エンドポイントのクライアント証明書認証が無効になります。これは、この値を未設定のままにするのとは異なります。設定可能な値は以下のとおりです。

- **service**: Red Hat Advanced Cluster Security for Kubernetes が生成するサービス証明書の CA。
- **user**: PKI 認証プロバイダーによって設定された CA。

11.2. 新規インストール中のエンドポイントの設定

roxctl CLI を使用して Red Hat Advanced Cluster Security for Kubernetes をインストールすると、**central-bundle** という名前のフォルダーが作成され、セントラルのデプロイに必要な YAML マニフェストとスクリプトが格納されます。

手順

1. **central-bundle** を生成した後、**./central-bundle/central/02-endpoints-config.yaml** ファイルを開きます。
2. このファイルで、キー **endpoints.yaml** の **data:** セクションにカスタム YAML 設定を追加します。YAML 設定用に 4 つのスペースインデントを維持していることを確認してください。
3. 通常どおりインストール手順を続行します。Red Hat Advanced Cluster Security for Kubernetes は、指定された設定を使用します。

注記

デフォルトのポート **8443** 以外の追加のポートを公開する場合は、それらの追加のポートでのトラフィックを許可するネットワークポリシーを作成する必要があります。

11.3. 既存のインスタンスのエンドポイントの設定

Red Hat Advanced Cluster Security for Kubernetes の既存のインスタンスのエンドポイントを設定できます。

手順

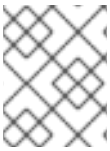
1. 既存の設定マップをダウンロードします。

```
$ oc -n stackrox get cm/central-endpoints -o go-template='{{index .data "endpoints.yaml"}}'  >
<directory_path>/central_endpoints.yaml
```

2. ダウンロードした **central_endpoints.yaml** ファイルで、カスタム YAML 設定を指定します。
3. 変更した **central_endpoints.yaml** 設定ファイルをアップロードして適用します。

```
$ oc -n stackrox create cm central-endpoints --from-file=endpoints.yaml=<directory-
path>/central-endpoints.yaml -o yaml --dry-run | \
oc label -f - --local -o yaml app.kubernetes.io/name=stackrox | \
oc apply -f -
```

4. セントラルを再起動します。



注記

デフォルトのポート **8443** 以外の追加のポートを公開する場合は、それらの追加のポートでのトラフィックを許可するネットワークポリシーを作成する必要があります。

11.3.1. セントラルコンテナーの再起動

セントラルコンテナーを強制終了するか、セントラル Pod を削除することで、セントラルコンテナーを再起動できます。

手順

- 次のコマンドを実行して、セントラルコンテナーを強制終了します。

注記

OpenShift Container Platform が変更を伝播し、セントラルコンテナーを再始動するまで、少なくとも 1 分間待機する必要があります。

```
$ oc -n stackrox exec deploy/central -c central -- kill 1
```

- または、次のコマンドを実行して セントラル Pod を削除します。

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

11.4. カスタムポートを介したトラフィックフローの有効化

同じクラスターで実行されている別のサービスまたは ingress コントローラーにポートを公開する場合は、クラスター内のサービスまたは ingress コントローラーのプロキシーからのトラフィックのみを許可する必要があります。それ以外の場合、ロードバランサーサービスを使用してポートを公開している場合は、外部ソースを含むすべてのソースからのトラフィックを許可することをお勧めします。このセクションにリストされている手順を使用して、すべてのソースからのトラフィックを許可します。

手順

1. **allow-ext-to-central** Kubernetes ネットワークポリシーのクローンを作成します。

```
$ oc -n stackrox get networkpolicy.networking.k8s.io/allow-ext-to-central -o yaml >
<directory_path>/allow-ext-to-central-custom-port.yaml
```

2. これを参照として使用してネットワークポリシーを作成し、そのポリシーで、公開するポート番号を指定します。ビルトインの **allow-ext-to-central** ポリシーを妨げないように、YAML ファイルの **metadata** セクションでネットワークポリシーの名前を変更してください。

第12章 PROMETHEUS による監視

Prometheus は、オープンソースのモニターリングおよびアラートプラットフォームです。これを使用して、Red Hat Advanced Cluster Security for Kubernetes のセントラルおよびセンサーコンポーネントの健全性と可用性を監視できます。

12.1. モニターリングの有効化

Kubernetes の Red Hat Cluster Security をモニターする前に、監視を有効にする必要があります。

手順

1. サービスにパッチを適用して、ポート番号 **9090** を公開します。
   a. センサーサービスにパッチを適用します。

```
$ oc -n stackrox patch svc/sensor -p '{"spec":{"ports":
[{"name":"monitoring","port":9090,"protocol":"TCP","targetPort":9090}]}}' 1
```

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

   b. セントラルサービスにパッチを適用します。

```
$ oc -n stackrox patch svc/central -p '{"spec":{"ports":
[{"name":"monitoring","port":9090,"protocol":"TCP","targetPort":9090}]}}'
```

2. Ingress を許可するようにネットワークポリシーを変更します。

```
$ oc apply -f - <<EOF 1
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  labels:
    app.kubernetes.io/name: stackrox
  name: allow-monitoring
  namespace: stackrox
spec:
  ingress:
  - ports:
    - port: 9090
      protocol: TCP
  podSelector:
    matchExpressions:
    - {key: app, operator: In, values: [central, sensor, collector]}
  policyTypes:
  - Ingress
EOF
```

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

12.1.1. デフォルトポートのカスタマイズ

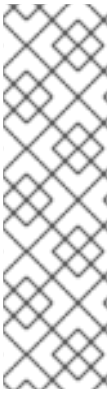
Red Hat Advanced Cluster Security for Kubernetes セントラルとセンサーで Prometheus メトリックに使用されるポートをカスタマイズするには、**ROX_METRICS_PORT** 環境変数を使用できます。

手順

- **ROX_METRICS_PORT** 環境変数を設定します。

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

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



注記

ROX_METRICS_PORT 環境変数の **<value>** は次のように指定できます。

- モニターリングを **disabled** にして、無効にします。
- **<port_number>** を使用して、ワイルドカードアドレスにバインドします。
- **<address>:<port_number>** は、特定のアドレスとポート番号を使用します。**[2001:db8::1234]:9090**. のように、角かっこを使用して IPv6 アドレスを指定することもできます。

第13章 監査ログの設定

Red Hat Advanced Cluster Security for Kubernetes は、Red Hat Advanced Cluster Security for Kubernetes で行われたすべての変更を確認するために使用できる監査ログ機能を提供します。監査ログには、Red Hat Advanced Cluster Security for Kubernetes の変更であるすべての **PUT** および **POST** イベントがキャプチャされます。この情報を使用して、問題のトラブルシューティングを行ったり、ロールや権限の変更などの重要なイベントを記録したりします。監査ログを使用すると、Red Hat Advanced Cluster Security for Kubernetes で発生したすべての正常なイベントと異常なイベントの全体像を把握できます。

注記

監査ロギングはデフォルトでは有効になっていません。監査ログを手動で有効にする必要があります。

警告

現在、監査ログメッセージのメッセージ配信保証はありません。

13.1. 監査ロギングの有効化

監査ログを有効にすると、変更があるたびに、Red Hat Advanced Cluster Security forKubernetes が設定されたシステムに HTTP POST メッセージ (JSON 形式) を送信します。

前提条件

- Red Hat Advanced Cluster Security for Kubernetes のログメッセージを処理するように Splunk または別の Webhook レシーバーを設定する。
- 自分のロールの **Notifiers** リソースで **write** 権限を有効にする必要がある。

手順

1. RHACS ポータルで、**Platform Configuration → Integrations** に移動します。
2. **Notifier Integrations** セクションまでスクロールダウンし、**Generic Webhook** または **Splunk** を選択します。
3. 必要な情報を入力し、**Enable Audit Logging** を有効にするトグルをオンにします。

13.2. 監査ログメッセージのサンプル

ログメッセージの形式は次のとおりです。

```
{
  "headers": {
    "Accept-Encoding": [
      "gzip"
    ],
```

```
    "Content-Length": [
      "586"
    ],
    "Content-Type": [
      "application/json"
    ],
    "User-Agent": [
      "Go-http-client/1.1"
    ]
  },
  "data": {
    "audit": {
      "interaction": "CREATE",
      "method": "UI",
      "request": {
        "endpoint": "/v1/notifiers",
        "method": "POST",
        "payload": {
          "@type": "storage.Notifier",
          "enabled": true,
          "generic": {
            "auditLoggingEnabled": true,
            "endpoint": "http://samplewebhookserver.com:8080"
          },
          "id": "b53232ee-b13e-47e0-b077-1e383c84aa07",
          "name": "Webhook",
          "type": "generic",
          "uiEndpoint": "https://localhost:8000"
        }
      },
      "status": "REQUEST_SUCCEEDED",
      "time": "2019-05-28T16:07:05.500171300Z",
      "user": {
        "friendlyName": "John Doe",
        "role": {
          "globalAccess": "READ_WRITE_ACCESS",
          "name": "Admin"
        },
        "username": "john.doe@example.com"
      }
    }
  }
}
```