



Red Hat Advanced Cluster Security for Kubernetes 4.5

roxctl CLI

roxctl CLI

roxctl CLI

法律上の通知

Copyright © 2024 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 のインストール	3
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	3
1.2. コンテナから ROXCTL CLI の実行	5
第2章 ROXCTL CLI の使用	6
2.1. 前提条件	6
2.2. 認証情報の取得	6
2.3. ROXCTL CLI を使用した認証	7
2.4. RHACS CLOUD SERVICE での ROXCTL CLI の設定と使用	9
第3章 セキュアクラスターの管理	11
3.1. 前提条件	11
3.2. SENSOR デプロイメントファイルの生成	11
3.3. SENSOR.SH スクリプトを使用した SENSOR のインストール	12
3.4. 既存のクラスターの SENSOR バンドルのダウンロード	12
3.5. クラスター統合の削除	12
第4章 ポリシーコンプライアンスの確認	14
4.1. 前提条件	14
4.2. 出力形式の設定	14
4.3. デプロイYAML ファイルの確認	15
4.4. イメージの確認	16
4.5. イメージスキャン結果の確認	16
第5章 デバッグの問題	18
5.1. 前提条件	18
5.2. ログの表示	18
5.3. 現在のログレベルの表示	18
5.4. ログレベルの変更	18
5.5. デバッグ情報の取得	19
第6章 ビルド時のネットワークポリシーの生成	20
6.1. 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	20
第7章 ROXCTL CLI を使用したイメージスキャン	22
7.1. リモートクラスターを使用したイメージのスキャン	22
7.2. ROXCTL IMAGE SCAN コマンドのオプション	23
第8章 ROXCTL CLI コマンドリファレンス	25
8.1. ROXCTL	25
8.2. ROXCTL CENTRAL	27
8.3. ROXCTL クラスター	49
8.4. ROXCTL COLLECTOR	52
8.5. ROXCTL COMPLETION	54
8.6. ROXCTL DECLARATIVE-CONFIG	56
8.7. ROXCTL DEPLOYMENT	66
8.8. ROXCTL HELM	69
8.9. ROXCTL IMAGE	73
8.10. ROXCTL NETPOL	78
8.11. ROXCTL SCANNER	82
8.12. ROXCTL SENSOR	86
8.13. ROXCTL VERSION	93

第1章 ROXCTL CLI のインストール

roxctl は、Red Hat Advanced Cluster Security for Kubernetes (RHACS) 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。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、コンテナイメージから **roxctl** CLI を実行できます。

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 をインストールすると、コマンドラインインターフェイスから RHACS を操作できます。**roxctl** は、Linux、Windows、または macOS にインストールできます。

1.1.1. Linux への roxctl CLI のインストール

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



注記

Linux 用の **roxctl** CLI は、**amd64**、**arm64**、**ppc64le**、**s390x** アーキテクチャーで使用できます。

手順

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

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

2. **roxctl** CLI をダウンロードします。

```
$ curl -L -f -o roxctl  
"https://mirror.openshift.com/pub/rhacs/assets/4.5.1/bin/Linux/roxctl${arch}"
```

3. **roxctl** バイナリーを実行可能にします。

```
$ chmod +x roxctl
```

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

```
$ echo $PATH
```

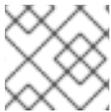
検証

- インストールした **roxctl** のバージョンを確認します。

```
$ roxctl version
```

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。



注記

macOS 用の **roxctl** CLI は、**amd64** および **arm64** アーキテクチャーで使用できます。

手順

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

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

2. **roxctl** CLI をダウンロードします。

```
$ curl -L -f -o roxctl
"https://mirror.openshift.com/pub/rhacs/assets/4.5.1/bin/Darwin/roxctl${arch}"
```

3. バイナリーからすべての拡張属性を削除します。

```
$ xattr -c roxctl
```

4. **roxctl** バイナリーを実行可能にします。

```
$ chmod +x roxctl
```

5. **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
```

1.1.3. Windows への roxctl CLI のインストール

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



注記

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

手順

- **roxctl** CLI をダウンロードします。

```
$ curl -f -O https://mirror.openshift.com/pub/rhacs/assets/4.5.1/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
```

1.2. コンテナから ROXCTL CLI の実行

roxctl クライアントは、RHACS **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 の最新のコンテナイメージをプルします。

```
$ docker pull registry.redhat.io/advanced-cluster-security/rhacs-roxctl-rhel8:4.5.1
```

CLI をインストールしたら、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-rhel8:4.5.1 \  
-e $ROX_CENTRAL_ADDRESS <command>
```



注記

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

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-rhel8:4.5.1 version
```

第2章 ROXCTL CLI の使用

2.1. 前提条件

- 以下のコマンドを使用して **ROX_ENDPOINT** 環境変数を設定している。

```
$ export ROX_ENDPOINT=<host:port> 1
```

- 1 **ROX_ENDPOINT** 環境変数に保存するホストおよびポートの情報。

2.2. 認証情報の取得

次の手順では、**roxctl Central whoami** コマンドを使用して、Central で認証ステータスとユーザープロフィールに関する情報を取得する方法を説明します。出力例は、ユーザーのロール、アクセス許可、さまざまな管理機能など、表示されることが予想されるデータを示しています。この手順では、Central 内でのアクセス権とロールを確認できます。

手順

- 次のコマンドを実行して、Central の現在の認証ステータスとユーザー情報に関する情報を取得します。

```
$ roxctl central whoami
```

出力例

```
UserID:
  <redacted>
User name:
  <redacted>
Roles:
  APIToken creator, Admin, Analyst, Continuous Integration, Network Graph Viewer, None,
  Sensor Creator, Vulnerability Management Approver, Vulnerability Management Requester,
  Vulnerability Manager, Vulnerability Report Creator
Access:
  rw Access
  rw Administration
  rw Alert
  rw CVE
  rw Cluster
  rw Compliance
  rw Deployment
  rw DeploymentExtension
  rw Detection
  rw Image
  rw Integration
  rw K8sRole
  rw K8sRoleBinding
  rw K8sSubject
  rw Namespace
  rw NetworkGraph
  rw NetworkPolicy
```

```
rw Node
rw Secret
rw ServiceAccount
rw VulnerabilityManagementApprovals
rw VulnerabilityManagementRequests
rw WatchedImage
rw WorkflowAdministration
```

出力を確認して、認証とユーザーの詳細が想定どおりであることを確認します。

2.3. ROXCTL CLI を使用した認証

認証には、API トークン、管理者パスワード、または **roxctl central login** コマンドを使用できます。

API トークンを効果的に使用するには、次のガイドラインに従ってください。

- 継続的インテグレーション (CI) を備えた運用環境で API トークンを使用します。各トークンには特定のアクセス許可が割り当てられ、実行できるアクションを制御できます。さらに、API トークンはブラウザーベースのログインなどの対話型プロセスを必要としないため、自動プロセスに最適です。これらのトークンの有効期間 (TTL) の値は1年で、有効期間が長くなり、シームレスな統合と運用効率が向上します。
- 管理者パスワードはテスト目的にのみ使用してください。これを実稼働環境で使用しないでください。
- **roxctl Central login** コマンドは、対話型かつローカルでのみ使用してください。

2.3.1. 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。

2.3.2. API トークンのエクスポートと保存

手順

1. 認証トークンを生成したら、次のコマンドを入力して、**ROX_API_TOKEN** 変数としてエクスポートします。

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

- (オプション): 次のコマンドを入力して、トークンをファイルに保存し、**--token-file** オプションとともに使用することもできます。

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

次のガイドラインに注意してください。

- **-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** 変数を設定しており、**--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検証に使用します。
- すでに **ROX_API_TOKEN** 変数を設定しており、**--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検証に使用します。

2.3.3. 認証プロバイダーを使用した roxctl での認証

Central で認証プロバイダーを設定し、**roxctl** CLI を使用してログインプロセスを開始できます。次の手順で説明するように、**ROX_ENDPOINT** 変数を設定し、**roxctl central login** コマンドでログインプロセスを開始し、ブラウザウィンドウで認証プロバイダーを選択して、**roxctl** CLI からトークン情報を取得します。

前提条件

- フラグメントモードまたはクエリーモードの OpenID Connect (OIDC) など、任意の認証プロバイダーを選択しました。

手順

1. 次のコマンドを実行して、**ROX_ENDPOINT** 変数を Central のホスト名とポートに設定します。

```
export ROX_ENDPOINT=<central_hostname:port>
```

2. 次のコマンドを実行して、Central へのログインプロセスを開始します。

```
$ roxctl central login
```

3. **roxctl** CLI 内では、URL が出力として出力され、ブラウザウィンドウにリダイレクトされ、使用する認証プロバイダーを選択できます。
4. 認証プロバイダーを使用してログインします。
ログインに成功すると、認証が成功したことがブラウザウィンドウに表示されるので、ブラウザウィンドウを閉じることができます。
5. **roxctl** CLI は、アクセストークン、アクセストークンの有効期限、更新トークンが発行されている場合はそのトークン、これらの値がローカルに保存されているという通知などの詳細を含むトークン情報を表示します。

出力例

■

Please complete the authorization flow in the browser with an auth provider of your choice.
If no browser window opens, please click on the following URL:
`http://127.0.0.1:xxxxx/login`

INFO: Received the following after the authorization flow from Central:

INFO: Access token: <redacted> **1**

INFO: Access token expiration: 2023-04-19 13:58:43 +0000 UTC **2**

INFO: Refresh token: <redacted> **3**

INFO: Storing these values under `$HOME/.roxctl/login...` **4**

- 1** アクセストークン。
- 2** アクセストークンの有効期限。
- 3** 更新トークン。
- 4** アクセストークンの値、アクセストークンの有効期限、および更新トークンがローカルに保存されるディレクトリー。

重要

設定が保存されるディレクトリーを決定するように環境を設定してください。デフォルトでは、設定は `$HOME/.roxctl/roxctl-config` ディレクトリーに保存されます。

- `$ROX_CONFIG_DIR` 環境変数を設定すると、設定は `$ROX_CONFIG_DIR/roxctl-config` ディレクトリーに保存されます。このオプションは、優先度が最も高くなります。
- `$XDG_RUNTIME_DIR` 環境変数を設定し、`$ROX_CONFIG_DIR` 変数が設定されていないと、設定は `$XDG_RUNTIME_DIR/roxctl-config` ディレクトリーに保存されます。
- `$ROX_CONFIG_DIR` または `$XDG_RUNTIME_DIR` 環境変数を設定しないと、設定は `$HOME/.roxctl/roxctl-config` ディレクトリーに保存されます。

2.4. RHACS CLOUD SERVICE での ROXCTL CLI の設定と使用

手順

- 次のコマンドを実行して `ROX_API_TOKEN` をエクスポートします。

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

- 次のコマンドを実行して `ROX_ENDPOINT` をエクスポートします。

```
$ export ROX_ENDPOINT=<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。
- Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service) で、Central アドレスを必要とする `roxctl` コマンドを使用する場合は、Red Hat Hybrid Cloud Console の **Instance Details** セクションに表示される **Central インスタンスアドレス** を使用します。たとえ

ば、**acs-data-ABCD12345.acs.rhcloud.com** の代わりに **acs-ABCD12345.acs.rhcloud.com** を使用します。

第3章 セキュアクラスターの管理

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 (RHACS) 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。

3.1. 前提条件

- 以下のコマンドを使用して **ROX_ENDPOINT** 環境変数を設定している。

```
$ export ROX_ENDPOINT=<host:port> ❶
```

- ❶ **ROX_ENDPOINT** 環境変数に保存するホストおよびポートの情報。

3.2. SENSOR デプロイメントファイルの生成

Kubernetes システムのファイルの生成

手順

- 次のコマンドを実行して、Kubernetes クラスターに必要な Sensor 設定を生成し、Central インスタンスに関連付けます。

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

OpenShift Container Platform システムのファイルの生成

手順

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

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

- ❶ **--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。



重要

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、次のガイドラインに従ってください。

- 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
```

3.3. SENSOR.SH スクリプトを使用した SENSOR のインストール

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。

手順

- Sensor 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。

3.4. 既存のクラスターの SENSOR バンドルのダウンロード

手順

- 次のコマンドを実行して、**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>
```

3.5. クラスター統合の削除

手順

- クラスターを削除する前に、Central から削除するクラスター名が正しいことを確認してください。

```
$ roxctl cluster delete --name=<cluster_name>
```



重要

インストール方法によっては、クラスター統合を削除しても、クラスター内で実行されている RHACS サービスは削除されません。サービスは、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。

第4章 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。

4.1. 前提条件

- 以下のコマンドを使用して **ROX_ENDPOINT** 環境変数を設定している。

```
$ export ROX_ENDPOINT=<host:port> 1
```

- ROX_ENDPOINT** 環境変数に保存するホストおよびポートの情報。

4.2. 出力形式の設定

roxctl deployment check または **roxctl image check** コマンドを使用してポリシーへの準拠をチェックする場合、コマンドに **-o** オプションを使用し、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。

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 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。
- image scan** 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。トラブルシューティングの目的でデータを収集する場合は、古い JSON 形式の出力を使用します。

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

オプション	説明	フォーマット
--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。 <pre>\$ roxctl deployment check -- file=<yaml_filename> \ -o table --headers POLICY- NAME,SEVERITY \ --row-jsonpath-expressions=" {results..violatedPolicies..name,results..violat edPolicies..severity}"</pre>	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

4.3. デプロイ YAML ファイルの確認

手順

- 次のコマンドを実行して、YAML デプロイメントファイル内のセキュリティポリシーのビルド時およびデプロイ時の違反を確認します。

```
$ roxctl deployment check --file=<yaml_filename> \ ①
--namespace=<cluster_namespace> \ ②
--cluster=<cluster_name_or_id> \ ③
--verbose ④
```

- ① **<yaml_filename>** には、ポリシー評価のために Central に送信する1つ以上のデプロイメントを含む YAML ファイルを指定します。 **--file** フラグ (**--file=<yaml_filename1>**、 **--file=<yaml_filename2>** など) を使用して、ポリシー評価のために Central に送信する複数の YAML ファイルを指定することもできます。
- ② **<cluster_namespace>** には、仕様に namespace がないデプロイメントのネットワークポリシー、ロールベースアクセス制御 (RBAC)、サービスなどのコンテキスト情報を使用して、デプロイメントを強化するための namespace を指定します。仕様で定義されている namespace は変更されません。デフォルト値は **default** です。
- ③ **<cluster_name_or_id>** には、クラスター固有の情報を使用した拡張デプロイメントを有効にするための評価のコンテキストとして使用するクラスター名または ID を指定します。
- ④ **--verbose** フラグを有効にすると、ポリシーチェック中に各デプロイメントに関する追加情報を受信できます。拡張情報には RBAC 権限レベルと 適用されるネットワークポリ

情報で戻ります。追加情報は、RHACS 権限レベルと、適用されるイメージポリシーの包括的なリストが含まれます。



注記

--verbose フラグを有効にするかどうかに関係なく、JSON 出力で各デプロイメントの追加情報を確認できます。

形式は API リファレンスで定義されています。Red Hat Advanced Cluster Security for Kubernetes (RHACS) 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。

このコマンドは、以下の項目を検証します。

- 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

4.4. イメージの確認

手順

- 次のコマンドを実行して、イメージ内のセキュリティポリシーのビルド時の違反を確認します。

```
$ roxctl image check --image=<image_name>
```

形式は API リファレンスで定義されています。Red Hat Advanced Cluster Security for Kubernetes (RHACS) 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。

関連情報

- [roxctl image](#)

4.5. 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。

手順

- 以下のコマンドを実行して、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。

```
$ roxctl image scan --image <image_name>
```

形式は API リファレンスで定義されています。Red Hat Advanced Cluster Security for Kubernetes (RHACS) 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。

関連情報

- [roxctl image](#)

第5章 デバッグの問題

Central は、情報をコンテナログに保存します。

5.1. 前提条件

- 以下のコマンドを使用して **ROX_ENDPOINT** 環境変数を設定している。

```
$ export ROX_ENDPOINT=<host:port> 1
```

- 1 **ROX_ENDPOINT** 環境変数に保存するホストおよびポートの情報。

5.2. ログの表示

oc または **kubect**l コマンドを使用して、Central Pod のログを表示できます。

手順

- **kubect**l を使用して Central Pod のログを表示するには、次のコマンドを実行します。

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

- **oc** を使用して Central Pod のログを表示するには、次のコマンドを実行します。

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

5.3. 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。

手順

- 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 central debug log
```

関連情報

- [roxctl central debug](#)

5.4. ログレベルの変更

手順

- 次のコマンドを実行して、ログレベルを変更します。

```
$ roxctl 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。

関連情報

- [roxctl central debug](#)

5.5. デバッグ情報の取得

手順

- 次のコマンドを実行して、問題を調査するためのデバッグ情報を収集します。

```
$ roxctl central debug dump
```

- RHACS 管理者パスワードまたは API トークンと Central アドレスを使用して診断バンドルを生成するには、[roxctl CLI を使用した診断バンドルの生成](#) の手順に従います。

関連情報

- [roxctl central debug](#)

第6章 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。

6.1. 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

roxctl CLI のビルトインネットワークポリシージェネレーターを使用して、ネットワークポリシーを生成できます。

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: 環境変数の使用](#)
 - [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 netpol generate -h
```

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

```
$ roxctl netpol generate <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 netpol generate` 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
<code>-h, --help</code>	<code>netpol</code> コマンドのヘルプテキストを表示します。
<code>-d, --output-dir <dir></code>	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。
<code>-f, --output-file <filename></code>	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。
<code>--fail</code>	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。
<code>--remove</code>	出力パスがすでに存在する場合は削除します。
<code>--strict</code>	警告をエラーとして扱います。デフォルト値は false です。
<code>--dnSPORT</code>	生成されたポリシーのEgressルールで使用するデフォルトのDNSポートを指定します。デフォルト値は 53 です。

第7章 ROXCTL CLI を使用したイメージスキャン

roxctl CLI を使用して、OpenShift Container Platform 統合イメージレジストリーなどのクラスタのローカルレジストリーなど、イメージレジストリーに保存されているイメージをスキャンできます。

7.1. リモートクラスタを使用したイメージのスキャン

委任スキャン設定で適切なクラスタを指定するか、次の手順で説明するクラスタパラメーターを使用して、リモートクラスタでクラスタのローカルレジストリーからイメージをスキャンできます。



重要

委任されたイメージスキャンの設定方法の詳細は、[委任されたイメージスキャンの設定](#)を参照してください。

手順

- 次のコマンドを実行して、リモートクラスタで指定されたイメージをスキャンします。

```
$ roxctl image scan \  
  --image=<image_registry>/<image_name> \ 1  
  --cluster=<cluster_detail> \ 2  
  [flags] 3
```

- 1** **<image_registry>** には、イメージが配置されているレジストリーを指定します (例: **image-registry.openshift-image-registry.svc:5000**)。 **<image_name>** には、スキャンするイメージの名前を指定します (例: **default/image-stream:latest**)。
- 2** **<cluster_detail>** には、リモートクラスタの名前または ID を指定します。たとえば、**remote** という名前を指定します。
- 3** オプション: **[flags]** には、コマンドの動作を変更するパラメーターを指定できます。

オプションのパラメーターの詳細は、[roxctl image scan command options](#) を参照してください。

出力例

```
{  
  "Id":  
  "sha256:3f439d7d71adb0a0c8e05257c091236ab00c6343bc44388d091450ff58664bf9", 1  
  "name": { 2  
    "registry": "image-registry.openshift-image-registry.svc:5000", 3  
    "remote": "default/image-stream", 4  
    "tag": "latest", 5  
    "fullName": "image-registry.openshift-image-registry.svc:5000/default/image-stream:latest"  
  }, 6  
  [...]
```

- 1** イメージのフィンガープリントとして機能するイメージの一意的識別子これは、イメージの完全性と信頼性を確保するのに役立ちます。

- ② イメージに関する詳細情報が含まれます。
- ③ イメージが保存されるイメージレジストリーの場所。
- ④ イメージへのリモートパス。
- ⑤ このイメージに関連付けられたバージョンまたはタグ。
- ⑥ レジストリー、リモートパス、およびタグを組み合わせたイメージの完全な名前。

7.2. ROXCTL IMAGE SCAN コマンドのオプション

roxctl image scan 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
--cluster string	イメージスキャンを特定のクラスターに委任します。
--compact-output	JSON 出力をコンパクトな形式で出力します。デフォルト値は false です。
-f, --force	Central のスキャンのキャッシュを無視し、Scanner から新たにもう一度強制的にプルします。デフォルト値は false です。
--headers strings	ヘッダーを表形式で出力します。デフォルト値には、 COMPONENT 、 VERSION 、 CVE 、 SEVERITY 、および LINK が含まれます。
--headers-as-comments	CSV 表形式出力にヘッダーをコメントとして出力します。デフォルト値は false です。
-h, --help	roxctl image scan コマンドのヘルプテキストを表示します。
-i, --image string	スキャンするイメージ名と参照を指定します。
-a, --include-snoozed	延期されたものと延期されていないものの両方の一般的な脆弱性とエクスポージャー (CVE) を返します。デフォルト値は false です。
--merge-output	表形式の出力で重複したセルを結合します。デフォルト値は true です。
--no-header	表形式のヘッダーを出力しません。デフォルト値は false です。

オプション	説明
-o, --output string	出力形式を指定します。形式を選択して結果の表示をカスタマイズできます。形式には、 テーブル 、 CSV 、 JSON 、および SARIF が含まれます。
-r, --retries int	操作がエラーで中止されるまでの再試行回数を設定します。デフォルト値は 3 です。
-d, --retry-delay int	次の再試行までの待機時間を秒単位で設定します。デフォルト値は 3 です。
--row-jsonpath-expressions string	JSON パス式を使用して、JSON オブジェクトから行を作成します。詳細は、 roxctl image scan --help コマンドを実行してください。

第8章 ROXCTL CLI コマンドリファレンス

8.1. ROXCTL

roxctl CLI で使用可能なコマンドとオプションのパラメーターを表示します。これらのコマンドを使用するには、管理者権限を持つアカウントが必要です。

使用方法

```
$ roxctl [command] [flags]
```

表8.1 利用可能なコマンド

コマンド	説明
central	Central サービスに関連するコマンド。
cluster	クラスターに関連するコマンド。
collector	Collector サービスに関連するコマンド。
completion	シェル補完スクリプトを生成します。
declarative-config	宣言型設定を管理します。
デプロイメント	デプロイメントに関連するコマンド。
helm	Red Hat Advanced Cluster Security for Kubernetes (RHACS) Helm チャートに関連するコマンド。
image	特定のイメージに対して実行できるコマンド。
netpol	ネットワークポリシーに関連するコマンド。
scanner	Scanner サービスに関連するコマンド。
sensor	セキュアクラスターに RHACS サービスをデプロイします。
version	現在の roxctl のバージョンを表示します。

8.1.1. roxctl コマンドのオプション

roxctl 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
--ca string	安全な接続のためのカスタム CA 証明書ファイルパスを指定します。あるいは、 ROX_CA_CERT_FILE 環境変数を使用して、ファイルパスを指定することもできます。
--direct-grpc	接続パフォーマンスを向上させるには、 --direct-grpc を設定します。あるいは、 ROX_DIRECT_GRPC_CLIENT 環境変数を true に設定することで、直接 gRPC を有効にできます。デフォルト値は false です。
-e, --endpoint string	サービスが接続するエンドポイントを設定します。あるいは、 ROX_ENDPOINT 環境変数を使用して、エンドポイントを設定することもできます。デフォルト値は localhost:8443 です。
--force-http1	すべての接続で HTTP/1 を強制的に使用します。あるいは、 ROX_CLIENT_FORCE_HTTP1 環境変数を true に設定することで、HTTP/1 の使用を強制できます。デフォルト値は false です。
--insecure	安全でない接続オプションを有効にします。あるいは、 ROX_INSECURE_CLIENT 環境変数を true に設定することで、安全でない接続オプションを有効にすることができます。デフォルト値は false です。
--insecure-skip-tls-verify	TLS 証明書の検証をスキップします。あるいは、 ROX_INSECURE_CLIENT_SKIP_TLS_VERIFY 環境変数を true に設定することで、TLS 証明書の検証をスキップできます。デフォルト値は false です。
--no-color	カラー出力を無効にします。あるいは、 ROX_NO_COLOR 環境変数を true に設定することで、カラー出力を無効にすることができます。デフォルト値は false です。
-p, --password string	Basic 認証のパスワードを指定します。あるいは、 ROX_ADMIN_PASSWORD 環境変数を使用して、パスワードを設定することもできます。
--plaintext	暗号化されていない接続を使用します。あるいは、 ROX_PLAINTEXT 環境変数を true に設定することで、暗号化されていない接続を有効にすることができます。デフォルト値は false です。

オプション	説明
-s, --server-name string	SNI に使用する TLS サーバー名を設定します。あるいは、 ROX_SERVER_NAME 環境変数を使用して、サーバー名を設定することもできます。
--token-file string	指定されたファイルで提供される API トークンを認証に使用します。あるいは、 ROX_API_TOKEN 環境変数を使用して、トークンを設定することもできます。

8.2. ROXCTL CENTRAL

Central サービスに関連するコマンド。

使用方法

```
$ roxctl central [command] [flags]
```

表8.2 利用可能なコマンド

コマンド	説明
backup	Red Hat Advanced Cluster Security for Kubernetes (RHACS) データベースと証明書のバックアップを作成します。
cert	Central サービスの証明書チェーンをダウンロードします。
db	データベース操作を制御します。
debug	Central サービスをデバッグします。
generate	Central のデプロイメントのオーケストレーターオブジェクトを含む必要な YAML 設定ファイルを生成します。
init-bundles	Central のバンドルを初期化します。
login	Central インスタンスにログインしてトークンを取得します。
userpki	ユーザー証明書認可プロバイダーを管理します。
whoami	現在のユーザーとその認証方法に関する情報を表示します。

8.2.1. 親コマンドから継承された roxctl central コマンドオプション

roxctl central コマンドは、roxctl の親コマンドから継承された次のオプションをサポートします。

オプション	説明
--ca string	安全な接続のためのカスタム CA 証明書ファイルパスを指定します。あるいは、 ROX_CA_CERT_FILE 環境変数を使用して、ファイルパスを指定することもできます。
--direct-grpc	接続パフォーマンスを向上させるには、 --direct-grpc を設定します。あるいは、 ROX_DIRECT_GRPC_CLIENT 環境変数を true に設定することで、直接 gRPC を有効にできます。デフォルト値は false です。
-e, --endpoint string	サービスが接続するエンドポイントを設定します。あるいは、 ROX_ENDPOINT 環境変数を使用して、エンドポイントを設定することもできます。デフォルト値は localhost:8443 です。
--force-http1	すべての接続で HTTP/1 を強制的に使用します。あるいは、 ROX_CLIENT_FORCE_HTTP1 環境変数を true に設定することで、HTTP/1 の使用を強制できます。デフォルト値は false です。
--insecure	安全でない接続オプションを有効にします。あるいは、 ROX_INSECURE_CLIENT 環境変数を true に設定することで、安全でない接続オプションを有効にすることができます。デフォルト値は false です。
--insecure-skip-tls-verify	TLS 証明書の検証をスキップします。あるいは、 ROX_INSECURE_CLIENT_SKIP_TLS_VERIFY 環境変数を true に設定することで、TLS 証明書の検証をスキップできます。デフォルト値は false です。
--no-color	カラー出力を無効にします。あるいは、 ROX_NO_COLOR 環境変数を true に設定することで、カラー出力を無効にすることができます。デフォルト値は false です。
-p, --password string	Basic 認証のパスワードを指定します。あるいは、 ROX_ADMIN_PASSWORD 環境変数を使用して、パスワードを設定することもできます。
--plaintext	暗号化されていない接続を使用します。あるいは、 ROX_PLAINTEXT 環境変数を true に設定することで、暗号化されていない接続を有効にすることができます。デフォルト値は false です。

オプション	説明
-s, --server-name string	SNI に使用する TLS サーバー名を設定します。あるいは、 ROX_SERVER_NAME 環境変数を使用して、サーバー名を設定することもできます。
--token-file string	指定されたファイルで提供される API トークンを認証に使用します。あるいは、 ROX_API_TOKEN 環境変数を使用して、トークンを設定することもできます。



注記

これらのオプションは、**roxctl central** コマンドのすべてのサブコマンドに適用されます。

8.2.2. roxctl central backup

RHACS データベースと証明書のバックアップを作成します。

使用方法

```
$ roxctl central backup [flags]
```

表8.3 オプション

オプション	説明
--certs-only	証明書のみをバックアップするように指定します。外部データベースを使用する場合、このオプションは証明書を含むバックアップバンドルを生成するために使用されます。デフォルト値は false です。
--output string	バックアップを保存する場所を指定します。動作は指定されたパスによって異なります。 <ul style="list-style-type: none"> パスがファイルパスの場合、バックアップはファイルに書き込まれ、ファイルがすでに存在する場合は上書きされます。ディレクトリが存在する必要がある場合があります。 パスがディレクトリの場合、バックアップはサーバーが指定するファイル名でこのディレクトリに保存されます。 この引数を省略した場合、バックアップはサーバーが指定するファイル名で、現在の作業ディレクトリに保存されます。

オプション	説明
-t, --timeout duration	API リクエストのタイムアウトを指定します。リクエストの最大期間を表します。デフォルト値は 1h0m0s です。

8.2.3. roxctl central cert

Central サービスの証明書チェーンをダウンロードします。

使用方法

```
$ roxctl central cert [flags]
```

表8.4 オプション

オプション	説明
--output string	PEM 証明書を保存するファイル名を指定します。- を使用して標準出力を生成できます。デフォルト値は - です。
--retry-timeout duration	API リクエストが再試行されるまでのタイムアウトを指定します。値が 0 の場合、再試行せずにリクエスト期間全体を待機することを意味します。デフォルト値は 20s です。
-t, --timeout duration	リクエストの最大期間を表す API リクエストのタイムアウトを指定します。デフォルト値は 1m0s です。

8.2.4. roxctl central login

Central インスタンスにログインしてトークンを取得します。

使用方法

```
$ roxctl central login [flags]
```

表8.5 オプション

オプション	説明
-t, --timeout duration	リクエストの最大期間を表す API リクエストのタイムアウトを指定します。デフォルト値は 5m0s です。

8.2.5. roxctl central whoami

現在のユーザーとその認証方法に関する情報を表示します。

使用方法

```
$ roxctl central whoami [flags]
```

表8.6 オプション

オプション	説明
--retry-timeout duration	API リクエストが再試行されるまでのタイムアウトを指定します。値が0の場合、再試行せずにリクエスト期間全体を待機することを意味します。デフォルト値は 20s です。
-t, --timeout duration	リクエストの最大期間を表す API リクエストのタイムアウトを指定します。デフォルト値は 1m0s です。

8.2.6. roxctl central db

データベース操作を制御します。

使用方法

```
$ roxctl central db [flags]
```

表8.7 オプション

オプション	説明
-t, --timeout duration	リクエストの最大期間を表す API リクエストのタイムアウトを指定します。デフォルト値は 1h0m0s です。

8.2.6.1. roxctl central db restore

以前のバックアップから RHACS データベースを復元します。

使用方法

```
$ roxctl central db restore <file> [flags] 1
```

1 **<file>** には、復元するデータベースバックアップファイルを指定します。

表8.8 オプション

オプション	説明
-f, --force	true に設定すると、復元は確認なしで実行されます。デフォルト値は false です。
--interrupt	true に設定すると、実行中の復元プロセスが中断され、続行できるようになります。デフォルト値は false です。

8.2.6.2. roxctl central db generate

Central データベースバンドルを生成します。

使用方法

```
$ roxctl central db generate [flags]
```

表8.9 オプション

オプション	説明
--debug	true に設定すると、テンプレートはローカルファイルシステムから読み取られます。デフォルト値は false です。
--debug-path string	ローカルファイルシステム内の Helm テンプレートへのパスを指定します。詳細は、 roxctl central db generate コマンドを実行してください。
--enable-pod-security-policies	true に設定すると、 PodSecurityPolicy リソースが作成されます。デフォルト値は true です。

8.2.6.3. roxctl central db generate k8s

Central のデータベースコンポーネントをデプロイするための Kubernetes YAML ファイルを生成します。

使用方法

```
$ roxctl central db generate k8s [flags]
```

表8.10 オプション

オプション	説明
--central-db-image string	使用する Central データベースイメージを指定します。指定しない場合は、 --image-defaults に対応するデフォルト値が使用されます。

オプション	説明
--image-defaults string	コンテナイメージのデフォルト設定を指定します。イメージのダウンロード元のリポジトリ、イメージ名、タグの形式を制御します。デフォルト値は、 development_build です。
--output-dir output directory	デプロイメントバンドルを保存するディレクトリーを指定します。デフォルト値は central-db-bundle です。

8.2.6.4. roxctl central db restore cancel

進行中の Central データベースの復元プロセスをキャンセルします。

使用方法

```
$ roxctl central db restore cancel [flags]
```

表8.11 オプション

オプション	説明
f, --force	true に設定すると、確認なしでキャンセルが実行されます。デフォルト値は false です。

8.2.6.5. roxctl central db restore status

進行中のデータベース復元プロセスに関する情報を表示します。

使用方法

```
$ roxctl central db restore status [flags]
```

8.2.6.6. roxctl central db generate k8s pvc

Central で永続ボリューム要求 (PVC) 用の Kubernetes YAML ファイルを生成します。

使用方法

```
$ roxctl central db generate k8s pvc [flags]
```

表8.12 オプション

オプション	説明
--name string	Central データベースの外部ボリューム名を指定します。デフォルト値は central-db です。

オプション	説明
--size uint32	Central データベースの外部ボリュームのサイズをギガバイト単位で指定します。デフォルト値は 100 です。
--storage-class string	Central データベースのストレージクラス名を指定します。デフォルトのストレージクラスが設定されている場合、これはオプションです。

8.2.6.7. roxctl central db generate openshift

Red Hat OpenShift クラスタに Central データベースインスタンスをデプロイするための OpenShift YAML マニフェストを生成します。

使用方法

```
$ roxctl central db generate openshift [flags]
```

表8.13 オプション

オプション	説明
--central-db-image string	使用する Central データベースイメージを指定します。指定しない場合は、 --image-defaults に対応するデフォルト値が使用されます。
--image-defaults string	コンテナイメージのデフォルト設定を指定します。イメージのダウンロード元のリポジトリ、イメージ名、タグの形式を制御します。デフォルト値は、 development_build です。
--openshift-version int	デプロイメントに Red Hat OpenShift メジャーバージョン 3 または 4 を指定します。デフォルト値は 3 です。
--output-dir output-directory	デプロイメントバンドルを保存するディレクトリーを指定します。デフォルト値は central-db-bundle です。

8.2.6.8. roxctl central db generate k8s hostpath

Central のホストパスボリュームタイプを使用して、データベースデプロイメント用の Kubernetes YAML マニフェストを生成します。

使用方法

```
$ roxctl central db generate k8s hostpath [flags]
```

表8.14 オプション

オプション	説明
<code>--hostpath string</code>	ホスト上のパスを指定します。デフォルト値は <code>/var/lib/stackrox-central-db</code> です。
<code>--node-selector-key string</code>	ノードセレクターキーを指定します。有効な値には、 kubernetes.io および hostname が含まれます。
<code>--node-selector-value string</code>	ノードセレクターの値を指定します。

8.2.6.9. roxctl central db generate openshift pvc

Central で永続ボリューム要求 (PVC) を使用してデータベースデプロイメント用の OpenShift YAML マニフェストを生成します。

使用方法

```
$ roxctl central db generate openshift pvc [flags]
```

表8.15 オプション

オプション	説明
<code>--name string</code>	Central データベースの外部ボリューム名を指定します。デフォルト値は central-db です。
<code>--size uint32</code>	Central データベースの外部ボリュームのサイズをギガバイト単位で指定します。デフォルト値は 100 です。
<code>--storage-class string</code>	Central データベースのストレージクラス名を指定します。デフォルトのストレージクラスが設定されている場合、これはオプションです。

8.2.6.10. roxctl central db generate openshift hostpath

ホストパス外部ボリュームを Central データベースに追加します。

使用方法

```
$ roxctl central db generate openshift hostpath [flags]
```

表8.16 オプション

オプション	説明
<code>--hostpath string</code>	ホスト上のパスを指定します。デフォルト値は <code>/var/lib/stackrox-central-db</code> です。

オプション	説明
--node-selector-key string	ノードセレクターキーを指定します。有効な値には、 kubernetes.io および hostname が含まれます。
--node-selector-value string	ノードセレクターの値を指定します。

8.2.7. roxctl central debug

Central サービスをデバッグします。

使用方法

```
$ roxctl central debug [flags]
```

8.2.7.1. roxctl central debug db

データベースのデバッグを制御します。

使用方法

```
$ roxctl central debug db [flags]
```

表8.17 オプション

オプション	説明
-t, --timeout duration	リクエストの最大期間を表す API リクエストのタイムアウトを指定します。デフォルト値は 1m0s です。

8.2.7.2. roxctl central debug log

現在のログレベルを取得します。

使用方法

```
$ roxctl central debug log [flags]
```

表8.18 オプション

オプション	説明
-l, --level string	モジュールを設定するログレベルを指定します。有効な値には、 Debug 、 Info 、 Warn 、 Error 、 Panic 、および Fatal が含まれます。

オプション	説明
-m, --modules strings	コマンドを適用するモジュールを指定します。
--retry-timeout duration	API リクエストが再試行されるまでのタイムアウトを指定します。値が0の場合、再試行せずにリクエスト期間全体を待機することを意味します。デフォルト値は 20s です。
-t, --timeout duration	API リクエストのタイムアウトを指定します。これは、リクエストの最大期間です。デフォルト値は 1m0s です。

8.2.7.3. roxctl central debug dump

Central のデバッグ情報を含むバンドルをダウンロードします。

使用方法

```
$ roxctl central debug dump [flags]
```

表8.19 オプション

オプション	説明
--logs	true に設定すると、ログは Central ダンプに含まれます。デフォルト値は false です。
--output-dir string	バンドルコンテンツの出力ディレクトリーを指定します。デフォルト値は、現在のディレクトリー内で自動的に生成されたディレクトリー名です。
-t, --timeout duration	API リクエストのタイムアウトを指定します。これは、リクエストの最大期間です。デフォルト値は 5m0s です。

8.2.7.4. roxctl central debug db stats

Central データベースの統計を制御します。

使用方法

```
$ roxctl central debug db stats [flags]
```

8.2.7.5. roxctl central debug authz-trace

デバッグの目的で、Central での承認トレースを有効または無効にします。

使用方法

■

```
$ roxctl central debug authz-trace [flags]
```

表8.20 オプション

オプション	説明
-t, --timeout duration	リクエストの最大期間を表す API リクエストのタイムアウトを指定します。デフォルト値は 20m0s です。

8.2.7.6. roxctl central debug db stats reset

Central データベースの統計をリセットします。

使用方法

```
$ roxctl central debug db stats reset [flags]
```

8.2.7.7. roxctl central debug download-diagnostics

プラットフォームに関する診断情報のスナップショットを含むバンドルをダウンロードします。

使用方法

```
$ roxctl central debug download-diagnostics [flags]
```

表8.21 オプション

オプション	説明
--clusters strings	ログを収集する Sensor クラスターのコンマ区切りのリストを指定します。
--output-dir string	診断バンドルを保存する出力ディレクトリーを指定します。
--since string	Sensor クラスターからログを収集するタイムスタンプを指定します。
-t, --timeout duration	API リクエストのタイムアウトを指定します。これにより、リクエストの最大期間が指定されます。デフォルト値は 5m0s です。

8.2.8. roxctl central generate

Central をデプロイするためのオーケストレーターオブジェクトを含む必要な YAML 設定ファイルを生成します。

使用方法

```
$ roxctl central generate [flags]
```

表8.22 オプション

オプション	説明
--backup-bundle string	キーと証明書を復元するバックアップバンドルへのパスを指定します。
--debug	true に設定すると、テンプレートはローカルファイルシステムから読み取られます。デフォルト値は false です。
--debug-path string	ローカルファイルシステム上の Helm テンプレートへのパスを指定します。詳細は、 roxctl central generate --help コマンドを実行してください。
--default-tls-certfile	デフォルトとして使用する PEM 証明書バンドルファイルを指定します。
--default-tls-keyfile	デフォルトとして使用する PEM 秘密鍵ファイルを指定します。
--enable-pod-security-policies	true に設定すると、 PodSecurityPolicy リソースが作成されます。デフォルト値は true です。
-p, --password string	管理者パスワードを指定します。デフォルト値は自動的に生成されます。
--plaintext-endpoints string	暗号化されていない公開に使用するポートまたはエンドポイントをコンマ区切りのリストとして指定します。

8.2.8.1. roxctl central generate k8s

Central を Kubernetes クラスターにデプロイするために必要な YAML 設定ファイルを生成します。

使用方法

```
$ roxctl central generate k8s [flags]
```

表8.23 オプション

オプション	説明
--central-db-image string	使用する Central データベースイメージを指定します。指定しない場合は、 --image-defaults に対応するデフォルト値が使用されます。

オプション	説明
--declarative-config-config-maps strings	Central で宣言型設定マウントとして追加する設定マップのリストを指定します。
--declarative-config-secrets strings	Central で宣言型設定マウントとして追加するシークレットのリストを指定します。
--enable-telemetry	Telemetry を有効にするか指定します。デフォルト値は false です。
--image-defaults string	コンテナイメージのデフォルト設定を指定します。指定された設定は、イメージのダウンロード元のリポジトリ、イメージ名、タグの形式を制御します。デフォルト値は、 development_build です。
--istio-support version	指定された Istio バージョンをサポートするデプロイメントファイルを生成します。有効な値は、 1.0 、 1.1 、 1.2 、 1.3 、 1.4 、 1.5 、 1.6 、および 1.7 です。
--lb-type load balancer type	Central を一時停止する方法を指定します。有効な値には、 lb 、 np 、 none が含まれます。デフォルト値は none です。
-i, --main-image string	使用するメインイメージを指定します。指定しない場合は、 --image-defaults に対応するデフォルト値が使用されます。
--offline	インターネットへの接続を避けて、RHACS をオフラインモードで実行するかどうかを指定します。デフォルト値は false です。
--output-dir output directory	デプロイメントバンドルを保存するディレクトリを指定します。デフォルト値は central-bundle です。

オプション	説明
<code>--output-format output format</code>	使用するデプロイメントツールを指定します。有効な値には、 kubectl 、 helm 、 helm-values などがあります。デフォルト値は kubectl です。
<code>--scanner-db-image string</code>	使用する Scanner データベースイメージを指定します。指定しない場合は、 --image-defaults に対応するデフォルト値が使用されます。
<code>--scanner-image string</code>	使用する Scanner イメージを指定します。指定しない場合は、 <code>--image-defaults</code> に対応するデフォルト値が使用されます。

8.2.8.2. roxctl central generate k8s pvc

Central で永続ボリューム要求 (PVC) 用の Kubernetes YAML ファイルを生成します。

使用方法

```
$ roxctl central generate k8s pvc [flags]
```

表8.24 オプション

オプション	説明
<code>--db-name string</code>	Central データベースの外部ボリューム名を指定します。デフォルト値は central-db です。
<code>--db-size uint32</code>	Central データベースの外部ボリュームのサイズをギガバイト単位で指定します。デフォルト値は 100 です。
<code>--db-storage-class string</code>	Central データベースのストレージクラス名を指定します。デフォルトのストレージクラスが設定されている場合、これはオプションです。
<code>--name string</code>	Central の外部ボリューム名を指定します。デフォルト値は stackrox-db です。
<code>--size uint32</code>	Central の外部ボリュームのサイズをギガバイト単位で指定します。デフォルト値は 100 です。

オプション	説明
--storage-class string	Central のストレージクラス名を指定します。デフォルトのストレージクラスが設定されている場合、これはオプションです。

8.2.8.3. roxctl central generate openshift

Central を Red Hat OpenShift クラスタにデプロイするために必要な YAML 設定ファイルを生成します。

使用方法

```
$ roxctl central generate openshift [flags]
```

表8.25 オプション

オプション	説明
--central-db-image string	使用する Central データベースイメージを指定します。指定しない場合、 --image-defaults に対応するデフォルト値が作成されます。
--declarative-config-config-maps strings	Central で宣言型設定マウントとして追加する設定マップのリストを指定します。
--declarative-config-secrets strings	Central で宣言型設定マウントとして追加するシークレットのリストを指定します。
--enable-telemetry	Telemetry を有効にするか指定します。デフォルト値は false です。
--image-defaults string	コンテナイメージのデフォルト設定を指定します。イメージのダウンロード元のリポジトリ、イメージ名、タグの形式を制御します。デフォルト値は、 development_build です。

オプション	説明
--istio-support version	指定された Istio バージョンをサポートするデプロイメントファイルを生成します。有効な値は、 1.0 、 1.1 、 1.2 、 1.3 、 1.4 、 1.5 、 1.6 、および 1.7 です。
--lb-type load balancer type	Central を公開する方法を指定します。有効な値には、 route 、 lb 、 np 、および none が含まれます。デフォルト値は none です。
-i, --main-image string	使用するメインイメージを指定します。指定しない場合は、 --image-defaults に対応するデフォルト値が使用されます。
--offline	インターネットへの接続を避けて、RHACS をオフラインモードで実行するかどうかを指定します。デフォルト値は false です。
--openshift-monitoring false true auto[=true]	Red Hat OpenShift 4 モニタリングとの統合を指定します。デフォルト値は auto です。
--openshift-version int	デプロイメントに Red Hat OpenShift メジャーバージョン 3 または 4 を指定します。
--output-dir output directory	デプロイメントバンドルを保存するディレクトリーを指定します。デフォルト値は central-bundle です。
--output-format output format	使用するデプロイメントツールを指定します。有効な値には、 kubectl 、 helm 、および helm-values が含まれます。デフォルト値は kubectl です。
--scanner-db-image string	使用する Scanner データベースイメージを指定します。指定しない場合は、 --image-defaults に対応するデフォルト値が使用されます。

オプション	説明
--scanner-image string	使用する Scanner イメージを指定します。指定しない場合は、 --image-defaults に対応するデフォルト値が使用されます。

8.2.8.4. roxctl central generate interactive

Central でインタラクティブなリソースを生成します。

使用方法

```
$ roxctl central generate interactive [flags]
```

8.2.8.5. roxctl central generate k8s hostpath

ホストパスボリュームタイプを使用して、Central インスタンスをデプロイするための Kubernetes YAML マニフェストを生成します。

使用方法

```
$ roxctl central generate k8s hostpath [flags]
```

表8.26 オプション

オプション	説明
--db-hostpath string	Central データベースのホスト上のパスを指定します。デフォルト値は /var/lib/stackrox-central です。
--db-node-selector-key string	Central データベースのノードセレクターキーを指定します。有効な値には、 kubernetes.io および hostname が含まれます。
--db-node-selector-value string	Central データベースのノードセレクター値を指定します。
--hostpath string	ホスト上のパスを指定します。デフォルト値は /var/lib/stackrox です。
--node-selector-key string	ノードセレクターキーを指定します。有効な値には、 kubernetes.io および hostname が含まれます。

オプション	説明
--node-selector-value string	ノードセレクターの値を指定します。

8.2.8.6. roxctl central generate openshift pvc

Central に永続ボリューム要求 (PVC) をデプロイするための OpenShift YAML マニフェストを生成します。

使用方法

```
$ roxctl central generate openshift pvc [flags]
```

表8.27 オプション

オプション	説明
--db-name string	Central データベースの外部ボリューム名を指定します。デフォルト値は central-db です。
--db-size uint32	Central データベースの外部ボリュームのサイズをギガバイト単位で指定します。デフォルト値は 100 です。
--db-storage-class string	Central データベースのストレージクラス名を指定します。デフォルトのストレージクラスが設定されている場合、これはオプションです。
--name string	Central の外部ボリューム名を指定します。デフォルト値は stackrox-db です。
--size uint32	Central の外部ボリュームのサイズをギガバイト単位で指定します。デフォルト値は 100 です。
--storage-class string	Central のストレージクラス名を指定します。デフォルトのストレージクラスが設定されている場合、これはオプションです。

8.2.8.7. roxctl central generate openshift hostpath

Red Hat OpenShift のデプロイメント定義にホストパス外部ボリュームを追加します。

使用方法

```
$ roxctl central generate openshift hostpath [flags]
```

表8.28 オプション

オプション	説明
--db-hostpath string	Central データベースのホスト上のパスを指定します。デフォルト値は /var/lib/stackrox-central です。
--db-node-selector-key string	ノードセレクターキーを指定します。有効な値には、 kubernetes.io と Central データベースの hostname が含まれます。
--db-node-selector-value string	Central データベースのノードセレクター値を指定します。
--hostpath string	ホスト上のパスを指定します。デフォルト値は /var/lib/stackrox です。
--node-selector-key string	ノードセレクターキーを指定します。有効な値には、 kubernetes.io および hostname が含まれます。
--node-selector-value string	ノードセレクターの値を指定します。

8.2.9. roxctl central init-bundles

Central でバンドルを初期化します。

使用方法

```
$ roxctl central init-bundles [flag]
```

表8.29 オプション

オプション	説明
--retry-timeout duration	API リクエストが再試行されるまでのタイムアウトを指定します。値 0 は、リクエスト期間全体が再試行せずに待機されることを意味します。デフォルト値は 20s です。
-t, --timeout duration	リクエストの最大期間を表す API リクエストのタイムアウトを指定します。デフォルト値は 1m0s です。

8.2.9.1. roxctl central init-bundles list

Central で利用可能な初期化バンドルをリスト表示します。

使用方法

```
$ roxctl central init-bundles list [flags]
```

8.2.9.2. roxctl central init-bundles revoke

Central で1つ以上のクラスター初期化バンドルを取り消します。

使用方法

```
$ roxctl central init-bundles revoke <init_bundle_ID or name> [<init_bundle_ID or name> ...] [flags]
```

1

- 1 **<init_bundle_ID or name>** には、取り消す初期化バンドルの ID または名前を指定します。複数の ID または名前をスペースで区切って指定できます。

8.2.9.3. roxctl central init-bundles fetch-ca

Central から認証局 (CA) バンドルを取得します。

使用方法

```
$ roxctl central init-bundles fetch-ca [flags]
```

表8.30 オプション

オプション	説明
--output string	CA 設定の保存に使用するファイルを指定します。

8.2.9.4. roxctl central init-bundles generate

新しいクラスター初期化バンドルを生成します。

使用方法

```
$ roxctl central init-bundles generate <init_bundle_name> [flags] 1
```

- 1 **<init_bundle_name>** には、生成する初期化バンドルの名前を指定します。

表8.31 オプション

オプション	説明
-------	----

オプション	説明
--output string	Helm 設定フォームで、新しく生成された初期化バンドルを保存するために使用するファイルを指定します。- を使用して標準出力を生成できます。
--output-secrets string	Kubernetes シークレット形式で新しく生成された初期化バンドルを保存するために使用するファイルを指定します。- を使用して標準を生成できます。

8.2.10. roxctl central userpki

ユーザー証明書認可プロバイダーを管理します。

使用方法

```
$ roxctl central userpki [flags]
```

8.2.10.1. roxctl central userpki list

すべてのユーザー証明書認証プロバイダーを表示します。

使用方法

```
$ roxctl central userpki list [flags]
```

表8.32 オプション

オプション	説明
-j, --json	JSON 出力を有効にします。デフォルト値は false です。
--retry-timeout duration	API リクエストが再試行されるまでのタイムアウトを指定します。値が 0 の場合、再試行せずにリクエスト期間全体を待機することを意味します。デフォルト値は 20s です。
-t, --timeout duration	リクエストの最大期間を表す API リクエストのタイムアウトを指定します。デフォルト値は 1m0s です。

8.2.10.2. roxctl central userpki create

新しいユーザー証明書認証プロバイダーを作成します。

使用方法

```
$ roxctl central userpki create name [flags]
```

表8.33 オプション

オプション	説明
-c, --cert strings	ルート CA 証明書の PEM ファイルを指定します。複数の証明書ファイルを指定できます。
--retry-timeout duration	API リクエストが再試行されるまでのタイムアウトを指定します。値が 0 の場合、再試行せずにリクエスト期間全体を待機することを意味します。デフォルト値は 20s です。
-r, --role string	このプロバイダーのユーザーの最小アクセスロールを指定します。
-t, --timeout duration	リクエストの最大期間を表す API リクエストのタイムアウトを指定します。デフォルト値は 1m0s です。

8.2.10.3. roxctl central userpki delete

ユーザー証明書認証プロバイダーを削除します。

使用方法

```
$ roxctl central userpki delete id|name [flags]
```

表8.34 オプション

オプション	説明
-f, --force	true に設定すると、確認なしで削除が続行されます。デフォルト値は false です。
--retry-timeout duration	API リクエストが再試行されるまでのタイムアウトを指定します。値が 0 の場合、再試行せずにリクエスト期間全体を待機することを意味します。デフォルト値は 20s です。
-t, --timeout duration	リクエストの最大期間を表す API リクエストのタイムアウトを指定します。デフォルト値は 1m0s です。

8.3. ROXCTL クラスター

クラスターに関連するコマンド。

使用方法

```
$ roxctl cluster [command] [flags]
```

表8.35 利用可能なコマンド

コマンド	説明
<code>delete</code>	Central から Sensor を削除します。

表8.36 オプション

オプション	説明
<code>--retry-timeout duration</code>	API リクエストの再試行タイムアウトを設定します。値 0 は、リクエスト期間全体が再試行なしで待機されることを意味します。デフォルト値は 20s です。
<code>-t, --timeout duration</code>	リクエストの最大期間を表す API リクエストのタイムアウトを設定します。デフォルト値は 1m0s です。

8.3.1. 親コマンドから継承された roxctl cluster コマンドオプション

`roxctl cluster` コマンドは、`roxctl` の親コマンドから継承された次のオプションをサポートします。

オプション	説明
<code>--ca string</code>	安全な接続のためのカスタム CA 証明書ファイルパスを指定します。あるいは、 ROX_CA_CERT_FILE 環境変数を使用して、ファイルパスを指定することもできます。
<code>--direct-grpc</code>	接続パフォーマンスを向上させるには、 <code>--direct-grpc</code> を設定します。あるいは、 ROX_DIRECT_GRPC_CLIENT 環境変数を true に設定することで、直接 gRPC を有効にできます。デフォルト値は false です。
<code>-e, --endpoint string</code>	サービスが接続するエンドポイントを設定します。あるいは、 ROX_ENDPOINT 環境変数を使用して、エンドポイントを設定することもできます。デフォルト値は localhost:8443 です。
<code>--force-http1</code>	すべての接続で HTTP/1 を強制的に使用します。あるいは、 ROX_CLIENT_FORCE_HTTP1 環境変数を true に設定することで、HTTP/1 の使用を強制できます。デフォルト値は false です。

オプション	説明
--insecure	安全でない接続オプションを有効にします。あるいは、 ROX_INSECURE_CLIENT 環境変数を true に設定することで、安全でない接続オプションを有効にすることができます。デフォルト値は false です。
--insecure-skip-tls-verify	TLS 証明書の検証をスキップします。あるいは、 ROX_INSECURE_CLIENT_SKIP_TLS_VERIFY 環境変数を true に設定することで、TLS 証明書の検証をスキップできます。デフォルト値は false です。
--no-color	カラー出力を無効にします。あるいは、 ROX_NO_COLOR 環境変数を true に設定することで、カラー出力を無効にすることができます。デフォルト値は false です。
-p, --password string	Basic 認証のパスワードを指定します。あるいは、 ROX_ADMIN_PASSWORD 環境変数を使用して、パスワードを設定することもできます。
--plaintext	暗号化されていない接続を使用します。あるいは、 ROX_PLAINTEXT 環境変数を true に設定することで、暗号化されていない接続を有効にすることができます。デフォルト値は false です。
-s, --server-name string	SNI に使用する TLS サーバー名を設定します。あるいは、 ROX_SERVER_NAME 環境変数を使用して、サーバー名を設定することもできます。
--token-file string	指定されたファイルで提供される API トークンを認証に使用します。あるいは、 ROX_API_TOKEN 環境変数を使用して、トークンを設定することもできます。



注記

これらのオプションは、**roxctl cluster** コマンドのすべてのサブコマンドに適用されます。

8.3.2. roxctl cluster delete

Central から Sensor を削除します。

使用方法

```
$ roxctl cluster delete [flags]
```

表8.37 オプション

オプション	説明
--name string	削除するクラスター名を指定します。

8.4. ROXCTL COLLECTOR

Collector サービスに関連するコマンド。

使用方法

```
$ roxctl collector [command] [flags]
```

表8.38 利用可能なコマンド

コマンド	説明
support-packages	Collector のサポートパッケージをアップロードします。

8.4.1. 親コマンドから継承された roxctl collector コマンドオプション

roxctl collector コマンドは、**roxctl** の親コマンドから継承された次のオプションをサポートします。

オプション	説明
--ca string	安全な接続のためのカスタム CA 証明書ファイルパスを指定します。あるいは、 ROX_CA_CERT_FILE 環境変数を使用して、ファイルパスを指定することもできます。
--direct-grpc	接続パフォーマンスを向上させるには、 --direct-grpc を設定します。あるいは、 ROX_DIRECT_GRPC_CLIENT 環境変数を true に設定することで、直接 gRPC を有効にできます。デフォルト値は false です。
-e, --endpoint string	サービスが接続するエンドポイントを設定します。あるいは、 ROX_ENDPOINT 環境変数を使用して、エンドポイントを設定することもできます。デフォルト値は localhost:8443 です。
--force-http1	すべての接続で HTTP/1 を強制的に使用します。あるいは、 ROX_CLIENT_FORCE_HTTP1 環境変数を true に設定することで、HTTP/1 の使用を強制できます。デフォルト値は false です。

オプション	説明
--insecure	安全でない接続オプションを有効にします。あるいは、 ROX_INSECURE_CLIENT 環境変数を true に設定することで、安全でない接続オプションを有効にすることができます。デフォルト値は false です。
--insecure-skip-tls-verify	TLS 証明書の検証をスキップします。あるいは、 ROX_INSECURE_CLIENT_SKIP_TLS_VERIFY 環境変数を true に設定することで、TLS 証明書の検証をスキップできます。デフォルト値は false です。
--no-color	カラー出力を無効にします。あるいは、 ROX_NO_COLOR 環境変数を true に設定することで、カラー出力を無効にすることができます。デフォルト値は false です。
-p, --password string	Basic 認証のパスワードを指定します。あるいは、 ROX_ADMIN_PASSWORD 環境変数を使用して、パスワードを設定することもできます。
--plaintext	暗号化されていない接続を使用します。あるいは、 ROX_PLAINTEXT 環境変数を true に設定することで、暗号化されていない接続を有効にすることができます。デフォルト値は false です。
-s, --server-name string	SNI に使用する TLS サーバー名を設定します。あるいは、 ROX_SERVER_NAME 環境変数を使用して、サーバー名を設定することもできます。
--token-file string	指定されたファイルで提供される API トークンを認証に使用します。あるいは、 ROX_API_TOKEN 環境変数を使用して、トークンを設定することもできます。



注記

これらのオプションは、**roxctl collector** コマンドのすべてのサブコマンドに適用されません。

8.4.2. roxctl collector support-packages

Collector のサポートパッケージをアップロードします。



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。

使用方法

```
$ roxctl collector support-packages [flags]
```

8.4.2.1. roxctl collector support-packages upload

Collector サポートパッケージから Central にファイルをアップロードします。

使用方法

```
$ roxctl collector support-packages upload [flags]
```

表8.39 オプション

オプション	説明
--overwrite	既存の異なるファイルを上書きするかどうかを指定します。デフォルト値は false です。
--retry-timeout duration	API リクエストが再試行されるまでのタイムアウトを設定します。値が 0 の場合、再試行せずにリクエスト期間全体を待機することを意味します。デフォルト値は 20s です。
-t, --timeout duration	API リクエストのタイムアウトを設定します。このオプションは、リクエストの最大期間を表します。デフォルト値は 1m0s です。

8.5. ROXCTL COMPLETION

シェル補完スクリプトを生成します。

使用方法

```
$ roxctl completion [bash|zsh|fish|powershell]
```

表8.40 サポートされているシェルタイプ

シェルタイプ	説明
bash	Bash シェルの補完スクリプトを生成します。
zsh	Zsh シェルの補完スクリプトを生成します。

シェルタイプ	説明
fish	Fish シェルの補完スクリプトを生成します。
powershell	PowerShell シェルの補完スクリプトを生成します。

8.5.1. 親コマンドから継承された roxctl completion コマンドオプション

roxctl completion コマンドは、**roxctl** の親コマンドから継承された次のオプションをサポートします。

オプション	説明
--ca string	安全な接続のためのカスタム CA 証明書ファイルパスを指定します。あるいは、 ROX_CA_CERT_FILE 環境変数を使用して、ファイルパスを指定することもできます。
--direct-grpc	接続パフォーマンスを向上させるには、 --direct-grpc を設定します。あるいは、 ROX_DIRECT_GRPC_CLIENT 環境変数を true に設定することで、直接 gRPC を有効にできます。デフォルト値は false です。
-e, --endpoint string	サービスが接続するエンドポイントを設定します。あるいは、 ROX_ENDPOINT 環境変数を使用して、エンドポイントを設定することもできます。デフォルト値は localhost:8443 です。
--force-http1	すべての接続で HTTP/1 を強制的に使用します。あるいは、 ROX_CLIENT_FORCE_HTTP1 環境変数を true に設定することで、HTTP/1 の使用を強制できます。デフォルト値は false です。
--insecure	安全でない接続オプションを有効にします。あるいは、 ROX_INSECURE_CLIENT 環境変数を true に設定することで、安全でない接続オプションを有効にすることができます。デフォルト値は false です。
--insecure-skip-tls-verify	TLS 証明書の検証をスキップします。あるいは、 ROX_INSECURE_CLIENT_SKIP_TLS_VERIFY 環境変数を true に設定することで、TLS 証明書の検証をスキップできます。デフォルト値は false です。

オプション	説明
--no-color	カラー出力を無効にします。あるいは、 ROX_NO_COLOR 環境変数を true に設定することで、カラー出力を無効にすることができます。デフォルト値は false です。
-p, --password string	Basic 認証のパスワードを指定します。あるいは、 ROX_ADMIN_PASSWORD 環境変数を使用して、パスワードを設定することもできます。
--plaintext	暗号化されていない接続を使用します。あるいは、 ROX_PLAINTEXT 環境変数を true に設定することで、暗号化されていない接続を有効にすることができます。デフォルト値は false です。
-s, --server-name string	SNI に使用する TLS サーバー名を設定します。あるいは、 ROX_SERVER_NAME 環境変数を使用して、サーバー名を設定することもできます。
--token-file string	指定されたファイルで提供される API トークンを認証に使用します。あるいは、 ROX_API_TOKEN 環境変数を使用して、トークンを設定することもできます。

8.6. ROXCTL DECLARATIVE-CONFIG

宣言型設定を管理します。

使用方法

```
$ roxctl declarative-config [command] [flags]
```

表8.41 利用可能なコマンド

コマンド	説明
create	宣言型設定を作成します。
lint	既存の宣言型設定 YAML ファイルを lint します。

8.6.1. 親コマンドから継承された roxctl declarative-config コマンドオプション

roxctl declarative-config コマンドは、**roxctl** の親コマンドから継承された次のオプションをサポートします。

オプション	説明
--ca string	安全な接続のためのカスタム CA 証明書ファイルパスを指定します。あるいは、 ROX_CA_CERT_FILE 環境変数を使用して、ファイルパスを指定することもできます。
--direct-grpc	接続パフォーマンスを向上させるには、 --direct-grpc を設定します。あるいは、 ROX_DIRECT_GRPC_CLIENT 環境変数を true に設定することで、直接 gRPC を有効にできます。デフォルト値は false です。
-e, --endpoint string	サービスが接続するエンドポイントを設定します。あるいは、 ROX_ENDPOINT 環境変数を使用して、エンドポイントを設定することもできます。デフォルト値は localhost:8443 です。
--force-http1	すべての接続で HTTP/1 を強制的に使用します。あるいは、 ROX_CLIENT_FORCE_HTTP1 環境変数を true に設定することで、HTTP/1 の使用を強制できます。デフォルト値は false です。
--insecure	安全でない接続オプションを有効にします。あるいは、 ROX_INSECURE_CLIENT 環境変数を true に設定することで、安全でない接続オプションを有効にすることができます。デフォルト値は false です。
--insecure-skip-tls-verify	TLS 証明書の検証をスキップします。あるいは、 ROX_INSECURE_CLIENT_SKIP_TLS_VERIFY 環境変数を true に設定することで、TLS 証明書の検証をスキップできます。デフォルト値は false です。
--no-color	カラー出力を無効にします。あるいは、 ROX_NO_COLOR 環境変数を true に設定することで、カラー出力を無効にすることができます。デフォルト値は false です。
-p, --password string	Basic 認証のパスワードを指定します。あるいは、 ROX_ADMIN_PASSWORD 環境変数を使用して、パスワードを設定することもできます。
--plaintext	暗号化されていない接続を使用します。あるいは、 ROX_PLAINTEXT 環境変数を true に設定することで、暗号化されていない接続を有効にすることができます。デフォルト値は false です。

オプション	説明
-s, --server-name string	SNI に使用する TLS サーバー名を設定します。あるいは、 ROX_SERVER_NAME 環境変数を使用して、サーバー名を設定することもできます。
--token-file string	指定されたファイルで提供される API トークンを認証に使用します。あるいは、 ROX_API_TOKEN 環境変数を使用して、トークンを設定することもできます。



注記

これらのオプションは、**roxctl declarative-config** コマンドのすべてのサブコマンドに適用されます。

8.6.2. roxctl declarative-config lint

既存の宣言型設定 YAML ファイルを lint します。

使用方法

```
$ roxctl declarative-config lint [flags]
```

表8.42 オプション

オプション	説明
--config-map string	--config-map string から宣言型設定を読み取ります。指定しない場合、設定は --file フラグを使用して指定された YAML ファイルから読み取られます。
-f, --file string	YAML 形式の宣言型設定を含むファイル。
--namespace string	設定マップの --namespace string から宣言型設定を読み取ります。指定しない場合、現在の Kubernetes 設定コンテキストで指定された namespace が使用されます。
--secret string	指定された --secret string から、宣言型設定を読み取ります。指定しない場合、設定は --file フラグを使用して指定された YAML ファイルから読み取られます。

8.6.3. roxctl declarative-config create

宣言型設定を作成します。

使用方法

```
$ roxctl declarative-config create [flags]
```

表8.43 オプション

オプション	説明
--config-map string	設定マップに宣言型設定 YAML を書き込みます。指定されず、 --secret フラグも指定されていない場合、生成された YAML は標準出力形式で出力されます。
--namespace string	宣言型設定 YAML を設定マップまたはシークレットに書き込む場合は必須です。指定しない場合は、現在の Kubernetes 設定のデフォルトの namespace が使用されます。
--secret string	Secret に宣言型設定 YAML を記述します。機密データにはシークレットを使用する必要があります。指定されず、 --config-map フラグも指定されていない場合、生成された YAML は標準出力形式で出力されます。

8.6.3.1. roxctl declarative-config create role

ロールの宣言型設定を作成します。

使用方法

```
$ roxctl declarative-config create role [flags]
```

表8.44 オプション

オプション	説明
--access-scope string	名前を指定すると、参照されるアクセス範囲を指定できます。
--description string	ロールの説明を設定します。
--name string	ロールの名前を指定します。
--permission-set string	名前を指定すると、参照される権限セットを指定できます。

8.6.3.2. roxctl declarative-config create notifier

Notifier の宣言型設定を作成します。

使用方法

```
$ roxctl declarative-config create notifier [flags]
```

表8.45 オプション

オプション	説明
--name string	Notifier の名前を指定します。

8.6.3.3. roxctl declarative-config create access-scope

アクセススコープの宣言型設定を作成します。

使用方法

```
$ roxctl declarative-config create access-scope [flags]
```

表8.46 オプション

オプション	説明
--cluster-label-selector requirement	クラスターのラベルに基づいて、ラベルセレクターを作成するための基準を指定します。キーと値のペアは要件を表し、このフラグを複数回使用して要件の組み合わせを作成できます。デフォルト値は [[]] です。詳細は、 roxctl declarative-config create access-scope --help コマンドを実行してください。
--description string	アクセススコープの説明を設定します。
--included included-object	アクセススコープに含めるクラスターとその namespace のリストを指定します。デフォルト値は [null] です。
--name string	アクセススコープの名前を指定します。
--namespace-label-selector requirement	namespace のラベルに基づいてラベルセレクターを作成するための基準を指定します。cluster-label-selector と同様に、要件の組み合わせに対してこのフラグを複数回使用できます。詳細は、 roxctl declarative-config create access-scope --help コマンドを実行してください。

8.6.3.4. roxctl declarative-config create auth-provider

認証プロバイダーの宣言型設定を作成します。

使用方法

```
$ roxctl declarative-config create auth-provider [flags]
```

表8.47 オプション

オプション	説明
--extra-ui-endpoints strings	認証プロバイダーが使用される追加のユーザーインターフェイス (UI) エンドポイントを指定します。予期される形式は <endpoint>:<port> です。
--groups-key strings	認証プロバイダー内に追加するグループのキーを設定します。キー、値、ロールのタプルは同じ長さである必要があります。詳細は、 roxctl declarative-config create auth-provider --help コマンドを実行してください。
--groups-role strings	認証プロバイダー内に追加するグループのロールを設定します。キー、値、ロールのタプルは同じ長さである必要があります。詳細は、 roxctl declarative-config create auth-provider --help コマンドを実行してください。
--groups-value strings	認証プロバイダー内に追加するグループの値を設定します。キー、値、ロールのタプルは同じ長さである必要があります。詳細は、 roxctl declarative-config create auth-provider --help コマンドを実行してください。
--minimum-access-role string	認証プロバイダーの最小アクセスロールを設定します。宣言的設定を使用して最小アクセスロールを設定しない場合は、このフィールドを空のままにすることができます。
--name string	認証プロバイダーの名前を指定します。

オプション	説明
--required-attributes stringToString	認証プロバイダーが認証中に返す必要がある属性のリストを設定します。デフォルト値は [] です。
--ui-endpoint string	認証プロバイダーが使用される UI エンドポイントを設定します。これは通常、RHACS が利用できるパブリックエンドポイントです。予期される形式は <endpoint>:<port> です。

8.6.3.5. roxctl declarative-config create permission-set

権限セットの宣言型設定を作成します。

使用方法

```
$ roxctl declarative-config create permission-set [flags]
```

表8.48 オプション

オプション	説明
--description string	権限セットの説明を設定します。
--name string	権限セットの名前を指定します。
--resource-with-access stringToString	リソースのリストとそれぞれのアクセスレベルを設定します。デフォルト値は [] です。詳細は、 roxctl declarative-config create permission-set --help コマンドを実行してください。

8.6.3.6. roxctl declarative-config create notifier splunk

Splunk Notifier の宣言型設定を作成します。

使用方法

```
$ roxctl declarative-config create notifier splunk [flags]
```

表8.49 オプション

オプション	説明
--audit-logging	監査ロギングを有効にします。デフォルト値は false です。
--source-types stringToString	Splunk ソースタイプをコンマ区切りの key=value ペアとして指定します。デフォルト値は [] です。
--splunk-endpoint string	Splunk HTTP エンドポイントを指定します。これは必須のオプションです。
--splunk-skip-tls-verify	Splunk への安全でない接続を使用します。デフォルト値は false です。
--splunk-token string	Splunk HTTP トークンを指定します。これは必須のオプションです。
--truncate int	Splunk の切り捨て制限を指定します。デフォルト値は 10000 です。

8.6.3.7. roxctl declarative-config create notifier generic

汎用 Notifier の宣言型設定を作成します。

使用方法

```
$ roxctl declarative-config create notifier generic [flags]
```

表8.50 オプション

オプション	説明
--audit-logging	監査ロギングを有効にします。デフォルト値は false です。
--extra-fields stringToString	追加のフィールドをコンマ区切りの key=value ペアとして指定します。デフォルト値は [] です。
--headers stringToString	ヘッダーをコンマ区切りの key=value ペアとして指定します。デフォルト値は [] です。
--webhook-cacert-file string	エンドポイント CA 証明書のファイル名を PEM 形式で指定します。
--webhook-endpoint string	Webhook エンドポイントの URL を指定します。

オプション	説明
--webhook-password string	Webhook エンドポイントの Basic 認証用のパスワードを指定します。指定されない場合、認証は行われません。 --webhook-username が必要です。
--webhook-skip-tls-verify	Webhook TLS 検証をスキップします。デフォルト値は false です。
--webhook-username string	Webhook エンドポイントの Basic 認証のユーザー名を指定します。指定しない場合、認証は行われません。 --webhook-password が必要です。

8.6.3.8. roxctl declarative-config create auth-provider iap

Identity-Aware Proxy (IAP) 識別子を使用して、認証プロバイダーの宣言型設定を作成します。

使用方法

```
$ roxctl declarative-config create auth-provider iap [flags]
```

表8.51 オプション

オプション	説明
--audience string	検証するターゲットグループを指定します。

8.6.3.9. roxctl declarative-config create auth-provider oidc

OpenID Connect (OIDC) 認証プロバイダーの宣言型設定を作成します。

使用方法

```
$ roxctl declarative-config create auth-provider oidc [flags]
```

表8.52 オプション

オプション	説明
--claim-mappings stringToString	認証プロバイダーのルールに含めるアイデンティティプロバイダー (IdP) トークンからの非標準クレームのリストを指定します。デフォルト値は [] です。
--client-id string	OIDC クライアントのクライアント ID を指定します。

オプション	説明
--client-secret string	OIDC クライアントのクライアントシークレットを指定します。
--disable-offline-access	OIDC IdP からの <code>offline_access</code> のリクエストを無効にします。OIDC IdP が offline_access スコープのセッション数を制限する場合は、このオプションを使用する必要があります。デフォルト値は false です。
--issuer string	OIDC クライアントの発行者を指定します。
--mode string	使用するコールバックモードを指定します。有効な値には、 auto 、 post 、 query 、および fragment が含まれます。デフォルト値は auto です。

8.6.3.10. roxctl declarative-config create auth-provider saml

SAML 認証プロバイダーの宣言型設定を作成します。

使用方法

```
$ roxctl declarative-config create auth-provider saml [flags]
```

表8.53 オプション

オプション	説明
--idp-cert string	SAML アイデンティティプロバイダー (IdP) 証明書を含むファイルを PEM 形式で指定します。
--idp-issuer string	IdP の発行者を指定します。
--metadata-url string	サービスプロバイダーのメタデータ URL を指定します。
--name-id-format string	名前 ID の形式を指定します。
--sp-issuer string	サービスプロバイダーの発行者を指定します。
--sso-url string	シングルサインオン (SSO) 用の IdP の URL を指定します。

8.6.3.11. roxctl declarative-config create auth-provider userpki

ユーザー PKI 認証プロバイダーの宣言型設定を作成します。

使用方法

```
$ roxctl declarative-config create auth-provider userpki [flags]
```

表8.54 オプション

オプション	説明
<code>--ca-file string</code>	認証局を含むファイルを PEM 形式で指定します。

8.6.3.12. roxctl declarative-config create auth-provider openshift-auth

OpenShift Container Platform OAuth 認証プロバイダーの宣言型設定を作成します。

使用方法

```
$ roxctl declarative-config create auth-provider openshift-auth [flags]
```

8.7. ROXCTL DEPLOYMENT

デプロイメントに関連するコマンド。

使用方法

```
$ roxctl deployment [command] [flags]
```

表8.55 利用可能なコマンド

コマンド	説明
<code>check</code>	デプロイメントにデプロイメント時間ポリシーの違反がないか確認します。

表8.56 オプション

オプション	説明
<code>-t, --timeout duration</code>	API リクエストのタイムアウトを設定します。このオプションは、リクエストの最大期間を表します。デフォルト値は 10m0s です。

8.7.1. 親コマンドから継承された roxctl deployment コマンドオプション

`roxctl deployment` コマンドは、`roxctl` の親コマンドから継承された次のオプションをサポートします。

オプション	説明
--ca string	安全な接続のためのカスタム CA 証明書ファイルパスを指定します。あるいは、 ROX_CA_CERT_FILE 環境変数を使用して、ファイルパスを指定することもできます。
--direct-grpc	接続パフォーマンスを向上させるには、 --direct-grpc を設定します。あるいは、 ROX_DIRECT_GRPC_CLIENT 環境変数を true に設定することで、直接 gRPC を有効にできます。デフォルト値は false です。
-e, --endpoint string	サービスが接続するエンドポイントを設定します。あるいは、 ROX_ENDPOINT 環境変数を使用して、エンドポイントを設定することもできます。デフォルト値は localhost:8443 です。
--force-http1	すべての接続で HTTP/1 を強制的に使用します。あるいは、 ROX_CLIENT_FORCE_HTTP1 環境変数を true に設定することで、HTTP/1 の使用を強制できます。デフォルト値は false です。
--insecure	安全でない接続オプションを有効にします。あるいは、 ROX_INSECURE_CLIENT 環境変数を true に設定することで、安全でない接続オプションを有効にすることができます。デフォルト値は false です。
--insecure-skip-tls-verify	TLS 証明書の検証をスキップします。あるいは、 ROX_INSECURE_CLIENT_SKIP_TLS_VERIFY 環境変数を true に設定することで、TLS 証明書の検証をスキップできます。デフォルト値は false です。
--no-color	カラー出力を無効にします。あるいは、 ROX_NO_COLOR 環境変数を true に設定することで、カラー出力を無効にすることができます。デフォルト値は false です。
-p, --password string	Basic 認証のパスワードを指定します。あるいは、 ROX_ADMIN_PASSWORD 環境変数を使用して、パスワードを設定することもできます。
--plaintext	暗号化されていない接続を使用します。あるいは、 ROX_PLAINTEXT 環境変数を true に設定することで、暗号化されていない接続を有効にすることができます。デフォルト値は false です。

オプション	説明
-s, --server-name string	SNI に使用する TLS サーバー名を設定します。あるいは、 ROX_SERVER_NAME 環境変数を使用して、サーバー名を設定することもできます。
--token-file string	指定されたファイルで提供される API トークンを認証に使用します。あるいは、 ROX_API_TOKEN 環境変数を使用して、トークンを設定することもできます。



注記

これらのオプションは、**roxctl deployment** コマンドのすべてのサブコマンドに適用されます。

8.7.2. roxctl deployment check

デプロイメント時間ポリシーの違反がないか、デプロイメントを確認します。

使用方法

```
$ roxctl deployment check [flags]
```

表8.57 オプション

オプション	説明
-c, --categories strings	実行するポリシーカテゴリーを定義します。デフォルトでは、すべてのポリシーカテゴリーが実行されます。
--cluster string	評価のコンテキストとして使用するクラスター名または ID を設定して、クラスター固有の情報を使用した拡張デプロイメントを有効にします。
--compact-output	JSON 出力をコンパクトな形式で出力します。デフォルト値は false です。
-f, --file stringArray	ポリシー評価のために Central に送信する YAML ファイルを指定します。
--force	イメージの Central キャッシュをバイパスし、強制的に Scanner から新たにプルします。デフォルト値は false です。

オプション	説明
--headers strings	表形式で出力するヘッダーを定義します。デフォルト値には、 POLICY 、 SEVERITY 、 BREAKS DEPLOY 、 DEPLOYMENT 、 DESCRIPTION 、 VIOLATION 、および REMEDIATION が含まれます。
--headers-as-comments	CSV 表形式出力にヘッダーをコメントとして出力します。デフォルト値は false です。
--junit-suite-name string	JUnit テストスイートの名前を設定します。デフォルト値は deployment-check です。
--merge-output	表形式の出力で重複したセルをマージします。デフォルト値は false です。
-n, --namespace string	namespace を指定して、仕様に namespace がないデプロイメントのネットワークポリシー、RBAC、サービスなどのコンテキスト情報を使用し、デプロイメントを強化します。仕様で定義されている namespace は変更されません。デフォルト値は default です。
--no-header	表形式の出力では、ヘッダーを出力しないでください。デフォルト値は false です。
-o, --output string	出力形式を選択します。出力形式には、 json 、 junit 、 sarif 、 table 、および csv が含まれます。デフォルト値は table です。
-r, --retries int	エラーとして終了するまでの再試行回数を設定します。デフォルト値は 3 です。
-d, --retry-delay int	再試行間の待機時間を秒単位で設定します。デフォルト値は 3 です。
--row-jsonpath-expressions string	JSON パス式を定義して、JSON オブジェクトから行を作成します。詳細は、 roxctl deployment check --help コマンドを実行してください。

8.8. ROXCTL HELM

Red Hat Advanced Cluster Security for Kubernetes (RHACS) Helm チャートに関連するコマンド。

使用方法

```
$ roxctl helm [command] [flags]
```

表8.58 利用可能なコマンド

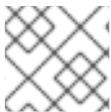
コマンド	説明
derive-local-values	クラスター設定からローカル Helm 値を導出します。
出力 (output)	Helm チャートを出力します。

8.8.1. 親コマンドから継承された roxctl helm コマンドオプション

roxctl helm コマンドは、roxctl の親コマンドから継承された次のオプションをサポートします。

オプション	説明
--ca string	安全な接続のためのカスタム CA 証明書ファイルパスを指定します。あるいは、 ROX_CA_CERT_FILE 環境変数を使用して、ファイルパスを指定することもできます。
--direct-grpc	接続パフォーマンスを向上させるには、 --direct-grpc を設定します。あるいは、 ROX_DIRECT_GRPC_CLIENT 環境変数を true に設定することで、直接 gRPC を有効にできます。デフォルト値は false です。
-e, --endpoint string	サービスが接続するエンドポイントを設定します。あるいは、 ROX_ENDPOINT 環境変数を使用して、エンドポイントを設定することもできます。デフォルト値は localhost:8443 です。
--force-http1	すべての接続で HTTP/1 を強制的に使用します。あるいは、 ROX_CLIENT_FORCE_HTTP1 環境変数を true に設定することで、HTTP/1 の使用を強制できます。デフォルト値は false です。
--insecure	安全でない接続オプションを有効にします。あるいは、 ROX_INSECURE_CLIENT 環境変数を true に設定することで、安全でない接続オプションを有効にすることができます。デフォルト値は false です。
--insecure-skip-tls-verify	TLS 証明書の検証をスキップします。あるいは、 ROX_INSECURE_CLIENT_SKIP_TLS_VERIFY 環境変数を true に設定することで、TLS 証明書の検証をスキップできます。デフォルト値は false です。

オプション	説明
--no-color	カラー出力を無効にします。あるいは、 ROX_NO_COLOR 環境変数を true に設定することで、カラー出力を無効にすることができます。デフォルト値は false です。
-p, --password string	Basic 認証のパスワードを指定します。あるいは、 ROX_ADMIN_PASSWORD 環境変数を使用して、パスワードを設定することもできます。
--plaintext	暗号化されていない接続を使用します。あるいは、 ROX_PLAINTEXT 環境変数を true に設定することで、暗号化されていない接続を有効にすることができます。デフォルト値は false です。
-s, --server-name string	SNI に使用する TLS サーバー名を設定します。あるいは、 ROX_SERVER_NAME 環境変数を使用して、サーバー名を設定することもできます。
--token-file string	指定されたファイルで提供される API トークンを認証に使用します。あるいは、 ROX_API_TOKEN 環境変数を使用して、トークンを設定することもできます。



注記

これらのオプションは、**roxctl helm** コマンドのすべてのサブコマンドに適用されます。

8.8.2. roxctl helm output

Helm チャートを出力します。

使用方法

```
$ roxctl helm output <central_services or secured_cluster_services> [flags] 1
```

- 1** **<central_services or secured_cluster_services>** には、Helm チャート出力を生成するための Central サービスまたはセキュアクラスターサービスへのパスを指定します。

表8.59 オプション

オプション	説明
--debug	ローカルファイルシステムからテンプレートを読み取ります。デフォルト値は false です。

オプション	説明
--debug-path string	ローカルファイルシステム上の Helm テンプレートへのパスを指定します。詳細は、 roxctl helm output --help コマンドを実行してください。
--image-defaults string	デフォルトのコンテナイメージを設定します。イメージ設定には、 development_build 、 stackrox.io 、 rhacs 、 opensource が含まれます。これは、イメージのダウンロード用リポジトリ、イメージ名、タグ形式に影響を及ぼします。デフォルト値は、 development_build です。
--output-dir string	Helm チャートの出力ディレクトリーへのパスを定義します。デフォルトのパスは ./stackrox-<chart name>-chart です。
--remove	出力ディレクトリーがすでに存在する場合は削除します。デフォルト値は false です。

8.8.3. roxctl helm derive-local-values

クラスター設定からローカル Helm 値を導出します。

使用方法

```
$ roxctl helm derive-local-values --output <path> \ ❶
<central_services> [flags] ❷
```

- ❶ **<path>** には、生成されたローカル値ファイルを保存するパスを指定します。
- ❷ **<central_services>** には、Central サービス設定ファイルへのパスを指定します。

表8.60 オプション

オプション	説明
--input string	YAML 入力を含むファイルまたはディレクトリーへのパスを指定します。
--output string	出力ファイルへのパスを定義します。
--output-dir string	出力ディレクトリーへのパスを定義します。

オプション	説明
--retry-timeout duration	API リクエストが再試行されるまでのタイムアウトを設定します。タイムアウト値は、リクエスト期間全体が再試行せずに待機することを示します。デフォルト値は 20s です。
-t, --timeout duration	リクエストの最大期間を表す API リクエストのタイムアウトを設定します。デフォルト値は 1m0s です。

8.9. ROXCTL IMAGE

特定のイメージに対して実行できるコマンド。

使用方法

```
$ roxctl image [command] [flags]
```

表8.61 利用可能なコマンド

コマンド	説明
check	イメージにビルド時間のポリシー違反がないか確認し、報告します。
scan	指定されたイメージをスキャンし、スキャン結果を返します。

表8.62 オプション

-t, --timeout duration	リクエストの最大期間を表す API リクエストのタイムアウトを設定します。デフォルト値は 10m0s です。
-------------------------------	---

8.9.1. 親コマンドから継承された roxctl image コマンドオプション

roxctl image コマンドは、**roxctl** の親コマンドから継承された次のオプションをサポートします。

オプション	説明
--ca string	安全な接続のためのカスタム CA 証明書ファイルパスを指定します。あるいは、 ROX_CA_CERT_FILE 環境変数を使用して、ファイルパスを指定することもできます。

オプション	説明
--direct-grpc	接続パフォーマンスを向上させるには、 --direct-grpc を設定します。あるいは、 ROX_DIRECT_GRPC_CLIENT 環境変数を true に設定することで、直接 gRPC を有効にできます。デフォルト値は false です。
-e, --endpoint string	サービスが接続するエンドポイントを設定します。あるいは、 ROX_ENDPOINT 環境変数を使用して、エンドポイントを設定することもできます。デフォルト値は localhost:8443 です。
--force-http1	すべての接続で HTTP/1 を強制的に使用します。あるいは、 ROX_CLIENT_FORCE_HTTP1 環境変数を true に設定することで、HTTP/1 の使用を強制できます。デフォルト値は false です。
--insecure	安全でない接続オプションを有効にします。あるいは、 ROX_INSECURE_CLIENT 環境変数を true に設定することで、安全でない接続オプションを有効にすることができます。デフォルト値は false です。
--insecure-skip-tls-verify	TLS 証明書の検証をスキップします。あるいは、 ROX_INSECURE_CLIENT_SKIP_TLS_VERIFY 環境変数を true に設定することで、TLS 証明書の検証をスキップできます。デフォルト値は false です。
--no-color	カラー出力を無効にします。あるいは、 ROX_NO_COLOR 環境変数を true に設定することで、カラー出力を無効にすることができます。デフォルト値は false です。
-p, --password string	Basic 認証のパスワードを指定します。あるいは、 ROX_ADMIN_PASSWORD 環境変数を使用して、パスワードを設定することもできます。
--plaintext	暗号化されていない接続を使用します。あるいは、 ROX_PLAINTEXT 環境変数を true に設定することで、暗号化されていない接続を有効にすることができます。デフォルト値は false です。
-s, --server-name string	SNI に使用する TLS サーバー名を設定します。あるいは、 ROX_SERVER_NAME 環境変数を使用して、サーバー名を設定することもできます。

オプション	説明
--token-file string	指定されたファイルで提供される API トークンを認証に使用します。あるいは、 ROX_API_TOKEN 環境変数を使用して、トークンを設定することもできます。



注記

これらのオプションは、**roxctl image** コマンドのすべてのサブコマンドに適用されません。

8.9.2. roxctl image scan

指定されたイメージをスキャンし、スキャン結果を返します。

使用方法

```
$ roxctl image scan [flags]
```

表8.63 オプション

オプション	説明
--cluster string	イメージスキャンを委任するクラスター名または ID を指定します。
--compact-output	JSON 出力をコンパクトな形式で出力します。デフォルト値は false です。
--fail	脆弱性が見つかった場合は失敗します。デフォルト値は false です。
-f, --force	Central のキャッシュを無視し、Scanner から新たにもう一度強制的にプルします。デフォルト値は false です。
--headers strings	表形式を出力するヘッダーを指定します。デフォルト値には、 COMPONENT 、 VERSION 、 CVE 、 S EVERITY 、および LINK が含まれます。
--headers-as-comments	CSV 表形式出力にヘッダーをコメントとして出力します。デフォルト値は false です。
-i, --image string	スキャンするイメージ名と参照を指定します。たとえば、 nginx:latest または nginx@sha256:... などです。

オプション	説明
-a, --include-snoozed	スヌーズされた CVE とスヌーズされていない CVE をスキャン結果に含めます。デフォルト値は false です。
--merge-output	表形式の出力で重複したセルを結合します。デフォルト値は true です。
--no-header	表形式の出力では、ヘッダーを出力しないでください。デフォルト値は false です。
-o, --output string	出力形式を指定します。出力形式には、 table 、 csv 、 json 、および sarif が含まれます。
-r, --retries int	エラーとして終了するまでの再試行回数を指定します。デフォルト値は 3 です。
-d, --retry-delay int	再試行間の待機時間を秒単位で設定します。デフォルト値は 3 です。
--row-jsonpath-expressions string	JSON オブジェクトから行を作成するには、JSON パス式を指定します。詳細は、 roxctl image scan --help コマンドを実行してください。
--severity strings	出力に追加する重大度のリスト。これを使用して、特定の重大度をフィルター処理します。デフォルト値には、 LOW 、 MODERATE 、 IMPORTANT 、および CRITICAL が含まれます。

8.9.3. roxctl image check

イメージにビルド時間のポリシー違反がないか確認し、報告します。

使用方法

```
$ roxctl image check [flags]
```

表8.64 オプション

オプション	説明
-c, --categories strings	実行するポリシーカテゴリーのリスト。デフォルトでは、すべてのポリシーカテゴリーが使用されます。

オプション	説明
--cluster string	評価のコンテキストとして使用するクラスター名または ID を定義します。
--compact-output	JSON 出力をコンパクトな形式で出力します。デフォルト値は false です。
-f, --force	イメージの Central キャッシュをバイパスし、強制的に Scanner から新たにプルします。デフォルト値は false です。
--headers strings	表形式に出力するヘッダーを定義します。デフォルト値には、 POLICY 、 SEVERITY 、 BREAKS BUILD 、 DESCRIPTION 、 VIOLATION 、および REMIEDIATION が含まれます。
--headers-as-comments	CSV 表形式出力にヘッダーをコメントとして出力します。デフォルト値は false です。
-i, --image string	イメージ名と参照を指定します。たとえば、 nginx:latest または nginx@sha256:... 。
--junit-suite-name string	JUnit テストスイートの名前を設定します。デフォルト値は image-check です。
--merge-output	表形式の出力で重複したセルを結合します。デフォルト値は false です。
--no-header	表形式の出力では、ヘッダーを出力しないでください。デフォルト値は false です。
-o, --output string	出力形式を選択します。出力形式には、 junit 、 sarif 、 table 、 csv 、および json が含まれます。デフォルト値は table です。
-r, --retries int	エラーとして終了するまでの再試行回数を設定します。デフォルト値は 3 です。
-d, --retry-delay int	再試行間の待機時間を秒単位で設定します。デフォルト値は 3 です。
--row-jsonpath-expressions string	JSON パス式を使用して、JSON オブジェクトから行を作成します。詳細は、 roxctl image check --help コマンドを実行してください。

オプション	説明
--send-notifications	違反が発生した場合に通知を送信するかどうかを定義します。デフォルト値は false です。

8.10. ROXCTL NETPOL

ネットワークポリシーに関連するコマンド。

使用方法

```
$ roxctl netpol [command] [flags]
```

表8.65 利用可能なコマンド

コマンド	説明
connectivity	ネットワークポリシーリソースの接続分析。
generate	デプロイメント情報に基づいて、ネットワークポリシーを推奨します。

8.10.1. 親コマンドから継承された roxctl netpol コマンドオプション

roxctl netpol コマンドは、**roxctl** の親コマンドから継承された次のオプションをサポートします。

オプション	説明
--ca string	安全な接続のためのカスタム CA 証明書ファイルパスを指定します。あるいは、 ROX_CA_CERT_FILE 環境変数を使用して、ファイルパスを指定することもできます。
--direct-grpc	接続パフォーマンスを向上させるには、 --direct-grpc を設定します。あるいは、 ROX_DIRECT_GRPC_CLIENT 環境変数を true に設定することで、直接 gRPC を有効にできます。デフォルト値は false です。
-e, --endpoint string	サービスが接続するエンドポイントを設定します。あるいは、 ROX_ENDPOINT 環境変数を使用して、エンドポイントを設定することもできます。デフォルト値は localhost:8443 です。
--force-http1	すべての接続で HTTP/1 を強制的に使用します。あるいは、 ROX_CLIENT_FORCE_HTTP1 環境変数を true に設定することで、HTTP/1 の使用を強制できます。デフォルト値は false です。

オプション	説明
--insecure	安全でない接続オプションを有効にします。あるいは、 ROX_INSECURE_CLIENT 環境変数を true に設定することで、安全でない接続オプションを有効にすることができます。デフォルト値は false です。
--insecure-skip-tls-verify	TLS 証明書の検証をスキップします。あるいは、 ROX_INSECURE_CLIENT_SKIP_TLS_VERIFY 環境変数を true に設定することで、TLS 証明書の検証をスキップできます。デフォルト値は false です。
--no-color	カラー出力を無効にします。あるいは、 ROX_NO_COLOR 環境変数を true に設定することで、カラー出力を無効にすることができます。デフォルト値は false です。
-p, --password string	Basic 認証のパスワードを指定します。あるいは、 ROX_ADMIN_PASSWORD 環境変数を使用して、パスワードを設定することもできます。
--plaintext	暗号化されていない接続を使用します。あるいは、 ROX_PLAINTEXT 環境変数を true に設定することで、暗号化されていない接続を有効にすることができます。デフォルト値は false です。
-s, --server-name string	SNI に使用する TLS サーバー名を設定します。あるいは、 ROX_SERVER_NAME 環境変数を使用して、サーバー名を設定することもできます。
--token-file string	指定されたファイルで提供される API トークンを認証に使用します。あるいは、 ROX_API_TOKEN 環境変数を使用して、トークンを設定することもできます。



注記

これらのオプションは、**roxctl netpol** コマンドのすべてのサブコマンドに適用されません。

8.10.2. roxctl netpol generate

デプロイメント情報に基づいて、ネットワークポリシーを推奨します。

使用方法

```
$ roxctl netpol generate <folder_path> [flags] 1
```

1 **<folder_path>** には、Kubernetes デプロイメントおよびサービス設定ファイルが含まれるディレクトリへのパスを指定します。

クトリーへのパスを指定します。

表8.66 オプション

オプション	説明
--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。
-d, --output-dir string	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。
-f, --output-file string	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。
--remove	出力パスがすでに存在する場合は削除します。デフォルト値は false です。
--strict	警告をエラーとして扱います。デフォルト値は false です。

8.10.3. roxctl netpol connectivity

ネットワークポリシーリソースの接続分析に関連するコマンド。

使用方法

```
$ roxctl netpol connectivity [flags]
```

8.10.3.1. roxctl netpol connectivity map

ネットワークポリシーやその他のリソースに基づいて、接続を分析します。

使用方法

```
$ roxctl netpol connectivity map <folder_path> [flags] 1
```

- 1** **<folder_path>** には、Kubernetes デプロイメントおよびサービス設定ファイルが含まれるディレクトリーへのパスを指定します。

表8.67 オプション

オプション	説明
--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。
--focus-workload string	出力内の指定されたワークロード名の接続に注目します。

オプション	説明
-f, --output-file string	出力された接続リストを特定のファイルに保存します。
-o, --output-format string	接続リストを特定の形式で設定します。サポートされている形式には、 txt 、 json 、 md 、 dot 、および csv が含まれます。デフォルト値は txt です。
--remove	出力パスがすでに存在する場合は削除します。デフォルト値は false です。
--save-to-file	接続リストの出力をデフォルトファイルに保存するかどうかを定義します。デフォルト値は false です。
--strict	警告をエラーとして扱います。デフォルト値は false です。

8.10.3.2. roxctl netpol connectivity diff

2つのネットワークポリシーディレクトリーとワークロードリソースを含むYAML マニフェストに基づいて、接続の違いをレポートします。

使用方法

```
$ roxctl netpol connectivity diff [flags]
```

表8.68 オプション

オプション	説明
--dir1 string	入力リソースの最初のディレクトリーパスを指定します。この値は必須です。
--dir2 string	最初のディレクトリーパスと比較する入力リソースの2番目のディレクトリーパスを指定します。この値は必須です。
--fail	最初のエラーで失敗します。デフォルト値は false です。
-f, --output-file string	connectivity difference コマンドの出力を特定のファイルに保存します。

オプション	説明
-o, --output-format string	connectivity difference コマンドの出力を特定の形式で設定します。サポートされている形式には、 txt 、 md 、 csv が含まれます。デフォルト値は txt です..
--remove	出力パスがすでに存在する場合は削除します。デフォルト値は false です。
--save-to-file	接続の違いの出力をデフォルトファイルに保存するかどうかを定義します。デフォルト値は false です。
--strict	警告をエラーとして扱います。デフォルト値は false です。

8.11. ROXCTL SCANNER

StackRox Scanner および Scanner V4 サービスに関連するコマンド。

使用方法

```
$ roxctl scanner [command] [flags]
```

表8.69 利用可能なコマンド

コマンド	説明
download-db	StackRox Scanner および Scanner V4 のオフライン脆弱性データベースをダウンロードします。
generate	StackRox Scanner および Scanner V4 をデプロイするために必要な YAML 設定ファイルを生成します。
upload-db	StackRox Scanner および Scanner V4 の脆弱性データベースをアップロードします。

8.11.1. 親コマンドから継承された roxctl scanner コマンドオプション

roxctl scanner コマンドは、**roxctl** の親コマンドから継承された次のオプションをサポートします。

オプション	説明
-------	----

オプション	説明
--ca string	安全な接続のためのカスタム CA 証明書ファイルパスを指定します。あるいは、 ROX_CA_CERT_FILE 環境変数を使用して、ファイルパスを指定することもできます。
--direct-grpc	接続パフォーマンスを向上させるには、 --direct-grpc を設定します。あるいは、 ROX_DIRECT_GRPC_CLIENT 環境変数を true に設定することで、直接 gRPC を有効にできます。デフォルト値は false です。
-e, --endpoint string	サービスが接続するエンドポイントを設定します。あるいは、 ROX_ENDPOINT 環境変数を使用して、エンドポイントを設定することもできます。デフォルト値は localhost:8443 です。
--force-http1	すべての接続で HTTP/1 を強制的に使用します。あるいは、 ROX_CLIENT_FORCE_HTTP1 環境変数を true に設定することで、HTTP/1 の使用を強制できます。デフォルト値は false です。
--insecure	安全でない接続オプションを有効にします。あるいは、 ROX_INSECURE_CLIENT 環境変数を true に設定することで、安全でない接続オプションを有効にすることができます。デフォルト値は false です。
--insecure-skip-tls-verify	TLS 証明書の検証をスキップします。あるいは、 ROX_INSECURE_CLIENT_SKIP_TLS_VERIFY 環境変数を true に設定することで、TLS 証明書の検証をスキップできます。デフォルト値は false です。
--no-color	カラー出力を無効にします。あるいは、 ROX_NO_COLOR 環境変数を true に設定することで、カラー出力を無効にすることができます。デフォルト値は false です。
-p, --password string	Basic 認証のパスワードを指定します。あるいは、 ROX_ADMIN_PASSWORD 環境変数を使用して、パスワードを設定することもできます。
--plaintext	暗号化されていない接続を使用します。あるいは、 ROX_PLAINTEXT 環境変数を true に設定することで、暗号化されていない接続を有効にすることができます。デフォルト値は false です。

オプション	説明
-s, --server-name string	SNI に使用する TLS サーバー名を設定します。あるいは、 ROX_SERVER_NAME 環境変数を使用して、サーバー名を設定することもできます。
--token-file string	指定されたファイルで提供される API トークンを認証に使用します。あるいは、 ROX_API_TOKEN 環境変数を使用して、トークンを設定することもできます。



注記

これらのオプションは、**roxctl scanner** コマンドのすべてのサブコマンドに適用されません。

8.11.2. roxctl scanner generate

Scanner をデプロイするために必要な YAML 設定ファイルを生成します。

使用方法

```
$ roxctl scanner generate [flags]
```

表8.70 オプション

オプション	説明
--cluster-type cluster type	Scanner を実行するクラスターのタイプを指定します。クラスターの種類には、 k8s および openshift が含まれます。デフォルト値は k8s です。
--enable-pod-security-policies	PodSecurityPolicy リソースを作成します。デフォルト値は true です。
--istio-support string	指定された Istio バージョンをサポートするデプロイメントファイルを生成します。有効なバージョンには、 1.0 、 1.1 、 1.2 、 1.3 、 1.4 、 1.5 、 1.6 、および 1.7 が含まれます。
--output-dir string	Scanner バンドルの出力ディレクトリーを指定します。デフォルト値を使用する場合は空白のままにします。
--retry-timeout duration	API リクエストが再試行されるまでのタイムアウトを設定します。値が 0 の場合、再試行せずにリクエスト期間全体を待機することを意味します。デフォルト値は 20s です。

オプション	説明
--scanner-image string	使用する Scanner イメージを指定します。サーバーのデフォルトを使用するには、空白のままにします。
-t, --timeout duration	リクエストの最大期間を表す API リクエストのタイムアウトを設定します。デフォルト値は 1m0s です。

8.11.3. roxctl scanner upload-db

Scanner の脆弱性データベースをアップロードします。

使用方法

```
$ roxctl scanner upload-db [flags]
```

表8.71 オプション

オプション	説明
--scanner-db-file string	ダンプした Scanner 定義 DB を含むファイルを指定します。
-t, --timeout duration	リクエストの最大期間を表す API リクエストのタイムアウトを設定します。デフォルト値は 10m0s です。

8.11.4. roxctl scanner download-db

StackRox Scanner または Scanner V4 のオフライン脆弱性データベースをダウンロードします。

このコマンドは、バージョン固有のオフライン脆弱性バンドルをダウンロードします。バージョンが指定されていない場合、システムは Central に問い合わせてバージョンを確認します。通信が失敗した場合、ダウンロードはデフォルトで **roxctl** 内に埋め込まれたバージョンになります。

デフォルトでは、特定されたバージョンとそれほど特定のでないバリエーションのデータベースのダウンロードが試行されます。たとえば、バージョン **4.4.1-extra** が指定されている場合、次のバージョンのバリエーションに対してダウンロードが試行されます。

- 4.4.1-extra
- 4.4.1
- 4.4

使用方法

```
$ roxctl scanner download-db [flags]
```

表8.72 オプション

オプション	説明
--force	出力ファイルがすでに存在する場合は、強制的に上書きします。デフォルト値は false です。
--scanner-db-file string	脆弱性データベースを保存する出力ファイル。デフォルト値は、ダウンロードされるリモートファイルの名前とパスです。
--skip-central	バージョンを検出するときに Central に連絡しないでください。デフォルト値は false です。
--skip-variants	決定されたバージョンのバリエーションの処理を試行しないでください。デフォルト値は false です。
-t, --timeout duration	リクエストの最大期間を表す API リクエストのタイムアウトを設定します。デフォルト値は 10m0s です。
--version 文字列	脆弱性データベースの特定バージョンまたはバージョンバリエーションをダウンロードします。デフォルトでは、バージョンは自動的に検出されます。

8.12. ROXCTL SENSOR

Red Hat Advanced Cluster Security for Kubernetes (RHACS) サービスをセキュアクラスターにデプロイします。

使用方法

```
$ roxctl sensor [command] [flags]
```

表8.73 利用可能なコマンド

コマンド	説明
generate	ファイルを生成して、セキュアクラスターに RHACS サービスをデプロイします。
generate-certs	Sensor、Collector、および Admission コントローラーの更新された証明書を含む YAML ファイルをダウンロードします。
get-bundle	クラスターに RHACS サービスをデプロイするためのファイルを含むバンドルをダウンロードします。

表8.74 オプション

オプション	説明
<code>--retry-timeout duration</code>	API リクエストが再試行されるまでのタイムアウトを設定します。値が0の場合、再試行せずにリクエスト期間全体を待機することを意味します。デフォルト値は 20s です。
<code>-t, --timeout duration</code>	リクエストの最大期間を表す API リクエストのタイムアウトを設定します。デフォルト値は 1m0s です。

8.12.1. 親コマンドから継承された `roxctl sensor` コマンドオプション

`roxctl sensor` コマンドは、`roxctl` の親コマンドから継承された次のオプションをサポートします。

オプション	説明
<code>--ca string</code>	安全な接続のためのカスタム CA 証明書ファイルパスを指定します。あるいは、 ROX_CA_CERT_FILE 環境変数を使用して、ファイルパスを指定することもできます。
<code>--direct-grpc</code>	接続パフォーマンスを向上させるには、 --direct-grpc を設定します。あるいは、 ROX_DIRECT_GRPC_CLIENT 環境変数を true に設定することで、直接 gRPC を有効にできます。デフォルト値は false です。
<code>-e, --endpoint string</code>	サービスが接続するエンドポイントを設定します。あるいは、 ROX_ENDPOINT 環境変数を使用して、エンドポイントを設定することもできます。デフォルト値は localhost:8443 です。
<code>--force-http1</code>	すべての接続で HTTP/1 を強制的に使用します。あるいは、 ROX_CLIENT_FORCE_HTTP1 環境変数を true に設定することで、HTTP/1 の使用を強制できます。デフォルト値は false です。
<code>--insecure</code>	安全でない接続オプションを有効にします。あるいは、 ROX_INSECURE_CLIENT 環境変数を true に設定することで、安全でない接続オプションを有効にすることができます。デフォルト値は false です。
<code>--insecure-skip-tls-verify</code>	TLS 証明書の検証をスキップします。あるいは、 ROX_INSECURE_CLIENT_SKIP_TLS_VERIFY 環境変数を true に設定することで、TLS 証明書の検証をスキップできます。デフォルト値は false です。

オプション	説明
--no-color	カラー出力を無効にします。あるいは、 ROX_NO_COLOR 環境変数を true に設定することで、カラー出力を無効にすることができます。デフォルト値は false です。
-p, --password string	Basic 認証のパスワードを指定します。あるいは、 ROX_ADMIN_PASSWORD 環境変数を使用して、パスワードを設定することもできます。
--plaintext	暗号化されていない接続を使用します。あるいは、 ROX_PLAINTEXT 環境変数を true に設定することで、暗号化されていない接続を有効にすることができます。デフォルト値は false です。
-s, --server-name string	SNI に使用する TLS サーバー名を設定します。あるいは、 ROX_SERVER_NAME 環境変数を使用して、サーバー名を設定することもできます。
--token-file string	指定されたファイルで提供される API トークンを認証に使用します。あるいは、 ROX_API_TOKEN 環境変数を使用して、トークンを設定することもできます。



注記

これらのオプションは、**roxctl sensor** コマンドのすべてのサブコマンドに適用されません。

8.12.2. roxctl sensor generate

ファイルを生成して、セキュアクラスターに RHACS サービスをデプロイします。

使用方法

```
$ roxctl sensor generate [flags]
```

表8.75 オプション

オプション	説明
--admission-controller-disable-bypass	アドミッションコントローラーのバイパスアノテーションを無効にします。デフォルト値は false です。

オプション	説明
--admission-controller-enforce-on-creates	アドミッションコントローラーのオブジェクト作成を強制するための動的な有効化。デフォルト値は false です。
--admission-controller-enforce-on-updates	アドミッションコントローラーでのオブジェクト更新の動的な強制を有効にします。デフォルト値は false です。
--admission-controller-listen-on-creates	デプロイメントの作成をリッスンするようにアドミッションコントローラー Webhook を設定します。デフォルト値は false です。
--admission-controller-listen-on-updates	デプロイメントの更新をリッスンするようにアドミッションコントローラー Webhook を設定します。デフォルト値は false です。
--admission-controller-scan-inline	アドミッションコントローラーを使用するときにスキャンをインラインで取得します。デフォルト値は false です。
--admission-controller-timeout int32	アドミッションコントローラーのタイムアウトを秒単位で設定します。デフォルト値は 3 です。
--central string	Sensor を接続するエンドポイントを設定します。デフォルト値は central.stackrox:443 です。
--collection-method collection method	ランタイムサポートに使用する収集方法を指定します。収集方法には、 none 、 default 、 ebpf 、および core_bpf が含まれます。デフォルト値は default です。
--collector-image-repository string	Collector のデプロイに使用するイメージリポジトリを設定します。指定されていない場合は、有効な --main-image repository 値に対応するデフォルト値が導出されます。

オプション	説明
--continue-if-exists	クラスターがすでに存在する場合でも、Sensor バンドルのダウンロードを続行します。デフォルト値は false です。
--create-upgrader-sa	Sensor の自動アップグレードを容易にするために、 cluster-admin 特権を持つアップグレードサービスアカウントを作成するかどうかを決定します。デフォルト値は true です。
--disable-tolerations	テイントノードの容認を無効にします。デフォルト値は false です。
--enable-pod-security-policies	PodSecurityPolicy リソースを作成します。デフォルト値は true です。
--istio-support string	指定された Istio バージョンをサポートするデプロイメントファイルを生成します。有効なバージョンには、 1.0 、 1.1 、 1.2 、 1.3 、 1.4 、 1.5 、 1.6 、 1.7 が含まれます。
--main-image-repository string	Sensor のデプロイに使用するイメージリポジトリを指定します。指定しない場合は、デフォルト値が使用されます。
--name string	クラスターを識別するためのクラスター名を設定します。
--output-dir string	バンドルコンテンツの出力ディレクトリを設定します。デフォルト値は、現在のディレクトリー内で自動的に生成されたディレクトリー名です。
--slim-collector string["true"]	デプロイメントバンドルで Collector-slim を使用します。有効な値には、 auto 、 true 、および false が含まれます。デフォルト値は auto です。

オプション	説明
-t, --timeout duration	リクエストの最大期間を表す API リクエストのタイムアウトを設定します。デフォルト値は 5m0s です。

8.12.2.1. roxctl sensor generate k8s

Kubernetes クラスタに RHACS サービスをデプロイするために必要なファイルを生成します。

使用方法

```
$ roxctl sensor generate k8s [flags]
```

表8.76 オプション

オプション	説明
--admission-controller-listen-on-events	Kubernetes イベントをリッスンするためにアドミッションコントローラー Webhook を有効にします。デフォルト値は true です。

8.12.2.2. roxctl sensor generate openshift

Red Hat OpenShift クラスタに RHACS サービスをデプロイするために必要なファイルを生成します。

使用方法

```
$ roxctl sensor generate openshift [flags]
```

表8.77 オプション

オプション	説明
<code>`--admission-controller-listen-on-events false`</code>	true
<code>auto[=true]`</code>	Kubernetes イベントをリッスンするためのアドミッションコントローラー Webhook を有効または無効にします。 デフォルト値は auto です。
<code>`--disable-audit-logs false`</code>	true

オプション	説明
auto[=true]	実行時検出のための監査ログ収集を有効または無効にします。デフォルト値は auto です。
--openshift-version int	デプロイメントファイルを生成する Red Hat OpenShift メジャーバージョンを指定します。

8.12.3. roxctl sensor get-bundle

RHACS サービスをクラスターにデプロイするためのファイルを含むバンドルをダウンロードします。

使用方法

```
$ roxctl sensor get-bundle <cluster_details> [flags] 1
```

1 **<cluster_details>** には、クラスター名または ID を指定します。

表8.78 オプション

オプション	説明
--create-upgrader-sa	Sensor の自動アップグレード用に、 cluster-admin 権限を持つアップグレードサービスアカウントを作成するかどうかを指定します。デフォルト値は true です。
--istio-support string	指定された Istio バージョンをサポートするデプロイメントファイルを生成します。有効なバージョンには 1.0 、 1.1 、 1.2 、 1.3 、 1.4 、 1.5 、 1.6 、および 1.7 が含まれません。
--output-dir string	バンドルコンテンツの出力ディレクトリーを指定します。デフォルト値は、現在のディレクトリー内で自動的に生成されたディレクトリー名です。
--slim-collector string[="true"]	デプロイメントバンドルで Collector-slim を使用します。有効な値には、 auto 、 true 、および false が含まれます。デフォルト値は auto です。

オプション	説明
-t, --timeout duration	リクエストの最大期間を表す API リクエストのタイムアウトを設定します。デフォルト値は 5m0s です。

8.12.4. roxctl sensor generate-certs

Sensor、Collector、および Admission コントローラーの更新された証明書を含む YAML ファイルをダウンロードします。

使用方法

```
$ roxctl sensor generate-certs <cluster_details> [flags] ❶
```

❶ **<cluster_details>** には、クラスター名または ID を指定します。

表8.79 オプション

オプション	説明
--output-dir string	YAML ファイルの出力ディレクトリーを指定します。デフォルト値は . です。

8.13. ROXCTL VERSION

現在の roxctl のバージョンを表示します。

使用方法

```
$ roxctl version [flags]
```

8.13.1. roxctl version コマンドのオプション

roxctl version コマンドは、次のオプションをサポートします。

オプション	説明
--json	拡張バージョン情報を JSON で表示します。デフォルト値は false です。

8.13.2. 親コマンドから継承された roxctl version コマンドオプション

roxctl version コマンドは、**roxctl** の親コマンドから継承された次のオプションをサポートします。

オプション	説明
--ca string	安全な接続のためのカスタム CA 証明書ファイルパスを指定します。あるいは、 ROX_CA_CERT_FILE 環境変数を使用して、ファイルパスを指定することもできます。
--direct-grpc	接続パフォーマンスを向上させるには、 --direct-grpc を設定します。あるいは、 ROX_DIRECT_GRPC_CLIENT 環境変数を true に設定することで、直接 gRPC を有効にできます。デフォルト値は false です。
-e, --endpoint string	サービスが接続するエンドポイントを設定します。あるいは、 ROX_ENDPOINT 環境変数を使用して、エンドポイントを設定することもできます。デフォルト値は localhost:8443 です。
--force-http1	すべての接続で HTTP/1 を強制的に使用します。あるいは、 ROX_CLIENT_FORCE_HTTP1 環境変数を true に設定することで、HTTP/1 の使用を強制できます。デフォルト値は false です。
--insecure	安全でない接続オプションを有効にします。あるいは、 ROX_INSECURE_CLIENT 環境変数を true に設定することで、安全でない接続オプションを有効にすることができます。デフォルト値は false です。
--insecure-skip-tls-verify	TLS 証明書の検証をスキップします。あるいは、 ROX_INSECURE_CLIENT_SKIP_TLS_VERIFY 環境変数を true に設定することで、TLS 証明書の検証をスキップできます。デフォルト値は false です。
--no-color	カラー出力を無効にします。あるいは、 ROX_NO_COLOR 環境変数を true に設定することで、カラー出力を無効にすることができます。デフォルト値は false です。
-p, --password string	Basic 認証のパスワードを指定します。あるいは、 ROX_ADMIN_PASSWORD 環境変数を使用して、パスワードを設定することもできます。
--plaintext	暗号化されていない接続を使用します。あるいは、 ROX_PLAINTEXT 環境変数を true に設定することで、暗号化されていない接続を有効にすることができます。デフォルト値は false です。

オプション	説明
-s, --server-name string	SNI に使用する TLS サーバー名を設定します。あるいは、 ROX_SERVER_NAME 環境変数を使用して、サーバー名を設定することもできます。
--token-file string	指定されたファイルで提供される API トークンを認証に使用します。あるいは、 ROX_API_TOKEN 環境変数を使用して、トークンを設定することもできます。