Red Hat Advanced Cluster Security for Kubernetes 4.4

roxctl CLI

roxctl CLI

Last Updated: 2024-06-27

Red Hat Advanced Cluster Security for Kubernetes 4.4 roxctl CLI

roxctl CLI

Legal Notice

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. INSTALLING THE ROXCTL CLI

roxctl is a command-line interface (CLI) for running commands on Red Hat Advanced Cluster Security for Kubernetes (RHACS). You can install the **roxctl** CLI by downloading the binary or you can run the **roxctl** CLI from a container image.

1.1. INSTALLING THE ROXCTL CLI BY DOWNLOADING THE BINARY

You can install the **roxctl** CLI to interact with RHACS from a command-line interface. You can install **roxctl** on Linux, Windows, or macOS.

1.1.1. Installing the roxctl CLI on Linux

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

NOTE

roxctl CLI for Linux is available for **amd64**, **ppc64le**, and **s390x** architectures.

Procedure

1. Determine the **roxctl** architecture for the target operating system:

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

2. Download the **roxctl** CLI:

```
$ curl -f -o roxctl "https://mirror.openshift.com/pub/rhacs/assets/4.4.3/bin/Linux/roxctl${arch}"
```

3. Make the **roxctl** binary executable:

```
$ chmod +x roxctl
```

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

```
$ echo $PATH
```

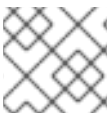
Verification

- Verify the **roxctl** version you have installed:

```
$ roxctl version
```

1.1.2. Installing the roxctl CLI on macOS

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



NOTE

roxctl CLI for macOS is available for the **amd64** architecture.

Procedure

1. Download the **roxctl** CLI:

```
$ curl -f -O https://mirror.openshift.com/pub/rhacs/assets/4.4.3/bin/Darwin/roxctl
```

2. Remove all extended attributes from the binary:

```
$ xattr -c roxctl
```

3. Make the **roxctl** binary executable:

```
$ chmod +x roxctl
```

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

```
$ echo $PATH
```

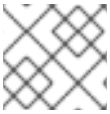
Verification

- Verify the **roxctl** version you have installed:

```
$ roxctl version
```

1.1.3. Installing the roxctl CLI on Windows

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



NOTE

roxctl CLI for Windows is available for the **amd64** architecture.

Procedure

- Download the **roxctl** CLI:

```
$ curl -f -O https://mirror.openshift.com/pub/rhacs/assets/4.4.3/bin/Windows/roxctl.exe
```

Verification

- Verify the **roxctl** version you have installed:

```
$ roxctl version
```

1.2. RUNNING THE ROXCTL CLI FROM A CONTAINER

The **roxctl** client is the default entry point in the RHACS **roxctl** image. To run the **roxctl** client in a container image:

Prerequisites

- You must first generate an authentication token from the RHACS portal.

Procedure

1. Log in to the **registry.redhat.io** registry.

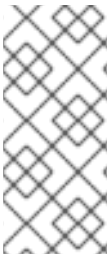
```
$ docker login registry.redhat.io
```

2. Pull the latest container image for the **roxctl** CLI.

```
$ docker pull registry.redhat.io/advanced-cluster-security/rhacs-roxctl-rhel8:4.4.3
```

After you install the CLI, you can run it by using the following command:

```
$ docker run -e ROX_API_TOKEN=$ROX_API_TOKEN \
  -it registry.redhat.io/advanced-cluster-security/rhacs-roxctl-rhel8:4.4.3 \
  -e $ROX_CENTRAL_ADDRESS <command>
```



NOTE

In Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service), when using **roxctl** commands that require the Central address, use the **Central instance address** as displayed in the **Instance Details** section of the Red Hat Hybrid Cloud Console. For example, use **acs-ABCD12345.acs.rhcloud.com** instead of **acs-data-ABCD12345.acs.rhcloud.com**.

Verification

- Verify the **roxctl** version you have installed.

```
$ docker run -it registry.redhat.io/advanced-cluster-security/rhacs-roxctl-rhel8:4.4.3 version
```

CHAPTER 2. USING THE ROXCTL CLI

2.1. PREREQUISITES

- You have configured the **ROX_ENDPOINT** environment variable using the following command:

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

1 The host and port information that you want to store in the **ROX_ENDPOINT** environment variable.

2.2. GETTING AUTHENTICATION INFORMATION

The following procedure describes how to use the **roxctl central whoami** command to retrieve information about your authentication status and user profile in Central. The example output illustrates the data you can expect to see, including user roles, access permissions, and various administrative functions. This step allows you to review your access and roles within Central.

Procedure

- Run the following command to get information about your current authentication status and user information in Central:

```
$ roxctl central whoami
```

Example output

```
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
```

Review the output to ensure that the authentication and user details are as expected.

2.3. AUTHENTICATING BY USING THE ROXCTL CLI

For authentication, you can use an API token, your administrator password, or the **roxctl central login** command.

Follow these guidelines for the effective use of API tokens:

- Use an API token in a production environment with continuous integration (CI). Each token is assigned specific access permissions, providing control over the actions it can perform. In addition, API tokens do not require interactive processes, such as browser-based logins, making them ideal for automated processes. These tokens have a time-to-live (TTL) value of 1 year, providing a longer validity period for seamless integration and operational efficiency.
- Use your administrator password only for testing purposes. Do not use it in the production environment.
- Use the **roxctl central login** command only for interactive, local uses.

NOTE

- To prevent privilege escalation, when you create a new token, your role's permissions limit the permission you can assign to that token. For example, if you only have **read** permission for the Integration resource, you cannot create a token with **write** permission.
- If you want a custom role to create tokens for other users to use, you must assign the required permissions to that custom role.
- Use short-lived tokens for machine-to-machine communication, such as CI/CD pipelines, scripts, and other automation. Also, use the **roxctl central login** command for human-to-machine communication, such as **roxctl** CLI or API access.

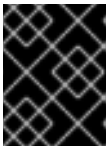
Additional resources

- Configuring API tokens
- Configuring short-lived access

2.3.1. Creating an API token

Procedure

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



IMPORTANT

Copy the generated token and securely store it. You will not be able to view it again.

2.3.2. Exporting and saving the API token

Procedure

1. After you have generated the authentication token, export it as the **ROX_API_TOKEN** variable by entering the following command:

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

2. (Optional): You can also save the token in a file and use it with the **--token-file** option by entering the following command:

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

Note the following guidelines:

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

2.3.3. Using an authentication provider to authenticate with roxctl

You can configure an authentication provider in Central and initiate the login process with the **roxctl** CLI. Set the **ROX_ENDPOINT** variable, initiate the login process with the **roxctl central login** command, select the authentication provider in a browser window, and retrieve the token information from the **roxctl** CLI as described in the following procedure.

Prerequisite

- You selected an authentication provider of your choice, such as OpenID Connect (OIDC) with fragment or query mode.

Procedure

1. Run the following command to set the **ROX_ENDPOINT** variable to Central hostname and port:

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

2. Run the following command to initiate the login process to Central:

```
$ roxctl central login
```

3. Within the **roxctl** CLI, a URL is printed as output and you are redirected to a browser window where you can select the authentication provider you want to use.

4. Log in with your authentication provider.
After you have successfully logged in, the browser window indicates that authentication was successful and you can close the browser window.

5. The **roxctl** CLI displays your token information including details such as the access token, the expiration time of the access token, the refresh token if one has been issued, and notification that these values are stored locally.

Example output

```
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. The access token.
2. The expiration time of the access token.
3. The refresh token.
4. The directory where values of the access token, the access token expiration time, and the refresh token are stored locally.

IMPORTANT

Ensure that you set the environment to determine the directory where the configuration is stored. By default, the configuration is stored in the **$HOME/.roxctl/roxctl-config** directory.

- If you set the **$ROX_CONFIG_DIR** environment variable, the configuration is stored in the **$ROX_CONFIG_DIR/roxctl-config** directory. This option has the highest priority.
- If you set the **$XDG_RUNTIME_DIR** environment variable and the **$ROX_CONFIG_DIR** variable is not set, the configuration is stored in the **$XDG_RUNTIME_DIR /roxctl-config** directory.
- If you do not set the **$ROX_CONFIG_DIR** or **$XDG_RUNTIME_DIR** environment variable, the configuration is stored in the **$HOME/.roxctl/roxctl-config** directory.

2.4. CONFIGURING AND USING THE ROXCTL CLI IN RHACS CLOUD SERVICE

Procedure

- Export the **ROX_API_TOKEN** by running the following command:

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

- Export the **ROX_ENDPOINT** by running the following command:

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

- You can use the **--help** option to get more information about the commands.
- In Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service), when using **roxctl** commands that require the Central address, use the **Central instance address** as displayed in the **Instance Details** section of the Red Hat Hybrid Cloud Console. For example, use **acs-ABCD12345.acs.rhcloud.com** instead of **acs-data-ABCD12345.acs.rhcloud.com**.

CHAPTER 3. MANAGING SECURED CLUSTERS

To secure a Kubernetes or an OpenShift Container Platform cluster, you must deploy Red Hat Advanced Cluster Security for Kubernetes (RHACS) 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.

3.1. PREREQUISITES

- You have configured the **ROX_ENDPOINT** environment variable using the following command:

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

1 The host and port information that you want to store in the **ROX_ENDPOINT** environment variable.

3.2. GENERATING SENSOR DEPLOYMENT FILES

Generating files for Kubernetes systems

Procedure

- Generate the required sensor configuration for your Kubernetes cluster and associate it with your Central instance by running the following command:

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

Generating files for OpenShift Container Platform systems

Procedure

- Generate the required sensor configuration for your OpenShift Container Platform cluster and associate it with your Central instance by running the following command:

```
$ roxctl sensor generate openshift --openshift-version <ocp_version> --name
<cluster_name> --central "$ROX_ENDPOINT" 1
```

1 For the **--openshift-version** option, specify the major OpenShift Container Platform version number for your cluster. For example, specify **3** for OpenShift Container Platform version **3.x** and specify **4** for OpenShift Container Platform version **4.x**.

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

Verify 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.

IMPORTANT

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

- 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
```

3.3. INSTALLING SENSOR BY USING THE SENSOR.SH SCRIPT

When you generate the Sensor deployment files, **roxctl** creates a directory called **sensor-<cluster_name>** in your working directory. The script to install Sensor is located in this directory.

Procedure

- Run the sensor installation script to install Sensor:

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

If you get a warning that you do not have the required permissions to install Sensor, follow the on-screen instructions, or contact your cluster administrator for help.

3.4. DOWNLOADING SENSOR BUNDLES FOR EXISTING CLUSTERS

Procedure

- Run the following command to download Sensor bundles for existing clusters by specifying a **cluster name** or **ID**:

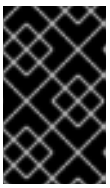
```
$ roxctl sensor get-bundle <cluster_name_or_id>
```

3.5. DELETING CLUSTER INTEGRATION

Procedure

- Before deleting the cluster, ensure you have the correct cluster name that you want to remove from Central:

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



IMPORTANT

Deleting the cluster integration does not remove the RHACS services running in the cluster, depending on the installation method. You can remove the services by running the **delete-sensor.sh** script from the Sensor installation bundle.

CHAPTER 4. CHECKING POLICY COMPLIANCE

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

4.1. PREREQUISITES

- You have configured the **ROX_ENDPOINT** environment variable using the following command:

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

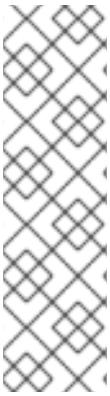
1 The host and port information that you want to store in the **ROX_ENDPOINT** environment variable.

4.2. CONFIGURING OUTPUT FORMAT

When you check policy compliance by using the **roxctl deployment check** or **roxctl image check** commands, you can specify the output format by using the **-o** option to the command and specifying the format as **json**, **table**, **csv**, or **junit**. This option determines how the output of a command is displayed in the terminal.

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

```
$ roxctl 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**. Use the old JSON format output when gathering data for troubleshooting purposes.

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

Option	Description	Formats
--compact-output	Use this option to display the JSON output in a compact format.	**json**
--headers	Use this option to specify custom headers.	**table** and **csv**
--no-header	Use this option to omit the header row from the output.	**table** and **csv**

Option	Description	Formats
--row-jsonpath-expressions	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: `$ roxctl deployment check --file=<yaml_filename> \ -o table --headers POLICY-NAME,SEVERITY \ --row-jsonpath-expressions="{results..violatedPolicies..name,results..violatedPolicies..severity}"`	**table** and **csv**
--merge-output	Use this options to merge table cells that have the same value.	**table**
headers-as-comment	Use this option to include the header row as a comment in the output.	**csv**
--junit-suite-name	Use this option to specify the name of the JUnit test suite.	**junit**

4.3. CHECKING DEPLOYMENT YAML FILES

Procedure

- Run the following command to check the build-time and deploy-time violations of your security policies in YAML deployment files:

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

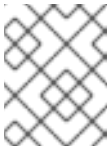
1. For the **<yaml_filename>**, specify the YAML file with one or more deployments to send to Central for policy evaluation. You can also specify multiple YAML files to send to Central for policy evaluation by using the **--file** flag, for example **--file=<yaml_filename1>**, **--file=<yaml_filename2>**, and so on.

2. For the **<cluster_namespace>**, specify a namespace to enhance deployments with context information such as network policies, role-based access controls (RBACs) and services for deployments that do not have a namespace in their specification. The namespace defined in the specification is not changed. The default value is **default**.

3. For the **<cluster_name_or_id>**, specify the cluster name or ID that you want to use as the context for the evaluation to enable extended deployments with cluster-specific information.

4. By enabling the **--verbose** flag, you receive additional information for each deployment during the policy check. The extended information includes the RBAC permission level and

during the policy check. The extended information includes the RBAC permission level and a comprehensive list of network policies that is applied.



NOTE

You can see the additional information for each deployment in your JSON output, regardless of whether you enable the **--verbose** flag or not.

The format is defined in the API reference. To cause Red Hat Advanced Cluster Security for Kubernetes (RHACS) 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.

This command validates the following items:

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

4.4. CHECKING IMAGES

Procedure

- Run the following command to check the build-time violations of your security policies in images:

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

The format is defined in the API reference. To cause Red Hat Advanced Cluster Security for Kubernetes (RHACS) 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.

Additional resources

- roxctl image

4.5. CHECKING IMAGE SCAN RESULTS

You can also check the scan results for specific images.

Procedure

- Run the following command to return the components and vulnerabilities found in the image in JSON format:

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

The format is defined in the API reference. To cause Red Hat Advanced Cluster Security for Kubernetes (RHACS) 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.

Additional resources

- roxctl image

CHAPTER 5. DEBUGGING ISSUES

Central saves information to its container logs.

5.1. PREREQUISITES

- You have configured the **ROX_ENDPOINT** environment variable using the following command:

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

1 The host and port information that you want to store in the **ROX_ENDPOINT** environment variable.

5.2. VIEWING THE LOGS

You can use either the **oc** or **kubectl** command to view the logs for the Central pod.

Procedure

- To view the logs for the Central pod by using **kubectl**, run the following command :

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

- To view the logs for the Central pod by using **oc**, run the following command :

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

5.3. VIEWING THE CURRENT LOG LEVEL

You can change the log level to see more or less information in Central logs.

Procedure

- Run the following command to view the current log level:

```
$ roxctl central debug log
```

Additional resources

- roxctl central debug

5.4. CHANGING THE LOG LEVEL

Procedure

- Run the following command to change the log level:

```
$ roxctl central debug log --level=<log_level> 1
```

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

Additional resources

- roxctl central debug

5.5. RETRIEVING DEBUGGING INFORMATION

Procedure

- Run the following command to gather the debugging information for investigating issues:

```
$ roxctl central debug dump
```

- To generate a diagnostic bundle with the RHACS administrator password or API token and central address, follow the procedure in Generating a diagnostic bundle by using the roxctl CLI .

Additional resources

- roxctl central debug

CHAPTER 6. GENERATING BUILD-TIME NETWORK POLICIES

The build-time network policy generator is included in the **roxctl** CLI. For the build-time network policy generation feature, **roxctl** CLI does not need to communicate with RHACS Central so you can use it in any development environment.

6.1. USING THE BUILD-TIME NETWORK POLICY GENERATOR

Prerequisites

1. The build-time network policy generator recursively scans the directory you specify when you run the command. Therefore, before you run the command, you must already have service manifests, config maps, and workload manifests such as **Pod**, **Deployment**, **ReplicaSet**, **Job**, **DaemonSet**, and **StatefulSet** as YAML files in the specified directory.

2. Verify that you can apply these YAML files as-is using the **kubectl apply -f** command. The build-time network policy generator does not work with files that use Helm-style templating.

3. Verify that the service network addresses are not hardcoded. Every workload that needs to connect to a service must specify the service network address as a variable. You can specify this variable by using the workload's resource environment variable or in a config map.

   - Example 1: using an environment variable
   - Example 2: using a config map
   - Example 3: using a config map

4. Service network addresses must match the following official regular expression pattern:

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

   1 In this pattern,

   - <svc> is the service name.
   - <ns> is the namespace where you defined the service.
   - <portNum> is the exposed service port number.

   Following are some examples that match the pattern:

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

Procedure

1. Verify that the build-time network policy generation feature is available by running the help command:

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

2. Generate the policies by using the **netpol generate** command:

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

1 Specify the path of the folder that has the Kubernetes manifests.

The **roxctl netpol generate** command supports the following options:

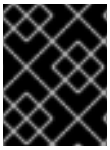
Option	Description
-h, --help	View the help text for the **netpol** command.
-d, --output-dir <dir>	Save the generated policies into a target folder. One file per policy.
-f, --output-file <filename>	Save and merge the generated policies into a single YAML file.
--fail	Fail on the first encountered error. The default value is **false**.
--remove	Remove the output path if it already exist.
--strict	Treat warnings as errors. The default value is **false**.

CHAPTER 7. IMAGE SCANNING BY USING THE ROXCTL CLI

You can scan images stored in image registries, including cluster local registries such as the OpenShift Container Platform integrated image registry by using the **roxctl** CLI.

7.1. SCANNING IMAGES BY USING A REMOTE CLUSTER

By specifying the appropriate cluster in the delegated scanning configuration or through the cluster parameter described in the following procedure, you can scan images from cluster local registries by using a remote cluster.



IMPORTANT

For more information about how to configure delegated image scanning, see Configuring delegated image scanning.

Procedure

- Run the following command to scan the specified image in a remote cluster:

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

1. For **<image_registry>**, specify the registry where the image is located, for example, **image-registry.openshift-image-registry.svc:5000**/. For **<image_name>**, specify the name of the image you want to scan, for example, **default/image-stream:latest**.
2. For **<cluster_detail>**, specify the name or ID of the remote cluster. For example, specify the name **remote**.
3. Optional: For **[flags]**, you can specify parameters to modify the behavior of the command.

   For more information about optional parameters, see roxctl image scan command options .

Example output

```
{
  "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. A unique identifier for the image that serves as a fingerprint for the image. It helps ensure the integrity and authenticity of the image.

2. Contains specific details about the image.
3. The location of the image registry where the image is stored.
4. The remote path to the image.
5. The version or tag associated with this image.
6. The complete name of the image, combining the registry, remote path, and tag.

7.2. ROXCTL IMAGE SCAN COMMAND OPTIONS

The **roxctl image scan** command supports the following options:

Option	Description
--cluster string	Delegate image scanning to a specific cluster.
--compact-output	Print the JSON output in a compact format. The default value is **false**.
-f, --force	Ignore Central's cache for the scan and force a fresh re-pull from Scanner. The default value is **false**.
--headers strings	Print the headers in a tabular format. Default values include **COMPONENT**,**VERSION**,**CVE**,**SEVERITY**, and **LINK**.
--headers-as-comments	Print the headers as comments in a CSV tabular output. The default value is **false**.
-h, --help	View the help text for the **roxctl image scan** command.
-i, --image string	Specify the image name and reference you want to scan.
-a, --include-snoozed	Return both snoozed and unsnoozed common vulnerabilities and exposures (CVEs). The default value is **false**.
--merge-output	Merge duplicate cells in a tabular output. The default value is **true**.
--no-header	Do not print headers for tabular format. The default value is **false**.

Option	Description
-o, **--output string**	Specify the output format. You can select a format to customize the display of results. Formats include **table**, **CSV**, **JSON**, and **SARIF**.
-r, **--retries int**	Set the number of retries before the operation is aborted with an error. The default value is **3**.
-d, **--retry-delay int**	Set the time in seconds to wait between retries. The default value is **3**.
--row-jsonpath-expressions string	Use the JSON path expressions to create rows from the JSON object. For more details, run the **roxctl image scan --help** command.

CHAPTER 8. ROXCTL CLI COMMAND REFERENCE

8.1. ROXCTL

Display the available commands and optional parameters for **roxctl** CLI. You must have an account with administrator privileges to use these commands.

Usage

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

Table 8.1. Available commands

Command	Description
central	Commands related to the Central service.
cluster	Commands related to a cluster.
collector	Commands related to the Collector service.
completion	Generate shell completion scripts.
declarative-config	Manage declarative configuration.
deployment	Commands related to deployments.
helm	Commands related to Red Hat Advanced Cluster Security for Kubernetes (RHACS) Helm Charts.
image	Commands that you can run on a specific image.
netpol	Commands related to network policies.
scanner	Commands related to the Scanner service.
sensor	Deploy RHACS services in secured clusters.
version	Display the current roxctl version.

8.1.1. roxctl command options

The **roxctl** command supports the following options:

Option	Description

Option	Description
--ca string	Specify a custom CA certificate file path for secure connections. Alternatively, you can specify the file path by using the **ROX_CA_CERT_FILE** environment variable.
--direct-grpc	Set **--direct-grpc** for improved connection performance. Alternatively, by setting the **ROX_DIRECT_GRPC_CLIENT** environment variable to **true**, you can enable direct gRPC . The default value is **false**.
-e, **--endpoint string**	Set the endpoint for the service to contact. Alternatively, you can set the endpoint by using the **ROX_ENDPOINT** environment variable. The default value is **localhost:8443**.
--force-http1	Force the use of HTTP/1 for all connections. Alternatively, by setting the **ROX_CLIENT_FORCE_HTTP1** environment variable to **true**, you can force the use of HTTP/1. The default value is **false**.
--insecure	Enable insecure connection options. Alternatively, by setting the **ROX_INSECURE_CLIENT** environment variable to **true**, you can enable insecure connection options. The default value is **false**.
--insecure-skip-tls-verify	Skip the TLS certificate validation. Alternatively, by setting the **ROX_INSECURE_CLIENT_SKIP_TLS_VERIFY** environment variable to **true**, you can skip the TLS certificate validation. The default value is **false**.
--no-color	Disable the color output. Alternatively, by setting the **ROX_NO_COLOR environment variable** to **true**, you can disable the color output. The default value is **false**.
-p, **--password string**	Specify the password for basic authentication. Alternatively, you can set the password by using the **ROX_ADMIN_PASSWORD** environment variable.
--plaintext	Use an unencrypted connection. Alternatively, by setting the **ROX_PLAINTEXT** environment variable to **true**, you can enable an unencrypted connection. The default value is **false**.

Option	Description
-s, **--server-name string**	Set the TLS server name to use for SNI. Alternatively, you can set the server name by using the **ROX_SERVER_NAME** environment variable.
--token-file string	Use the API token provided in the specified file for authentication. Alternatively, you can set the token by using the **ROX_API_TOKEN** environment variable.

8.2. ROXCTL CENTRAL

Commands related to the Central service.

Usage

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

Table 8.2. Available commands

Command	Description
backup	Create a backup of the Red Hat Advanced Cluster Security for Kubernetes (RHACS) database and the certificates.
cert	Download the certificate chain for the Central service.
db	Control the database operations.
debug	Debug the Central service.
generate	Generate the required YAML configuration files containing the orchestrator objects for the deployment of Central.
init-bundles	Initialize bundles for Central.
login	Log in to the Central instance to obtain a token.
userpki	Manage the user certificate authorization providers.
whoami	Display information about the current user and their authentication method.

8.2.1. roxctl central command options inherited from the parent command

The **roxctl central** command supports the following options inherited from the parent **roxctl** command:

Option	Description
--ca string	Specify a custom CA certificate file path for secure connections. Alternatively, you can specify the file path by using the **ROX_CA_CERT_FILE** environment variable.
--direct-grpc	Set **--direct-grpc** for improved connection performance. Alternatively, by setting the **ROX_DIRECT_GRPC_CLIENT** environment variable to **true**, you can enable direct gRPC . The default value is **false**.
-e, **--endpoint string**	Set the endpoint for the service to contact. Alternatively, you can set the endpoint by using the **ROX_ENDPOINT** environment variable. The default value is **localhost:8443**.
--force-http1	Force the use of HTTP/1 for all connections. Alternatively, by setting the **ROX_CLIENT_FORCE_HTTP1** environment variable to **true**, you can force the use of HTTP/1. The default value is **false**.
--insecure	Enable insecure connection options. Alternatively, by setting the **ROX_INSECURE_CLIENT** environment variable to **true**, you can enable insecure connection options. The default value is **false**.
--insecure-skip-tls-verify	Skip the TLS certificate validation. Alternatively, by setting the **ROX_INSECURE_CLIENT_SKIP_TLS_VERIFY** environment variable to **true**, you can skip the TLS certificate validation. The default value is **false**.
--no-color	Disable the color output. Alternatively, by setting the **ROX_NO_COLOR environment variable** to **true**, you can disable the color output. The default value is **false**.
-p, **--password string**	Specify the password for basic authentication. Alternatively, you can set the password by using the **ROX_ADMIN_PASSWORD** environment variable.
--plaintext	Use an unencrypted connection. Alternatively, by setting the **ROX_PLAINTEXT** environment variable to **true**, you can enable an unencrypted connection. The default value is **false**.

Option	Description
-s, **--server-name string**	Set the TLS server name to use for SNI. Alternatively, you can set the server name by using the **ROX_SERVER_NAME** environment variable.
--token-file string	Use the API token provided in the specified file for authentication. Alternatively, you can set the token by using the **ROX_API_TOKEN** environment variable.

NOTE

These options are applicable to all the sub-commands of the **roxctl central** command.

8.2.2. roxctl central backup

Create a backup of the RHACS database and certificates.

Usage

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

Table 8.3. Options

Option	Description
--certs-only	Specify to only back up the certificates. When using an external database, this option is used to generate a backup bundle with certificates. The default value is **false**.
--output string	Specify where you want to save the backup. The behavior depends on the specified path: • If the path is a file path, the backup is written to the file and overwrites it if it already exists. The directory must exist. • If the path is a directory, the backup is saved in this directory under the file name that the server specifies. • If this argument is omitted, the backup is saved in the current working directory under the file name that the server specifies.
-t, **--timeout duration**	Specify the timeout for API requests. It represents the maximum duration of a request. The default value is **1h0m0s**.

8.2.3. roxctl central cert

Download the certificate chain for the Central service.

Usage

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

Table 8.4. Options

Option	Description
--output string	Specify the file name to which you want to save the PEM certificate. You can generate a standard output by using **-**. The default value is **-**.
--retry-timeout duration	Specify the timeout after which API requests are retried. A value of zero means that the entire request duration is waited for without retrying. The default value is **20s**.
-t, --timeout duration	Specify the timeout for API requests representing the maximum duration of a request. The default value is **1m0s**.

8.2.4. roxctl central login

Login to the Central instance to obtain a token.

Usage

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

Table 8.5. Options

Option	Description
-t, --timeout duration	Specify the timeout for API requests representing the maximum duration of a request. The default value is **5m0s**.

8.2.5. roxctl central whoami

Display information about the current user and their authentication method.

Usage

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

Table 8.6. Options

Option	Description
--retry-timeout duration	Specify the timeout after which API requests are retried. A value of zero means that the entire request duration is waited for without retrying. The default value is **20s**.
-t, --timeout duration	Specify the timeout for API requests representing the maximum duration of a request. The default value is **1m0s**.

8.2.6. roxctl central db

Control the database operations.

Usage

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

Table 8.7. Options

Option	Description
-t, --timeout duration	Specify the timeout for API requests representing the maximum duration of a request. The default value is **1h0m0s**.

8.2.6.1. roxctl central db restore

Restore the RHACS database from a previous backup.

Usage

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

1 For **<file>**, specify the database backup file that you want to restore.

Table 8.8. Options

Option	Description
-f, --force	If set to **true**, the restoration is performed without confirmation. The default value is **false**.
--interrupt	If set to **true**, it interrupts the running restore process to allow it to continue. The default value is **false**.

8.2.6.2. roxctl central db generate

Generate a Central database bundle.

Usage

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

Table 8.9. Options

Option	Description
--debug	If set to **true**, templates are read from the local file system. The default value is **false**.
--debug-path string	Specify the path to the Helm templates in your local file system. For more details, run the **roxctl central db generate** command.
--enable-pod-security-policies	If set to **true**, **PodSecurityPolicy** resources are created. The default value is **true**.

8.2.6.3. roxctl central db generate k8s

Generate Kubernetes YAML files for deploying Central's database components.

Usage

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

Table 8.10. Options

Option	Description
--central-db-image string	Specify the Central database image that you want to use. If not specified, a default value corresponding to the **--image-defaults** is used.
--image-defaults string	Specify the default settings for container images. It controls the repositories from which the images are downloaded, the image names and the format of the tags. The default value is **development_build**.
--output-dir output directory	Specify the directory to which you want to save the deployment bundle. The default value is **central-db-bundle**.

8.2.6.4. roxctl central db restore cancel

Cancel the ongoing Central database restore process.

Usage

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

Table 8.11. Options

Option	Description
f, **--force**	If set to **true**, proceed with the cancellation without confirmation. The default value is **false**.

8.2.6.5. roxctl central db restore status

Display information about the ongoing database restore process.

Usage

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

8.2.6.6. roxctl central db generate k8s pvc

Generate Kubernetes YAML files for persistent volume claims (PVCs) in Central.

Usage

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

Table 8.12. Options

Option	Description
--name string	Specify the external volume name for the Central database. The default value is **central-db**.
--size uint32	Specify the external volume size in gigabytes for the Central database. The default value is **100**.
--storage-class string	Specify the storage class name for the Central database. This is optional if you have a default storage class configured.

8.2.6.7. roxctl central db generate openshift

Generate an OpenShift YAML manifest for deploying a Central database instance on a Red Hat OpenShift cluster.

Usage

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

Table 8.13. Options

Option	Description
--central-db-image string	Specify the Central database image that you want to use. If not specified, a default value corresponding to the **--image-defaults** is used.
--image-defaults string	Specify the default settings for container images. It controls the repositories from which the images are downloaded, the image names and the format of the tags. The default value is **development_build**.
--openshift-version int	Specify the Red Hat OpenShift major version 3 or 4 for the deployment. The default value is **3**.
--output-dir output-directory	Specify the directory to which you want to save the deployment bundle. The default value is **central-db-bundle**.

8.2.6.8. roxctl central db generate k8s hostpath

Generate a Kubernetes YAML manifest for a database deployment with a hostpath volume type in Central.

Usage

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

Table 8.14. Options

Option	Description
--hostpath string	Specify the path on the host. The default value is **/var/lib/stackrox-central-db**.
--node-selector-key string	Specify the node selector key. Valid values include **kubernetes.io** and **hostname**.
--node-selector-value string	Specify the node selector value.

8.2.6.9. roxctl central db generate openshift pvc

Generate an OpenShift YAML manifest for a database deployment with a persistent volume claim (PVC) in Central.

Usage

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

Table 8.15. Options

Option	Description
--name string	Specify the external volume name for the Central database. The default value is **central-db**.
--size uint32	Specify the external volume size in gigabytes for the Central database. The default value is **100**.
--storage-class string	Specify the storage class name for the Central database. This is optional if you have a default storage class configured.

8.2.6.10. roxctl central db generate openshift hostpath

Add a hostpath external volume to the Central database.

Usage

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

Table 8.16. Options

Option	Description
--hostpath string	Specify the path on the host. The default value is **/var/lib/stackrox-central-db**.
--node-selector-key string	Specify the node selector key. Valid values include **kubernetes.io** and **hostname**.
--node-selector-value string	Specify the node selector value.

8.2.7. roxctl central debug

Debug the Central service.

Usage

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

8.2.7.1. roxctl central debug db

Control the debugging of the database.

Usage

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

Table 8.17. Options

Option	Description
-t, **--timeout duration**	Specify the timeout for API requests representing the maximum duration of a request. The default value is **1m0s**.

8.2.7.2. roxctl central debug log

Retrieve the current log level.

Usage

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

Table 8.18. Options

Option	Description
-l, **--level string**	Specify the log level to which you want to set the modules. Valid values include **Debug**, **Info**, **Warn**, **Error**, **Panic**, and **Fatal**.
-m, **--modules strings**	Specify the modules to which you want to apply the command.
--retry-timeout duration	Specify the timeout after which API requests are retried. A value of zero means that the entire request duration is waited for without retrying. The default value is **20s**.
-t, **--timeout duration**	Specify the timeout for API requests, which is the maximum duration of a request. The default value is **1m0s**.

8.2.7.3. roxctl central debug dump

Download a bundle containing the debug information for Central.

Usage

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

Table 8.19. Options

Option	Description
--logs	If set to **true**, logs are included in the Central dump. The default value is **false**.

Option	Description
--output-dir string	Specify the output directory for the bundle content. The default value is an automatically generated directory name within the current directory.
-t, --timeout duration	Specify the timeout for API requests, which is the maximum duration of a request. The default value is **5m0s**.

8.2.7.4. roxctl central debug db stats

Control the statistics of the Central database.

Usage

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

8.2.7.5. roxctl central debug authz-trace

Enable or disable authorization tracing in Central for debugging purposes.

Usage

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

Table 8.20. Options

Option	Description
-t, --timeout duration	Specify the timeout for API requests representing the maximum duration of a request. The default value is **20m0s**.

8.2.7.6. roxctl central debug db stats reset

Reset the statistics of the Central database.

Usage

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

8.2.7.7. roxctl central debug download-diagnostics

Download a bundle containing a snapshot of diagnostic information about the platform.

Usage

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

Table 8.21. Options

Option	Description
--clusters strings	Specify a comma-separated list of the Sensor clusters from which you want to collect the logs.
--output-dir string	Specify the output directory in which you want to save the diagnostic bundle.
--since string	Specify the timestamp from which you want to collect the logs from the Sensor clusters.
-t, **--timeout duration**	Specify the timeout for API requests, which specifies the maximum duration of a request. The default value is **5m0s**.

8.2.8. roxctl central generate

Generate the required YAML configuration files that contain the orchestrator objects to deploy Central.

Usage

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

Table 8.22. Options

Option	Description
--backup-bundle string	Specify the path to the backup bundle from which you want to restore the keys and certificates.
--debug	If set to **true**, templates are read from the local file system. The default value is **false**.
--debug-path string	Specify the path to Helm templates on your local file system. For more details, run the **roxctl central generate --help** command.
--default-tls-certfile	Specify the PEM certificate bundle file that you want to use as the default.
--default-tls-keyfile	Specify the PEM private key file that you want to use as the default.
--enable-pod-security-policies	If set to **true**, **PodSecurityPolicy** resources are created. The default value is **true**.
-p, **--password string**	Specify the administrator password. The default value is automatically generated.

Option	Description
--plaintext-endpoints string	Specify the ports or endpoints you want to use for unencrypted exposure as a comma-separated list.

8.2.8.1. roxctl central generate k8s

Generate the required YAML configuration files to deploy Central into a Kubernetes cluster.

Usage

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

Table 8.23. Options

Option	Description
--central-db-image string	Specify the Central database image you want to use. If not specified, a default value corresponding to the **--image-defaults** is used.
--declarative-config-config-maps strings	Specify a list of configuration maps that you want to add as declarative configuration mounts in Central.
--declarative-config-secrets strings	Specify a list of secrets that you want to add as declarative configuration mounts in Central.
--enable-telemetry	Specify whether you want to enable telemetry. The default value is **false**.
--image-defaults string	Specify the default settings for container images. The specified settings control the repositories from which the images are downloaded, the image names and the format of the tags. The default value is **development_build**.

Option	Description
--istio-support version	Generate deployment files that support the specified Istio version. Valid values include **1.0**, **1.1**, **1.2**, **1.3**, **1.4**, **1.5**, **1.6**, and **1.7**.
--lb-type load balancer type	Specify the method in which you want to suspend Central. Valid values include **lb**, **np** and **none**. The default value is **none**.
-i, **--main-image string**	Specify the main image that you want to use. If not specified, a default value corresponding to the **--image-defaults** is used.
--offline	Specify whether you want to run RHACS in offline mode, avoiding a connection to the Internet. The default value is **false**.
--output-dir output directory	Specify the directory to which you want to save the deployment bundle. The default value is **central-bundle**.
--output-format output format	Specify the deployment tool that you want to use. Valid values include **kubectl**, **helm**, and **helm-values**. The default value is **kubectl**.
--scanner-db-image string	Specify the Scanner database image that you want to use. If not specified, a default value corresponding to the **--image-defaults** is used.
--scanner-image string	Specify the Scanner image that you want to use. If not specified, a default value corresponding to the `--image-defaults" is used.

8.2.8.2. roxctl central generate k8s pvc

Generate Kubernetes YAML files for persistent volume claims (PVCs) in Central.

Usage

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

Table 8.24. Options

Option	Description
--db-name string	Specify the external volume name for the Central database. The default value is **central-db**.
--db-size uint32	Specify the external volume size in gigabytes for the Central database. The default value is **100**.
--db-storage-class string	Specify the storage class name for the Central database. This is optional if you have a default storage class configured.
--name string	Specify the external volume name for Central. The default value is **stackrox-db**.
--size uint32	Specify the external volume size in gigabytes for Central. The default value is **100**.
--storage-class string	Specify the storage class name for Central. This is optional if you have a default storage class configured.

8.2.8.3. roxctl central generate openshift

Generate the required YAML configuration files to deploy Central in a Red Hat OpenShift cluster.

Usage

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

Table 8.25. Options

Option	Description
--central-db-image string	Specify the Central database image that you want to use. If not specified, a default value is created corresponding to the **--image-defaults**.
--declarative-config-config-maps strings	Specify a list of configuration maps that you want to add as declarative configuration mounts in Central.

Option	Description
--declarative-config-secrets strings	Specify a list of secrets that you want to add as declarative configuration mounts in Central.
--enable-telemetry	Specify whether you want to enable telemetry. The default value is **false**.
--image-defaults string	Specify the default settings for container images. It controls the repositories from which the images are downloaded, the image names and the format of the tags. The default value is **development_build**.
--istio-support version	Generate deployment files that support the specified Istio version. Valid values include **1.0**, **1.1**, **1.2**, **1.3**, **1.4**, **1.5**, **1.6**, and **1.7**.
--lb-type load balancer type	Specify the method of exposing Central. Valid values include **route**, **lb**, **np** and **none**. The default value is **none**.
-i, **--main-image string**	Specify the main image that you want to use. If not specified, a default value corresponding to **--image-defaults** is used.
--offline	Specify whether you want to run RHACS in offline mode, avoiding a connection to the Internet. The default value is **false**.
--openshift-monitoring false|true|auto[=true]	Specify integration with Red Hat OpenShift 4 monitoring. The default value is **auto**.
--openshift-version int	Specify the Red Hat OpenShift major version 3 or 4 for the deployment.

Option	Description
--output-dir output directory	Specify the directory to which you want to save the deployment bundle. The default value is **central-bundle**.
--output-format output format	Specify the deployment tool that you want to use. Valid values include **kubectl**, **helm** and **helm-values**. The default value is **kubectl**.
--scanner-db-image string	Specify the Scanner database image that you want to use. If not specified, a default value corresponding to the **--image-defaults** is used.
--scanner-image string	Specify the Scanner image that you want to use. If not specified, a default value corresponding to **--image-defaults** is used.

8.2.8.4. roxctl central generate interactive

Generate interactive resources in Central.

Usage

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

8.2.8.5. roxctl central generate k8s hostpath

Generate a Kubernetes YAML manifest for deploying a Central instance by using the hostpath volume type.

Usage

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

Table 8.26. Options

Option	Description
--db-hostpath string	Specify the path on the host for the Central database. The default value is **/var/lib/stackrox-central**.

Option	Description
--db-node-selector-key string	Specify the node selector key for the Central database. Valid values include **kubernetes.io** and **hostname**.
--db-node-selector-value string	Specify the node selector value for the Central database.
--hostpath string	Specify the path on the host. The default value is **/var/lib/stackrox**.
--node-selector-key string	Specify the node selector key. Valid values include **kubernetes.io** and **hostname**.
--node-selector-value string	Specify the node selector value.

8.2.8.6. roxctl central generate openshift pvc

Generate a OpenShift YAML manifest for deploying a persistent volume claim (PVC) in Central.

Usage

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

Table 8.27. Options

Option	Description
--db-name string	Specify the external volume name for the Central database. The default value is **central-db**.
--db-size uint32	Specify the external volume size in gigabytes for the Central database. The default value is **100**.
--db-storage-class string	Specify the storage class name for the Central database. This is optional if you have a default storage class configured.
--name string	Specify the external volume name for Central. The default value is **stackrox-db**.
--size uint32	Specify the external volume size in gigabytes for Central. The default value is **100**.

Option	Description
--storage-class string	Specify the storage class name for Central. This is optional if you have a default storage class configured.

8.2.8.7. roxctl central generate openshift hostpath

Add a hostpath external volume to the deployment definition in Red Hat OpenShift.

Usage

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

Table 8.28. Options

Option	Description
--db-hostpath string	Specify the path on the host for the Central database. The default value is **/var/lib/stackrox-central**.
--db-node-selector-key string	Specify the node selector key. Valid values include **kubernetes.io** and **hostname** for the Central database.
--db-node-selector-value string	Specify the node selector value for the Central database.
--hostpath string	Specify the path on the host. The default value is **/var/lib/stackrox**.
--node-selector-key string	Specify the node selector key. Valid values include **kubernetes.io** and **hostname**.
--node-selector-value string	Specify the node selector value.

8.2.9. roxctl central init-bundles

Initialize bundles in Central.

Usage

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

Table 8.29. Options

Option	Description
--retry-timeout duration	Specify the timeout after which API requests are retried. A value of **0s** means that the entire request duration is waited for without retrying. The default value is **20s**.
-t, --timeout duration	Specify the timeout for API requests representing the maximum duration of a request. The default value is **1m0s**.

8.2.9.1. roxctl central init-bundles list

List the available initialization bundles in Central.

Usage

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

8.2.9.2. roxctl central init-bundles revoke

Revoke one or more cluster initialization bundles in Central.

Usage

```
$ roxctl central init-bundles revoke <init_bundle_ID or name> [<init_bundle_ID or name> ...] [flags]
```
1

1 For **<init_bundle_ID or name>**, specify the ID or the name of the initialization bundle that you want to revoke. You can provide multiple IDs or names separated by using spaces.

8.2.9.3. roxctl central init-bundles fetch-ca

Fetch the certificate authority (CA) bundle from Central.

Usage

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

Table 8.30. Options

Option	Description
--output string	Specify the file that you want to use for storing the CA configuration.

8.2.9.4. roxctl central init-bundles generate

Generate a new cluster initialization bundle.

Usage

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

1 For **<init_bundle_name>**, specify the name for the initialization bundle you want to generate.

Table 8.31. Options

Option	Description
--output string	Specify the file you want to use for storing the newly generated initialization bundle in the Helm configuration form. You can generate a standard output by using **-**.
--output-secrets string	Specify the file that you want to use for storing the newly generated initialization bundle in Kubernetes secret form. You can generate a standard by using **-**.

8.2.10. roxctl central userpki

Manage the user certificate authorization providers.

Usage

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

8.2.10.1. roxctl central userpki list

Display all the user certificate authentication providers.

Usage

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

Table 8.32. Options

Option	Description
-j, **--json**	Enable the JSON output. The default value is **false**.
--retry-timeout duration	Specify the timeout after which API requests are retried. A value of zero means that the entire request duration is waited for without retrying. The default value is **20s**.
-t, **--timeout duration**	Specify the timeout for API requests representing the maximum duration of a request. The default value is **1m0s**.

8.2.10.2. roxctl central userpki create

Create a new user certificate authentication provider.

Usage

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

Table 8.33. Options

Option	Description
-c, **--cert strings**	Specify the PEM files of the root CA certificates. You can specify several certificate files.
--retry-timeout duration	Specify the timeout after which API requests are retried. A value of zero means that the entire request duration is waited for without retrying. The default value is **20s**.
-r, **--role string**	Specify the minimum access role for users of this provider.
-t, **--timeout duration**	Specify the timeout for API requests representing the maximum duration of a request. The default value is **1m0s**.

8.2.10.3. roxctl central userpki delete

Delete a user certificate authentication provider.

Usage

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

Table 8.34. Options

Option	Description
-f, **--force**	If set to **true**, proceed with the deletion without confirmation. The default value is **false**.
--retry-timeout duration	Specify the timeout after which API requests are retried. A value of zero means that the entire request duration is waited for without retrying. The default value is **20s**.
-t, **--timeout duration**	Specify the timeout for API requests representing the maximum duration of a request. The default value is **1m0s**.

8.3. ROXCTL CLUSTER

Commands related to a cluster.

Usage

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

Table 8.35. Available commands

Command	Description
delete	Remove Sensor from Central.

Table 8.36. Options

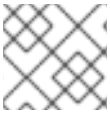
Option	Description
--retry-timeout duration	Set the retry timeout for API requests. A value of zero means the full request duration is awaited without retry. The default value is **20s**.
-t, **--timeout duration**	Set the timeout for API requests representing the maximum duration of a request. The default value is **1m0s**.

8.3.1. roxctl cluster command options inherited from the parent command

The **roxctl cluster** command supports the following options inherited from the parent **roxctl** command:

Option	Description
--ca string	Specify a custom CA certificate file path for secure connections. Alternatively, you can specify the file path by using the **ROX_CA_CERT_FILE** environment variable.
--direct-grpc	Set **--direct-grpc** for improved connection performance. Alternatively, by setting the **ROX_DIRECT_GRPC_CLIENT** environment variable to **true**, you can enable direct gRPC . The default value is **false**.
-e, **--endpoint string**	Set the endpoint for the service to contact. Alternatively, you can set the endpoint by using the **ROX_ENDPOINT** environment variable. The default value is **localhost:8443**.

Option	Description
--force-http1	Force the use of HTTP/1 for all connections. Alternatively, by setting the **ROX_CLIENT_FORCE_HTTP1** environment variable to **true**, you can force the use of HTTP/1. The default value is **false**.
--insecure	Enable insecure connection options. Alternatively, by setting the **ROX_INSECURE_CLIENT** environment variable to **true**, you can enable insecure connection options. The default value is **false**.
--insecure-skip-tls-verify	Skip the TLS certificate validation. Alternatively, by setting the **ROX_INSECURE_CLIENT_SKIP_TLS_VERIFY** environment variable to **true**, you can skip the TLS certificate validation. The default value is **false**.
--no-color	Disable the color output. Alternatively, by setting the **ROX_NO_COLOR environment variable** to **true**, you can disable the color output. The default value is **false**.
-p, **--password string**	Specify the password for basic authentication. Alternatively, you can set the password by using the **ROX_ADMIN_PASSWORD** environment variable.
--plaintext	Use an unencrypted connection. Alternatively, by setting the **ROX_PLAINTEXT** environment variable to **true**, you can enable an unencrypted connection. The default value is **false**.
-s, **--server-name string**	Set the TLS server name to use for SNI. Alternatively, you can set the server name by using the **ROX_SERVER_NAME** environment variable.
--token-file string	Use the API token provided in the specified file for authentication. Alternatively, you can set the token by using the **ROX_API_TOKEN** environment variable.



NOTE

These options are applicable to all the sub-commands of the **roxctl cluster** command.

8.3.2. roxctl cluster delete

Remove Sensor from Central.

Usage

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

Table 8.37. Options

Option	Description
--name string	Specify the cluster name to delete.

8.4. ROXCTL COLLECTOR

Commands related to the Collector service.

Usage

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

Table 8.38. Available commands

Command	Description
support-packages	Upload support packages for Collector.

8.4.1. roxctl collector command options inherited from the parent command

The **roxctl collector** command supports the following options inherited from the parent **roxctl** command:

Option	Description
--ca string	Specify a custom CA certificate file path for secure connections. Alternatively, you can specify the file path by using the **ROX_CA_CERT_FILE** environment variable.
--direct-grpc	Set **--direct-grpc** for improved connection performance. Alternatively, by setting the **ROX_DIRECT_GRPC_CLIENT** environment variable to **true**, you can enable direct gRPC . The default value is **false**.
-e, **--endpoint string**	Set the endpoint for the service to contact. Alternatively, you can set the endpoint by using the **ROX_ENDPOINT** environment variable. The default value is **localhost:8443**.

Option	Description
--force-http1	Force the use of HTTP/1 for all connections. Alternatively, by setting the **ROX_CLIENT_FORCE_HTTP1** environment variable to **true**, you can force the use of HTTP/1. The default value is **false**.
--insecure	Enable insecure connection options. Alternatively, by setting the **ROX_INSECURE_CLIENT** environment variable to **true**, you can enable insecure connection options. The default value is **false**.
--insecure-skip-tls-verify	Skip the TLS certificate validation. Alternatively, by setting the **ROX_INSECURE_CLIENT_SKIP_TLS_VERIFY** environment variable to **true**, you can skip the TLS certificate validation. The default value is **false**.
--no-color	Disable the color output. Alternatively, by setting the **ROX_NO_COLOR environment variable** to **true**, you can disable the color output. The default value is **false**.
-p, **--password string**	Specify the password for basic authentication. Alternatively, you can set the password by using the **ROX_ADMIN_PASSWORD** environment variable.
--plaintext	Use an unencrypted connection. Alternatively, by setting the **ROX_PLAINTEXT** environment variable to **true**, you can enable an unencrypted connection. The default value is **false**.
-s, **--server-name string**	Set the TLS server name to use for SNI. Alternatively, you can set the server name by using the **ROX_SERVER_NAME** environment variable.
--token-file string	Use the API token provided in the specified file for authentication. Alternatively, you can set the token by using the **ROX_API_TOKEN** environment variable.

NOTE

These options are applicable to all the sub-commands of the **roxctl collector** command.

8.4.2. roxctl collector support-packages

Upload support packages for Collector.

Usage

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

8.4.2.1. roxctl collector support-packages upload

Upload files from a Collector support package to Central.

Usage

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

Table 8.39. Options

Option	Description
--overwrite	Specify whether you want to overwrite existing but different files. The default value is **false**.
--retry-timeout duration	Set the timeout after which API requests are retried. A value of zero means that the entire request duration is waited for without retrying. The default value is **20s**.
-t, **--timeout duration**	Set the timeout for API requests. This option represents the maximum duration of a request. The default value is **1m0s**.

8.5. ROXCTL COMPLETION

Generate shell completion scripts.

Usage

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

Table 8.40. Supported shell types

Shell type	Description
bash	Generate a completion script for the Bash shell.
zsh	Generate a completion script for the Zsh shell.
fish	Generate a completion script for the Fish shell.
powershell	Generate a completion script for the PowerShell shell.

8.5.1. roxctl completion command options inherited from the parent command

The **roxctl completion** command supports the following options inherited from the parent **roxctl** command:

Option	Description
--ca string	Specify a custom CA certificate file path for secure connections. Alternatively, you can specify the file path by using the **ROX_CA_CERT_FILE** environment variable.
--direct-grpc	Set **--direct-grpc** for improved connection performance. Alternatively, by setting the **ROX_DIRECT_GRPC_CLIENT** environment variable to **true**, you can enable direct gRPC . The default value is **false**.
-e, **--endpoint string**	Set the endpoint for the service to contact. Alternatively, you can set the endpoint by using the **ROX_ENDPOINT** environment variable. The default value is **localhost:8443**.
--force-http1	Force the use of HTTP/1 for all connections. Alternatively, by setting the **ROX_CLIENT_FORCE_HTTP1** environment variable to **true**, you can force the use of HTTP/1. The default value is **false**.
--insecure	Enable insecure connection options. Alternatively, by setting the **ROX_INSECURE_CLIENT** environment variable to **true**, you can enable insecure connection options. The default value is **false**.
--insecure-skip-tls-verify	Skip the TLS certificate validation. Alternatively, by setting the **ROX_INSECURE_CLIENT_SKIP_TLS_VERIFY** environment variable to **true**, you can skip the TLS certificate validation. The default value is **false**.
--no-color	Disable the color output. Alternatively, by setting the **ROX_NO_COLOR environment variable** to **true**, you can disable the color output. The default value is **false**.
-p, **--password string**	Specify the password for basic authentication. Alternatively, you can set the password by using the **ROX_ADMIN_PASSWORD** environment variable.

Option	Description
--plaintext	Use an unencrypted connection. Alternatively, by setting the **ROX_PLAINTEXT** environment variable to **true**, you can enable an unencrypted connection. The default value is **false**.
-s, **--server-name string**	Set the TLS server name to use for SNI. Alternatively, you can set the server name by using the **ROX_SERVER_NAME** environment variable.
--token-file string	Use the API token provided in the specified file for authentication. Alternatively, you can set the token by using the **ROX_API_TOKEN** environment variable.

8.6. ROXCTL DECLARATIVE-CONFIG

Manage the declarative configuration.

Usage

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

Table 8.41. Available commands

Command	Description
create	Create declarative configurations.
lint	Lint an existing declarative configuration YAML file.

8.6.1. roxctl declarative-config command options inherited from the parent command

The **roxctl declarative-config** command supports the following options inherited from the parent **roxctl** command:

Option	Description
--ca string	Specify a custom CA certificate file path for secure connections. Alternatively, you can specify the file path by using the **ROX_CA_CERT_FILE** environment variable.

Option	Description
--direct-grpc	Set **--direct-grpc** for improved connection performance. Alternatively, by setting the **ROX_DIRECT_GRPC_CLIENT** environment variable to **true**, you can enable direct gRPC . The default value is **false**.
-e, **--endpoint string**	Set the endpoint for the service to contact. Alternatively, you can set the endpoint by using the **ROX_ENDPOINT** environment variable. The default value is **localhost:8443**.
--force-http1	Force the use of HTTP/1 for all connections. Alternatively, by setting the **ROX_CLIENT_FORCE_HTTP1** environment variable to **true**, you can force the use of HTTP/1. The default value is **false**.
--insecure	Enable insecure connection options. Alternatively, by setting the **ROX_INSECURE_CLIENT** environment variable to **true**, you can enable insecure connection options. The default value is **false**.
--insecure-skip-tls-verify	Skip the TLS certificate validation. Alternatively, by setting the **ROX_INSECURE_CLIENT_SKIP_TLS_VERIFY** environment variable to **true**, you can skip the TLS certificate validation. The default value is **false**.
--no-color	Disable the color output. Alternatively, by setting the **ROX_NO_COLOR environment variable** to **true**, you can disable the color output. The default value is **false**.
-p, **--password string**	Specify the password for basic authentication. Alternatively, you can set the password by using the **ROX_ADMIN_PASSWORD** environment variable.
--plaintext	Use an unencrypted connection. Alternatively, by setting the **ROX_PLAINTEXT** environment variable to **true**, you can enable an unencrypted connection. The default value is **false**.
-s, **--server-name string**	Set the TLS server name to use for SNI. Alternatively, you can set the server name by using the **ROX_SERVER_NAME** environment variable.

Option	Description
--token-file string	Use the API token provided in the specified file for authentication. Alternatively, you can set the token by using the **ROX_API_TOKEN** environment variable.

NOTE

These options are applicable to all the sub-commands of the **roxctl declarative-config** command.

8.6.2. roxctl declarative-config lint

Lint an existing declarative configuration YAML file.

Usage

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

Table 8.42. Options

Option	Description
--config-map string	Read the declarative configuration from the **--config-map string**. If not specified, the configuration is read from the YAML file specified by using the **--file** flag.
-f, --file string	File containing the declarative configuration in YAML format.
--namespace string	Read the declarative configuration from the **--namespace string** of the configuration map. If not specified, the namespace specified in the current Kubernetes configuration context is used.
--secret string	Read the declarative configuration from the specified **--secret string**. If not specified, the configuration is read from the YAML file specified by using the **--file** flag.

8.6.3. roxctl declarative-config create

Create declarative configurations.

Usage

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

Table 8.43. Options

Option	Description
--config-map string	Write the declarative configuration YAML in the configuration map. If not specified and the **--secret** flag is also not specified, the generated YAML is printed in the standard output format.
--namespace string	Required if you want to write the declarative configuration YAML to a configuration map or secret. If not specified, the default namespace in the current Kubernetes configuration is used.
--secret string	Write the declarative configuration YAML in the Secret. You must use secrets for sensitive data. If not specified and the **--config-map** flag is also not specified, the generated YAML is printed in the standard output format.

8.6.3.1. roxctl declarative-config create role

Create a declarative configuration for a role.

Usage

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

Table 8.44. Options

Option	Description
--access-scope string	By providing the name, you can specify the referenced access scope.
--description string	Set a description for the role.
--name string	Specify the name of the role.
--permission-set string	By providing the name, you can specify the referenced permission set.

8.6.3.2. roxctl declarative-config create notifier

Create a declarative configuration for a notifier.

Usage

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

Table 8.45. Options

Option	Description
--name string	Specify the name of the notifier.

8.6.3.3. roxctl declarative-config create access-scope

Create a declarative configuration for an access scope.

Usage

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

Table 8.46. Options

Option	Description
--cluster-label-selector requirement	Specify the criteria for creating a label selector based on the cluster's labels. The key-value pairs represent requirements, and you can use this flag multiple times to create a combination of requirements. The default value is **[[ ]]**. For more details, run the **roxctl declarative-config create access-scope --help** command.
--description string	Set a description for the access scope.
--included included-object	Specify a list of clusters and their namespaces that you want to include in the access scope. The default value is **[null]**.
--name string	Specify the name of the access scope.
--namespace-label-selector requirement	Specify the criteria for creating a label selector based on the namespace's labels. Similar to the cluster-label-selector, you can use this flag multiple times for the combination of requirements. For more details, run the **roxctl declarative-config create access-scope --help** command.

8.6.3.4. roxctl declarative-config create auth-provider

Create a declarative configuration for an authentication provider.

Usage

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

Table 8.47. Options

Option	Description
--extra-ui-endpoints strings	Specify additional user interface (UI) endpoints from which the authentication provider is used. The expected format is **<endpoint>:<port>**.
--groups-key strings	Set the keys of the groups that you want to add within the authentication provider. The tuples of key, value and role should have the same length. For more details, run the **roxctl declarative-config create auth-provider --help** command.
--groups-role strings	Set the role of the groups that you want to add within the authentication provider. The tuples of key, value and role should have the same length. For more details, run the **roxctl declarative-config create auth-provider --help** command.
--groups-value strings	Set the values of the groups that you want to add within the authentication provider. The tuples of key, value and role should have the same length. For more details, run the **roxctl declarative-config create auth-provider --help** command.

Option	Description
--minimum-access-role string	Set the minimum access role of the authentication provider. You can leave this field empty if you do not want to configure the minimum access role by using the declarative configuration.
--name string	Specify the name of the authentication provider.
--required-attributes stringToString	Set a list of attributes that the authentication provider must return during authentication. The default value is **[]**.
--ui-endpoint string	Set the UI endpoint from which the authentication provider is used. This is usually the public endpoint where RHACS is available. The expected format is **<endpoint>:<port>**.

8.6.3.5. roxctl declarative-config create permission-set

Create a declarative configuration for a permission set.

Usage

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

Table 8.48. Options

Option	Description
--description string	Set the description of the permission set.
--name string	Specify the name of the permission set.
--resource-with-access stringToString	Set a list of resources with their respective access levels. The default value is **[]**. For more details, run the **roxctl declarative-config create permission-set --help** command.

8.6.3.6. roxctl declarative-config create notifier splunk

Create a declarative configuration for a splunk notifier.

Usage

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

Table 8.49. Options

Option	Description
--audit-logging	Enable audit logging. The default value is **false**.
--source-types stringToString	Specify Splunk source types as comma-separated **key=value** pairs. The default value is **[]**.
--splunk-endpoint string	Specify the Splunk HTTP endpoint. This is a mandatory option.
--splunk-skip-tls-verify	Use an insecure connection to Splunk. The default value is **false**.
--splunk-token string	Specify the Splunk HTTP token. This is a mandatory option.
--truncate int	Specify the Splunk truncate limit. The default value is **10000**.

8.6.3.7. roxctl declarative-config create notifier generic

Create a declarative configuration for a generic notifier.

Usage

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

Table 8.50. Options

Option	Description
--audit-logging	Enable audit logging. The default value is **false**.
--extra-fields stringToString	Specify additional fields as comma-separated **key=value** pairs. The default value is **[]**.
--headers stringToString	Specify headers as comma-separated **key=value** pairs. The default value is **[]**.

Option	Description
--webhook-cacert-file string	Specify the file name of the endpoint CA certificate in PEM format.
--webhook-endpoint string	Specify the URL of the webhook endpoint.
--webhook-password string	Specify the password for basic authentication of the webhook endpoint. No authentication if not specified. Requires **--webhook-username**.
--webhook-skip-tls-verify	Skip webhook TLS verification. The default value is **false**.
--webhook-username string	Specify the username for basic authentication of the webhook endpoint. No authentication occurs if not specified. Requires **--webhook-password**.

8.6.3.8. roxctl declarative-config create auth-provider iap

Create a declarative configuration for an authentication provider with the identity-aware proxy (IAP) identifier.

Usage

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

Table 8.51. Options

Option	Description
--audience string	Specify the target group that you want to validate.

8.6.3.9. roxctl declarative-config create auth-provider oidc

Create a declarative configuration for an OpenID Connect (OIDC) authentication provider.

Usage

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

Table 8.52. Options

Option	Description

Option	Description
--claim-mappings stringToString	Specify a list of non-standard claims from the identity provider (IdP) token that you want to include in the authentication provider's rules. The default value is **[]**.
--client-id string	Specify the client ID of the OIDC client.
--client-secret string	Specify the client secret of the OIDC client.
--disable-offline-access	Disable the request for the offline_access from the OIDC IdP. You need to use this option if the OIDC IdP limits the number of sessions with the **offline_access** scope. The default value is **false**.
--issuer string	Specify the issuer of the OIDC client.
--mode string	Specify the callback mode that you want to use. Valid values include **auto**, **post**, **query** and **fragment**. The default value is **auto**.

8.6.3.10. roxctl declarative-config create auth-provider saml

Create a declarative configuration for a SAML authentication provider.

Usage

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

Table 8.53. Options

Option	Description
--idp-cert string	Specify the file containing the SAML identity provider (IdP) certificate in PEM format.
--idp-issuer string	Specify the issuer of the IdP.
--metadata-url string	Specify the metadata URL of the service provider.

Option	Description
--name-id-format string	Specify the format of the name ID.
--sp-issuer string	Specify the issuer of the service provider.
--sso-url string	Specify the URL of the IdP for single sign-on (SSO).

8.6.3.11. roxctl declarative-config create auth-provider userpki

Create a declarative configuration for an user PKI authentication provider.

Usage

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

Table 8.54. Options

Option	Description
--ca-file string	Specify the file containing the certification authorities in PEM format.

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

Create a declarative configuration for an OpenShift Container Platform OAuth authentication provider.

Usage

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

8.7. ROXCTL DEPLOYMENT

Commands related to deployments.

Usage

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

Table 8.55. Available commands

Command	Description
check	Check the deployments for violations of the deployment time policy.

Table 8.56. Options

Option	Description
-t, **--timeout duration**	Set the timeout for API requests. This option represents the maximum duration of a request. The default value is **10m0s**.

8.7.1. roxctl deployment command options inherited from the parent command

The **roxctl deployment** command supports the following options inherited from the parent **roxctl** command:

Option	Description
--ca string	Specify a custom CA certificate file path for secure connections. Alternatively, you can specify the file path by using the **ROX_CA_CERT_FILE** environment variable.
--direct-grpc	Set **--direct-grpc** for improved connection performance. Alternatively, by setting the **ROX_DIRECT_GRPC_CLIENT** environment variable to **true**, you can enable direct gRPC . The default value is **false**.
-e, **--endpoint string**	Set the endpoint for the service to contact. Alternatively, you can set the endpoint by using the **ROX_ENDPOINT** environment variable. The default value is **localhost:8443**.
--force-http1	Force the use of HTTP/1 for all connections. Alternatively, by setting the **ROX_CLIENT_FORCE_HTTP1** environment variable to **true**, you can force the use of HTTP/1. The default value is **false**.
--insecure	Enable insecure connection options. Alternatively, by setting the **ROX_INSECURE_CLIENT** environment variable to **true**, you can enable insecure connection options. The default value is **false**.
--insecure-skip-tls-verify	Skip the TLS certificate validation. Alternatively, by setting the **ROX_INSECURE_CLIENT_SKIP_TLS_VERIFY** environment variable to **true**, you can skip the TLS certificate validation. The default value is **false**.

Option	Description
--no-color	Disable the color output. Alternatively, by setting the **ROX_NO_COLOR environment variable** to **true**, you can disable the color output. The default value is **false**.
-p, **--password string**	Specify the password for basic authentication. Alternatively, you can set the password by using the **ROX_ADMIN_PASSWORD** environment variable.
--plaintext	Use an unencrypted connection. Alternatively, by setting the **ROX_PLAINTEXT** environment variable to **true**, you can enable an unencrypted connection. The default value is **false**.
-s, **--server-name string**	Set the TLS server name to use for SNI. Alternatively, you can set the server name by using the **ROX_SERVER_NAME** environment variable.
--token-file string	Use the API token provided in the specified file for authentication. Alternatively, you can set the token by using the **ROX_API_TOKEN** environment variable.

NOTE

These options are applicable to all the sub-commands of the **roxctl deployment** command.

8.7.2. roxctl deployment check

Check deployments for violations of the deployment time policy.

Usage

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

Table 8.57. Options

Option	Description
-c, **--categories strings**	Define the policy categories that you want to execute. By default, all policy categories are executed.
--cluster string	Set the cluster name or ID that you want to use as the context for the evaluation to enable extended deployments with cluster-specific information.

Option	Description
--compact-output	Print the JSON output in compact form. The default value is **false**.
-f, **--file stringArray**	Specify the YAML files to send to Central for policy evaluation.
--force	Bypass the Central cache for images and force a new pull from Scanner. The default value is **false**.
--headers strings	Define headers that you want to print in the tabular output. The default values include **POLICY**, **SEVERITY**, **BREAKS DEPLOY**, **DEPLOYMENT**, **DESCRIPTION**, **VIOLATION**, and **REMEDIATION**.
--headers-as-comments	Print headers as comments in the CSV tabular output. The default value is **false**.
--junit-suite-name string	Set the name of the JUnit test suite. The default value is **deployment-check**.
--merge-output	Merge duplicate cells in the tabular output. The default value is **false**.
-n, **--namespace string**	Specify a namespace to enhance deployments with context information such as network policies, RBACs and services for deployments that do not have a namespace in their specification. The namespace defined in the specification is not changed. The default value is **default**.
--no-header	Do not print headers for a tabular output. The default value is **false**.
-o, **--output string**	Choose the output format. Output formats include **json**, **junit**, **sarif**, **table**, and **csv**. The default value is **table**.
-r, **--retries int**	Set the number of retries before exiting as an error. The default value is **3**.
-d, **--retry-delay int**	Set the time to wait between retries in seconds. The default value is **3**.
--row-jsonpath-expressions string	Define the JSON path expressions to create a row from the JSON object. For more details, run the **roxctl deployment check --help** command.

8.8. ROXCTL HELM

Commands related to Red Hat Advanced Cluster Security for Kubernetes (RHACS) Helm Charts.

Usage

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

Table 8.58. Available commands

Command	Description
derive-local-values	Derive local Helm values from the cluster configuration.
output	Output a Helm chart.

8.8.1. roxctl helm command options inherited from the parent command

The **roxctl helm** command supports the following options inherited from the parent **roxctl** command:

Option	Description
--ca string	Specify a custom CA certificate file path for secure connections. Alternatively, you can specify the file path by using the **ROX_CA_CERT_FILE** environment variable.
--direct-grpc	Set **--direct-grpc** for improved connection performance. Alternatively, by setting the **ROX_DIRECT_GRPC_CLIENT** environment variable to **true**, you can enable direct gRPC . The default value is **false**.
-e, **--endpoint string**	Set the endpoint for the service to contact. Alternatively, you can set the endpoint by using the **ROX_ENDPOINT** environment variable. The default value is **localhost:8443**.
--force-http1	Force the use of HTTP/1 for all connections. Alternatively, by setting the **ROX_CLIENT_FORCE_HTTP1** environment variable to **true**, you can force the use of HTTP/1. The default value is **false**.
--insecure	Enable insecure connection options. Alternatively, by setting the **ROX_INSECURE_CLIENT** environment variable to **true**, you can enable insecure connection options. The default value is **false**.

Option	Description
--insecure-skip-tls-verify	Skip the TLS certificate validation. Alternatively, by setting the **ROX_INSECURE_CLIENT_SKIP_TLS_VERIFY** environment variable to **true**, you can skip the TLS certificate validation. The default value is **false**.
--no-color	Disable the color output. Alternatively, by setting the **ROX_NO_COLOR environment variable** to **true**, you can disable the color output. The default value is **false**.
-p, **--password string**	Specify the password for basic authentication. Alternatively, you can set the password by using the **ROX_ADMIN_PASSWORD** environment variable.
--plaintext	Use an unencrypted connection. Alternatively, by setting the **ROX_PLAINTEXT** environment variable to **true**, you can enable an unencrypted connection. The default value is **false**.
-s, **--server-name string**	Set the TLS server name to use for SNI. Alternatively, you can set the server name by using the **ROX_SERVER_NAME** environment variable.
--token-file string	Use the API token provided in the specified file for authentication. Alternatively, you can set the token by using the **ROX_API_TOKEN** environment variable.

NOTE

These options are applicable to all the sub-commands of the **roxctl helm** command.

8.8.2. roxctl helm output

Output a Helm chart.

Usage

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

1 For **<central_services or secured_cluster_services>**, specify the path to either the central services or the secured cluster services to generate a Helm chart output.

Table 8.59. Options

Option	Description
--debug	Read templates from the local filesystem. The default value is **false**.
--debug-path string	Specify the path to the Helm templates on your local filesystem. For more details, run the **roxctl helm output --help** command.
--image-defaults string	Set the default container image settings. Image settings include **development_build**, **stackrox.io**, **rhacs**, and **opensource**. It influences repositories for image downloads, image names, and tag formats. The default value is **development_build**.
--output-dir string	Define the path to the output directory for the Helm chart. The default path is **./stackrox-<chart name>-chart**.
--remove	Remove the output directory if it already exists. The default value is **false**.

8.8.3. roxctl helm derive-local-values

Derive local Helm values from the cluster configuration.

Usage

```
$ roxctl helm derive-local-values --output <path> \ 1
<central_services> [flags] 2
```

1 For the **<path>**, specify the path where you want to save the generated local values file.

2 For the **<central_services>**, specify the path to the central services configuration file.

Table 8.60. Options

Option	Description
--input string	Specify the path to the file or directory containing the YAML input.
--output string	Define the path to the output file.
--output-dir string	Define the path to the output directory.

Option	Description
--retry-timeout duration	Set the timeout after which API requests are retried. The timeout value indicates that the entire request duration is waited for without retrying. The default value is **20s**.
-t, --timeout duration	Set the timeout for API requests representing the maximum duration of a request. The default value is **1m0s**.

8.9. ROXCTL IMAGE

Commands that you can run on a specific image.

Usage

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

Table 8.61. Available commands

Command	Description
check	Check images for build time policy violations, and report them.
scan	Scan the specified image, and return the scan results.

Table 8.62. Options

-t, --timeout duration	**Set the timeout for API requests representing the maximum duration of a request. The default value is 10m0s.**

8.9.1. roxctl image command options inherited from the parent command

The **roxctl image** command supports the following options inherited from the parent **roxctl** command:

Option	Description
--ca string	Specify a custom CA certificate file path for secure connections. Alternatively, you can specify the file path by using the **ROX_CA_CERT_FILE** environment variable.

Option	Description
--direct-grpc	Set **--direct-grpc** for improved connection performance. Alternatively, by setting the **ROX_DIRECT_GRPC_CLIENT** environment variable to **true**, you can enable direct gRPC . The default value is **false**.
-e, **--endpoint string**	Set the endpoint for the service to contact. Alternatively, you can set the endpoint by using the **ROX_ENDPOINT** environment variable. The default value is **localhost:8443**.
--force-http1	Force the use of HTTP/1 for all connections. Alternatively, by setting the **ROX_CLIENT_FORCE_HTTP1** environment variable to **true**, you can force the use of HTTP/1. The default value is **false**.
--insecure	Enable insecure connection options. Alternatively, by setting the **ROX_INSECURE_CLIENT** environment variable to **true**, you can enable insecure connection options. The default value is **false**.
--insecure-skip-tls-verify	Skip the TLS certificate validation. Alternatively, by setting the **ROX_INSECURE_CLIENT_SKIP_TLS_VERIFY** environment variable to **true**, you can skip the TLS certificate validation. The default value is **false**.
--no-color	Disable the color output. Alternatively, by setting the **ROX_NO_COLOR environment variable** to **true**, you can disable the color output. The default value is **false**.
-p, **--password string**	Specify the password for basic authentication. Alternatively, you can set the password by using the **ROX_ADMIN_PASSWORD** environment variable.
--plaintext	Use an unencrypted connection. Alternatively, by setting the **ROX_PLAINTEXT** environment variable to **true**, you can enable an unencrypted connection. The default value is **false**.
-s, **--server-name string**	Set the TLS server name to use for SNI. Alternatively, you can set the server name by using the **ROX_SERVER_NAME** environment variable.

Option	Description
--token-file string	Use the API token provided in the specified file for authentication. Alternatively, you can set the token by using the **ROX_API_TOKEN** environment variable.

NOTE

These options are applicable to all the sub-commands of the **roxctl image** command.

8.9.2. roxctl image scan

Scan the specified image, and return the scan results.

Usage

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

Table 8.63. Options

Option	Description
--cluster string	Specify the cluster name or ID to which you want to delegate the image scan.
--compact-output	Print JSON output in a compact format. The default value is **false**.
-f, **--force**	Ignore Central's cache and force a fresh re-pull from Scanner. The default value is **false**.
--headers strings	Specify the headers to print in a tabular output. The default values include **COMPONENT**, **VERSION**,**CVE**,**SEVERITY**, and **LINK**.
--headers-as-comments	Print headers as comments in a CSV tabular output. The default value is **false**.
-i, **--image string**	Specify the image name and reference to scan. For example, **nginx:latest** or **nginx@sha256:...**.
-a, **--include-snoozed**	Include snoozed and unsnoozed CVEs in the scan results. The default value is **false**.
--merge-output	Merge duplicate cells in a tabular output. The default value is **true**.

Option	Description
--no-header	Do not print headers for a tabular output. The default value is **false**.
-o, **--output string**	Specify the output format. Output formats include **table**, **csv**, **json**, and **sarif**.
-r, **--retries int**	Specify the number of retries before exiting as an error. The default value is **3**.
-d, **--retry-delay int**	Set the time to wait between retries in seconds. The default value is **3**.
--row-jsonpath-expressions string	Specify JSON path expressions to create a row from the JSON object. For more details, run the **roxctl image scan --help** command.
--severity strings	List of severities to include in the output. Use this to filter for specific severities. The default values include **LOW**, **MODERATE**, **IMPORTANT**, and **CRITICAL**.

8.9.3. roxctl image check

Check images for build time policy violations, and report them.

Usage

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

Table 8.64. Options

Option	Description
-c, **--categories strings**	List of the policy categories that you want to execute. By default, all the policy categories are used.
--cluster string	Define the cluster name or ID that you want to use as the context for evaluation.
--compact-output	Print JSON output in a compact format. The default value is **false**.
-f, **--force**	Bypass the Central cache for the image and force a new pull from the Scanner. The default value is **false**.

Option	Description
--headers strings	Define headers to print in a tabular output. The default values include **POLICY**, **SEVERITY**, **BREAKS BUILD**, **DESCRIPTION**, **VIOLATION**, and **REMEDIATION**.
--headers-as-comments	Print headers as comments in a CSV tabular output. The default value is **false**.
-i, **--image string**	Specify the image name and reference. For example, **nginx:latest** or **nginx@sha256:...)**.
--junit-suite-name string	Set the name of the JUnit test suite. Default value is **image-check**.
--merge-output	Merge duplicate cells in a tabular output. The default value is **false**.
--no-header	Do not print headers for a tabular output. The default value is **false**.
-o, **--output string**	Choose the output format. Output formats include **junit**, **sarif**, **table**, **csv**, and **json**. The default value is **table**.
-r, **--retries int**	Set the number of retries before exiting as an error. The default value is **3**.
-d, **--retry-delay int**	Set the time to wait between retries in seconds. The default value is **3**.
--row-jsonpath-expressions string	Create a row from the JSON object by using JSON path expression. For more details, run the **roxctl image check --help** command.
--send-notifications	Define whether you want to send notifications in the event of violations. The default value is **false**.

8.10. ROXCTL NETPOL

Commands related to the network policies.

Usage

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

Table 8.65. Available commands

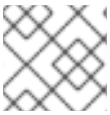
Command	Description
connectivity	Connectivity analysis of the network policy resources.
generate	Recommend network policies based on the deployment information.

8.10.1. roxctl netpol command options inherited from the parent command

The **roxctl netpol** command supports the following options inherited from the parent **roxctl** command:

Option	Description
--ca string	Specify a custom CA certificate file path for secure connections. Alternatively, you can specify the file path by using the **ROX_CA_CERT_FILE** environment variable.
--direct-grpc	Set **--direct-grpc** for improved connection performance. Alternatively, by setting the **ROX_DIRECT_GRPC_CLIENT** environment variable to **true**, you can enable direct gRPC . The default value is **false**.
-e, **--endpoint string**	Set the endpoint for the service to contact. Alternatively, you can set the endpoint by using the **ROX_ENDPOINT** environment variable. The default value is **localhost:8443**.
--force-http1	Force the use of HTTP/1 for all connections. Alternatively, by setting the **ROX_CLIENT_FORCE_HTTP1** environment variable to **true**, you can force the use of HTTP/1. The default value is **false**.
--insecure	Enable insecure connection options. Alternatively, by setting the **ROX_INSECURE_CLIENT** environment variable to **true**, you can enable insecure connection options. The default value is **false**.
--insecure-skip-tls-verify	Skip the TLS certificate validation. Alternatively, by setting the **ROX_INSECURE_CLIENT_SKIP_TLS_VERIFY** environment variable to **true**, you can skip the TLS certificate validation. The default value is **false**.

Option	Description
--no-color	Disable the color output. Alternatively, by setting the **ROX_NO_COLOR environment variable** to **true**, you can disable the color output. The default value is **false**.
-p, **--password string**	Specify the password for basic authentication. Alternatively, you can set the password by using the **ROX_ADMIN_PASSWORD** environment variable.
--plaintext	Use an unencrypted connection. Alternatively, by setting the **ROX_PLAINTEXT** environment variable to **true**, you can enable an unencrypted connection. The default value is **false**.
-s, **--server-name string**	Set the TLS server name to use for SNI. Alternatively, you can set the server name by using the **ROX_SERVER_NAME** environment variable.
--token-file string	Use the API token provided in the specified file for authentication. Alternatively, you can set the token by using the **ROX_API_TOKEN** environment variable.



NOTE

These options are applicable to all the sub-commands of the **roxctl netpol** command.

8.10.2. roxctl netpol generate

Recommend network policies based on the deployment information.

Usage

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

1 For **<folder_path>**, specify the path to the directory containing your Kubernetes deployment and service configuration files.

Table 8.66. Options

Option	Description
--fail	Fail on the first encountered error. The default value is **false**.

Option	Description
-d, **--output-dir string**	Save generated policies into the target folder.
-f, **--output-file string**	Save and merge generated policies into a single YAML file.
--remove	Remove the output path if it already exists. The default value is **false**.
--strict	Treat warnings as errors. The default value is **false**.

8.10.3. roxctl netpol connectivity

Commands related to the connectivity analysis of the network policy resources.

Usage

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

8.10.3.1. roxctl netpol connectivity map

Analyze connectivity based on the network policies and other resources.

Usage

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

1 For **<folder_path>**, specify the path to the directory containing your Kubernetes deployment and service configuration files.

Table 8.67. Options

Option	Description
--fail	Fail on the first encountered error. The default value is **false**.
--focus-workload string	Focus on connections of the specified workload name in the output.
-f, **--output-file string**	Save the connections list output into a specific file.
-o, **--output-format string**	Configure the connections list in a specific format. Supported formats include **txt**, **json**, **md**, **dot**, and **csv**. The default value is **txt**.

Option	Description
--remove	Remove the output path if it already exists. The default value is **false**.
--save-to-file	Define whether you want to save the output of the connection list in the default file. The default value is **false**.
--strict	Treat warnings as errors. The default value is **false**.

8.10.3.2. roxctl netpol connectivity diff

Report connectivity differences based on two network policy directories and YAML manifests with workload resources.

Usage

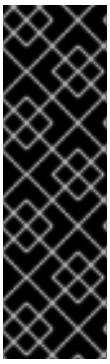
```
$ roxctl netpol connectivity diff [flags]
```

Table 8.68. Options

Option	Description
--dir1 string	Specify the first directory path of the input resources. This value is mandatory.
--dir2 string	Specify the second directory path of the input resources that you want to compare with the first directory path. This value is mandatory.
--fail	Fail on the first encounter. The default value is **false**.
-f, **--output-file string**	Save the output of the connectivity difference command into a specific file.
-o, **--output-format string**	Configure the output of the connectivity difference command in a specific format. Supported formats include **txt**, **md**, **csv**. The default value is **txt**..
--remove	Remove the output path if it already exists. The default value is **false**.
--save-to-file	Define whether you want to store the output of the connectivity differences in the default file. The default value is **false**.
--strict	Treat warnings as errors. The default value is **false**.

8.11. ROXCTL SCANNER

Commands related to the StackRox Scanner and Scanner V4 services.



IMPORTANT

Scanner V4 is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

Usage

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

Table 8.69. Available commands

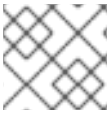
Command	Description
download-db	Download the offline vulnerability database for StackRox Scanner and Scanner V4.
generate	Generate the required YAML configuration files to deploy the StackRox Scanner and Scanner V4.
upload-db	Upload a vulnerability database for the StackRox Scanner and Scanner V4.

8.11.1. roxctl scanner command options inherited from the parent command

The **roxctl scanner** command supports the following options inherited from the parent **roxctl** command:

Option	Description
--ca string	Specify a custom CA certificate file path for secure connections. Alternatively, you can specify the file path by using the **ROX_CA_CERT_FILE** environment variable.
--direct-grpc	Set **--direct-grpc** for improved connection performance. Alternatively, by setting the **ROX_DIRECT_GRPC_CLIENT** environment variable to **true**, you can enable direct gRPC . The default value is **false**.

Option	Description
-e, **--endpoint string**	Set the endpoint for the service to contact. Alternatively, you can set the endpoint by using the **ROX_ENDPOINT** environment variable. The default value is **localhost:8443**.
--force-http1	Force the use of HTTP/1 for all connections. Alternatively, by setting the **ROX_CLIENT_FORCE_HTTP1** environment variable to **true**, you can force the use of HTTP/1. The default value is **false**.
--insecure	Enable insecure connection options. Alternatively, by setting the **ROX_INSECURE_CLIENT** environment variable to **true**, you can enable insecure connection options. The default value is **false**.
--insecure-skip-tls-verify	Skip the TLS certificate validation. Alternatively, by setting the **ROX_INSECURE_CLIENT_SKIP_TLS_VERIFY** environment variable to **true**, you can skip the TLS certificate validation. The default value is **false**.
--no-color	Disable the color output. Alternatively, by setting the **ROX_NO_COLOR environment variable** to **true**, you can disable the color output. The default value is **false**.
-p, **--password string**	Specify the password for basic authentication. Alternatively, you can set the password by using the **ROX_ADMIN_PASSWORD** environment variable.
--plaintext	Use an unencrypted connection. Alternatively, by setting the **ROX_PLAINTEXT** environment variable to **true**, you can enable an unencrypted connection. The default value is **false**.
-s, **--server-name string**	Set the TLS server name to use for SNI. Alternatively, you can set the server name by using the **ROX_SERVER_NAME** environment variable.
--token-file string	Use the API token provided in the specified file for authentication. Alternatively, you can set the token by using the **ROX_API_TOKEN** environment variable.



NOTE

These options are applicable to all the sub-commands of the **roxctl scanner** command.

8.11.2. roxctl scanner generate

Generate the required YAML configuration files to deploy Scanner.

Usage

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

Table 8.70. Options

Option	Description
--cluster-type cluster type	Specify the type of cluster on which you want to run Scanner. Cluster types include **k8s** and **openshift**. The default value is **k8s**.
--enable-pod-security-policies	Create **PodSecurityPolicy** resources. The default value is **true**.
--istio-support string	Generate deployment files that support the specified Istio version. Valid versions include **1.0**, **1.1**, **1.2**, **1.3**, **1.4**, **1.5**, **1.6**, and **1.7**.
--output-dir string	Specify the output directory for the Scanner bundle. Leave blank to use the default value.
--retry-timeout duration	Set the timeout after which API requests are retried. A value of zero means that the entire request duration is waited for without retrying. The default value is **20s**.
--scanner-image string	Specify the Scanner image that you want to use. Leave blank to use the server default.
-t, **--timeout duration**	Set the timeout for API requests representing the maximum duration of a request. The default value is **1m0s**.

8.11.3. roxctl scanner upload-db

Upload a vulnerability database for Scanner.

Usage

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

Table 8.71. Options

Option	Description
--scanner-db-file string	Specify the file containing the dumped Scanner definitions DB.
-t, --timeout duration	Set the timeout for API requests representing the maximum duration of a request. The default value is **10m0s**.

8.11.4. roxctl scanner download-db

Download the offline vulnerability database for StackRox Scanner or Scanner V4.

IMPORTANT

Scanner V4 is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

This command downloads version-specific offline vulnerability bundles. The system contacts Central to determine the version if one is not specified. If communication fails, the download defaults to the version embedded within **roxctl**.

By default, it will attempt to download the database for the determined version and less-specific variants. For example, if version **4.4.1-extra** is specified, downloads will be attempted for the following version variants:

- 4.4.1-extra
- 4.4.1
- 4.4

Usage

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

Table 8.72. Options

Option	Description
--force	Force overwriting the output file if it already exists. The default value is **false**.

Option	Description
--scanner-db-file string	Output file to save the vulnerability database to. The default value is the name and path of the remote file that is downloaded.
--skip-central	Do not contact Central when detecting the version. The default value is **false**.
--skip-variants	Do not attempt to process variants of the determined version. The default value is **false**.
-t, --timeout duration	Set the timeout for API requests representing the maximum duration of a request. The default value is **10m0s**.
--version string	Download a specific version or version variant of the vulnerability database. By default, the version is automatically detected.

8.12. ROXCTL SENSOR

Deploy Red Hat Advanced Cluster Security for Kubernetes (RHACS) services in secured clusters.

Usage

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

Table 8.73. Available commands

Command	Description
generate	Generate files to deploy RHACS services in secured clusters.
generate-certs	Download a YAML file with renewed certificates for Sensor, Collector, and Admission controller.
get-bundle	Download a bundle with the files to deploy RHACS services in a cluster.

Table 8.74. Options

Option	Description

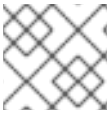
Option	Description
--retry-timeout duration	Set the timeout after which API requests are retried. A value of zero means that the entire request duration is waited for without retrying. The default value is **20s**.
-t, **--timeout duration**	Set the timeout for API requests representing the maximum duration of a request. The default value is **1m0s**.

8.12.1. roxctl sensor command options inherited from the parent command

The **roxctl sensor** command supports the following options inherited from the parent **roxctl** command:

Option	Description
--ca string	Specify a custom CA certificate file path for secure connections. Alternatively, you can specify the file path by using the **ROX_CA_CERT_FILE** environment variable.
--direct-grpc	Set **--direct-grpc** for improved connection performance. Alternatively, by setting the **ROX_DIRECT_GRPC_CLIENT** environment variable to **true**, you can enable direct gRPC . The default value is **false**.
-e, **--endpoint string**	Set the endpoint for the service to contact. Alternatively, you can set the endpoint by using the **ROX_ENDPOINT** environment variable. The default value is **localhost:8443**.
--force-http1	Force the use of HTTP/1 for all connections. Alternatively, by setting the **ROX_CLIENT_FORCE_HTTP1** environment variable to **true**, you can force the use of HTTP/1. The default value is **false**.
--insecure	Enable insecure connection options. Alternatively, by setting the **ROX_INSECURE_CLIENT** environment variable to **true**, you can enable insecure connection options. The default value is **false**.
--insecure-skip-tls-verify	Skip the TLS certificate validation. Alternatively, by setting the **ROX_INSECURE_CLIENT_SKIP_TLS_VERIFY** environment variable to **true**, you can skip the TLS certificate validation. The default value is **false**.

Option	Description
--no-color	Disable the color output. Alternatively, by setting the **ROX_NO_COLOR environment variable** to **true**, you can disable the color output. The default value is **false**.
-p, **--password string**	Specify the password for basic authentication. Alternatively, you can set the password by using the **ROX_ADMIN_PASSWORD** environment variable.
--plaintext	Use an unencrypted connection. Alternatively, by setting the **ROX_PLAINTEXT** environment variable to **true**, you can enable an unencrypted connection. The default value is **false**.
-s, **--server-name string**	Set the TLS server name to use for SNI. Alternatively, you can set the server name by using the **ROX_SERVER_NAME** environment variable.
--token-file string	Use the API token provided in the specified file for authentication. Alternatively, you can set the token by using the **ROX_API_TOKEN** environment variable.



NOTE

These options are applicable to all the sub-commands of the **roxctl sensor** command.

8.12.2. roxctl sensor generate

Generate files to deploy RHACS services in secured clusters.

Usage

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

Table 8.75. Options

Option	Description
--admission-controller-disable-bypass	Disable the bypass annotations for the admission controller. The default value is **false**.
--admission-controller-enforce-on-creates	Dynamic enable for enforcing on object creation in the admission controller. The default value is **false**.

Option	Description
--admission-controller-enforce-on-updates	Enable dynamic enforcement of object updates in the admission controller. The default value is **false**.
--admission-controller-listen-on-creates	Configure the admission controller webhook to listen to deployment creation. The default value is **false**.
--admission-controller-listen-on-updates	Configure the admission controller webhook to listen to deployment updates. The default value is **false**.
--admission-controller-scan-inline	Get scans inline when using the admission controller. The default value is **false**.
--admission-controller-timeout int32	Set the timeout in seconds for the admission controller. The default value is **3**.
--central string	Set the endpoint to which you want to connect Sensor. The default value is **central.stackrox:443**.
--collection-method collection method	Specify the collection method that you want to use for runtime support. Collection methods include **none**, **default**, **ebpf** and **core_bpf**. The default value is **default**.
--collector-image-repository string	Set the image repository that you want to use to deploy Collector. If not specified, a default value corresponding to the effective **--main-image repository** value is derived.
--continue-if-exists	Continue with downloading the sensor bundle even if the cluster already exists. The default value is **false**.

Option	Description
--create-upgrader-sa	Decide whether to create the upgrader service account with **cluster-admin** privileges to facilitate automated sensor upgrades. The default value is **true**.
--disable-tolerations	Disable tolerations for tainted nodes. The default value is **false**.
--enable-pod-security-policies	Create **PodSecurityPolicy** resources. The default value is **true**.
--istio-support string	Generate deployment files that support the specified Istio version. Valid versions include **1.0**, **1.1**, **1.2**, **1.3**, **1.4**, **1.5**, **1.6**, **1.7**.
--main-image-repository string	Specify the image repository that you want to use to deploy Sensor. If not specified, a default value is used.
--name string	Set the cluster name to identify the cluster.
--output-dir string	Set the output directory for the bundle contents. The default value is an automatically generated directory name inside the current directory.
--slim-collector string[="true"]	Use Collector-slim in the deployment bundle. Valid values include **auto**, **true**, and **false**. The default value is **auto**.
-t, **--timeout duration**	Set the timeout for API requests representing the maximum duration of a request. The default value is **5m0s**.

8.12.2.1. roxctl sensor generate k8s

Generate the required files to deploy RHACS services in a Kubernetes cluster.

Usage

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

Table 8.76. Options

Option	Description
--admission-controller-listen-on-events	Enable admission controller webhook to listen to Kubernetes events. The default value is **true**.

8.12.2.2. roxctl sensor generate openshift

Generate the required files to deploy RHACS services in a Red Hat OpenShift cluster.

Usage

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

Table 8.77. Options

Option	Description
`--admission-controller-listen-on-events false	true
auto[=true]`	**Enable or disable the admission controller webhook to listen to Kubernetes events**. The default value is **auto**.
`--disable-audit-logs false	true
auto[=true]`	Enable or disable audit log collection for runtime detection. The default value is **auto**.
--openshift-version int	Specify the Red Hat OpenShift major version for which you want to generate the deployment files.

8.12.3. roxctl sensor get-bundle

Download a bundle with the files to deploy RHACS services into a cluster.

Usage

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

1 For **<cluster_details>**, specify the cluster name or ID.

Table 8.78. Options

Option	Description
--create-upgrader-sa	Specify whether to create the upgrader service account with **cluster-admin** privileges for automated Sensor upgrades. The default value is **true**.
--istio-support string	Generate deployment files that support the specified Istio version. Valid versions include **1.0**, **1.1**, **1.2**, **1.3**, **1.4**, **1.5**, **1.6**, and **1.7**.
--output-dir string	Specify the output directory for the bundle contents. The default value is an automatically generated directory name inside the current directory.
--slim-collector string[="true"]	Use Collector-slim in the deployment bundle. Valid values include **auto**, **true** and **false**. The default value is **auto**.
-t, **--timeout duration**	Set the timeout for API requests representing the maximum duration of a request. The default value is **5m0s**.

8.12.4. roxctl sensor generate-certs

Download a YAML file with renewed certificates for Sensor, Collector, and Admission controller.

Usage

```
$ roxctl sensor generate-certs <cluster_details> [flags] 1
```

1 For **<cluster_details>**, specify the cluster name or ID.

Table 8.79. Options

Option	Description
--output-dir string	Specify the output directory for the YAML file. The default value is **.**.

8.13. ROXCTL VERSION

Display the current roxctl version.

Usage

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

8.13.1. roxctl version command options

The **roxctl version** command supports the following option:

Option	Description
--json	Display the extended version information as JSON. The default value is **false**.

8.13.2. roxctl version command options inherited from the parent command

The **roxctl version** command supports the following options inherited from the parent **roxctl** command:

Option	Description
--ca string	Specify a custom CA certificate file path for secure connections. Alternatively, you can specify the file path by using the **ROX_CA_CERT_FILE** environment variable.
--direct-grpc	Set **--direct-grpc** for improved connection performance. Alternatively, by setting the **ROX_DIRECT_GRPC_CLIENT** environment variable to **true**, you can enable direct gRPC . The default value is **false**.
-e, **--endpoint string**	Set the endpoint for the service to contact. Alternatively, you can set the endpoint by using the **ROX_ENDPOINT** environment variable. The default value is **localhost:8443**.
--force-http1	Force the use of HTTP/1 for all connections. Alternatively, by setting the **ROX_CLIENT_FORCE_HTTP1** environment variable to **true**, you can force the use of HTTP/1. The default value is **false**.
--insecure	Enable insecure connection options. Alternatively, by setting the **ROX_INSECURE_CLIENT** environment variable to **true**, you can enable insecure connection options. The default value is **false**.

Option	Description
--insecure-skip-tls-verify	Skip the TLS certificate validation. Alternatively, by setting the **ROX_INSECURE_CLIENT_SKIP_TLS_VERIFY** environment variable to **true**, you can skip the TLS certificate validation. The default value is **false**.
--no-color	Disable the color output. Alternatively, by setting the **ROX_NO_COLOR environment variable** to **true**, you can disable the color output. The default value is **false**.
-p, **--password string**	Specify the password for basic authentication. Alternatively, you can set the password by using the **ROX_ADMIN_PASSWORD** environment variable.
--plaintext	Use an unencrypted connection. Alternatively, by setting the **ROX_PLAINTEXT** environment variable to **true**, you can enable an unencrypted connection. The default value is **false**.
-s, **--server-name string**	Set the TLS server name to use for SNI. Alternatively, you can set the server name by using the **ROX_SERVER_NAME** environment variable.
--token-file string	Use the API token provided in the specified file for authentication. Alternatively, you can set the token by using the **ROX_API_TOKEN** environment variable.