Red Hat Advanced Cluster Security for Kubernetes 4.5

設定

Red Hat Advanced Cluster Security for Kubernetes の設定

Last Updated: 2024-09-16

Red Hat Advanced Cluster Security for Kubernetes 4.5 設定

Red Hat Advanced Cluster Security for Kubernetes の設定

法律上の通知

概要

このドキュメントでは、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、Central 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 Sensor を設定する必要もがあります。

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

- Operator を使用して Red Hat Advanced Cluster Security for Kubernetes をインストールする場合:

1. 次のコマンドを入力して、Central サービスがインストールされる namespace に適切な TLS 証明書が含まれる **central-default-tls-cert** シークレットを作成します。

```
oc -n <namespace> create secret tls central-default-tls-cert --cert <tls-cert.pem> --key <tls-key.pem>
```

- 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

- Operator を使用して Red Hat Advanced Cluster Security for Kubernetes をインストールした場合。

  1. 次のコマンドを入力して、Central サービスがインストールされている namespace に適切な TLS 証明書が含まれる **central-default-tls-cert** シークレットを作成します。

     ```
     oc -n <namespace> create secret tls central-default-tls-cert --cert <tls-cert.pem> --key
     <tls-key.pem>
     ```

- 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 --reuse-values \(1)
      -f values-private.yaml
     ```

     (1) **value-private.yaml** ファイルには必要な設定値がすべて含まれているわけではないため、このパラメーターを使用する必要があります。

- **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、Central 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

Central 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 -
```

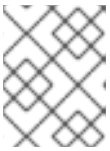
3. Central コンテナーを再起動します。

1.1.4.1. Central コンテナーの再起動

Central コンテナーを強制終了するか、Central Pod を削除して、Central 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、Central 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、Central 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。

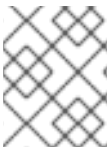
```
$ oc -n stackrox exec deploy/central -c central -- kill 1
```

- または、次のコマンドを実行して Central 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な SENSOR の設定

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 Sensor 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 Sensor を設定する必要はありません。

1.2.1. Sensor バンドルのダウンロード

Sensor バンドルには、Sensor のインストールに必要な設定ファイルとスクリプトが含まれています。Sensor バンドルは RHACS ポータルからダウンロードできます。

手順

1. RHACS ポータルで、**Platform Configuration → Clusters** に移動します。
2. **New Cluster** をクリックして、クラスターの名前を指定します。
3. 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**)。

4. **Next** をクリックして先に進みます。
5. **Download YAML File and Keys**クリックします。

1.2.2. 新規 Sensor のデプロイ時にカスタムの証明書を信頼するように Sensor を設定する手順

前提条件

- Sensor バンドルをダウンロードしている。

手順

- **sensor.sh** スクリプトを使用している場合:
  1. Sensor バンドルを展開します。

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

  2. **sensor.sh** スクリプトを実行します。

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

  Sensor (**./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:

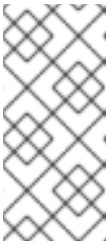
1. Sensor バンドルを展開します。

```
$ 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 Sensor 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 Sensor を設定する手順

前提条件

- Sensor バンドルをダウンロードしている。

手順

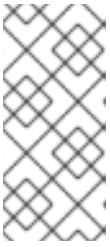
1. Sensor バンドルを展開します。

```
$ 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 Sensor のデプロイを続行します。

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

1.2.3.1. Sensor コンテナーの再起動

コンテナーを強制終了するか、Sensor Pod を削除することで、Sensor 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、Sensor 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、Sensor 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 Sensor 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

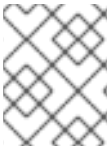
カスタム信頼済み証明書を Red Hat Cluster Security for Kubernetes に追加する方法を学びます。

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。

- Central 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。
- Central に使用するカスタム証明書を信頼する Sensor。

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、Central 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 Scanner と Sensor に伝播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

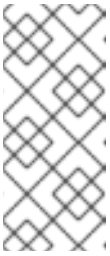
- 新規インストールを行う場合は、**ca-setup.sh** スクリプトが **central-bundle/central/scripts/ca-setup.sh** の **scripts** ディレクトリーにあります。
- OpenShift Container Platform クラスターにログインしたのと同じターミナルで **ca-setup.sh** スクリプトを実行する必要があります。

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>
```

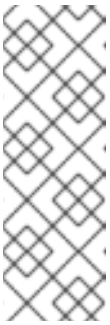


注記

- 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

- 拡張子が **.crt** または **.pem** の PEM エンコードされた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、Central 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、Central と Scanner の両方を再起動する必要があります。

2.2.1. Central コンテナーの再起動

Central コンテナーを強制終了するか、Central Pod を削除して、Central 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、Central 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、Central 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 Central Pod を削除します。

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

2.2.2. Scanner コンテナーの再起動

Pod を削除すると、Scanner コンテナーを再起動できます。

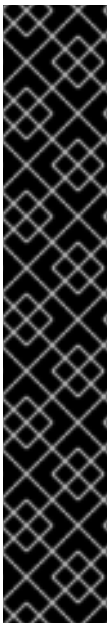
手順

- 次のコマンドを実行して Scanner 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、Central を設定すると、CA は、作成する新しい Sensor デプロイメントバンドルに含まれます。

- Central への接続中に既存の Sensor が問題を報告した場合は、Sensor 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 Sensor 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/
```

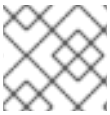
- Helm を使用して新しい Sensor をデプロイする場合は、追加のスクリプトを実行する必要はありません。

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証明書を再発行またはローテーションする必要があります。証明書の有効期限を表示するには、RHACS ポータルで **Platform Configuration → Clusters** を選択し、**Credential Expiration** 列を表示します。

3.1. CENTRAL の内部証明書の再発行

Central 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、Central インストールに固有のものです。Central 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。

Operator ベースのインストールの場合は、RHACS バージョン 4.3.4 以降、Operator により、すべての Central コンポーネントのサービス Transport Layer Security (TLS) 証明書が、有効期限が切れる 6 カ月前に自動的にローテーションされます。以下の条件が適用されます。

- シークレット内の証明書のローテーションによって、コンポーネントが証明書を自動的に再ロードすることはありません。ただし、リロードは通常、RHACS アップグレードの一部として、またはノードの再起動の結果として Pod が交換されるときに発生します。どちらのイベントも少なくとも 6 カ月ごとに発生しない場合は、古い (メモリー内) サービス証明書の有効期限が切れる前に Pod を再起動する必要があります。たとえば、**central**、**central-db**、**scanner**、または **scanner-db** のいずれかの値を含む **app** ラベルを持つ Pod を削除できます。
- CA 証明書は更新されません。有効期限は 5 年間です。
- セキュアクラスターのコンポーネントによって使用される init バンドル内のサービス証明書は更新されません。初期バンドルは定期的にローテーションする必要があります。

Operator ベースではないインストールの場合、TLS 証明書を手動でローテーションする必要があります。証明書を手動でローテーションする手順は、次のセクションに記載されています。

前提条件

- 証明書を再発行またはローテーションするには、**ServiceIdentity** リソースの **write** 権限が必要である。

手順

1. RHACS ポータルで、証明書の有効期限を通知するバナー内のリンクをクリックして、新しい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。シークレットには、証明書とキーの値が含まれます。
2. 次のコマンドを実行して、Central 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>
```

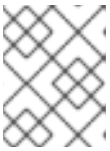
3. Central を再起動して、変更を適用します。

3.1.1. Central コンテナーの再起動

Central コンテナーを強制終了するか、Central Pod を削除して、Central 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、Central 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、Central 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 Central Pod を削除します。

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

3.2. SCANNER の内部証明書の再発行

Scanner には、Central との通信に使用する証明書が組み込まれています。

Scanner 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. 新しい YAML 設定ファイルを、Scanner をインストールしたクラスターに適用します。

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

3. Scanner を再起動して変更を適用します。

3.2.1. Scanner および Scanner DB コンテナーの再起動

Pod を削除すると、Scanner と Scanner DB コンテナーを再起動できます。

手順

- Scanner および Scanner 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. SENSOR、COLLECTOR、および ADMISSION コントローラーの内部証明書の再発行

Sensor、Collector、および Admission コントローラーは、証明書を使用して相互に通信し、Central と通信します。

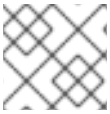
証明書を置き換えるには、以下のいずれかの方法を使用します。

- セキュアクラスターに init バンドルを作成、ダウンロード、インストールします。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。
- 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アクラスターの内部証明書の再発行

セキュアクラスターには、Collector、Sensor、および Admission Control コンポーネントが含まれています。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。

Central 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。

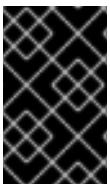


注記

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。複数のセキュアクラスターで同じバンドルを使用できます。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

- 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** をクリックする。

e. クラスター初期化バンドルの名前を入力し、**Generate** をクリックする。

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
```

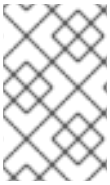
次のステップ

- セキュアクラスターごとに必要なリソースを作成するには、次のコマンドを実行します。

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

3.3.2. 自動アップグレードを使用したセキュアクラスターの内部証明書の再発行

自動アップグレードを使用して、Sensor、Collector、および Admission 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「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認証情報を作成します。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 サービスが新しい認証情報の使用を開始すると、通知は消えます。

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。

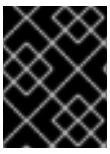
Red Hat Advanced Cluster Security for Kubernetes をオフラインモードでデプロイして操作するには:

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、セキュア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、オフラインで使用するためのイメージのダウンロード で説明されているように、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、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、Scanner 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. イメージのバージョン

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。最新バージョンには次のイメージが含まれています。

表5.1 Red Hat Advanced Cluster Security for Kubernetes のイメージ

Image	説明	現行バージョン

Image	説明	現行バージョン
Main	Central、Sensor、Admission コントローラー、および Compliance コンポーネントが含まれます。継続的インテグレーション (CI) システムで使用する **roxctl** も含まれます。	**registry.redhat.io/advanced-cluster-security/rhacs-main-rhel8:4.5.1**
Central DB	Central にデータベースストレージを提供する PostgreSQL インスタンス。	**registry.redhat.io/advanced-cluster-security/rhacs-central-db-rhel8:4.5.1**
Scanner	イメージおよびノードをスキャンします。	1. **registry.redhat.io/advanced-cluster-security/rhacs-scanner-rhel8:4.5.1** 2. **registry.redhat.io/advanced-cluster-security/rhacs-scanner-slim-rhel8:4.5.1**
Scanner DB	イメージのスキャン結果および脆弱性の定義を格納します。	**registry.redhat.io/advanced-cluster-security/rhacs-scanner-db-rhel8:4.5.1**
Scanner V4	イメージをスキャンします。	**registry.redhat.io/advanced-cluster-security/rhacs-scanner-v4-rhel8:4.5.1**
Scanner V4 DB	Scanner V4 のイメージスキャン結果と脆弱性定義を保存します。	**registry.redhat.io/advanced-cluster-security/rhacs-scanner-v4-db-rhel8:4.5.1**
Collector	Kubernetes または OpenShift Container Platform クラスターでランタイムアクティビティーを収集します。	1. **registry.redhat.io/advanced-cluster-security/rhacs-collector-rhel8:4.5.1** 2. **registry.redhat.io/advanced-cluster-security/rhacs-collector-slim-rhel8:4.5.1**

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-rhel8:4.5.1
<your_registry>/rhacs-main-rhel8:4.5.1
```

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-rhel8:4.5.1
<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. central-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 **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。

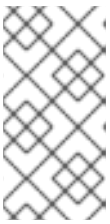
5.2.2. roxctl CLI を使用したオフラインモードの有効化

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-rhel8:4.5.1
```



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である **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-rhel8:4.5.1** になります。

```
Enter Scanner DB image to use (if unset, the default will be used): <your_registry>/rhacs-scanner-db-rhel8:4.5.1
```

```
Enter Scanner image to use (if unset, the default will be used): <your_registry>/rhacs-scanner-rhel8:4.5.1
```

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 Sensor 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 Collector 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 SCANNER 定義の更新

Scanner は脆弱性のデータベースを維持します。Red Hat Advanced Cluster Security for Kubernetes (RHACS) が通常のモードで実行されると、Central がインターネットから最新の脆弱性データを取得し、Scanner が Central から脆弱性データを取得します。

しかし、RHACS をオフラインモードで使用している場合は、脆弱性データを手動で更新する必要があります。脆弱性データを手動で更新するには、定義ファイルを Central にアップロードする必要があります。その後、Scanner が Central から脆弱性データを取得します。

Scanner は、オンラインモードとオフラインモードの両方で、デフォルトで 5 分ごとに Central からの新しいデータをチェックします。オンラインモードでは、Central も約 5 - 20 分ごとにインターネットからの新しいデータをチェックします。

オフラインデータソースは約 3 時間ごとに更新されます。データが Central にアップロードされると、Scanner はデータをダウンロードし、ローカルの脆弱性データベースを更新します。

オフラインモードで定義を更新するには、次の手順を実行します。

1. 定義をダウンロードします。
2. 定義を Central にアップロードします。

5.3.1. Scanner 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、Scanner が使用する脆弱性定義データベースをダウンロードし、Central にアップロードできます。

前提条件

- Scanner 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す。

手順

- 定義をダウンロードするには、次のいずれかの操作を実行します。
  - 推奨: RHACS バージョン 4.4 以降では、**roxctl scanner download-db --scanner-db-file scanner-vuln-updates.zip** コマンドを使用して定義をダウンロードします。
  - 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。

関連情報

- roxctl scanner download-db

5.3.2. Central への定義のアップロード

Scanner 定義を Central 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を使用します。Red Hat は実稼働環境では認証トークンを使用することを推奨します。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です。

5.3.2.1. API トークンを使用して Central に定義をアップロードする

API トークンを使用して、Scanner が使用する脆弱性定義データベースを Central 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 Central 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、Scanner が使用する脆弱性定義データベースを Central 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

注記

サポートパッケージは非推奨であり、バージョン 4.5 以降を実行しているセキュアクラスターには影響しません。サポートパッケージのアップロードは、バージョン 4.4 以前のセキュアクラスターにのみ影響します。

Collector は、セキュア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監視するには、Collector に eBPF プログラムの形式のプローブが必要です。

CORE_BPF 収集方法を使用した場合、プローブがカーネルバージョンに固有のものではないため、基礎となるカーネルの更新後もプローブを引き続き使用できます。この収集方法では、サポートパッケージを提供または更新する必要がありません。

代わりに、収集方法 **EBPF** を使用した場合、プローブが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ものになります。Collector イメージには、リリース時のサポート対象カーネル用のビルトインプローブセットが含まれています。ただし、それ以降のカーネルでは、新しいプローブが必要になります。

Red Hat Advanced Cluster Security for Kubernetes が通常モードで実行されている (インターネットに接続されている) 場合、必要なプローブ組み込まれていないと、Collector が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、Central にアップロードできます。次に、Collector はこれらのプローブを Central からダウンロードします。

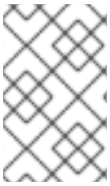
Collector は、次の順序で新しいプローブを確認します。確認の対象は次のとおりです。

1. 既存の Collector イメージ。
2. カーネルサポートパッケージ (Central 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。Collector は、Central のネットワーク接続を使用してプローブを確認し、ダウンロードします。

Collector は確認後に新しいプローブを取得しなかった場合、**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、Central にアップロードできます。その後、Collector がこれらのプローブを Central 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



注記

サポートパッケージは非推奨であり、バージョン 4.5 以降を実行しているセキュアクラスターには影響しません。サポートパッケージのアップロードは、バージョン 4.4 以前のセキュアクラスターにのみ影響します。

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、Central 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 Central 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 Central 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

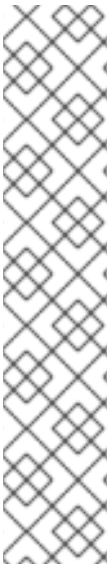
1. **ROX_API_TOKEN** および **ROX_CENTRAL_ADDRESS** 環境変数を設定します。

```
$ 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 Central 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。
- Central 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、Central 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。Central 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

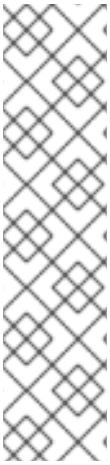
Red Hat Advanced Cluster Security for Kubernetes アラートの保持期間を設定する方法を学びます。

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

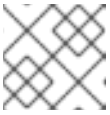
RHACS ポータルを使用することで、アラート保持の設定を行うことができます。

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

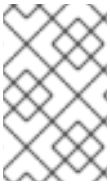
1. RHACS ポータルで、**Platform Configuration → System Configuration** に移動します。
2. **System Configuration** ビューのヘッダーで、**Edit** をクリックします。
3. **Data Retention Configuration** セクションで、各タイプのデータの日数を更新します。
   - **すべてのランタイム違反**
   - **解決されたデプロイフェーズ違反**
   - **削除されたデプロイメントのランタイム違反**
   - **デプロイされなくなったイメージ**



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。



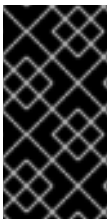
注記

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 Central 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、Central 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、Central 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 Central 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 セキュアクラスターの自動アップグレードの設定

各セキュア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により、各セキュアクラスターをアップグレードする手動タスクが自動化され、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、Central 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 Sensor、Collector、および Compliance 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ア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 Sensor 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、セキュア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 Sensor イメージと Collector 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。
- Sensor 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、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。このアカウントを削除すると、Sensor に十分な権限がないため、今後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

すべてのセキュア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 Collector 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

セキュア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リスト表示されます。

アップグレードステータス	説明
Central バージョンで最新	セキュアクラスターが Central 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	Sensor と Collector 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	Sensor と Collector 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。
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、セキュアクラスターのステータスメッセージが **Upgrade failed. Retry upgrade** に変わります。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** ビューでセキュアクラスターの行を確認します。

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、sensor-upgrader 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。Red Hat Advanced Cluster Security for Kubernete 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、セキュア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ア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ない場合は、**Clusters** ビューを使用して、セキュアクラスターのアップグレードを管理できます。

セキュア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、以下を実行します。

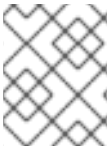
手順

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. **Upgrade** をクリックします。

第9章 RHACS からの非アクティブなクラスターの自動削除の設定

Red Hat Advanced Cluster Security for Kubernetes (RHACS) には、アクティブなクラスターのみをモニターできるように、非アクティブなクラスターを自動的に削除するようにシステムを設定するオプションが用意されています。インストールされ、Central とのハンドシェイクを少なくとも 1 回実行したクラスターのみが最初にモニターされることに注意してください。この機能が有効になっている場合、Central がクラスター内の Sensor に到達できなかった時間が **Decomissioned cluster age** フィールドで設定された期間に達した場合、クラスターは RHACS で非アクティブであると見なされます。その後、Central は非アクティブなクラスターをモニターしなくなります。**Platform Configuration → System Configuration** ページで、**Decommissioned cluster age** フィールドを設定できます。この機能を設定するときに、クラスターのラベルを追加して、クラスターが非アクティブになった場合でも RHACS がクラスターをモニターし続けるようにすることができます。

RHACS からの非アクティブなクラスターの削除は、デフォルトで無効になっています。この設定を有効にするには、次の手順で説明するように、**Decommissioned cluster age** フィールドにゼロ以外の数値を入力します。**Decommissioned cluster age** フィールドは、クラスターが非アクティブであると見なされるまでに到達不能な状態を維持できる日数を示します。クラスターが非アクティブの場合は、**Clusters** ページにクラスターのステータスが表示されます。非アクティブなクラスターは **unhealthy** ラベルで示され、ウィンドウには、非アクティブな状態が続く場合にクラスターが RHACS から削除されるまでの日数が表示されます。クラスターが RHACS から削除された後、そのアクションは Central ログの **info** ログに記録されます。



注記

この設定を有効にしてからクラスターが削除されるまで、24 時間の猶予期間があります。Central をホストするクラスターが削除されることはありません。

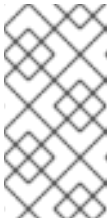
9.1. クラスター廃止の設定

非アクティブなクラスターを RHACS から自動的に削除するように RHACS を設定できます。非アクティブなクラスターは、インストールされ、Central とのハンドシェイクを少なくとも 1 回実行したにもかかわらず、指定された期間、Sensor から到達できなかったクラスターです。クラスターにラベルを付けて、アクセスできないときに削除されないようにすることもできます。

手順

1. RHACS ポータルで、**Platform Configuration → System Configuration** に移動します。
2. **System Configuration** ヘッダーで、**Edit** をクリックします。
3. **Cluster deletion** セクションでは、次のフィールドを設定できます。
   - **Decommissioned cluster age**: RHACS からの削除が検討される前にクラスターに到達できない日数。この日数の間、Central がクラスター上の Sensor にアクセスできない場合は、クラスターとそのすべてのリソースが削除されます。この機能を無効のままにする (デフォルトの動作) には、このフィールドに **0** を入力します。この機能を有効にするには、**90** などのゼロ以外の数値を入力して、到達不能日数を設定します。
   - **Ignore clusters which have labels**: クラスターが削除されないようにするために、このセクションにキーおよび値を入力してラベルを設定できます。このラベルが付いたクラスターは、**Decommissioned cluster age** フィールドで設定された日数の間到達できなかったとしても、削除されません。
     - **Key**: クラスターに使用するラベルを入力します。

- **値**: キーに関連付けられた値を入力します。
  たとえば、実稼働クラスターが削除されないよう保持するために、**cluster-type** のキーと **production** の値を設定できます。



注記

Cluster deletion セクションで、Clusters which have Sensor Status: Unhealthy クリックして、Clusters リストページに移動します。このページはフィルタリングされ、削除の対象となる非アクティブなクラスターと、RHACS からの削除の時間枠が表示されます。

4. Save をクリックします。

注記

API を使用してこのオプションを表示および設定するには、**/v1/config** および **/v1/config/private** エンドポイントの要求ペイロードで **decommissionedClusterRetention** 設定を使用します。詳細は、RHACS ポータルで **Help → API reference** に移動して、**ConfigService** オブジェクトの API ドキュメントを参照してください。

9.2. 非アクティブなクラスターの表示

非アクティブなクラスターは、インストールされ、少なくとも 1 回は Central とのハンドシェイクを実行したが、指定された期間、Sensor から到達できなかったクラスターです。この手順を使用して、これらのクラスターのリストを表示します。

手順

1. RHACS ポータルで、**Platform Configuration → System Configuration** に移動します。
2. **Cluster deletion** セクションで、**Clusters which have Sensor Status: Unhealthy**クリックして、**Clusters** リストページに移動します。このページはフィルター処理され、RHACS からの削除の対象となる非アクティブなクラスターと、削除の時間枠が表示されます。

注記

クラスターが非アクティブであると見なされた後にこの機能が有効になっている場合、削除までの日数のカウントは、機能が有効になった時点からではなく、クラスターが非アクティブになった時点から開始されます。削除したくない非アクティブなクラスターがある場合は、「クラスターの廃止の設定」セクションの説明に従ってラベルを設定できます。これらのラベルを持つクラスターは、システムが非アクティブなクラスターを削除するときに無視されます。

第10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:

- Central および Scanner 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。
- Central と Scanner 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、セキュアクラスターで実行されている Collector が実行時に追加の eBPF プローブをダウンロードする必要がある場合:
  - Collector は Sensor に接続してダウンロードを試みます。
  - 次に、Sensor はこのリクエストを Central に転送します。
  - Central 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。

10.1. 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。

注記

OpenShift Container Platform クラスターにグローバルプロキシーを設定している場合、Operator Lifecycle Manager (OLM) はクラスター全体のプロキシーで管理する Operator を自動的に設定します。ただし、インストールされた Operator をグローバルプロキシーを上書きするか、カスタム認証局 (CA) 証明書を挿入するように設定することもできます。

詳細は、Operator Lifecycle Manager でのプロキシーサポートの設定 を参照してください。

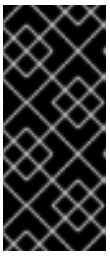
手順

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 Central と Scanner 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、Central Pod と Scanner Pod を再起動する必要があります。

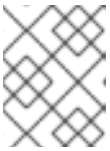
10.2. インストール中にプロキシーを設定する

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、Central と Scanner 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11章 診断バンドルの生成

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、診断バンドルを生成して、そのデータを送信してください。

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お願い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 に転送するときは、必ず暗号化されたチャネルを使用してください。

11.1. 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。

- Central 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: 各セキュア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、各セキュアクラスターに関する詳細。

11.2. 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

前提条件

- 診断バンドルを生成するために、**DebugLogs** リソースの **read** 権限がある。

手順

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

11.3. ROXCTL CLI を使用した診断バンドルの生成

roxctl CLI を使用して、Red Hat Advanced Cluster Security for Kubernetes (RHACS) の管理者パスワードまたは API トークンと中央アドレスを含む診断バンドルを生成できます。

前提条件

- 診断バンドルを生成するために、**Administration** リソースの **read** パーミッションを用意する。これは、バージョン 3.73.0 よりも古い **DebugLogs** リソースのバージョンで必要です。
- RHACS 管理者パスワード、API トークン、および中央アドレスを設定している。

手順

- RHACS 管理者パスワードを使用して診断バンドルを生成するには、以下の手順を実行します。
  1. 以下のコマンドを実行して、環境変数 **ROX_PASSWORD** および **ROX_CENTRAL_ADDRESS** を設定します。

     ```
     $ export ROX_PASSWORD=<rox_password> && export
     ROX_CENTRAL_ADDRESS=<address>:<port_number>
     ```
     (1)

     (1) **<rox_password>** には、RHACS 管理者パスワードを指定します。

  2. 次のコマンドを実行して、RHACS 管理者パスワードを使用して診断バンドルを生成します。

     ```
     $ roxctl -e "$ROX_CENTRAL_ADDRESS" -p "$ROX_PASSWORD" central debug
     download-diagnostics
     ```

- API トークンを使用して診断バンドルを生成するには、以下の手順を実行します。
  1. 以下のコマンドを実行して **ROX_API_TOKEN** 環境変数を設定します。

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

  2. 以下のコマンドを実行して API トークンを使用して診断バンドルを生成します。

```
$ roxctl -e "$ROX_CENTRAL_ADDRESS" central debug download-diagnostics
```

第12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。

12.1. 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 Central 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、Central 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。Red Hat は、エンドポイントで HTTP プロトコルと gRPC 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推奨します。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 Central 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、Central 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。

12.2. 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、Central 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。

12.3. 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. Central を再起動します。

注記

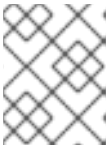
デフォルトのポート **8443** 以外の追加のポートを公開する場合は、それらの追加のポートでのトラフィックを許可するネットワークポリシーを作成する必要があります。

12.3.1. Central コンテナーの再起動

Central コンテナーを強制終了するか、Central Pod を削除して、Central 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、Central 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、Central 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 Central Pod を削除します。

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

12.4. 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推奨します。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。

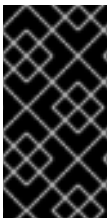
第13章 RHACS のモニタリング

Red Hat OpenShift の組み込みモニタリングを使用するか、カスタムの Prometheus モニタリングを使用して、Red Hat Advanced Cluster Security for Kubernetes (RHACS) を監視できます。

Red Hat OpenShift で RHACS を使用する場合、OpenShift Container Platform には、コアプラットフォームコンポーネントの監視を提供する、事前に設定およびインストールされた自己更新型のモニタリングスタックが組み込まれています。RHACS は、暗号化および認証されたエンドポイントを介して Red Hat OpenShift モニタリングにメトリクスを公開します。

13.1. RED HAT OPENSHIFT を使用したモニタリング

Red Hat OpenShift を使用したモニタリングはデフォルトで有効になっています。このデフォルトの動作には設定は必要ありません。



重要

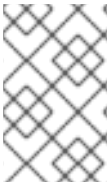
以前に Prometheus Operator でモニタリングを設定している場合は、カスタムの **ServiceMonitor** リソースを削除することを検討してください。RHACS には、Red Hat OpenShift のモニタリング用に事前設定された **ServiceMonitor** が付属しています。複数の **ServiceMonitor** を使用すると、スクレイピングが重複する可能性があります。

Red Hat OpenShift を使用したモニタリングは、Scanner ではサポートされていません。Scanner を監視する場合は、まずデフォルトの Red Hat OpenShift モニタリングを無効にする必要があります。次に、カスタム Prometheus モニタリングを設定します。

Red Hat OpenShift モニタリングの無効化の詳細は、「RHACS Operator を使用した Central サービスの Red Hat OpenShift モニタリングの無効化」または「Helm を使用した Central サービスの Red Hat OpenShift モニタリングの無効化」を参照してください。Prometheus の設定に関する詳細は、「カスタム Prometheus のモニタリング」を参照してください。

13.2. カスタム PROMETHEUS のモニタリング

Prometheus は、オープンソースのモニタ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、RHACS の Central コンポーネントと Sensor コンポーネントの正常性と可用性を監視できます。モニタリングを有効にすると、RHACS はポート番号 9090 に新しいモニタリングサービスを作成し、そのポートへの受信接続を許可するネットワークポリシーを作成します。



注記

このモニタリングサービスは、TLS によって暗号化されず、承認のないエンドポイントを公開します。これは、Red Hat OpenShift モニタリングを使用しない場合にのみ使用してください。

Red Hat OpenShift を使用している場合は、カスタムの Prometheus モニタリングを使用する前に、デフォルトのモニタリングを無効にする必要があります。Kubernetes を使用している場合は、この手順を実行する必要はありません。

13.2.1. RHACS Operator を使用した Central サービスの Red Hat OpenShift モニタリングの無効化

Operator を使用してデフォルトのモニタリングを無効にするには、次の例に示すように、**Central** カスタムリソースの設定を変更します。設定オプションの詳細は、「関連情報」セクションの「Operator を使用した Central 設定オプション」を参照してください。

手順

1. OpenShift Container Platform Web コンソールで、**Operators → Installed Operators** ページに移動します。
2. インストールされている Operator の一覧から RHACS Operator を選択します。
3. **Central** タブをクリックします。
4. Central インスタンスのリストから、監視を有効にする Central インスタンスをクリックします。
5. **YAML** タブをクリックし、次の例に示すように YAML 設定を更新します。

```
monitoring:
  openshift:
    enabled: false
```

13.2.2. Helm を使用した Central サービスの Red Hat OpenShift モニタリングの無効化

Helm を使用してデフォルトのモニタリングを無効にするには、**central-services** Helm チャートの設定オプションを変更します。設定オプションの詳細は、「関連情報」セクションのドキュメントを参照してください。

手順

1. 以下の値で設定ファイルを更新します。

```
monitoring.openshift.enabled: false
```

2. **helm upgrade** コマンドを実行し、設定ファイルを指定します。

13.2.3. RHACS Operator を使用した Central サービスの監視

Central カスタムリソースの設定を変更することで、Central サービス、Central および Scanner を監視できます。設定オプションの詳細は、「関連情報」セクションの「Operator を使用した Central 設定オプション」を参照してください。

手順

1. OpenShift Container Platform Web コンソールで、**Operators → Installed Operators** ページに移動します。
2. インストールされている Operator のリストから、Red Hat Advanced Cluster Security for Kubernetes Operator を選択します。
3. **Central** タブをクリックします。
4. Central インスタンスのリストから、監視を有効にする Central インスタンスをクリックします。

5. **YAML** タブをクリックし、YAML 設定を更新します。
   - Central を監視するには、**Central** カスタムリソースの **central.monitoring.exposeEndpoint** 設定オプションを有効にします。
   - Scanner を監視するには、**Central** カスタムリソースの **scanner.monitoring.exposeEndpoint** 設定オプションを有効にします。
6. **Save** をクリックします。

13.3. HELM を使用した CENTRAL サービスの監視

central-services の Helm チャートの設定オプションを変更することで、Central サービス (Central と Scanner) を監視できます。詳細は、「関連情報」セクションの「central-services Helm チャートをデプロイした後の設定オプションの変更」を参照してください。

手順

1. 次の値を使用して、**values-public.yaml** 設定ファイルを更新します。

```
central.exposeMonitoring: true
scanner.exposeMonitoring: true
```

2. **helm upgrade** コマンドを実行し、設定ファイルを指定します。

13.3.1. Prometheus サービスモニターを使用した Central の監視

Prometheus Operator を使用している場合は、サービスモニターを使用して Red Hat Advanced Cluster Security for Kubernetes (RHACS) からメトリクスを取得できます。

注記

Prometheus オペレーターを使用していない場合は、RHACS からデータを受信するように Prometheus 設定ファイルを編集する必要があります。

手順

1. 次の内容を含む新しい **servicemonitor.yaml** ファイルを作成します。

```
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  name: prometheus-stackrox
  namespace: stackrox
spec:
  endpoints:
    - interval: 30s
      port: monitoring
      scheme: http
  selector:
    matchLabels:
      app.kubernetes.io/name: <stackrox-service> 1
```

1 ラベルは、監視する **Service** リソースと一致する必要があります。たとえば、**central** または **scanner** です。

2. YAML をクラスターに適用します。

```
$ oc apply -f servicemonitor.yaml 1
```

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

検証

- 次のコマンドを実行して、サービスモニターのステータスを確認します。

```
$ oc get servicemonitor --namespace stackrox 1
```

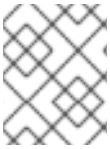
1 Kubernetes を使用する場合は、**oc** の代わりに **kubectl** を入力します。

13.4. 関連情報

- Operator を使用した Central 設定オプション
- central-services Helm チャートをデプロイした後の設定オプションの変更
- Helm ドキュメント

第14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。

14.1. 監査ログの有効化

監査ログを有効にすると、変更があるたびに、Red Hat Advanced Cluster Security for Kubernetes 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。

14.2. 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",
        "source": {
          "requestAddr": "10.131.0.7:58276",
          "xForwardedFor": "8.8.8.8",
        },
        "sourceIp": "8.8.8.8",
        "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"
      }
    }
  }
}
```


リクエストの送信元 IP アドレスがソースパラメーターに表示されるため、監査ログリクエストを調査し、その送信元を特定することが容易になります。

リクエストの送信元 IP アドレスを決定するために、RHACS は次のパラメーターを使用します。

- **xForwardedFor**: X-Forwarded-For ヘッダー。

- **requestAddr**: リモートアドレスヘッダー。
- **sourceIp**: HTTP リクエストの IP アドレス。

重要

ソース IP アドレスの判別は、Central を外部に公開する方法によって異なります。次のオプションを検討できます。

- Kubernetes 外部ロードバランサーサービスタイプを使用して Google Kubernetes Engine (GKE) または Amazon Elastic Kubernetes Service (Amazon EKS) で Central を実行している場合など、Central をロードバランサーの背後で公開する場合は、クライアントソース IP の保持 を参照してください。
- X-Forwarded-For ヘッダー を使用してリクエストを転送する Ingress コントローラーの背後で Central を公開する場合、設定を変更する必要はありません。
- TLS パススルールートを使用して Central を公開する場合、クライアントの送信元 IP アドレスを特定できません。クラスター内部 IP アドレスは、クライアントの送信元 IP アドレスとして送信元パラメーターに表示されます。

第15章 API トークンの設定

Red Hat Advanced Cluster Security for Kubernetes (RHACS) では、一部のシステム統合、認証プロセス、およびシステム機能に API トークンが必要です。RHACS Web インターフェイスを使用してトークンを設定できます。

15.1. API トークンの作成

手順

1. RHACS ポータルで、**Platform Configuration → Integrations** に移動します。
2. **Authentication Tokens** カテゴリーまでスクロールし、**API Token** をクリックします。
3. **Generate Token** をクリックします。
4. トークンの名前を入力し、必要なレベルのアクセスを提供するロールを選択します (たとえば、**Continuous Integration** または **Sensor Creator**)。
5. **Generate** をクリックします。

重要

生成されたトークンをコピーして安全に保存します。再度表示することはできません。

15.2. API トークンの有効期限について

API トークンの有効期限は作成日から 1 年です。RHACS は、トークンの有効期限が 1 週間以内に切れる場合、Web インターフェイスで、またログメッセージを Central に送信することで警告します。ログメッセージプロセスは 1 時間に 1 回実行されます。このプロセスは 1 日に 1 回、期限切れになるトークンをリストし、それぞれのトークンに対してログメッセージを作成します。ログメッセージは 1 日に 1 回発行され、Central ログに表示されます。

ログの形式は次の例に示すとおりです。

```
Warn: API Token [token name] (ID [token ID]) will expire in less than X days.
```

次の表に示す環境変数を設定することで、ログメッセージプロセスのデフォルト設定を変更できます。

環境変数	デフォルト値	説明
ROX_TOKEN_EXPIRATION_NOTIFIER_INTERVAL	1h (1 時間)	トークンをリストし、ログを作成するログメッセージバックグラウンドループが実行される頻度。
ROX_TOKEN_EXPIRATION_NOTIFIER_BACKOFF_INTERVAL	24h (1 日)	ループがトークンをリストし、通知を発行する頻度。
ROX_TOKEN_EXPIRATION_DETECTION_WINDOW	168h (1 週間)	通知が生成されるトークンの有効期限が切れるまでの期間。

第16章 宣言型設定の使用

宣言型設定を使用すると、設定をリポジトリー内のファイルに保存して更新し、システムに適用できます。宣言型設定は、GitOps ワークフローを使用している場合などに便利です。現在、Red Hat Advanced Cluster Security for Kubernetes (RHACS) では、認証プロバイダー、ロール、権限セット、アクセススコープなど、認証および認可リソースに対して宣言型設定を使用できます。

宣言型設定を使用するには、認証および認可リソースに関する設定情報を含む YAML ファイルを作成します。これらのファイルまたは設定は、Central インストール中にマウントポイントを使用して RHACS に追加されます。RHACS のインストール時にマウントポイントを設定する方法の詳細は、「関連情報」セクションのインストールドキュメントを参照してください。

宣言型設定で使用される設定ファイルは、リソースのタイプに応じて、config map またはシークレットに保存されます。セキュリティーを強化するために、認証プロバイダーの設定をシークレットに保存します。他の設定を config map に保存できます。

単一の config map またはシークレットには、複数のリソースタイプの複数の設定を含めることができます。これにより、Central インスタンスのボリュームマウントの数を制限できます。

16.1. 宣言的設定から作成されたリソースの制限事項

リソースは他のリソースを参照できるため (たとえば、ロールは権限セットとアクセススコープの両方を参照できます)、参照には次の制限が適用されます。

- 宣言的設定は、同様に宣言的に作成されたリソース、またはシステム RHACS リソースのみを参照できます。たとえば、**Admin** または **Analyst** システムロールや権限セットなどのリソースです。
- リソース間のすべての参照では、リソースを識別するために名前が使用されます。したがって、同じリソースタイプ内のすべての名前は一意である必要があります。
- 宣言的設定から作成されたリソースは、宣言的設定ファイルを変更することによってのみ変更または削除できます。RHACS ポータルまたは API を使用してこれらのリソースを変更することはできません。

16.2. 宣言型設定の作成

roxctl を使用して、設定を保存する YAML ファイルを作成し、そのファイルから config map を作成し、config map を適用します。

前提条件

- Central のインストール中に、config map またはシークレットのマウントを追加しました。この例では、config map は "declarative-configs" と呼ばれます。詳細は、「関連情報」セクションに記載されているインストールドキュメントを参照してください。

手順

1. 次のコマンドを入力して権限セットを作成します。この例では、"restricted" という名前のアクセス許可セットを作成し、**permission-set.yaml** ファイルとして保存します。これは、**Administration** リソースの読み取りおよび書き込みアクセスと、アクセスリソースへの読み取りアクセスを設定します。

```
$ roxctl declarative-config create permission-set \
```

```
--name="restricted" \
--description="Restriction permission set that only allows \
access to Administration and Access resources" \
--resource-with-access=Administration=READ_WRITE_ACCESS \
--resource-with-access=Access=READ_ACCESS > permission-set.yaml
```

2. 次のコマンドを入力して、**Administration** と **Access** リソースへのアクセスを許可するロールを作成します。この例では、"restricted" という名前のロールを作成し、**role.yaml** ファイルとして保存します。

```
$ roxctl declarative-config create role \
--name="restricted" \
--description="Restricted role that only allows access to Administration and Access" \
--permission-set="restricted" \
--access-scope="Unrestricted" > role.yaml
```

3. 次のコマンドを入力して、前の手順で作成した 2 つの YAML ファイルから config map を作成します。この例では、**declarative-configurations** config map を作成します。

```
$ kubectl create configmap declarative-configurations \ 1
--from-file permission-set.yaml --from-file role.yaml \
-o yaml --namespace=stackrox > declarative-configs.yaml
```

1 OpenShift Container Platform の場合は、**oc create** を使用します。

4. 次のコマンドを入力して、config map を適用します。

```
$ kubectl apply -f declarative-configs.yaml 1
```

1 OpenShift Container Platform の場合は、**oc apply** を使用します。

config map を適用すると、Central から抽出された設定情報によってリソースが作成されます。

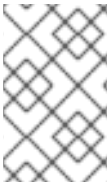
注記

次の段落で説明するように、監視間隔は 5 秒ですが、config map から中央マウントへの変更の伝播に遅延が発生する可能性があります。

次の間隔を設定して、宣言的設定が Central とどのように対話するかを指定できます。

- 設定監視間隔: Central が変更をチェックする間隔は 5 秒ごとです。この間隔は、**ROX_DECLAATIVE_CONFIG_WATCH_INTERVAL** 環境変数を使用して設定できます。
- 調整間隔: デフォルトでは、Central との宣言的な設定調整は 20 秒ごとに行われます。この間隔は、**ROX_DECLARATIVE_CONFIG_RECONCILE_INTERVAL** 環境変数を使用して設定できます。

宣言型設定を使用して認証および認可リソースを作成した後、RHACS Web ポータルの **アクセス制御** ページでそれらを表示できます。リソースが宣言的設定を使用して作成された場合、**Origin** フィールドは **Declarative** を示します。



注記

RHACS Web ポータルの宣言的設定から作成されたリソースを編集することはできません。これらのリソースに変更を加えるには、設定ファイルを直接編集する必要があります。

宣言的設定のステータスを表示するには、**Platform Configuration → System Health** に移動し、**Declarative configuration** セクションまでスクロールします。

16.3. 宣言的な設定例

次の例をガイドとして使用して、宣言型設定を作成できます。**roxctl declarative-config lint** コマンドを使用して、設定が有効であることを確認します。

16.3.1. 宣言型設定認証プロバイダーの例

宣言型設定認証プロバイダーの例

```
name: A sample auth provider
minimumRole: Analyst (1)
uiEndpoint: central.custom-domain.com:443 (2)
extraUIEndpoints: (3)
  - central-alt.custom-domain.com:443
groups: (4)
  - key: email (5)
    value: example@example.com
    role: Admin (6)
  - key: groups
    value: reviewers
    role: Analyst
requiredAttributes: (7)
  - key: org_id
    value: "12345"
claimMappings: (8)
  - path: org_id
    value: my_org_id
oidc: (9)
  issuer: sample.issuer.com (10)
  mode: auto (11)
  clientID: CLIENT_ID
  clientSecret: CLIENT_SECRET
clientSecret: CLIENT_SECRET
iap: (12)
  audience: audience
saml: (13)
  spIssuer: sample.issuer.com
  metadataURL: sample.provider.com/metadata
saml: (14)
  spIssuer: sample.issuer.com
  cert: | (15)
  ssoURL: saml.provider.com
  idpIssuer: idp.issuer.com
```

```
userpki:
  certificateAuthorities: | 16
  certificate 17
openshift: 18
  enable: true
```

1. ログインしているユーザーにデフォルトで割り当てられる最小限のロールを指定します。空白のままにすると、値は **None** になります。
2. Central インスタンスのユーザーインターフェイスエンドポイントを使用します。
3. Central インスタンスが別のエンドポイントに公開されている場合は、ここで指定します。
4. これらのフィールドは、属性に基づいてユーザーを特定のロールにマップします。
5. キーには、認証プロバイダーから返された任意のクレームを指定できます。
6. ユーザーに与えられるロールを識別します。デフォルトのロールまたは宣言的に作成されたロールを使用できます。
7. オプション: 認証プロバイダーから返された属性が必要な場合は、これらのフィールドを使用します。たとえば、対象者が特定の組織やグループに限定されている場合などです。
8. オプション: アイデンティティープロバイダーから返されたクレームをカスタムクレームにマップする必要がある場合は、これらのフィールドを使用します。
9. このセクションは、OpenID Connect (OIDC) 認証プロバイダーの場合にのみ必要です。
10. トークンの予想される発行者を識別します。
11. OIDC コールバックモードを識別します。可能な値は、**auto**、**post**、**query**、および **fragment** です。推奨される値は **auto** です。
12. このセクションは、Google Identity-Aware Proxy (IAP) 認証プロバイダーの場合にのみ必要です。
13. このセクションは、Security Assertion Markup Language (SAML) 2.0 動的設定認証プロバイダーの場合にのみ必要です。
14. このセクションは、SAML 2.0 静的設定認証プロバイダーの場合にのみ必要です。
15. 証明書を Privacy Enhanced Mail (PEM) 形式で含めます。
16. このセクションは、ユーザー証明書による認証の場合にのみ必要です。
17. PEM 形式の証明書を含めます。
18. このセクションは、OpenShift Auth 認証プロバイダーにのみ必要です。

16.3.2. 宣言的な設定権限セットの例

宣言的な設定権限セットの例

```
name: A sample permission set
description: A sample permission set created declaratively
resources:
```

```
- resource: Integration (1)
  access: READ_ACCESS (2)
- resource: Administration
  access: READ_WRITE_ACCESS
```

(1) サポートされているリソースの完全なリストについては、**Access Control → Permission Sets** に移動してください。

(2) アクセスは **READ_ACCESS** または **READ_WRITE_ACCESS** のいずれかになります。

16.3.3. 宣言型設定のアクセス範囲の例

宣言型設定のアクセス範囲の例

```
name: A sample access scope
description: A sample access scope created declaratively
rules:
    included:
        - cluster: secured-cluster-A (1)
          namespaces:
            - namespaceA
        - cluster: secured-cluster-B (2)
    clusterLabelSelectors:
        - requirements:
            key: kubernetes.io/metadata.name
            operator: IN (3)
            values:
            - production
            - staging
            - environment
```

(1) アクセス範囲内に特定の namespace のみが含まれるクラスターを識別します。

(2) すべての namespace がアクセススコープに含まれるクラスターを識別します。

(3) ラベルの選択に使用する Operator を特定します。有効な値は、**IN**、**NOT_IN**、**EXISTS**、および **NOT_EXISTS** です。

16.3.4. 宣言型設定ロールの例

宣言型設定ロールの例

```
name: A sample role
description: A sample role created declaratively
permissionSet: A sample permission set (1)
accessScope: Unrestricted (2)
```

(1) パーミッションセットの名前。システムパーミッションセットのいずれか、または宣言で作成されたパーミッションセットのいずれかになります。

(2)

アクセススコープの名前。システムアクセススコープの 1 つまたは宣言で作成されたアクセススコープのいずれかです。

16.4. 宣言型設定のトラブルシューティング

Platform Configuration → System Health ページの **Declarative configuration** セクションに表示されるエラーメッセージを使用すると、トラブルシューティングに役立ちます。**roxctl declarative-config** コマンドには、設定ファイルを検証し、エラーを検出するための **lint** オプションも含まれています。

Platform Configuration → System Health ページの **Declarative configuration** セクションに表示されるエラーメッセージは、宣言型設定の問題に関する情報を提供します。宣言型設定に関する問題は、以下の条件によって発生する可能性があります。

- 設定ファイルの形式が有効な YAML ではありません。
- 設定ファイルには、権限セット内の無効なアクセスなど、無効な値が含まれています。
- リソース名が一意でない、設定にリソースへの無効な参照が含まれているなど、無効なストレージ制約が存在します。

設定ファイルを検証するには、設定ファイルのエラーを確認し、設定ファイルの作成および更新時に無効なストレージ制約がないことを確認するには、**roxctl declarative-config lint** コマンドを使用します。

削除中のストレージ制約のトラブルシューティングを行うには、リソースが **Declarative Orphaned** としてマークされているかどうかを確認します。これは、リソースによって参照された宣言的設定が削除されたことを示します (たとえば、ロールによって参照された権限セットの宣言的設定が削除された場合)。このエラーを修正するには、新しい権限セットを指すようにリソースを編集するか、削除された宣言的設定を復元します。

16.5. 関連情報

- カスタマイズされた Helm チャートを使用して Central をインストールする (Red Hat OpenShift)
- カスタマイズされた Helm チャートを使用して Central をインストールする (他の Kubernetes プラットフォーム)

第17章 RHACS インスタンスへのユーザーの招待

ユーザーを Red Hat Advanced Cluster Security for Kubernetes (RHACS) に招待することで、クラスター内で適切なユーザーに適切なアクセス権を確実に割り当てることができます。ロールを割り当て、認証プロバイダーを定義することにより、1人以上のユーザーを招待できます。

17.1. アクセス制御の設定と招待の送信

RHACS ポータルでアクセス制御を設定すると、RHACS インスタンスにユーザーを招待できます。

手順

1. RHACS ポータルで、**Platform Configuration → Access Control → Auth providers**タブに移動し、**Invite users** をクリックします。
2. **Invite users** ダイアログボックスで、次の情報を入力します。
   - **Emails to invite**: 招待するユーザーのメールアドレスを1つ以上入力します。対象の受信者に関連付けられた有効なメールアドレスであることを確認してください。
   - **Provider**: ドロップダウンリストから、招待された各ユーザーに使用するプロバイダーを選択します。

     重要

     - 利用可能な認証プロバイダーが1つしかない場合は、デフォルトで選択されます。
     - 複数の認証プロバイダーが使用可能で、そのうちの少なくとも1つが **Red Hat SSO** または **Default Internal SSO** である場合、そのプロバイダーがデフォルトで選択されます。
     - 複数の認証プロバイダーが使用可能であるが、そのどれも **Red Hat SSO** または **Default Internal SSO** ではない場合は、手動で1つを選択するように求められます。
     - 認証プロバイダーを設定していない場合は、警告メッセージが表示され、フォームが無効になります。**Access Control** セクションに進み、認証プロバイダーを設定します。
   - **Role**: ドロップダウンリストから、招待された各ユーザーに割り当てるロールを選択します。
3. **Invite users** をクリックします。
4. 確認ダイアログボックスで、選択したロールでユーザーが作成されたことを示す確認が表示されます。
5. 1つ以上のメールアドレスとメッセージを独自のメールクライアントで作成したメールにコピーし、ユーザーに送信します。
6. **Done** をクリックします。

検証

1. RHACS ポータルで、**Platform Configuration → Access Control → Auth providers**タブに移動します。
2. ユーザーの招待に使用した認証プロバイダーを選択します。
3. **Rules** セクションまでスクロールダウンします。
4. ユーザーのメールと認証プロバイダーのロールがリストに追加されていることを確認します。