Red Hat Advanced Cluster Security for Kubernetes 4.0

Installing

Installing Red Hat Advanced Cluster Security for Kubernetes

Last Updated: 2024-02-28

Red Hat Advanced Cluster Security for Kubernetes 4.0 Installing

Installing Red Hat Advanced Cluster Security for Kubernetes

Legal Notice

Abstract

This document describes how to install Red Hat Advanced Cluster Security for Kubernetes by using the Operator, Helm charts, or the roxctl CLI.

Table of Contents

CHAPTER 1. SUPPORTED PLATFORMS AND INSTALLATION METHODS

Red Hat Advanced Cluster Security for Kubernetes (RHACS) is supported on OpenShift Container Platform and Kubernetes platforms. For more information about supported self-managed and managed platforms, see Red Hat Advanced Cluster Security for Kubernetes Support Policy .

1.1. INSTALLATION METHODS FOR DIFFERENT PLATFORMS

You can perform different types of installations on different platforms.

NOTE

Not all installation methods are supported for all platforms.

Table 1.1. Supported platforms and recommended installation methods

Platform type [1]	Platform [2]	Supported for Central	Supported for Secured Clusters	Supported installation methods	Installation steps
Managed service platform	Red Hat OpenShift Dedicated (OSD)	Yes	Yes	Operator (recommended), Helm charts, or **roxctl** CLI [3]	● High-level overview of installing RHACS on Red Hat OpenShift
	Azure Red Hat OpenShift (ARO)	Yes	Yes		
	Red Hat OpenShift Service on AWS (ROSA)	Yes	Yes		
	Amazon Elastic Kubernetes Service (Amazon EKS)	Limited [4]	Yes	Helm charts (recommended), or **roxctl** CLI [3]	● High-level overview of installing RHACS on other platforms
	Google Kubernetes Engine (Google GKE)	Limited [4]	Yes		
	Microsoft Azure Kubernetes Service (Microsoft AKS)	Limited [4]	Yes		

Platform type [1]	Platform [2]	Supported for Central	Supported for Secured Clusters	Supported installation methods	Installation steps
Self-managed platform	Red Hat OpenShift Container Platform (OCP) 4.x	Yes	Yes	Operator (recommended), Helm charts, or **roxctl** CLI [3]	● High-level overview of installing RHACS on Red Hat OpenShift
	Red Hat OpenShift Kubernetes Engine (OKE) 4.x	No	Yes		

1. The availability of support for each platform depends on the overarching lifecycle of the platform and the end-of-life date.

2. For more information about supported self-managed and managed platforms, see Red Hat Advanced Cluster Security for Kubernetes Support Policy.

3. Do not use the **roxctl** installation method unless you have specific requirements for following this installation method.

4. RHACS Central is tested, qualified, and is fully supported exclusively on OpenShift Container Platform 4. Deployment and use of Central in environments that are not OpenShift Container Platform 4 is possible, but support is limited to the RHACS product software only and not to the underlying infrastructure provider. As part of problem diagnosis and isolation, it is necessary to reproduce problems in an OpenShift Container Platform 4 environment. If an issue is specific to a provider and cluster that is not OpenShift Container Platform 4, Red Hat provides commercially reasonable support to isolate issues. You are expected to open a case with your respective provider. For instructions, see the Red Hat 3rd Party Support Policy .

1.2. INSTALLATION METHODS FOR DIFFERENT ARCHITECTURES

Red Hat Advanced Cluster Security for Kubernetes (RHACS) supports the following architectures.

Table 1.2. Supported architectures and recommended installation methods

Supported architectures	Supported for Central	Supported for Secured Clusters	Supported installation methods
AMD64	Yes	Yes	Operator (preferred), Helm charts, or **roxctl** CLI (not recommended)
ppc64le (IBM Power)	No	Yes (OpenShift Container Platform version 4.12 and later)	Operator is the only supported install method.

Supported architectures	Supported for Central	Supported for Secured Clusters	Supported installation methods
s390x (IBM zSystems and IBM® LinuxONE)	No	Yes (OpenShift Container Platform versions 4.10, 4.12 and later)	

1.3. SUPPORTED BROWSERS FOR RHACS

Red Hat Advanced Cluster Security for Kubernetes (RHACS) browser support complies with Red Hat policy and includes the following browsers:

- Google Chrome
- Mozilla Firefox
- Apple Safari
- Microsoft Edge

CHAPTER 2. INSTALLING RHACS ON RED HAT OPENSHIFT

2.1. HIGH-LEVEL OVERVIEW OF INSTALLING RHACS ON RED HAT OPENSHIFT

Red Hat Advanced Cluster Security for Kubernetes (RHACS) provides security services for your self-managed Red Hat OpenShift Kubernetes systems.

Before you install:

- Understand the installation platforms and methods.
- Understand Red Hat Advanced Cluster Security for Kubernetes architecture .
- Review the prerequisites.

The following list provides a high-level overview of installation steps:

1. Install Central services on a cluster using the Operator, Helm charts, or the **roxctl** CLI.
2. Generate and apply an init bundle.
3. Install secured cluster resources on each of your secured clusters.

2.2. PREREQUISITES FOR RHACS ON RED HAT OPENSHIFT

Before you install RHACS for OpenShift Container Platform or another OCP-compatible supported Kubernetes platform, ensure you have met the prerequisites.

2.2.1. General requirements

RHACS has some system requirements that must be met before installing.

WARNING

You must not install Red Hat Advanced Cluster Security for Kubernetes on:

- Amazon Elastic File System (Amazon EFS). Use the Amazon Elastic Block Store (Amazon EBS) with the default **gp2** volume type instead.
- Older CPUs that do not have the Streaming SIMD Extensions (SSE) 4.2 instruction set. For example, Intel processors older than *Sandy Bridge* and AMD processors older than *Bulldozer*. (These processors were released in 2011.)

To install Red Hat Advanced Cluster Security for Kubernetes, you must have:

- OpenShift Container Platform version 4.10 or later. For more information about supported self-managed and managed OpenShift Container Platform versions, see Red Hat Advanced Cluster Security for Kubernetes Support Policy.
- Cluster nodes with a supported operating system:
    - Red Hat Enterprise Linux CoreOS (RHCOS), Red Hat Enterprise Linux (RHEL).
    - **Processor and memory**: 2 CPU cores and at least 3GiB of RAM.

        NOTE

        For deploying Central, use a machine type with four or more cores and apply scheduling policies to launch Central on such nodes.

    - **Architectures**: AMD64, ppc64le, or s390x.

        NOTE

        For ppc64le, or s390x architectures, you can only install RHACS Secured cluster services on IBM Power, IBM zSystems, and IBM® LinuxONE clusters. Central is not supported at this time.

- Persistent storage by using persistent volume claim (PVC).

    IMPORTANT

    You must not use Ceph FS storage with Red Hat Advanced Cluster Security for Kubernetes. Red Hat recommends using RBD block mode PVCs for Red Hat Advanced Cluster Security for Kubernetes.

    - Use Solid-State Drives (SSDs) for best performance. However, you can use another storage type if you do not have SSDs available.

To install using Helm charts:

- You must have Helm command-line interface (CLI) v3.2 or newer, if you are installing or configuring Red Hat Advanced Cluster Security for Kubernetes using Helm charts. Use the **helm version** command to verify the version of Helm you have installed.
- The Red Hat OpenShift CLI (**oc**).
- You must have access to the Red Hat Container Registry. For information about downloading images from **registry.redhat.io**, see Red Hat Container Registry Authentication .

2.2.2. Prerequisites for installing Central

A containerized service called Central handles API interactions and user interface (Portal) access while a containerized service called Central DB (PostgreSQL 13) handles data persistence.

Both Central and Central DB require persistent storage:

- You can provide storage with a persistent volume claim (PVC).

NOTE

You can use a hostPath volume for storage only if all your hosts (or a group of hosts) mount a shared file system, such as an NFS share or a storage appliance. Otherwise, your data is only saved on a single node. Red Hat does not recommend using a hostPath volume.

- Use Solid-State Drives (SSD) for best performance. However, you can use another storage type if you do not have SSDs available.
- If you use a web proxy or firewall, you must configure bypass rules to allow traffic for the **definitions.stackrox.io** and **collector-modules.stackrox.io** domains and enable Red Hat Advanced Cluster Security for Kubernetes to trust your web proxy or firewall. Otherwise, updates for vulnerability definitions and kernel support packages will fail.
  Red Hat Advanced Cluster Security for Kubernetes requires access to:
  - **definitions.stackrox.io** for downloading updated vulnerability definitions. Vulnerability definition updates allow Red Hat Advanced Cluster Security for Kubernetes to maintain up-to-date vulnerability data when new vulnerabilities are discovered or additional data sources are added.
  - **collector-modules.stackrox.io** to download updated kernel support packages. Updated Kernel support packages ensure that Red Hat Advanced Cluster Security for Kubernetes can monitor the latest operating systems and collect data about the network traffic and processes running inside the containers. Without these updates, Red Hat Advanced Cluster Security for Kubernetes might fail to monitor containers if you add new nodes in your cluster or if you update your nodes' operating system.

NOTE

For security reasons, you should deploy Central in a cluster with limited administrative access.

Memory and storage requirements

The following table lists the minimum memory and storage values required to install and run Central.

Central	CPU	Memory	Storage
Request	1.5 cores	4 GiB	100 GiB
Limit	4 cores	8 GiB	100 GiB

Central DB	CPU	Memory	Storage
Request	4 cores	8 GiB	100 GiB
Limit	8 cores	16 GiB	100 GiB

Sizing guidelines

Use the following compute resources and storage values depending upon the number of nodes in your cluster.

Nodes	Deployments	Central CPU	Central Memory	Central Storage
Up to 100	Up to 1000	2 cores	4 GiB	100 GiB
Up to 500	Up to 2000	4 cores	8 GiB	100 GiB
More than 500	More than 2000	8 cores	12 - 16 GiB	100 - 200 GiB

Nodes	Deployments	Central DB CPU	Central DB Memory	Central DB Storage
Up to 100	Up to 1000	2 cores	4 GiB	100 GiB
Up to 500	Up to 2000	4 cores	8 GiB	100 GiB
More than 500	More than 2000	8 cores	12 - 16 GiB	100 - 200 GiB

2.2.3. Prerequisites for installing Scanner

Red Hat Advanced Cluster Security for Kubernetes includes an image vulnerability scanner called Scanner. This service scans images that are not already scanned by scanners integrated into image registries.

Memory and storage requirements

Scanner	CPU	Memory
Request	1.2 cores	2700 MiB
Limit	5 cores	8000 MiB

2.2.4. Prerequisites for installing Sensor

Sensor monitors your Kubernetes and OpenShift Container Platform clusters. These services currently deploy in a single deployment, which handles interactions with the Kubernetes API and coordinates with Collector.

Memory and storage requirements

Sensor	CPU	Memory
Request	2 cores	4 GiB
Limit	4 cores	8 GiB

2.2.5. Prerequisites for installing Admission controller

The Admission controller prevents users from creating workloads that violate policies you configure.

Memory and storage requirements

By default, the admission control service runs 3 replicas. The following table lists the request and limits for each replica.

Admission controller	CPU	Memory
Request	.05 cores	100 MiB
Limit	.5 cores	500 MiB

2.2.6. Prerequisites for installing Collector

Collector monitors runtime activity on each node in your secured clusters. It connects to Sensor to report this information.

CAUTION

To install Collector on systems that have Unified Extensible Firmware Interface (UEFI) and that have Secure Boot enabled, you must use eBPF probes because kernel modules are unsigned, and the UEFI firmware cannot load unsigned packages. Collector identifies Secure Boot status at the start and switches to eBPF probes if required.

Memory and storage requirements

Collector	CPU	Memory
Request	.05 cores	320 MiB
Limit	.75 cores	1 GiB

NOTE

Collector uses a mutable image tag (**<version>-latest**), so you get support for newer Linux kernel versions more easily. There is no change in code, pre-existing kernel modules, or eBPF programs for image updates. Updates only add a single image layer with support for new kernel versions published after the initial release.

2.3. INSTALLING CENTRAL SERVICES FOR RHACS ON RED HAT OPENSHIFT

Central is the resource that contains the RHACS application management interface and services. It handles data persistence, API interactions, and RHACS portal access. You can use the same Central instance to secure multiple OpenShift Container Platform or Kubernetes clusters.

You can install Central on your OpenShift Container Platform or Kubernetes cluster by using one of the following methods:

- Install using the Operator
- Install using Helm charts

- Install using the **roxctl** CLI (do not use this method unless you have a specific installation need that requires using it)

2.3.1. Install Central using the Operator

2.3.1.1. Installing the Red Hat Advanced Cluster Security for Kubernetes Operator

Using the OperatorHub provided with OpenShift Container Platform is the easiest way to install Red Hat Advanced Cluster Security for Kubernetes.

Prerequisites

- You have access to an OpenShift Container Platform cluster using an account with Operator installation permissions.

- You must be using OpenShift Container Platform 4.10 or later. For more information, see Red Hat Advanced Cluster Security for Kubernetes Support Policy.

Procedure

1. Navigate in the web console to the **Operators → OperatorHub** page.

2. If Red Hat Advanced Cluster Security for Kubernetes is not displayed, enter **Advanced Cluster Security** into the **Filter by keyword** box to find the Red Hat Advanced Cluster Security for Kubernetes Operator.

3. Select the **Red Hat Advanced Cluster Security for Kubernetes Operator**to view the details page.

4. Read the information about the Operator, and then click **Install**.

5. On the **Install Operator** page:

    - Keep the default value for **Installation mode** as **All namespaces on the cluster**.

    - Choose a specific namespace in which to install the Operator for the **Installed namespace** field. Install the Red Hat Advanced Cluster Security for Kubernetes Operator in the **rhacs-operator** namespace.

    - Select automatic or manual updates for **Update approval**.
      If you choose automatic updates, when a new version of the Operator is available, Operator Lifecycle Manager (OLM) automatically upgrades the running instance of your Operator.

      If you choose manual updates, when a newer version of the Operator is available, OLM creates an update request. As a cluster administrator, you must manually approve the update request to update the Operator to the latest version.

      > **IMPORTANT**
      >
      > If you choose manual updates, you must update the RHACS Operator in all secured clusters when you update the RHACS Operator in the cluster where Central is installed. The secured clusters and the cluster where Central is installed must have the same version to ensure optimal functionality.

6. Click **Install**.

Verification

- After the installation completes, navigate to **Operators → Installed Operators** to verify that the Red Hat Advanced Cluster Security for Kubernetes Operator is listed with the status of **Succeeded**.

Next Step

- Install, configure, and deploy the **Central** custom resource.

2.3.1.2. Installing Central using the Operator method

The main component of Red Hat Advanced Cluster Security for Kubernetes is called Central. You can install Central on OpenShift Container Platform by using the **Central** custom resource. You deploy Central only once, and you can monitor multiple separate clusters by using the same Central installation.

IMPORTANT

When you install Red Hat Advanced Cluster Security for Kubernetes for the first time, you must first install the **Central** custom resource because the **SecuredCluster** custom resource installation is dependent on certificates that Central generates.

Prerequisites

- You must be using OpenShift Container Platform 4.10 or later. For more information, see Red Hat Advanced Cluster Security for Kubernetes Support Policy.

Procedure

1. On the OpenShift Container Platform web console, navigate to the **Operators → Installed Operators** page.

2. Select the Red Hat Advanced Cluster Security for Kubernetes Operator from the list of installed Operators.

3. If you have installed the Operator in the recommended namespace, OpenShift Container Platform lists the project as **rhacs-operator**. Select **Project: rhacs-operator → Create project**.

WARNING

- If you have installed the Operator in a different namespace, OpenShift Container Platform shows the name of that namespace rather than **rhacs-operator**.

- You must install the Red Hat Advanced Cluster Security for Kubernetes **Central** custom resource in its own project and not in the **rhacs-operator** and **openshift-operator** projects, or in the project in which you have installed the Red Hat Advanced Cluster Security for Kubernetes Operator.

4. Enter the new project name (for example, **stackrox**), and click **Create**. Red Hat recommends that you use **stackrox** as the project name.
5. Under the **Provided APIs** section, select **Central**. Click **Create Central**.
6. Enter a name for your **Central** custom resource and add any labels you want to apply. Otherwise, accept the default values for the available options.
7. Click **Create**.

NOTE

If you are using the cluster-wide proxy, Red Hat Advanced Cluster Security for Kubernetes uses that proxy configuration to connect to the external services.

Next Steps

1. Verify Central installation.
2. Optional: Configure Central options.
3. Generate an init bundle containing the cluster secrets that allows communication between the **Central** and **SecuredCluster** resources. You need to download this bundle, use it to generate resources on the clusters you want to secure, and securely store it.
4. Install secured cluster services on each cluster you want to monitor.

2.3.1.3. Installing Central with an external database using the Operator method

IMPORTANT

External PostgreSQL support 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 .

The main component of Red Hat Advanced Cluster Security for Kubernetes is called Central. You can install Central on OpenShift Container Platform by using the **Central** custom resource. You deploy Central only once, and you can monitor multiple separate clusters by using the same Central installation.

IMPORTANT

When you install Red Hat Advanced Cluster Security for Kubernetes for the first time, you must first install the **Central** custom resource because the **SecuredCluster** custom resource installation is dependent on certificates that Central generates.

Prerequisites

- You must be using OpenShift Container Platform 4.10 or later. For more information, see Red Hat Advanced Cluster Security for Kubernetes Support Policy.

- You must provision a database that supports PostgreSQL 13 and you must only use it for RHACS.
- You must have a superuser role with permissions to create and delete databases.

NOTE

RHACS 4.0 does not support a multi-tenant database and PgBouncer.

Procedure

1. On the OpenShift Container Platform web console, navigate to the **Operators** → **Installed Operators** page.
2. Select the Red Hat Advanced Cluster Security for Kubernetes Operator from the list of installed Operators.
3. If you have installed the Operator in the recommended namespace, OpenShift Container Platform lists the project as **rhacs-operator**. Select **Project: rhacs-operator** → **Create project**.

WARNING

- If you have installed the Operator in a different namespace, OpenShift Container Platform shows the name of that namespace rather than **rhacs-operator**.
- You must install the Red Hat Advanced Cluster Security for Kubernetes **Central** custom resource in its own project and not in the **rhacs-operator** and **openshift-operator** projects, or in the project in which you have installed the Red Hat Advanced Cluster Security for Kubernetes Operator.

4. Enter the new project name (for example, **stackrox**), and click **Create**. Red Hat recommends that you use **stackrox** as the project name.
5. Create a password secret in the deployed namespace by using the OpenShift Container Platform web console or the terminal.
   - On the OpenShift Container Platform web console, go to the **Workloads** → **Secrets** page. Create a **Key/Value secret** with the key **password** and the value as the path of a plain text file containing the password for the superuser of the provisioned database.
   - Or, run the following command in your terminal:

```
$ oc create secret generic external-db-password \ 1
  --from-file=password=<password.txt> 2
```

1 If you use Kubernetes, enter **kubectl** instead of **oc**.

2 Replace **password.txt** with the path of the file which has the plain text password.

6. Return to the Red Hat Advanced Cluster Security for Kubernetes operator page in the OpenShift Container Platform web console. Under the **Provided APIs** section, select **Central**. Click **Create Central**.

7. Enter a name for your **Central** custom resource and add any labels you want to apply.

8. Navigate to **Central Component Settings** → **Central DB Settings**.

9. For **Administrator Password** specify the referenced secret as **external-db-password** (or the secret name of the password created previously).

10. For **Connection String (Technology Preview)** specify the connection string in **keyword=value** format, for example, **host=<host> port=5432 user=postgres sslmode=verify-ca**

11. For **Persistence** → **PersistentVolumeClaim** → **Claim Name**, remove **central-db**.

12. If necessary, you can specify a Certificate Authority for Central to trust the database certificate. To add this, go to the YAML view and add a TLS block under the top-level spec, as shown in the following example:

```
spec:
  tls:
    additionalCAs:
    - name: db-ca
      content: |
        <certificate>
```

13. Click **Create**.

NOTE

If you are using the cluster-wide proxy, Red Hat Advanced Cluster Security for Kubernetes uses that proxy configuration to connect to the external services.

Next Steps

1. Verify Central installation.

2. Optional: Configure Central options.

3. Generate an init bundle containing the cluster secrets that allows communication between the **Central** and **SecuredCluster** resources. You need to download this bundle, use it to generate resources on the clusters you want to secure, and securely store it.

4. Install secured cluster services on each cluster you want to monitor.

Additional resources

- Central configuration options
- PostgreSQL Connection String Docs

2.3.1.4. Verifying Central installation using the Operator method

After Central finishes installing, log in to the RHACS portal to verify the successful installation of Central.

Procedure

1. On the OpenShift Container Platform web console, navigate to the **Operators** → **Installed Operators** page.
2. Select the Red Hat Advanced Cluster Security for Kubernetes Operator from the list of installed Operators.
3. Select the **Central** tab.
4. From the **Centrals** list, select **stackrox-central-services** to view its details.
5. To get the password for the **admin** user, you can either:
   - Click the link under **Admin Password Secret Reference**.
   - Use the Red Hat OpenShift CLI to enter the command listed under **Admin Credentials Info**:

     ```
     $ oc -n stackrox get secret central-htpasswd -o go-template='{{index .data "password" |
     base64decode}}'
     ```

6. Find the link to the RHACS portal by using the Red Hat OpenShift CLI command:

   ```
   $ oc -n stackrox get route central -o jsonpath="{.status.ingress[0].host}"
   ```

   Alternatively, you can use the Red Hat Advanced Cluster Security for Kubernetes web console to find the link to the RHACS portal by performing the following commands:

   a. Navigate to **Networking** → **Routes**.

   b. Find the **central** Route and click on the RHACS portal link under the **Location** column.

7. Log in to the RHACS portal using the username **admin** and the password that you retrieved in a previous step. Until RHACS is completely configured (for example, you have the **Central** resource and at least one **SecuredCluster** resource installed and configured), no data is available in the dashboard. The **SecuredCluster** resource can be installed and configured on the same cluster as the **Central** resource. Clusters with the **SecuredCluster** resource are similar to managed clusters in Red Hat Advanced Cluster Management (RHACM).

Next Steps

1. Optional: Configure central settings.
2. Generate an init bundle containing the cluster secrets that allows communication between the **Central** and **SecuredCluster** resources. You need to download this bundle, use it to generate resources on the clusters you want to secure, and securely store it.
3. Install secured cluster services on each cluster you want to monitor.

2.3.2. Install Central using Helm charts

You can install Central using Helm charts without any customization, using the default values, or by using Helm charts with additional customizations of configuration parameters.

2.3.2.1. Install Central using Helm charts without customization

You can install RHACS on your cluster without any customizations. You must add the Helm chart repository and install the **central-services** Helm chart to install the centralized components of Central and Scanner.

2.3.2.1.1. Adding the Helm chart repository

Procedure

- Add the RHACS charts repository.

```
$ helm repo add rhacs https://mirror.openshift.com/pub/rhacs/charts/
```

The Helm repository for Red Hat Advanced Cluster Security for Kubernetes includes Helm charts for installing different components, including:

- Central services Helm chart (**central-services**) for installing the centralized components (Central and Scanner).

  NOTE

  You deploy centralized components only once and you can monitor multiple separate clusters by using the same installation.

- Secured Cluster Services Helm chart (**secured-cluster-services**) for installing the per-cluster (Sensor and Admission controller) and per-node (Collector) components.

  NOTE

  Deploy the per-cluster components into each cluster that you want to monitor and deploy the per-node components in all nodes that you want to monitor.

Verification

- Run the following command to verify the added chart repository:

```
$ helm search repo -l rhacs/
```

2.3.2.1.2. Installing the central-services Helm chart without customizations

Use the following instructions to install the **central-services** Helm chart to deploy the centralized components (Central and Scanner).

Prerequisites

- You must have access to the Red Hat Container Registry. For information about downloading images from **registry.redhat.io**, see Red Hat Container Registry Authentication .

Procedure

- Run the following command to install Central services and expose Central using a route:

```
$ helm install -n stackrox \
  --create-namespace stackrox-central-services rhacs/central-services \
  --set imagePullSecrets.username=<username> \
  --set imagePullSecrets.password=<password> \
  --set central.exposure.route.enabled=true
```

- Or, run the following command to install Central services and expose Central using a load balancer:

```
$ helm install -n stackrox \
  --create-namespace stackrox-central-services rhacs/central-services \
  --set imagePullSecrets.username=<username> \
  --set imagePullSecrets.password=<password> \
  --set central.exposure.loadBalancer.enabled=true
```

- Or, run the following command to install Central services and expose Central using port forward:

```
$ helm install -n stackrox \
  --create-namespace stackrox-central-services rhacs/central-services \
    --set imagePullSecrets.username=<username> \
  --set imagePullSecrets.password=<password>
```

IMPORTANT

- If you are installing Red Hat Advanced Cluster Security for Kubernetes in a cluster that requires a proxy to connect to external services, you must specify your proxy configuration by using the **proxyConfig** parameter. For example:

```
env:
  proxyConfig: |
    url: http://proxy.name:port
    username: username
    password: password
    excludes:
    - some.domain
```

- If you already created one or more image pull secrets in the namespace in which you are installing, instead of using a username and password, you can use **--set imagePullSecrets.useExisting="<pull-secret-1;pull-secret-2>"**.

- Do not use image pull secrets:
  - If you are pulling your images from **quay.io/stackrox-io** or a registry in a private network that does not require authentication. Use use **--set imagePullSecrets.allowNone=true** instead of specifying a username and password.
  - If you already configured image pull secrets in the default service account in the namespace you are installing. Use **--set imagePullSecrets.useFromDefaultServiceAccount=true** instead of specifying a username and password.

The output of the installation command includes:

- An automatically generated administrator password.
- Instructions on storing all the configuration values.
- Any warnings that Helm generates.

2.3.2.2. Install Central using Helm charts with customizations

You can install RHACS on your Red Hat OpenShift cluster with customizations by using Helm chart configuration parameters with the **helm install** and **helm upgrade** commands. You can specify these parameters by using the **--set** option or by creating YAML configuration files.

Create the following files for configuring the Helm chart for installing Red Hat Advanced Cluster Security for Kubernetes:

- Public configuration file **values-public.yaml**: Use this file to save all non-sensitive configuration options.
- Private configuration file **values-private.yaml**: Use this file to save all sensitive configuration options. Ensure that you store this file securely.

2.3.2.2.1. Private configuration file

This section lists the configurable parameters of the **values-private.yaml** file. There are no default values for these parameters.

2.3.2.2.1.1. Image pull secrets

The credentials that are required for pulling images from the registry depend on the following factors:

- If you are using a custom registry, you must specify these parameters:
  - **imagePullSecrets.username**
  - **imagePullSecrets.password**
  - **image.registry**
- If you do not use a username and password to log in to the custom registry, you must specify one of the following parameters:
  - **imagePullSecrets.allowNone**
  - **imagePullSecrets.useExisting**
  - **imagePullSecrets.useFromDefaultServiceAccount**

Parameter	Description
imagePullSecrets.username	The username of the account that is used to log in to the registry.

Parameter	Description
imagePullSecrets.password	The password of the account that is used to log in to the registry.
imagePullSecrets.allowNone	Use **true** if you are using a custom registry and it allows pulling images without credentials.
imagePullSecrets.useExisting	A comma-separated list of secrets as values. For example, **secret1, secret2, secretN**. Use this option if you have already created pre-existing image pull secrets with the given name in the target namespace.
imagePullSecrets.useFromDefaultServiceAccount	Use **true** if you have already configured the default service account in the target namespace with sufficiently scoped image pull secrets.

2.3.2.2.1.2. Proxy configuration

If you are installing Red Hat Advanced Cluster Security for Kubernetes in a cluster that requires a proxy to connect to external services, you must specify your proxy configuration by using the **proxyConfig** parameter. For example:

```
env:
  proxyConfig: |
    url: http://proxy.name:port
    username: username
    password: password
    excludes:
    - some.domain
```

Parameter	Description
env.proxyConfig	Your proxy configuration.

2.3.2.2.1.3. Central

Configurable parameters for Central.

For a new installation, you can skip the following parameters:

- **central.jwtSigner.key**
- **central.serviceTLS.cert**
- **central.serviceTLS.key**
- **central.adminPassword.value**
- **central.adminPassword.htpasswd**

- **central.db.serviceTLS.cert**
- **central.db.serviceTLS.key**
- **central.db.password.value**
- When you do not specify values for these parameters the Helm chart autogenerates values for them.
- If you want to modify these values you can use the **helm upgrade** command and specify the values using the **--set** option.

IMPORTANT

For setting the administrator password, you can only use either **central.adminPassword.value** or **central.adminPassword.htpasswd**, but not both.

Parameter	Description
central.jwtSigner.key	A private key which Red Hat Advanced Cluster Security for Kubernetes should use for signing JSON web tokens (JWTs) for authentication.
central.serviceTLS.cert	An internal certificate that the Central service should use for deploying Central.
central.serviceTLS.key	The private key of the internal certificate that the Central service should use.
central.defaultTLS.cert	The user-facing certificate that Central should use. Red Hat Advanced Cluster Security for Kubernetes uses this certificate for RHACS portal. ● For a new installation, you must provide a certificate, otherwise, Red Hat Advanced Cluster Security for Kubernetes installs Central by using a self-signed certificate. ● If you are upgrading, Red Hat Advanced Cluster Security for Kubernetes uses the existing certificate and its key.
central.defaultTLS.key	The private key of the user-facing certificate that Central should use. ● For a new installation, you must provide the private key, otherwise, Red Hat Advanced Cluster Security for Kubernetes installs Central by using a self-signed certificate. ● If you are upgrading, Red Hat Advanced Cluster Security for Kubernetes uses the existing certificate and its key.

Parameter	Description
central.db.password.value	Connection password for Central database.
central.adminPassword.value	Administrator password for logging into Red Hat Advanced Cluster Security for Kubernetes.
central.adminPassword.htpasswd	Administrator password for logging into Red Hat Advanced Cluster Security for Kubernetes. This password is stored in hashed format using bcrypt.
central.db.serviceTLS.cert	An internal certificate that the Central DB service should use for deploying Central DB.
central.db.serviceTLS.key	The private key of the internal certificate that the Central DB service should use.
central.db.password.value	The password used to connect to the Central DB.

NOTE

If you are using **central.adminPassword.htpasswd** parameter, you must use a bcrypt encoded password hash. You can run the command **htpasswd -nB admin** to generate a password hash. For example,

```
htpasswd: |
  admin:<bcrypt-hash>
```

2.3.2.2.1.4. Scanner

Configurable parameters for Scanner.

For a new installation, you can skip the following parameters and the Helm chart autogenerates values for them. Otherwise, if you are upgrading to a new version, specify the values for the following parameters:

- **scanner.dbPassword.value**
- **scanner.serviceTLS.cert**
- **scanner.serviceTLS.key**
- **scanner.dbServiceTLS.cert**
- **scanner.dbServiceTLS.key**

Parameter	Description

Parameter	Description
scanner.dbPassword.value	The password to use for authentication with Scanner database. Do not modify this parameter because Red Hat Advanced Cluster Security for Kubernetes automatically creates and uses its value internally.
scanner.serviceTLS.cert	An internal certificate that the Scanner service should use for deploying Scanner.
scanner.serviceTLS.key	The private key of the internal certificate that the Scanner service should use.
scanner.dbServiceTLS.cert	An internal certificate that the Scanner-db service should use for deploying Scanner database.
scanner.dbServiceTLS.key	The private key of the internal certificate that the Scanner-db service should use.

2.3.2.2.2. Public configuration file

This section lists the configurable parameters of the **values-public.yaml** file.

2.3.2.2.2.1. Image pull secrets

Image pull secrets are the credentials required for pulling images from your registry.

Parameter	Description
imagePullSecrets.allowNone	Use **true** if you are using a custom registry and it allows pulling images without credentials.
imagePullSecrets.useExisting	A comma-seprated list of secrets as values. For example, **secret1, secret2**. Use this option if you have already created pre-existing image pull secrets with the given name in the target namespace.
imagePullSecrets.useFromDefaultServiceAccount	Use **true** if you have already configured the default service account in the target namespace with sufficiently scoped image pull secrets.

2.3.2.2.2.2. Image

Image declares the configuration to set up the main registry, which the Helm chart uses to resolve images for the **central.image**, **scanner.image**, and **scanner.dbImage** parameters.

Parameter	Description
image.registry	Address of your image registry. Either use a hostname, such as **registry.redhat.io**, or a remote registry hostname, such as **us.gcr.io/stackrox-mirror**.

2.3.2.2.2.3. Environment variables

Red Hat Advanced Cluster Security for Kubernetes automatically detects your cluster environment and sets values for **env.openshift**, **env.istio**, and **env.platform**. Only set these values to override the automatic cluster environment detection.

Parameter	Description
env.openshift	Use **true** for installing on an OpenShift Container Platform cluster and overriding automatic cluster environment detection.
env.istio	Use **true** for installing on an Istio enabled cluster and overriding automatic cluster environment detection.
env.platform	The platform on which you are installing Red Hat Advanced Cluster Security for Kubernetes. Set its value to **default** or **gke** to specify cluster platform and override automatic cluster environment detection.
env.offlineMode	Use **true** to use Red Hat Advanced Cluster Security for Kubernetes in offline mode.

2.3.2.2.2.4. Additional trusted certificate authorities

The Red Hat Advanced Cluster Security for Kubernetes automatically references the system root certificates to trust. When Central or Scanner must reach out to services that use certificates issued by an authority in your organization or a globally trusted partner organization, you can add trust for these services by specifying the root certificate authority to trust by using the following parameter:

Parameter	Description
additionalCAs.<certificate_name>	Specify the PEM encoded certificate of the root certificate authority to trust.

2.3.2.2.2.5. Central

Configurable parameters for Central.

- You must specify a persistent storage option as either **hostPath** or **persistentVolumeClaim**.
- For exposing Central deployment for external access. You must specify one parameter, either **central.exposure.loadBalancer**, **central.exposure.nodePort**, or **central.exposure.route**. When you do not specify any value for these parameters, you must manually expose Central or access it by using port-forwarding.

The following table includes settings for an external PostgreSQL database (Technology Preview).

IMPORTANT

External PostgreSQL support 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 .

Parameter	Description
central.endpointsConfig	The endpoint configuration options for Central.
central.nodeSelector	If the node selector selects tainted nodes, use this parameter to specify a taint toleration key, value, and effect for Central. This parameter is mainly used for infrastructure nodes.
central.tolerations	If the node selector selects tainted nodes, use this parameter to specify a taint toleration key, value, and effect for Central. This parameter is mainly used for infrastructure nodes.
central.exposeMonitoring	Specify **true** to expose Prometheus metrics endpoint for Central on port number **9090**.
central.image.registry	A custom registry that overrides the global **image.registry** parameter for the Central image.
central.image.name	The custom image name that overrides the default Central image name (**main**).
central.image.tag	The custom image tag that overrides the default tag for Central image. If you specify your own image tag during a new installation, you must manually increment this tag when you to upgrade to a new version by running the **helm upgrade** command. If you mirror Central images in your own registry, do not modify the original image tags.

Parameter	Description
central.image.fullRef	Full reference including registry address, image name, and image tag for the Central image. Setting a value for this parameter overrides the **central.image.registry**, **central.image.name**, and **central.image.tag** parameters.
central.resources.requests.memory	The memory request for Central to override the default value.
central.resources.requests.cpu	The CPU request for Central to override the default value.
central.resources.limits.memory	The memory limit for Central to override the default value.
central.resources.limits.cpu	The CPU limit for Central to override the default value.
central.persistence.hostPath	The path on the node where RHACS should create a database volume. Red Hat does not recommend using this option.
central.persistence.persistentVolumeClaim.claimName	The name of the persistent volume claim (PVC) you are using.
central.persistence.persistentVolumeClaim.createClaim	Use **true** to create a new PVC, or**false** to use an existing claim.
central.persistence.persistentVolumeClaim.size	The size (in GiB) of the persistent volume managed by the specified claim.
central.exposure.loadBalancer.enabled	Use **true** to expose Central by using a load balancer.
central.exposure.loadBalancer.port	The port number on which to expose Central. The default port number is 443.
central.exposure.nodePort.enabled	Use **true** to expose Central by using the node port service.
central.exposure.nodePort.port	The port number on which to expose Central. When you skip this parameter, OpenShift Container Platform automatically assigns a port number. Red Hat recommends that you do not specify a port number if you are exposing Red Hat Advanced Cluster Security for Kubernetes by using a node port.

Parameter	Description
central.exposure.route.enabled	Use **true** to expose Central by using a route. This parameter is only available for OpenShift Container Platform clusters.
central.db.external	(Technology Preview) Use **true** to specify that Central DB should not be deployed and that an external database will be used.
central.db.source.connectionString	(Technology Preview) The connection string for Central to use to connect to the database. This is only used when **central.db.external** is set to true. The connection string must be in keyword/value format as described in the PostgreSQL documentation in "Additional resources". ● Only PostgreSQL 13 is supported. ● Connections through PgBouncer are not supported. ● User must be superuser with ability to create and delete databases.
central.db.source.minConns	The minimum number of connections to the database to be established.
central.db.source.maxConns	The maximum number of connections to the database to be established.
central.db.source.statementTimeoutMs	The number of milliseconds a single query or transaction can be active against the database.
central.db.postgresConfig	The postgresql.conf to be used for Central DB as described in the PostgreSQL documentation in "Additional resources".
central.db.hbaConfig	The pg_hba.conf to be used for Central DB as described in the PostgreSQL documentation in "Additional resources".
central.db.nodeSelector	Specify a node selector label as **label-key: label-value** to force Central DB to only schedule on nodes with the specified label.
central.db.image.registry	A custom registry that overrides the global **image.registry** parameter for the Central DB image.

Parameter	Description
central.db.image.name	The custom image name that overrides the default Central DB image name (**central-db**).
central.db.image.tag	The custom image tag that overrides the default tag for Central DB image. If you specify your own image tag during a new installation, you must manually increment this tag when you to upgrade to a new version by running the **helm upgrade** command. If you mirror Central DB images in your own registry, do not modify the original image tags.
central.db.image.fullRef	Full reference including registry address, image name, and image tag for the Central DB image. Setting a value for this parameter overrides the **central.db.image.registry**, **central.db.image.name**, and **central.db.image.tag** parameters.
central.db.resources.requests.memory	The memory request for Central DB to override the default value.
central.db.resources.requests.cpu	The CPU request for Central DB to override the default value.
central.db.resources.limits.memory	The memory limit for Central DB to override the default value.
central.db.resources.limits.cpu	The CPU limit for Central DB to override the default value.
central.db.persistence.hostPath	The path on the node where RHACS should create a database volume. Red Hat does not recommend using this option.
central.db.persistence.persistentVolumeClaim.claimName	The name of the persistent volume claim (PVC) you are using.
central.db.persistence.persistentVolumeClaim.createClaim	Use **true** to create a new persistent volume claim, or **false** to use an existing claim.
central.db.persistence.persistentVolumeClaim.size	The size (in GiB) of the persistent volume managed by the specified claim.

2.3.2.2.2.6. Scanner

Configurable parameters for Scanner.

Parameter	Description
scanner.disable	Use **true** to install Red Hat Advanced Cluster Security for Kubernetes without Scanner. When you use it with the **helm upgrade** command, Helm removes existing Scanner deployment.
scanner.exposeMonitoring	Specify **true** to expose Prometheus metrics endpoint for Scanner on port number **9090**.
scanner.replicas	The number of replicas to create for the Scanner deployment. When you use it with the **scanner.autoscaling** parameter, this value sets the initial number of replicas.
scanner.logLevel	Configure the log level for Scanner. Red Hat recommends that you not change the log level's default value (**INFO**).
scanner.nodeSelector	Specify a node selector label as **label-key: label-value** to force Scanner to only schedule on nodes with the specified label.
scanner.tolerations	If the node selector selects tainted nodes, use this parameter to specify a taint toleration key, value, and effect for Scanner. This parameter is mainly used for infrastructure nodes.
scanner.autoscaling.disable	Use **true** to disable autoscaling for Scanner deployment. When you disable autoscaling, the **minReplicas** and **maxReplicas** parameters do not have any effect.
scanner.autoscaling.minReplicas	The minimum number of replicas for autoscaling.
scanner.autoscaling.maxReplicas	The maximum number of replicas for autoscaling.
scanner.resources.requests.memory	The memory request for Scanner to override the default value.
scanner.resources.requests.cpu	The CPU request for Scanner to override the default value.
scanner.resources.limits.memory	The memory limit for Scanner to override the default value.
scanner.resources.limits.cpu	The CPU limit for Scanner to override the default value.

Parameter	Description
scanner.dbResources.requests.memory	The memory request for Scanner database deployment to override the default values.
scanner.dbResources.requests.cpu	The CPU request for Scanner database deployment to override the default values.
scanner.dbResources.limits.memory	The memory limit for Scanner database deployment to override the default values.
scanner.dbResources.limits.cpu	The CPU limit for Scanner database deployment to override the default values.
scanner.image.registry	A custom registry for the Scanner image.
scanner.image.name	The custom image name that overrides the default Scanner image name (**scanner**).
scanner.dbImage.registry	A custom registry for the Scanner DB image.
scanner.dbImage.name	The custom image name that overrides the default Scanner DB image name (**scanner-db**).
scanner.dbNodeSelector	Specify a node selector label as **label-key: label-value** to force Scanner DB to only schedule on nodes with the specified label.
scanner.dbTolerations	If the node selector selects tainted nodes, use this parameter to specify a taint toleration key, value, and effect for Scanner DB. This parameter is mainly used for infrastructure nodes.

2.3.2.2.2.7. Customization

Use these parameters to specify additional attributes for all objects that Red Hat Advanced Cluster Security for Kubernetes creates.

Parameter	Description
customize.labels	A custom label to attach to all objects.
customize.annotations	A custom annotation to attach to all objects.
customize.podLabels	A custom label to attach to all deployments.
customize.podAnnotations	A custom annotation to attach to all deployments.

Parameter	Description
customize.envVars	A custom environment variable for all containers in all objects.
customize.central.labels	A custom label to attach to all objects that Central creates.
customize.central.annotations	A custom annotation to attach to all objects that Central creates.
customize.central.podLabels	A custom label to attach to all Central deployments.
customize.central.podAnnotations	A custom annotation to attach to all Central deployments.
customize.central.envVars	A custom environment variable for all Central containers.
customize.scanner.labels	A custom label to attach to all objects that Scanner creates.
customize.scanner.annotations	A custom annotation to attach to all objects that Scanner creates.
customize.scanner.podLabels	A custom label to attach to all Scanner deployments.
customize.scanner.podAnnotations	A custom annotation to attach to all Scanner deployments.
customize.scanner.envVars	A custom environment variable for all Scanner containers.
customize.scanner-db.labels	A custom label to attach to all objects that Scanner DB creates.
customize.scanner-db.annotations	A custom annotation to attach to all objects that Scanner DB creates.
customize.scanner-db.podLabels	A custom label to attach to all Scanner DB deployments.
customize.scanner-db.podAnnotations	A custom annotation to attach to all Scanner DB deployments.
customize.scanner-db.envVars	A custom environment variable for all Scanner DB containers.

You can also use:

- the **customize.other.service/*.labels** and the **customize.other.service/*.annotations** parameters, to specify labels and annotations for all objects.
- or, provide a specific service name, for example, **customize.other.service/central-loadbalancer.labels** and **customize.other.service/central-loadbalancer.annotations** as parameters and set their value.

2.3.2.2.2.8. Advanced customization

IMPORTANT

The parameters specified in this section are for information only. Red Hat does not support Red Hat Advanced Cluster Security for Kubernetes instances with modified namespace and release names.

Parameter	Description
allowNonstandardNamespace	Use **true** to deploy Red Hat Advanced Cluster Security for Kubernetes into a namespace other than the default namespace **stackrox**.
allowNonstandardReleaseName	Use **true** to deploy Red Hat Advanced Cluster Security for Kubernetes with a release name other than the default **stackrox-central-services**.

2.3.2.2.3. Installing the central-services Helm chart

After you configure the **values-public.yaml** and **values-private.yaml** files, install the **central-services** Helm chart to deploy the centralized components (Central and Scanner).

Procedure

- Run the following command:

```
$ helm install -n stackrox --create-namespace \
  stackrox-central-services rhacs/central-services \
  -f <path_to_values_public.yaml> -f <path_to_values_private.yaml> 1
```

1 Use the **-f** option to specify the paths for your YAML configuration files.

2.3.2.3. Changing configuration options after deploying the central-services Helm chart

You can make changes to any configuration options after you have deployed the **central-services** Helm chart.

Procedure

1. Update the **values-public.yaml** and **values-private.yaml** configuration files with new values.

2. Run the **helm upgrade** command and specify the configuration files using the **-f** option:

```
$ helm upgrade -n stackrox \
  stackrox-central-services rhacs/central-services \
  -f <path_to_values_public.yaml> \
  -f <path_to_values_private.yaml>
```

NOTE

You can also specify configuration values using the **--set** or **--set-file** parameters. However, these options are not saved, and it requires you to manually specify all the options again whenever you make changes.

2.3.3. Install Central using the roxctl CLI

WARNING

For production environments, Red Hat recommends using the Operator or Helm charts to install RHACS. Do not use the **roxctl** install method unless you have a specific installation need that requires using this method.

2.3.3.1. Installing the roxctl CLI

To install Red Hat Advanced Cluster Security for Kubernetes you must install the **roxctl** CLI by downloading the binary. You can install **roxctl** on Linux, Windows, or macOS.

2.3.3.1.1. Installing the roxctl CLI on Linux

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

Procedure

1. Download the latest version of the **roxctl** CLI:

```
$ curl -O https://mirror.openshift.com/pub/rhacs/assets/4.0.5/bin/Linux/roxctl
```

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

```
$ chmod +x roxctl
```

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

```
$ echo $PATH
```

Verification

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

```
$ roxctl version
```

2.3.3.1.2. Installing the roxctl CLI on macOS

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

Procedure

1. Download the latest version of the **roxctl** CLI:

```
$ curl -O https://mirror.openshift.com/pub/rhacs/assets/4.0.5/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
```

2.3.3.1.3. Installing the roxctl CLI on Windows

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

Procedure

- Download the latest version of the **roxctl** CLI:

```
$ curl -O https://mirror.openshift.com/pub/rhacs/assets/4.0.5/bin/Windows/roxctl.exe
```

Verification

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

```
$ roxctl version
```

2.3.3.2. Using the interactive installer

Use the interactive installer to generate the required secrets, deployment configurations, and deployment scripts for your environment.

Procedure

1. Run the interactive install command:

```
$ roxctl central generate interactive
```

> **IMPORTANT**
>
> Installing Red Hat Advanced Cluster Security for Kubernetes using **roxctl** CLI creates PodSecurityPolicy (PSP) objects by default for backward compatibility. If you install RHACS on Kubernetes versions 1.25 and newer or OpenShift Container Platform version 4.12 and newer, you must disable the PSP object creation. To do this, specify **--enable-pod-security-policies** option as **false** for the **roxctl central generate** and **roxctl sensor generate** commands.

2. Press **Enter** to accept the default value for a prompt or enter custom values as required.

```
Enter path to the backup bundle from which to restore keys and certificates (optional):
Enter PEM cert bundle file (optional): 1
Enter administrator password (default: autogenerated):
Enter orchestrator (k8s, openshift): openshift
Enter the directory to output the deployment bundle to (default: "central-bundle"):
Enter the OpenShift major version (3 or 4) to deploy on (default: "0"): 4
Enter Istio version when deploying into an Istio-enabled cluster (leave empty when not
running Istio) (optional):
Enter the method of exposing Central (route, lb, np, none) (default: "none"): route 2
Enter main image to use (default: "stackrox.io/main:3.0.61.1"):
Enter whether to run StackRox in offline mode, which avoids reaching out to the Internet
(default: "false"):
Enter whether to enable telemetry (default: "true"):
Enter the deployment tool to use (kubectl, helm, helm-values) (default: "kubectl"):
Enter Scanner DB image to use (default: "stackrox.io/scanner-db:2.15.2"):
Enter Scanner image to use (default: "stackrox.io/scanner:2.15.2"):
Enter Central volume type (hostpath, pvc): pvc 3
Enter external volume name (default: "stackrox-db"):
Enter external volume size in Gi (default: "100"):
Enter storage class name (optional if you have a default StorageClass configured):
```

1 If you want to add a custom TLS certificate, provide the file path for the PEM-encoded certificate. When you specify a custom certificate the interactive installer also prompts you to provide a PEM private key for the custom certificate you are using.

2 To use the RHACS portal, you must expose Central by using a route, a load balancer or a node port.

3 If you plan to install Red Hat Advanced Cluster Security for Kubernetes on OpenShift Container Platform with a hostPath volume, you must modify the SELinux policy.

WARNING

On OpenShift Container Platform, for using a hostPath volume, you must modify the SELinux policy to allow access to the directory, which the host and the container share. It is because SELinux blocks directory sharing by default. To modify the SELinux policy, run the following command:

```
$ sudo chcon -Rt svirt_sandbox_file_t <full_volume_path>
```

However, Red Hat does not recommend modifying the SELinux policy, instead use PVC when installing on OpenShift Container Platform.

On completion, the installer creates a folder named central-bundle, which contains the necessary YAML manifests and scripts to deploy Central. In addition, it shows on-screen instructions for the scripts you need to run to deploy additional trusted certificate authorities, Central and Scanner, and the authentication instructions for logging into the RHACS portal along with the autogenerated password if you did not provide one when answering the prompts.

2.3.3.3. Running the Central installation scripts

After you run the interactive installer, you can run the **setup.sh** script to install Central.

Procedure

1. Run the **setup.sh** script to configure image registry access:

```
$ ./central-bundle/central/scripts/setup.sh
```

2. Create the necessary resources:

```
$ oc create -R -f central-bundle/central
```

3. Check the deployment progress:

```
$ oc get pod -n stackrox -w
```

4. After Central is running, find the RHACS portal IP address and open it in your browser. Depending on the exposure method you selected when answering the prompts, use one of the following methods to get the IP address.

Exposure method	Command	Address	Example
Route	**oc -n stackrox get route central**	The address under the **HOST/PORT** column in the output	**https://central-stackrox.example.route**

Exposure method	Command	Address	Example
Node Port	**oc get node -owide && oc -n stackrox get svc central-loadbalancer**	IP or hostname of any node, on the port shown for the service	**https://198.51.100.0:31489**
Load Balancer	**oc -n stackrox get svc central-loadbalancer**	EXTERNAL-IP or hostname shown for the service, on port 443	**https://192.0.2.0**
None	**central-bundle/central/scripts/port-forward.sh 8443**	**https://localhost:8443**	**https://localhost:8443**

NOTE

If you have selected autogenerated password during the interactive install, you can run the following command to see it for logging into Central:

```
$ cat central-bundle/password
```

2.4. OPTIONAL - CONFIGURING CENTRAL CONFIGURATION OPTIONS FOR RHACS USING THE OPERATOR

This topic provides information about optional configuration options that you can configure using the Operator.

2.4.1. Central configuration options using the Operator

When you create a Central instance, the Operator lists the following configuration options for the **Central** custom resource.

The following table includes settings for an external PostgreSQL database (Technology Preview).

IMPORTANT

External PostgreSQL support 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 .

2.4.1.1. Central settings

Parameter	Description
central.adminPasswordSecret	Specify a secret that contains the administrator password in the **password** data item. If omitted, the operator autogenerates a password and stores it in the **password** item in the **central-htpasswd** secret.
central.defaultTLSSecret	By default, Central only serves an internal TLS certificate, which means that you need to handle TLS termination at the ingress or load balancer level. If you want to terminate TLS in Central and serve a custom server certificate, you can specify a secret containing the certificate and private key.
central.adminPasswordGenerationDisabled	Set this parameter to **true** to disable the automatic administrator password generation. Use this only after you perform the first-time setup of alternative authentication methods. Do not use this for initial installation. Otherwise, you must reinstall the custom resource to log back in.
central.tolerations	If the node selector selects tainted nodes, use this parameter to specify a taint toleration key, value, and effect for Central. This parameter is mainly used for infrastructure nodes.
central.exposure.loadBalancer.enabled	Set this to **true** to expose Central through a load balancer.
central.exposure.loadBalancer.port	Use this parameter to specify a custom port for your load balancer.
central.exposure.loadBalancer.ip	Use this parameter to specify a static IP address reserved for your load balancer.
central.exposure.route.enabled	Set this to **true** to expose Central through an OpenShift route. The default value is **false**.
central.exposure.route.host	Specify a custom hostname to use for Central's route. Leave this unset to accept the default value that OpenShift Container Platform provides.
central.exposure.nodeport.enabled	Set this to **true** to expose Central through a node port. The default value is**false**.
central.exposure.nodeport.port	Use this to specify an explicit node port.
central.monitoring.exposeEndpoint	Use **Enabled** to enable monitoring for Central. When you enable monitoring, RHACS creates a new monitoring service on port number **9090**. The default value is **Disabled**.
central.nodeSelector	If you want this component to only run on specific nodes, you can configure a node selector by using this parameter.

Parameter	Description
central.persistence.hostPath.path	Specify a host path to store persistent data in a directory on the host. Red Hat does not recommend using this. If you need to use host path, you must use it with a node selector.
central.persistence.persistentVolumeClaim.claimName	The name of the PVC to manage persistent data. If no PVC with the given name exists, it will be created. The default value is **stackrox-db** if not set. To prevent data losses the PVC is not removed automatically with Central`s deletion.
central.persistence.persistentVolumeClaim.size	The size of the persistent volume when created through the claim. This is automatically generated by default.
central.persistence.persistentVolumeClaim.storageClassName	The name of the storage class to use for the PVC. If your cluster is not configured with a default storage class, you must provide a value for this parameter.
central.resources.limits	Use this parameter to override the default resource limits for the Central.
central.resources.requests	Use this parameter to override the default resource requests for the Central.
central.imagePullSecrets	Use this parameter to specify the image pull secrets for the Central image.
central.db.passwordSecret.name	Specify a secret that has the database password in the **password** data item. Only use this parameter if you want to specify a connection string manually. If omitted, the operator auto-generates a password and stores it in the **password** item in the **central-db-password** secret.
central.db.connectionString	(Technology Preview): Setting this parameter will not deploy Central DB, and Central will connect using the specified connection string. If you specify a value for this parameter, you must also specify a value for **central.db.passwordSecret.name**. This parameter has the following constraints: ● Connection string must be in keyword/value format as described in the PostgreSQL documentation. For more information, see the links in the **Additional resources** section. ● Only PostgreSQL 13 is supported. ● Connections through PGBouncer are not supported. ● User must be a superuser who can create and delete databases.

Parameter	Description
central.db.tolerations	If the node selector selects tainted nodes, use this parameter to specify a taint toleration key, value, and effect for Central DB. This parameter is mainly used for infrastructure nodes.
central.db.persistence.hostPath.path	Specify a host path to store persistent data in a directory on the host. Red Hat does not recommend using this. If you need to use host path, you must use it with a node selector.
central.db.persistence.persistentVolumeClaim.claimName	The name of the PVC to manage persistent data. If no PVC with the given name exists, it will be created. The default value is **central-db** if not set. To prevent data loss, the PVC is not removed automatically with Central DB’s deletion.
central.db.persistence.persistentVolumeClaim.size	The size of the persistent volume when created through the claim. This is automatically generated by default.
central.db.persistence.persistentVolumeClaim.storageClassName	The name of the storage class to use for the PVC. If your cluster is not configured with a default storage class, you must provide a value for this parameter.
central.db.resources.limits	Use this parameter to override the default resource limits for the Central DB.
central.db.resources.requests	Use this parameter to override the default resource requests for the Central DB.

2.4.1.2. Scanner settings

Parameter	Description
scanner.analyzer.nodeSelector	If you want this scanner to only run on specific nodes, you can configure a node selector by using this parameter.
scanner.analyzer.tolerations	If the node selector selects tainted nodes, use this parameter to specify a taint toleration key, value, and effect for Scanner. This parameter is mainly used for infrastructure nodes.
scanner.analyzer.resources.limits	Use this parameter to override the default resource limits for the scanner.
scanner.analyzer.resources.requests	Use this parameter to override the default resource requests for the scanner.
scanner.analyzer.scaling.autoScaling	When enabled, the number of analyzer replicas is managed dynamically based on the load, within the limits specified.

Parameter	Description
scanner.analyzer.scaling.maxReplicas	Specifies the maximum replicas to be used the analyzer autoscaling configuration
scanner.analyzer.scaling.minReplicas	Specifies the minimum replicas to be used the analyzer autoscaling configuration
scanner.analyzer.scaling.replicas	When autoscaling is disabled, the number of replicas will always be configured to match this value.
scanner.db.nodeSelector	If you want this component to only run on specific nodes, you can configure a node selector by using this parameter.
scanner.db.tolerations	If the node selector selects tainted nodes, use this parameter to specify a taint toleration key, value, and effect for Scanner DB. This parameter is mainly used for infrastructure nodes.
scanner.db.resources.limits	Use this parameter to override the default resource limits for the scanner.
scanner.db.resources.requests	Use this parameter to override the default resource requests for the scanner.
scanner.monitoring.exposeEndpoint	Use **Enabled** to enable monitoring for Scanner. When you enable monitoring, RHACS creates a new monitoring service on port number **9090**. The default value is **Disabled**.
scanner.scannerComponent	If you do not want to deploy Scanner, you can disable it by using this parameter. If you disable Scanner, all other settings in this section have no effect. Red Hat does not recommend disabling Red Hat Advanced Cluster Security for Kubernetes Scanner.

2.4.1.3. General and miscellaneous settings

Parameter	Description
tls.additionalCAs	Additional Trusted CA certificates for the secured cluster to trust. These certificates are typically used when integrating with services using a private certificate authority.
misc.createSCCs	Specify **true** to create **SecurityContextConstraints** (SCCs) for Central. Setting to **true** might cause issues in some environments.
customize.annotations	Allows specifying custom annotations for the Central deployment.
customize.envVars	Advanced settings to configure environment variables.

Parameter	Description
egress.connectivityPolicy	Configures whether RHACS should run in online or offline mode. In offline mode, automatic updates of vulnerability definitions and kernel modules are disabled.

Additional resources

- Connection Strings - PostgreSQL Docs
- Parameter Interaction via the Configuration File - PostgreSQL Docs
- The pg_hba.conf File - PostgreSQL Docs

2.5. GENERATING AND APPLYING AN INIT BUNDLE FOR RHACS ON RED HAT OPENSHIFT

Before you install the **SecuredCluster** resource on a cluster, you must create an init bundle. The cluster that has **SecuredCluster** installed and configured then uses this bundle to authenticate with Central. You can create an init bundle by using either the RHACS portal or the **roxctl** CLI. You then apply the init bundle by using it to create resources.

To configure an init bundle for RHACS Cloud Service, see the following resources:

- Generating an init bundle for secured clusters (Red Hat Cloud)
- Applying an init bundle for secured clusters (Red Hat Cloud)
- Generating an init bundle for Kubernetes secured clusters
- Applying an init bundle for Kubernetes secured clusters

NOTE

You must have the **Admin** user role to create an init bundle.

2.5.1. Generating an init bundle

2.5.1.1. Generating an init bundle by using the RHACS portal

You can create an init bundle containing secrets by using the RHACS portal.

NOTE

You must have the **Admin** user role to create an init bundle.

Procedure

1. Find the address of the RHACS portal based on your exposure method:

   a. For a route:

```
$ oc get route central -n stackrox
```

   b. For a load balancer:

      ```
      $ oc get service central-loadbalancer -n stackrox
      ```

   c. For port forward:

      i. Run the following command:

         ```
         $ oc port-forward svc/central 18443:443 -n stackrox
         ```

      ii. Navigate to **https://localhost:18443/**.

2. On the RHACS portal, navigate to **Platform Configuration → Integrations**.

3. Navigate to the **Authentication Tokens** section and click on **Cluster Init Bundle**.

4. Click **Generate bundle**.

5. Enter a name for the cluster init bundle and click **Generate**.

   a. If you are installing using Helm charts, click **Download Helm Values File** to download the generated bundle.

   b. If you are installing using the Operator, click **Download Kubernetes Secret File** to download the generated bundle.

IMPORTANT

Store this bundle securely because it contains secrets. You can use the same bundle to create multiple secured clusters.

Next steps

1. Apply the init bundle by creating a resource on the secured cluster.

2. Install secured cluster services on each cluster.

2.5.1.2. Generating an init bundle by using the roxctl CLI

You can create an init bundle with secrets by using the **roxctl** CLI.

NOTE

You must have the **Admin** user role to create init bundles.

Prerequisites

You have configured the **ROX_API_TOKEN** and the **ROX_CENTRAL_ADDRESS** environment variables.

- Set the **ROX_API_TOKEN** and the **ROX_CENTRAL_ADDRESS** environment variables:

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

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

Procedure

- Run the following command to generate a cluster init bundle containing secrets:
  For Helm installations:

  ```
  $ roxctl -e "$ROX_CENTRAL_ADDRESS" \
    central init-bundles generate <cluster_init_bundle_name> \
    --output cluster_init_bundle.yaml
  ```

  For Operator installations:

  ```
  $ roxctl -e "$ROX_CENTRAL_ADDRESS" \
    central init-bundles generate <cluster_init_bundle_name> \
    --output-secrets cluster_init_bundle.yaml
  ```

  > **IMPORTANT**
  >
  > Ensure that you store this bundle securely because it contains secrets. You can use the same bundle to set up multiple secured clusters.

Next Step

- Use the Red Hat OpenShift CLI to create resources using the init bundle.

2.5.1.3. Creating resources by using the init bundle

Before you install secured clusters, you must use the init bundle to create the required resources on the cluster that will allow the services on the secured clusters to communicate with Central.

> **NOTE**
>
> If you are installing by using Helm charts, do not perform this step. Complete the installation by using Helm; See "Installing RHACS on secured clusters by using Helm charts" in the additional resources section.

Prerequisites

- You must have generated an init bundle containing secrets.

Procedure

To create resources, perform one of the following steps:

- In the OpenShift Container Platform web console, in the top menu, click **+** to open the **Import YAML** page. You can drag the init bundle file or copy and paste its contents into the editor, and then click **Create**.

- Using the Red Hat OpenShift CLI, run the following command to create the resources:

  ```
  $ oc create -f <init_bundle>.yaml \ 1
    -n <stackrox> 2
  ```

1 Specify the file name of the init bundle containing the secrets.

2 Specify the name of the project where Central services are installed.

Next Step

- Install RHACS secured cluster services in all clusters that you want to monitor.

Additional resources

- Installing RHACS on secured clusters by using Helm charts

2.6. INSTALLING SECURED CLUSTER SERVICES FOR RHACS ON RED HAT OPENSHIFT

This section describes the installation procedure for installing Red Hat Advanced Cluster Security for Kubernetes on your secured clusters.

You can install RHACS on your secured clusters by using one of the following methods:

- Install using the Operator
- Install using Helm charts
- Install using the **roxctl** CLI (do not use this method unless you have a specific installation need that requires using it)

2.6.1. Installing RHACS on secured clusters by using the Operator

2.6.1.1. Installing secured cluster services

You can install secured cluster services on your clusters by using the **SecuredCluster** custom resource. You must install the secured cluster services on every cluster in your environment that you want to monitor.

CAUTION

When you install secured cluster services, Collector is also installed. To install Collector on systems that have Unified Extensible Firmware Interface (UEFI) and that have Secure Boot enabled, you must use eBPF probes because kernel modules are unsigned, and the UEFI firmware cannot load unsigned packages. Collector identifies Secure Boot status at the start and switches to eBPF probes if required.

Prerequisites

- If you are using OpenShift Container Platform, you must install version 4.10 or later.
- You have installed the RHACS Operator.
- You have generated an init bundle and applied it to the cluster.

Procedure

1. On the OpenShift Container Platform web console, navigate to the **Operators** → **Installed Operators** page.
2. Click the RHACS Operator.
3. Click **Secured Cluster** from the central navigation menu in the **Operator details** page.
4. Click **Create SecuredCluster**.
5. Select one of the following options in the **Configure via** field:
   - **Form view**: Use this option if you want to use the on-screen fields to configure the secured cluster and do not need to change any other fields.
   - **YAML view**: Use this view to set up the secured cluster using the YAML file. The YAML file is displayed in the window and you can edit fields in it. If you select this option, when you are finished editing the file, click **Create**.
6. If you are using **Form view**, enter the new project name by accepting or editing the default name. The default value is **stackrox-secured-cluster-services**.
7. Optional: Add any labels for the cluster.
8. Enter a unique name for your **SecuredCluster** custom resource.
9. For **Central Endpoint**, enter the address and port number of your Central instance. For example, if Central is available at **https://central.example.com**, then specify the central endpoint as **central.example.com:443**. The default value of **central.stackrox.svc:443** only works when you install secured cluster services and Central in the same cluster. Do not use the default value when you are configuring multiple clusters. Instead, use the hostname when configuring the **Central Endpoint** value for each cluster.
   - Only if you are installing secured cluster services and Central in the same cluster, use **central.stackrox.svc:443**.
10. Accept the default values or configure custom values if needed. For example, you may need to configure TLS if you are using custom certificates or untrusted CAs.
11. Click **Create**.

Next step

1. Optional: Configure additional secured cluster settings.
2. Verify installation.

2.6.2. Installing RHACS on secured clusters by using Helm charts

You can install RHACS on secured clusters by using Helm charts with no customization, using the default values, or with customizations of configuration parameters.

2.6.2.1. Installing RHACS on secured clusters by using Helm charts without customizations

2.6.2.1.1. Adding the Helm chart repository

Procedure

- Add the RHACS charts repository.

```
$ helm repo add rhacs https://mirror.openshift.com/pub/rhacs/charts/
```

The Helm repository for Red Hat Advanced Cluster Security for Kubernetes includes Helm charts for installing different components, including:

- Central services Helm chart (**central-services**) for installing the centralized components (Central and Scanner).

  NOTE

  You deploy centralized components only once and you can monitor multiple separate clusters by using the same installation.

- Secured Cluster Services Helm chart (**secured-cluster-services**) for installing the per-cluster (Sensor and Admission controller) and per-node (Collector) components.

  NOTE

  Deploy the per-cluster components into each cluster that you want to monitor and deploy the per-node components in all nodes that you want to monitor.

Verification

- Run the following command to verify the added chart repository:

```
$ helm search repo -l rhacs/
```

2.6.2.1.2. Installing the secured-cluster-services Helm chart without customization

Use the following instructions to install the **secured-cluster-services** Helm chart to deploy the per-cluster and per-node components (Sensor, Admission controller, and Collector).

CAUTION

To install Collector on systems that have Unified Extensible Firmware Interface (UEFI) and that have Secure Boot enabled, you must use eBPF probes because kernel modules are unsigned, and the UEFI firmware cannot load unsigned packages. Collector identifies Secure Boot status at the start and switches to eBPF probes if required.

Prerequisites

- You must have generated RHACS init bundle for your cluster.
- You must have the address and the port number that you are exposing the Central service on.

Procedure

- Run the following command on your Kubernetes based clusters:

```
$ helm install -n stackrox --create-namespace \
    stackrox-secured-cluster-services rhacs/secured-cluster-services \
  -f <path_to_cluster_init_bundle.yaml> \ ❶
  --set clusterName=<name_of_the_secured_cluster> \
  --set centralEndpoint=<endpoint_of_central_service> ❷
```

1. Use the **-f** option to specify the path for the init bundle.

2. Specify the address and port number for Central. For example, **acs.domain.com:443**.

- Run the following command on OpenShift Container Platform clusters:

```
$ helm install -n stackrox --create-namespace \
  stackrox-secured-cluster-services rhacs/secured-cluster-services \
  -f <path_to_cluster_init_bundle.yaml> \ ❶
  --set clusterName=<name_of_the_secured_cluster> \
  --set centralEndpoint=<endpoint_of_central_service> ❷
  --set scanner.disable=false
```

1. Use the **-f** option to specify the path for the init bundle.

2. Specify the address and port number for Central. For example, **acs.domain.com:443**.

Additional resources

- Generating and applying an init bundle for RHACS on Red Hat OpenShift

2.6.2.2. Configuring the secured-cluster-services Helm chart with customizations

This section describes Helm chart configuration parameters that you can use with the **helm install** and **helm upgrade** commands. You can specify these parameters by using the **--set** option or by creating YAML configuration files.

Create the following files for configuring the Helm chart for installing Red Hat Advanced Cluster Security for Kubernetes:

- Public configuration file **values-public.yaml**: Use this file to save all non-sensitive configuration options.

- Private configuration file **values-private.yaml**: Use this file to save all sensitive configuration options. Ensure that you store this file securely.

IMPORTANT

While using the **secured-cluster-services** Helm chart, do not modify the **values.yaml** file that is part of the chart.

2.6.2.2.1. Configuration parameters

Parameter	Description
clusterName	Name of your cluster.

Parameter	Description
centralEndpoint	Address, including port number, of the Central endpoint. If you are using a non-gRPC capable load balancer, use the WebSocket protocol by prefixing the endpoint address with **wss://**. When configuring multiple clusters, use the hostname for the address (for example, **central.example.com:443**).
sensor.endpoint	Address of the Sensor endpoint including port number.
sensor.imagePullPolicy	Image pull policy for the Sensor container.
sensor.serviceTLS.cert	The internal service-to-service TLS certificate that Sensor uses.
sensor.serviceTLS.key	The internal service-to-service TLS certificate key that Sensor uses.
sensor.resources.requests.memory	The memory request for the Sensor container. Use this parameter to override the default value.
sensor.resources.requests.cpu	The CPU request for the Sensor container. Use this parameter to override the default value.
sensor.resources.limits.memory	The memory limit for the Sensor container. Use this parameter to override the default value.
sensor.resources.limits.cpu	The CPU limit for the Sensor container. Use this parameter to override the default value.
sensor.nodeSelector	Specify a node selector label as **label-key: label-value** to force Sensor to only schedule on nodes with the specified label.
sensor.tolerations	If the node selector selects tainted nodes, use this parameter to specify a taint toleration key, value, and effect for Sensor. This parameter is mainly used for infrastructure nodes.
image.main.name	The name of the **main** image.
image.collector.name	The name of the Collector image.
image.main.registry	Address of the registry you are using for the main image.

Parameter	Description
image.collector.registry	Address of the registry you are using for the Collector image.
image.main.pullPolicy	Image pull policy for **main** images.
image.collector.pullPolicy	Image pull policy for the Collector images.
image.main.tag	Tag of **main** image to use.
image.collector.tag	Tag of **collector** image to use.
collector.collectionMethod	Either **EBPF**, **KERNEL_MODULE**, or **NO_COLLECTION**.
collector.imagePullPolicy	Image pull policy for the Collector container.
collector.complianceImagePullPolicy	Image pull policy for the Compliance container.
collector.disableTaintTolerations	If you specify **false**, tolerations are applied to Collector, and the collector pods can schedule onto all nodes with taints. If you specify it as **true**, no tolerations are applied, and the collector pods are not scheduled onto nodes with taints.
collector.resources.requests.memory	The memory request for the Collector container. Use this parameter to override the default value.
collector.resources.requests.cpu	The CPU request for the Collector container. Use this parameter to override the default value.
collector.resources.limits.memory	The memory limit for the Collector container. Use this parameter to override the default value.
collector.resources.limits.cpu	The CPU limit for the Collector container. Use this parameter to override the default value.
collector.complianceResources.requests.memory	The memory request for the Compliance container. Use this parameter to override the default value.
collector.complianceResources.requests.cpu	The CPU request for the Compliance container. Use this parameter to override the default value.
collector.complianceResources.limits.memory	The memory limit for the Compliance container. Use this parameter to override the default value.
collector.complianceResources.limits.cpu	The CPU limit for the Compliance container. Use this parameter to override the default value.

Parameter	Description
collector.serviceTLS.cert	The internal service-to-service TLS certificate that Collector uses.
collector.serviceTLS.key	The internal service-to-service TLS certificate key that Collector uses.
admissionControl.listenOnCreates	This setting controls whether Kubernetes is configured to contact Red Hat Advanced Cluster Security for Kubernetes with **AdmissionReview** requests for workload creation events.
admissionControl.listenOnUpdates	When you set this parameter as **false**, Red Hat Advanced Cluster Security for Kubernetes creates the **ValidatingWebhookConfiguration** in a way that causes the Kubernetes API server not to send object update events. Since the volume of object updates is usually higher than the object creates, leaving this as **false** limits the load on the admission control service and decreases the chances of a malfunctioning admission control service.
admissionControl.listenOnEvents	This setting controls whether the cluster is configured to contact Red Hat Advanced Cluster Security for Kubernetes with **AdmissionReview** requests for Kubernetes **exec** and **portforward** events. Red Hat Advanced Cluster Security for Kubernetes does not support this feature on OpenShift Container Platform 3.11. For more information, see Red Hat Advanced Cluster Security for Kubernetes Support Policy.
admissionControl.dynamic.enforceOnCreates	This setting controls whether Red Hat Advanced Cluster Security for Kubernetes evaluates policies; if it is disabled, all AdmissionReview requests are automatically accepted.
admissionControl.dynamic.enforceOnUpdates	This setting controls the behavior of the admission control service. You must specify **listenOnUpdates** as **true** for this to work.

Parameter	Description
admissionControl.dynamic.scanInline	If you set this option to **true**, the admission control service requests an image scan before making an admission decision. Since image scans take several seconds, enable this option only if you can ensure that all images used in your cluster are scanned before deployment (for example, by a CI integration during image build). This option corresponds to the **Contact image scanners** option in the RHACS Portal.
admissionControl.dynamic.disableBypass	Set it to **true** to disable bypassing the Admission controller.
admissionControl.dynamic.timeout	The maximum time, in seconds, Red Hat Advanced Cluster Security for Kubernetes should wait while evaluating admission review requests. Use this to set request timeouts when you enable image scanning. If the image scan runs longer than the specified time, Red Hat Advanced Cluster Security for Kubernetes accepts the request.
admissionControl.resources.requests.memory	The memory request for the Admission Control container. Use this parameter to override the default value.
admissionControl.resources.requests.cpu	The CPU request for the Admission Control container. Use this parameter to override the default value.
admissionControl.resources.limits.memory	The memory limit for the Admission Control container. Use this parameter to override the default value.
admissionControl.resources.limits.cpu	The CPU limit for the Admission Control container. Use this parameter to override the default value.
admissionControl.nodeSelector	Specify a node selector label as **label-key: label-value** to force Admission Control to only schedule on nodes with the specified label.
admissionControl.tolerations	If the node selector selects tainted nodes, use this parameter to specify a taint toleration key, value, and effect for Admission Control. This parameter is mainly used for infrastructure nodes.
admissionControl.serviceTLS.cert	The internal service-to-service TLS certificate that Admission Control uses.

Parameter	Description
admissionControl.serviceTLS.key	The internal service-to-service TLS certificate key that Admission Control uses.
registryOverride	Use this parameter to override the default **docker.io** registry. Specify the name of your registry if you are using some other registry.
collector.disableTaintTolerations	If you specify **false**, tolerations are applied to Collector, and the Collector pods can schedule onto all nodes with taints. If you specify it as **true**, no tolerations are applied, and the Collector pods are not scheduled onto nodes with taints.
createUpgraderServiceAccount	Specify **true** to create the **sensor-upgrader** account. By default, Red Hat Advanced Cluster Security for Kubernetes creates a service account called **sensor-upgrader** in each secured cluster. This account is highly privileged but is only used during upgrades. If you do not create this account, you must complete future upgrades manually if the Sensor does not have enough permissions.
createSecrets	Specify **false** to skip the orchestrator secret creation for the Sensor, Collector, and Admission controller.
collector.slimMode	Specify **true** if you want to use a slim Collector image for deploying Collector. Using slim Collector images requires Central to provide the matching eBPF probe or kernel module. If you are running Red Hat Advanced Cluster Security for Kubernetes in offline mode, you must download a kernel support package from stackrox.io and upload it to Central for slim Collectors to function. Otherwise, you must ensure that Central can access the online probe repository hosted at https://collector-modules.stackrox.io/.
sensor.resources	Resource specification for Sensor.
admissionControl.resources	Resource specification for Admission controller.
collector.resources	Resource specification for Collector.
collector.complianceResources	Resource specification for Collector’s Compliance container.

Parameter	Description
exposeMonitoring	If you set this option to **true**, Red Hat Advanced Cluster Security for Kubernetes exposes Prometheus metrics endpoints on port number 9090 for the Sensor, Collector, and the Admission controller.
auditLogs.disableCollection	If you set this option to **true**, Red Hat Advanced Cluster Security for Kubernetes disables the audit log detection features used to detect access and modifications to configuration maps and secrets.
scanner.disable	If you set this option to **false**, Red Hat Advanced Cluster Security for Kubernetes deploys a lightweight scanner and Scanner DB in the secured cluster to allow scanning images on OpenShift Container Registry. Enabling Scanner is only supported on OpenShift. Defaults to **true**
scanner.dbTolerations	If the node selector selects tainted nodes, use this parameter to specify a taint toleration key, value, and effect for Scanner DB.
scanner.replicas	Resource specification for Collector’s Compliance container.
scanner.logLevel	Setting this parameter allows you to modify the scanner log level. Use this option only for troubleshooting purposes.
scanner.autoscaling.disable	If you set this option to **true**, Red Hat Advanced Cluster Security for Kubernetes disables autoscaling on the Scanner deployment.
scanner.autoscaling.minReplicas	The minimum number of replicas for autoscaling. Defaults to 2.
scanner.autoscaling.maxReplicas	The maximum number of replicas for autoscaling. Defaults to 5.
scanner.nodeSelector	Specify a node selector label as **label-key: label-value** to force Scanner to only schedule on nodes with the specified label.
scanner.tolerations	If the node selector selects tainted nodes, use this parameter to specify a taint toleration key, value, and effect for Scanner.

Parameter	Description
scanner.dbNodeSelector	Specify a node selector label as **label-key: label-value** to force Scanner DB to only schedule on nodes with the specified label.
scanner.dbTolerations	If the node selector selects tainted nodes, use this parameter to specify a taint toleration key, value, and effect for Scanner DB.
scanner.resources.requests.memory	The memory request for the Scanner container. Use this parameter to override the default value.
scanner.resources.requests.cpu	The CPU request for the Scanner container. Use this parameter to override the default value.
scanner.resources.limits.memory	The memory limit for the Scanner container. Use this parameter to override the default value.
scanner.resources.limits.cpu	The CPU limit for the Scanner container. Use this parameter to override the default value.
scanner.dbResources.requests.memory	The memory request for the Scanner DB container. Use this parameter to override the default value.
scanner.dbResources.requests.cpu	The CPU request for the Scanner DB container. Use this parameter to override the default value.
scanner.dbResources.limits.memory	The memory limit for the Scanner DB container. Use this parameter to override the default value.
scanner.dbResources.limits.cpu	The CPU limit for the Scanner DB container. Use this parameter to override the default value.

2.6.2.2.1.1. Environment variables

You can specify environment variables for Sensor and Admission controller in the following format:

```
customize:
  envVars:
    ENV_VAR1: "value1"
    ENV_VAR2: "value2"
```

The **customize** setting allows you to specify custom Kubernetes metadata (labels and annotations) for all objects created by this Helm chart and additional pod labels, pod annotations, and container environment variables for workloads.

The configuration is hierarchical, in the sense that metadata defined at a more generic scope (for example, for all objects) can be overridden by metadata defined at a narrower scope (for example, only for the Sensor deployment).

2.6.2.2.2. Installing the secured-cluster-services Helm chart

After you configure the **values-public.yaml** and **values-private.yaml** files, install the **secured-cluster-services** Helm chart to deploy the per-cluster and per-node components (Sensor, Admission controller, and Collector).

CAUTION

To install Collector on systems that have Unified Extensible Firmware Interface (UEFI) and that have Secure Boot enabled, you must use eBPF probes because kernel modules are unsigned, and the UEFI firmware cannot load unsigned packages. Collector identifies Secure Boot status at the start and switches to eBPF probes if required.

Prerequisites

- You must have generated RHACS init bundle for your cluster.
- You must have the address and the port number that you are exposing the Central service on.

Procedure

- Run the following command:

```
$ helm install -n stackrox --create-namespace \
  stackrox-secured-cluster-services rhacs/secured-cluster-services \
  -f <name_of_cluster_init_bundle.yaml> \
  -f <path_to_values_public.yaml> -f <path_to_values_private.yaml> 1
```

1 Use the **-f** option to specify the paths for your YAML configuration files.

NOTE

To deploy **secured-cluster-services** Helm chart by using a continuous integration (CI) system, pass the init bundle YAML file as an environment variable to the **helm install** command:

```
$ helm install ... -f <(echo "$INIT_BUNDLE_YAML_SECRET") 1
```

1 If you are using base64 encoded variables, use the **helm install … -f <(echo "$INIT_BUNDLE_YAML_SECRET" | base64 --decode)** command instead.

Additional resources

- Generating and applying an init bundle for RHACS on Red Hat OpenShift

2.6.2.3. Changing configuration options after deploying the secured-cluster-services Helm chart

You can make changes to any configuration options after you have deployed the **secured-cluster-services** Helm chart.

Procedure

1. Update the **values-public.yaml** and **values-private.yaml** configuration files with new values.

2. Run the **helm upgrade** command and specify the configuration files using the **-f** option:

   ```
   $ helm upgrade -n stackrox \
     stackrox-secured-cluster-services rhacs/secured-cluster-services \
     --reuse-values \ 1
     -f <path_to_values_public.yaml> \
     -f <path_to_values_private.yaml>
   ```

   1 You must specify the **--reuse-values** parameter, otherwise the Helm upgrade command resets all previously configured settings.

NOTE

You can also specify configuration values using the **--set** or **--set-file** parameters. However, these options are not saved, and it requires you to manually specify all the options again whenever you make changes.

2.6.3. Installing RHACS on secured clusters by using the roxctl CLI

To install RHACS on secured clusters by using the CLI, perform the following steps:

1. Install the **roxctl** CLI

2. Install Sensor.

2.6.3.1. Installing the roxctl CLI

You must first download the binary. You can install **roxctl** on Linux, Windows, or macOS.

2.6.3.1.1. Installing the roxctl CLI on Linux

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

Procedure

1. Download the latest version of the **roxctl** CLI:

   ```
   $ curl -O https://mirror.openshift.com/pub/rhacs/assets/4.0.5/bin/Linux/roxctl
   ```

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

   ```
   $ chmod +x roxctl
   ```

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

   ```
   $ echo $PATH
   ```

Verification

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

```
$ roxctl version
```

2.6.3.1.2. Installing the roxctl CLI on macOS

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

Procedure

1. Download the latest version of the **roxctl** CLI:

```
$ curl -O https://mirror.openshift.com/pub/rhacs/assets/4.0.5/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
```

2.6.3.1.3. Installing the roxctl CLI on Windows

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

Procedure

- Download the latest version of the **roxctl** CLI:

```
$ curl -O https://mirror.openshift.com/pub/rhacs/assets/4.0.5/bin/Windows/roxctl.exe
```

Verification

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

```
$ roxctl version
```

2.6.3.2. Installing Sensor

To monitor a cluster, you must deploy Sensor. You must deploy Sensor into each cluster that you want to monitor. The following steps describe adding Sensor by using the RHACS portal.

Prerequisites

- You must have already installed Central services, or you can access Central services by selecting your **ACS instance** on Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service).

Procedure

1. On your secured cluster, in the RHACS portal, navigate to **Platform Configuration → Clusters**.
2. Select **+ New Cluster**.
3. Specify a name for the cluster.
4. Provide appropriate values for the fields based on where you are deploying the Sensor.
    - If you are deploying Sensor in the same cluster, accept the default values for all the fields.
    - If you are deploying into a different cluster, replace **central.stackrox.svc:443** with a load balancer, node port, or other address, including the port number, that is accessible from the other cluster.
    - If you are using a non-gRPC capable load balancer, such as HAProxy, AWS Application Load Balancer (ALB), or AWS Elastic Load Balancing (ELB), use the WebSocket Secure (**wss**) protocol. To use **wss**:
        - Prefix the address with **wss://**.
        - Add the port number after the address, for example, **wss://stackrox-central.example.com:443**.
5. Click **Next** to continue with the Sensor setup.
6. Click **Download YAML File and Keys** to download the cluster bundle (zip archive).

    IMPORTANT

    The cluster bundle zip archive includes unique configurations and keys for each cluster. Do not reuse the same files in another cluster.

7. From a system that has access to the monitored cluster, unzip and run the **sensor** script from the cluster bundle:

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

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

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

After Sensor is deployed, it contacts Central and provides cluster information.

Verification

1. Return to the RHACS portal and check if the deployment is successful. If successful, when viewing your list of clusters in **Platform Configuration → Clusters**, the cluster status displays a green checkmark and a **Healthy** status. If you do not see a green checkmark, use the following command to check for problems:

   - On OpenShift Container Platform, enter the following command:

     ```
     $ oc get pod -n stackrox -w
     ```

   - On Kubernetes, enter the following command:

     ```
     $ kubectl get pod -n stackrox -w
     ```

2. Click **Finish** to close the window.

After installation, Sensor starts reporting security information to RHACS and the RHACS portal dashboard begins showing deployments, images, and policy violations from the cluster on which you have installed the Sensor.

2.7. VERIFYING INSTALLATION OF RHACS ON RED HAT OPENSHIFT

Provides steps to verify that RHACS is properly installed.

2.7.1. Verifying installation

After you complete the installation, run a few vulnerable applications and navigate to the RHACS portal to evaluate the results of security assessments and policy violations.

> **NOTE**
>
> The sample applications listed in the following section contain critical vulnerabilities and they are specifically designed to verify the build and deploy-time assessment features of Red Hat Advanced Cluster Security for Kubernetes.

To verify installation:

1. Find the address of the RHACS portal based on your exposure method:

   a. For a route:

      ```
      $ oc get route central -n stackrox
      ```

   b. For a load balancer:

      ```
      $ oc get service central-loadbalancer -n stackrox
      ```

   c. For port forward:

      i. Run the following command:

         ```
         $ oc port-forward svc/central 18443:443 -n stackrox
         ```

   ii. Navigate to **https://localhost:18443/**.

2. Using the Red Hat OpenShift CLI, create a new project:

```
$ oc new-project test
```

3. Start some applications with critical vulnerabilities:

```
$ oc run shell --labels=app=shellshock,team=test-team \
  --image=vulnerables/cve-2014-6271 -n test
$ oc run samba --labels=app=rce \
  --image=vulnerables/cve-2017-7494 -n test
```

Red Hat Advanced Cluster Security for Kubernetes automatically scans these deployments for security risks and policy violations as soon as they are submitted to the cluster. Navigate to the RHACS portal to view the violations. You can log in to the RHACS portal by using the default username **admin** and the generated password.

CHAPTER 3. INSTALLING RHACS ON OTHER PLATFORMS

3.1. HIGH-LEVEL OVERVIEW OF INSTALLING RHACS ON OTHER PLATFORMS

Red Hat Advanced Cluster Security for Kubernetes (RHACS) provides security services for self-managed RHACS on platforms such as Amazon Elastic Kubernetes Service (Amazon EKS), Google Kubernetes Engine (Google GKE), and Microsoft Azure Kubernetes Service (Microsoft AKS).

Before you install:

- Understand the installation platforms and methods.
- Understand Red Hat Advanced Cluster Security for Kubernetes architecture .
- Review the prerequisites.

The following list provides a high-level overview of installation steps:

1. Install Central services on a cluster using Helm charts or the **roxctl** CLI.
2. Generate and apply an init bundle.
3. Install secured cluster resources on each of your secured clusters.

3.2. PREREQUISITES FOR RHACS ON OTHER PLATFORMS

Before installing RHACS on other platforms such as Amazon Elastic Kubernetes Service (Amazon EKS), Google Kubernetes Engine (Google GKE), and Microsoft Azure Kubernetes Service (Microsoft AKS), ensure that you have met the prerequisites.

3.2.1. General requirements

RHACS has some system requirements that must be met before installing.

WARNING

You must not install Red Hat Advanced Cluster Security for Kubernetes on:

- Amazon Elastic File System (Amazon EFS). Use the Amazon Elastic Block Store (Amazon EBS) with the default **gp2** volume type instead.
- Older CPUs that do not have the Streaming SIMD Extensions (SSE) 4.2 instruction set. For example, Intel processors older than *Sandy Bridge* and AMD processors older than *Bulldozer*. (These processors were released in 2011.)

To install Red Hat Advanced Cluster Security for Kubernetes, you must have:

- OpenShift Container Platform version 4.10 or later. For more information about supported self-managed and managed OpenShift Container Platform versions, see Red Hat Advanced Cluster Security for Kubernetes Support Policy.
- Cluster nodes with a supported operating system:
  - Red Hat Enterprise Linux CoreOS (RHCOS), Red Hat Enterprise Linux (RHEL).
- A supported managed Kubernetes platform. For more information, see Red Hat Advanced Cluster Security for Kubernetes Support Policy.
- Cluster nodes with a supported operating system:
  - **Operating system**: Amazon Linux, CentOS, Container-Optimized OS from Google, Red Hat Enterprise Linux CoreOS (RHCOS), Debian, Red Hat Enterprise Linux (RHEL), or Ubuntu.
  - **Processor and memory**: 2 CPU cores and at least 3GiB of RAM.

    NOTE

    For deploying Central, use a machine type with four or more cores and apply scheduling policies to launch Central on such nodes.

  - **Architectures**: AMD64, ppc64le, or s390x.

    NOTE

    For ppc64le, or s390x architectures, you can only install RHACS Secured cluster services on IBM Power, IBM zSystems, and IBM® LinuxONE clusters. Central is not supported at this time.

- Persistent storage by using persistent volume claim (PVC).

  IMPORTANT

  You must not use Ceph FS storage with Red Hat Advanced Cluster Security for Kubernetes. Red Hat recommends using RBD block mode PVCs for Red Hat Advanced Cluster Security for Kubernetes.

  - Use Solid-State Drives (SSDs) for best performance. However, you can use another storage type if you do not have SSDs available.

To install using Helm charts:

- You must have Helm command-line interface (CLI) v3.2 or newer, if you are installing or configuring Red Hat Advanced Cluster Security for Kubernetes using Helm charts. Use the **helm version** command to verify the version of Helm you have installed.
- The Red Hat OpenShift CLI (**oc**).
- You must have access to the Red Hat Container Registry. For information about downloading images from **registry.redhat.io**, see Red Hat Container Registry Authentication .

3.2.2. Prerequisites for installing Central

A containerized service called Central handles API interactions and user interface (Portal) access while a containerized service called Central DB (PostgreSQL 13) handles data persistence.

Both Central and Central DB require persistent storage:

- You can provide storage with a persistent volume claim (PVC).

  NOTE

  You can use a hostPath volume for storage only if all your hosts (or a group of hosts) mount a shared file system, such as an NFS share or a storage appliance. Otherwise, your data is only saved on a single node. Red Hat does not recommend using a hostPath volume.

- Use Solid-State Drives (SSD) for best performance. However, you can use another storage type if you do not have SSDs available.

- If you use a web proxy or firewall, you must configure bypass rules to allow traffic for the **definitions.stackrox.io** and **collector-modules.stackrox.io** domains and enable Red Hat Advanced Cluster Security for Kubernetes to trust your web proxy or firewall. Otherwise, updates for vulnerability definitions and kernel support packages will fail.
  Red Hat Advanced Cluster Security for Kubernetes requires access to:

  - **definitions.stackrox.io** for downloading updated vulnerability definitions. Vulnerability definition updates allow Red Hat Advanced Cluster Security for Kubernetes to maintain up-to-date vulnerability data when new vulnerabilities are discovered or additional data sources are added.

  - **collector-modules.stackrox.io** to download updated kernel support packages. Updated Kernel support packages ensure that Red Hat Advanced Cluster Security for Kubernetes can monitor the latest operating systems and collect data about the network traffic and processes running inside the containers. Without these updates, Red Hat Advanced Cluster Security for Kubernetes might fail to monitor containers if you add new nodes in your cluster or if you update your nodes' operating system.

NOTE

For security reasons, you should deploy Central in a cluster with limited administrative access.

Memory and storage requirements
The following table lists the minimum memory and storage values required to install and run Central.

Central	CPU	Memory	Storage
Request	1.5 cores	4 GiB	100 GiB
Limit	4 cores	8 GiB	100 GiB

Central DB	CPU	Memory	Storage
Request	4 cores	8 GiB	100 GiB
Limit	8 cores	16 GiB	100 GiB

Sizing guidelines
Use the following compute resources and storage values depending upon the number of nodes in your cluster.

Nodes	Deployments	Central CPU	Central Memory	Central Storage
Up to 100	Up to 1000	2 cores	4 GiB	100 GiB
Up to 500	Up to 2000	4 cores	8 GiB	100 GiB
More than 500	More than 2000	8 cores	12 - 16 GiB	100 - 200 GiB

Nodes	Deployments	Central DB CPU	Central DB Memory	Central DB Storage
Up to 100	Up to 1000	2 cores	4 GiB	100 GiB
Up to 500	Up to 2000	4 cores	8 GiB	100 GiB
More than 500	More than 2000	8 cores	12 - 16 GiB	100 - 200 GiB

3.2.3. Prerequisites for installing Scanner

Red Hat Advanced Cluster Security for Kubernetes includes an image vulnerability scanner called Scanner. This service scans images that are not already scanned by scanners integrated into image registries.

Memory and storage requirements

Scanner	CPU	Memory
Request	1.2 cores	2700 MiB
Limit	5 cores	8000 MiB

3.2.4. Prerequisites for installing Sensor

Sensor monitors your Kubernetes and OpenShift Container Platform clusters. These services currently deploy in a single deployment, which handles interactions with the Kubernetes API and coordinates with Collector.

Memory and storage requirements

Sensor	CPU	Memory
Request	2 cores	4 GiB
Limit	4 cores	8 GiB

3.2.5. Prerequisites for installing Admission controller

The Admission controller prevents users from creating workloads that violate policies you configure.

Memory and storage requirements
By default, the admission control service runs 3 replicas. The following table lists the request and limits for each replica.

Admission controller	CPU	Memory
Request	.05 cores	100 MiB
Limit	.5 cores	500 MiB

3.2.6. Prerequisites for installing Collector

Collector monitors runtime activity on each node in your secured clusters. It connects to Sensor to report this information.

CAUTION

To install Collector on systems that have Unified Extensible Firmware Interface (UEFI) and that have Secure Boot enabled, you must use eBPF probes because kernel modules are unsigned, and the UEFI firmware cannot load unsigned packages. Collector identifies Secure Boot status at the start and switches to eBPF probes if required.

Memory and storage requirements

Collector	CPU	Memory
Request	.05 cores	320 MiB
Limit	.75 cores	1 GiB

NOTE

Collector uses a mutable image tag (**<version>-latest**), so you get support for newer Linux kernel versions more easily. There is no change in code, pre-existing kernel modules, or eBPF programs for image updates. Updates only add a single image layer with support for new kernel versions published after the initial release.

3.3. INSTALLING CENTRAL SERVICES FOR RHACS ON OTHER PLATFORMS

Central is the resource that contains the RHACS application management interface and services. It handles data persistence, API interactions, and RHACS portal access. You can use the same Central instance to secure multiple OpenShift Container Platform or Kubernetes clusters.

You can install Central by using one of the following methods:

- Install using Helm charts
- Install using the **roxctl** CLI (do not use this method unless you have a specific installation need that requires using it)

3.3.1. Install Central using Helm charts

You can install Central using Helm charts without any customization, using the default values, or by using Helm charts with additional customizations of configuration parameters.

3.3.1.1. Install Central using Helm charts without customization

You can install RHACS on your Red Hat OpenShift cluster without any customizations. You must add the Helm chart repository and install the **central-services** Helm chart to install the centralized components of Central and Scanner.

3.3.1.1.1. Adding the Helm chart repository

Procedure

- Add the RHACS charts repository.

```
$ helm repo add rhacs https://mirror.openshift.com/pub/rhacs/charts/
```

The Helm repository for Red Hat Advanced Cluster Security for Kubernetes includes Helm charts for installing different components, including:

- Central services Helm chart (**central-services**) for installing the centralized components (Central and Scanner).

  > **NOTE**
  >
  > You deploy centralized components only once and you can monitor multiple separate clusters by using the same installation.

- Secured Cluster Services Helm chart (**secured-cluster-services**) for installing the per-cluster (Sensor and Admission controller) and per-node (Collector) components.

  > **NOTE**
  >
  > Deploy the per-cluster components into each cluster that you want to monitor and deploy the per-node components in all nodes that you want to monitor.

Verification

- Run the following command to verify the added chart repository:

  ```
  $ helm search repo -l rhacs/
  ```

3.3.1.1.2. Installing the central-services Helm chart without customizations

Use the following instructions to install the **central-services** Helm chart to deploy the centralized components (Central and Scanner).

Prerequisites

- You must have access to the Red Hat Container Registry. For information about downloading images from **registry.redhat.io**, see Red Hat Container Registry Authentication .

Procedure

- Run the following command to install Central services and expose Central using a route:

  ```
  $ helm install -n stackrox \
    --create-namespace stackrox-central-services rhacs/central-services \
    --set imagePullSecrets.username=<username> \
    --set imagePullSecrets.password=<password> \
    --set central.exposure.route.enabled=true
  ```

- Or, run the following command to install Central services and expose Central using a load balancer:

  ```
  $ helm install -n stackrox \
    --create-namespace stackrox-central-services rhacs/central-services \
    --set imagePullSecrets.username=<username> \
    --set imagePullSecrets.password=<password> \
    --set central.exposure.loadBalancer.enabled=true
  ```

- Or, run the following command to install Central services and expose Central using port forward:

  ```
  $ helm install -n stackrox \
    --create-namespace stackrox-central-services rhacs/central-services \
      --set imagePullSecrets.username=<username> \
    --set imagePullSecrets.password=<password>
  ```

IMPORTANT

- If you are installing Red Hat Advanced Cluster Security for Kubernetes in a cluster that requires a proxy to connect to external services, you must specify your proxy configuration by using the **proxyConfig** parameter. For example:

```
env:
  proxyConfig: |
    url: http://proxy.name:port
    username: username
    password: password
    excludes:
    - some.domain
```

- If you already created one or more image pull secrets in the namespace in which you are installing, instead of using a username and password, you can use **--set imagePullSecrets.useExisting="<pull-secret-1;pull-secret-2>"**.
- Do not use image pull secrets:
  - If you are pulling your images from **quay.io/stackrox-io** or a registry in a private network that does not require authentication. Use use **--set imagePullSecrets.allowNone=true** instead of specifying a username and password.
  - If you already configured image pull secrets in the default service account in the namespace you are installing. Use **--set imagePullSecrets.useFromDefaultServiceAccount=true** instead of specifying a username and password.

The output of the installation command includes:

- An automatically generated administrator password.
- Instructions on storing all the configuration values.
- Any warnings that Helm generates.

3.3.1.2. Install Central using Helm charts with customizations

You can install RHACS on your Red Hat OpenShift cluster with customizations by using Helm chart configuration parameters with the **helm install** and **helm upgrade** commands. You can specify these parameters by using the **--set** option or by creating YAML configuration files.

Create the following files for configuring the Helm chart for installing Red Hat Advanced Cluster Security for Kubernetes:

- Public configuration file **values-public.yaml**: Use this file to save all non-sensitive configuration options.
- Private configuration file **values-private.yaml**: Use this file to save all sensitive configuration options. Ensure that you store this file securely.

3.3.1.2.1. Private configuration file

This section lists the configurable parameters of the **values-private.yaml** file. There are no default values for these parameters.

3.3.1.2.1.1. Image pull secrets

The credentials that are required for pulling images from the registry depend on the following factors:

- If you are using a custom registry, you must specify these parameters:
  - **imagePullSecrets.username**
  - **imagePullSecrets.password**
  - **image.registry**
- If you do not use a username and password to log in to the custom registry, you must specify one of the following parameters:
  - **imagePullSecrets.allowNone**
  - **imagePullSecrets.useExisting**
  - **imagePullSecrets.useFromDefaultServiceAccount**

Parameter	Description
imagePullSecrets.username	The username of the account that is used to log in to the registry.
imagePullSecrets.password	The password of the account that is used to log in to the registry.
imagePullSecrets.allowNone	Use **true** if you are using a custom registry and it allows pulling images without credentials.
imagePullSecrets.useExisting	A comma-separated list of secrets as values. For example, **secret1, secret2, secretN**. Use this option if you have already created pre-existing image pull secrets with the given name in the target namespace.
imagePullSecrets.useFromDefaultServiceAccount	Use **true** if you have already configured the default service account in the target namespace with sufficiently scoped image pull secrets.

3.3.1.2.1.2. Proxy configuration

If you are installing Red Hat Advanced Cluster Security for Kubernetes in a cluster that requires a proxy to connect to external services, you must specify your proxy configuration by using the **proxyConfig** parameter. For example:

```
env:
  proxyConfig: |
url: http://proxy.name:port
username: username
password: password
excludes:
- some.domain
```

Parameter	Description
env.proxyConfig	Your proxy configuration.

3.3.1.2.1.3. Central

Configurable parameters for Central.

For a new installation, you can skip the following parameters:

- **central.jwtSigner.key**
- **central.serviceTLS.cert**
- **central.serviceTLS.key**
- **central.adminPassword.value**
- **central.adminPassword.htpasswd**
- **central.db.serviceTLS.cert**
- **central.db.serviceTLS.key**
- **central.db.password.value**
- When you do not specify values for these parameters the Helm chart autogenerates values for them.
- If you want to modify these values you can use the **helm upgrade** command and specify the values using the **--set** option.

IMPORTANT

For setting the administrator password, you can only use either **central.adminPassword.value** or **central.adminPassword.htpasswd**, but not both.

Parameter	Description
central.jwtSigner.key	A private key which Red Hat Advanced Cluster Security for Kubernetes should use for signing JSON web tokens (JWTs) for authentication.
central.serviceTLS.cert	An internal certificate that the Central service should use for deploying Central.

Parameter	Description
central.serviceTLS.key	The private key of the internal certificate that the Central service should use.
central.defaultTLS.cert	The user-facing certificate that Central should use. Red Hat Advanced Cluster Security for Kubernetes uses this certificate for RHACS portal. ● For a new installation, you must provide a certificate, otherwise, Red Hat Advanced Cluster Security for Kubernetes installs Central by using a self-signed certificate. ● If you are upgrading, Red Hat Advanced Cluster Security for Kubernetes uses the existing certificate and its key.
central.defaultTLS.key	The private key of the user-facing certificate that Central should use. ● For a new installation, you must provide the private key, otherwise, Red Hat Advanced Cluster Security for Kubernetes installs Central by using a self-signed certificate. ● If you are upgrading, Red Hat Advanced Cluster Security for Kubernetes uses the existing certificate and its key.
central.db.password.value	Connection password for Central database.
central.adminPassword.value	Administrator password for logging into Red Hat Advanced Cluster Security for Kubernetes.
central.adminPassword.htpasswd	Administrator password for logging into Red Hat Advanced Cluster Security for Kubernetes. This password is stored in hashed format using bcrypt.
central.db.serviceTLS.cert	An internal certificate that the Central DB service should use for deploying Central DB.
central.db.serviceTLS.key	The private key of the internal certificate that the Central DB service should use.
central.db.password.value	The password used to connect to the Central DB.

NOTE

If you are using **central.adminPassword.htpasswd** parameter, you must use a bcrypt encoded password hash. You can run the command **htpasswd -nB admin** to generate a password hash. For example,

```
htpasswd: |
  admin:<bcrypt-hash>
```

3.3.1.2.1.4. Scanner

Configurable parameters for Scanner.

For a new installation, you can skip the following parameters and the Helm chart autogenerates values for them. Otherwise, if you are upgrading to a new version, specify the values for the following parameters:

- **scanner.dbPassword.value**
- **scanner.serviceTLS.cert**
- **scanner.serviceTLS.key**
- **scanner.dbServiceTLS.cert**
- **scanner.dbServiceTLS.key**

Parameter	Description
scanner.dbPassword.value	The password to use for authentication with Scanner database. Do not modify this parameter because Red Hat Advanced Cluster Security for Kubernetes automatically creates and uses its value internally.
scanner.serviceTLS.cert	An internal certificate that the Scanner service should use for deploying Scanner.
scanner.serviceTLS.key	The private key of the internal certificate that the Scanner service should use.
scanner.dbServiceTLS.cert	An internal certificate that the Scanner-db service should use for deploying Scanner database.
scanner.dbServiceTLS.key	The private key of the internal certificate that the Scanner-db service should use.

3.3.1.2.2. Public configuration file

This section lists the configurable parameters of the **values-public.yaml** file.

3.3.1.2.2.1. Image pull secrets

Image pull secrets are the credentials required for pulling images from your registry.

Parameter	Description
imagePullSecrets.allowNone	Use **true** if you are using a custom registry and it allows pulling images without credentials.
imagePullSecrets.useExisting	A comma-seprated list of secrets as values. For example, **secret1, secret2**. Use this option if you have already created pre-existing image pull secrets with the given name in the target namespace.
imagePullSecrets.useFromDefaultServiceAccount	Use **true** if you have already configured the default service account in the target namespace with sufficiently scoped image pull secrets.

3.3.1.2.2.2. Image

Image declares the configuration to set up the main registry, which the Helm chart uses to resolve images for the **central.image**, **scanner.image**, and **scanner.dbImage** parameters.

Parameter	Description
image.registry	Address of your image registry. Either use a hostname, such as **registry.redhat.io**, or a remote registry hostname, such as **us.gcr.io/stackrox-mirror**.

3.3.1.2.2.3. Environment variables

Red Hat Advanced Cluster Security for Kubernetes automatically detects your cluster environment and sets values for **env.openshift**, **env.istio**, and **env.platform**. Only set these values to override the automatic cluster environment detection.

Parameter	Description
env.openshift	Use **true** for installing on an OpenShift Container Platform cluster and overriding automatic cluster environment detection.
env.istio	Use **true** for installing on an Istio enabled cluster and overriding automatic cluster environment detection.
env.platform	The platform on which you are installing Red Hat Advanced Cluster Security for Kubernetes. Set its value to **default** or **gke** to specify cluster platform and override automatic cluster environment detection.

Parameter	Description
env.offlineMode	Use **true** to use Red Hat Advanced Cluster Security for Kubernetes in offline mode.

3.3.1.2.2.4. Additional trusted certificate authorities

The Red Hat Advanced Cluster Security for Kubernetes automatically references the system root certificates to trust. When Central or Scanner must reach out to services that use certificates issued by an authority in your organization or a globally trusted partner organization, you can add trust for these services by specifying the root certificate authority to trust by using the following parameter:

Parameter	Description
additionalCAs.<certificate_name>	Specify the PEM encoded certificate of the root certificate authority to trust.

3.3.1.2.2.5. Central

Configurable parameters for Central.

- You must specify a persistent storage option as either **hostPath** or **persistentVolumeClaim**.
- For exposing Central deployment for external access. You must specify one parameter, either **central.exposure.loadBalancer**, **central.exposure.nodePort**, or **central.exposure.route**. When you do not specify any value for these parameters, you must manually expose Central or access it by using port-forwarding.

The following table includes settings for an external PostgreSQL database (Technology Preview).

IMPORTANT

External PostgreSQL support 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 .

Parameter	Description
central.endpointsConfig	The endpoint configuration options for Central.
central.nodeSelector	If the node selector selects tainted nodes, use this parameter to specify a taint toleration key, value, and effect for Central. This parameter is mainly used for infrastructure nodes.

Parameter	Description
central.tolerations	If the node selector selects tainted nodes, use this parameter to specify a taint toleration key, value, and effect for Central. This parameter is mainly used for infrastructure nodes.
central.exposeMonitoring	Specify **true** to expose Prometheus metrics endpoint for Central on port number **9090**.
central.image.registry	A custom registry that overrides the global **image.registry** parameter for the Central image.
central.image.name	The custom image name that overrides the default Central image name (**main**).
central.image.tag	The custom image tag that overrides the default tag for Central image. If you specify your own image tag during a new installation, you must manually increment this tag when you to upgrade to a new version by running the **helm upgrade** command. If you mirror Central images in your own registry, do not modify the original image tags.
central.image.fullRef	Full reference including registry address, image name, and image tag for the Central image. Setting a value for this parameter overrides the **central.image.registry**, **central.image.name**, and **central.image.tag** parameters.
central.resources.requests.memory	The memory request for Central to override the default value.
central.resources.requests.cpu	The CPU request for Central to override the default value.
central.resources.limits.memory	The memory limit for Central to override the default value.
central.resources.limits.cpu	The CPU limit for Central to override the default value.
central.persistence.hostPath	The path on the node where RHACS should create a database volume. Red Hat does not recommend using this option.

Parameter	Description
central.persistence.persistentVolumeClaim.claimName	The name of the persistent volume claim (PVC) you are using.
central.persistence.persistentVolumeClaim.createClaim	Use **true** to create a new PVC, or**false** to use an existing claim.
central.persistence.persistentVolumeClaim.size	The size (in GiB) of the persistent volume managed by the specified claim.
central.exposure.loadBalancer.enabled	Use **true** to expose Central by using a load balancer.
central.exposure.loadBalancer.port	The port number on which to expose Central. The default port number is 443.
central.exposure.nodePort.enabled	Use **true** to expose Central by using the node port service.
central.exposure.nodePort.port	The port number on which to expose Central. When you skip this parameter, OpenShift Container Platform automatically assigns a port number. Red Hat recommends that you do not specify a port number if you are exposing Red Hat Advanced Cluster Security for Kubernetes by using a node port.
central.exposure.route.enabled	Use **true** to expose Central by using a route. This parameter is only available for OpenShift Container Platform clusters.
central.db.external	(Technology Preview) Use **true** to specify that Central DB should not be deployed and that an external database will be used.
central.db.source.connectionString	(Technology Preview) The connection string for Central to use to connect to the database. This is only used when **central.db.external** is set to true. The connection string must be in keyword/value format as described in the PostgreSQL documentation in "Additional resources". ● Only PostgreSQL 13 is supported. ● Connections through PgBouncer are not supported. ● User must be superuser with ability to create and delete databases.

Parameter	Description
central.db.source.minConns	The minimum number of connections to the database to be established.
central.db.source.maxConns	The maximum number of connections to the database to be established.
central.db.source.statementTimeoutMs	The number of milliseconds a single query or transaction can be active against the database.
central.db.postgresConfig	The postgresql.conf to be used for Central DB as described in the PostgreSQL documentation in "Additional resources".
central.db.hbaConfig	The pg_hba.conf to be used for Central DB as described in the PostgreSQL documentation in "Additional resources".
central.db.nodeSelector	Specify a node selector label as **label-key: label-value** to force Central DB to only schedule on nodes with the specified label.
central.db.image.registry	A custom registry that overrides the global **image.registry** parameter for the Central DB image.
central.db.image.name	The custom image name that overrides the default Central DB image name (**central-db**).
central.db.image.tag	The custom image tag that overrides the default tag for Central DB image. If you specify your own image tag during a new installation, you must manually increment this tag when you to upgrade to a new version by running the **helm upgrade** command. If you mirror Central DB images in your own registry, do not modify the original image tags.
central.db.image.fullRef	Full reference including registry address, image name, and image tag for the Central DB image. Setting a value for this parameter overrides the **central.db.image.registry**, **central.db.image.name**, and **central.db.image.tag** parameters.
central.db.resources.requests.memory	The memory request for Central DB to override the default value.
central.db.resources.requests.cpu	The CPU request for Central DB to override the default value.

Parameter	Description
central.db.resources.limits.memory	The memory limit for Central DB to override the default value.
central.db.resources.limits.cpu	The CPU limit for Central DB to override the default value.
central.db.persistence.hostPath	The path on the node where RHACS should create a database volume. Red Hat does not recommend using this option.
central.db.persistence.persistentVolumeClaim.claimName	The name of the persistent volume claim (PVC) you are using.
central.db.persistence.persistentVolumeClaim.createClaim	Use **true** to create a new persistent volume claim, or **false** to use an existing claim.
central.db.persistence.persistentVolumeClaim.size	The size (in GiB) of the persistent volume managed by the specified claim.

3.3.1.2.2.6. Scanner

Configurable parameters for Scanner.

Parameter	Description
scanner.disable	Use **true** to install Red Hat Advanced Cluster Security for Kubernetes without Scanner. When you use it with the **helm upgrade** command, Helm removes existing Scanner deployment.
scanner.exposeMonitoring	Specify **true** to expose Prometheus metrics endpoint for Scanner on port number **9090**.
scanner.replicas	The number of replicas to create for the Scanner deployment. When you use it with the **scanner.autoscaling** parameter, this value sets the initial number of replicas.
scanner.logLevel	Configure the log level for Scanner. Red Hat recommends that you not change the log level's default value (**INFO**).
scanner.nodeSelector	Specify a node selector label as **label-key: label-value** to force Scanner to only schedule on nodes with the specified label.

Parameter	Description
scanner.tolerations	If the node selector selects tainted nodes, use this parameter to specify a taint toleration key, value, and effect for Scanner. This parameter is mainly used for infrastructure nodes.
scanner.autoscaling.disable	Use **true** to disable autoscaling for Scanner deployment. When you disable autoscaling, the **minReplicas** and **maxReplicas** parameters do not have any effect.
scanner.autoscaling.minReplicas	The minimum number of replicas for autoscaling.
scanner.autoscaling.maxReplicas	The maximum number of replicas for autoscaling.
scanner.resources.requests.memory	The memory request for Scanner to override the default value.
scanner.resources.requests.cpu	The CPU request for Scanner to override the default value.
scanner.resources.limits.memory	The memory limit for Scanner to override the default value.
scanner.resources.limits.cpu	The CPU limit for Scanner to override the default value.
scanner.dbResources.requests.memory	The memory request for Scanner database deployment to override the default values.
scanner.dbResources.requests.cpu	The CPU request for Scanner database deployment to override the default values.
scanner.dbResources.limits.memory	The memory limit for Scanner database deployment to override the default values.
scanner.dbResources.limits.cpu	The CPU limit for Scanner database deployment to override the default values.
scanner.image.registry	A custom registry for the Scanner image.
scanner.image.name	The custom image name that overrides the default Scanner image name (**scanner**).
scanner.dbImage.registry	A custom registry for the Scanner DB image.
scanner.dbImage.name	The custom image name that overrides the default Scanner DB image name (**scanner-db**).

Parameter	Description
scanner.dbNodeSelector	Specify a node selector label as **label-key: label-value** to force Scanner DB to only schedule on nodes with the specified label.
scanner.dbTolerations	If the node selector selects tainted nodes, use this parameter to specify a taint toleration key, value, and effect for Scanner DB. This parameter is mainly used for infrastructure nodes.

3.3.1.2.2.7. Customization

Use these parameters to specify additional attributes for all objects that Red Hat Advanced Cluster Security for Kubernetes creates.

Parameter	Description
customize.labels	A custom label to attach to all objects.
customize.annotations	A custom annotation to attach to all objects.
customize.podLabels	A custom label to attach to all deployments.
customize.podAnnotations	A custom annotation to attach to all deployments.
customize.envVars	A custom environment variable for all containers in all objects.
customize.central.labels	A custom label to attach to all objects that Central creates.
customize.central.annotations	A custom annotation to attach to all objects that Central creates.
customize.central.podLabels	A custom label to attach to all Central deployments.
customize.central.podAnnotations	A custom annotation to attach to all Central deployments.
customize.central.envVars	A custom environment variable for all Central containers.
customize.scanner.labels	A custom label to attach to all objects that Scanner creates.
customize.scanner.annotations	A custom annotation to attach to all objects that Scanner creates.

Parameter	Description
customize.scanner.podLabels	A custom label to attach to all Scanner deployments.
customize.scanner.podAnnotations	A custom annotation to attach to all Scanner deployments.
customize.scanner.envVars	A custom environment variable for all Scanner containers.
customize.scanner-db.labels	A custom label to attach to all objects that Scanner DB creates.
customize.scanner-db.annotations	A custom annotation to attach to all objects that Scanner DB creates.
customize.scanner-db.podLabels	A custom label to attach to all Scanner DB deployments.
customize.scanner-db.podAnnotations	A custom annotation to attach to all Scanner DB deployments.
customize.scanner-db.envVars	A custom environment variable for all Scanner DB containers.

You can also use:

- the **customize.other.service/*.labels** and the **customize.other.service/*.annotations** parameters, to specify labels and annotations for all objects.
- or, provide a specific service name, for example, **customize.other.service/central-loadbalancer.labels** and **customize.other.service/central-loadbalancer.annotations** as parameters and set their value.

3.3.1.2.2.8. Advanced customization

IMPORTANT

The parameters specified in this section are for information only. Red Hat does not support Red Hat Advanced Cluster Security for Kubernetes instances with modified namespace and release names.

Parameter	Description
allowNonstandardNamespace	Use **true** to deploy Red Hat Advanced Cluster Security for Kubernetes into a namespace other than the default namespace **stackrox**.

Parameter	Description
allowNonstandardReleaseName	Use **true** to deploy Red Hat Advanced Cluster Security for Kubernetes with a release name other than the default **stackrox-central-services**.

Additional resources

- Connection Strings - PostgreSQL Docs
- Parameter Interaction via the Configuration File - PostgreSQL Docs
- The pg_hba.conf File - PostgreSQL Docs

3.3.1.2.3. Installing the central-services Helm chart

After you configure the **values-public.yaml** and **values-private.yaml** files, install the **central-services** Helm chart to deploy the centralized components (Central and Scanner).

Procedure

- Run the following command:

```
$ helm install -n stackrox --create-namespace \
  stackrox-central-services rhacs/central-services \
  -f <path_to_values_public.yaml> -f <path_to_values_private.yaml> ①
```

① Use the **-f** option to specify the paths for your YAML configuration files.

3.3.1.3. Changing configuration options after deploying the central-services Helm chart

You can make changes to any configuration options after you have deployed the **central-services** Helm chart.

Procedure

1. Update the **values-public.yaml** and **values-private.yaml** configuration files with new values.
2. Run the **helm upgrade** command and specify the configuration files using the **-f** option:

```
$ helm upgrade -n stackrox \
  stackrox-central-services rhacs/central-services \
  -f <path_to_values_public.yaml> \
  -f <path_to_values_private.yaml>
```

NOTE

You can also specify configuration values using the **--set** or **--set-file** parameters. However, these options are not saved, and it requires you to manually specify all the options again whenever you make changes.

3.3.2. Install Central using the roxctl CLI

WARNING

For production environments, Red Hat recommends using the Operator or Helm charts to install RHACS. Do not use the **roxctl** install method unless you have a specific installation need that requires using this method.

3.3.2.1. Installing the roxctl CLI

To install Red Hat Advanced Cluster Security for Kubernetes you must install the **roxctl** CLI by downloading the binary. You can install **roxctl** on Linux, Windows, or macOS.

3.3.2.1.1. Installing the roxctl CLI on Linux

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

Procedure

1. Download the latest version of the **roxctl** CLI:

```
$ curl -O https://mirror.openshift.com/pub/rhacs/assets/4.0.5/bin/Linux/roxctl
```

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

```
$ chmod +x roxctl
```

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

```
$ echo $PATH
```

Verification

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

```
$ roxctl version
```

3.3.2.1.2. Installing the roxctl CLI on macOS

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

Procedure

1. Download the latest version of the **roxctl** CLI:

```
$ curl -O https://mirror.openshift.com/pub/rhacs/assets/4.0.5/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
```

3.3.2.1.3. Installing the roxctl CLI on Windows

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

Procedure

- Download the latest version of the **roxctl** CLI:

```
$ curl -O https://mirror.openshift.com/pub/rhacs/assets/4.0.5/bin/Windows/roxctl.exe
```

Verification

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

```
$ roxctl version
```

3.3.2.2. Using the interactive installer

Use the interactive installer to generate the required secrets, deployment configurations, and deployment scripts for your environment.

Procedure

1. Run the interactive install command:

```
$ roxctl central generate interactive
```

IMPORTANT

Installing Red Hat Advanced Cluster Security for Kubernetes using **roxctl** CLI creates PodSecurityPolicy (PSP) objects by default for backward compatibility. If you install RHACS on Kubernetes versions 1.25 and newer or OpenShift Container Platform version 4.12 and newer, you must disable the PSP object creation. To do this, specify **--enable-pod-security-policies** option as **false** for the **roxctl central generate** and **roxctl sensor generate** commands.

2. Press **Enter** to accept the default value for a prompt or enter custom values as required.

```
Enter path to the backup bundle from which to restore keys and certificates (optional):
Enter PEM cert bundle file (optional): ①
Enter administrator password (default: autogenerated):
Enter orchestrator (k8s, openshift): openshift
Enter the directory to output the deployment bundle to (default: "central-bundle"):
Enter the OpenShift major version (3 or 4) to deploy on (default: "0"): 4
Enter Istio version when deploying into an Istio-enabled cluster (leave empty when not
running Istio) (optional):
Enter the method of exposing Central (route, lb, np, none) (default: "none"): route ②
Enter main image to use (default: "stackrox.io/main:3.0.61.1"):
Enter whether to run StackRox in offline mode, which avoids reaching out to the Internet
(default: "false"):
Enter whether to enable telemetry (default: "true"):
Enter the deployment tool to use (kubectl, helm, helm-values) (default: "kubectl"):
Enter Scanner DB image to use (default: "stackrox.io/scanner-db:2.15.2"):
Enter Scanner image to use (default: "stackrox.io/scanner:2.15.2"):
Enter Central volume type (hostpath, pvc): pvc ③
Enter external volume name (default: "stackrox-db"):
Enter external volume size in Gi (default: "100"):
Enter storage class name (optional if you have a default StorageClass configured):
```

① If you want to add a custom TLS certificate, provide the file path for the PEM-encoded certificate. When you specify a custom certificate the interactive installer also prompts you to provide a PEM private key for the custom certificate you are using.

② To use the RHACS portal, you must expose Central by using a route, a load balancer or a node port.

③ If you plan to install Red Hat Advanced Cluster Security for Kubernetes on OpenShift Container Platform with a hostPath volume, you must modify the SELinux policy.

WARNING

On OpenShift Container Platform, for using a hostPath volume, you must modify the SELinux policy to allow access to the directory, which the host and the container share. It is because SELinux blocks directory sharing by default. To modify the SELinux policy, run the following command:

```
$ sudo chcon -Rt svirt_sandbox_file_t <full_volume_path>
```

However, Red Hat does not recommend modifying the SELinux policy, instead use PVC when installing on OpenShift Container Platform.

On completion, the installer creates a folder named central-bundle, which contains the necessary YAML manifests and scripts to deploy Central. In addition, it shows on-screen instructions for the scripts you need to run to deploy additional trusted certificate authorities, Central and Scanner, and the authentication instructions for logging into the RHACS portal along with the autogenerated password if you did not provide one when answering the prompts.

3.3.2.3. Running the Central installation scripts

After you run the interactive installer, you can run the **setup.sh** script to install Central.

Procedure

1. Run the **setup.sh** script to configure image registry access:

```
$ ./central-bundle/central/scripts/setup.sh
```

2. Create the necessary resources:

```
$ oc create -R -f central-bundle/central
```

3. Check the deployment progress:

```
$ oc get pod -n stackrox -w
```

4. After Central is running, find the RHACS portal IP address and open it in your browser. Depending on the exposure method you selected when answering the prompts, use one of the following methods to get the IP address.

Exposure method	Command	Address	Example
Route	**oc -n stackrox get route central**	The address under the **HOST/PORT** column in the output	**https://central-stackrox.example.route**

Exposure method	Command	Address	Example
Node Port	**oc get node -owide && oc -n stackrox get svc central-loadbalancer**	IP or hostname of any node, on the port shown for the service	**https://198.51.100.0:31489**
Load Balancer	**oc -n stackrox get svc central-loadbalancer**	EXTERNAL-IP or hostname shown for the service, on port 443	**https://192.0.2.0**
None	**central-bundle/central/scripts/port-forward.sh 8443**	**https://localhost:8443**	**https://localhost:8443**

NOTE

If you have selected autogenerated password during the interactive install, you can run the following command to see it for logging into Central:

```
$ cat central-bundle/password
```

3.4. GENERATING AND APPLYING AN INIT BUNDLE FOR RHACS ON OTHER PLATFORMS

Before you install the **SecuredCluster** resource on a cluster, you must create an init bundle. The cluster that has **SecuredCluster** installed and configured then uses this bundle to authenticate with Central. You can create an init bundle by using either the RHACS portal or the **roxctl** CLI. You then apply the init bundle by using it to create resources.

NOTE

You must have the **Admin** user role to create an init bundle.

3.4.1. Generating an init bundle

3.4.1.1. Generating an init bundle by using the RHACS portal

You can create an init bundle containing secrets by using the RHACS portal.

NOTE

You must have the **Admin** user role to create an init bundle.

Procedure

1. Find the address of the RHACS portal based on your exposure method:

a. For a route:

```
$ oc get route central -n stackrox
```

b. For a load balancer:

```
$ oc get service central-loadbalancer -n stackrox
```

c. For port forward:

i. Run the following command:

```
$ oc port-forward svc/central 18443:443 -n stackrox
```

ii. Navigate to **https://localhost:18443/**.

2. On the RHACS portal, navigate to **Platform Configuration → Integrations**.

3. Navigate to the **Authentication Tokens** section and click on **Cluster Init Bundle**.

4. Click **Generate bundle**.

5. Enter a name for the cluster init bundle and click **Generate**.

   a. If you are installing using Helm charts, click **Download Helm Values File** to download the generated bundle.

   b. If you are installing using the Operator, click **Download Kubernetes Secret File** to download the generated bundle.

IMPORTANT

Store this bundle securely because it contains secrets. You can use the same bundle to create multiple secured clusters.

Next steps

1. Apply the init bundle by creating a resource on the secured cluster.

2. Install secured cluster services on each cluster.

3.4.1.2. Generating an init bundle by using the roxctl CLI

You can create an init bundle with secrets by using the **roxctl** CLI.

NOTE

You must have the **Admin** user role to create init bundles.

Prerequisites

You have configured the **ROX_API_TOKEN** and the **ROX_CENTRAL_ADDRESS** environment variables.

- Set the **ROX_API_TOKEN** and the **ROX_CENTRAL_ADDRESS** environment variables:

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

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

Procedure

- Run the following command to generate a cluster init bundle containing secrets:
  For Helm installations:

```
$ roxctl -e "$ROX_CENTRAL_ADDRESS" \
  central init-bundles generate <cluster_init_bundle_name> \
  --output cluster_init_bundle.yaml
```

For Operator installations:

```
$ roxctl -e "$ROX_CENTRAL_ADDRESS" \
  central init-bundles generate <cluster_init_bundle_name> \
  --output-secrets cluster_init_bundle.yaml
```

> **IMPORTANT**
>
> Ensure that you store this bundle securely because it contains secrets. You can use the same bundle to set up multiple secured clusters.

Next Step

- Use the Red Hat OpenShift CLI to create resources using the init bundle.

3.4.1.3. Creating resources by using the init bundle

Before you install secured clusters, you must use the init bundle to create the required resources on the cluster that will allow the services on the secured clusters to communicate with Central.

> **NOTE**
>
> If you are installing by using Helm charts, do not perform this step. Complete the installation by using Helm; See "Installing RHACS on secured clusters by using Helm charts" in the additional resources section.

Prerequisites

- You must have generated an init bundle containing secrets.

Procedure

To create resources, perform one of the following steps:

- In the OpenShift Container Platform web console, in the top menu, click **+** to open the **Import YAML** page. You can drag the init bundle file or copy and paste its contents into the editor, and then click **Create**.

- Using the Red Hat OpenShift CLI, run the following command to create the resources:

```
$ oc create -f <init_bundle>.yaml \ ❶
  -n <stackrox> ❷
```

❶ Specify the file name of the init bundle containing the secrets.

❷ Specify the name of the project where Central services are installed.

- Using the **kubectl** CLI, run the following commands to create the resources:

```
$ kubectl create namespace stackrox ❶
$ kubectl create -f <init_bundle>.yaml \ ❷
  -n <stackrox> ❸
```

❶ Create the project where secured cluster resources will be installed. This example uses **stackrox**.

❷ Specify the file name of the init bundle containing the secrets.

❸ Specify the project name that you created. This example uses **stackrox**.

Next Step

- Install RHACS secured cluster services in all clusters that you want to monitor.

3.5. INSTALLING SECURED CLUSTER SERVICES FOR RHACS ON OTHER PLATFORMS

You can install RHACS on your secured clusters for platforms such as Amazon Elastic Kubernetes Service (Amazon EKS), Google Kubernetes Engine (Google GKE), and Microsoft Azure Kubernetes Service (Microsoft AKS).

3.5.1. Installing RHACS on secured clusters by using Helm charts

You can install RHACS on secured clusters by using Helm charts with no customization, using the default values, or with customizations of configuration parameters.

3.5.1.1. Installing RHACS on secured clusters by using Helm charts without customizations

3.5.1.1.1. Adding the Helm chart repository

Procedure

- Add the RHACS charts repository.

```
$ helm repo add rhacs https://mirror.openshift.com/pub/rhacs/charts/
```

The Helm repository for Red Hat Advanced Cluster Security for Kubernetes includes Helm charts for installing different components, including:

- Central services Helm chart (**central-services**) for installing the centralized components (Central and Scanner).

  NOTE

  You deploy centralized components only once and you can monitor multiple separate clusters by using the same installation.

- Secured Cluster Services Helm chart (**secured-cluster-services**) for installing the per-cluster (Sensor and Admission controller) and per-node (Collector) components.

  NOTE

  Deploy the per-cluster components into each cluster that you want to monitor and deploy the per-node components in all nodes that you want to monitor.

Verification

- Run the following command to verify the added chart repository:

```
$ helm search repo -l rhacs/
```

3.5.1.1.2. Installing the secured-cluster-services Helm chart without customization

Use the following instructions to install the **secured-cluster-services** Helm chart to deploy the per-cluster and per-node components (Sensor, Admission controller, and Collector).

CAUTION

To install Collector on systems that have Unified Extensible Firmware Interface (UEFI) and that have Secure Boot enabled, you must use eBPF probes because kernel modules are unsigned, and the UEFI firmware cannot load unsigned packages. Collector identifies Secure Boot status at the start and switches to eBPF probes if required.

Prerequisites

- You must have generated RHACS init bundle for your cluster.
- You must have the address and the port number that you are exposing the Central service on.

Procedure

- Run the following command on your Kubernetes based clusters:

```
$ helm install -n stackrox --create-namespace \
    stackrox-secured-cluster-services rhacs/secured-cluster-services \
    -f <path_to_cluster_init_bundle.yaml> \ 1
    --set clusterName=<name_of_the_secured_cluster> \
    --set centralEndpoint=<endpoint_of_central_service> 2
```

1. Use the **-f** option to specify the path for the init bundle.
2. Specify the address and port number for Central. For example, **acs.domain.com:443**.

- Run the following command on OpenShift Container Platform clusters:

```
$ helm install -n stackrox --create-namespace \
    stackrox-secured-cluster-services rhacs/secured-cluster-services \
    -f <path_to_cluster_init_bundle.yaml> \ 1
    --set clusterName=<name_of_the_secured_cluster> \
    --set centralEndpoint=<endpoint_of_central_service> 2
    --set scanner.disable=false
```

1 Use the **-f** option to specify the path for the init bundle.

2 Specify the address and port number for Central. For example, **acs.domain.com:443**.

Additional resources

- Generating and applying an init bundle for RHACS on other platforms

3.5.1.2. Configuring the secured-cluster-services Helm chart with customizations

This section describes Helm chart configuration parameters that you can use with the **helm install** and **helm upgrade** commands. You can specify these parameters by using the **--set** option or by creating YAML configuration files.

Create the following files for configuring the Helm chart for installing Red Hat Advanced Cluster Security for Kubernetes:

- Public configuration file **values-public.yaml**: Use this file to save all non-sensitive configuration options.

- Private configuration file **values-private.yaml**: Use this file to save all sensitive configuration options. Ensure that you store this file securely.

IMPORTANT

While using the **secured-cluster-services** Helm chart, do not modify the **values.yaml** file that is part of the chart.

3.5.1.2.1. Configuration parameters

Parameter	Description
clusterName	Name of your cluster.
centralEndpoint	Address, including port number, of the Central endpoint. If you are using a non-gRPC capable load balancer, use the WebSocket protocol by prefixing the endpoint address with **wss://**. When configuring multiple clusters, use the hostname for the address (for example, **central.example.com:443**).

Parameter	Description
sensor.endpoint	Address of the Sensor endpoint including port number.
sensor.imagePullPolicy	Image pull policy for the Sensor container.
sensor.serviceTLS.cert	The internal service-to-service TLS certificate that Sensor uses.
sensor.serviceTLS.key	The internal service-to-service TLS certificate key that Sensor uses.
sensor.resources.requests.memory	The memory request for the Sensor container. Use this parameter to override the default value.
sensor.resources.requests.cpu	The CPU request for the Sensor container. Use this parameter to override the default value.
sensor.resources.limits.memory	The memory limit for the Sensor container. Use this parameter to override the default value.
sensor.resources.limits.cpu	The CPU limit for the Sensor container. Use this parameter to override the default value.
sensor.nodeSelector	Specify a node selector label as **label-key: label-value** to force Sensor to only schedule on nodes with the specified label.
sensor.tolerations	If the node selector selects tainted nodes, use this parameter to specify a taint toleration key, value, and effect for Sensor. This parameter is mainly used for infrastructure nodes.
image.main.name	The name of the **main** image.
image.collector.name	The name of the Collector image.
image.main.registry	Address of the registry you are using for the main image.
image.collector.registry	Address of the registry you are using for the Collector image.
image.main.pullPolicy	Image pull policy for **main** images.
image.collector.pullPolicy	Image pull policy for the Collector images.
image.main.tag	Tag of **main** image to use.

Parameter	Description
image.collector.tag	Tag of **collector** image to use.
collector.collectionMethod	Either **EBPF**, **KERNEL_MODULE**, or **NO_COLLECTION**.
collector.imagePullPolicy	Image pull policy for the Collector container.
collector.complianceImagePullPolicy	Image pull policy for the Compliance container.
collector.disableTaintTolerations	If you specify **false**, tolerations are applied to Collector, and the collector pods can schedule onto all nodes with taints. If you specify it as **true**, no tolerations are applied, and the collector pods are not scheduled onto nodes with taints.
collector.resources.requests.memory	The memory request for the Collector container. Use this parameter to override the default value.
collector.resources.requests.cpu	The CPU request for the Collector container. Use this parameter to override the default value.
collector.resources.limits.memory	The memory limit for the Collector container. Use this parameter to override the default value.
collector.resources.limits.cpu	The CPU limit for the Collector container. Use this parameter to override the default value.
collector.complianceResources.requests.memory	The memory request for the Compliance container. Use this parameter to override the default value.
collector.complianceResources.requests.cpu	The CPU request for the Compliance container. Use this parameter to override the default value.
collector.complianceResources.limits.memory	The memory limit for the Compliance container. Use this parameter to override the default value.
collector.complianceResources.limits.cpu	The CPU limit for the Compliance container. Use this parameter to override the default value.
collector.serviceTLS.cert	The internal service-to-service TLS certificate that Collector uses.
collector.serviceTLS.key	The internal service-to-service TLS certificate key that Collector uses.

Parameter	Description
admissionControl.listenOnCreates	This setting controls whether Kubernetes is configured to contact Red Hat Advanced Cluster Security for Kubernetes with **AdmissionReview** requests for workload creation events.
admissionControl.listenOnUpdates	When you set this parameter as **false**, Red Hat Advanced Cluster Security for Kubernetes creates the **ValidatingWebhookConfiguration** in a way that causes the Kubernetes API server not to send object update events. Since the volume of object updates is usually higher than the object creates, leaving this as **false** limits the load on the admission control service and decreases the chances of a malfunctioning admission control service.
admissionControl.listenOnEvents	This setting controls whether the cluster is configured to contact Red Hat Advanced Cluster Security for Kubernetes with **AdmissionReview** requests for Kubernetes **exec** and **portforward** events. Red Hat Advanced Cluster Security for Kubernetes does not support this feature on OpenShift Container Platform 3.11. For more information, see Red Hat Advanced Cluster Security for Kubernetes Support Policy.
admissionControl.dynamic.enforceOnCreates	This setting controls whether Red Hat Advanced Cluster Security for Kubernetes evaluates policies; if it is disabled, all AdmissionReview requests are automatically accepted.
admissionControl.dynamic.enforceOnUpdates	This setting controls the behavior of the admission control service. You must specify **listenOnUpdates** as **true** for this to work.
admissionControl.dynamic.scanInline	If you set this option to **true**, the admission control service requests an image scan before making an admission decision. Since image scans take several seconds, enable this option only if you can ensure that all images used in your cluster are scanned before deployment (for example, by a CI integration during image build). This option corresponds to the **Contact image scanners** option in the RHACS Portal.
admissionControl.dynamic.disableBypass	Set it to **true** to disable bypassing the Admission controller.

Parameter	Description
admissionControl.dynamic.timeout	The maximum time, in seconds, Red Hat Advanced Cluster Security for Kubernetes should wait while evaluating admission review requests. Use this to set request timeouts when you enable image scanning. If the image scan runs longer than the specified time, Red Hat Advanced Cluster Security for Kubernetes accepts the request.
admissionControl.resources.requests.memory	The memory request for the Admission Control container. Use this parameter to override the default value.
admissionControl.resources.requests.cpu	The CPU request for the Admission Control container. Use this parameter to override the default value.
admissionControl.resources.limits.memory	The memory limit for the Admission Control container. Use this parameter to override the default value.
admissionControl.resources.limits.cpu	The CPU limit for the Admission Control container. Use this parameter to override the default value.
admissionControl.nodeSelector	Specify a node selector label as **label-key: label-value** to force Admission Control to only schedule on nodes with the specified label.
admissionControl.tolerations	If the node selector selects tainted nodes, use this parameter to specify a taint toleration key, value, and effect for Admission Control. This parameter is mainly used for infrastructure nodes.
admissionControl.serviceTLS.cert	The internal service-to-service TLS certificate that Admission Control uses.
admissionControl.serviceTLS.key	The internal service-to-service TLS certificate key that Admission Control uses.
registryOverride	Use this parameter to override the default **docker.io** registry. Specify the name of your registry if you are using some other registry.
collector.disableTaintTolerations	If you specify **false**, tolerations are applied to Collector, and the Collector pods can schedule onto all nodes with taints. If you specify it as **true**, no tolerations are applied, and the Collector pods are not scheduled onto nodes with taints.

Parameter	Description
createUpgraderServiceAccount	Specify **true** to create the **sensor-upgrader** account. By default, Red Hat Advanced Cluster Security for Kubernetes creates a service account called **sensor-upgrader** in each secured cluster. This account is highly privileged but is only used during upgrades. If you do not create this account, you must complete future upgrades manually if the Sensor does not have enough permissions.
createSecrets	Specify **false** to skip the orchestrator secret creation for the Sensor, Collector, and Admission controller.
collector.slimMode	Specify **true** if you want to use a slim Collector image for deploying Collector. Using slim Collector images requires Central to provide the matching eBPF probe or kernel module. If you are running Red Hat Advanced Cluster Security for Kubernetes in offline mode, you must download a kernel support package from stackrox.io and upload it to Central for slim Collectors to function. Otherwise, you must ensure that Central can access the online probe repository hosted at https://collector-modules.stackrox.io/.
sensor.resources	Resource specification for Sensor.
admissionControl.resources	Resource specification for Admission controller.
collector.resources	Resource specification for Collector.
collector.complianceResources	Resource specification for Collector’s Compliance container.
exposeMonitoring	If you set this option to **true**, Red Hat Advanced Cluster Security for Kubernetes exposes Prometheus metrics endpoints on port number 9090 for the Sensor, Collector, and the Admission controller.
auditLogs.disableCollection	If you set this option to **true**, Red Hat Advanced Cluster Security for Kubernetes disables the audit log detection features used to detect access and modifications to configuration maps and secrets.
scanner.disable	If you set this option to **false**, Red Hat Advanced Cluster Security for Kubernetes deploys a lightweight scanner and Scanner DB in the secured cluster to allow scanning images on OpenShift Container Registry. Enabling Scanner is only supported on OpenShift. Defaults to **true**

Parameter	Description
scanner.dbTolerations	If the node selector selects tainted nodes, use this parameter to specify a taint toleration key, value, and effect for Scanner DB.
scanner.replicas	Resource specification for Collector's Compliance container.
scanner.logLevel	Setting this parameter allows you to modify the scanner log level. Use this option only for troubleshooting purposes.
scanner.autoscaling.disable	If you set this option to **true**, Red Hat Advanced Cluster Security for Kubernetes disables autoscaling on the Scanner deployment.
scanner.autoscaling.minReplicas	The minimum number of replicas for autoscaling. Defaults to 2.
scanner.autoscaling.maxReplicas	The maximum number of replicas for autoscaling. Defaults to 5.
scanner.nodeSelector	Specify a node selector label as **label-key: label-value** to force Scanner to only schedule on nodes with the specified label.
scanner.tolerations	If the node selector selects tainted nodes, use this parameter to specify a taint toleration key, value, and effect for Scanner.
scanner.dbNodeSelector	Specify a node selector label as **label-key: label-value** to force Scanner DB to only schedule on nodes with the specified label.
scanner.dbTolerations	If the node selector selects tainted nodes, use this parameter to specify a taint toleration key, value, and effect for Scanner DB.
scanner.resources.requests.memory	The memory request for the Scanner container. Use this parameter to override the default value.
scanner.resources.requests.cpu	The CPU request for the Scanner container. Use this parameter to override the default value.
scanner.resources.limits.memory	The memory limit for the Scanner container. Use this parameter to override the default value.
scanner.resources.limits.cpu	The CPU limit for the Scanner container. Use this parameter to override the default value.

Parameter	Description
scanner.dbResources.requests.memory	The memory request for the Scanner DB container. Use this parameter to override the default value.
scanner.dbResources.requests.cpu	The CPU request for the Scanner DB container. Use this parameter to override the default value.
scanner.dbResources.limits.memory	The memory limit for the Scanner DB container. Use this parameter to override the default value.
scanner.dbResources.limits.cpu	The CPU limit for the Scanner DB container. Use this parameter to override the default value.

3.5.1.2.1.1. Environment variables

You can specify environment variables for Sensor and Admission controller in the following format:

```
customize:
  envVars:
    ENV_VAR1: "value1"
    ENV_VAR2: "value2"
```

The **customize** setting allows you to specify custom Kubernetes metadata (labels and annotations) for all objects created by this Helm chart and additional pod labels, pod annotations, and container environment variables for workloads.

The configuration is hierarchical, in the sense that metadata defined at a more generic scope (for example, for all objects) can be overridden by metadata defined at a narrower scope (for example, only for the Sensor deployment).

3.5.1.2.2. Installing the secured-cluster-services Helm chart

After you configure the **values-public.yaml** and **values-private.yaml** files, install the **secured-cluster-services** Helm chart to deploy the per-cluster and per-node components (Sensor, Admission controller, and Collector).

CAUTION

To install Collector on systems that have Unified Extensible Firmware Interface (UEFI) and that have Secure Boot enabled, you must use eBPF probes because kernel modules are unsigned, and the UEFI firmware cannot load unsigned packages. Collector identifies Secure Boot status at the start and switches to eBPF probes if required.

Prerequisites

- You must have generated RHACS init bundle for your cluster.

- You must have the address and the port number that you are exposing the Central service on.

Procedure

- Run the following command:

```
$ helm install -n stackrox --create-namespace \
  stackrox-secured-cluster-services rhacs/secured-cluster-services \
  -f <name_of_cluster_init_bundle.yaml> \
  -f <path_to_values_public.yaml> -f <path_to_values_private.yaml> 1
```

1 Use the **-f** option to specify the paths for your YAML configuration files.

NOTE

To deploy **secured-cluster-services** Helm chart by using a continuous integration (CI) system, pass the init bundle YAML file as an environment variable to the **helm install** command:

```
$ helm install ... -f <(echo "$INIT_BUNDLE_YAML_SECRET") 1
```

1 If you are using base64 encoded variables, use the **helm install … -f <(echo "$INIT_BUNDLE_YAML_SECRET" | base64 --decode)** command instead.

Additional resources

- Generating and applying an init bundle for RHACS on other platforms

3.5.1.3. Changing configuration options after deploying the secured-cluster-services Helm chart

You can make changes to any configuration options after you have deployed the **secured-cluster-services** Helm chart.

Procedure

1. Update the **values-public.yaml** and **values-private.yaml** configuration files with new values.

2. Run the **helm upgrade** command and specify the configuration files using the **-f** option:

```
$ helm upgrade -n stackrox \
  stackrox-secured-cluster-services rhacs/secured-cluster-services \
  --reuse-values \ 1
  -f <path_to_values_public.yaml> \
  -f <path_to_values_private.yaml>
```

1 You must specify the **--reuse-values** parameter, otherwise the Helm upgrade command resets all previously configured settings.

NOTE

You can also specify configuration values using the **--set** or **--set-file** parameters. However, these options are not saved, and it requires you to manually specify all the options again whenever you make changes.

3.5.2. Installing RHACS on secured clusters by using the roxctl CLI

To install RHACS on secured clusters by using the CLI, perform the following steps:

1. Install the **roxctl** CLI
2. Install Sensor.

3.5.2.1. Installing the roxctl CLI

You must first download the binary. You can install **roxctl** on Linux, Windows, or macOS.

3.5.2.1.1. Installing the roxctl CLI on Linux

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

Procedure

1. Download the latest version of the **roxctl** CLI:

```
$ curl -O https://mirror.openshift.com/pub/rhacs/assets/4.0.5/bin/Linux/roxctl
```

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

```
$ chmod +x roxctl
```

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

```
$ echo $PATH
```

Verification

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

```
$ roxctl version
```

3.5.2.1.2. Installing the roxctl CLI on macOS

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

Procedure

1. Download the latest version of the **roxctl** CLI:

```
$ curl -O https://mirror.openshift.com/pub/rhacs/assets/4.0.5/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
  ```

3.5.2.1.3. Installing the roxctl CLI on Windows

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

Procedure

- Download the latest version of the **roxctl** CLI:

  ```
  $ curl -O https://mirror.openshift.com/pub/rhacs/assets/4.0.5/bin/Windows/roxctl.exe
  ```

Verification

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

  ```
  $ roxctl version
  ```

3.5.2.2. Installing Sensor

To monitor a cluster, you must deploy Sensor. You must deploy Sensor into each cluster that you want to monitor. The following steps describe adding Sensor by using the RHACS portal.

Prerequisites

- You must have already installed Central services, or you can access Central services by selecting your **ACS instance** on Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service).

Procedure

1. On your secured cluster, in the RHACS portal, navigate to **Platform Configuration → Clusters**.
2. Select **+ New Cluster**.
3. Specify a name for the cluster.
4. Provide appropriate values for the fields based on where you are deploying the Sensor.
   - If you are deploying Sensor in the same cluster, accept the default values for all the fields.

- If you are deploying into a different cluster, replace **central.stackrox.svc:443** with a load balancer, node port, or other address, including the port number, that is accessible from the other cluster.
- If you are using a non-gRPC capable load balancer, such as HAProxy, AWS Application Load Balancer (ALB), or AWS Elastic Load Balancing (ELB), use the WebSocket Secure (**wss**) protocol. To use **wss**:
  - Prefix the address with **wss:**//.
  - Add the port number after the address, for example, **wss://stackrox-central.example.com:443**.

5. Click **Next** to continue with the Sensor setup.
6. Click **Download YAML File and Keys** to download the cluster bundle (zip archive).

IMPORTANT

The cluster bundle zip archive includes unique configurations and keys for each cluster. Do not reuse the same files in another cluster.

7. From a system that has access to the monitored cluster, unzip and run the **sensor** script from the cluster bundle:

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

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

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

After Sensor is deployed, it contacts Central and provides cluster information.

Verification

1. Return to the RHACS portal and check if the deployment is successful. If successful, when viewing your list of clusters in **Platform Configuration → Clusters**, the cluster status displays a green checkmark and a **Healthy** status. If you do not see a green checkmark, use the following command to check for problems:
   - On Kubernetes, enter the following command:

     ```
     $ kubectl get pod -n stackrox -w
     ```

2. Click **Finish** to close the window.

After installation, Sensor starts reporting security information to RHACS and the RHACS portal dashboard begins showing deployments, images, and policy violations from the cluster on which you have installed the Sensor.

3.6. VERIFYING INSTALLATION OF RHACS ON OTHER PLATFORMS

Provides steps to verify that RHACS is properly installed.

3.6.1. Verifying installation

After you complete the installation, run a few vulnerable applications and navigate to the RHACS portal to evaluate the results of security assessments and policy violations.

> **NOTE**
>
> The sample applications listed in the following section contain critical vulnerabilities and they are specifically designed to verify the build and deploy-time assessment features of Red Hat Advanced Cluster Security for Kubernetes.

To verify installation:

1. Find the address of the RHACS portal based on your exposure method:
   a. For a load balancer:

      ```
      $ kubectl get service central-loadbalancer -n stackrox
      ```

   b. For port forward:
      i. Run the following command:

         ```
         $ kubectl port-forward svc/central 18443:443 -n stackrox
         ```

      ii. Navigate to **https://localhost:18443/**.
2. Create a new namespace:

   ```
   $ kubectl create namespace test
   ```

3. Start some applications with critical vulnerabilities:

   ```
   $ kubectl run shell --labels=app=shellshock,team=test-team \
     --image=vulnerables/cve-2014-6271 -n test
   $ kubectl run samba --labels=app=rce \
     --image=vulnerables/cve-2017-7494 -n test
   ```

Red Hat Advanced Cluster Security for Kubernetes automatically scans these deployments for security risks and policy violations as soon as they are submitted to the cluster. Navigate to the RHACS portal to view the violations. You can log in to the RHACS portal by using the default username **admin** and the generated password.

CHAPTER 4. UNINSTALLING RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES

When you install Red Hat Advanced Cluster Security for Kubernetes, it creates:

- A namespace called **rhacs-operator** where the Operator is installed, if you chose the Operator method of installation
- A namespace called **stackrox**, or another namespace where you created the Central and SecuredCluster custom resources
- **PodSecurityPolicy** and Kubernetes role-based access control (RBAC) objects for all components
- Additional labels on namespaces, for use in generated network policies
- An application custom resource definition (CRD), if it does not exist

Uninstalling Red Hat Advanced Cluster Security for Kubernetes involves deleting all of these items.

4.1. DELETING NAMESPACE

You can delete the namespace that Red Hat Advanced Cluster Security for Kubernetes creates by using the OpenShift Container Platform or Kubernetes command-line interface.

Procedure

- Delete the **stackrox** namespace:
  - On OpenShift Container Platform:

    ```
    $ oc delete namespace stackrox
    ```

  - On Kubernetes:

    ```
    $ kubectl delete namespace stackrox
    ```

NOTE

If you installed RHACS in a different namespace, use the name of that namespace in the **delete** command.

4.2. DELETING GLOBAL RESOURCES

You can delete the global resources that Red Hat Advanced Cluster Security for Kubernetes creates, by using the OpenShift Container Platform or Kubernetes command-line interface.

Procedure

- Delete global resources:
  - On OpenShift Container Platform:

```
$ oc get clusterrole,clusterrolebinding,role,rolebinding,psp -o name | grep stackrox |
xargs oc delete --wait
```

```
$ oc delete scc -l "app.kubernetes.io/name=stackrox"
```

```
$ oc delete ValidatingWebhookConfiguration stackrox
```

- On Kubernetes:

```
$ kubectl get clusterrole,clusterrolebinding,role,rolebinding,psp -o name | grep stackrox |
xargs kubectl delete --wait
```

```
$ kubectl delete ValidatingWebhookConfiguration stackrox
```

4.3. DELETING LABELS AND ANNOTATIONS

You can delete the labels and annotations that Red Hat Advanced Cluster Security for Kubernetes creates, by using the OpenShift Container Platform or Kubernetes command-line interface.

Procedure

- Delete labels and annotations:
  - On OpenShift Container Platform:

```
$ for namespace in $(oc get ns | tail -n +2 | awk '{print $1}'); do     oc label namespace
$namespace namespace.metadata.stackrox.io/id-;     oc label namespace $namespace
namespace.metadata.stackrox.io/name-;     oc annotate namespace $namespace
modified-by.stackrox.io/namespace-label-patcher-;   done
```

  - On Kubernetes:

```
$ for namespace in $(kubectl get ns | tail -n +2 | awk '{print $1}'); do     kubectl label
namespace $namespace namespace.metadata.stackrox.io/id-;     kubectl label
namespace $namespace namespace.metadata.stackrox.io/name-;     kubectl annotate
namespace $namespace modified-by.stackrox.io/namespace-label-patcher-;   done
```