Red Hat Advanced Cluster Security for Kubernetes 3.74

roxctl CLI

roxctl CLI

Last Updated: 2023-08-26

Red Hat Advanced Cluster Security for Kubernetes 3.74 roxctl CLI

roxctl CLI

法律上の通知

Copyright © 2023 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at
http://creativecommons.org/licenses/by-sa/3.0/
. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

このドキュメントでは、roxctl の構文と操作を含め、roxctl コマンドラインインターフェイスをインストールして使用する方法を説明します。一般的なコマンドの例をいくつか示します。

目次

第1章 ROXCTL CLI の使用を開始する 3
1.1. ROXCTL CLI のインストール 3
1.2. ROXCTL CLI を使用した認証 5
1.3. ROXCTL CLI の使用 6

第1章 ROXCTL CLI の使用を開始する

roxctl は、Red Hat Advanced Cluster Security for Kubernetes でコマンドを実行するためのコマンドラインインターフェイス (CLI) です。このトピックでは、**roxctl** 構文および操作を説明し、いくつかの一般的な例を示します。

1.1. ROXCTL CLI のインストール

バイナリーをダウンロードして **roxctl** CLI をインストールするか、コンテナーイメージから **roxctl** CLI を実行できます。

1.1.1. バイナリーをダウンロードして roxctl CLI をインストール

roxctl CLI をインストールして、コマンドラインインターフェイスから Red Hat Advanced Cluster Security for Kubernetes と対話できます。**roxctl** は、Linux、Windows、または macOS にインストールできます。

1.1.1.1. Linux への roxctl CLI のインストール

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

手順

1. **roxctl** CLI の最新バージョンをダウンロードします。

```
$ curl -O https://mirror.openshift.com/pub/rhacs/assets/3.74.4/bin/Linux/roxctl
```

2. **roxctl** バイナリーを実行可能にします。

```
$ chmod +x roxctl
```

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

```
$ echo $PATH
```

検証

- インストールした **roxctl** のバージョンを確認します。

```
$ roxctl version
```

1.1.1.2. macOS への roxctl CLI のインストール

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

手順

1. **roxctl** CLI の最新バージョンをダウンロードします。

```
$ curl -O https://mirror.openshift.com/pub/rhacs/assets/3.74.4/bin/Darwin/roxctl
```

2. バイナリーからすべての拡張属性を削除します。

```
$ xattr -c roxctl
```

3. **roxctl** バイナリーを実行可能にします。

```
$ chmod +x roxctl
```

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

```
$ echo $PATH
```

検証

- インストールした **roxctl** のバージョンを確認します。

```
$ roxctl version
```

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

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

手順

- **roxctl** CLI の最新バージョンをダウンロードします。

```
$ curl -O https://mirror.openshift.com/pub/rhacs/assets/3.74.4/bin/Windows/roxctl.exe
```

検証

- インストールした **roxctl** のバージョンを確認します。

```
$ roxctl version
```

1.1.2. コンテナーから roxctl CLI の実行

roxctl クライアントは、Red Hat Advanced Cluster Security for Kubernetes の **roxctl** イメージのデフォルトエントリーポイントです。コンテナーイメージで **roxctl** クライアントを実行するには、以下を行います。

前提条件

- はじめに、RHACS ポータルから認証トークンを生成している。

手順

1. **registry.redhat.io** レジストリーにログインします。

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

2. **roxctl** CLI の最新のコンテナーイメージをプルします。

```
$ docker pull registry.redhat.io/advanced-cluster-security/rhacs-roxctl-rhel8:3.74.4
```

CLI をインストールした後、次のコマンドを使用して CLI を実行できます。

```
$ docker run -e ROX_API_TOKEN=$ROX_API_TOKEN \
  -it registry.redhat.io/advanced-cluster-security/rhacs-roxctl-rhel8:3.74.4 \
  -e $ROX_CENTRAL_ADDRESS <command>
```

検証

- インストールした **roxctl** のバージョンを確認します。

```
$ docker run -it registry.redhat.io/advanced-cluster-security/rhacs-roxctl-rhel8:3.74.4 version
```

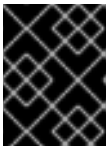
1.2. ROXCTL CLI を使用した認証

認証には、認証トークンまたは管理者パスワードを使用できます。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。

次の手順を使用して、認証トークンを生成します。

手順

1. RHACS ポータルに移動します。
2. **Platform Configuration** → **Integrations** に移動します。
3. **Authentication Tokens** カテゴリーまで下にスクロールし、**API Token** をクリックします。
4. **Generate Token** をクリックします。
5. 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**)。
6. **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。

注記

認証トークンを生成したら、それを **ROX_API_TOKEN** 変数としてエクスポートします。

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

トークンをファイルに保存して、**-token-file** オプションとともに使用することもできます。以下に例を示します。

```
$ roxctl central debug dump --token-file <token_file>
```

- **-password** (**-p**) オプションと **--token-file** オプションの両方を同時に使用することはできません。
- すでに **ROX_API_TOKEN** 変数を設定し、**--token-file** オプションを指定している場合、**roxctl** CLI は指定されたトークンファイルを認証に使用します。
- すでに **ROX_API_TOKEN** 変数を設定し、**--password** オプションを指定している場合、**roxctl** CLI は指定されたパスワードを認証に使用します。

1.3. ROXCTL CLI の使用

以下のセクションで、CLI を使用して一般的なタスクを実行する方法を確認します。

注記

- これらのコマンドを使用する前に、次の変数をエクスポートします。

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

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

- **--help** オプションを使用して、コマンドに関する詳細情報を取得できます。

1.3.1. Central のデータベースの管理

Central は、以下に関する情報を保存します。

- クラスターで観察されたアクティビティー
- 統合されたイメージレジストリーまたはスキャナーから取得された情報
- Red Hat Advanced Cluster Security for Kubernetes 設定

roxctl CLI を使用して、Central のデータベースをバックアップおよび復元できます。

Central データベースのバックアップ
次のコマンドを実行して、Central のデータベースをバックアップします。

```
$ roxctl -e "$ROX_CENTRAL_ADDRESS" central backup
```

Central データベースの復元

次のコマンドを実行して、Central のデータベースを復元します。

```
$ roxctl -e "$ROX_CENTRAL_ADDRESS" central db restore <backup_filename>
```

1.3.2. 保護されたクラスターの管理

Kubernetes または OpenShift Container Platform クラスターを保護するには、Red Hat Advanced Cluster Security for Kubernetes サービスをクラスターにデプロイする必要があります。**Platform Configuration → Clusters** ビューに移動して RHACS ポータルでデプロイメントファイルを生成するか、**roxctl** CLI を使用できます。

Sensor デプロイメントファイルの生成

Kubernetes

```
$ roxctl -e "$ROX_CENTRAL_ADDRESS" sensor generate k8s --name <cluster_name> --central
"$ROX_CENTRAL_ADDRESS"
```

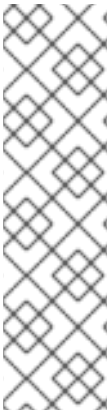
OpenShift Container Platform

```
$ roxctl -e "$ROX_CENTRAL_ADDRESS" sensor generate openshift --openshift-version <ocp-
version> --name <cluster_name> --central "$ROX_CENTRAL_ADDRESS" (1)
```

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

--help の出力を読んで、システムアーキテクチャーに応じて使用する必要のある他のオプションを確認してください。

--central に提供するエンドポイントに、Red Hat Advanced Cluster Security for Kubernetes サービスをデプロイしているクラスターから到達できることを確認してください。



注記

HAProxy、AWS Application Load Balancer (ALB)、AWS Elastic Load Balancing (ELB) などの gRPC 非対応のロードバランサーを使用している場合は、以下を行います。

- WebSocket Secure (**wss**) プロトコルを使用します。**wss** を使用するには、アドレスの前に **wss://** を付けます。
- アドレスの後にポート番号を追加します。次に例を示します。

```
$ roxctl sensor generate k8s --central wss://stackrox-central.example.com:443
```

生成された YAML ファイルを使用して、Sensor をインストールする
Sensor デプロイメントファイルを生成すると、**roxctl** は作業ディレクトリーに **sensor-<cluster_name>** というディレクトリーを作成します。Sensor をインストールするスクリプトは、このディレクトリーにあります。センサーインストールスクリプトを実行して、Sensor をインストールします。

```
$ ./sensor-<cluster_name>/sensor.sh
```

–

Sensor のインストールに必要な権限がないという警告が表示された場合は、画面の指示に従うか、クラスター管理者に連絡してください。

既存のクラスターのセンサーバーンドルをダウンロード
次のコマンドを使用して、クラスター名または ID を指定して、既存のクラスターの Sensor バンドルをダウンロードします。

```
$ roxctl sensor get-bundle <cluster_name_or_id>
```

クラスター統合の削除

```
$ roxctl -e "$ROX_CENTRAL_ADDRESS" cluster delete --name=<cluster_name>
```

重要

クラスター統合を削除しても、クラスターで実行されている Red Hat Advanced Cluster Security for Kubernetes サービスは削除されません。これらは、Sensor インストールバンドルから **delete-sensor.sh** スクリプトを実行することで削除できます。

1.3.3. ポリシーコンプライアンスの確認

roxctl CLI を使用して、ポリシーに準拠しているかどうかデプロイメント YAML ファイルおよびイメージを確認できます。

出力形式の設定
deployment check、**image check**、または **image scan** コマンドを使用してポリシーコンプライアンスをチェックする場合は、**-o** オプションを使用して出力形式を指定できます。このオプションは、コマンドの出力が端末にどのように表示されるかを決定します。

コマンドに **-o** オプションを追加し、形式を **json**、**table**、**csv**、または **junit** として指定することにより、出力形式を変更できます。

たとえば、次のコマンドはデプロイメントをチェックしてから、結果を **csv** 形式で表示します。

```
$ roxctl -e "$ROX_CENTRAL_ADDRESS" \
  deployment check --file =<yaml_filename> \
  -o csv
```

注記

出力形式に **-o** オプションを指定しない場合は、次のデフォルトの動作が使用されます。

- **deployment check** および **image check** コマンドの形式は **table** です。
- **imagescan** コマンドのデフォルトの出力形式は **json** です。これは、古いバージョンの CLI との互換性を目的とした古い JSON 形式の出力です。新しい JSON 形式で出力を取得するには、**-o json** などのフォーマットでオプションを指定します。

出力を設定するためにさまざまなオプションを使用できます。次の表に、オプションとそれらが使用可能な形式を示します。

オプション	説明	フォーマット
--compact-output	このオプションを使用して、JSON 出力をコンパクトな形式で表示します。	**json**
--headers	このオプションを使用して、カスタムヘッダーを指定します。	**table** および **csv**
--no-header	このオプションを使用して、出力からヘッダー行を省略します。	**table** および **csv**
--row-jsonpath-expressions	このオプションを使用して GJSON パス を指定し、出力から特定のアイテムを選択します。たとえば、デプロイメントチェックの **ポリシー名** および **重大度** を取得するには、次のコマンドを使用します。 $ roxctl -e "$ROX_CENTRAL_ADDRESS" \ deployment check --file=<yaml_filename> \ -o table --headers POLICY-NAME,SEVERITY \ --row-jsonpath-expressions="{results.#.violatedPolicies.#.name,results.#.violatedPolicies.#.severity}"	**table** および **csv**
--merge-output	このオプションを使用して、同じ値を持つテーブルセルを結合します。	**table**
headers-as-comment	このオプションを使用して、ヘッダー行をコメントとして出力に含めます。	**csv**
--junit-suite-name	このオプションを使用して、JUnit テストスイートの名前を指定します。	**junit**

デプロイ YAML ファイルを確認する
次のコマンドは、YAML デプロイメントファイル内のセキュリティーポリシーのビルド時およびデプロイ時の違反をチェックします。次のコマンドを使用して、以下を検証します。

- リソース制限や特権オプションなど、YAML ファイルの設定オプション
- コンポーネントや脆弱性など、YAML ファイルで使用されるイメージの側面

```
$ roxctl -e "$ROX_CENTRAL_ADDRESS" deployment check --file=<yaml_filename>
```

イメージを確認する
次のコマンドは、イメージ内のセキュリティーポリシーのビルド時違反をチェックします。

```
$ roxctl -e "$ROX_CENTRAL_ADDRESS" image check --image=<image_name>
```

イメージスキャン結果の確認
特定のイメージのスキャン結果を確認することもできます。

次のコマンドは、イメージで見つかったコンポーネントおよび脆弱性を JSON 形式で返します。形式は API リファレンスで定義されています。

```
$ roxctl -e "$ROX_CENTRAL_ADDRESS" image scan --image <image_name>
```

Red Hat Advanced Cluster Security for Kubernetes で、関連付けられたレジストリーとスキャナーからイメージメタデータおよびイメージスキャン結果を再プルするには、**-force** オプションを追加します。

注記

特定のイメージスキャン結果を確認するには、**Image** リソースの **read** および **write** の両方の権限を持つトークンが必要です。デフォルトの **継続的インテグレーション** システムのロールには、すでに必要な権限があります。

1.3.4. デバッグの問題

中央ログレベルの管理
Central は、情報をコンテナーログに保存します。

ログの表示
次を実行すると、Central のコンテナーログを確認できます。

Kubernetes

```
$ kubectl logs -n stackrox <central_pod>
```

OpenShift Container Platform

```
$ oc logs -n stackrox <central_pod>
```

現在のログレベルの表示
ログレベルを変更して、Central ログの情報を増減できます。次のコマンドを実行して、現在のログレベルを表示します。

```
$ roxctl -e "$ROX_CENTRAL_ADDRESS" central debug log
```

ログレベルの変更
次のコマンドを実行して、ログレベルを変更します。

```
$ roxctl -e "$ROX_CENTRAL_ADDRESS" central debug log --level=<log_level> (1)
```

(1) **<log_level>** の許容値は、**Panic**、**Fatal**、**Error**、**Warn**、**Info**、および **Debug** です。

デバッグ情報の取得
問題を調査するためのデバッグ情報を収集するには、次のコマンドを実行します。

```
$ roxctl -e "$ROX_CENTRAL_ADDRESS" central debug dump
```

1.3.5. ビルド時のネットワークポリシーの生成

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

前提条件

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

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

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

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

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

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

   (1) このパターンでは、

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

   以下は、パターンに一致するいくつかの例です。

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

手順

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

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

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

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

(1) Kubernetes マニフェストがあるフォルダーのパスを指定します。

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

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

関連情報

- ビルド時のネットワークポリシージェネレーターの使用