Red Hat Advanced Cluster Security for Kubernetes 3.71

roxctl CLI

roxctl CLI
roxctl CLI
Abstract

This document describes how to install and use the roxctl command-line interface, including the roxctl syntax and operations. It provides some common command examples.
# Table of Contents

## CHAPTER 1. GETTING STARTED WITH THE ROXCTL CLI  ......................................................... 3

1.1. INSTALLING THE ROXCTL CLI  ................................................................. 3
1.1.1. Installing the roxctl CLI by downloading the binary  3
1.1.1.1. Installing the roxctl CLI on Linux  3
1.1.1.2. Installing the roxctl CLI on macOS  3
1.1.1.3. Installing the roxctl CLI on Windows  4
1.1.2. Running the roxctl CLI from a container  4

1.2. AUTHENTICATING USING THE ROXCTL CLI  ..................................................... 5

1.3. USING THE ROXCTL CLI  ................................................................. 6
1.3.1. Managing Central’s database  ................................................................. 6
1.3.2. Managing secured clusters  ................................................................. 7
1.3.2.1. Generating Sensor deployment YAML file  ........................................ 7
1.3.2.2. Downloading Sensor bundle for existing clusters  7
1.3.2.3. Deleting cluster integration  ............................................................... 7
1.3.3. Checking policy compliance  ................................................................. 8
1.3.3.1. Configuring output format  ............................................................... 8
1.3.3.2. Checking deployment YAML files  .................................................... 9
1.3.3.3. Checking images  ....................................................................... 9
1.3.3.4. Checking image scan results  ........................................................... 9
1.3.4. Debugging issues  ................................................................. 10
1.3.4.1. Managing Central log level  ............................................................. 10
1.3.4.2. Viewing the logs  ..................................................................... 10
1.3.4.3. Viewing current log level  ............................................................... 10
1.3.4.4. Changing the log level  ................................................................. 10
1.3.4.5. Retrieving debugging information  .................................................. 10
CHAPTER 1. GETTING STARTED WITH THE ROXCTL CLI

roxctl is a command-line interface (CLI) for running commands on Red Hat Advanced Cluster Security for Kubernetes. This topic describes roxctl syntax, operations, and provides some common examples.

1.1. INSTALLING THE ROXCTL CLI

You can install the roxctl CLI by downloading the binary. Or, you can run the roxctl CLI from a container image.

1.1.1. Installing the roxctl CLI by downloading the binary

You can install the roxctl CLI to interact with Red Hat Advanced Cluster Security for Kubernetes from a command-line interface. You can install roxctl on Linux, Windows, or macOS.

1.1.1.1. Installing the roxctl CLI on Linux

You can install the roxctl CLI binary on Linux by using the following procedure.

Procedure

1. Download the latest version of the roxctl CLI:

   $ curl -O https://mirror.openshift.com/pub/rhacs/assets/3.71.0/bin/Linux/roxctl

2. Make the roxctl binary executable:

   $ chmod +x roxctl

3. Place the roxctl binary in a directory that is on your PATH:
   To check your PATH, execute the following command:

   $ echo $PATH

Verification

- Verify the roxctl version you have installed:

   $ roxctl version

1.1.1.2. Installing the roxctl CLI on macOS

You can install the roxctl CLI binary on macOS by using the following procedure.

Procedure

1. Download the latest version of the roxctl CLI:

   $ curl -O https://mirror.openshift.com/pub/rhacs/assets/3.71.0/bin/Darwin/roxctl

2. Remove all extended attributes from the binary:
3. Make the `roxctl` binary executable:
   ```bash
   $ chmod +x roxctl
   ```

4. Place the `roxctl` binary in a directory that is on your `PATH`:
   To check your `PATH`, execute the following command:
   ```bash
   $ echo $PATH
   ```

Verification

- Verify the `roxctl` version you have installed:
  ```bash
  $ roxctl version
  ```

1.1.3. Installing the roxctl CLI on Windows

You can install the `roxctl` CLI binary on Windows by using the following procedure.

Procedure

- Download the latest version of the `roxctl` CLI:
  ```bash
  $ curl -O https://mirror.openshift.com/pub/rhacs/assets/3.71.0/bin/Windows/roxctl.exe
  ```

Verification

- Verify the `roxctl` version you have installed:
  ```bash
  $ roxctl version
  ```

1.1.2. Running the roxctl CLI from a container

The `roxctl` client is the default entry point in Red Hat Advanced Cluster Security for Kubernetes `roxctl` image. To run the `roxctl` client in a container image:

Procedure

1. Log in to the `registry.redhat.io` registry.
   ```bash
   $ docker login registry.redhat.io
   ```

2. Pull the latest container image for the `roxctl` CLI.
   ```bash
   $ docker pull registry.redhat.io/advanced-cluster-security/rhacs-roxctl-rhel8:3.71.0
   ```

After you install the CLI, you can run it by using the following command:
1.2. AUTHENTICATING USING THE ROXCTL CLI

For authentication, you can use an authentication token or your administrator password. Red Hat recommends using an authentication token in a production environment because each token is assigned specific access control permissions.

Use the following steps to generate an authentication token.

**Procedure**

1. Navigate to the RHACS portal.
2. Go to Platform Configuration → Integrations.
3. Scroll down to the Authentication Tokens category, and click API Token.
4. Click Generate Token.
5. Enter a name for the token and select a role that provides the required level of access (for example, Continuous Integration or Sensor Creator).
6. Click Generate.

**IMPORTANT**

Copy the generated token and securely store it. You will not be able to view it again.
NOTE
After you have generated the authentication token, export it as `ROX_API_TOKEN` variable:

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

You can also save the token in a file and use it with the `--token-file` option. For example:

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

- You cannot use both the `-password (-p)` and the `--token-file` options simultaneously.
- If you have already set `ROX_API_TOKEN` variable, and specify the `--token-file` option, the `roxctl` CLI uses the specified token file for authentication.
- If you have already set `ROX_API_TOKEN` variable, and specify the `--password` option, the `roxctl` CLI uses the specified password for authentication.

1.3. USING THE ROXCTL CLI

Review the following sections to learn how to complete common tasks using the CLI.

NOTE
- Export the following variables before using these commands:

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

- You can use the `--help` option to get more information about the commands.

1.3.1. Managing Central’s database

Central stores information about:

- Activity observed in your clusters,
- Information retrieved from integrated image registries or scanners, and

You can back up and restore Central’s database by using the `roxctl` CLI.

**Backing up Central database**

Run the following command to back up Central’s database:

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

**Restoring Central database**

Run the following command to restore Central’s database:
1.3.2. Managing secured clusters

To secure a Kubernetes or an OpenShift Container Platform cluster, you must deploy Red Hat Advanced Cluster Security for Kubernetes services into the cluster. You can generate deployment files in the RHACS portal by navigating to the Platform Configuration → Clusters view, or you can use the roxctl CLI.

Generating Sensor deployment YAML file

Kubernetes

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

OpenShift Container Platform

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

Read the --help output to see other options that you might need to use depending on your system architecture.

Make sure that the endpoint you provide for --central can be reached from the cluster where you are deploying Red Hat Advanced Cluster Security for Kubernetes services.

NOTE

If you are using a non-gRPC capable load balancer, such as HAProxy, AWS Application Load Balancer (ALB), or AWS Elastic Load Balancing (ELB):

- Use the WebSocket Secure (wss) protocol. To use wss, prefix the address with wss://, and
- Add the port number after the address, for example:

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

Downloading Sensor bundle for existing clusters

Use the following command to download Sensor bundles for existing clusters by specifying a cluster name or ID.

$ roxctl sensor get-bundle <cluster_name_or_id>

Deleting cluster integration

$ roxctl -e "$ROX_CENTRAL_ADDRESS" cluster delete --name=<cluster_name>
IMPORTANT
Deleting cluster integration will not remove Red Hat Advanced Cluster Security for Kubernetes services running in the cluster. You can remove them by running the delete-sensor.sh script from the Sensor installation bundle.

1.3.3. Checking policy compliance
You can use the roxctl CLI to check deployment YAML files and images for policy compliance.

Configuring output format
When you check policy compliance by using the deployment check, image check, or image scan commands, you can specify the output format by using the -o option. This option determines how the output of a command is displayed in the terminal.

You can change the output format by adding the -o option to the command and specifying the format as json, table, csv, or junit.

For example, the following command checks a deployment and then displays the result in csv format:

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

NOTE
When you do not specify the -o option for the output format, the following default behavior is used:

- The format for the deployment check and the image check commands is table.
- The default output format for the image scan command is json. This is the old JSON format output for compatibility with older versions of the CLI. To get the output in the new JSON format, specify the option with format, as -o json.

Different options are available to configure the output. The following table lists the options and the format in which they are available.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Formats</th>
</tr>
</thead>
<tbody>
<tr>
<td>--compact-output</td>
<td>Use this option to display the JSON output in a compact format.</td>
<td>json</td>
</tr>
<tr>
<td>--headers</td>
<td>Use this option to specify custom headers.</td>
<td>table and csv</td>
</tr>
<tr>
<td>--no-header</td>
<td>Use this option to omit the header row from the output.</td>
<td>table and csv</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
<td>Formats</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>--row-jsonpath-expressions</td>
<td>Use this option to specify GJSON paths to select specific items from the output. For example, to get the Policy name and Severity for a deployment check, use the following command:</td>
<td>table and csv</td>
</tr>
<tr>
<td></td>
<td>$ roxctl -e &quot;$ROX_CENTRAL_ADDRESS&quot; \</td>
<td></td>
</tr>
<tr>
<td></td>
<td>deployment check --file=&lt;yaml_filename&gt; \</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-o table --headers POLICY-NAME,SEVERITY \</td>
<td></td>
</tr>
<tr>
<td></td>
<td>--row-jsonpath-expressions=&quot;\</td>
<td></td>
</tr>
<tr>
<td></td>
<td>{results.#.violatedPolicies.#.name,results.#.violatedPolicies.#.severity}&quot;</td>
<td></td>
</tr>
<tr>
<td>--merge-output</td>
<td>Use this option to merge table cells that have the same value.</td>
<td>table</td>
</tr>
<tr>
<td>headers-as-comment</td>
<td>Use this option to include the header row as a comment in the output.</td>
<td>csv</td>
</tr>
<tr>
<td>--junit-suite-name</td>
<td>Use this option to specify the name of the JUnit test suite.</td>
<td>junit</td>
</tr>
</tbody>
</table>

### Checking deployment YAML files
The following command checks build-time and deploy-time violations of your security policies in YAML deployment files. Use this command to validate:

- Configuration options in a YAML file, such as resource limits or privilege options; or
- Aspects of the images used in a YAML file, such as components or vulnerabilities.

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

### Checking images
The following command checks build-time violations of your security policies in images.

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

### Checking image scan results
You can also check the scan results for specific images.

The following command returns the components and vulnerabilities found in the image in JSON format. The format is defined in the API reference.

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

To cause Red Hat Advanced Cluster Security for Kubernetes to re-pull image metadata and image scan results from the associated registry and scanner, add the `--force` option.
NOTE
To check specific image scan results, you must have a token with both read and write permissions for the Image resource. The default Continuous Integration system role already has the required permissions.

1.3.4. Debugging issues

Managing Central log level
Central saves information to its container logs.

Viewing the logs
You can see the container logs for Central by running:

Kubernetes

$ kubectl logs -n stackrox <central_pod>

OpenShift Container Platform

$ oc logs -n stackrox <central_pod>

Viewing current log level
You can change the log level to see more or less information in Central logs. Run the following command to view the current log level:

$ roxctl -e "$ROX_CENTRAL_ADDRESS" central debug log

Changing the log level
Run the following command to change the log level:

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

① The acceptable values for <log_level> are Panic, Fatal, Error, Warn, Info, and Debug.

Retrieving debugging information
To gather debugging information for investigating issues, run the following command:

$ roxctl -e "$ROX_CENTRAL_ADDRESS" central debug dump