Red Hat Advanced Cluster Security for Kubernetes 4.4

配置

为 Kubernetes 配置 Red Hat Advanced Cluster Security

Last Updated: 2024-07-02

Red Hat Advanced Cluster Security for Kubernetes 4.4 配置

为 Kubernetes 配置 Red Hat Advanced Cluster Security

法律通告

摘要

本文档描述了如何执行常见的配置任务，包括配置证书、自动升级和代理设置。它还包含有关启用监控和日志记录的信息。

目录

第 1 章 添加自定义证书

了解如何在 Red Hat Advanced Cluster Security for Kubernetes 中使用自定义 TLS 证书。设置证书后，用户和 API 客户端不必在连接到 Central 时绕过证书安全警告。

1.1. 添加自定义安全证书

您可以在安装过程中或在现有的 Red Hat Advanced Cluster Security for Kubernetes 部署中应用安全证书。

1.1.1. 添加自定义证书的先决条件

前提条件

- 您必须已经有 PEM 编码的私钥和证书文件。
- 证书文件应以人类可读的块开头和结束。例如：

```
-----BEGIN CERTIFICATE-----
MIICLDCCAdKgAwIBAgIBADAKBggqhkjOPQQDAjB9MQswCQYDVQQGEwJCRTEPMA0G
...
l4wOuDwKQa+upc8GftXE2C//4mKANBC6It01gUaTIpo=
-----END CERTIFICATE-----
```

- 证书文件可以包含单个（叶）证书，也可以是证书链。

警告

- 如果证书不是由可信 root 直接签名的证书，您必须提供完整的证书链，包括任何中间证书。
- 链中的所有证书都必须按顺序进行，以便叶证书是第一个证书，root 证书是链中的最后一个证书。

- 如果您使用不是全局可信的自定义证书，还必须将 Sensor 配置为信任您的自定义证书。

1.1.2. 在新安装过程中添加自定义证书

流程

- 如果要使用 Operator 安装 Red Hat Advanced Cluster Security for Kubernetes：
  1. 输入以下命令，创建一个 **central-default-tls-cert** secret，在要安装 Central 服务的命名空间中包含适当的 TLS 证书：

```
oc -n <namespace> create secret tls central-default-tls-cert --cert <tls-cert.pem> --key
<tls-key.pem>
```

- 如果要使用 Helm 安装 Red Hat Advanced Cluster Security for Kubernetes：

  1. 在 **values-private.yaml** 文件中添加自定义证书及其密钥：

```
central:
  # Configure a default TLS certificate (public cert + private key) for central
  defaultTLS:
    cert: |
      -----BEGIN CERTIFICATE-----

EXAMPLE!MIIMIICLDCCAdKgAwIBAgIBADAKBggqhkjOPQQDAjB9MQswCQYDVQQGE
wJCRTEPMA0G
      ...
      -----END CERTIFICATE-----
    key: |
      -----BEGIN EC PRIVATE KEY-----
      EXAMPLE!MHcl4wOuDwKQa+upc8GftXE2C//4mKANBC6It01gUaTIpo=
      ...
      -----END EC PRIVATE KEY-----
```

  2. 在安装过程中提供配置文件：

```
$ helm install -n stackrox --create-namespace stackrox-central-services rhacs/central-services -f values-private.yaml
```

- 如果您要使用 **roxctl** CLI 安装 Red Hat Advanced Cluster Security for Kubernetes，请在运行安装程序时提供证书和密钥文件：

  - 对于非交互式安装程序，请使用 **--default-tls-cert** 和 **--default-tls-key** 选项：

```
$ roxctl central generate --default-tls-cert "cert.pem" --default-tls-key "key.pem"
```

  - 对于交互式安装程序，在输入提示时提供证书和密钥文件：

```
...
Enter PEM cert bundle file (optional): <cert.pem>
Enter PEM private key file (optional): <key.pem>
Enter administrator password (default: autogenerated):
Enter orchestrator (k8s, openshift): openshift
...
```

1.1.3. 为现有实例添加自定义证书

流程

- 如果使用 Operator 安装 Red Hat Advanced Cluster Security for Kubernetes：

  1. 输入以下命令，创建一个 **central-default-tls-cert** secret，在安装了 Central 服务的命名空间中包含适当的 TLS 证书：

```
oc -n <namespace> create secret tls central-default-tls-cert --cert <tls-cert.pem> --key <tls-key.pem>
```

- 如果您使用 Helm 安装 Red Hat Advanced Cluster Security for Kubernetes：

1. 在 **values-private.yaml** 文件中添加自定义证书及其密钥：

```
central:
  # Configure a default TLS certificate (public cert + private key) for central
  defaultTLS:
    cert: |
      -----BEGIN CERTIFICATE-----

EXAMPLE!MIIMIICLDCCAdKgAwIBAgIBADAKBggqhkjOPQQDAjB9MQswCQYDVQQGE
wJCRTEPMA0G
      ...
      -----END CERTIFICATE-----
    key: |
      -----BEGIN EC PRIVATE KEY-----
      EXAMPLE!MHcl4wOuDwKQa+upc8GftXE2C//4mKANBC6It01gUaTIpo=
      ...
      -----END EC PRIVATE KEY-----
```

2. 使用 **helm upgrade** 命令并提供更新的配置文件：

```
$ helm upgrade -n stackrox --create-namespace stackrox-central-services \
 rhacs/central-services --reuse-values \ 1
 -f values-private.yaml
```

1 您必须使用此参数，因为 **values-private.yaml** 文件不包含所有所需的配置值。

- 如果您使用 **roxctl** CLI 安装 Red Hat Advanced Cluster Security for Kubernetes：
  - 从 PEM 编码的密钥和证书文件创建并应用 TLS secret：

```
$ oc -n stackrox create secret tls central-default-tls-cert \
  --cert <server_cert.pem> \
  --key <server_key.pem> \
  --dry-run -o yaml | oc apply -f -
```

运行此命令后，Central 会自动应用新密钥和证书，而无需重启 pod。可能需要一分钟时间来传播更改。

1.1.4. 为现有实例更新自定义证书

如果将自定义证书用于 Central，您可以执行以下步骤来更新证书。

流程

1. 删除现有的自定义证书 secret：

```
$ oc delete secret central-default-tls-cert
```

2. 创建新 secret：

```
$ oc -n stackrox create secret tls central-default-tls-cert \
  --cert <server_cert.pem> \
  --key <server_key.pem> \
```

```
--dry-run -o yaml | oc apply -f -
```

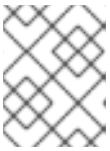
3. 重启 Central 容器。

1.1.4.1. 重启 Central 容器

您可以通过终止 Central 容器或删除 Central pod 来重启 Central 容器。

流程

- 运行以下命令以终止 Central 容器：



注意

您必须至少等待 1 分钟，直到 OpenShift Container Platform 传播您的更改并重启 Central 容器。

```
$ oc -n stackrox exec deploy/central -c central -- kill 1
```

- 或者，运行以下命令来删除 Central pod：

```
$ oc -n stackrox delete pod -lapp=central
```

1.2. 配置 SENSOR 以信任自定义证书

如果您使用不全局信任的自定义证书，您必须将 Sensor 配置为信任您的自定义证书。否则，您可能会遇到错误。具体类型的错误会因您的设置和您使用的证书可能会有所不同。通常，它是一个与 **x509** 验证相关的错误。

注意

如果您使用全局可信证书，则不需要将 Sensor 配置为信任您的自定义证书。

1.2.1. 下载 Sensor 捆绑包

Sensor 捆绑包包括安装 Sensor 所需的配置文件和脚本。您可以从 RHACS 门户下载 Sensor 捆绑包。

流程

1. 在 RHACS 门户中，进入 **Platform Configuration → Clusters**。
2. 点 **New Cluster** 并为集群指定一个名称。
3. 如果您要在同一集群中部署 Sensor，请接受所有字段的默认值。否则，如果您要部署到不同的集群中，请将地址 **central.stackrox.svc:443** 替换为负载均衡器、节点端口或其他地址（包括端口号），该地址可从您要安装的其他集群访问。

注意

如果您使用一个支持非 gRPC 的负载均衡器，如 HAProxy、AWS Application Load Balancer (ALB) 或 AWS Elastic Load Balancing (ELB)，请使用 WebSocket Secure (**wss**) 协议。使用 **ws** ：

1. 使用 **wss://** 为地址加上前缀，以及
2. 在地址后添加端口号，例如 **ws://stackrox-central.example.com:443**。

4. 点 **Next** 继续。
5. 点 **Download YAML File and Keys**。

1.2.2. 在部署新的 Sensor 时将 Sensor 配置为信任自定义证书

前提条件

- 您已下载了 Sensor 捆绑包。

流程

- 如果您使用 **sensor.sh** 脚本：
  1. 解压 Sensor 捆绑包：

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

  2. 运行 **sensor.sh** 脚本：

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

     当您运行传感器(**./sensor/sensor.sh**)脚本时，证书会自动应用。在运行 **sensor.sh** 脚本前，您还可以将额外的自定义证书放在 **sensor/additional-cas/** 目录中。

- 如果您不使用 **sensor.sh** 脚本：
  1. 解压 Sensor 捆绑包：

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

  2. 运行以下命令来创建 secret：

     ```
     $ ./sensor/ca-setup-sensor.sh -d sensor/additional-cas/ 1
     ```

     1 使用 **-d** 选项指定包含自定义证书的目录。

注意

如果得到 "secret already exists" 错误消息，请使用 **-u** 选项重新运行脚本：

```
$ ./sensor/ca-setup-sensor.sh -d sensor/additional-cas/ -u
```

3. 使用 YAML 文件继续 Sensor 部署。

1.2.3. 配置现有 Sensor 以信任自定义证书

前提条件

- 您已下载了 Sensor 捆绑包。

流程

1. 解压 Sensor 捆绑包：

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

2. 运行以下命令来创建 secret：

```
$ ./sensor/ca-setup-sensor.sh -d sensor/additional-cas/ 1
```

1 使用 **-d** 选项指定包含自定义证书的目录。

注意

如果得到 "secret already exists" 错误消息，请使用 **-u** 选项重新运行脚本：

```
$ ./sensor/ca-setup-sensor.sh -d sensor/additional-cas/ -u
```

3. 使用 YAML 文件继续 Sensor 部署。

如果将证书添加到现有传感器中，您必须重启 Sensor 容器。

1.2.3.1. 重启 Sensor 容器

您可以通过终止容器或删除 Sensor pod 来重启 Sensor 容器。

流程

- 运行以下命令以终止 Sensor 容器：

注意

您必须至少等待 1 分钟，直到 OpenShift Container Platform 或 Kubernetes 传播您的更改并重启 Sensor 容器。

  - 在 OpenShift Container Platform 中：

```
$ oc -n stackrox deploy/sensor -c sensor -- kill 1
```

  - 对于 Kubernetes：

```
$ kubectl -n stackrox deploy/sensor -c sensor -- kill 1
```

- 或者，运行以下命令来删除 Sensor pod：
  - 在 OpenShift Container Platform 中：

    ```
    $ oc -n stackrox delete pod -lapp=sensor
    ```

  - 对于 Kubernetes：

    ```
    $ kubectl -n stackrox delete pod -lapp=sensor
    ```

第 2 章 添加可信证书颁发机构

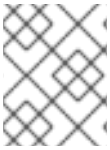
了解如何在 Red Hat Advanced Cluster Security for Kubernetes 中添加自定义可信证书颁发机构。

如果您在网络或自签名证书中使用企业证书颁发机构(CA)，您必须将 CA 的 root 证书添加到 Red Hat Advanced Cluster Security for Kubernetes 中作为可信 root CA。

添加可信 root CA 允许：

- 与其它工具集成时，中央和扫描器以信任远程服务器。
- 信任用于 Central 的自定义证书。

您可以在安装过程中或现有部署上添加额外的 CA。



注意

您必须首先在部署了 Central 的集群中配置可信 CA，然后将更改传播到 Scanner 和 Sensor。

2.1. 配置其他 CA

添加自定义 CA：

流程

1. 下载 **ca-setup.sh** 脚本。

注意

- 如果要进行新安装，您可以在 **scripts** 目录中找到 **ca-setup.sh** 脚本（**central-bundle/central/scripts/ca-setup.sh**）。
- 您必须在登录到 OpenShift Container Platform 集群的同一终端中运行 **ca-setup.sh** 脚本。

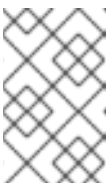
2. 使 **ca-setup.sh** 脚本可执行：

```
$ chmod +x ca-setup.sh
```

3. 要添加：

   a. 单个证书，使用 **-f** （文件）选项：

```
$ ./ca-setup.sh -f <certificate>
```

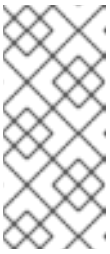


注意

- 您必须使用 PEM 编码的证书文件（具有任何扩展名）。
- 您还可以使用 **-u** (update)选项和 **-f** 选项更新之前添加的任何证书。

   b. 一次性移动目录中的所有证书，然后使用 **-d** （目录）选项：

```
$ ./ca-setup.sh -d <directory_name>
```



注意

- 您必须使用带有 **.crt** 或 **.pem** 扩展名的 PEM 编码证书文件。
- 每个文件必须仅包含单个证书。
- 您还可以使用 **-u**（更新）选项和 **-d** 选项更新任何之前添加的证书。

2.2. 传播更改

配置可信 CA 后，您必须使 Red Hat Advanced Cluster Security for Kubernetes 服务信任它们。

- 如果您在安装后配置了可信 CA，则必须重启 Central。
- 另外，如果您要添加证书以与镜像 registry 集成，则必须重启 Central 和 Scanner。

2.2.1. 重启 Central 容器

您可以通过终止 Central 容器或删除 Central pod 来重启 Central 容器。

流程

- 运行以下命令以终止 Central 容器：

注意

您必须至少等待 1 分钟，直到 OpenShift Container Platform 传播您的更改并重启 Central 容器。

```
$ oc -n stackrox exec deploy/central -c central -- kill 1
```

- 或者，运行以下命令来删除 Central pod：

```
$ oc -n stackrox delete pod -lapp=central
```

2.2.2. 重启 Scanner 容器

您可以通过删除 pod 来重启 Scanner 容器。

流程

- 运行以下命令以删除 Scanner pod：
  - 在 OpenShift Container Platform 中：

```
$ oc delete pod -n stackrox -l app=scanner
```

  - 对于 Kubernetes：

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

重要

添加可信 CA 并配置了 Central 后，CA 将包含在您创建的任何新的 Sensor 部署捆绑包中。

- 如果在连接到 Central 时现有 Sensor 报告问题，您必须生成 Sensor 部署 YAML 文件并更新现有集群。
- 如果要使用 **sensor.sh** 脚本部署新的 Sensor，请在运行 **sensor.sh** 脚本前运行以下命令：

  ```
  $ ./ca-setup-sensor.sh -d ./additional-cas/
  ```

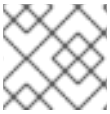
- 如果要使用 Helm 部署新的 Sensor，则不必运行任何其他脚本。

第 3 章 重新签发内部证书

Red Hat Advanced Cluster Security for Kubernetes 的每个组件都使用 X.509 证书向其他组件验证其自身。这些证书具有过期日期，您必须在证书过期前重新发布或轮转证书。您可以通过在 RHACS 门户中选择 **Platform Configuration → Clusters** 来查看证书过期日期，并查看 **Credential Expiration** 列。

3.1. 为 CENTRAL 重复内部证书

Central 在与其他 Red Hat Advanced Cluster Security for Kubernetes 服务通信时使用内置服务器证书进行身份验证。此证书对 Central 安装是唯一的。RHACS 门户显示 Central 证书即将过期时的信息横幅。



注意

信息横幅仅在证书过期日期前 15 天出现。

对于基于 Operator 的安装，从 RHACS 版本 4.3.4 开始，Operator 会在过期前自动轮转所有 Central 组件的服务传输层安全(TLS)证书 6 个月。适用以下条件：

- 在 secret 中轮转证书不会触发组件自动重新载入它们。但是，当 pod 作为 RHACS 升级的一部分或因为节点重启而被替换时，通常会重新载入。如果这些事件至少每 6 个月发生一次，则必须在旧的（内存中）服务证书过期前重启 pod。例如，您可以删除具有 **app** 标签的 pod，其包含 central、 **central -db**、Scanner 或 **scanner -db** 的值之一。
- CA 证书不会更新。它们有效期为 5 年。
- 安全集群组件使用的 init 捆绑包中的服务证书不会更新。您必须定期轮转 init 捆绑包。

对于基于非 Operator 的安装，您必须手动轮转 TLS 证书。以下部分包含了手动轮转证书的说明。

前提条件

- 要重新发布或轮转证书，您必须具有 **ServiceIdentity** 资源的写入权限。

流程

1. **在 RHACS 门户中，点击横幅中的链接，该链接声明证书过期时间下载 YAML 配置文件，其中包含新 secret。secret 包括证书和密钥值。**

2. **运行以下命令，将新的 YAML 配置文件应用到安装 Central 的集群：**

```
$ oc apply -f <secret_file.yaml>
```

3. **重启 Central 以应用更改。**

3.1.1. 重启 Central 容器

您可以通过终止 Central 容器或删除 Central pod 来重启 Central 容器。

流程

- 运行以下命令以终止 Central 容器：

注意

您必须至少等待 1 分钟，直到 OpenShift Container Platform 传播您的更改并重启 Central 容器。

```
$ oc -n stackrox exec deploy/central -c central -- kill 1
```

- 或者，运行以下命令来删除 Central pod：

```
$ oc -n stackrox delete pod -lapp=central
```

3.2. 为 SCANNER 修复内部证书

扫描程序有一个内置证书，用于与 Central 通信。

当 Scanner 证书即将过期时，RHACS 门户会显示信息横幅。

注意

信息横幅仅在证书到期日期前 15 天出现。

前提条件

- 要重新发布证书，您必须具有 ServiceIdentity 资源的写入权限。

流程

1. 点横幅中的链接下载 YAML 配置文件，其中包含一个新的 OpenShift Container Platform secret，包括证书和密钥值。

2. 将新的 YAML 配置文件应用到安装 Scanner 的集群。

```
$ oc apply -f <secret_file.yaml>
```

3. 重启 Scanner 以应用更改。

3.2.1. 重启 Scanner 和 Scanner DB 容器

您可以通过删除 pod 来重启 Scanner 和 Scanner DB 容器。

流程

- 要删除 Scanner 和 Scanner DB pod，请运行以下命令：
  - 在 OpenShift Container Platform 中：

```
$ oc delete pod -n stackrox -l app=scanner; oc -n stackrox delete pod -l
app=scanner-db
```

  - 对于 Kubernetes：

```
$ kubectl delete pod -n stackrox -l app=scanner; kubectl -n stackrox delete pod -l
app=scanner-db
```

3.3. 为 SENSOR、COLLECTOR 和 ADMISSION 控制器恢复内部证书

Sensor、Collector 和 Admission 控制器使用证书相互通信，并与 Central 通信。

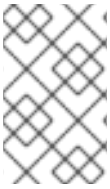
要替换证书，请使用以下方法之一：

- 在安全集群中创建、下载并安装 init 捆绑包。您必须具有 Admin 用户角色才能创建 init 捆绑包。
- 使用自动升级功能。自动升级仅适用于使用 roxctl CLI 的静态清单部署。

3.3.1. 使用 init 捆绑包为安全集群恢复内部证书

安全集群包含 Collector、Sensor 和 Admission Control 组件。这些组件在与其他 Red Hat Advanced Cluster Security for Kubernetes 组件通信时使用内置服务器证书进行身份验证。

RHACS 门户显示 Central 证书即将过期时的信息横幅。



注意

信息横幅仅在证书到期日期前 15 天出现。

前提条件

- 要重新发布证书，您必须具有 ServiceIdentity 资源的写入权限。

重要

安全地存储此捆绑包，因为它包含 secret。您可以在多个安全集群中使用相同的捆绑包。您必须具有 Admin 用户角色才能创建 init 捆绑包。

流程

- 使用 RHACS 门户生成 init 捆绑包：

a. 选择 Platform Configuration → Clusters。

b. 单击 Manage Tokens。

c. 进入 Authentication Tokens 部分，再点 Cluster Init Bundle。

d. 点 Generate bundle。

e. 为集群 init 捆绑包输入一个名称并点 Generate。

f. 要下载生成的捆绑包，请点 Download Kubernetes secrets file。

- 要使用 roxctl CLI 生成 init 捆绑包，请运行以下命令：

```
$ roxctl -e <endpoint> -p <admin_password> central init-bundle generate
<bundle_name> --output-secrets init-bundle.yaml
```

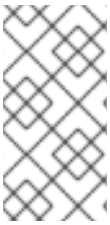
后续步骤

- 要在每个安全集群中创建所需资源，请运行以下命令：

```
$ oc -n stackrox apply -f <init-bundle.yaml>
```

3.3.2. 使用自动升级为安全集群恢复内部证书

您可以使用自动升级为 Sensor、Collector 和 Admission 控制器重新发布内部证书。



注意

自动升级仅适用于使用 roxctl CLI 的基于静态清单的部署。*请参阅安装* 章节中的"使用 roxctl CLI 安装"部分中的"安装 Central"。

前提条件

- 您必须为所有集群启用自动升级。
- 要重新发布证书，您必须具有 ServiceIdentity 资源的写入权限。

流程

1. 在 RHACS 门户中，进入 Platform Configuration → Clusters。
2. 在 Clusters 视图中，选择一个 Cluster 来查看其详情。

3. 在集群详情面板中，选择到 **Apply credentials by using an automatic upgrade** 的链接。

注意

当您应用自动升级时，**Red Hat Advanced Cluster Security for Kubernetes** 在所选集群中创建新凭证。但是，您仍会看到通知。当每个 **Red Hat Advanced Cluster Security for Kubernetes** 服务在服务重启后使用新凭证时，通知会退出。

第 4 章 添加安全通知

使用 Red Hat Advanced Cluster Security for Kubernetes，您可以添加用户登录时看到的安全公告。您还可以在 RHACS 门户的顶部或底部设置机构范围内的消息或声明者。

此消息可作为企业政策的提醒，并通知员工的相应策略。或者，您可能因为法律原因显示这些消息，例如，警告用户正在审核其操作。

4.1. 添加自定义登录信息

在登录警告或不格式的用户有关其操作后，显示警告消息。

前提条件

- 您必须具有带有 read 权限的 Config 角色，才能查看登录消息配置选项。
- 您需要具有带有 write 权限的 Config 角色来修改、启用或禁用登录消息。

流程

1. 在 RHACS 门户中，进入 Platform Configuration → System Configuration。
2. 在 System Configuration 视图 标头中，单击 Edit。
3. 在 Login Configuration 部分中，输入您的登录信息。
4. 要启用登录信息，请在 Login Configuration 部分中打开切换。
5. 点 Save。

4.2. 添加自定义标头和页脚

您可以将自定义文本放在标头和页脚中，并配置文本及其背景颜色。

前提条件

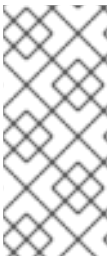
- 您必须具有具有 read 权限的 Config 角色，才能查看自定义标头和页脚配置选项。
- 您必须具有具有 write 权限的 Config 角色才能修改、启用或禁用自定义标头和页脚。

流程

1. 在 RHACS 门户中，进入 Platform Configuration → System Configuration。
2. 在 System Configuration 视图 标头中，单击 Edit。
3. 在 Header Configuration 和 Footer Configuration 部分下，输入标头和页脚文本。
4. 自定义标头和页脚 文本、大小 和 Background Color。
5. 要启用标头，请在 Header Configuration 部分打开切换。
6. 要启用页脚，请在 Footer Configuration 部分中打开切换。
7. 点 Save。

第 5 章 启用离线模式

您可以通过启用离线模式，使用 Red Hat Advanced Cluster Security for Kubernetes for 集群没有连接到互联网。在离线模式下，Red Hat Advanced Cluster Security for Kubernetes 组件没有连接到互联网上的地址或主机。



注意

Red Hat Advanced Cluster Security for Kubernetes 不决定用户提供的主机名、IP 地址或其他资源是否在互联网上。例如，如果您尝试与互联网上托管的 Docker registry 集成，Red Hat Advanced Cluster Security for Kubernetes 将不会阻止此请求。

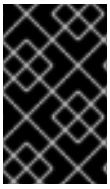
以离线模式部署和操作 Red Hat Advanced Cluster Security for Kubernetes：

1. 下载 RHACS 镜像并在集群中安装它们。如果使用 OpenShift Container Platform，您可以使用 Operator Lifecycle Manager (OLM) 和 OperatorHub 将镜像下载到连接到互联网的工作站。然后，工作站会将镜像推送到安全集群的镜像 registry。对于其他平台，您可以使用 Skopeo 或 Docker 等程序从远程 registry 拉取镜像并将其推送到您自己的私有 registry，如 下载镜像 中所述。

2. 在安装过程中启用离线模式。

3. （可选）通过上传新的定义文件来根据情况更新扫描器的漏洞列表。

4. （可选）当需要时，通过上传新的内核支持软件包，在更多内核版本上添加对运行时集合的支持。



重要

您只能在安装过程中启用离线模式，而不在升级过程中启用。

5.1. 下载镜像以离线使用

5.1.1. 直接下载镜像

您可以手动拉取、重新标记并将 Red Hat Advanced Cluster Security for Kubernetes 镜像推送到

registry。镜像捆绑包的当前版本中包含的镜像有：

- registry.redhat.io/advanced-cluster-security/rhacs-main-rhel8:4.4.3
- registry.redhat.io/advanced-cluster-security/rhacs-scanner-rhel8:4.4.3
- registry.redhat.io/advanced-cluster-security/rhacs-scanner-db-rhel8:4.4.3
- registry.redhat.io/advanced-cluster-security/rhacs-collector-rhel8:4.4.3
- registry.redhat.io/advanced-cluster-security/rhacs-collector-slim-rhel8:4.4.3

5.1.1.1. 重新标记镜像

您可以使用 Docker 命令行界面下载和重新标记镜像。

重要

当重新标记镜像时，您必须维护镜像的名称和标签。例如，使用：

```
$ docker tag registry.redhat.io/advanced-cluster-security/rhacs-main-rhel8:4.4.3
<your_registry>/rhacs-main-rhel8:4.4.3
```

不要重新标签类似以下示例：

```
$ docker tag registry.redhat.io/advanced-cluster-security/rhacs-main-rhel8:4.4.3
<your_registry>/other-name:latest
```

流程

1. 登录到 registry：

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

2. 拉取镜像：

```
$ docker pull <image>
```

3. 重新标记镜像：

```
$ docker tag <image> <new_image>
```

4. 将更新的镜像推送到 registry 中：

```
$ docker push <new_image>
```

5.2. 在安装过程中启用离线模式

您可在安装 Red Hat Advanced Cluster Security for Kubernetes 过程中启用离线模式。

5.2.1. 使用 Helm 配置启用离线模式

当使用 Helm chart 安装 Red Hat Advanced Cluster Security for Kubernetes 时，您可以在安装过程中启用离线模式。

流程

1. 安装 central-services Helm Chart 时，在 values-public.yaml 配置文件中将 env.offlineMode 环境变量值设置为 true。

2. 安装 secured-cluster-services Helm Chart 时，在 values-public.yaml 配置文件中将 config.offlineMode 参数的值设置为 true。

5.2.2. 使用 roxctl CLI 启用离线模式

在使用 roxctl CLI 安装 Red Hat Advanced Cluster Security for Kubernetes 时，您可以启用离线模式。

流程

1.

如果您使用默认互联网连接的 registry (registry.redhat.io)，请在回答 镜像以使用 提示时提供推送 Red Hat Advanced Cluster Security for Kubernetes 镜像的位置：

```
Enter main image to use (if unset, the default will be used): <your_registry>/rhacs-main-rhel8:4.4.3
```

注意

默认镜像取决于您对 Enter default container images settings: 提示的回答：如果您输入 rhacs，默认选项，默认镜像为 registry.redhat.io/advanced-cluster-security/rhacs-main-rhel8:4.4.3。

```
Enter Scanner DB image to use (if unset, the default will be used): <your_registry>/rhacs-scanner-db-rhel8:4.4.3
```

```
Enter Scanner image to use (if unset, the default will be used): <your_registry>/rhacs-scanner-rhel8:4.4.3
```

2. 要启用离线模式，在回答 Enter whether to run StackRox in offline mode 提问时输入 true：

```
Enter whether to run StackRox in offline mode, which avoids reaching out to the internet (default: "false"): true
```

3. 之后，当您在 RHACS 门户的 Platform Configuration → Clusters 视图中将 Sensor 添加到远程集群中时，您必须在 Collector Image Repository 字段中指定 Collector 镜像名称。

5.3. 在离线模式下更新扫描器定义

扫描程序包含本地漏洞定义数据库。当 Red Hat Advanced Cluster Security for Kubernetes 以普通模式（连接到互联网）运行时，Scanner 从互联网获取新的漏洞定义并更新其数据库。

但是，当您以离线模式使用 Red Hat Advanced Cluster Security for Kubernetes 时，必须通过将其上传到 Central 来手动更新 Scanner 定义。

当 Red Hat Advanced Cluster Security for Kubernetes 以离线模式运行时，Scanner 会检查来自 Central 的新定义。如果有新的定义，Scanner 从 Central 下载新定义，将它们标记为默认值，然后使用更新的定义来扫描镜像。

以离线模式更新定义：

1. 下载定义。
2. 将定义上传到 Central。

5.3.1. 下载扫描器定义

如果您以离线模式运行 Red Hat Advanced Cluster Security for Kubernetes，您可以下载 Scanner 使用的漏洞定义数据库，然后将其上传到 Central。

前提条件

- 要下载 Scanner 定义，您需要有可访问互联网的系统。

流程

- 要下载定义，请执行以下操作之一：
  - 建议：从 RHACS 版本 4.4 开始，使用 roxctl scanner download-db --scanner-db-file scanner-vuln-updates.zip 命令来下载定义。
  - 进入 https://install.stackrox.io/scanner/scanner-vuln-updates.zip 下载定义。

其他资源

- roxctl scanner download-db

5.3.2. 将定义上传到 Central

要将 Scanner 定义上传到 Central，您可以使用 API 令牌或管理员密码。红帽建议在生产环境中使用身份验证令牌，因为每个令牌被分配特定的访问控制权限。

5.3.2.1. 使用 API 令牌将定义上传到 Central

您可以使用 API 令牌将 Scanner 使用的漏洞定义数据库上传到 Central。

前提条件

- 您必须具有带有管理员角色的 API 令牌。
- 您必须已安装了 roxctl 命令行界面(CLI)。

流程

1. 设置 ROX_API_TOKEN 和 ROX_CENTRAL_ADDRESS 环境变量：

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

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

2. 运行以下命令来上传定义文件：

```
$ roxctl scanner upload-db \
  -e "$ROX_CENTRAL_ADDRESS" \
  --scanner-db-file=<compressed_scanner_definitions.zip>
```

5.3.2.1.1. 其他资源

- 使用 roxctl CLI 进行身份验证

5.3.2.2. 使用管理员密码将定义上传到 Central

您可以使用 Red Hat Advanced Cluster Security for Kubernetes 管理员密码将 Scanner 使用的漏洞定义数据库上传到 Central。

前提条件

- 您必须具有管理员密码。
- 您必须已安装了 roxctl 命令行界面(CLI)。

流程

1. 设置 **ROX_CENTRAL_ADDRESS** 环境变量：

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

2. 运行以下命令来上传定义文件：

```
$ roxctl scanner upload-db \
  -p <your_administrator_password> \
  -e "$ROX_CENTRAL_ADDRESS" \
  --scanner-db-file=<compressed_scanner_definitions.zip>
```

5.4. 以离线模式更新内核支持软件包

收集器监控安全集群中每个节点的运行时活动。要监控活动，Collector 需要以 eBPF 程序形式的探测。

使用 CORE_BPF 集合方法时，探测不特定于任何内核版本，在更新底层内核后仍可使用。这个集合方法不要求您提供或更新支持软件包。

反之，当您使用集合方法 EBPF 时，探测特定于主机上安装的 Linux 内核版本。Collector 镜像包含一组内置探测，用于发行版本支持的内核。但是，后续的内核将需要较新的探测。

当 Red Hat Advanced Cluster Security for Kubernetes 以正常模式（连接到互联网）运行时，如果没有构建所需的探测，Collector 会自动下载新的探测。

在离线模式下，您可以手动下载包含所有最新且受支持的 Linux 内核版本探测的软件包，并将它们上传到 Central。然后，收集器从 Central 下载这些探测。

收集器按照以下顺序检查新探测：它会检查：

1. 现有的 Collector 镜像。
2. 内核支持软件包（如果您已向 Central 上传了一个）。

3. 互联网上可用的红帽操作服务器。收集器使用 Central 的网络连接来检查和下载探测。

如果 Collector 在检查后没有获得新的探测，它会报告 CrashLoopBackoff 事件。

如果您的网络配置限制出站流量，您可以手动下载包含所有最新和支持的 Linux 内核版本探测的软件包，并将其上传到 Central。然后，收集器从 Central 下载这些探测，从而避免任何出站互联网访问。

5.4.1. 下载内核支持软件包

如果您以离线模式运行 Red Hat Advanced Cluster Security for Kubernetes，您可以下载包含所有最新和支持的 Linux 内核版本探测的软件包，然后将其上传到 Central。

流程

- 从 https://install.stackrox.io/collector/support-packages/index.html 查看并下载可用支持软件包。内核支持列表根据 Red Hat Advanced Cluster Security for Kubernetes 版本对软件包进行分类。

5.4.2. 将内核支持软件包上传到 Central

您可以将内核支持软件包上传包含所有最新且受支持的 Linux 内核版本的探测到 Central。

前提条件

- 您必须具有带有管理员角色的 API 令牌。
- 您必须已安装了 roxctl 命令行界面(CLI)。

流程

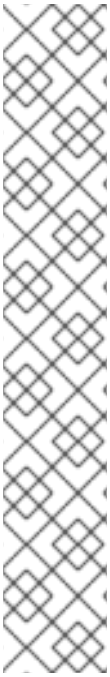
1. 设置 ROX_API_TOKEN 和 ROX_CENTRAL_ADDRESS 环境变量：

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

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

2. 运行以下命令来上传内核支持软件包：

```
$ roxctl collector support-packages upload <package_file> \
  -e "$ROX_CENTRAL_ADDRESS"
```



注意

- 当您上传包含上传到 Central 的内容的新支持软件包时，只会上传新文件。
- 当您上传一个新的支持软件包，其中包含名称相同的文件，但与 Central 上存在的内容不同，roxctl 会显示警告消息，且不会覆盖文件。
  - 您可以将 --overwrite 选项与 upload 命令一起使用来覆盖文件。
- 当您上传包含所需探测的支持软件包时，Central 不会发出任何出站请求（到互联网）以下载这个探测。Central 使用 support 软件包中的探测。

第 6 章 启用警报数据保留

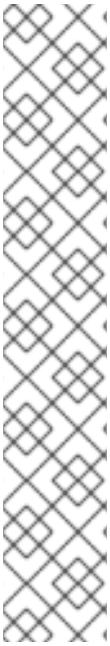
了解如何为 Red Hat Advanced Cluster Security for Kubernetes 警报配置保留周期。

使用 Red Hat Advanced Cluster Security for Kubernetes，您可以配置时间来保持历史警报存储。Red Hat Advanced Cluster Security for Kubernetes 随后在指定时间后删除旧的警报。

通过自动删除不再需要的警报，您可以节省存储成本。

您可以配置保留周期的警报包括：

- 运行时警报，未解析（主动）和解析。
- 不适用于当前部署的过时的部署时警报。



注意

- 数据保留设置会被默认启用。您可在安装后更改这些设置。
- 当您升级 Red Hat Advanced Cluster Security for Kubernetes 时，除非之前启用了数据保留设置，否则不会应用数据保留设置。
- 您可以使用 RHACS 门户或 API 配置警报保留设置。
- 删除过程每小时运行一次。目前，您无法更改它。

6.1. 配置警报数据保留

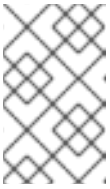
您可以使用 RHACS 门户配置警报保留设置。

前提条件

- 您必须具有带有 read 和 write 权限的 Config 角色，才能配置数据保留。

流程

1. 在 RHACS 门户中，进入 Platform Configuration → System Configuration。
2. 在 System Configuration 视图 标头中，单击 Edit。
3. 在 Data Retention Configuration 部分下，更新每种数据类型的天数：
   - 所有运行时冲突
   - 已解决 Deploy-Phase Violations
   - 用于删除部署的运行时冲突
   - 镜像没有部署长



注意

要永久保存数据类型，请将保留周期设置为 0 天。

4. 点 Save。

注意

要使用 Red Hat Advanced Cluster Security for Kubernetes API 配置警报数据保留，请参阅 API 参考文档中的 ConfigService 组中的 PutConfig API 和相关 API。

第 7 章 通过 HTTP 公开 RHACS 门户

启用未加密的 HTTP 服务器，通过入口控制器、第 7 层负载均衡器、Istio 或其他解决方案公开 RHACS 门户。

如果您使用入口控制器、Istio 或首选未加密的 HTTP 后端的 7 层负载均衡器，您可以配置 Red Hat Advanced Cluster Security for Kubernetes 来通过 HTTP 公开 RHACS 门户。这样做可让 RHACS 门户通过纯文本后端提供。

重要

要通过 HTTP 公开 RHACS 门户，您必须使用入口控制器、第 7 层负载均衡器或 Istio 来使用 HTTPS 加密外部流量。使用普通 HTTP 将 RHACS 门户直接公开给外部客户端是不安全的。

您可以在安装过程中或现有部署中通过 HTTP 公开 RHACS 门户。

7.1. 前提条件

- 要指定 HTTP 端点，您必须使用 < endpoints_spec >。它是以 < type>@<addr>:<port> 格式的单端点规格列表，其中：
  - type 为 grpc 或 http。在大多数用例中，使用 http 作为类型可以正常工作。对于高级用例，您可以使用 grpc 或省略其值。如果省略了 类型 的值，您可以在代理中配置两个端点，一个用于 gRPC，另一个用于 HTTP。这两个端点都指向 Central 上相同的公开 HTTP 端口。但是，大多数代理都不支持在同一外部端口上传输 gRPC 和 HTTP 流量。
  - addr 是要公开 Central 的 IP 地址。如果需要 HTTP 端点只能通过端口转发访问，可以省略它，也可以使用 localhost 或 127.0.0.1。
  - port 是 Central 在其中公开的端口。
  - 以下是几个有效的 < endpoints_spec> 值：
    - 8080

- http@8080
- :8081
- grpc@:8081
- localhost:8080
- http@localhost:8080
- http@8080,grpc@8081
- 8080, grpc@:8081, http@0.0.0.0:8082

7.2. 在安装过程中通过 HTTP 公开 RHACS 门户

如果您要使用 roxctl CLI 安装 Red Hat Advanced Cluster Security for Kubernetes，请使用带有 roxctl central generate interactive 命令的 --plaintext-endpoints 选项在安装过程中启用 HTTP 服务器。

流程

- 运行以下命令在互动安装过程中指定 HTTP 端点：

```
$ roxctl central generate interactive \
  --plaintext-endpoints=<endpoints_spec> 1
```

1 以 < type>@<addr>:<port> 的形式的端点规格。详情请查看先决条件部分。

7.3. 通过 HTTP 公开 RHACS 门户用于现有部署

您可以在现有 Red Hat Advanced Cluster Security for Kubernetes 部署中启用 HTTP 服务器。

流程

1. 创建补丁并定义 ROX_PLAINTEXT_ENDPOINTS 环境变量：

```
$ CENTRAL_PLAINTEXT_PATCH='
spec:
  template:
    spec:
      containers:
      - name: central
        env:
        - name: ROX_PLAINTEXT_ENDPOINTS
          value: <endpoints_spec> 1
'
```

1. 以 < type>@<addr>:<port> 的形式的端点规格。详情请查看先决条件部分。

2. 将 ROX_PLAINTEXT_ENDPOINTS 环境变量添加到 Central 部署中：

```
$ oc -n stackrox patch deploy/central -p "$CENTRAL_PLAINTEXT_PATCH"
```

第 8 章 为安全集群配置自动升级

您可以为每个安全集群自动化升级过程，并从 RHACS 门户查看升级状态。

通过自动执行每个安全集群的手动任务，自动升级自动升级，从而更轻松地保持最新状态。

通过自动升级，在升级 Central 后，在所有安全集群中升级 Sensor、Collector 和 Compliance 服务，会自动升级到最新版本。

Red Hat Advanced Cluster Security for Kubernetes 还允许从 RHACS 门户集中管理所有安全集群。新的集群视图显示有关所有安全集群、每个集群的 Sensor 版本和升级状态的信息。您还可以使用此视图有选择地升级安全集群或更改其配置。

注意

- 默认启用自动升级功能。
- 如果使用私有镜像 registry，您必须首先将 Sensor 和 Collector 镜像推送到私有 registry。
- Sensor 必须使用默认 RBAC 权限运行。
- 自动升级不会保留对集群中运行的任何 Red Hat Advanced Cluster Security for Kubernetes 服务所做的任何补丁。但是，它会保留添加到任何 Red Hat Advanced Cluster Security for Kubernetes 对象的所有标签和注解。
- 默认情况下，Red Hat Advanced Cluster Security for Kubernetes 在每个安全集群中创建一个名为 sensor-upgrader 的服务帐户。此帐户具有高特权，但仅在升级过程中使用。如果删除了这个帐户，Sensor 没有足够权限，您必须手动完成将来的升级。

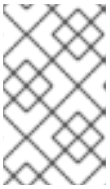
8.1. 启用自动升级

您可以为所有安全集群启用自动升级，以便在所有安全集群中自动将 Collector 和 Compliance 服务升级到最新版本。

流程

1. **在 RHACS 门户中，进入 Platform Configuration → Clusters。**

2. **打开 Automatically upgrade secured clusters 切换开关。**



注意

对于新安装，默认启用 Automatically upgrade secured clusters 切换。

8.2. 禁用自动升级

如果要手动管理安全集群升级，您可以禁用自动升级。

流程

1. **在 RHACS 门户中，进入 Platform Configuration → Clusters。**

2. **关闭 Automatically upgrade secured clusters 切换开关。**

注意

对于新安装，默认启用 Automatically upgrade secured clusters 切换。

8.3. 自动升级状态

Clusters 视图列出了所有集群及其升级状态。

升级状态	Description
Central 版本最多最新	安全集群运行与 Central 相同的版本。
可用的升级	Sensor 和 Collector 提供了新版本。
升级失败。重试升级。	以前的自动升级失败。

升级状态	Description
需要手动升级	Sensor 和 Collector 版本早于 2.5.29.0 版本。您必须手动升级安全集群。
pre-flight 检查完成	升级正在进行。在执行自动升级前，升级安装程序会运行 pre-flight 检查。在 pre-flight 检查过程中，安装程序会验证是否满足某些条件，然后只启动升级过程。

8.4. 自动升级失败

有时，Red Hat Advanced Cluster Security for Kubernetes 自动升级可能无法安装。当升级失败时，安全集群的状态信息将变为 Upgrade 失败。重试升级。要查看有关失败的更多信息，并了解升级失败的原因，您可以在 Clusters 视图中检查安全的集群行。

失败的一些常见原因是：

- 因为缺少或不可调度的镜像，sensor-upgrader 部署可能没有运行。
- pre-flight 检查可能会失败，因为 RBAC 权限不足，或者因为集群状态无法识别。如果您编辑了 Red Hat Advanced Cluster Security for Kubernetes 服务配置，或缺少 auto-upgrade.stackrox.io/component 标签，会出现这种情况。
- 执行升级可能会出现错误。如果发生这种情况，升级安装程序会自动尝试回滚升级。

注意

有时，回滚也可以失败。在这种情况下，查看集群日志以识别问题或联系支持。

在识别并修复升级失败的根本原因后，您可以使用 Retry Upgrade 选项来升级安全集群。

8.5. 从 RHACS 门户手动升级安全集群

如果您不想启用自动升级，您可以使用 Clusters 视图管理安全集群升级。

为安全集群手动触发升级：

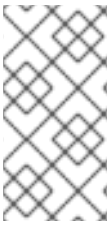
流程

1. 在 RHACS 门户中，进入 Platform Configuration → Clusters。
2. 在您要升级的集群的升级状态列中选择 Upgrade available 选项。
3. 要一次升级多个集群，请在您要更新的集群的 Cluster 列中选择复选框。
4. 单击 Upgrade。

第 9 章 配置从 RHACS 自动删除非活跃集群

Red Hat Advanced Cluster Security for Kubernetes (RHACS)提供了将您的系统配置为从 RHACS 中自动删除非活跃集群的选项，以便您可以只监控活跃集群。请注意，只有安装和执行带有 Central 的握手的集群最初会被监控。如果启用了这个功能，当 Central 在 Decomissioned cluster age 字段中配置的时间无法访问 Sensor 时，集群在 RHACS 中被视为非活跃。然后，Central 将不再监控非活跃的集群。您可以在 Platform Configuration → System Configuration 页面中配置 Decommissioned cluster age 字段。在配置此功能时，您可以为集群添加标签，以便 RHACS 继续监控集群，即使它变为非活跃状态。

默认禁用从 RHACS 中删除非活跃集群。要启用此设置，请在 Decommissioned cluster age 字段中输入非零数字，如以下步骤所述。Decommissioned cluster age 字段指示集群在被视为非活跃前可以保持无法访问的天数。当集群非活跃时，Clusters 页面会显示集群的状态。非活跃集群使用 不健康 标签表示，如果继续保持非活跃，窗口会显示从 RHACS 中删除集群的天数。从 RHACS 中删除集群后，该操作会记录在 Central 日志中作为一条 info 日志。



注意

在启用此设置后，在集群被删除前有一个 24 小时的宽限期。用于托管 Central 的集群永远不会被删除。

9.1. 配置集群停用

您可以将 RHACS 配置为从 RHACS 中自动删除非活跃集群。非活跃集群是那些已安装并执行具有 Central 的握手一次，但 Sensor 在指定时间段内无法访问。您还可以标记集群，以便在无法访问时不会删除它们。

流程

1. 在 RHACS 门户中，进入 Platform Configuration → System Configuration。
2. 在 System Configuration 标头中，点 Edit。
3. 在 Cluster deletion 部分，您可以配置以下字段：

- 停用的集群年龄 ：集群在考虑从 RHACS 中删除前无法访问的天数。如果集群中的 Central 无法达到这个天数，集群及其所有资源都会从 RHACS 中删除。要禁用此功能（这是默认行为），请在此字段中输入 0。要启用此功能，请输入非零数字，如 90，以配置无法访问的天数。

- 忽略具有标签：要阻止集群被删除的集群，您可以通过输入本节中的键和值来配置标签。具有此标签的集群不会被删除，即使它们在 Decommissioned cluster age 字段中设定的天数内无法访问。

  - Key ：输入用于集群的标签。

  - Value ：输入与键关联的值。

  例如，若要从移除保留生产集群，您可以配置 cluster-type 键和 production 的值。

注意

在 Cluster deletion 部分，点 Clusters，其具有 Sensor Status: Unhealthy 以进入 Clusters 列表页面。本页被过滤，以显示可能删除以及从 RHACS 中删除的时间线的非活跃集群。

4. 点 Save。

注意

要使用 API 查看并配置此选项，对于 /v1/config 和 /v1/config/private 端点，在请求的正文中使用 decommissionedClusterRetention 设置。如需更多信息，请参阅 RHACS 门户中的 Help → API 参考中的 ConfigService 对象的 API 文档。

9.2. 查看非活跃集群

非活跃集群是安装和执行具有 Central 的握手的集群，但 Sensor 在指定时间段内无法访问。使用这个流程查看这些集群的列表。

流程

1. 在 RHACS 门户中，进入 Platform Configuration → System Configuration。

2. 在 Cluster deletion 部分，点 Clusters，其具有 Sensor Status: Unhealthy 以进入 Clusters 列表页面。本页被过滤，以显示可以从 RHACS 中删除的非活跃集群以及删除的时间

线。

注意

如果在集群被视为非活跃后启用此功能，则从集群变为非活跃的时间开始删除的天数，而不是从启用该功能的时间开始。如果您不希望删除任何非活跃集群，您可以配置标签，如"配置集群停用"部分中所述。当系统删除非活跃集群时，会忽略具有这些标签的集群。

第 10 章 为外部网络访问配置代理

如果您的网络配置限制了通过代理的出站流量，您可以在 Red Hat Advanced Cluster Security for Kubernetes 中配置代理设置，以通过代理路由流量。

当您将代理与 Red Hat Advanced Cluster Security for Kubernetes 搭配使用时：

- 来自 Central 和 Scanner 的所有传出 HTTP、HTTPS 和其他 TCP 流量都通过代理。
- Central 和 Scanner 之间的流量不会通过代理。
- 代理配置不会影响其他 Red Hat Advanced Cluster Security for Kubernetes 组件。
- 当您没有使用离线模式时，在安全集群中运行的 Collector 需要在运行时下载额外的 eBPF 探测：
  - 收集器尝试通过联系 Sensor 来下载它们。
  - 然后，Sensor 将这个请求转发到 Central。
  - Central 使用代理在 https://collector-modules.stackrox.io 找到模块或探测。

10.1. 在现有部署上配置代理

要在现有部署中配置代理，您必须将 proxy-config secret 导出为 YAML 文件，更新该文件中的代理配置，并将其上传为 secret。

注意

如果您在 OpenShift Container Platform 集群上配置了全局代理，Operator Lifecycle Manager (OLM)会自动配置使用集群范围代理管理的 Operator。但是，您还可以配置已安装的 Operator 来覆盖全局代理或注入自定义证书颁发机构(CA)证书。

如需更多信息，请参阅 Operator Lifecycle Manager 中的配置代理支持。

流程

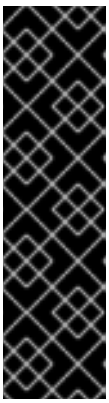
1. 将现有 secret 保存为 YAML 文件：

```
$ oc -n stackrox get secret proxy-config \
  -o go-template='{{index .data "config.yaml" | \
  base64decode}}{{"\n"}}' > /tmp/proxy-config.yaml
```

2. 编辑 YAML 配置文件中要修改的字段，如在安装过程中 Configure proxy 中指定的。

3. 保存更改后，运行以下命令替换 secret：

```
$ oc -n stackrox create secret generic proxy-config \
  --from-file=config.yaml=/tmp/proxy-config.yaml -o yaml --dry-run | \
  oc label -f - --local -o yaml app.kubernetes.io/name=stackrox | \
  oc apply -f -
```



重要

- 您必须至少等待 1 分钟，直到 OpenShift Container Platform 将更改传播到 Central 和 Scanner。
- 如果在更改代理配置后看到传出连接的问题，您必须重启您的 Central 和 Scanner pod。

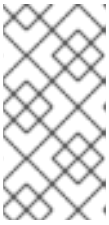
10.2. 在安装过程中配置代理

当使用 roxctl 命令行界面(CLI)或 Helm 安装 Red Hat Advanced Cluster Security for Kubernetes 时，您可以在安装过程中指定代理配置。

当使用 roxctl central generate 命令运行安装程序时，安装程序会为您的环境生成 secret 和部署配置文件。您可以通过编辑生成的配置 secret (YAML)文件来配置代理。目前，您无法使用 roxctl CLI 配置代理。配置存储在 Kubernetes secret 中，它由 Central 和 Scanner 共享。

流程

1. 从部署捆绑包目录中打开配置文件 central/proxy-config-secret.yaml。



注意

如果您使用 Helm，则配置文件位于 central/templates/proxy-config-secret.yaml。

2. 编辑配置文件中要修改的字段：

```
apiVersion: v1
kind: Secret
metadata:
  namespace: stackrox
  name: proxy-config
type: Opaque
stringData:
  config.yaml: |- (1)
    # # NOTE: Both central and scanner should be restarted if this secret is changed.
    # # While it is possible that some components will pick up the new proxy
configuration
    # # without a restart, it cannot be guaranteed that this will apply to every possible
    # # integration etc.
    # url: http://proxy.name:port (2)
    # username: username (3)
    # password: password (4)
    # # If the following value is set to true, the proxy wil NOT be excluded for the default
hosts:
    # # - *.stackrox, *.stackrox.svc
    # # - localhost, localhost.localdomain, 127.0.0.0/8, ::1
    # # - *.local
    # omitDefaultExcludes: false
    # excludes:  # hostnames (may include * components) for which you do not (5)
    # # want to use a proxy, like in-cluster repositories.
    # - some.domain
    # # The following configuration sections allow specifying a different proxy to be
used for HTTP(S) connections.
    # # If they are omitted, the above configuration is used for HTTP(S) connections as
well as TCP connections.
    # # If only the `http` section is given, it will be used for HTTPS connections as well.
    # # Note: in most cases, a single, global proxy configuration is sufficient.
```

```
# http:
#   url: http://http-proxy.name:port 6
#   username: username 7
#   password: password 8
# https:
#   url: http://https-proxy.name:port 9
#   username: username 10
#   password: password 11
```

3 4 7 8 10 11

在开始以及在 http 和 https 部分中，添加 username 和 password 是可选的。

2 6 9

url 选项支持以下 URL 方案：

- http:// 用于 HTTP 代理。
- https:// 用于支持 TLS 的 HTTP 代理。
- SOCKS5 代理的 socks5://。

5

excludes 列表可以包含 DNS 名称（带有或不带有 * 通配符）、IP 地址或 IP 块（例如：10.0.0.0/8）。此列表中的值适用于所有传出连接，而不考虑协议。

1

stringData 部分中的 |- 行表示配置数据的开头。

注意

- 首次打开文件时，所有值都会被注释掉（在每行的开头使用 # 符号）。以双 hash 符号（# #）开头的行包含配置键的说明。
- 确保在编辑字段时，您可以维护相对于 config.yaml: |- 行的两个空格的缩进级别。

3. 编辑配置文件后，您可以继续正常安装。更新的配置指示 Red Hat Advanced Cluster Security for Kubernetes 使用在提供的地址和端口号上运行的代理。

第 11 章 生成诊断捆绑包

您可以生成诊断捆绑包并发送这些数据，以便支持团队能够深入了解 Red Hat Advanced Cluster Security for Kubernetes 组件的状态和健康状况。

红帽可能会要求您在调查 Red Hat Advanced Cluster Security for Kubernetes 的问题时发送诊断捆绑包。您可以生成诊断捆绑包，并在发送前检查其数据。

注意

诊断捆绑包是未加密的，具体取决于您环境中的集群数量，捆绑包大小介于 100 KB 到 1 MB 之间。始终使用加密通道将这些数据传回红帽。

11.1. 诊断捆绑包数据

当您生成诊断捆绑包时，它包括以下数据：

- 中央堆配置文件.
- 系统日志 ：所有 Red Hat Advanced Cluster Security for Kubernetes 组件的日志（用于最后 20 分钟）和最近崩溃组件的日志（从崩溃前到 20 分钟）。系统日志取决于您的环境大小。对于大型部署，数据仅包含关键错误的组件的日志文件，如高重启计数。
- Red Hat Advanced Cluster Security for Kubernetes 组件的 YAML 定义：这些数据不包括 Kubernetes secret。
- OpenShift Container Platform 或 Kubernetes 事件：有关与 stackrox 命名空间中的对象相关的事件详情。
- 在线 Telemetry 数据，其中包括：
  - 存储信息 ：有关数据库大小以及附加卷中的可用空间大小的详细信息。
  - Red Hat Advanced Cluster Security for Kubernetes 组件健康信息：有关 Red Hat

Advanced Cluster Security for Kubernetes 组件版本、其内存用量以及任何报告的错误。

- 粗颗粒的使用统计数据：有关 API 端点调用数量和报告的错误状态的信息。它不包括 API 请求中发送的实际数据。

- 节点信息 ：有关每个安全集群中节点的详细信息。它包括内核和操作系统版本、资源压力和污点。

- 环境信息 ：有关每个安全集群的详细信息，包括 Kubernetes 或 OpenShift Container Platform 版本、Istio 版本（如果适用）、云供应商类型和其他类似信息。

11.2. 使用 RHACS 门户生成诊断捆绑包

您可以使用 RHACS 门户中的系统健康仪表板生成诊断捆绑包。

先决条件

- 要生成诊断捆绑包，您需要 DebugLogs 资源的 read 权限。

流程

1. 在 RHACS 门户中，选择 Platform Configuration → System Health。
2. 在 System Health view 标头上，点 Generate Diagnostic Bundle。
3. 对于 Filter by clusters 下拉菜单，选择要为其生成诊断数据的集群。
4. 对于 Filter by starting time，指定您要包含诊断数据的日期和时间（以 UTC 格式）。
5. 点 Download Diagnostic Bundle。

11.3. 使用 ROXCTL CLI 生成诊断捆绑包

您可以使用 roxctl CLI 使用 Red Hat Advanced Cluster Security for Kubernetes (RHACS)管理员密码或 API 令牌和中央地址生成诊断捆绑包。

先决条件

- 要生成诊断捆绑包，您需要对 Administration 资源具有读权限。这是比版本 3.73.0 更早的 DebugLogs 资源版本所必需的。
- 您必须已配置了 RHACS 管理员密码或 API 令牌和中央地址。

流程

- 要使用 RHACS 管理员密码生成诊断捆绑包，请执行以下步骤：

  1. 运行以下命令来配置 ROX_PASSWORD 和 ROX_CENTRAL_ADDRESS 环境变量：

     ```
     $ export ROX_PASSWORD=<rox_password> && export
     ROX_CENTRAL_ADDRESS=<address>:<port_number> 1
     ```

     1 对于 <rox_password>，请指定 RHACS 管理员密码。

  2. 运行以下命令，以使用 RHACS 管理员密码生成诊断捆绑包：

     ```
     $ roxctl -e "$ROX_CENTRAL_ADDRESS" -p "$ROX_PASSWORD" central debug
     download-diagnostics
     ```

- 要使用 API 令牌生成诊断捆绑包，请执行以下步骤：

  1. 运行以下命令来配置 ROX_API_TOKEN 环境变量：

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

  2. 运行以下命令，以使用 API 令牌生成诊断捆绑包：

```
$ roxctl -e "$ROX_CENTRAL_ADDRESS" central debug download-diagnostics
```

第 12 章 配置端点

了解如何使用 YAML 配置文件为 Red Hat Advanced Cluster Security for Kubernetes (RHACS)配置端点。

您可以使用 YAML 配置文件来配置公开的端点。您可以使用此配置文件为 Red Hat Advanced Cluster Security for Kubernetes 定义一个或多个端点，并自定义每个端点的 TLS 设置，或为特定端点禁用 TLS。您还可以定义是否需要客户端身份验证，以及要接受哪些客户端证书。

12.1. 自定义 YAML 配置

Red Hat Advanced Cluster Security for Kubernetes 使用 YAML 配置作为 ConfigMap，以便更轻松地更改和管理配置。

使用自定义 YAML 配置文件时，您可以为每个端点配置以下内容：

- 要使用的协议，如 HTTP、gRPC 或两者。
- 启用或禁用 TLS。
- 指定服务器证书。
- 要信任客户端身份验证的客户端证书颁发机构(CA)。
- 指定是否需要客户端证书身份验证(mTLS)。

您可以使用配置文件在安装过程中或现有 Red Hat Advanced Cluster Security for Kubernetes 实例指定端点。但是，如果您公开除默认端口 8443 以外的任何其他端口，您必须创建允许这些额外端口上流量的网络策略。

以下是 Red Hat Advanced Cluster Security for Kubernetes 的 endpoint .yaml 配置文件示例：

```
# Sample endpoints.yaml configuration for Central.
```

```
#
# # CAREFUL: If the following line is uncommented, do not expose the default endpoint on
port 8443 by default.
# #         This will break normal operation.
# disableDefault: true # if true, do not serve on :8443 (1)
endpoints: (2)
  # Serve plaintext HTTP only on port 8080
  - listen: ":8080" (3)
    # Backend protocols, possible values are 'http' and 'grpc'. If unset or empty, assume both.
    protocols: (4)
      - http
    tls: (5)
      # Disable TLS. If this is not specified, assume TLS is enabled.
      disable: true (6)
  # Serve HTTP and  gRPC for sensors only on port 8444
  - listen: ":8444" (7)
    tls: (8)
      # Which TLS certificates to serve, possible values are 'service' (For  service certificates
that Red Hat Advanced Cluster Security for Kubernetes generates)
      # and 'default' (user-configured default TLS certificate). If unset or empty, assume both.
      serverCerts: (9)
        - default
        - service
      # Client authentication settings.
      clientAuth: (10)
        # Enforce TLS client authentication. If unset, do not enforce, only request certificates
        # opportunistically.
        required: true (11)
        # Which TLS client CAs to serve, possible values are 'service' (CA for service
        # certificates that Red Hat Advanced Cluster Security for Kubernetes generates)
and 'user' (CAs for PKI auth providers). If unset or empty, assume both.
        certAuthorities: (12)
        # if not set, assume ["user", "service"]
          - service
```

(1)

使用 true 禁用默认端口号 8443 的风险。默认值为 false; 将它改为 true 可能会破坏现有功能。

(2)

用于公开 Central 的额外端点列表。

(3) (7)

要侦听的地址和端口号。如果使用 端点，则必须指定这个值。您可以使用格式 端口、:port，或 address:port 来指定值。例如，

- 8080 或 :8080 - 监听所有接口中的端口 8080。

- 0.0.0.0:8080 - 监听所有 IPv4 （非 IPv6）接口上的端口 8080。
- 127.0.0.1:8080 - 仅监听本地回送设备上的端口 8080。

4

用于指定端点的协议。可接受值为 http 和 grpc。如果没有指定值，则 Central 侦听指定端口上的 HTTP 和 gRPC 流量。如果要只为 RHACS 门户公开端点，请使用 http。但是，您将无法将端点用于服务到服务通信或 roxctl CLI，因为这些客户端需要 gRPC 和 HTTP。红帽建议不要指定这个键的值，为端点启用 HTTP 和 gRPC 协议。如果只想将端点限制为 Red Hat Advanced Cluster Security for Kubernetes 服务，请使用 clientAuth 选项。

5 8

使用它来指定端点的 TLS 设置。如果没有指定值，Red Hat Advanced Cluster Security for Kubernetes 会使用以下所有嵌套键的默认设置启用 TLS。

6

使用 true 在指定端点中禁用 TLS。默认值为 false。当设置为 true 时，您无法为 serverCerts 和 clientAuth 指定值。

9

指定配置服务器 TLS 证书的源列表。serverCerts 列表是独立于顺序的，这意味着列表中的第一个项目决定了 Central 默认使用的证书，如果没有匹配的 SNI （服务器名称 Indication）。您可以使用此选项指定多个证书，Central 会自动根据 SNI 选择正确的证书。可接受值为：

- 默认 ：如果已存在，请使用已配置的自定义 TLS 证书。
- 服务 ： 使用 Red Hat Advanced Cluster Security for Kubernetes 生成的内部服务证书。

10

使用它来配置启用了 TLS 的端点客户端证书身份验证的行为。

11

使用 true 只允许具有有效客户端证书的客户端。默认值为 false。您可以将 true 与服务的 certAuthorities 设置一起使用，以便只允许 Red Hat Advanced Cluster Security for Kubernetes 服务连接到此端点。

12

验证客户端证书的 CA 列表。默认值为 ["service", "user"]。certAuthorities 列表是按顺序独立的，这意味着此列表中项目的位置无关紧要。另外，将它设置为空 list [] 会禁用端点的客户端证书身份验证，这与保留这个值未设置不同。可接受值为：

- Service: Red Hat Advanced Cluster Security for Kubernetes 生成的服务证书的 CA。
- 用户 ：由 PKI 身份验证提供程序配置的 CA。

12.2. 在新安装过程中配置端点

当使用 roxctl CLI 安装 Red Hat Advanced Cluster Security for Kubernetes 时，它会创建一个名为 central-bundle 的文件夹，其中包含部署 Central 所需的 YAML 清单和脚本。

流程

1. 生成 central-bundle 后，打开 ./central-bundle/central/02-endpoints-config.yaml 文件。
2. 在这个文件中，将自定义 YAML 配置添加到密钥 endpoint .yaml 的 data: 部分。确保您为 YAML 配置维护 4 个空格缩进。
3. 照常继续安装说明。Red Hat Advanced Cluster Security for Kubernetes 使用指定的配置。

注意

如果您公开了除默认端口 8443 以外的任何其他端口，您必须创建允许这些额外端口上流量的网络策略。

12.3. 为现有实例配置端点

您可以为 Red Hat Advanced Cluster Security for Kubernetes 的现有实例配置端点。

流程

1. 下载现有配置映射：

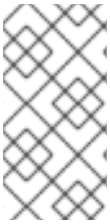
```
$ oc -n stackrox get cm/central-endpoints -o go-template='{{index .data
"endpoints.yaml"}}'  > <directory_path>/central_endpoints.yaml
```

2. 在下载的 central_endpoints.yaml 文件中，指定您的自定义 YAML 配置。

3. 上传并应用修改后的 central_endpoints.yaml 配置文件：

```
$ oc -n stackrox create cm central-endpoints --from-file=endpoints.yaml=<directory-
path>/central-endpoints.yaml -o yaml --dry-run | \
oc label -f - --local -o yaml app.kubernetes.io/name=stackrox | \
oc apply -f -
```

4. 重启 Central。



注意

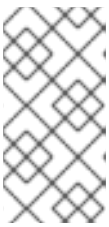
如果您公开了除默认端口 8443 以外的任何其他端口，您必须创建允许这些额外端口上流量的网络策略。

12.3.1. 重启 Central 容器

您可以通过终止 Central 容器或删除 Central pod 来重启 Central 容器。

流程

- 运行以下命令以终止 Central 容器：



注意

您必须至少等待 1 分钟，直到 OpenShift Container Platform 传播您的更改并重启 Central 容器。

```
$ oc -n stackrox exec deploy/central -c central -- kill 1
```

- 或者，运行以下命令来删除 Central pod：

```
$ oc -n stackrox delete pod -lapp=central
```

12.4. 启用通过自定义端口的流量流

如果要将端口公开给同一集群或入口控制器中运行的另一个服务，则必须只允许来自集群中的服务的流量或从入口控制器的代理进行流量。否则，如果您使用负载均衡器服务公开端口，您可能需要允许来自所有源的流量，包括外部源。使用本节中列出的流程允许来自所有源的流量。

流程

1. 克隆 allow-ext-to-central Kubernetes 网络策略：

```
$ oc -n stackrox get networkpolicy.networking.k8s.io/allow-ext-to-central -o yaml > <directory_path>/allow-ext-to-central-custom-port.yaml
```

2. 使用它作为创建网络策略的引用，在该策略中指定您要公开的端口号。确保在 YAML 文件的 metadata 部分中更改网络策略的名称，使其不会影响内置的 allow-ext-to-central 策略。

第 13 章 监控 RHACS

您可以使用 Red Hat OpenShift 的内置监控或使用自定义 Prometheus 监控来监控 Red Hat Advanced Cluster Security for Kubernetes (RHACS)。

如果您将 RHACS 与 Red Hat OpenShift 搭配使用，OpenShift Container Platform 包括一个预配置、预安装和自我更新的监控堆栈，为核心平台组件提供监控。RHACS 通过加密和验证的端点向 Red Hat OpenShift 监控公开指标。

13.1. 使用 RED HAT OPENSHIFT 进行监控

默认启用 Red Hat OpenShift 监控。这个默认行为不需要配置。

重要

如果您之前使用 Prometheus Operator 配置监控，请考虑删除自定义 ServiceMonitor 资源。RHACS 附带了一个预配置的 ServiceMonitor，用于 Red Hat OpenShift 监控。多个 ServiceMonitor 可能会导致重复的 scraping。

Scanner 不支持使用 Red Hat OpenShift 进行监控。如果要监控扫描器，您必须首先禁用默认的 Red Hat OpenShift 监控。然后，配置自定义 Prometheus 监控。

有关禁用 Red Hat OpenShift 监控的更多信息，请参阅"使用 RHACS Operator 为 Central 服务禁用 Red Hat OpenShift 监控"或"使用 Helm 禁用中央服务的 Red Hat OpenShift 监控"。有关配置 Prometheus 的更多信息，请参阅"监控自定义 Prometheus"。

13.2. 使用自定义 PROMETHEUS 监控

Prometheus 是一个开源监控和警报平台。您可以使用它来监控 RHACS 的 Central 和 Sensor 组件的健康状态和可用性。当您启用监控时，RHACS 在端口号 9090 和允许到该端口的入站连接上创建一个新的监控服务。

注意

此监控服务公开不由 TLS 加密且没有授权的端点。仅在您不想使用 Red Hat OpenShift 监控时才使用它。

在使用自定义 Prometheus 监控前，如果有 Red Hat OpenShift，则必须禁用默认的监控。如果使用 Kubernetes，则不需要执行此步骤。

13.2.1. 使用 RHACS Operator 为中央服务禁用 Red Hat OpenShift 监控

要使用 Operator 禁用默认监控，请更改 Central 自定义资源的配置，如下例所示。如需有关配置选项的更多信息，请参阅"添加资源"部分中的"Central 配置选项使用 Operator"。

流程

1. 在 OpenShift Container Platform Web 控制台中，进入 Operators → Installed Operators 页面。
2. 从安装的 Operator 列表中选择 RHACS Operator。
3. 点 Central 选项卡。
4. 从 Central 实例列表中，点您要为其启用监控的 Central 实例。
5. 点 YAML 选项卡并更新 YAML 配置，如下例所示：

```
monitoring:
  openshift:
    enabled: false
```

13.2.2. 使用 Helm 为 Central 服务禁用 Red Hat OpenShift 监控

要使用 Helm 禁用默认监控，请更改 central-services Helm Chart 中的配置选项。有关配置选项的更多信息，请参阅"添加资源"部分中的文档。

流程

1. 使用以下值更新配置文件：

```
monitoring.openshift.enabled: false
```

2. 运行 helm upgrade 命令并指定配置文件。

13.2.3. 使用 RHACS Operator 监控 Central 服务

您可以通过更改 Central 自定义资源的配置来监控 Central 服务、Central 和 Scanner。如需有关配置选项的更多信息，请参阅"添加资源"部分中的"Central 配置选项使用 Operator"。

流程

1. 在 OpenShift Container Platform Web 控制台中，进入 Operators → Installed Operators 页面。
2. 从安装的 Operator 列表中选择 Red Hat Advanced Cluster Security for Kubernetes Operator。
3. 点 Central 选项卡。
4. 从 Central 实例列表中，点您要为其启用监控的 Central 实例。
5. 点 YAML 选项卡并更新 YAML 配置：
   - 对于监控 Central，请为 Central 自定义资源启用 central.monitoring.exposeEndpoint 配置选项。
   - 对于监控扫描器，请为 Central 自定义资源启用 scanner.monitoring.exposeEndpoint 配置选项。
6. 点击 Save。

13.3. 使用 HELM 监控 CENTRAL 服务

您可以通过更改 central-services Helm Chart 中的配置选项来监控 Central 服务、Central 和 Scanner。如需更多信息，请参阅"添加资源"部分的"部署 central-services Helm Chart"后

的"Changing 配置选项"。

流程

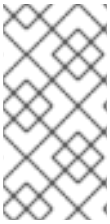
1. 使用以下值更新 values-public.yaml 配置文件：

```
central.exposeMonitoring: true
scanner.exposeMonitoring: true
```

2. 运行 helm upgrade 命令并指定配置文件。

13.3.1. 使用 Prometheus 服务监控器监控 Central

如果使用 Prometheus Operator，您可以使用服务监控器从 Red Hat Advanced Cluster Security for Kubernetes (RHACS)中提取指标。



注意

如果没有使用 Prometheus operator，您必须编辑 Prometheus 配置文件以从 RHACS 接收数据。

流程

1. 使用以下内容创建新的 servicemonitor.yaml 文件：

```
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  name: prometheus-stackrox
  namespace: stackrox
spec:
  endpoints:
    - interval: 30s
      port: monitoring
      scheme: http
  selector:
    matchLabels:
      app.kubernetes.io/name: <stackrox-service> 1
```

1. 标签必须与要监控的 Service 资源匹配。例如，中央 或 扫描程序。

2. 将 YAML 应用到集群：

```
$ oc apply -f servicemonitor.yaml 1
```

1 如果使用 Kubernetes，请输入 kubectl 而不是 oc。

验证

- 运行以下命令来检查服务监控器的状态：

```
$ oc get servicemonitor --namespace stackrox 1
```

1 如果使用 Kubernetes，请输入 kubectl 而不是 oc。

13.4. 其他资源

- 使用 Operator 的中央配置选项
- 在部署 central-services Helm Chart 后更改配置选项
- Helm 文档

第 14 章 配置审计日志记录

Red Hat Advanced Cluster Security for Kubernetes 提供了审计日志记录功能，可用于检查 Red Hat Advanced Cluster Security for Kubernetes 中所做的所有更改。审计日志捕获所有 PUT 和 POST 事件，这些事件会修改 Red Hat Advanced Cluster Security for Kubernetes。使用此信息对问题进行故障排除或记录重要事件，如对角色和权限的更改。使用审计日志记录时，您可以了解 Red Hat Advanced Cluster Security for Kubernetes 上发生的所有正常和异常事件。

注意

默认情况下不启用审计日志记录。您必须手动启用审计日志记录。

警告

目前，没有消息发送保证用于审计日志消息。

14.1. 启用审计日志记录

当您启用审计日志记录时，每次有修改时，Red Hat Advanced Cluster Security for Kubernetes 会向配置的系统发送 HTTP POST 信息(JSON 格式)。

前提条件

- 配置 Splunk 或其他 webhook 接收器来处理 Red Hat Advanced Cluster Security for Kubernetes 日志消息。
- 您必须在角色的 Notifiers 资源上启用 写入权限。

流程

1. 在 RHACS 门户中，进入 Platform Configuration → Integrations。
2. 向下滚动到 Notifier Integrations 部分，然后选择 Generic Webhook 或 Splunk。

3. 填写所需信息，再打开 **Enable Audit Logging** 切换开关。

14.2. 审计日志消息示例

日志消息的格式如下：


```
{
 "headers": {
  "Accept-Encoding": [
   "gzip"
  ],
  "Content-Length": [
   "586"
  ],
  "Content-Type": [
   "application/json"
  ],
  "User-Agent": [
   "Go-http-client/1.1"
  ]
 },
 "data": {
  "audit": {
   "interaction": "CREATE",
   "method": "UI",
   "request": {
    "endpoint": "/v1/notifiers",
    "method": "POST",
    "source": {
     "requestAddr": "10.131.0.7:58276",
     "xForwardedFor": "8.8.8.8",
    },
    "sourceIp": "8.8.8.8",
    "payload": {
     "@type": "storage.Notifier",
     "enabled": true,
     "generic": {
      "auditLoggingEnabled": true,
      "endpoint": "http://samplewebhookserver.com:8080"
     },
     "id": "b53232ee-b13e-47e0-b077-1e383c84aa07",
     "name": "Webhook",
     "type": "generic",
     "uiEndpoint": "https://localhost:8000"
    }
   },
   "status": "REQUEST_SUCCEEDED",
   "time": "2019-05-28T16:07:05.500171300Z",
   "user": {
    "friendlyName": "John Doe",
    "role": {
```

```
        "globalAccess": "READ_WRITE_ACCESS",
        "name": "Admin"
      },
      "username": "john.doe@example.com"
    }
  }
 }
}
```

请求的源 IP 地址显示在源参数中，这有助于调查审计日志请求并识别其原始卷。

要确定请求的源 IP 地址，RHACS 使用以下参数：

- XForwardedFor ：X-Forwarded-For 标头。
- requestAddr ：远程地址标头。
- SourceIp ：HTTP 请求的 IP 地址。

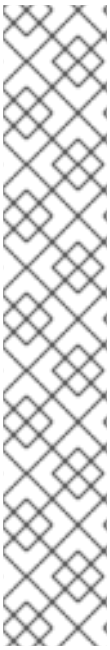
重要

源 IP 地址的确定取决于您如何向外部公开 Central。您可以考虑以下选项：

- 如果您在负载均衡器后面公开 Central，例如，如果您使用 Kubernetes External Load Balancer 服务类型在 Google Kubernetes Engine (GKE)或 Amazon Elastic Kubernetes Service (Amazon EKS)上运行 Central，请参阅 保留客户端源 IP。
- 如果您在 Ingress Controller 后面公开了使用 X-Forwarded-For 标头来转发请求 的 Central，则不需要进行任何配置更改。
- 如果使用 TLS passthrough 路由公开 Central，则无法确定客户端的源 IP 地址。源参数中会显示集群内部 IP 地址作为客户端的源 IP 地址。

第 15 章 配置 API 令牌

Red Hat Advanced Cluster Security for Kubernetes (RHACS)需要 API 令牌进行一些系统集成、身份验证流程和系统功能。您可以使用 RHACS Web 界面配置令牌。



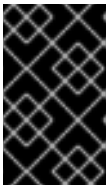
注意

- 为防止特权升级，在创建新令牌时，您的权限将限制您可以分配给该令牌的权限。例如，如果您只有 Integration 资源的 read 权限，则无法创建 具有写入权限的令牌。
- 如果您希望自定义角色为其他用户创建令牌，则必须为该自定义角色分配所需的权限。
- 将短期令牌用于机器到机器的通信，如 CI/CD 管道、脚本和其他自动化。另外，使用 roxctl central login 命令进行人工到机器通信，如 roxctl CLI 或 API 访问。

15.1. 创建 API 令牌

流程

1. 在 RHACS 门户中，进入 Platform Configuration → Integrations。
2. 滚动到 Authentication Tokens 类别，然后点 API Token。
3. 点 Generate Token。
4. 输入令牌的名称并选择提供所需访问级别的角色（例如： Continuous Integration 或 Sensor Creator）。
5. 点 Generate。



重要

复制生成的令牌并安全地存储它。您将无法再次查看它。

其他资源

- 使用身份验证供应商与 roxctl 进行身份验证
- 配置短期访问

15.2. 关于 API 令牌过期

API 令牌自创建日期起一年后过期。RHACS 在 web 界面中提醒您，当令牌在不到一周时将日志消息发送到 Central。日志消息进程每小时运行一次。每天，进程会列出即将过期的令牌，并为每个令牌创建一个日志消息。日志消息每天发出一次，并显示在 Central 日志中。

日志具有以下格式，如下例所示：

> **Warn: API Token [token name] (ID [token ID]) will expire in less than X days.**

您可以通过配置下表中显示的环境变量来更改日志消息进程的默认设置：

环境变量	默认值	描述
ROX_TOKEN_EXPIRATION_NOTIFIER_INTERVAL	1h (1 小时)	列出令牌并创建日志将运行日志消息后台循环的频率。
ROX_TOKEN_EXPIRATION_NOTIFIER_BACKOFF_INTERVAL	24h (1 天)	循环列出令牌和问题通知的频率。
ROX_TOKEN_EXPIRATION_DETECTION_WINDOW	168H (1 week)	导致生成通知的令牌过期前的时间段。

第 16 章 使用声明配置

通过声明性配置，您可以通过将其存储在存储库中的文件中来更新配置，并将其应用到系统。例如，如果您使用 GitOps 工作流，声明配置很有用。目前，您可以使用 Red Hat Advanced Cluster Security for Kubernetes (RHACS)中的声明配置进行身份验证和授权资源，如身份验证供应商、角色、权限集和访问范围。

要使用声明性配置，您可以创建 YAML 文件，其中包含有关身份验证和授权资源的配置信息。这些文件或配置通过使用中央安装过程中的挂载点添加到 RHACS 中。有关安装 RHACS 时配置挂载点的更多信息，请参阅"附加资源"部分中的安装文档。

与声明性配置一起使用的配置文件存储在配置映射或 secret 中，具体取决于资源类型。将身份验证供应商的配置存储在 secret 中以提高安全性。您可以在配置映射中存储其他配置。

单个配置映射或 secret 可以包含多个资源类型的配置。这可让您限制 Central 实例的卷挂载数量。

16.1. 从声明性配置创建的资源的限制

因为资源可以引用其他资源（例如，角色可以引用权限集和访问范围），所以应用以下限制：

- 声明性配置只能引用由声明性或系统 RHACS 资源创建的资源；例如，一个资源，如 Admin 或 analyst 系统角色或权限集。
- 资源之间的所有引用都使用名称来识别资源，因此同一资源类型中的所有名称都必须是唯一的。
- 从声明性配置创建的资源只能通过更改声明性配置文件来修改或删除。您不能使用 RHACS 门户或 API 更改这些资源。

16.2. 创建声明性配置

使用 roxctl 创建存储配置的 YAML 文件，从文件中创建配置映射并应用配置映射。

前提条件

-

您已在安装 Central 过程中为配置映射或 secret 添加了挂载。在本例中，配置映射名为 "declarative-configs"。如需更多信息，请参阅"附加资源"部分中列出的安装文档。

流程

1. 运行以下命令来创建权限集。本例创建一个名为 "restricted" 的权限集，并保存为 permission-set.yaml 文件。它为 Administration 资源设置读写访问权限，以及对 Access 资源的读访问权限。

```
$ roxctl declarative-config create permission-set \
--name="restricted" \
--description="Restriction permission set that only allows \
access to Administration and Access resources" \
--resource-with-access=Administration=READ_WRITE_ACCESS \
--resource-with-access=Access=READ_ACCESS > permission-set.yaml
```

2. 输入以下命令，创建允许访问 Administration 和 Access 资源的角色。本例创建一个名为 "restricted" 的角色，并保存为 role.yaml 文件。

```
$ roxctl declarative-config create role \
--name="restricted" \
--description="Restricted role that only allows access to Administration and Access" \
--permission-set="restricted" \
--access-scope="Unrestricted" > role.yaml
```

3. 输入以下命令，从上一步中创建的两个 YAML 文件创建配置映射。这个示例创建 declarative-configurations 配置映射。

```
$ kubectl create configmap declarative-configurations \ 1
--from-file permission-set.yaml --from-file role.yaml \
-o yaml --namespace=stackrox > declarative-configs.yaml
```

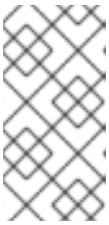
1. 对于 OpenShift Container Platform，使用 oc create。

4. 输入以下命令应用配置映射：

```
$ kubectl apply -f declarative-configs.yaml 1
```

1.

应用配置映射后，从 Central 中提取的配置信息会创建资源。



注意

虽然监视间隔为 5 秒，如以下段落中所述，但可将配置映射的更改传播到 Central 挂载时可能会有延迟。

您可以配置以下间隔来指定声明性配置如何与 Central 交互：

- 配置监视间隔： Central 检查更改的时间间隔每 5 秒进行一次。您可以使用 ROX_DECLARATIVE_CONFIG_WATCH_INTERVAL 环境变量配置此间隔。
- 协调间隔： 默认情况下，声明性配置与 Central 协调每 20 秒进行。您可以使用 ROX_DECLARATIVE_CONFIG_RECONCILE_INTERVAL 环境变量配置此间隔。

使用声明性配置创建身份验证和授权资源后，您可以在 RHACS web 门户的 Access Control 页面中查看它们。Origin 字段指示 资源 是使用声明性配置创建的 Declarative。

注意

您不能编辑从 RHACS web 门户中声明配置创建的资源。您必须直接编辑配置文件，以更改这些资源。

您可以通过进入到 Platform Configuration → System Health 并滚动到 Declarative 配置部分 来查看声明配置的状态。

16.3. 声明性配置示例

您可以使用以下示例创建声明配置作为指南。使用 roxctl declarative-config lint 命令来验证您的配置是否有效。

16.3.1. 声明性配置身份验证供应商示例

声明性配置身份验证供应商示例

```
name: A sample auth provider
minimumRole: Analyst (1)
uiEndpoint: central.custom-domain.com:443 (2)
extraUIEndpoints: (3)
  - central-alt.custom-domain.com:443
groups: (4)
  - key: email (5)
    value: example@example.com
    role: Admin (6)
  - key: groups
    value: reviewers
    role: Analyst
requiredAttributes: (7)
  - key: org_id
    value: "12345"
claimMappings: (8)
  - path: org_id
    value: my_org_id
oidc: (9)
  issuer: sample.issuer.com (10)
  mode: auto (11)
  clientID: CLIENT_ID
  clientSecret: CLIENT_SECRET
clientSecret: CLIENT_SECRET
iap: (12)
  audience: audience
saml: (13)
  spIssuer: sample.issuer.com
  metadataURL: sample.provider.com/metadata
saml: (14)
  spIssuer: sample.issuer.com
  cert: | (15)
  ssoURL: saml.provider.com
  idpIssuer: idp.issuer.com
userpki:
  certificateAuthorities: | (16)
  certificate (17)
openshift: (18)
  enable: true
```

(1) 标识默认分配给任何用户登录的最小角色。如果留空，则值为 None。

2

使用 Central 实例的用户界面端点。

3

如果您的 Central 实例公开给不同的端点，请在此处指定它们。

4

这些字段根据用户的属性将用户映射到特定的角色。

5

密钥可以是从身份验证提供程序返回的任何声明。

6

标识用户被授予的角色。您可以使用默认角色或声明性创建的角色。

7

可选：如果需要从身份验证供应商返回的属性，请使用这些字段；例如，如果使用者仅限于特定的机构或组。

8

可选：如果从身份提供程序返回的声明应映射到自定义声明，请使用这些字段。

9

本节只适用于 OpenID Connect (OIDC)身份验证供应商。

10

标识令牌的预期签发者。

11

标识 OIDC 回调模式。可能的值有 auto、post、query 和 fragment。首选的值为 auto。

12

本节只适用于 Google Identity-Aware Proxy (IAP)身份验证供应商。

13

14

本节只适用于 SAML 2.0 静态配置身份验证供应商。

15

以 Privacy Enhanced Mail (PEM)格式包括证书。

16

本节只适用于使用用户证书进行身份验证。

17

以 PEM 格式包含证书。

18

本节只适用于 OpenShift Auth 身份验证提供程序。

16.3.2. 声明性配置权限集示例

声明性配置权限集示例

```
name: A sample permission set
description: A sample permission set created declaratively
resources:
- resource: Integration 1
  access: READ_ACCESS 2
- resource: Administration
  access: READ_WRITE_ACCESS
```

1

如需支持的资源的完整列表，请转至 Access Control → Permission Sets。

2

访问可以是 READ_ACCESS 或 READ_WRITE_ACCESS。

16.3.3. 声明性配置访问范围示例

声明性配置访问范围示例

```
name: A sample access scope
description: A sample access scope created declaratively
rules:
  included:
    - cluster: secured-cluster-A (1)
      namespaces:
        - namespaceA
    - cluster: secured-cluster-B (2)
  clusterLabelSelectors:
    - requirements:
        key: kubernetes.io/metadata.name
        operator: IN (3)
        values:
        - production
        - staging
        - environment
```

(1) 标识仅在访问范围内包括特定命名空间的集群。

(2) 标识在访问范围内包括所有命名空间的集群。

(3) 标识要用于标签选择的 Operator。有效值为 IN,NOT_IN,EXISTS, 和 NOT_EXISTS。

16.3.4. 声明性配置角色示例

声明性配置角色示例

```
name: A sample role
description: A sample role created declaratively
permissionSet: A sample permission set (1)
accessScope: Unrestricted (2)
```

-

1

权限集的名称； 可以是系统权限集之一，也可以是声明性创建的权限集。

2

访问范围的名称；可以是系统访问范围之一，也可以是声明性创建的访问范围。

16.4. 声明性配置故障排除

您可以使用 Platform Configuration → System Health 页面的 Declarative configuration 部分显示的错误消息来帮助进行故障排除。roxctl declarative-config 命令还包括一个 lint 选项来验证配置文件，并帮助您检测错误。

平台 Configuration → System Health 页面的 Declarative configuration 部分显示的错误消息提供有关声明性配置的问题的信息。声明性配置的问题可能是由以下条件造成的：

- 配置文件的格式不是有效的 YAML。
- 配置文件包含无效值，如权限集中的无效访问。
- 存在无效的存储约束，如资源名称不是唯一的，或者配置包含对资源的无效引用。

要验证配置文件，请检查配置文件中的错误，并确保创建和更新配置文件时没有无效的存储限制，请使用 roxctl declarative-config lint 命令。

要在删除过程中对存储约束进行故障排除，请检查资源是否已标记为 Declarative Orphaned。这表示资源引用的声明性配置已被删除（例如，如果删除了由角色引用的权限集的声明配置）。要更正此错误，请编辑资源以指向新的权限集，或恢复已删除的声明性配置。

16.5. 其他资源

- 使用自定义使用 Helm chart 安装 Central (Red Hat OpenShift)
- 使用带有自定义（其他 Kubernetes 平台）的 Helm chart 安装 Central

第 17 章 将用户邀请到 RHACS 实例

通过邀请用户访问 Red Hat Advanced Cluster Security for Kubernetes (RHACS)，您可以确保正确的用户在集群中拥有适当的访问权限。您可以通过分配角色并定义身份验证提供程序来邀请一个或多个用户。

17.1. 配置访问控制和发送邀请

通过在 RHACS 门户中配置访问控制，您可以邀请用户访问 RHACS 实例。

流程

1. 在 RHACS 门户中，进入 **Platform Configuration** → **Access Control** → **Auth provider** 选项卡，然后点 **Invite users**。

2. 在 **Invite users** 对话框中提供以下信息：

   - **邀请电子邮件** ：输入您要邀请的一个或多个用户的电子邮件地址。确保它们是与预期接收者关联的有效电子邮件地址。

   - **Provider:** 从下拉列表中选择您要用于每个邀请用户的供应商。

重要

- 如果您只有一个身份验证供应商可用，则默认选择它。
- 如果有多个身份验证提供程序可用，并且至少有一个身份验证提供程序是 Red Hat SSO 或 Default Internal SSO，则默认选择该提供程序。
- 如果有多个身份验证提供程序可用，但它们都不是 Red Hat SSO 或 Default Internal SSO，则会提示您输入手动选择一个。
- 如果您还没有设置身份验证提供程序，则会出现警告消息，并禁用表单。点链接，它带您进入 Access Control 部分来配置身份验证供应商。

- 角色 ：从下拉列表中选择要分配给每个邀请用户的角色。

3. 点 Invite users。
4. 在确认对话框中，您会收到一条确认用户已使用所选角色创建的确认。
5. 将一个或多个电子邮件地址以及邮件复制到您在您自己的电子邮件客户端中创建的电子邮件中，并将其发送给用户。
6. 点 Done。

验证

1. 在 RHACS 门户中，进入 Platform Configuration → Access Control → Auth provider 选项卡。
2. 选择用于邀请用户的身份验证提供程序。

3. 向下滚动到 **Rules** 部分。

4. 验证用户电子邮件和验证供应商角色是否已添加到列表中。