Red Hat Advanced Cluster Security for Kubernetes 3.72

Integrating

Integrating Red Hat Advanced Cluster Security for Kubernetes

Last Updated: 2023-05-02

Red Hat Advanced Cluster Security for Kubernetes 3.72 Integrating

Integrating Red Hat Advanced Cluster Security for Kubernetes

Legal Notice

Copyright © 2023 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

Abstract

This document describes how to configure common integrations in Red Hat Advanced Cluster Security for Kubernetes, including integrations with image registries, Slack, PagerDuty, JIRA, email, and by using generic webhooks.

Table of Contents

CHAPTER 1. INTEGRATING WITH IMAGE REGISTRIES 5
1.1. AUTOMATIC CONFIGURATION 5
1.2. AMAZON ECR INTEGRATIONS 6
1.3. MANUALLY CONFIGURING IMAGE REGISTRIES 6
1.3.1. Manually configuring OpenShift Container Platform registry 6
1.3.2. Manually configuring Amazon Elastic Container Registry 7
1.3.2.1. Using assumerole with Amazon ECR 8
1.3.2.1.1. Configuring AssumeRole with container IAM 8
1.3.2.1.2. Configuring AssumeRole without container IAM 9
1.3.2.1.3. Configuring AssumeRole in RHACS 10
1.3.3. Manually configuring Google Container Registry 11
1.3.4. Manually configuring Google Artifact Registry 12
1.3.5. Manually configuring Microsoft Azure Container Registry 12
1.3.6. Manually configuring JFrog Artifactory 13
1.3.7. Manually configuring Quay Container Registry 14
1.4. ADDITIONAL RESOURCES 15
1.4.1. Manually configuring IBM Cloud Container Registry 15
1.4.2. Manually configuring Red Hat Container Registry 15

CHAPTER 2. INTEGRATING WITH CI SYSTEMS 17
2.1. CONFIGURING BUILD POLICIES 17
2.1.1. Checking existing build-phase policies 17
2.1.2. Creating a new system policy 17
2.2. CONFIGURING REGISTRY INTEGRATION 19
2.2.1. Checking for existing registry integration 19
2.2.1.1. Additional resources 20
2.3. CONFIGURING ACCESS 20
2.3.1. Authenticating using the roxctl CLI 20
2.3.2. Installing the roxctl CLI by downloading the binary 21
2.3.2.1. Installing the roxctl CLI on Linux 21
2.3.2.2. Installing the roxctl CLI on macOS 21
2.3.2.3. Installing the roxctl CLI on Windows 22
2.3.3. Running the roxctl CLI from a container 22
2.4. INTEGRATING WITH YOUR CI PIPELINE 23
2.4.1. Using Jenkins 23
2.4.2. Using CircleCI 23

CHAPTER 3. INTEGRATING WITH PAGERDUTY 26
3.1. CONFIGURING PAGERDUTY 26
3.2. CONFIGURING RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES 26
3.3. CONFIGURING POLICY NOTIFICATIONS 27

CHAPTER 4. INTEGRATING WITH SLACK 28
4.1. CONFIGURING SLACK 28
4.1.1. Sending alerts to different Slack channels 28
4.2. CONFIGURING RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES 29
4.3. CONFIGURING POLICY NOTIFICATIONS 29

CHAPTER 5. INTEGRATING BY USING GENERIC WEBHOOKS 31
5.1. CONFIGURING INTEGRATIONS BY USING WEBHOOKS 31
5.2. CONFIGURING POLICY NOTIFICATIONS 32

CHAPTER 6. INTEGRATING WITH QRADAR 34
6.1. CONFIGURING INTEGRATIONS BY USING WEBHOOKS 34
6.2. CONFIGURING POLICY NOTIFICATIONS 35

CHAPTER 7. INTEGRATING WITH SERVICENOW 37
7.1. CONFIGURING INTEGRATIONS BY USING WEBHOOKS 37
7.2. CONFIGURING POLICY NOTIFICATIONS 38

CHAPTER 8. INTEGRATING WITH SUMO LOGIC 39
8.1. CONFIGURING SUMO LOGIC 39
8.2. CONFIGURING RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES 39
8.3. CONFIGURING POLICY NOTIFICATIONS 40
8.4. VIEWING ALERTS IN SUMO LOGIC 40

CHAPTER 9. INTEGRATING WITH GOOGLE CLOUD STORAGE 41
9.1. CONFIGURING RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES 41
9.1.1. Perform on-demand backups on Google Cloud Storage 42
9.1.1.1. Additional resources 42

CHAPTER 10. INTEGRATING BY USING THE SYSLOG PROTOCOL 43
10.1. CONFIGURING SYSLOG INTEGRATION WITH RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES 43

CHAPTER 11. INTEGRATING WITH AMAZON S3 45
11.1. CONFIGURING AMAZON S3 INTEGRATION IN RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES 45
11.2. PERFORMING ON-DEMAND BACKUPS ON AMAZON S3 46
11.3. ADDITIONAL RESOURCES 46

CHAPTER 12. INTEGRATING WITH GOOGLE CLOUD SECURITY COMMAND CENTER 47
12.1. CONFIGURING GOOGLE CLOUD SCC 47
12.2. CONFIGURING RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES FOR INTEGRATING WITH GOOGLE CLOUD SCC 47
12.3. CONFIGURING POLICY NOTIFICATIONS 48

CHAPTER 13. INTEGRATING WITH SPLUNK 49
13.1. USING THE HTTP EVENT COLLECTOR 49
13.1.1. Adding an HTTP event collector in Splunk 49
13.1.1.1. Enabling HTTP event collector 50
13.1.2. Configuring Splunk integration in Red Hat Advanced Cluster Security for Kubernetes 50
13.1.3. Configuring policy notifications 51
13.2. USING THE STACKROX KUBERNETES SECURITY PLATFORM ADD-ON 51
13.2.1. Installing and configuring the Splunk add-on 51

CHAPTER 14. INTEGRATING WITH IMAGE VULNERABILITY SCANNERS 53
14.1. INTEGRATING WITH COREOS CLAIR 53
14.2. INTEGRATING WITH GOOGLE CONTAINER REGISTRY 53
14.3. INTEGRATING WITH QUAY CONTAINER REGISTRY TO SCAN IMAGES 54

CHAPTER 15. INTEGRATING WITH JIRA 56
15.1. CONFIGURING JIRA 56
15.2. CONFIGURING RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES 56
15.2.1. Creating issues in different Jira projects 57
15.2.2. Configuring custom priorities in Jira 57
15.3. CONFIGURING POLICY NOTIFICATIONS 58
15.4. TROUBLESHOOTING JIRA INTEGRATION 59

CHAPTER 16. INTEGRATING WITH EMAIL 60
16.1. CONFIGURING THE EMAIL PLUGIN 60
16.2. CONFIGURING POLICY NOTIFICATIONS 61

CHAPTER 1. INTEGRATING WITH IMAGE REGISTRIES

Red Hat Advanced Cluster Security for Kubernetes (RHACS) integrates with a variety of image registries so that you can understand your images and apply security policies for image usage.

When you integrate with image registries, you can view important image details, such as image creation date and Dockerfile details (including image layers).

After you integrate RHACS with your registry, you can scan images, view image components, and apply security policies to images before or after deployment.

NOTE

When you integrate with an image registry, RHACS does not scan all images in your registry. RHACS only scans the images when you:

- Use the images in deployments
- Use the **roxctl** CLI to check images
- Use a continuous integration (CI) system to enforce security policies

You can integrate RHACS with major image registries, including:

- Amazon Elastic Container Registry (ECR)
- Docker Hub
- Google Container Registry (GCR)
- Google Artifact Registry
- IBM Cloud Container Registry (ICR)
- JFrog Artifactory
- Microsoft Azure Container Registry (ACR)
- Red Hat Quay
- Red Hat container registries
- Sonatype Nexus
- Any other registry that uses the Docker Registry HTTP API

1.1. AUTOMATIC CONFIGURATION

Red Hat Advanced Cluster Security for Kubernetes includes default integrations with standard registries, such as Docker Hub and others. It can also automatically configure integrations based on artifacts found in the monitored clusters, such as image pull secrets. Usually, you do not need to configure registry integrations manually.

IMPORTANT

If you are using a GCR registry, Red Hat Advanced Cluster Security for Kubernetes does not create a registry integration automatically.

1.2. AMAZON ECR INTEGRATIONS

For Amazon ECR integrations, Red Hat Advanced Cluster Security for Kubernetes automatically generates ECR registry integrations if the following conditions are met:

- The cloud provider for the cluster is AWS.
- The nodes in your cluster have an Instance Identity and Access Management (IAM) Role association and the Instance Metadata Service is available in the nodes. For example, when using Amazon Elastic Kubernetes Service (EKS) to manage your cluster, this role is known as the EKS Node IAM role.
- The Instance IAM role has IAM policies granting access to the ECR registries from which you are deploying.

If the listed conditions are met, Red Hat Advanced Cluster Security for Kubernetes monitors deployments that pull from ECR registries and automatically generates ECR integrations for them. You can edit these integrations after they are automatically generated.

1.3. MANUALLY CONFIGURING IMAGE REGISTRIES

If you are using GCR, you must manually create image registry integrations.

1.3.1. Manually configuring OpenShift Container Platform registry

You can integrate Red Hat Advanced Cluster Security for Kubernetes with OpenShift Container Platform built-in container image registry.

Prerequisites

- You need a username and a password for authentication with the OpenShift Container Platform registry.

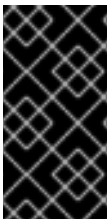
Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Integrations**.
2. Under the **Image Integrations** section, select **Generic Docker Registry**.
3. Click **New integration**.
4. Enter the details for the following fields:
    a. **Integration name**: The name of the integration.
    b. **Endpoint**: The address of the registry.
    c. **Username** and **Password**.

5. If you are not using a TLS certificate when connecting to the registry, select **Disable TLS certificate validation (insecure)**.
6. Select **Create integration without testing**to create the integration without testing the connection to the registry.
7. Select **Test** to test that the integration with the selected registry is working.
8. Select **Save**.

1.3.2. Manually configuring Amazon Elastic Container Registry

You can use Red Hat Advanced Cluster Security for Kubernetes to create and modify Amazon Elastic Container Registry (ECR) integrations manually. If you are deploying from Amazon ECR, integrations for the Amazon ECR registries are usually automatically generated. However, you might want to create integrations on your own to scan images outside deployments. You can also modify the parameters of an automatically-generated integration. For example, you can change the authentication method used by an automatically-generated Amazon ECR integration to use AssumeRole authentication or other authorization models.



IMPORTANT

To erase changes you made to an automatically-generated ECR integration, delete the integration, and Red Hat Advanced Cluster Security for Kubernetes creates a new integration for you with the automatically-generated parameters when you deploy images from Amazon ECR.

Prerequisites

- You must have an Amazon Identity and Access Management (IAM) access key ID and a secret access key. Alternatively, you can use a node-level IAM proxy such as **kiam** or **kube2iam**.
- The access key must have read access to ECR. See How do I create an AWS access key? for more information.
- If you are running Red Hat Advanced Cluster Security for Kubernetes in Amazon Elastic Kubernetes Service (EKS) and want to integrate with an ECR from a separate Amazon account, you must first set a repository policy statement in your ECR. Follow the instructions at Setting a repository policy statement and for **Actions**, choose the following scopes of the Amazon ECR API operations:
  - ecr:BatchCheckLayerAvailability
  - ecr:BatchGetImage
  - ecr:DescribeImages
  - ecr:GetDownloadUrlForLayer
  - ecr:ListImages

Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Integrations**.
2. Under the **Image Integrations** section, select **Amazon ECR**.

3. Click **New integration**, or click one of the automatically-generated integrations to open it, then click **Edit**.

4. Enter or modify the details for the following fields:

   a. **Update stored credentials**: Clear this box if you are modifying an integration without updating the credentials such as access keys and passwords.

   b. **Integration name**: The name of the integration.

   c. **Registry ID**: The ID of the registry.

   d. **Endpoint**: The address of the registry. This field is not enabled when the AssumeRole option is selected.

   e. **Region**: The region for the registry; for example, **us-west-1**.

5. If you are using IAM, select **Use Container IAM role**. Otherwise, clear the **Use Container IAM role** box and enter the **Access key ID** and **Secret access key**.

6. If you are using AssumeRole authentication, select **Use AssumeRole** and enter the details for the following fields:

   a. **AssumeRole ID**: The ID of the role to assume.

   b. **AssumeRole External ID** (optional): If you are using an external ID with AssumeRole, you can enter it here.

7. Select **Create integration without testing** to create the integration without testing the connection to the registry.

8. Select **Test** to test that the integration with the selected registry is working.

9. Select **Save**.

1.3.2.1. Using assumerole with Amazon ECR

You can use AssumeRole to grant access to AWS resources without manually configuring each user's permissions. Instead, you can define a role with the desired permissions so that the user is granted access to assume that role. **AssumeRole** enables you to grant, revoke, or otherwise generally manage more fine-grained permissions.

1.3.2.1.1. Configuring AssumeRole with container IAM

Before you can use AssumeRole with Red Hat Advanced Cluster Security for Kubernetes, you must first configure it.

Procedure

1. Enable the IAM OIDC provider for your EKS cluster:

   ```
   $ eksctl utils associate-iam-oidc-provider --cluster <cluster name> --approve
   ```

2. Create an IAM role for your EKS cluster.

3. Associate the newly created role with a service account:

```
$ kubectl -n stackrox annotate sa central eks.amazonaws.com/role-
arn=arn:aws:iam::67890:role/<role-name>
```

4. Restart Central to apply the changes.

```
$ kubectl -n stackrox delete pod -l app=central
```

5. Assign the role to a policy that allows the role to assume another role as required:

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "VisualEditor0",
         "Effect": "Allow",
         "Action": "sts:AssumeRole",
         "Resource": "arn:aws:iam::<ecr-registry>:role/<assumerole-readonly>" (1)
      }
   ]
}
```

(1) Replace **<assumerole-readonly>** with the role you want to assume.

6. Update the trust relationship for the role you want to assume:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::<ecr-registry>:role/<role-name>" (1)
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

(1) The **<role-name>** should match with the new role you have created earlier.

1.3.2.1.2. Configuring AssumeRole without container IAM

To use AssumeRole without container IAM, you must use an access and a secret key to authenticate as an AWS user with programmatic access.

Procedure

1. Depending on whether the AssumeRole user is in the same account as the ECR registry or in a different account, you must either:

- Create a new role with the desired permissions if the user for which you want to assume role is in the same account as the ECR registry.

NOTE

When creating the role, you can choose any trusted entity as required. However, you must modify it after creation.

- Or, you must provide permissions to access the ECR registry and define its trust relationship if the user is in a different account than the ECR registry:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": "arn:aws:iam::<ecr-registry>:role/<assumerole-readonly>" 1
        }
    ]
}
```

1 Replace **<assumerole-readonly>** with the role you want to assume.

2. Configure the trust relationship of the role by including the user ARN under the **Principal** field:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::<ecr-registry>:user/<role-name>"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

1.3.2.1.3. Configuring AssumeRole in RHACS

After configuring AssumeRole in ECR, you can integrate Red Hat Advanced Cluster Security for Kubernetes with Amazon Elastic Container Registry (ECR) by using AssumeRole.

Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Integrations**.
2. Under the **Image Integrations** section, select **Amazon ECR**.

3. Click **New Integration**.
4. Enter the details for the following fields:
   a. **Integration Name**: The name of the integration.
   b. **Registry ID**: The ID of the registry.
   c. **Region**: The region for the registry; for example, **us-west-1**.
5. If you are using IAM, select **Use container IAM role**. Otherwise, clear the **Use custom IAM role** box and enter the **Access key ID** and **Secret access key**.
6. If you are using AssumeRole, select **Use AssumeRole** and enter the details for the following fields:
   a. **AssumeRole ID**: The ID of the role to assume.
   b. **AssumeRole External ID** (optional): If you are using an external ID with AssumeRole, you can enter it here.
7. Select **Test** to test that the integration with the selected registry is working.
8. Select **Save**.

1.3.3. Manually configuring Google Container Registry

You can integrate Red Hat Advanced Cluster Security for Kubernetes with Google Container Registry (GCR).

Prerequisites

- You must have a service account key.
- The associated service account must have access to the registry. See Configuring access control for information about granting users and other projects access to GCR.
- If you are using GCR Container Analysis, you must also grant the following roles to the service account:
  - Container Analysis Notes Viewer
  - Container Analysis Occurrences Viewer
  - Storage Object Viewer

Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Integrations**.
2. Under the **Image Integrations** section, select **Google Container Registry**.
3. Click **New integration**.
4. Enter the details for the following fields:
   a. **Integration name**: The name of the integration.

b. **Type**: Select **Registry**.

c. **Registry Endpoint**: The address of the registry.

d. **Project**: The Google Cloud project name.

e. **Service account key (JSON)**Your service account key for authentication.

5. Select **Create integration without testing**to create the integration without testing the connection to the registry.

6. Select **Test** to test that the integration with the selected registry is working.

7. Select **Save**.

1.3.4. Manually configuring Google Artifact Registry

You can integrate Red Hat Advanced Cluster Security for Kubernetes with Google Artifact Registry.

Prerequisites

- You need a service account key with the **Artifact Registry Reader** Identity and Access Management (IAM) role **roles/artifactregistry.reader**.

Procedure

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

2. Under the **Image Integrations** section, select **Google Artifact Registry**.

3. Click **New integration**.

4. Enter the details for the following fields:

a. **Integration name**: The name of the integration.

b. **Registry endpoint**: The address of the registry.

c. **Project**: The Google Cloud project name.

d. **Service account key (JSON)**Your service account key for authentication.

5. Select **Create integration without testing**to create the integration without testing the connection to the registry.

6. Select **Test** to test that the integration with the selected registry is working.

7. Select **Save**.

1.3.5. Manually configuring Microsoft Azure Container Registry

You can integrate Red Hat Advanced Cluster Security for Kubernetes with Microsoft Azure Container Registry.

Prerequisites

- You must have a username and a password for authentication.

Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Integrations**.
2. Under the **Image Integrations** section, select **Microsoft Azure Container Registry**.
3. Click **New integration**.
4. Enter the details for the following fields:
   a. **Integration name**: The name of the integration.
   b. **Endpoint**: The address of the registry.
   c. **Username** and **Password**.
5. Select **Create integration without testing** to create the integration without testing the connection to the registry.
6. Select **Test** to test that the integration with the selected registry is working.
7. Select **Save**.

1.3.6. Manually configuring JFrog Artifactory

You can integrate Red Hat Advanced Cluster Security for Kubernetes with JFrog Artifactory.

Prerequisites

- You must have a username and a password for authentication with JFrog Artifactory.

Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Integrations**.
2. Under the **Image Integrations** section, select **JFrog Artifactory**.
3. Click **New integration**.
4. Enter the details for the following fields:
   a. **Integration name**: The name of the integration.
   b. **Endpoint**: The address of the registry.
   c. **Username** and **Password**.
5. If you are not using a TLS certificate when connecting to the registry, select **Disable TLS certificate validation (insecure)**.
6. Select **Create integration without testing** to create the integration without testing the connection to the registry.
7. Select **Test** to test that the integration with the selected registry is working.

8. Select **Save**.

1.3.7. Manually configuring Quay Container Registry

You can integrate Red Hat Advanced Cluster Security for Kubernetes (RHACS) with Quay Container Registry. You can integrate with Quay by using the following methods:

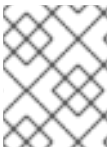
- Integrating with the Quay public repository (registry): This method does not require authentication.
- Integrating with a Quay private registry by using a robot account: This method requires that you create a robot account to use with Quay (recommended). See the Quay documentation for more information.
- Integrating with Quay to use the Quay scanner rather than the RHACS scanner: This method uses the API and requires an OAuth token for authentication. See "Integrating with Quay Container Registry to scan images" in the "Additional Resources" section.

Prerequisites

- For authentication with a Quay private registry, you need the credentials associated with a robot account or an OAuth token (deprecated).

Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Integrations**.
2. Under the **Image Integrations** section, select **Red Hat Quay.io**.
3. Click **New integration**.
4. Enter the **Integration name.**
5. Enter the **Endpoint**, or the address of the registry.
    a. If you are integrating with the Quay public repository, under **Type**, select **Registry**, and then go to the next step.
    b. If you are integrating with a Quay private registry, under **Type**, select **Registry** and enter information in the following fields:
        - **Robot username**: If you are accessing the registry by using a Quay robot account, enter the user name in the format ***<namespace>+<accountname>***.
        - **Robot password**: If you are accessing the registry by using a Quay robot account, enter the password for the robot account user name.
        - **OAuth token**: If you are accessing the registry by using an OAuth token (deprecated), enter it in this field.
6. Optional: If you are not using a TLS certificate when connecting to the registry, select **Disable TLS certificate validation (insecure)**.
7. Optional: To create the integration without testing, select **Create integration without testing**
8. Select **Save**.



NOTE

If you are editing a Quay integration but do not want to update your credentials, verify that **Update stored credentials** is not selected.

1.4. ADDITIONAL RESOURCES

- Integrating with Quay Container Registry to scan images

1.4.1. Manually configuring IBM Cloud Container Registry

You can integrate Red Hat Advanced Cluster Security for Kubernetes with IBM Cloud Container Registry.

Prerequisites

- You must have an API key for authentication with the IBM Cloud Container Registry.

Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Integrations**.
2. Under the **Image Integrations** section, select **IBM Cloud Container Registry**.
3. Click **New integration**.
4. Enter the details for the following fields:
   a. **Integration name**: The name of the integration.
   b. **Endpoint**: The address of the registry.
   c. **API key**.
5. Select **Test** to test that the integration with the selected registry is working.
6. Select **Save**.

1.4.2. Manually configuring Red Hat Container Registry

You can integrate Red Hat Advanced Cluster Security for Kubernetes with Red Hat Container Registry.

Prerequisites

- You must have a username and a password for authentication with the Red Hat Container Registry.

Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Integrations**.
2. Under the **Image Integrations** section, select **Red Hat Registry**.
3. Click **New integration**.

4. Enter the details for the following fields:
    a. **Integration name**: The name of the integration.
    b. **Endpoint**: The address of the registry.
    c. **Username** and **Password**.
5. Select **Create integration without testing**to create the integration without testing the connection to the registry.
6. Select **Test** to test that the integration with the selected registry is working.
7. Select **Save**.

CHAPTER 2. INTEGRATING WITH CI SYSTEMS

Red Hat Advanced Cluster Security for Kubernetes (RHACS) integrates with a variety of continuous integration (CI) products and allows you to apply build-time and deploy-time security rules before you deploy images.

Red Hat Advanced Cluster Security for Kubernetes integrates into CI pipelines after images are built and pushed to a registry. Pushing the image first allows developers to continue testing their artifacts while dealing with any policy violations alongside any other CI test failures, linter violations, or other problems.

If possible, you should configure the version control system to block pull or merge requests from being merged if the build stage, which includes Red Hat Advanced Cluster Security for Kubernetes checks, fails.

The integration with your CI product functions by contacting your Red Hat Advanced Cluster Security for Kubernetes installation to check whether the image complies with build-phase policies you have configured. If there are policy violations, a detailed message is displayed on the console log, including the policy description, rationale, and remediation instructions. Each policy includes an optional enforcement setting; if you mark a policy for build-phase enforcement, failure of that policy causes the client to exit with a nonzero error code.

To integrate Red Hat Advanced Cluster Security for Kubernetes with your CI system, follow these steps:

1. Configure build policies.
2. Configure a registry integration.
3. Configure access to your Red Hat Advanced Cluster Security for Kubernetes instance.
4. Integrate with your CI pipeline.

2.1. CONFIGURING BUILD POLICIES

To check Red Hat Advanced Cluster Security for Kubernetes policies during builds, you must first configure policies that apply to the build phase of the container lifecycle. And then you must integrate with the registry that images are pushed to during the build.

2.1.1. Checking existing build-phase policies

Use the RHACS portal to check any existing build-phase policies that you have configured in Red Hat Advanced Cluster Security for Kubernetes.

Procedure

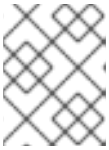
1. On the RHACS portal, navigate to **Platform Configuration → Policies**.
2. Use global search to search for **Lifecycle Stage:Build**.

2.1.2. Creating a new system policy

In addition to using the default policies, you can also create custom policies in Red Hat Advanced Cluster Security for Kubernetes.

Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Policies**.
2. Click **+ New Policy**.
3. Enter the **Name** for the policy.
4. Select a **Severity** level for the policy: Critical, High, Medium, or Low.
5. Choose the **Lifecycle Stages** for which the policy is applicable, from **Build**, **Deploy**, or **Runtime**. You can select more than one stage.

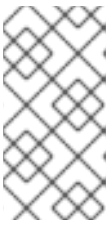


NOTE

If you create a new policy for integrating with a CI system, select **Build** as the lifecycle stage.

   - Build-time policies apply to image fields such as CVEs and Dockerfile instructions.
   - Deploy-time policies can include all build-time policy criteria. They can also have data from your cluster configurations, such as running in privileged mode or mounting the Docker daemon socket.
   - Runtime policies can include all build-time and deploy-time policy criteria, as well as data about process executions during runtime.

6. Enter information about the policy in the **Description**, **Rationale**, and **Remediation** fields. When CI validates the build, the data from these fields is displayed. Therefore, include all information explaining the policy.
7. Select a category from the **Categories** drop-down menu.
8. Select a notifier from the **Notifications** drop-down menu that receives alert notifications when a violation occurs for this policy.



NOTE

You must integrate Red Hat Advanced Cluster Security for Kubernetes with your notification providers, such as webhooks, Jira, or PagerDuty, to receive alert notifications. Notifiers only show up if you have integrated any notification providers with Red Hat Advanced Cluster Security for Kubernetes.

9. Use **Restrict to Scope** to enable this policy only for a specific cluster, namespace, or label. You can add multiple scopes and also use regular expressions in RE2 Syntax for namespaces and labels.
10. Use **Exclude by Scope** to exclude deployments, clusters, namespaces, and labels. This field indicates that the policy will not apply to the entities that you specify. You can add multiple scopes and also use regular expressions in RE2 Syntax for namespaces and labels. However, you cannot use regular expressions for selecting deployments.
11. For **Excluded Images (Build Lifecycle only)**, select all the images from the list for which you do not want to trigger a violation for the policy.

NOTE

The **Excluded Images (Build Lifecycle only)** setting only applies when you check images in a continuous integration system (the Build lifecycle stage). It does not have any effect if you use this policy to check running deployments (the Deploy lifecycle stage) or runtime activities (the Runtime lifecycle stage).

12. In the **Policy Criteria** section, configure the attributes that will trigger the policy.
13. Select **Next** on the panel header.
14. The new policy panel shows a preview of the violations that are triggered if you enable the policy.
15. Select **Next** on the panel header.
16. Choose the enforcement behavior for the policy. Enforcement settings are only available for the stages that you selected for the **Lifecycle Stages** option. Select **ON** to enforce policy and report a violation. Select **OFF** to only report a violation.

NOTE

The enforcement behavior is different for each lifecycle stage.

- For the **Build** stage, Red Hat Advanced Cluster Security for Kubernetes fails your CI builds when images match the conditions of the policy.
- For the **Deploy** stage, Red Hat Advanced Cluster Security for Kubernetes blocks the creation of deployments that match the conditions of the policy. In clusters with admission controller enforcement, the Kubernetes or OpenShift Container Platform API server blocks all noncompliant deployments. In other clusters, Red Hat Advanced Cluster Security for Kubernetes edits noncompliant deployments to prevent pods from being scheduled.
- For the **Runtime** stage, Red Hat Advanced Cluster Security for Kubernetes stops all pods that match the conditions of the policy.

WARNING

Policy enforcement can impact running applications or development processes. Before you enable enforcement options, inform all stakeholders and plan how to respond to the automated enforcement actions.

2.2. CONFIGURING REGISTRY INTEGRATION

To scan images, you must provide Red Hat Advanced Cluster Security for Kubernetes with access to the image registry you are using in your build pipeline.

2.2.1. Checking for existing registry integration

You can use the RHACS portal to check if you have already integrated with a registry.

Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Integrations**.
2. Under the **Image Integration** section, look for highlighted **Registry** tiles. The tiles also list the number of items already configured for that tile.

If none of the Registry tiles are highlighted, you must first integrate with an image registry.

2.2.1.1. Additional resources

- Integrating with image registries

2.3. CONFIGURING ACCESS

Red Hat Advanced Cluster Security for Kubernetes provides a command-line interface (CLI) **roxctl** to make it easy to integrate Red Hat Advanced Cluster Security for Kubernetes policies into your build pipeline. The **roxctl** CLI prints detailed information about problems and how to fix them so that developers can maintain high standards in the early phases of the container lifecycle.

To securely authenticate to the Red Hat Advanced Cluster Security for Kubernetes API server, you must create an API token.

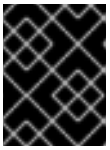
2.3.1. Authenticating using the roxctl CLI

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

Use the following steps to generate an authentication token.

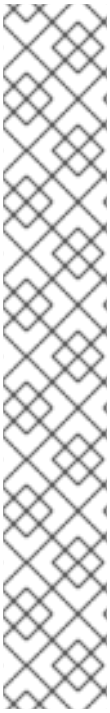
Procedure

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



IMPORTANT

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



NOTE

After you have generated the authentication token, export it as **ROX_API_TOKEN** variable:

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

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

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

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

2.3.2. Installing the roxctl CLI by downloading the binary

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

2.3.2.1. Installing the roxctl CLI on Linux

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

Procedure

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

```
$ curl -O https://mirror.openshift.com/pub/rhacs/assets/3.72.3/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
```

2.3.2.2. Installing the roxctl CLI on macOS

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

Procedure

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

```
$ curl -O https://mirror.openshift.com/pub/rhacs/assets/3.72.3/bin/Darwin/roxctl
```

2. Remove all extended attributes from the binary:

```
$ xattr -c roxctl
```

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

```
$ chmod +x roxctl
```

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

```
$ echo $PATH
```

Verification

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

```
$ roxctl version
```

2.3.2.3. 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/3.72.3/bin/Windows/roxctl.exe
```

Verification

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

```
$ roxctl version
```

2.3.3. Running the roxctl CLI from a container

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

Prerequisites

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

Procedure

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

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

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

```
$ docker pull registry.redhat.io/advanced-cluster-security/rhacs-roxctl-rhel8:3.72.3
```

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

```
$ docker run -e ROX_API_TOKEN=$ROX_API_TOKEN \
  -it registry.redhat.io/advanced-cluster-security/rhacs-roxctl-rhel8:3.72.3 \
  -e $ROX_CENTRAL_ADDRESS <command>
```

Verification

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

```
$ docker run -it registry.redhat.io/advanced-cluster-security/rhacs-roxctl-rhel8:3.72.3 version
```

2.4. INTEGRATING WITH YOUR CI PIPELINE

After you have finished these procedures, the next step is to integrate with your CI pipeline.

Each CI system might require a slightly different configuration.

2.4.1. Using Jenkins

Use the StackRox Container Image Scanner Jenkins plugin for integrating with Jenkins. You can use this plugin in both Jenkins freestyle projects and pipelines.

2.4.2. Using CircleCI

You can integrate Red Hat Advanced Cluster Security for Kubernetes with CircleCI.

Prerequisetes

- You have a token with **read** and **write** permissions for the **Image** resource.
- You have a username and password for your Docker Hub account.

Procedure

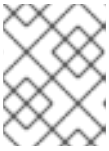
1. Log in to CircleCI and open an existing project or create a new project.
2. Click **Project Settings**.
3. Click **Environment variables**.
4. Click **Add variable** and create the following three environment variables:

- **Name**: **STACKROX_CENTRAL_HOST** - The DNS name or IP address of Central.
- **Name**: **ROX_API_TOKEN** - The API token to access Red Hat Advanced Cluster Security for Kubernetes.
- **Name**: **DOCKERHUB_PASSWORD** - The password for your Docker Hub account.
- **Name**: **DOCKERHUB_USER** - The username for your Docker Hub account.

5. Create a directory called **.circleci** in the root directory of your local code repository for your selected project, if you do not already have a CircleCI configuration file.
6. Create a **config.yml** configuration file with the following lines in the **.circleci** directory:

```
version: 2
jobs:
  check-policy-compliance:
    docker:
      - image: 'circleci/node:latest'
        auth:
          username: $DOCKERHUB_USER
          password: $DOCKERHUB_PASSWORD
    steps:
      - checkout
      - run:
          name: Install roxctl
          command: |
            curl -H "Authorization: Bearer $ROX_API_TOKEN"
https://$STACKROX_CENTRAL_HOST:443/api/cli/download/roxctl-linux -o roxctl && chmod
+x ./roxctl
      - run:
          name: Scan images for policy deviations and vulnerabilities
          command: |
            ./roxctl image check --endpoint "$STACKROX_CENTRAL_HOST:443" --image "
<your_registry/repo/image_name>" 1
      - run:
          name: Scan deployment files for policy deviations
          command: |
            ./roxctl image check --endpoint "$STACKROX_CENTRAL_HOST:443" --image "
<your_deployment_file>" 2
            # Important note: This step assumes the YAML file you'd like to test is located in the
project.
workflows:
  version: 2
  build_and_test:
    jobs:
      - check-policy-compliance
```

1. Replace **<your_registry/repo/image_name>** with your registry and image path.
2. Replace **<your_deployment_file>** with the path to your deployment file.



NOTE

If you already have a **config.yml** file for CircleCI in your repository, add a new jobs section with the specified details in your existing configuration file.

7. After you commit the configuration file to your repository, navigate to the **Jobs** queue in your CircleCI dashboard to verify the build policy enforcement.

CHAPTER 3. INTEGRATING WITH PAGERDUTY

If you are using PagerDuty, you can forward alerts from Red Hat Advanced Cluster Security for Kubernetes to PagerDuty.

The following steps represent a high-level workflow for integrating Red Hat Advanced Cluster Security for Kubernetes with PagerDuty:

1. Add a new API service in PagerDuty and get the integration key.
2. Use the integration key to set up notifications in Red Hat Advanced Cluster Security for Kubernetes.
3. Identify the policies you want to send notifications for, and update the notification settings for those policies.

3.1. CONFIGURING PAGERDUTY

Start integrating with PagerDuty by creating a new service and by getting the integration key.

Procedure

1. Navigate to **Configuration → Services**.
2. Select **Add Services**.
3. Under **General Settings**, specify a **Name** and **Description**.
4. Under **Integration Setting**, click **Use our API Directly** with **Events v2 API** selected for the **Integration Type** drop-down menu.
5. Under **Incident Settings**, select an **Escalation Policy**, and configure notification settings and incident timeouts.
6. Accept default settings for **Incident Behavior** and **Alert Grouping**, or configure them as required.
7. Click **Add Service**.
8. From the **Service Details** page, make note of the **Integration Key**.

3.2. CONFIGURING RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES

Create a new integration in Red Hat Advanced Cluster Security for Kubernetes by using the integration key.

Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Integrations**.
2. Scroll down to the **Notifier Integrations** section and select **PagerDuty**.
3. Click **New Integration** (**add** icon).

4. Enter a name for **Integration Name**.
5. Enter the integration key in the **PagerDuty integration key** field.
6. Click **Test** (**checkmark** icon) to validate that the integration with PagerDuty is working.
7. Click **Create** (**save** icon) to create the configuration.

3.3. CONFIGURING POLICY NOTIFICATIONS

Enable alert notifications for system policies.

Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Policies**.
2. Select one or more policies for which you want to send alerts.
3. Under **Bulk actions**, select **Enable notification**.
4. In the **Enable notification** window, select the PagerDuty notifier.

NOTE

If you have not configured any other integrations, the system displays a message that no notifiers are configured.

5. Click **Enable**.

NOTE

- Red Hat Advanced Cluster Security for Kubernetes sends notifications on an opt-in basis. To receive notifications, you must first assign a notifier to the policy.
- Notifications are only sent once for a given alert. If you have assigned a notifier to a policy, you will not receive a notification unless a violation generates a new alert.
- Red Hat Advanced Cluster Security for Kubernetes creates a new alert for the following scenarios:
  - A policy violation occurs for the first time in a deployment.
  - A runtime-phase policy violation occurs in a deployment after you resolved the previous runtime alert for a policy in that deployment.

CHAPTER 4. INTEGRATING WITH SLACK

If you are using Slack, you can forward alerts from Red Hat Advanced Cluster Security for Kubernetes to Slack.

The following steps represent a high-level workflow for integrating Red Hat Advanced Cluster Security for Kubernetes with Slack:

1. Create a new Slack app, enable incoming webhooks, and get a webhook URL.
2. Use the webhook URL to integrate Slack with Red Hat Advanced Cluster Security for Kubernetes.
3. Identify policies for which you want to send notifications, and update the notification settings for those policies.

4.1. CONFIGURING SLACK

Start by creating a new Slack app, and get the webhook URL.

Prerequisites

1. You need an administrator account or a user account with permissions to create webhooks.

Procedure

1. Create a new Slack app:

NOTE

If you want to use an existing Slack app, go to https://api.slack.com/apps and select an app.

   a. Navigate to https://api.slack.com/apps/new.

   b. Enter the **App Name** and choose a **Development Slack Workspace** to install your app.

   c. Click **Create App**.

2. On the settings page, **Basic Information** section, select **Incoming Webhooks** (under **Add features and functionality**).
3. Turn on the **Activate Incoming Webhooks** toggle.
4. Select **Add New Webhook to Workspace**.
5. Choose a **channel** that the app will post to, and then select **Authorize**. The page refreshes and you are sent back to your app settings page.
6. Copy the webhook URL located in the **Webhook URLs for Your Workspace** section.

For more details, see the Slack documentation topic, Getting started with Incoming Webhooks.

4.1.1. Sending alerts to different Slack channels

You can configure Red Hat Advanced Cluster Security for Kubernetes to send notifications to different Slack channels so that they directly go to the right team.

Procedure

1. After you configure incoming webhooks, add an annotation similar to the following in your deployment YAML file:

```
example.com/slack-webhook:
https://hooks.slack.com/services/T00000000/B00000000/XXXXXXXXXXXXXXXXXXXXXXXX
```

2. Use the annotation key **example.com/slack-webhook** in the **Label/Annotation Key For Slack Webhook** field when you configure Red Hat Advanced Cluster Security for Kubernetes.

After the configuration is complete, if a deployment has the annotation that you configured in the YAML file, Red Hat Advanced Cluster Security for Kubernetes sends the alert to the webhook URL you specified for that annotation. Otherwise, it sends the alert to the default webhook URL.

4.2. CONFIGURING RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES

Create a new integration in Red Hat Advanced Cluster Security for Kubernetes by using the webhook URL.

Procedure

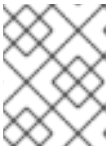
1. On the RHACS portal, navigate to **Platform Configuration → Integrations**.
2. Scroll down to the **Notifier Integrations** section and select **Slack**.
3. Click **New Integration** (**add** icon).
4. Enter a name for **Integration Name**.
5. Enter the generated webhook URL in the **Default Slack Webhook** field.
6. Select **Test** (**checkmark** icon) to test that the integration with Slack is working.
7. Select **Create** (**save** icon) to create the configuration.

4.3. CONFIGURING POLICY NOTIFICATIONS

Enable alert notifications for system policies.

Procedure

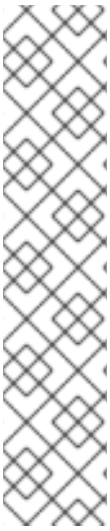
1. On the RHACS portal, navigate to **Platform Configuration → Policies**.
2. Select one or more policies for which you want to send alerts.
3. Under **Bulk actions**, select **Enable notification**.
4. In the **Enable notification** window, select the Slack notifier.



NOTE

If you have not configured any other integrations, the system displays a message that no notifiers are configured.

5. Click **Enable**.



NOTE

- Red Hat Advanced Cluster Security for Kubernetes sends notifications on an opt-in basis. To receive notifications, you must first assign a notifier to the policy.
- Notifications are only sent once for a given alert. If you have assigned a notifier to a policy, you will not receive a notification unless a violation generates a new alert.
- Red Hat Advanced Cluster Security for Kubernetes creates a new alert for the following scenarios:
  - A policy violation occurs for the first time in a deployment.
  - A runtime-phase policy violation occurs in a deployment after you resolved the previous runtime alert for a policy in that deployment.

CHAPTER 5. INTEGRATING BY USING GENERIC WEBHOOKS

With Red Hat Advanced Cluster Security for Kubernetes, you can send alert notifications as JSON messages to any webhook receiver. When a violation occurs, Red Hat Advanced Cluster Security for Kubernetes makes an HTTP POST request on the configured URL. The POST request body includes JSON-formatted information about the alert.

The webhook POST request's JSON data includes a **v1.Alert** object and any custom fields that you configure, as shown in the following example:

```
{
  "alert": {
    "id": "<id>",
    "time": "<timestamp>",
    "policy": {
      "name": "<name>",
      ...
    },
    ...
  },
  "<custom_field_1>": "<custom_value_1>"
}
```

You can create multiple webhooks. For example, you can create one webhook for receiving all audit logs and another webhook for alert notifications.

To forward alerts from Red Hat Advanced Cluster Security for Kubernetes to any webhook receiver:

1. Set up a webhook URL to receive alerts.
2. Use the webhook URL to set up notifications in Red Hat Advanced Cluster Security for Kubernetes.
3. Identify the policies you want to send notifications for, and update the notification settings for those policies.

5.1. CONFIGURING INTEGRATIONS BY USING WEBHOOKS

Create a new integration in Red Hat Advanced Cluster Security for Kubernetes by using the webhook URL.

Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Integrations**.
2. Scroll down to the **Notifier Integrations** section and select **Generic Webhook**.
3. Click **New integration**.
4. Enter a name for **Integration name**.
5. Enter the webhook URL in the **Endpoint** field.
6. If your webhook receiver uses an untrusted certificate, enter a CA certificate in the **CA certificate** field. Otherwise, leave it blank.

NOTE

The server certificate used by the webhook receiver must be valid for the endpoint DNS name. You can click **Skip TLS verification** to ignore this validation. Red Hat does not suggest turning off TLS verification. Without TLS verification, data could be intercepted by an unintended recipient.

7. Optional: Click **Enable audit logging** to receive alerts about all the changes made in Red Hat Advanced Cluster Security for Kubernetes.

NOTE

Red Hat suggests using separate webhooks for alerts and audit logs to handle these messages differently.

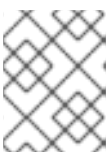
8. To authenticate with the webhook receiver, enter details for one of the following:
   - **Username** and **Password** for basic HTTP authentication
   - Custom **Header**, for example: **Authorization: Bearer <access_token>**
9. Use **Extra fields** to include additional key-value pairs in the JSON object that Red Hat Advanced Cluster Security for Kubernetes sends. For example, if your webhook receiver accepts objects from multiple sources, you can add **"source": "rhacs"** as an extra field and filter on this value to identify all alerts from Red Hat Advanced Cluster Security for Kubernetes.
10. Select **Test** to send a test message to verify that the integration with your generic webhook is working.
11. Select **Save** to create the configuration.

5.2. CONFIGURING POLICY NOTIFICATIONS

Enable alert notifications for system policies.

Procedure

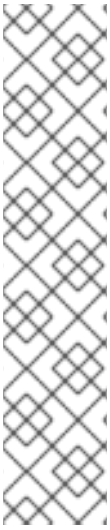
1. On the RHACS portal, navigate to **Platform Configuration → Policies**.
2. Select one or more policies for which you want to send alerts.
3. Under **Bulk actions**, select **Enable notification**.
4. In the **Enable notification** window, select the webhook notifier.



NOTE

If you have not configured any other integrations, the system displays a message that no notifiers are configured.

5. Click **Enable**.



NOTE

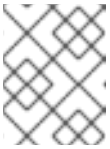
- Red Hat Advanced Cluster Security for Kubernetes sends notifications on an opt-in basis. To receive notifications, you must first assign a notifier to the policy.
- Notifications are only sent once for a given alert. If you have assigned a notifier to a policy, you will not receive a notification unless a violation generates a new alert.
- Red Hat Advanced Cluster Security for Kubernetes creates a new alert for the following scenarios:
  - A policy violation occurs for the first time in a deployment.
  - A runtime-phase policy violation occurs in a deployment after you resolved the previous runtime alert for a policy in that deployment.

CHAPTER 6. INTEGRATING WITH QRADAR

You can configure Red Hat Advanced Cluster Security for Kubernetes to send events to QRadar by configuring a generic webhook integration in RHACS.

The following steps represent a high-level workflow for integrating RHACS with QRadar:

1. In RHACS:
   a. Configure the generic webhook.



NOTE

When configuring the integration in RHACS, in the **Endpoint** field, use the following example as a guide: **<URL to QRadar Box>:<Port of Integration>**.

   b. Identify policies for which you want to send notifications, and update the notification settings for those policies.

2. If QRadar does not automatically detect the log source, add an RHACS log source on the QRadar Console. For more information on configuring QRadar and RHACS, see the Red Hat Advanced Cluster Security for Kubernetes IBM resource.

6.1. CONFIGURING INTEGRATIONS BY USING WEBHOOKS

Create a new integration in Red Hat Advanced Cluster Security for Kubernetes by using the webhook URL.

Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Integrations**.
2. Scroll down to the **Notifier Integrations** section and select **Generic Webhook**.
3. Click **New integration**.
4. Enter a name for **Integration name**.
5. Enter the webhook URL in the **Endpoint** field.
6. If your webhook receiver uses an untrusted certificate, enter a CA certificate in the **CA certificate** field. Otherwise, leave it blank.

NOTE

The server certificate used by the webhook receiver must be valid for the endpoint DNS name. You can click **Skip TLS verification** to ignore this validation. Red Hat does not suggest turning off TLS verification. Without TLS verification, data could be intercepted by an unintended recipient.

7. Optional: Click **Enable audit logging** to receive alerts about all the changes made in Red Hat Advanced Cluster Security for Kubernetes.

NOTE

Red Hat suggests using separate webhooks for alerts and audit logs to handle these messages differently.

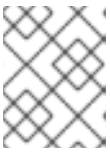
8. To authenticate with the webhook receiver, enter details for one of the following:
   - **Username** and **Password** for basic HTTP authentication
   - Custom **Header**, for example: **Authorization: Bearer <access_token>**
9. Use **Extra fields** to include additional key-value pairs in the JSON object that Red Hat Advanced Cluster Security for Kubernetes sends. For example, if your webhook receiver accepts objects from multiple sources, you can add **"source": "rhacs"** as an extra field and filter on this value to identify all alerts from Red Hat Advanced Cluster Security for Kubernetes.
10. Select **Test** to send a test message to verify that the integration with your generic webhook is working.
11. Select **Save** to create the configuration.

6.2. CONFIGURING POLICY NOTIFICATIONS

Enable alert notifications for system policies.

Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Policies**.
2. Select one or more policies for which you want to send alerts.
3. Under **Bulk actions**, select **Enable notification**.
4. In the **Enable notification** window, select the webhook notifier.



NOTE

If you have not configured any other integrations, the system displays a message that no notifiers are configured.

5. Click **Enable**.

NOTE

- Red Hat Advanced Cluster Security for Kubernetes sends notifications on an opt-in basis. To receive notifications, you must first assign a notifier to the policy.
- Notifications are only sent once for a given alert. If you have assigned a notifier to a policy, you will not receive a notification unless a violation generates a new alert.
- Red Hat Advanced Cluster Security for Kubernetes creates a new alert for the following scenarios:
    - A policy violation occurs for the first time in a deployment.
    - A runtime-phase policy violation occurs in a deployment after you resolved the previous runtime alert for a policy in that deployment.

CHAPTER 7. INTEGRATING WITH SERVICENOW

You can configure Red Hat Advanced Cluster Security for Kubernetes to send events to ServiceNow by configuring a generic webhook integration in RHACS.

The following steps represent a high-level workflow for integrating RHACS with ServiceNow:

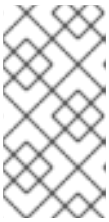
1. In ServiceNow, configure a REST API endpoint to use in RHACS. For more information that includes steps for ServiceNow configuration, see How to integrate Red Hat Advanced Cluster Security for Kubernetes with ServiceNow.
2. In RHACS:
   a. Configure the generic webhook.
   b. Identify policies for which you want to send notifications, and update the notification settings for those policies.

7.1. CONFIGURING INTEGRATIONS BY USING WEBHOOKS

Create a new integration in Red Hat Advanced Cluster Security for Kubernetes by using the webhook URL.

Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Integrations**.
2. Scroll down to the **Notifier Integrations** section and select **Generic Webhook**.
3. Click **New integration**.
4. Enter a name for **Integration name**.
5. Enter the webhook URL in the **Endpoint** field.
6. If your webhook receiver uses an untrusted certificate, enter a CA certificate in the **CA certificate** field. Otherwise, leave it blank.



NOTE

The server certificate used by the webhook receiver must be valid for the endpoint DNS name. You can click **Skip TLS verification** to ignore this validation. Red Hat does not suggest turning off TLS verification. Without TLS verification, data could be intercepted by an unintended recipient.

7. Optional: Click **Enable audit logging** to receive alerts about all the changes made in Red Hat Advanced Cluster Security for Kubernetes.

NOTE

Red Hat suggests using separate webhooks for alerts and audit logs to handle these messages differently.

8. To authenticate with the webhook receiver, enter details for one of the following:

- **Username** and **Password** for basic HTTP authentication
- Custom **Header**, for example: **Authorization: Bearer <access_token>**

9. Use **Extra fields** to include additional key-value pairs in the JSON object that Red Hat Advanced Cluster Security for Kubernetes sends. For example, if your webhook receiver accepts objects from multiple sources, you can add **"source": "rhacs"** as an extra field and filter on this value to identify all alerts from Red Hat Advanced Cluster Security for Kubernetes.
10. Select **Test** to send a test message to verify that the integration with your generic webhook is working.
11. Select **Save** to create the configuration.

7.2. CONFIGURING POLICY NOTIFICATIONS

Enable alert notifications for system policies.

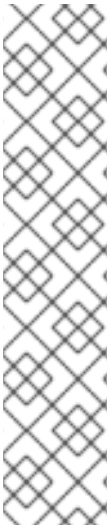
Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Policies**.
2. Select one or more policies for which you want to send alerts.
3. Under **Bulk actions**, select **Enable notification**.
4. In the **Enable notification** window, select the webhook notifier.

NOTE

If you have not configured any other integrations, the system displays a message that no notifiers are configured.

5. Click **Enable**.



NOTE

- Red Hat Advanced Cluster Security for Kubernetes sends notifications on an opt-in basis. To receive notifications, you must first assign a notifier to the policy.
- Notifications are only sent once for a given alert. If you have assigned a notifier to a policy, you will not receive a notification unless a violation generates a new alert.
- Red Hat Advanced Cluster Security for Kubernetes creates a new alert for the following scenarios:
  - A policy violation occurs for the first time in a deployment.
  - A runtime-phase policy violation occurs in a deployment after you resolved the previous runtime alert for a policy in that deployment.

CHAPTER 8. INTEGRATING WITH SUMO LOGIC

If you are using Sumo Logic, you can forward alerts from Red Hat Advanced Cluster Security for Kubernetes to Sumo Logic.

The following steps represent a high-level workflow for integrating Red Hat Advanced Cluster Security for Kubernetes with Sumo Logic:

1. Add a new Custom App in Sumo Logic, set the HTTP source, and get the HTTP URL.
2. Use the HTTP URL to integrate Sumo Logic with Red Hat Advanced Cluster Security for Kubernetes.
3. Identify the policies you want to send notifications for, and update the notification settings for those policies.

8.1. CONFIGURING SUMO LOGIC

Use the **Setup Wizard** to set up **Streaming Data** and get the HTTP URL.

Procedure

1. Log in to your Sumo Logic Home page and select **Setup Wizard**.
2. Move your cursor over to **Set Up Streaming Data** and select **Get Started**.
3. On the Select Data Type page, select **Your Custom App**.
4. On the Set Up Collection page, select **HTTP Source**.
5. Enter a name for **Source Category**, for example, **rhacs** and click **Continue**.
6. **Copy** the generated URL.

8.2. CONFIGURING RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES

Create a new integration in Red Hat Advanced Cluster Security for Kubernetes by using the HTTP URL.

Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Integrations**.
2. Scroll down to the **Notifier Integrations** section and select **Sumo Logic**.
3. Click **New Integration** (**add** icon).
4. Enter a name for **Integration Name**.
5. Enter the generated HTTP URL in the **HTTP Collector Source Address** field.
6. Click **Test** (**checkmark** icon) to test that the integration with Sumo Logic is working.
7. Click **Create** (**save** icon) to create the configuration.

8.3. CONFIGURING POLICY NOTIFICATIONS

Enable alert notifications for system policies.

Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Policies**.
2. Select one or more policies for which you want to send alerts.
3. Under **Bulk actions**, select **Enable notification**.
4. In the **Enable notification** window, select the Sumo Logic notifier.

NOTE

If you have not configured any other integrations, the system displays a message that no notifiers are configured.

5. Click **Enable**.

NOTE

- Red Hat Advanced Cluster Security for Kubernetes sends notifications on an opt-in basis. To receive notifications, you must first assign a notifier to the policy.
- Notifications are only sent once for a given alert. If you have assigned a notifier to a policy, you will not receive a notification unless a violation generates a new alert.
- Red Hat Advanced Cluster Security for Kubernetes creates a new alert for the following scenarios:
  - A policy violation occurs for the first time in a deployment.
  - A runtime-phase policy violation occurs in a deployment after you resolved the previous runtime alert for a policy in that deployment.

8.4. VIEWING ALERTS IN SUMO LOGIC

You can view alerts from Red Hat Advanced Cluster Security for Kubernetes in Sumo Logic.

1. Log in to your Sumo Logic Home page and click **Log Search**.
2. In the search box, enter **_sourceCategory=rhacs**. Make sure to use the same **Source Category** name that you entered while configuring Sumo Logic.
3. Select the time and then click **Start**.

CHAPTER 9. INTEGRATING WITH GOOGLE CLOUD STORAGE

You can integrate with Google Cloud Storage (GCS) to enable data backups. You can use these backups for data restoration in the case of an infrastructure disaster, or corrupt data. After you integrate with GCS, you can schedule daily or weekly backups and do manual on-demand backups.

The backup includes the Red Hat Advanced Cluster Security for Kubernetes entire database, which includes all configurations, resources, events, and certificates. Make sure that backups are stored securely.

NOTE

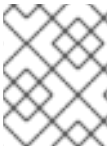
If you are using Red Hat Advanced Cluster Security for Kubernetes version 3.0.53 or older, the backup does not include certificates.

9.1. CONFIGURING RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES

To configure data backups on Google Cloud Storage (GCS), create an integration in Red Hat Advanced Cluster Security for Kubernetes.

Prerequisites

- An existing **bucket**. To create a new bucket, see the official Google Cloud Storage documentation topic Creating storage buckets.
- A **service account** with the **Storage Object Admin** IAM role in the storage bucket you want to use. See the official Google Cloud Storage documentation topic Using Cloud IAM permissions.
- A **service account key file** (JSON) for the Service account you are using. See the official Google Cloud documentation topics Creating a service account and Creating service account keys.



NOTE

Currently, Red Hat Advanced Cluster Security for Kubernetes does not support using Workload Identity to authenticate to GCS.

Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Integrations**.
2. Scroll down to the **External backups** section and select **Google Cloud Storage**.
3. Click **New Integration** (**add** icon).
4. Enter a name for **Integration Name**.
5. Enter the number of backups to retain in the **Backups To Retain** box.
6. For **Schedule**, select the backup frequency (daily or weekly) and the time to run the backup process.
7. Enter the **Bucket** name in which you want to store the backup.

8. In the **Service Account JSON** field, enter the contents of your service account key file.
9. Select **Test** (**checkmark** icon) to confirm that the integration with GCS is working.
10. Select **Create** (**save** icon) to create the configuration.

Once configured, Red Hat Advanced Cluster Security for Kubernetes automatically backs up all data according to the specified schedule.

9.1.1. Perform on-demand backups on Google Cloud Storage

Uses the RHACS portal to trigger manual backups of Red Hat Advanced Cluster Security for Kubernetes on Google Cloud Storage.

Prerequisites

- You must have already integrated Red Hat Advanced Cluster Security for Kubernetes with Google Cloud Storage.

Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Integrations**.
2. Under the **External backups** section, click **Google Cloud Storage**.
3. Select the integration name for the GCS bucket in which you want to do a backup.
4. Click **Trigger Backup**.

NOTE

Currently, when you select the **Trigger Backup** option, there is no notification. However, Red Hat Advanced Cluster Security for Kubernetes begins the backup task in the background.

9.1.1.1. Additional resources

- Backing up Red Hat Advanced Cluster Security for Kubernetes
- Restoring from a backup

CHAPTER 10. INTEGRATING BY USING THE SYSLOG PROTOCOL

Syslog is an event logging protocol that applications use to send messages to a central location, such as a SIEM or a syslog collector, for data retention and security investigations. With Red Hat Advanced Cluster Security for Kubernetes, you can send alerts and audit events using the syslog protocol.

NOTE

- Forwarding events by using the syslog protocol requires the Red Hat Advanced Cluster Security for Kubernetes version 3.0.52 or newer.
- When you use the syslog integration, Red Hat Advanced Cluster Security for Kubernetes forwards both violation alerts that you configure and all audit events.
- Currently, Red Hat Advanced Cluster Security for Kubernetes only supports **CEF** (Common Event Format).

The following steps represent a high-level workflow for integrating Red Hat Advanced Cluster Security for Kubernetes with a syslog events receiver:

1. Set up a syslog events receiver to receive alerts.
2. Use the receiver's address and port number to set up notifications in the Red Hat Advanced Cluster Security for Kubernetes.

After the configuration, Red Hat Advanced Cluster Security for Kubernetes automatically sends all violations and audit events to the configured syslog receiver.

10.1. CONFIGURING SYSLOG INTEGRATION WITH RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES

Create a new syslog integration in Red Hat Advanced Cluster Security for Kubernetes.

Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Integrations**.
2. Scroll down to the **Notifier Integrations** section and select **Syslog**.
3. Click **New Integration** (add icon).
4. Enter a name for **Integration Name**.
5. Select the **Logging Facility** value from **local0** through **local7**.
6. Enter your **Receiver Host** address and **Receiver Port** number.
7. If you are using TLS, turn on the **Use TLS** toggle.
8. If your syslog receiver uses a certificate that is not trusted, turn on the **Disable TLS Certificate Validation (Insecure)** toggle. Otherwise, leave this toggle off.

9. Select **Test** (**checkmark** icon) to send a test message to verify that the integration with your generic webhook is working.

10. Select **Create** (**save** icon) to create the configuration.

CHAPTER 11. INTEGRATING WITH AMAZON S3

You can integrate Red Hat Advanced Cluster Security for Kubernetes with Amazon S3 to enable data backups. You can use these backups for data restoration in the case of an infrastructure disaster or corrupt data. After you integrate with Amazon S3, you can schedule daily or weekly backups and do manual on-demand backups.

The backup includes the entire Red Hat Advanced Cluster Security for Kubernetes database, which includes all configurations, resources, events, and certificates. Make sure that backups are stored securely.

IMPORTANT

- If you are using Red Hat Advanced Cluster Security for Kubernetes version 3.0.53 or older, the backup does not include certificates.
- If your Amazon S3 is part of an air-gapped environment, you must add your AWS root CA as a trusted certificate authority in Red Hat Advanced Cluster Security for Kubernetes.

11.1. CONFIGURING AMAZON S3 INTEGRATION IN RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES

To configure Amazon S3 backups, create a new integration in Red Hat Advanced Cluster Security for Kubernetes.

Prerequisites

- An existing S3 Bucket. To create a new bucket with required permissions, see the Amazon documentation topic Creating a bucket.
- **Read**, **write**, and **delete** permissions for the S3 bucket, the **Access key ID**, and the **Secret access key**.
- If you are using **KIAM**, **kube2iam** or another proxy, then an **IAM role** that has the **read**, **write**, and **delete** permissions.

Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Integrations**.
2. Scroll down to the **External backups** section and select **Amazon S3**.
3. Click **New Integration** (**add** icon).
4. Enter a name for **Integration Name**.
5. Enter the number of backups to retain in the **Backups To Retain** box.
6. For **Schedule**, select the backup frequency as daily or weekly and the time to run the backup process.
7. Enter the **Bucket** name where you want to store the backup.

8. Optionally, enter an **Object Prefix** if you want to save the backups in a specific folder structure. For more details, see the Amazon documentation topic Working with object metadata.
9. Enter the **Endpoint** for the bucket if you are using a non-public S3 instance, otherwise leave it blank.
10. Enter the **Region** for the bucket.
11. Turn on the **Use Container IAM Role** toggle or enter the **Access Key ID**, and the **Secret Access Key**.
12. Select **Test** (**checkmark** icon) to confirm that the integration with Amazon S3 is working.
13. Select **Create** (**save** icon) to create the configuration.

Once configured, Red Hat Advanced Cluster Security for Kubernetes automatically backs up all data according to the specified schedule.

11.2. PERFORMING ON-DEMAND BACKUPS ON AMAZON S3

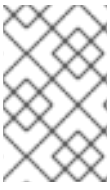
Uses the RHACS portal to trigger manual backups of Red Hat Advanced Cluster Security for Kubernetes on Amazon S3.

Prerequisites

- You must have already integrated Red Hat Advanced Cluster Security for Kubernetes with Amazon S3.

Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Integrations**.
2. Under the **External backups** section, click **Amazon S3**.
3. Select the integration name for the S3 bucket where you want to do a backup.
4. Click **Trigger Backup**.



NOTE

Currently, when you select the **Trigger Backup** option, there is no notification. However, Red Hat Advanced Cluster Security for Kubernetes begins the backup task in the background.

11.3. ADDITIONAL RESOURCES

- Backing up Red Hat Advanced Cluster Security for Kubernetes
- Restoring from a backup

CHAPTER 12. INTEGRATING WITH GOOGLE CLOUD SECURITY COMMAND CENTER

If you are using Google Cloud Security Command Center (Cloud SCC), you can forward alerts from Red Hat Advanced Cluster Security for Kubernetes to Cloud SCC. This guide explains how to integrate Red Hat Advanced Cluster Security for Kubernetes with Cloud SCC.

The following steps represent a high-level workflow for integrating Red Hat Advanced Cluster Security for Kubernetes with Cloud SCC.

1. Register a new security source with Google Cloud.
2. Provide the source ID and service account key to Red Hat Advanced Cluster Security for Kubernetes.
3. Identify the policies you want to send notifications for, and update the notification settings for those policies.

12.1. CONFIGURING GOOGLE CLOUD SCC

Start by adding Red Hat Advanced Cluster Security for Kubernetes as a trusted Cloud SCC source.

Procedure

1. Follow the Adding vulnerability and threat sources to Cloud Security Command Center guide and add Red Hat Advanced Cluster Security for Kubernetes as a trusted Cloud SCC source. Make a note of the **Source ID** that Google Cloud creates for your Red Hat Advanced Cluster Security for Kubernetes integration. If you do not see a source ID after registering, you can find it on the Cloud SCC Security Sources page .
2. Create a key for the service account you created, or the existing account you used, in the previous step. See Google Cloud's guide to creating and managing service account keys for details.

12.2. CONFIGURING RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES FOR INTEGRATING WITH GOOGLE CLOUD SCC

Create a new Google Cloud SCC integration in Red Hat Advanced Cluster Security for Kubernetes by using the **Source ID** and **service account key**.

Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Integrations**.
2. Scroll down to the **Notifier Integrations** section and select **Google Cloud SCC**.
3. Click **New Integration** (**add** icon).
4. Enter a name for **Integration Name**.
5. Enter the **Cloud SCC Source ID** and **Service Account Key (JSON)**.
6. Select **Create** (**save** icon) to create the configuration.

12.3. CONFIGURING POLICY NOTIFICATIONS

Enable alert notifications for system policies.

Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Policies**.
2. Select one or more policies for which you want to send alerts.
3. Under **Bulk actions**, select **Enable notification**.
4. In the **Enable notification** window, select the Google Cloud SCC notifier.

NOTE

If you have not configured any other integrations, the system displays a message that no notifiers are configured.

5. Click **Enable**.

NOTE

- Red Hat Advanced Cluster Security for Kubernetes sends notifications on an opt-in basis. To receive notifications, you must first assign a notifier to the policy.
- Notifications are only sent once for a given alert. If you have assigned a notifier to a policy, you will not receive a notification unless a violation generates a new alert.
- Red Hat Advanced Cluster Security for Kubernetes creates a new alert for the following scenarios:
  - A policy violation occurs for the first time in a deployment.
  - A runtime-phase policy violation occurs in a deployment after you resolved the previous runtime alert for a policy in that deployment.

CHAPTER 13. INTEGRATING WITH SPLUNK

If you are using Splunk, you can forward alerts from Red Hat Advanced Cluster Security for Kubernetes to Splunk and view vulnerability and compliance related data from within Splunk.

Depending on your use case, you can integrate Red Hat Advanced Cluster Security for Kubernetes with Splunk by using the following ways:

- By using an HTTP event collector in Splunk
  - Use the event collector option to forward alerts and audit log data
- By using the StackRox Kubernetes Security Platform add-on
  - Use the add-on to pull vulnerability detection and compliance data into Splunk

NOTE

The StackRox Kubernetes Security Platform add-on is only available if you are using Red Hat Advanced Cluster Security for Kubernetes version 3.0.51.0 or newer.

You can use one or both of these integration options to integrate the Red Hat Advanced Cluster Security for Kubernetes with Splunk.

13.1. USING THE HTTP EVENT COLLECTOR

You can forward alerts from Red Hat Advanced Cluster Security for Kubernetes to Splunk by using an HTTP event collector.

To integrate Red Hat Advanced Cluster Security for Kubernetes with Splunk by using the HTTP event collector, follow these steps:

1. Add a new HTTP event collector in Splunk and get the token value.
2. Use the token value to set up notifications in Red Hat Advanced Cluster Security for Kubernetes.
3. Identify policies for which you want to send notifications, and update the notification settings for those policies.

13.1.1. Adding an HTTP event collector in Splunk

Add a new HTTP event collector for your Splunk instance, and get the token.

Procedure

1. In your Splunk dashboard, navigate to **Settings → Add Data**.
2. Click **Monitor**.
3. On the **Add Data** page, click **HTTP Event Collector**.
4. Enter a **Name** for the event collector and then click **Next >**.

5. Accept the default **Input Settings** and click **Review >**.
6. Review the event collector properties and click **Submit >**.
7. Copy the **Token Value** for the event collector. You need this token value to configure integration with Splunk in Red Hat Advanced Cluster Security for Kubernetes.

13.1.1.1. Enabling HTTP event collector

You must enable HTTP event collector tokens before you can receive events.

Procedure

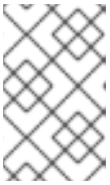
1. In your Splunk dashboard, navigate to **Settings → Data inputs**.
2. Click **HTTP Event Collector**.
3. Click **Global Settings**.
4. In the dialog that opens, click **Enabled** and then click **Save**.

13.1.2. Configuring Splunk integration in Red Hat Advanced Cluster Security for Kubernetes

Create a new Splunk integration in Red Hat Advanced Cluster Security for Kubernetes by using the token value.

Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Integrations**.
2. Scroll down to the **Notifier Integrations** section and select **Splunk**.
3. Click **New Integration** (**add** icon).
4. Enter a name for **Integration Name**.
5. Enter your Splunk URL in the **HTTP Event Collector URL** field. You must specify the port number if it is not **443** for HTTPS or **80** for HTTP. You must also add the URL path **/services/collector/event** at the end of the URL. For example, **https://<splunk-server-path>:8088/services/collector/event**.
6. Enter your token in the **HTTP Event Collector Token** field.



NOTE

If you are using Red Hat Advanced Cluster Security for Kubernetes version 3.0.57 or newer, you can specify custom **Source Type for Alert** events and **Source Type for Audit** events.

7. Select **Test** (**checkmark** icon) to send a test message to verify that the integration with Splunk is working.
8. Select **Create** (**save** icon) to create the configuration.

13.1.3. Configuring policy notifications

Enable alert notifications for system policies.

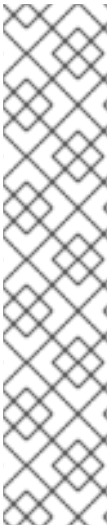
Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Policies**.
2. Select one or more policies for which you want to send alerts.
3. Under **Bulk actions**, select **Enable notification**.
4. In the **Enable notification** window, select the Splunk notifier.

NOTE

If you have not configured any other integrations, the system displays a message that no notifiers are configured.

5. Click **Enable**.



NOTE

- Red Hat Advanced Cluster Security for Kubernetes sends notifications on an opt-in basis. To receive notifications, you must first assign a notifier to the policy.
- Notifications are only sent once for a given alert. If you have assigned a notifier to a policy, you will not receive a notification unless a violation generates a new alert.
- Red Hat Advanced Cluster Security for Kubernetes creates a new alert for the following scenarios:
  - A policy violation occurs for the first time in a deployment.
  - A runtime-phase policy violation occurs in a deployment after you resolved the previous runtime alert for a policy in that deployment.

13.2. USING THE STACKROX KUBERNETES SECURITY PLATFORM ADD-ON

You can use the StackRox Kubernetes Security Platform add-on to forward the vulnerability detection and compliance related data from the Red Hat Advanced Cluster Security for Kubernetes to Splunk.

Begin by generating an API token with read permission for all resources in Red Hat Advanced Cluster Security for Kubernetes and then use that token to install and configure the add-on.

13.2.1. Installing and configuring the Splunk add-on

You can install the StackRox Kubernetes Security Platform add-on from your Splunk instance.

Prerequisites

- You must have an API token with **read** permission for all resources of Red Hat Advanced Cluster Security for Kubernetes. You can assign the **Analyst** system role to grant this level of access. The **Analyst** role has read permissions for all resources.

Procedure

1. Download the StackRox Kubernetes Security Platform add-on from Splunkbase.
2. Navigate to the Splunk home page on your Splunk instance.
3. Navigate to **Apps → Manage Apps**.
4. Select **Install app from file**
5. In the **Upload app** pop-up box, select **Choose File** and select the StackRox Kubernetes Security Platform add-on file.
6. Click **Upload**.
7. Click **Restart Splunk**, and confirm to restart.
8. After Splunk restarts, select **StackRox** from the **Apps** menu.
9. Click **Create New Input**
10. Either select **StackRox Compliance** to pull compliance data or **StackRox Vulnerability Management** to pull vulnerability data into Splunk.
11. Enter a **Name** for the input.
12. Select an **Interval** to pull data from Red Hat Advanced Cluster Security for Kubernetes. For example, every 14400 seconds.
13. Select the Splunk **Index** to which you want to send the data.
14. For **Central Endpoint**, enter the IP address or the name of your Central instance.
15. Enter the **API token** you have generated for the add-on.
16. Click **Add**.

CHAPTER 14. INTEGRATING WITH IMAGE VULNERABILITY SCANNERS

Red Hat Advanced Cluster Security for Kubernetes (RHACS) integrates with various vulnerability scanners to enable you to import your container images and monitor them for vulnerabilities. You can set up Red Hat Advanced Cluster Security for Kubernetes to obtain image vulnerability data from many open-source and commercial container image vulnerability scanners, including:

- Clair
- Google Container Analysis
- Red Hat Quay

If you are using one of these products in your DevOps workflow, you can use the RHACS portal to configure a connection with RHACS. Once integrated, your image vulnerabilities are surfaced in the RHACS portal and you can triage them easily.

14.1. INTEGRATING WITH COREOS CLAIR

You can integrate Red Hat Advanced Cluster Security for Kubernetes with CoreOS Clair for the static analysis of vulnerabilities in your images.

Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Integrations**.
2. Under the **Image Integrations** section, select **CoreOS Clair**.
3. Click **New integration**.
4. Enter the details for the following fields:
    a. **Integration name**: The name of the integration.
    b. **Endpoint**: The address of the scanner.
5. (Optional) If you are not using a TLS certificate when connecting to the registry, select **Disable TLS certificate validation (insecure)**.
6. (Optional) Select **Test** to test that the integration with the selected registry is working.
7. Select **Save**.

14.2. INTEGRATING WITH GOOGLE CONTAINER REGISTRY

You can integrate Red Hat Advanced Cluster Security for Kubernetes with Google Container Registry (GCR) for container analysis and vulnerability scanning.

Prerequisites

- You must have a service account key for the Google Container Registry.
- The associated service account has access to the registry. See Configuring access control for information about granting users and other projects access to GCR.

- If you are using GCR Container Analysis, you have granted the following roles to the service account:
  - Container Analysis Notes Viewer
  - Container Analysis Occurrences Viewer
  - Storage Object Viewer

Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Integrations**.
2. Under the **Image Integrations** section, select **Google Container Registry**.
   The **Configure image integration** modal box opens.
3. Click **New Integration**.
4. Enter the details for the following fields:
   a. **Integration Name**: The name of the integration.
   b. **Types**: Select **Scanner**.
   c. **Registry Endpoint**: The address of the registry.
   d. **Project**: The Google Cloud project name.
   e. **Service Account Key (JSON)** Your service account key for authentication.
5. Select **Test** (**checkmark** icon) to test that the integration with the selected registry is working.
6. Select **Create** (**save** icon) to create the configuration.

14.3. INTEGRATING WITH QUAY CONTAINER REGISTRY TO SCAN IMAGES

You can integrate Red Hat Advanced Cluster Security for Kubernetes with Quay Container Registry for scanning images.

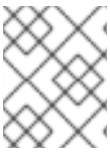
Prerequisites

- You must have an OAuth token for authentication with the Quay Container Registry to scan images.

Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Integrations**.
2. Under the **Image Integrations** section, select **Red Hat Quay.io**.
3. Click **New integration**.
4. Enter the **Integration name.**

5. Under **Type**, select **Scanner**. (If you are also integrating with the registry, select **Scanner + Registry**.) Enter information in the following fields:
    - **Endpoint**: Enter the address of the registry.
    - **OAuth token**: Enter the OAuth token that RHACS uses to authenticate by using the API.
    - Optional: **Robot username**: If you are configuring **Scanner + Registry** and are accessing the registry by using a Quay robot account, enter the user name in the format ***<namespace>+<accountname>***.
    - Optional: **Robot password**: If you are configuring **Scanner + Registry** and are accessing the registry by using a Quay robot account, enter the password for the robot account user name.
6. Optional: If you are not using a TLS certificate when connecting to the registry, select **Disable TLS certificate validation (insecure)**.
7. Optional: To create the integration without testing, select **Create integration without testing**.
8. Select **Save**.



NOTE

If you are editing a Quay integration but do not want to update your credentials, verify that **Update stored credentials** is not selected.

CHAPTER 15. INTEGRATING WITH JIRA

If you are using Jira, you can forward alerts from Red Hat Advanced Cluster Security for Kubernetes to Jira.

The following steps represent a high-level workflow for integrating Red Hat Advanced Cluster Security for Kubernetes with Jira:

1. Setup a user in Jira.
2. Use the Jira URL, username, and password to integrate Jira with Red Hat Advanced Cluster Security for Kubernetes.
3. Identify policies for which you want to send notifications, and update the notification settings for those policies.

15.1. CONFIGURING JIRA

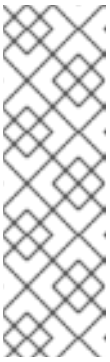
Start by creating a new user, and assign appropriate roles and permissions.

Prerequisites

- You need a Jira account with permissions to create and edit issues in the project with which you are integrating.

Procedure

- Create a user in Jira which have access to the projects for which you want to create issues:
  - To create a new user, see the Jira documentation topic Create, edit, or remove a user .
  - To give users access to project roles and applications, see the Jira documentation topic Assign users to groups, project roles, and applications .



NOTE

If you are using Jira Software Cloud, after you create the user, you must create a token for the user:

1. Go to https://id.atlassian.com/manage/api-tokens, to generate a new token.
2. Use the token as password when you configure Red Hat Advanced Cluster Security for Kubernetes.

15.2. CONFIGURING RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES

Create a new integration in Red Hat Advanced Cluster Security for Kubernetes by using the Jira server URL and user credentials.

Procedure

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

2. Scroll down to the **Notifier Integrations** section and select **Jira Software**.
3. Click **New Integration**.
4. Enter a name for **Integration Name**.
5. Enter the user credentials in the **Username** and **Password or API Token** boxes.
6. For **Issue Type**, enter a valid Jira Issue Type, for example **Task**, **Sub-task**, or **Bug**.
7. Enter the Jira server URL in the **Jira URL** box.
8. Enter the key of the project in which you want to create issues in the **Default Project** box.
9. Use the **Annotation Key For Project** box to create issues in different Jira projects.
10. If you use custom priorities in your Jira project, use the **Priority Mapping** toggle to configure custom priorities.
11. If you use mandatory custom fields in your JIRA project, enter them as JSON values in the **Default Fields JSON (Necessary If Required Fields)** box. For example:

```
{
  "customfield_10004": 3,
  "customfield_20005": "Alerts",
}
```

12. Select **Test** (**checkmark** icon) to test that the integration with Jira is working.
13. Select **Create** (**save** icon) to create the configuration.

15.2.1. Creating issues in different Jira projects

You can configure Red Hat Advanced Cluster Security for Kubernetes to to create issues in different Jira projects so that they directly go to the right team.

Prerequisites

- You must have an account with access to each project that you want to send the alerts to.

Procedure

1. Add an annotation similar to the following in your deployment YAML file:

```
jira/project-key: <jira_project_key>
```

2. Use the annotation key **jira/project-key** in the **Annotation Key For Project** field when you configure Red Hat Advanced Cluster Security for Kubernetes.

After the configuration is complete, if a deployment has an annotation in the YAML file, Red Hat Advanced Cluster Security for Kubernetes sends the alert to the project specified for that annotation. Otherwise, an alert is sent to the default project.

15.2.2. Configuring custom priorities in Jira

If you are using custom priorities in your Jira project, you can configure them in Red Hat Advanced Cluster Security for Kubernetes.

Procedure

1. While configuring Jira integration in Red Hat Advanced Cluster Security for Kubernetes, turn on the **Priority Mapping** toggle. Red Hat Advanced Cluster Security for Kubernetes gets the JIRA project schema, and auto fills the values for the **CRITICAL_SEVERITY**, **HIGH_SEVERITY**, **MEDIUM_SEVERITY**, and **LOW_SEVERITY** fields.
2. Verify or update the priority values based on your JIRA project configuration.
3. Select **Test** (**checkmark** icon) to test that the integration with Jira is working.
4. Select **Create** (**save** icon) to create the configuration.

NOTE

If you get an error, follow the instructions in the Troubleshooting Jira integration section.

15.3. CONFIGURING POLICY NOTIFICATIONS

Enable alert notifications for system policies.

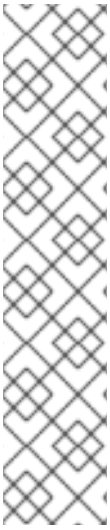
Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Policies**.
2. Select one or more policies for which you want to send alerts.
3. Under **Bulk actions**, select **Enable notification**.
4. In the **Enable notification** window, select the Jira notifier.

NOTE

If you have not configured any other integrations, the system displays a message that no notifiers are configured.

5. Click **Enable**.



NOTE

- Red Hat Advanced Cluster Security for Kubernetes sends notifications on an opt-in basis. To receive notifications, you must first assign a notifier to the policy.
- Notifications are only sent once for a given alert. If you have assigned a notifier to a policy, you will not receive a notification unless a violation generates a new alert.
- Red Hat Advanced Cluster Security for Kubernetes creates a new alert for the following scenarios:
  - A policy violation occurs for the first time in a deployment.
  - A runtime-phase policy violation occurs in a deployment after you resolved the previous runtime alert for a policy in that deployment.

15.4. TROUBLESHOOTING JIRA INTEGRATION

If you are using custom priorities or mandatory custom fields in your Jira project, you may get an error when you try to integrate Red Hat Advanced Cluster Security for Kubernetes with Jira Software. This error might be because of the mismatch between the severity and the priority field values.

If you do not know the custom priority values in your JIRA project, use the **roxctl** CLI to enable debug logging for JIRA integration.

Procedure

1. To get the custom priority values from your JIRA project, run the following command to turn on debug logging for JIRA integration:

```
$ roxctl -e "$ROX_CENTRAL_ADDRESS" central debug log --level Debug --modules
notifiers/jira
```

2. Follow the instructions to configure Red Hat Advanced Cluster Security for Kubernetes for Jira integration. When you test the integration, even if the integration test fails, the generated log includes your JIRA project schema and the custom priorities.
3. To save the debugging information as a compressed **.zip** file, run the following command:

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

4. Unzip the **.zip** file to retrieve the custom priority values in use in your JIRA project.
5. To turn off debug logging, run the following command:

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

6. Configure Red Hat Advanced Cluster Security for Kubernetes for Jira integration again and use the priority values to configure custom priorities.

CHAPTER 16. INTEGRATING WITH EMAIL

Configure Red Hat Advanced Cluster Security for Kubernetes (RHACS) to send alerts about policy violations to a standard email provider.

You can use email as a notification method by forwarding alerts from RHACS to a standard email provider. To forward alerts from the RHACS platform to an email address, you can use the **Default Recipient** field to send email to a standard and centralized team, or use deployment annotations to specify an audience for notifications.

With annotation keys, you can define an audience to notify about policy violations that are associated with a deployment or namespace. If the deployment has an annotation, the annotation's value overrides the default value. If the namespace has an annotation, the namespace's value overrides the default value.

- If a deployment has an annotation key and a defined audience, an email is sent to the audience who is defined by the key.
- If a deployment does not have an annotation key, the namespace is checked for an annotation key and an email is sent to the defined audience.
- If no annotation keys exist, an email is sent to the default recipient that is defined in the integration.

16.1. CONFIGURING THE EMAIL PLUGIN

The RHACS notifier can send email to a recipient specified in the integration, or it can use annotations to determine the recipient.

To use an annotation to dynamically determine an email recipient:

1. Add an annotation similar to the following example in your deployment YAML file, where **email** is the **Annotation key** that you specify in your email integration.

```
annotations:
  email: <email_address>
```

2. Use the annotation key **email** in the **Annotation key for recipient** field when you configure RHACS.

NOTE

You can create an annotation for the deployment or the namespace.

If you configured the deployment or namespace with an annotation, the RHACS platform sends the alert to the email specified in the annotation. Otherwise, it sends the alert to the default recipient.

Procedure

1. Navigate to **Platform Configuration → Integrations**.
2. Under the **Notifier Integrations** section, select **Email**.
3. Select **New Integration**.

4. In the **Integration name** field, enter a name for your email integration.

5. In the **Email server** field, enter the address of your email server. The email server address includes fully qualified domain name (FQDN) and the port number; for example, **smtp.example.com:465**.

6. Optional: If you are using unauthenticated SMTP, select **Enable unauthenticated SMTP**. This is insecure and not recommended, but might be required for some integrations. For example, you might need to enable this option if you use an internal server for notifications that does not require authentication.

NOTE

You cannot change an existing email integration that uses authentication to enable unauthenticated SMTP. You must delete the existing integration and create a new one with **Enable unauthenticated SMTP** selected.

7. Enter the user name and password of a service account that is used for authentication.

8. Optional: Enter the name that you want to appear in the **FROM** header of email notifications in the **From** field; for example, **Security Alerts**.

9. Specify the email address that you want to appear in the **SENDER** header of email notifications in the **Sender** field.

10. Specify the email address that will receive the notifications in the **Default recipient** field.

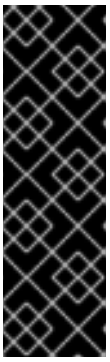
11. Optional: Enter an annotation key in **Annotation key for recipient** If you provide an annotation and the deployment or the namespace has a key with this value, then notifications will be sent to the email address in the annotation. Otherwise, notifications are sent to the email specified in the **Default Recipient** field.

12. Optional: Select **Disable TLS certificate validation (insecure)** to send email without TLS. You should not disable TLS unless you are using StartTLS.

NOTE

Use TLS for email notifications. Without TLS, all email is sent unencrypted.

13. Optional: To use StartTLS, select either **Login** or **Plain** from the **Use STARTTLS (requires TLS to be disabled)** drop-down menu.



IMPORTANT

With StartTLS, credentials are passed in plain text to the email server before the session encryption is established.

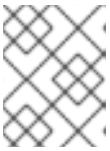
- StartTLS with the **Login** parameter sends authentication credentials in a **base64** encoded string.
- StartTLS with the **Plain** parameter sends authentication credentials to your mail relay in plain text.

16.2. CONFIGURING POLICY NOTIFICATIONS

Enable alert notifications for system policies.

Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Policies**.
2. Select one or more policies for which you want to send alerts.
3. Under **Bulk actions**, select **Enable notification**.
4. In the **Enable notification** window, select the email notifier.



NOTE

If you have not configured any other integrations, the system displays a message that no notifiers are configured.

5. Click **Enable**.

NOTE

- Red Hat Advanced Cluster Security for Kubernetes sends notifications on an opt-in basis. To receive notifications, you must first assign a notifier to the policy.
- Notifications are only sent once for a given alert. If you have assigned a notifier to a policy, you will not receive a notification unless a violation generates a new alert.
- Red Hat Advanced Cluster Security for Kubernetes creates a new alert for the following scenarios:
  - A policy violation occurs for the first time in a deployment.
  - A runtime-phase policy violation occurs in a deployment after you resolved the previous runtime alert for a policy in that deployment.