Red Hat Advanced Cluster Security for Kubernetes 3.70

安装

安装 Red Hat Advanced Cluster Security for Kubernetes

Last Updated: 2023-02-27

Red Hat Advanced Cluster Security for Kubernetes 3.70 安装

安装 Red Hat Advanced Cluster Security for Kubernetes

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

法律通告

摘要

本文档论述了如何使用 Operator、Helm chart 或 roxctl CLI 安装 Red Hat Advanced Cluster Security for Kubernetes。

目录

第 1 章 安装 RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES 的先决条件

1.1. 常规要求

要安装 Red Hat Advanced Cluster Security for Kubernetes，您必须有：

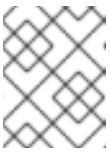
- OpenShift Container Platform 版本 4.5 或更高版本用于 OpenShift Container Platform 安装。

警告

您不能在以下环境中安装 Red Hat Advanced Cluster Security for Kubernetes：

- Amazon Elastic File System(Amazon EFS)。使用带有默认 **gp2** 卷类型的 Amazon Elastic Block Store(Amazon EBS)。
- 没有 SIMD 扩展 (SSE) 4.2 指令集的旧 CPU。例如，比 *Sandy Bridge* 和 AMD 处理器旧的 Intel 处理器（比 *Bulldozer* 旧）。（这些处理器在 2011 年发布。）

- 具有受支持的操作系统的集群节点。如需更多信息，请参阅 Red Hat Advanced Cluster Security for Kubernetes 支持政策。
  - **操作系统** ：Amazon Linux、CentOS、Container-Optimized OS from Google、Red Hat Enterprise Linux CoreOS(RHCOS)、Debian、Red Hat Enterprise Linux(RHEL) 或 Ubuntu。
  - **处理器和内存** ：2 个 CPU 内核和至少 3GiB RAM。



注意

对于部署中心，请使用带有 4 个或更多内核的机器类型，并应用调度策略在这样的节点上启动中心。

- 使用持久性卷声明(PVC)的持久性存储。

重要

您不能在 Red Hat Advanced Cluster Security for Kubernetes 中使用 Ceph FS 存储。红帽建议为 Red Hat Advanced Cluster Security for Kubernetes 使用 RBD 块模式 PVC。

  - 使用固态驱动器(SSD)以获得最佳性能。但是，如果您没有 SSD，也可以使用另一个存储类型。

- 如果您要使用 Helm chart 安装和配置 Red Hat Advanced Cluster Security for Kubernetes，则 Helm 命令行界面(CLI)v3.2 或更新版本。使用 **helm version** 命令验证已安装的 Helm 版本。

- OpenShift Container Platform CLI (**oc**)。
- 您必须具有在中心集群中配置部署所需的权限。
- 您必须有权访问 Red Hat Container Registry。有关从 **registry.redhat.io** 下载镜像的详情，请参考 Red Hat Container Registry 身份验证 。

1.2. 安装 CENTRAL 的先决条件

一个名为 Central 的单一容器化服务处理数据持久性、API 交互和用户界面(Portal)访问。

中心需要持久性存储：

- 您可以使用持久性卷声明(PVC)提供存储。

注意

只有在所有主机（或一组主机）挂载共享文件系统（如 NFS 共享或存储设备）时，您可以使用 hostPath 卷进行存储。否则，您的数据只保存在一个节点中。红帽不推荐使用 hostPath 卷。

- 使用固态磁盘(SSD)以获得最佳性能。但是，如果您没有 SSD，也可以使用另一个存储类型。
- 如果使用 web 代理或防火墙，您必须配置绕过规则，以允许 **definitions.stackrox.io** 和 **collector-modules.stackrox.io** 域的流量并启用 Red Hat Advanced Cluster Security for Kubernetes 来信任您的 web 代理或防火墙。否则，对漏洞定义和内核支持软件包更新将失败。Red Hat Advanced Cluster Security for Kubernetes 需要访问：
  - **definitions.stackrox.io**，用于下载更新的漏洞定义。漏洞定义更新允许 Red Hat Advanced Cluster Security for Kubernetes 在发现新漏洞或其他数据源时维护最新的漏洞数据。
  - **collector-modules.stackrox.io**，用于下载更新的内核支持软件包。更新了内核支持软件包，确保 Red Hat Advanced Cluster Security for Kubernetes 可以监控最新的操作系统，并收集与容器内运行的网络流量和进程相关的数据。如果没有这些更新，当在集群中添加新节点，或者更新节点的操作系统后，Red Hat Advanced Cluster Security for Kubernetes 可能无法监控容器。

注意

为安全起见，您应该在具有有限的管理访问权限的集群中部署 Central。

内存和存储要求

下表列出了安装和运行 Central 所需的最小内存和存储值。

Central	CPU	内存	存储
Request（请求）	1.5 个内核	4 GiB	100 GiB
限制	4 个核	8 GiB	100 GiB

大小指南

根据集群中的节点数量，使用以下计算资源和存储值。

节点	Deployments	CPU	内存	存储
最多 100	最多 1000	2 个内核	4 GiB	100 GiB
最多 500	最多 2000	4 个核	8 GiB	100 GiB
最多 500	最多 2000	8 个内核	12 - 16 GiB	100 - 200 GiB

1.3. 安装扫描器的先决条件

Red Hat Advanced Cluster Security for Kubernetes 包括一个称为 Scanner 的镜像漏洞策略。此服务扫描未被扫描程序集成到镜像 registry 中的镜像。

内存和存储要求

扫描程序	CPU	内存
Request（请求）	1.2 个内核	2700 MiB
限制	5 个内核	8000 MiB

1.4. 安装 SENSOR 的先决条件

Sensor 监控 Kubernetes 和 OpenShift Container Platform 集群。这些服务目前部署到单个部署中，该服务处理与 Kubernetes API 的交互，并与 Collector 协调。

内存和存储要求

Sensor	CPU	内存
Request（请求）	1 个内核	1 GiB
限制	2 个内核	4 GiB

1.5. 安装 ADMISSION CONTROLLER 的先决条件

Admission 控制器可防止用户创建违反您配置策略的工作负载。

内存和存储要求
默认情况下，准入控制服务运行 3 个副本。下表列出了每个副本的请求和限制。

准入控制器	CPU	内存
Request（请求）	.05 个内核	100 MiB
限制	.5 个内核	500 MiB

1.6. 安装 COLLECTOR 的先决条件

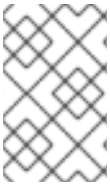
收集器监控安全集群中每个节点的运行时活动。它连接到 Sensor 来报告此信息。

小心

要在具有统一可扩展固件接口(UEFI)以及启用了安全引导机制的系统中安装 Collector，您必须使用 eBPF 探测，因为内核模块没有被签名，且 UEFI 固件无法加载未签名的软件包。收集器在启动时用来识别安全引导状态，并切换到 eBPF 探测（如果需要）。

内存和存储要求

Collector	CPU	内存
Request（请求）	.05 个内核	320 MiB
限制	.75 个内核	1 GiB



注意

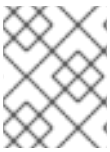
收集器使用 mutable 镜像标签(**<version>-latest**)，因此您可以更轻松地获得对较新的 Linux 内核版本的支持。对于镜像更新，代码、预先存在的内核模块或 eBPF 程序没有改变。更新只添加对初始发布后发布的新内核版本的支持的单个镜像层。

第 2 章 安装平台和方法

Red Hat Advanced Cluster Security for Kubernetes 在各种平台上被支持。本主题提供每个平台的信息以及安装文档的链接。

2.1. 不同平台的安装方法

您可以在不同的平台上执行不同类型的安装。



注意

不是所有安装选项都支持所有平台，如下表所示。红帽建议不要使用 **roxctl** 安装方法，除非您有需要使用此方法的特定安装需要。

表 2.1. 自我管理的平台

平台	支持的安装方法
Red Hat OpenShift Container Platform (OCP) 4.x	• Operator (推荐) • Helm • **roxctl**
Red Hat OpenShift Container Platform (OCP) 3.11.z	• Helm （推荐） • **roxctl**
Red Hat OpenShift Kubernetes Engine (OKE) 4.x	• Operator (推荐) • Helm • **roxctl**

表 2.2. 受管服务平台

平台	支持的安装方法
Red Hat OpenShift Dedicated (OSD)	• Operator (推荐) • Helm • **roxctl**

平台	支持的安装方法
Azure Red Hat OpenShift (ARO)	• Operator (推荐) • Helm • **roxctl**
Red Hat OpenShift Service on AWS (ROSA)	• Operator (推荐) • Helm • **roxctl**
Amazon Elastic Kubernetes Service (Amazon EKS)	• Helm （推荐） • **roxctl**
Google Kubernetes Engine (Google GKE)	• Helm （推荐） • **roxctl**
Microsoft Azure Kubernetes Service (Microsoft AKS)	• Helm （推荐） • **roxctl**

第 3 章 使用 OPERATOR 安装

Red Hat Advanced Cluster Security for Kubernetes(RHACS)在 OpenShift Container Platform 或 Kubernetes 集群中安装一组服务。本节介绍了使用 Operator 在 OpenShift Container Platform 或 Kubernetes 集群中安装 Red Hat Advanced Cluster Security for Kubernetes 的安装过程。

安装前：

- 了解 Red Hat Advanced Cluster Security for Kubernetes 架构 。
- 查看安装 Red Hat Advanced Cluster Security for Kubernetes 的先决条件 。

Red Hat Advanced Cluster Security for Kubernetes Operator 包括以下两个自定义资源：

1. **Central** - 中央资源是以下服务的逻辑分组：
   - **Central**：Central 是 Red Hat Advanced Cluster Security for Kubernetes 应用程序管理界面和服务。它处理数据持久性、API 互动和用户界面(RHACS Portal)访问。您可以使用同一中实例来保护多个 OpenShift Container Platform 或 Kubernetes 集群。
   - **Scanner** ：扫描程序是红帽开发的、经过认证的漏洞扫描程序，用于扫描容器镜像及其关联的数据库。它分析所有镜像层，以检查来自常见漏洞和暴露(CVE)列表中的已知漏洞。扫描程序还会识别由软件包管理器和多种编程语言相依性安装的软件包中的漏洞。
2. **SecuredCluster** - 安全集群资源是以下服务的逻辑分组：
   - **Sensor** ：传感器是负责分析和监控集群的服务。它处理与 OpenShift Container Platform 或 Kubernetes API 服务器交互以进行策略检测和实施，并与 Collector 协调。
   - **Collector** ：收集器分析和监控集群节点上的容器活动。它收集容器运行时和网络活动的信息。然后，它将收集的数据发送到 Sensor。
   - **Admission Control** ：准入控制器可防止用户创建违反 Red Hat Advanced Cluster Security for Kubernetes 中的安全策略的工作负载。

以下步骤代表了使用 Operator 安装 Red Hat Advanced Cluster Security for Kubernetes 的高级别工作流：

1. 在您要安装中心的集群中，从 OperatorHub 安装 Red Hat Advanced Cluster Security for Kubernetes Operator。
2. 配置和部署 **Central** 自定义资源。
3. 生成并应用 init 捆绑包。init 捆绑包包含在 Central 和安全集群间提供链接的 secret。
4. 在您要监控的所有集群中安装 Red Hat Advanced Cluster Security for Kubernetes Operator。
5. 在您要监控的每个集群中 配置和部署 **SecuredCluster** 自定义资源。

3.1. 安装 RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES OPERATOR

使用 OpenShift Container Platform 提供的 OperatorHub 是安装 Red Hat Advanced Cluster Security for Kubernetes 的最简单方法。

先决条件

- 您可以使用具有 Operator 安装权限的账户访问 OpenShift Container Platform 集群。
- 您必须使用 OpenShift Container Platform 4.6 或更高版本。

流程

1. 在 Web 控制台中进入 **Operators → OperatorHub** 页面。
2. 如果没有显示 Red Hat Advanced Cluster Security for Kubernetes，在 **Filter by keyword** 框中输入 **Advanced Cluster Security** 来查找 Red Hat Advanced Cluster Security for Kubernetes Operator。
3. 选择 **Red Hat Advanced Cluster Security for Kubernetes Operator** 查看详情页。
4. 阅读 Operator 信息并单击 **Install**。
5. 在 **Install Operator** 页面中：
   - 保留**安装模式**的默认值 **All namespaces on the cluster**。
   - 选择要在其中为 **Installed namespace** 字段安装 Operator 的特定命名空间。红帽建议在 **rhacs-operator** 命名空间中安装 Red Hat Advanced Cluster Security for Kubernetes Operator。
   - 为**更新批准**选择自动或手工。
     如果选择自动更新，当有新版 Operator 可用时，Operator Lifecycle Manager(OLM)会自动升级 Operator 的运行实例。

     如果选择手动更新，则当有新版 Operator 可用时，OLM 会创建更新请求。作为集群管理员，您必须手动批准该更新请求，才能将 Operator 更新至新版本。

     重要

     如果选择手动更新，则当安装了 Central 的集群中更新 RHACS Operator 时，您应该在所有安全集群中更新 RHACS Operator。安装 Central 的安全集群和集群应该具有相同的版本，以确保最佳功能。
6. 点 **Install**。

验证

- 安装完成后，进入到 **Operators → Installed Operators**，以验证 Red Hat Advanced Cluster Security for Kubernetes Operator 的状态为 **Succeeded**。

下一步

- 安装、配置和部署 **Central** 自定义资源。

3.2. 安装 CENTRAL

Red Hat Advanced Cluster Security for Kubernetes 的主要组件名为 Central。您可以使用 Central 自定义资源在 OpenShift Container Platform 上安装 **Central**。您只需要部署 Central 一次，并使用同一 Central 安装监控多个独立集群。

重要

当您首次安装 Red Hat Advanced Cluster Security for Kubernetes 时，您必须首先安装 **Central** 自定义资源，因为 **SecuredCluster** 自定义资源安装取决于 Central 生成的证书。

先决条件

- 您必须使用 OpenShift Container Platform 4.6 或更高版本。

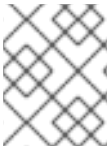
流程

1. 在 OpenShift Container Platform web 控制台中进入到 **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. 如果您在推荐的命名空间中安装了 Operator，OpenShift Container Platform 会将项目列为 **rhacs-operator**。选择 **Project: rhacs-operator → Create project**。

警告

- 如果您在不同的命名空间中安装了 Operator，则 OpenShift Container Platform 会显示该命名空间的名称，而不是 **rhacs-operator**。
- 您必须在自己的项目中安装 Red Hat Advanced Cluster Security for Kubernetes **Central** 自定义资源，而不是在 **rhacs-operator** 和 **openshift-operator** 项目中安装 Red Hat Advanced Cluster Security for Kubernetes Operator。

4. 输入新项目名称（如 **stackrox**），然后点 **Create**。红帽建议使用 **stackrox** 作为项目名称。
5. 在 **Provided APIs** 部分下，选择 **Central**。点 **Create Central**。
6. 输入您的 **Central** 自定义资源的名称并添加您要应用的任何标签。否则，接受可用选项的默认值。
7. 点 **Create**。



注意

如果使用集群范围的代理，Red Hat Advanced Cluster Security for Kubernetes 会使用该代理配置连接到外部服务。

后续步骤

1. 验证中央安装。
2. 可选：配置中央选项。
3. 生成 init 捆绑包。

其他资源

- 配置集群范围代理

3.3. 验证中央安装

安装中心后，登录到 RHACS 门户以验证中央安装是否成功。

流程

1. 在 OpenShift Container Platform web 控制台中进入到 **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. 选择 **Central** 选项卡。
4. 从 **Centrals** 列表中，选择 **stackrox-central-services** 以查看其详细信息。
5. 要获取 **admin** 用户的密码，您可以：
   - 点 **Admin Password Secret Reference** 下的链接。
   - 使用 OpenShift Container Platform CLI 进入 **Admin Credentials Info** 下列出的命令：

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

6. 使用 OpenShift Container Platform CLI 命令查找到 RHACS 门户的链接：

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

   另外，您可以执行以下命令，使用 Red Hat Advanced Cluster Security for Kubernetes web 控制台查找到 RHACS 门户的链接：

   a. 进入 **Networking → Routes**。

   b. 找到 **central** 路由，再点 **Location** 列下的 RHACS 门户链接。

7. 使用用户名 **admin** 和密码您在上一步中检索的密码登录 RHACS 门户。在 Red Hat Advanced Cluster Security for Kubernetes 被完全配置（例如，您具有 **Central** 资源，且至少有一个 **SecuredCluster** 资源已安装）前，仪表板中没有可用的数据。**SecuredCluster** 资源可以在与 **Central** 资源相同的集群中安装和配置。带有 **SecuredCluster** 资源的集群与 Red Hat Advanced Cluster Management(RHACM)中的受管集群类似。

后续步骤

1. 可选：配置中央设置。
2. 生成包含集群 secret 的 init 捆绑包，它允许在 **Central** 和 **SecuredCluster** 资源之间的通信。您需要下载这个捆绑包，使用它来在您要保护的集群中生成资源，并安全地存储它。

3.4. CENTRAL 配置选项

当您创建 Central 实例时，Operator 列出了 **Central** 自定义资源的以下配置选项。

3.4.1. Central 设置

参数	描述
central.adminPasswordSecret	指定在 **password** 密码数据项中包含管理员密码的 secret。如果省略，Operator 会自动生成密码，并将其存储在 **central-htpasswd** secret 的 **password** 项中。
central.defaultTLSSecret	默认情况下，Central 仅提供内部 TLS 证书，这意味着您需要在入口或负载均衡器级别处理 TLS 终止。如果要在 Central 中终止 TLS 并提供自定义服务器证书，您可以指定包含证书和私钥的 secret。
central.adminPasswordGenerationDisabled	将此参数设置为 **true** 以禁用自动管理员密码生成。仅在执行替代验证方法首次设置后使用它。不要将它用于初始安装。否则，您必须重新安装自定义资源才能重新登录。
central.tolerations	如果节点选择器选择污点节点，请使用此参数指定 taint toleration key、value 和 effect。此参数主要用于基础架构节点。
central.exposure.loadBalancer.enabled	把它设置为 **true**，以通过负载均衡器公开 Central。
central.exposure.loadBalancer.port	使用此参数为您的负载均衡器指定自定义端口。
central.exposure.loadBalancer.ip	使用这个参数为您的负载均衡器指定保留的静态 IP 地址。
central.exposure.route.enabled	把它设置为 **true**，以通过 OpenShift 路由公开 Central。默认值为 **false**。
central.exposure.nodeport.enabled	把它设置为 **true**，以通过节点端口公开 Central。默认值为 **false**。
central.exposure.nodeport.port	使用此选项指定显式节点端口。
central.nodeSelector	如果您希望此组件只在特定节点上运行，您可以使用此参数配置节点选择器。
central.persistence.hostPath.path	指定将持久数据存储在主机上的主机路径。红帽不推荐使用这个方法。如果需要使用主机路径，则必须将其与节点选择器一起使用。
central.persistence.persistentVolumeClaim.claimName	要管理的持久性数据的 PVC 名称。如果没有具有指定名称的 PVC，则会创建它。如果没有设置，则默认值为 **stackrox-db**。为防止数据丢失 PVC，使用中心删除操作不会自动删除。
central.persistence.persistentVolumeClaim.size	通过声明创建持久性卷的大小。默认情况下会自动生成。

参数	描述
central.persistence.persistentVolumeClaim.storageClassName	用于 PVC 的存储类的名称。如果您的集群没有配置默认存储类，则必须为此参数提供一个值。
central.resources.limits	使用此参数覆盖 Central 的默认资源限值。
central.resources.requests	使用此参数覆盖 Central 的默认资源请求。
central.imagePullSecrets	使用此参数指定 Central 镜像的镜像 pull secret。

3.4.2. 扫描程序设置

参数	描述
scanner.analyzer.nodeSelector	如果您希望此扫描程序仅在特定节点上运行，您可以使用此参数配置节点选择器。
scanner.analyzer.tolerations	如果节点选择器选择污点节点，请使用此参数为 Scanner 指定污点容限键、值和效果。此参数主要用于基础架构节点。
scanner.analyzer.resources.limits	使用此参数覆盖扫描程序的默认资源限值。
scanner.analyzer.resources.requests	使用此参数覆盖扫描程序的默认资源请求。
scanner.analyzer.scaling.autoScaling	启用后，分析器副本数量会根据指定的限值来动态管理。
scanner.analyzer.scaling.maxReplicas	指定使用分析器自动扩展配置的最大副本
scanner.analyzer.scaling.minReplicas	指定使用分析器自动扩展配置的最小副本
scanner.analyzer.scaling.replicas	禁用自动扩展时，副本数始终配置为与此值匹配。
scanner.db.nodeSelector	如果您希望此组件只在特定节点上运行，您可以使用此参数配置节点选择器。

参数	描述
scanner.db.tolerations	如果节点选择器选择污点节点，请使用此参数为 Scanner DB 指定污点容限键、值和效果。此参数主要用于基础架构节点。
scanner.db.resources.limits	使用此参数覆盖扫描程序的默认资源限值。
scanner.db.resources.requests	使用此参数覆盖扫描程序的默认资源请求。
scanner.scannerComponent	如果您不想部署 Scanner，可以使用此参数来禁用它。如果禁用扫描器，本节中的所有其他设置都无效。红帽不推荐为 Kubernetes 扫描器禁用 Red Hat Advanced Cluster Security。

3.4.3. 常规设置和各种设置

参数	描述
tls.additionalCAs	要信任的安全集群的其他可信 CA 证书。这通常在与使用私有证书颁发机构的服务集成时使用。
misc.createSCCs	指定 **true** 为 Central 创建 **SecurityContextConstraints** (SCC)。它可能会在某些环境中出现问题。

3.5. 生成 INIT 捆绑包

在集群中安装 **SecuredCluster** 资源前，您必须创建一个 init 捆绑包。安装并配置 **SecuredCluster** 的集群，然后使用此捆绑包与 Central 进行身份验证。

您可以使用 RHACS 门户（推荐）或使用 roxctl CLI 创建 init 捆绑包。

3.5.1. 使用 RHACS 门户生成 init 捆绑包

您可以使用 RHACS 门户创建包含 secret 的 init 捆绑包。

流程

1. 根据您的暴露的方法查找 RHACS 门户地址：
   a. 对于路由：

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

   b. 对于负载均衡器：

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

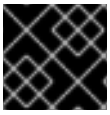
c. 对于端口转发：

i. 运行以下命令：

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

ii. 进入到 **https://localhost:18443/**。

2. 在 RHACS 门户网站中，进入 **Platform Configuration → Integrations**。
3. 进入 **Authentication Tokens** 部分，再点 **Cluster Init Bundle**。
4. 点 **Generate bundle**。
5. 为集群 init 捆绑包输入一个名称并点 **Generate**。
6. 点 **Download Kubernetes Secret File** 下载生成的捆绑包。



重要

安全地存储此捆绑包，因为它包含 secret。您可以使用同一捆绑包来创建多个安全集群。

下一步

1. 使用 OpenShift Container Platform CLI 使用 init 捆绑包创建资源。
2. 在您要监控的所有集群中安装 Red Hat Advanced Cluster Security for Kubernetes。

3.5.2. 使用 roxctl CLI 生成 init 捆绑包

您可以使用 **roxctl** CLI 创建带有 secret 的 init 捆绑包。

先决条件

您已配置了 **ROX_API_TOKEN** 和 **ROX_CENTRAL_ADDRESS** 环境变量。

- 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>
```

流程

- 运行以下命令以生成包含 secret 的集群 init 捆绑包：

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

重要

请确定您安全地存储此捆绑包，因为它包含 secret。您可以使用同一捆绑包来设置多个安全集群。

3.5.3. 其他资源

- 安装 **roxctl** CLI
- 使用 **roxctl** CLI

3.6. 使用 INIT 捆绑包创建资源

在安装安全集群前，您必须使用 init 捆绑包在集群中创建所需的资源，以允许安全集群上的服务与 Central 通信。

先决条件

- 您必须生成了一个包含 secret 的 init 捆绑包。

流程

- 使用 OpenShift Container Platform CLI 运行以下命令来创建资源：

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

1. 指定包含 secret 的 init 捆绑包的文件名。
2. 指定安装中心的项目的名称。

下一步

- 在您要监控的所有集群中安装 Red Hat Advanced Cluster Security for Kubernetes。

3.7. 安装安全的集群服务

您可以使用 **SecuredCluster** 自定义资源在集群中安装安全的集群服务。您必须在要监控的环境中的每个集群中安装安全集群服务。

小心

要在具有统一可扩展固件接口(UEFI)以及启用了安全引导机制的系统中安装 Collector，您必须使用 eBPF 探测，因为内核模块没有被签名，且 UEFI 固件无法加载未签名的软件包。收集器在启动时用来识别安全引导状态，并切换到 eBPF 探测（如果需要）。

先决条件

- 您必须使用 OpenShift Container Platform 4.6 或更高版本。
- 您必须生成一个 init 捆绑包，且已经使用 init 捆绑包创建所需的资源。

流程

1. 在 OpenShift Container Platform web 控制台中进入到 **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. 默认情况下，OpenShift Container Platform 将项目列为 **rhacs-operator**。选择 **Project: rhacs-operator → Create project**。

警告

您必须在自己的项目中安装 Red Hat Advanced Cluster Security for Kubernetes **SecuredCluster** 资源，而不是默认的 **openshift-operators** 项目。

4. 将新项目名称输入为 **stackrox** 或一些其他名称，然后点 **Create**。
5. 在 **Provided APIs** 部分中，选择 **Secured Cluster**。
6. 选择 **Create SecuredCluster**。
7. 输入您的 **SecuredCluster** 自定义资源名称。
8. 对于 **Central 端点**，请输入您的 Central 实例的地址和端口号。例如，如果 Central 位于 **https://central.example.com**，则将中央端点指定为 **central.example.com:443**。**central.stackrox.svc:443** 的默认值只有您在同一集群中安装了安全集群服务和 Central 时才可以正常工作。
9. 根据需要，选择使用默认值或为相关的选项设置自定义值。
10. 点 **Create**。

后续步骤

1. 可选：配置其他安全的集群设置。
2. 验证 Red Hat Advanced Cluster Security for Kubernetes 安装。

3.8. 安全的集群配置选项

当您创建 Central 实例时，Operator 列出了 **Central** 自定义资源的以下配置选项。

3.8.1. 所需的配置设置

参数	描述
centralEndpoint	用于连接的 Central 实例的端点，包括端口号。如果使用一个支持非 gRPC 的负载均衡器，请使用带有 **ws://** 的端点地址的 WebSocket 协议。如果您没有为此参数指定值，则 Sensor 会尝试连接到在同一命名空间中运行的 Central 实例。
clusterName	此集群的唯一名称，显示在 RHACS 门户中。使用此参数设置名称后，您将无法再次更改它。要更改名称，您必须删除并重新创建对象。

3.8.2. 准入控制器设置

参数	描述
admissionControl.listenOnCreates	指定 **true** 以启用创建对象的防止策略强制。默认值为 **false**。
admissionControl.listenOnEvents	指定 **true** 来为 Kubernetes 事件启用监控和强制实施，如**port-forward** 和 **exec** 事件。它用于通过 Kubernetes API 控制资源访问。默认值为 **true**。
admissionControl.listenOnUpdates	指定 **true** 来为对象更新启用防止策略强制。除非将 **Listen On Creates** 设为 **true**， 否则它不会生效。默认值为 **false**。
admissionControl.nodeSelector	如果您希望此组件只在特定节点上运行，您可以使用此参数配置节点选择器。
admissionControl.tolerations	如果节点选择器选择污点节点，请使用此参数指定污点容限键、值以及 Admission Control 的效果。此参数主要用于基础架构节点。
admissionControl.resources.limits	使用此参数覆盖准入控制器的默认资源限值。
admissionControl.resources.requests	使用此参数覆盖准入控制器的默认资源请求。
admissionControl.bypass	使用以下值之一配置绕过准入控制器强制： • **BreakGlassAnnotation** 允许通过 **admission.stackrox.io/break-glass** 注解绕过准入控制器。 • **Disabled** 禁用安全集群绕过准入控制器强制实施的功能。 默认值为 **BreakGlassAnnotation**。
admissionControl.contactImageScanners	使用以下值之一指定准入控制器是否必须连接到镜像扫描程序： • 如果缺少镜像的扫描结果，**ScanIfMissing**。 • **DoNotScanInline** 用来在处理准入请求时跳过扫描镜像。 默认值为 **DoNotScanInline**。
admissionControl.timeoutSeconds	在将 Red Hat Advanced Cluster Security for Kubernetes 标记为失败前，使用这个参数指定 Red Hat Advanced Cluster Security for Kubernetes 的最大秒数。

3.8.3. 扫描程序配置

使用 Scanner 配置设置修改 OpenShift Container Registry(OCR)的本地集群扫描程序。

参数	描述
scanner.analyzer.nodeSelector	将节点选择器标签指定为 **label-key: label-value**，以强制 Scanner 仅调度到具有指定标签的节点。
scanner.analyzer.resources.requests.memory	Scanner 容器的内存请求。使用此参数覆盖默认值。
scanner.analyzer.resources.requests.cpu	Scanner 容器的 CPU 请求。使用此参数覆盖默认值。
scanner.analyzer.resources.limits.memory	Scanner 容器的内存限值。使用此参数覆盖默认值。
scanner.analyzer.resources.limits.cpu	Scanner 容器的 CPU 限制。使用此参数覆盖默认值。
scanner.scaling.autoscaling	如果将此选项设置为 **Disabled**，Red Hat Advanced Cluster Security for Kubernetes 会禁用 Scanner 部署的自动扩展。默认值为 **Enabled**。
scanner.scaling.minReplicas	自动扩展的最小副本数。默认值为 **2**。
scanner.scaling.maxReplicas	自动扩展的最大副本数。默认值为 **5**。
scanner.scaling.replicas	默认副本数。默认值为 **3**。
scanner.Tolerations	如果节点选择器选择污点节点，请使用此参数为 Scanner 指定污点容限键、值和效果。
scanner.db.nodeSelector	将节点选择器标签指定为 **label-key: label-value**，以强制 Scanner DB 仅调度到具有指定标签的节点。
scanner.db.resources.requests.memory	Scanner DB 容器的内存请求。使用此参数覆盖默认值。
scanner.db.resources.requests.cpu	Scanner DB 容器的 CPU 请求。使用此参数覆盖默认值。
scanner.db.resources.limits.memory	Scanner DB 容器的内存限值。使用此参数覆盖默认值。
scanner.db.resources.limits.cpu	Scanner DB 容器的 CPU 限制。使用此参数覆盖默认值。

参数	描述
scanner.db.tolerations	如果节点选择器选择污点节点，请使用此参数为 Scanner DB 指定污点容限键、值和效果。
scanner.scannerComponent	如果将此选项设置为 **Disabled**，Red Hat Advanced Cluster Security for Kubernetes 不会部署 Scanner 部署。不要在 OpenShift Container Platform 集群上禁用 Scanner。默认值为 **AutoSense**。

3.8.4. 镜像配置

在使用自定义 registry 时使用镜像配置设置。

参数	描述
imagePullSecrets.name	拉取镜像时考虑的其他镜像 pull secret。

3.8.5. 针对每个节点的设置

针对每个节点的设置为在集群中的节点上运行的组件定义了一组配置设置，用于保护集群的安全。这些组件是 Collector 和 Compliance。

参数	描述
perNode.collector.collection	系统级数据收集的方法。默认值为 **KernelModule**。红帽建议您使用 **KernelModule** 作为这个参数的值。如果您选择 **NoCollection**，则无法查看有关网络活动和进程执行的任何信息。选项包括 **NoCollection**、**EBPF** 和 **KernelModule**。
perNode.collector.imageFlavor	用于 Collector 的镜像类型。您可以将它指定为 **Regular** 或 **Slim**。**Regular** 镜像大小较大，但包含大多数内核的内核模块。如果使用 **Slim** 镜像类型，您必须确保您的 Central 实例连接到互联网，或定期接收 Collector 支持软件包更新。默认值为 **Slim**。
perNode.collector.resources.limits	使用此参数覆盖 Collector 的默认资源限值。
perNode.collector.resources.requests	使用此参数覆盖 Collector 的默认资源请求。
perNode.compliance.resources.requests	使用此参数覆盖 Compliance 的默认资源请求。
perNode.compliance.resources.limits	使用此参数覆盖 Compliance 的默认资源限值。

3.8.6. 污点容限设置

参数	描述
taintToleration	为确保对集群进行全面监控，Red Hat Advanced Cluster Security for Kubernetes 在每个节点上运行服务，包括污点节点。如果您不希望此行为，将此参数设置为 **AvoidTaints**。

3.8.7. Sensor 配置

此配置定义了 Sensor 组件的设置，该组件的设置在集群的一个节点上运行。

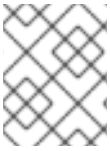
参数	描述
sensor.nodeSelector	如果您希望 Sensor 仅在特定节点上运行，您可以配置节点选择器。
sensor.tolerations	如果节点选择器选择污点节点，请使用此参数指定污点容限键、值和 Sensor 的效果。此参数主要用于基础架构节点。
sensor.resources.limits	使用这个参数覆盖 Sensor 的默认资源限值。
sensor.resources.requests	使用这个参数覆盖 Sensor 的默认资源请求。

3.8.8. 常规设置和各种设置

参数	描述
tls.additionalCAs	安全集群的其他可信 CA 证书。这些证书在使用私有证书颁发机构与服务集成时使用。
misc.createSCCs	把它设置为 **true**，以便为 Central 创建 SCC。它可能会在某些环境中出现问题。
customize.annotations	允许为 Central 部署指定自定义注解。
customize.envVars	用于配置环境变量的高级设置。
egress.connectivityPolicy	配置 Red Hat Advanced Cluster Security for Kubernetes 是否应该以在线或离线模式运行。在离线模式下，禁用对漏洞定义和内核模块的自动更新。

3.9. 验证安装

完成安装后，运行几个存在安全漏洞的应用程序并进入 RHACS 门户来评估安全评估结果和策略违反结果。



注意

以下部分中列出的示例应用程序包含关键漏洞，它们旨在验证 Red Hat Advanced Cluster Security for Kubernetes 的构建和部署时间评估功能。

验证安装：

1. 根据您的暴露的方法查找 RHACS 门户地址：
   a. 对于路由：

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

   b. 对于负载均衡器：

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

   c. 对于端口转发：
      i. 运行以下命令：

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

      ii. 进入到 **https://localhost:18443/**。
2. 使用 OpenShift Container Platform CLI 创建新项目：

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

3. 使用关键漏洞启动一些应用程序：

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

Red Hat Advanced Cluster Security for Kubernetes 会在向集群提交后自动扫描这些部署以了解安全风险以及策略违反情况。进入 RHACS 门户以查看违反情况。您可以使用默认用户名 **admin** 和生成的密码登录到 RHACS 门户。

3.10. 在 RHACS 中添加新集群

要在 Red Hat Advanced Cluster Security for Kubernetes 中添加更多集群，您必须在要添加的每个集群中安装 Red Hat Advanced Cluster Security for Kubernetes Operator。

以下步骤代表了向 Red Hat Advanced Cluster Security for Kubernetes 添加额外集群的高级流程：

1. 在集群中安装 Red Hat Advanced Cluster Security for Kubernetes Operator 。
2. 使用现有的 init 捆绑包或生成新的 init 捆绑包。
3. 使用 init 捆绑包在集群中创建资源。
4. 在集群上安装安全的集群服务。

第 4 章 使用 HELM CHART 安装

4.1. 使用 HELM CHART 快速安装

Red Hat Advanced Cluster Security for Kubernetes 在 OpenShift Container Platform 集群中安装一组服务。本节论述了在没有自定义的情况下在 OpenShift Container Platform 集群中安装 Red Hat Advanced Cluster Security for Kubernetes 的安装过程。

以下步骤代表了快速安装 Red Hat Advanced Cluster Security for Kubernetes 的高级别安装流程：

1. 添加 Red Hat Advanced Cluster Security for Kubernetes Helm Chart 仓库。
2. 安装 **central-services** Helm Chart 来安装 集中组件 （Central 和 Scanner）。
3. 生成 init 捆绑包。
4. 安装 **secured-cluster-services** Helm chart，以安装 per-cluster（针对每个集群）和per-node（针对每个节点）组件（Sensor、Admission Controller 和 Collector）。

安装前：

- 了解 Red Hat Advanced Cluster Security for Kubernetes 架构 。
- 查看安装 Red Hat Advanced Cluster Security for Kubernetes 的先决条件 。

4.1.1. 添加 Helm Chart 仓库

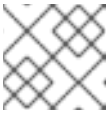
流程

- 添加 Red Hat Advanced Cluster Security for Kubernetes charts 软件仓库。

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

Red Hat Advanced Cluster Security for Kubernetes 的 Helm 仓库包括两个用于安装不同组件的 Helm chart。

  - 用于安装集中组件（Central 和 Scanner）的中央服务Helm Chart（**central-services**）。



注意

您只部署集中式组件一次，并可使用同一安装监控多个独立集群。

  - 安全集群服务 Helm Chart (**secured-cluster-services**) 用于安装针对每个集群（Sensor 和 Admission 控制器）和针对每个节点(Collector) 的组件。

注意

将 per-cluster 组件部署到要监控的每个集群中，并在要监控的所有节点中部署 per-node 组件。

验证

- 运行以下命令来验证添加的 chart 存储库：

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

4.1.2. 在不自定义的情况下安装中央服务 Helm Chart

使用以下说明安装 **central-services** Helm Chart 以部署集中组件（Central 和 Scanner）。

流程

- 运行以下命令安装 Central 服务并使用一个路由来公开 Central：

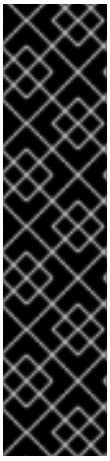
```
$ helm install -n stackrox \
  --create-namespace stackrox-central-services rhacs/central-services \
  --set imagePullSecrets.allowNone=true \
  --set central.exposure.route.enabled=true
```

- 或者，运行以下命令安装 Central 服务并使用一个负载均衡器来公开 Central：

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

- 或者，运行以下命令安装 Central 服务并使用一个端口转发来公开 Central：

```
$ helm install -n stackrox \
  --create-namespace stackrox-central-services rhacs/central-services \
  --set imagePullSecrets.allowNone=true
```



重要

如果要在需要使用代理连接到外部服务的集群中安装 Red Hat Advanced Cluster Security for Kubernetes，则必须使用 **proxyConfig** 参数指定代理配置。例如：

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

安装命令的输出包括：

- 自动生成的管理员密码。
- 关于存储所有配置值的说明。
- Helm 生成的任何警告。

4.1.3. 生成 init 捆绑包

在集群中安装 **SecuredCluster** 资源前，您必须创建一个 init 捆绑包。安装并配置 **SecuredCluster** 的集群，然后使用此捆绑包与 Central 进行身份验证。

4.1.3.1. 使用 roxctl CLI 生成 init 捆绑包

您可以使用 **roxctl** CLI 创建带有 secret 的 init 捆绑包。

先决条件

您已配置了 **ROX_API_TOKEN** 和 **ROX_CENTRAL_ADDRESS** 环境变量。

- 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>
```

流程

- 运行以下命令以生成包含 secret 的集群 init 捆绑包：

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

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

重要

请确定您安全地存储此捆绑包，因为它包含 secret。您可以使用同一捆绑包来设置多个安全集群。

其他资源

- 安装 **roxctl** CLI
- 使用 RHACS 门户生成 init 捆绑包

4.1.4. 在不使用自定义配置的情况下安装 secured-cluster-services Helm chart

使用以下说明安装 **secure-cluster-services** Helm chart，以部署 per-cluster 和 per-node 组件（Sensor、Admission Controller 和 Collector）。

小心

要在具有统一可扩展固件接口(UEFI)以及启用了安全引导机制的系统中安装 Collector，您必须使用 eBPF 探测，因为内核模块没有被签名，且 UEFI 固件无法加载未签名的软件包。收集器在启动时用来识别安全引导状态，并切换到 eBPF 探测（如果需要）。

先决条件

- 您必须有用于公开 Central 服务的地址和端口号。

流程

- 在其他基于 Kubernetes 的集群上运行以下命令：

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

1. 使用 **-f** 选项指定 init 捆绑包的路径。
2. 指定 Central 的地址和端口号。例如, **acs.domain.com:443**。

- 在 OpenShift Container Platform 集群中运行以下命令：

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

1. 使用 **-f** 选项指定 init 捆绑包的路径。
2. 指定 Central 的地址和端口号。例如, **acs.domain.com:443**。

4.1.5. 验证安装

完成安装后，运行几个存在安全漏洞的应用程序并进入 RHACS 门户来评估安全评估结果和策略违反结果。

注意

以下部分中列出的示例应用程序包含关键漏洞，它们旨在验证 Red Hat Advanced Cluster Security for Kubernetes 的构建和部署时间评估功能。

验证安装：

1. 根据您的暴露的方法查找 RHACS 门户地址：
   a. 对于路由：

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

   b. 对于负载均衡器：

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

c. 对于端口转发：

i. 运行以下命令：

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

ii. 进入到 **https://localhost:18443/**。

2. 使用 OpenShift Container Platform CLI 创建新项目：

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

3. 使用关键漏洞启动一些应用程序：

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

Red Hat Advanced Cluster Security for Kubernetes 会在向集群提交后自动扫描这些部署以了解安全风险以及策略违反情况。进入 RHACS 门户以查看违反情况。您可以使用默认用户名 **admin** 和生成的密码登录到 RHACS 门户。

4.1.6. 其他资源

- 通过 Helm chart 使用自定义安装

4.2. 通过 HELM CHART 使用自定义安装

安装流程：

1. 为 Kubernetes Helm Chart 仓库添加 Red Hat Advanced Cluster Security。
2. 配置 **central-services** Helm Chart。
3. 安装 **central-services** Helm Chart 来安装 集中组件 （Central 和 Scanner）。
4. 生成 init 捆绑包。
5. 配置 **secure-cluster-services** Helm Chart。
6. 安装 **secured-cluster-services** Helm chart，以安装 per-cluster（针对每个集群）和per-node（针对每个节点）组件（Sensor、Admission Controller 和 Collector）。

安装前：

- 了解 Red Hat Advanced Cluster Security for Kubernetes 架构 。
- 查看安装 Red Hat Advanced Cluster Security for Kubernetes 的先决条件 。

4.2.1. 添加 Helm Chart 仓库

流程

- 添加 Red Hat Advanced Cluster Security for Kubernetes charts 软件仓库。

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

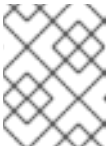
Red Hat Advanced Cluster Security for Kubernetes 的 Helm 仓库包括两个用于安装不同组件的 Helm chart。

  - 用于安装集中组件（Central 和 Scanner）的中央服务Helm Chart（**central-services**）。

注意

您只部署集中式组件一次，并可使用同一安装监控多个独立集群。

  - 安全集群服务 Helm Chart (**secured-cluster-services**) 用于安装针对每个集群（Sensor 和 Admission 控制器）和针对每个节点(Collector) 的组件。



注意

将 per-cluster 组件部署到要监控的每个集群中，并在要监控的所有节点中部署 per-node 组件。

验证

- 运行以下命令来验证添加的 chart 存储库：

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

4.2.2. 配置 central-services Helm chart

本节论述了可用于 **helm install** 和 **helm upgrade** 命令的 Helm Chart 配置参数。您可以使用 **--set** 选项或创建 YAML 配置文件来指定这些参数。

创建以下文件来配置 Helm chart 来安装 Red Hat Advanced Cluster Security for Kubernetes：

- 公共配置文件 **values-public.yaml** ：使用此文件保存所有非敏感配置选项。
- 专用配置文件 **values-private.yaml** ：使用此文件保存所有敏感配置选项。请确定您安全地存储这个文件。

4.2.2.1. 专用配置文件

本节列出了 **values-private.yaml** 文件的可配置参数。这些参数没有默认值。

4.2.2.1.1. 镜像 pull secret

从 registry 中拉取镜像所需的凭证取决于以下因素：

- 如果使用自定义 registry，您必须指定这些参数：
  - **imagePullSecrets.username**
  - **imagePullSecrets.password**
  - **image.registry**

- 如果不使用用户名和密码登录到自定义 registry，您必须指定以下参数之一：
  - **imagePullSecrets.allowNone**
  - **imagePullSecrets.useExisting**
  - **imagePullSecrets.useFromDefaultServiceAccount**

参数	描述
imagePullSecrets.username	用于登录到 registry 的帐户的用户名。
imagePullSecrets.password	用于登录到 registry 的帐户的密码。
imagePullSecrets.allowNone	如果您使用自定义 registry，且允许在没有凭证的情况下拉取镜像，请使用 **true**。
imagePullSecrets.useExisting	以逗号分隔的 secret 列表作为值。例如，**secret1, secret2, secretN**。如果您已在目标命名空间中创建了预先存在的镜像 pull secret，则使用此选项。
imagePullSecrets.useFromDefaultServiceAccount	如果您已经在目标命名空间中配置了具有足够范围的镜像 pull secret 的默认服务帐户，请使用 **true**。

4.2.2.1.2. 代理配置

如果要在需要使用代理连接到外部服务的集群中安装 Red Hat Advanced Cluster Security for Kubernetes，则必须使用 **proxyConfig** 参数指定代理配置。例如：

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

参数	描述
env.proxyConfig	您的代理配置。

4.2.2.1.3. Central

Central 的可配置参数。

对于新安装，您可以跳过以下参数：

- **central.jwtSigner.key**
- **central.serviceTLS.cert**

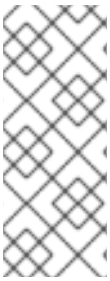
- **central.serviceTLS.key**
- **central.adminPassword.value**
- **central.adminPassword.htpasswd**
- 当您没有为这些参数指定值时，Helm Chart 会为它们自动生成值。
- 如果要修改这些值，您可以使用 **helm upgrade** 命令并使用 **--set** 选项指定值。

重要

对于设置管理员密码，您只能使用 **central.adminPassword.value** 或 **central.adminPassword.htpasswd**，但不能同时使用两者。

参数	描述
central.jwtSigner.key	Red Hat Advanced Cluster Security for Kubernetes 应使用的私钥签名 JSON Web 令牌(JWT)进行验证。
central.serviceTLS.cert	Central 服务应用于部署中心的内部证书。
central.serviceTLS.key	Central 服务应使用的内部证书的私钥。
central.defaultTLS.cert	Central 应该使用的用户面向用户的证书。Red Hat Advanced Cluster Security for Kubernetes 将这个证书用于 RHACS 门户。 • 对于新安装，您必须提供一个证书，否则 Red Hat Advanced Cluster Security for Kubernetes 通过使用自签名证书安装 Central。 • 如果要升级，Red Hat Advanced Cluster Security for Kubernetes 会使用现有证书及其密钥。
central.defaultTLS.key	Central 应使用面向用户的证书的私钥。 • 对于新安装，您必须提供私钥，否则 Red Hat Advanced Cluster Security for Kubernetes 使用自签名证书安装中心。 • 如果要升级，Red Hat Advanced Cluster Security for Kubernetes 会使用现有证书及其密钥。
central.adminPassword.value	用于登录到 Red Hat Advanced Cluster Security for Kubernetes 的管理员密码。
central.adminPassword.htpasswd	用于登录到 Red Hat Advanced Cluster Security for Kubernetes 的管理员密码。此密码以散列格式存储，使用 bcrypt。

参数	描述



注意

如果使用 **central.adminPassword.htpasswd** 参数，则必须使用 bcrypt 编码的密码哈希。您可以运行 **htpasswd -nB admin** 命令来生成密码哈希。例如，

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

4.2.2.1.4. 扫描程序

扫描程序的可配置参数。

对于新的安装，您可以跳过以下参数，以及 Helm Chart 自动生成值。否则，如果您升级到新版本，请指定以下参数的值：

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

参数	描述
scanner.dbPassword.value	用于通过 Scanner 数据库进行身份验证的密码。不要修改此参数，因为 Red Hat Advanced Cluster Security for Kubernetes 会自动创建和使用其值。
scanner.serviceTLS.cert	扫描程序服务用于部署扫描器的内部证书。
scanner.serviceTLS.key	Scanner 服务使用的内部证书的私钥。
scanner.dbServiceTLS.cert	Scanner-db 服务应用于部署 Scanner 数据库的内部证书。
scanner.dbServiceTLS.key	Scanner-db 服务应使用的内部证书的私钥。

4.2.2.2. 公共配置文件

本节列出了 **values-public.yaml** 文件的可配置参数。

4.2.2.2.1. 镜像 pull secret

镜像拉取 secret 是从 registry 中拉取镜像所需的凭证。

参数	描述
imagePullSecrets.allowNone	如果您使用自定义 registry，且允许在没有凭证的情况下拉取镜像，请使用 **true**。
imagePullSecrets.useExisting	以逗号分隔的 secret 列表作为值。例如，**secret1, secret2**.如果您已在目标命名空间中创建了预先存在的镜像 pull secret，则使用此选项。
imagePullSecrets.useFromDefaultServiceAccount	如果您已经在目标命名空间中配置了具有足够范围的镜像 pull secret 的默认服务帐户，请使用 **true**。

4.2.2.2.2. Image

镜像声明了配置来设置主 registry，Helm Chart 用来为 **central.image**、**scanner.image** 和 **scanner.dbImage** 参数解析镜像。

参数	描述
image.registry	镜像 registry 的地址。使用主机名，如 **registry.redhat.io** 或远程 registry 主机名，如 **us.gcr.io/stackrox-mirror**。

4.2.2.2.3. 环境变量

Red Hat Advanced Cluster Security for Kubernetes 会自动检测到集群环境，并为 **env.openshift**、**env.istio** 和 **env.platform** 设置值。仅设置这些值来覆盖自动集群环境检测。

参数	描述
env.openshift	使用 **true** 在 OpenShift Container Platform 集群上安装并覆盖自动集群环境检测。
env.istio	使用 **true** 在启用了 Istio 的集群上安装并覆盖自动集群环境检测。
env.platform	安装 Red Hat Advanced Cluster Security for Kubernetes 的平台。将其值设为 **default** 或 **gke** 以指定集群平台并覆盖自动集群环境检测。
env.offlineMode	使用 **true** 以离线模式使用 Red Hat Advanced Cluster Security for Kubernetes。

4.2.2.2.4. 其他可信证书颁发机构

Red Hat Advanced Cluster Security for Kubernetes 会自动引用要信任的系统根证书。当 Central 或 Scanner 必须联系到使用您机构中授权或全局可信合作伙伴机构发布的证书的服务时，您可以使用以下参数来指定对这些服务的信任：

参数	描述
additionalCAs.<certificate_name>	指定要信任的根证书颁发机构的 PEM 编码证书。

4.2.2.2.5. Central

Central 的可配置参数。

- 您必须将持久性存储选项指定为 **hostPath** 或 **persistentVolumeClaim**。
- 用于公开外部访问的中央部署。您必须指定一个参数，可以是 **central.exposure.loadBalancer**、**central.exposure.nodePort** 或 **central.exposure.route**。如果没有为这些参数指定任何值，您必须手动公开 Central，或使用端口转发（port-forwarding）访问它。

参数	描述
central.disableTelemetry	使用 **true** 来禁用在线遥测数据收集。
central.endpointsConfig	Central 的端点配置选项。
central.nodeSelector	如果节点选择器选择污点节点，请使用此参数指定 taint toleration key、value 和 effect。此参数主要用于基础架构节点。
central.tolerations	如果节点选择器选择污点节点，请使用此参数指定 taint toleration key、value 和 effect。此参数主要用于基础架构节点。
central.exposeMonitoring	指定 **true**，以在端口号 **9090** 上为 Central 公开 Prometheus 指标端点。
central.image.registry	用于覆盖 Central 镜像的全局 **image.registry** 参数的自定义 registry。
central.image.name	覆盖默认 Central 镜像名称（**main**）的自定义镜像名称。
central.image.tag	覆盖 Central 镜像默认标签的自定义镜像标签。如果在新安装过程中指定了自己的镜像标签，则您必须在运行 **helm upgrade** 命令升级到新版本时手动增加此标签。如果您 mirror 了自己的 registry 中的镜像，请不要修改原始镜像标签。

参数	描述
central.image.fullRef	Central 镜像的完整参考，包括 registry 地址、镜像名称和镜像标签。为此参数设置值会覆盖 **central.image.registry**、**central.image.name** 和 **central.image.tag** 参数。
central.resources.requests.memory	Central 的内存请求，以覆盖默认值。
central.resources.requests.cpu	Central 的 CPU 请求，以覆盖默认值。
central.resources.limits.memory	Central 的内存限值来覆盖默认值。
central.resources.limits.cpu	Central 的 CPU 限制，以覆盖默认值。
central.persistence.hostPath	Red Hat Advanced Cluster Security for Kubernetes 应该创建数据库卷的节点的路径。红帽不推荐使用这个选项。
central.persistence.persistentVolumeClaim.claimName	您要使用的持久性卷声明(PVC)的名称。
central.persistence.persistentVolumeClaim.createClaim	使用 **true** 创建一个新的持久性卷声明，或 **false** 来使用现有的声明。
central.persistence.persistentVolumeClaim.size	由指定声明管理的持久性卷的大小（以 GiB 为单位）。
central.exposure.loadBalancer.enabled	使用 **true** 来通过使用负载均衡器公开 Central。
central.exposure.loadBalancer.port	要公开 Central 的端口号。默认端口号为 443。
central.exposure.nodePort.enabled	使用 **true** 通过节点端口服务公开 Central。
central.exposure.nodePort.port	要公开 Central 的端口号。当您跳过此参数时，OpenShift Container Platform 会自动分配一个端口号。如果您使用节点端口公开 Red Hat Advanced Cluster Security for Kubernetes，红帽建议您不要指定端口号。
central.exposure.route.enabled	使用 **true** 通过路由公开 Central。此参数仅适用于 OpenShift Container Platform 集群。

4.2.2.2.6. 扫描程序

扫描程序的可配置参数。

参数	描述
scanner.disable	使用 **true** 在没有扫描器的情况下安装 Red Hat Advanced Cluster Security for Kubernetes。当您与 **helm upgrade** 命令一起使用时，Helm 会移除现有的 Scanner 部署。
scanner.replicas	为 Scanner 部署创建的副本数。当您将其与 **scanner.autoscaling** 参数搭配使用时，这个值会设置初始副本数。
scanner.logLevel	为 Scanner 配置日志级别。红帽建议您不要更改日志级别的默认值 (**INFO**)。
scanner.nodeSelector	将节点选择器标签指定为 **label-key: label-value**，以强制 Scanner 仅调度到具有指定标签的节点。
scanner.tolerations	如果节点选择器选择污点节点，请使用此参数为 Scanner 指定污点容限键、值和效果。此参数主要用于基础架构节点。
scanner.autoscaling.disable	使用 **true** 为 Scanner 部署禁用自动扩展。禁用自动扩展时，**minReplicas** 和 **maxReplicas** 参数没有任何效果。
scanner.autoscaling.minReplicas	自动扩展的最小副本数。
scanner.autoscaling.maxReplicas	自动扩展的最大副本数。
scanner.resources.requests.memory	扫描器的内存请求，以覆盖默认值。
scanner.resources.requests.cpu	扫描器的 CPU 请求，以覆盖默认值。
scanner.resources.limits.memory	扫描器的内存限值，以覆盖默认值。
scanner.resources.limits.cpu	扫描器的 CPU 限制，以覆盖默认值。
scanner.dbResources.requests.memory	Scanner 数据库部署的内存请求，以覆盖默认值。
scanner.dbResources.requests.cpu	扫描数据库部署的 CPU 请求，以覆盖默认值。
scanner.dbResources.limits.memory	Scanner 数据库部署的内存限值，以覆盖默认值。
scanner.dbResources.limits.cpu	扫描数据库部署的 CPU 限制，以覆盖默认值。
scanner.image.registry	Scanner 镜像的自定义 registry。

参数	描述
scanner.image.name	覆盖默认扫描程序镜像名称（**scanner**）的自定义镜像名称。
scanner.dbImage.registry	Scanner DB 镜像的自定义 registry。
scanner.dbImage.name	覆盖默认 Scanner DB 镜像名称 (**scanner-db**) 的自定义镜像名称。
scanner.dbNodeSelector	将节点选择器标签指定为 **label-key: label-value**，以强制 Scanner DB 仅调度到具有指定标签的节点。
scanner.dbTolerations	如果节点选择器选择污点节点，请使用此参数为 Scanner DB 指定污点容限键、值和效果。此参数主要用于基础架构节点。

4.2.2.2.7. 自定义

使用这些参数为 Red Hat Advanced Cluster Security for Kubernetes 创建的所有对象指定附加属性。

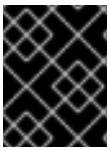
参数	描述
customize.labels	附加到所有对象的自定义标签。
customize.annotations	附加到所有对象的自定义注解。
customize.podLabels	附加到所有部署的自定义标签。
customize.podAnnotations	附加到所有部署的自定义注解。
customize.envVars	所有对象中所有容器的自定义环境变量。
customize.central.labels	附加到 Central 创建的所有对象的自定义标签。
customize.central.annotations	附加到中央创建的所有对象的自定义注解。
customize.central.podLabels	附加到所有中央部署的自定义标签。
customize.central.podAnnotations	附加到所有中央部署的自定义注解。
customize.central.envVars	所有中央容器的自定义环境变量。
customize.scanner.labels	附加到 Scanner 创建的所有对象的自定义标签。
customize.scanner.annotations	附加到 Scanner 创建的所有对象的自定义注解。

参数	描述
customize.scanner.podLabels	附加到所有 Scanner 部署的自定义标签。
customize.scanner.podAnnotations	附加到所有 Scanner 部署的自定义注解。
customize.scanner.envVars	所有 Scanner 容器的自定义环境变量。
customize.scanner-db.labels	附加到 Scanner DB 创建的所有对象的自定义标签。
customize.scanner-db.annotations	附加到 Scanner DB 创建的所有对象的自定义注解。
customize.scanner-db.podLabels	附加到所有 Scanner DB 部署的自定义标签。
customize.scanner-db.podAnnotations	附加到所有 Scanner DB 部署的自定义注解。
customize.scanner-db.envVars	所有 Scanner DB 容器的自定义环境变量。

您还可以使用：

- **customize.other.service/*.labels** 和 **customize.other.service/*.annotations** 参数，为所有对象指定标签和注解。
- 或者，提供特定的服务名称，例如 **customize.other.service/central-loadbalancer.labels** 和 **customize.other.service/central-loadbalancer.annotations** 作为参数，并设置它们的值。

4.2.2.2.8. 高级自定义



重要

本节中指定的参数仅用于信息。红帽不支持带有修改后的命名空间和发行版本名称的 Red Hat Advanced Cluster Security for Kubernetes 实例。

参数	描述
allowNonstandardNamespace	使用 **true** 将 Red Hat Advanced Cluster Security for Kubernetes 部署到默认命名空间 **stackrox** 以外的命名空间中。
allowNonstandardReleaseName	使用 **true** 使用默认 **stackrox-central-services** 之外的发行版本名称部署 Red Hat Advanced Cluster Security for Kubernetes。

4.2.3. 安装 central-services Helm chart

配置 **values-public.yaml** 和 **values-private.yaml** 文件后，安装 **central-services** Helm Chart 来部署集中式组件（Central 和 Scanner）。

流程

- 运行以下命令：

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

1 使用 **-f** 选项指定 YAML 配置文件的路径。

4.2.3.1. 在部署 central-services Helm Chart 后更改配置选项

在部署 **central-services** Helm Chart 后，您可以对任何配置选项进行更改。

流程

1. 使用新值更新 **values-public.yaml** 和 **values-private.yaml** 配置文件。
2. 运行 **helm upgrade** 命令并使用 **-f** 选项指定配置文件：

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

注意

您还可以使用 **--set** 或 **--set-file** 参数指定配置值。但是，这些选项不会被保存，需要您在每次进行更改时手动指定所有选项。

4.2.4. 生成 init 捆绑包

在集群中安装 **SecuredCluster** 资源前，您必须创建一个 init 捆绑包。安装并配置 **SecuredCluster** 的集群，然后使用此捆绑包与 Central 进行身份验证。

您可以使用 **roxctl** CLI 或 RHACS 门户创建 init 捆绑包。

4.2.4.1. 使用 roxctl CLI 生成 init 捆绑包

您可以使用 **roxctl** CLI 创建带有 secret 的 init 捆绑包。

先决条件

您已配置了 **ROX_API_TOKEN** 和 **ROX_CENTRAL_ADDRESS** 环境变量。

- 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>
```

流程

- 运行以下命令以生成包含 secret 的集群 init 捆绑包：

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

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

重要

请确定您安全地存储此捆绑包，因为它包含 secret。您可以使用同一捆绑包来设置多个安全集群。

其他资源

- 安装 **roxctl** CLI

4.2.4.2. 使用 RHACS 门户生成 init 捆绑包

您可以使用 RHACS 门户创建包含 secret 的 init 捆绑包。

流程

1. 根据您的暴露的方法查找 RHACS 门户地址：
   a. 对于路由：

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

   b. 对于负载均衡器：

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

   c. 对于端口转发：
      i. 运行以下命令：

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

      ii. 进入到 **https://localhost:18443/**。
2. 在 RHACS 门户网站中，进入 **Platform Configuration → Integrations**。
3. 进入 **Authentication Tokens** 部分，再点 **Cluster Init Bundle**。
4. 点 **Generate bundle**。
5. 为集群 init 捆绑包输入一个名称并点 **Generate**。
6. 点 **Download Helm Values File** 下载生成的捆绑包。

7. 点 **Download Kubernetes Secret File** 下载生成的捆绑包。

重要

安全地存储此捆绑包，因为它包含 secret。您可以使用同一捆绑包来创建多个安全集群。

下一步

1. 使用 OpenShift Container Platform CLI 使用 init 捆绑包创建资源。
2. 在您要监控的所有集群中安装 Red Hat Advanced Cluster Security for Kubernetes。

4.2.5. 配置 secured-cluster-services Helm chart

本节论述了可用于 **helm install** 和 **helm upgrade** 命令的 Helm Chart 配置参数。您可以使用 **--set** 选项或创建 YAML 配置文件来指定这些参数。

创建以下文件来配置 Helm chart 来安装 Red Hat Advanced Cluster Security for Kubernetes：

- 公共配置文件 **values-public.yaml** ：使用此文件保存所有非敏感配置选项。
- 专用配置文件 **values-private.yaml** ：使用此文件保存所有敏感配置选项。请确定您安全地存储这个文件。

重要

在使用 **secured-cluster-services** Helm Chart 时，不要修改属于 chart 的 **values.yaml** 文件。

4.2.5.1. 配置参数

参数	描述
clusterName	集群的名称。
centralEndpoint	中央端点的地址，包括端口号。如果使用一个支持非 gRPC 的负载均衡器，请使用带有 **ws://** 的端点地址的 WebSocket 协议。
sensor.endpoint	Sensor 端点的地址，包括端口号。
sensor.imagePullPolicy	Sensor 容器的镜像拉取策略。
sensor.serviceTLS.cert	Sensor 使用的内部服务到服务 TLS 证书。
sensor.serviceTLS.key	Sensor 使用的内部服务到服务 TLS 证书密钥。
sensor.resources.requests.memory	Sensor 容器的内存请求。使用此参数覆盖默认值。
sensor.resources.requests.cpu	Sensor 容器的 CPU 请求。使用此参数覆盖默认值。

参数	描述
sensor.resources.limits.memory	Sensor 容器的内存限值。使用此参数覆盖默认值。
sensor.resources.limits.cpu	Sensor 容器的 CPU 限制。使用此参数覆盖默认值。
sensor.nodeSelector	将节点选择器标签指定为 **label-key: label-value**，以强制 Sensor 仅调度到具有指定标签的节点。
sensor.tolerations	如果节点选择器选择污点节点，请使用此参数指定污点容限键、值和 Sensor 的效果。此参数主要用于基础架构节点。
image.main.name	**main（主）** 镜像的名称。
image.collector.name	Collector 镜像的名称。
image.main.registry	用于主镜像的 registry 地址。
image.collector.registry	用于 Collector 镜像的 registry 地址。
image.main.pullPolicy	**main** 镜像的镜像拉取策略。
image.collector.pullPolicy	Collector 镜像的镜像拉取策略。
image.main.tag	使用 **main** 镜像标签。
image.collector.tag	使用 **collector** 镜像标签。
collector.collectionMethod	**EBPF**、**KERNEL_MODULE** 或 **NO_COLLECTION**。
collector.imagePullPolicy	Collector 容器的镜像拉取策略。
collector.complianceImagePullPolicy	Compliance 容器的镜像拉取策略。
collector.disableTaintTolerations	如果指定了 **false**，则容限应用到 Collector，并且收集器 pod 可以调度到具有污点的所有节点上。如果将其指定为 **true**，则不会应用任何容限，且收集器 pod 不会调度到具有污点的节点。
collector.resources.requests.memory	Collector 容器的内存请求。使用此参数覆盖默认值。
collector.resources.requests.cpu	Collector 容器的 CPU 请求。使用此参数覆盖默认值。
collector.resources.limits.memory	Collector 容器的内存限值。使用此参数覆盖默认值。

参数	描述
collector.resources.limits.cpu	Collector 容器的 CPU 限制。使用此参数覆盖默认值。
collector.complianceResources.requests.memory	Compliance 容器的内存请求。使用此参数覆盖默认值。
collector.complianceResources.requests.cpu	Compliance 容器的 CPU 请求。使用此参数覆盖默认值。
collector.complianceResources.limits.memory	Compliance 容器的内存限值。使用此参数覆盖默认值。
collector.complianceResources.limits.cpu	Compliance 容器的 CPU 限制。使用此参数覆盖默认值。
collector.serviceTLS.cert	Collector 使用的内部服务到服务的 TLS 证书。
collector.serviceTLS.key	Collector 使用的内部服务到服务的 TLS 证书密钥。
admissionControl.listenOnCreates	此设置控制 Kubernetes 是否配置为联系 Red Hat Advanced Cluster Security for Kubernetes，并带有 **AdmissionReview** 请求，用于工作负载创建事件。
admissionControl.listenOnUpdates	当将此参数设置为 **false** 时，Red Hat Advanced Cluster Security for Kubernetes 会以 Kubernetes API 服务器不发送对象更新事件的方式创建 **ValidatingWebhookConfiguration**。由于对象更新的卷通常高于对象创建的，所以保留此项为 **false** 会限制准入控制服务的负载，并减少准入控制服务的几率。
admissionControl.listenOnEvents	此设置控制集群是否被配置为联系 Red Hat Advanced Cluster Security for Kubernetes，使用 **AdmissionReview** 请求用于 **exec** 和 **portforward** 事件。Red Hat Advanced Cluster Security for Kubernetes 不支持 OpenShift Container Platform 3.11 中的此功能。
admissionControl.dynamic.enforceOnCreates	此设置控制 Red Hat Advanced Cluster Security for Kubernetes 是否评估策略 ; 如果禁用，则会自动接受所有 AdmissionReview 请求。
admissionControl.dynamic.enforceOnUpdates	此设置控制准入控制服务的行为。您必须把 **listenOnUpdates** 指定为 **true** 才能正常工作。

参数	描述
admissionControl.dynamic.scanInline	如果将这个选项设置为 **true**，则准入控制服务会在做出准入决策前请求镜像扫描。由于镜像扫描需要几秒钟，因此只有在您确保部署前扫描集群中使用的所有镜像（例如，在镜像构建期间通过 CI 集成），才启用此选项。这个选项与 RHACS 门户中的 **Contact image scanners** 选项对应。
admissionControl.dynamic.disableBypass	将它设置为 **true** 以禁用绕过 Admission Controller。
admissionControl.dynamic.timeout	在评估准入检查请求时，Red Hat Advanced Cluster Security for Kubernetes 应该等待的最大时间（以秒为单位）。使用它来设置启用镜像扫描时的请求超时。如果镜像扫描的运行时间超过指定时间，Red Hat Advanced Cluster Security for Kubernetes 接受了请求。
admissionControl.resources.requests.memory	Admission Control 容器的内存请求。使用此参数覆盖默认值。
admissionControl.resources.requests.cpu	Admission Control 容器的 CPU 请求。使用此参数覆盖默认值。
admissionControl.resources.limits.memory	Admission Control 容器的内存限值。使用此参数覆盖默认值。
admissionControl.resources.limits.cpu	Admission Control 容器的 CPU 限制。使用此参数覆盖默认值。
admissionControl.nodeSelector	将节点选择器标签指定为 **label-key: label-value**，以强制 Admission Control 仅调度到具有指定标签的节点。
admissionControl.tolerations	如果节点选择器选择污点节点，请使用此参数指定污点容限键、值以及 Admission Control 的效果。此参数主要用于基础架构节点。
admissionControl.serviceTLS.cert	Admission Control 使用的内部服务到服务的 TLS 证书。
admissionControl.serviceTLS.key	Admission Control 使用的内部服务对服务的 TLS 证书密钥。
registryOverride	使用此参数覆盖默认的 **docker.io** registry。如果使用其他 registry，请指定 registry 的名称。

参数	描述
collector.disableTaintTolerations	如果指定了 **false**，则容限应用到 Collector，Collector pod 可以调度到具有污点的所有节点上。如果您将其指定为 **true**，则不会应用任何容限，Collector pod 不会调度到具有污点的节点。
createUpgraderServiceAccount	指定 **true** 以创建 **sensor-upgrader** 帐户。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。如果您没有创建这个帐户，当 Sensor 没有足够权限时，则必须手动完成将来的升级。
createSecrets	指定 **false** 以跳过 Sensor、Collector 和 Admission Controller 的编配 secret 创建。
collector.slimMode	如果要使用 slim Collector 镜像部署 Collector，请指定 **true**。使用 slim Collector 镜像需要 Central 来提供匹配的内核模块或 eBPF 探测。如果您以离线模式运行 Red Hat Advanced Cluster Security for Kubernetes，您必须从 stackrox.io 下载内核支持软件包，并将其上传到 Central slim Collectors 才能正常工作。否则，您必须确保 Central 可以访问托管在 https://collector-modules.stackrox.io/ 的在线探测存储库。
sensor.resources	Sensor 的资源规格。
admissionControl.resources	Admission Controller 的资源规格。
collector.resources	Collector 的资源规格。
collector.complianceResources	Collector 的 Compliance 容器的资源规格。
exposeMonitoring	如果将此选项设置为 **true**，Red Hat Advanced Cluster Security for Kubernetes 会在 Sensor、Collector 和 Admission Controller 的端口号 9090 上公开 Prometheus 指标端点。
auditLogs.disableCollection	如果将此选项设置为 **true**，Red Hat Advanced Cluster Security for Kubernetes 会禁用用于检测对配置映射和 secret 的访问和修改的审计日志检测功能。
scanner.disable	如果将此选项设置为 **false**，Red Hat Advanced Cluster Security for Kubernetes 会在安全集群中部署轻量级扫描程序和扫描器数据库，以允许扫描 OpenShift Container Registry 上的镜像。只有在 OpenShift 上才支持启用扫描器。默认值为 **true**

参数	描述
scanner.dbTolerations	如果节点选择器选择污点节点，请使用此参数为 Scanner DB 指定污点容限键、值和效果。
scanner.replicas	Collector 的 Compliance 容器的资源规格。
scanner.logLevel	通过设置此参数，您可以修改扫描程序日志级别。使用这个选项仅用于故障排除目的。
scanner.autoscaling.disable	如果将此选项设置为 **true**，Red Hat Advanced Cluster Security for Kubernetes 会禁用 Scanner 部署的自动扩展。
scanner.autoscaling.minReplicas	自动扩展的最小副本数。默认值为 2。
scanner.autoscaling.maxReplicas	自动扩展的最大副本数。默认值为 5。
scanner.nodeSelector	将节点选择器标签指定为 **label-key: label-value**，以强制 Scanner 仅调度到具有指定标签的节点。
scanner.tolerations	如果节点选择器选择污点节点，请使用此参数为 Scanner 指定污点容限键、值和效果。
scanner.dbNodeSelector	将节点选择器标签指定为 **label-key: label-value**，以强制 Scanner DB 仅调度到具有指定标签的节点。
scanner.dbTolerations	如果节点选择器选择污点节点，请使用此参数为 Scanner DB 指定污点容限键、值和效果。
scanner.resources.requests.memory	Scanner 容器的内存请求。使用此参数覆盖默认值。
scanner.resources.requests.cpu	Scanner 容器的 CPU 请求。使用此参数覆盖默认值。
scanner.resources.limits.memory	Scanner 容器的内存限值。使用此参数覆盖默认值。
scanner.resources.limits.cpu	Scanner 容器的 CPU 限制。使用此参数覆盖默认值。
scanner.dbResources.requests.memory	Scanner DB 容器的内存请求。使用此参数覆盖默认值。
scanner.dbResources.requests.cpu	Scanner DB 容器的 CPU 请求。使用此参数覆盖默认值。
scanner.dbResources.limits.memory	Scanner DB 容器的内存限值。使用此参数覆盖默认值。

参数	描述
scanner.dbResources.limits.cpu	Scanner DB 容器的 CPU 限制。使用此参数覆盖默认值。

4.2.5.1.1. 环境变量

您可以采用以下格式指定 Sensor 和 Admission Controller 的环境变量：

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

通过 **customize** 设置，您可以为此 Helm Chart 创建的所有对象指定自定义 Kubernetes 元数据（标签和注解）以及工作负载的其他 pod 标签、Pod 注解和容器环境变量。

配置是分层的，在更通用范围（例如，所有对象）中定义的元数据被覆盖为更通用范围的元数据（例如，仅适用于 Sensor 部署）。

4.2.6. 安装 secured-cluster-services Helm chart

配置 **values-public.yaml** 和 **values-private.yaml** 文件后，安装 **secure-cluster-services** Helm chart 以部署每个集群和每个节点组件（Sensor、Admission Controller 和 Collector）。

小心

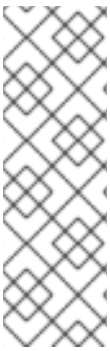
要在具有统一可扩展固件接口(UEFI)以及启用了安全引导机制的系统中安装 Collector，您必须使用 eBPF 探测，因为内核模块没有被签名，且 UEFI 固件无法加载未签名的软件包。收集器在启动时用来识别安全引导状态，并切换到 eBPF 探测（如果需要）。

流程

- 运行以下命令：

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

(1) 使用 **-f** 选项指定 YAML 配置文件的路径。



注意

要使用持续集成(CI)系统部署 **secure-cluster-services** Helm Chart，请将 init 捆绑包 YAML 文件作为环境变量传递给 **helm install** 命令：

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

(1) 如果您使用 base64 编码变量，请使用 **helm install ... -f <(echo "$INIT_BUNDLE_YAML_SECRET" | base64 --decode)** 命令。

4.2.6.1. 在部署 secure-cluster-services Helm chart 后更改配置选项

在部署 **secure-cluster-services** Helm Chart 后，您可以对任何配置选项进行更改。

流程

1. 使用新值更新 **values-public.yaml** 和 **values-private.yaml** 配置文件。
2. 运行 **helm upgrade** 命令并使用 **-f** 选项指定配置文件：

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

1 您必须指定 **--reuse-values** 参数，否则 Helm upgrade 命令重置所有之前配置的设置。

注意

您还可以使用 **--set** 或 **--set-file** 参数指定配置值。但是，这些选项不会被保存，需要您在每次进行更改时手动指定所有选项。

4.2.7. 验证安装

完成安装后，运行几个存在安全漏洞的应用程序并进入 RHACS 门户来评估安全评估结果和策略违反结果。

注意

以下部分中列出的示例应用程序包含关键漏洞，它们旨在验证 Red Hat Advanced Cluster Security for Kubernetes 的构建和部署时间评估功能。

验证安装：

1. 根据您的暴露的方法查找 RHACS 门户地址：
   a. 对于路由：

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

   b. 对于负载均衡器：

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

   c. 对于端口转发：
      i. 运行以下命令：

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

      ii. 进入到 **https://localhost:18443/**。

2. 使用 OpenShift Container Platform CLI 创建新项目：

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

3. 使用关键漏洞启动一些应用程序：

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

Red Hat Advanced Cluster Security for Kubernetes 会在向集群提交后自动扫描这些部署以了解安全风险以及策略违反情况。进入 RHACS 门户以查看违反情况。您可以使用默认用户名 **admin** 和生成的密码登录到 RHACS 门户。

第 5 章 使用 ROXCTL CLI 安装

Red Hat Advanced Cluster Security for Kubernetes 在 OpenShift Container Platform 集群中安装一组服务。本节论述了使用 **roxctl** CLI 在 OpenShift Container Platform 集群中安装 Red Hat Advanced Cluster Security for Kubernetes 的安装过程。

警告

对于生产环境，红帽建议使用 Helm chart 为 Kubernetes 安装 Red Hat Advanced Cluster Security。除非有需要使用此方法的特定安装需要，否则不要使用 **roxctl** 安装方法。

安装流程：

1. 安装 **roxctl** CLI。
2. 使用 **roxctl** CLI 互动安装程序安装 集中组件 （Central 和 Scanner）。
3. 安装 Sensor 以监控集群。

安装前：

- 了解 Red Hat Advanced Cluster Security for Kubernetes 架构 。
- 查看安装 Red Hat Advanced Cluster Security for Kubernetes 的先决条件 。

5.1. 安装 ROXCTL CLI

要安装 Red Hat Advanced Cluster Security for Kubernetes，您必须下载二进制文件来安装 **roxctl** CLI。您可以在 Linux、Windows 或 macOS 上安装 **roxctl**。

5.2. 在 LINUX 中安装 ROXCTL CLI

您可以按照以下流程在 Linux 上安装 **roxctl** CLI 二进制文件。

流程

1. 下载 **roxctl** CLI 的最新版本：

```
$ curl -O https://mirror.openshift.com/pub/rhacs/assets/3.70.2/bin/Linux/roxctl
```

2. 使 **roxctl** 二进制文件可执行：

```
$ chmod +x roxctl
```

3. 将 **roxctl** 二进制文件放到 **PATH** 中的目录中：
   要查看您的 **PATH**，请执行以下命令：

```
$ echo $PATH
```

验证

- 验证您已安装的 **roxctl** 版本：

```
$ roxctl version
```

5.2.1. 在 macOS 上安装 roxctl CLI

您可以按照以下流程在 macOS 中安装 **roxctl** CLI 二进制文件。

流程

1. 下载 **roxctl** CLI 的最新版本：

```
$ curl -O https://mirror.openshift.com/pub/rhacs/assets/3.70.2/bin/Darwin/roxctl
```

2. 从二进制文件中删除所有扩展属性：

```
$ xattr -c roxctl
```

3. 使 **roxctl** 二进制文件可执行：

```
$ chmod +x roxctl
```

4. 将 **roxctl** 二进制文件放到 **PATH** 中的目录中：
要查看您的 **PATH**，请执行以下命令：

```
$ echo $PATH
```

验证

- 验证您已安装的 **roxctl** 版本：

```
$ roxctl version
```

5.2.2. 在 Windows 上安装 roxctl CLI

您可以按照以下流程在 Windows 上安装 **roxctl** CLI 二进制文件。

流程

- 下载 **roxctl** CLI 的最新版本：

```
$ curl -O https://mirror.openshift.com/pub/rhacs/assets/3.70.2/bin/Windows/roxctl.exe
```

验证

- 验证您已安装的 **roxctl** 版本：

```
$ roxctl version
```

5.3. 安装 CENTRAL

Red Hat Advanced Cluster Security for Kubernetes 的主要组件名为 Central。您可以使用交互式安装程序在 OpenShift Container Platform 上安装 Central。您只需要部署 Central 一次，并使用同一安装监控多个独立集群。

5.3.1. 使用交互式安装程序

使用交互式安装程序为您的环境生成所需的 secret、部署配置和部署脚本。

流程

1. 运行交互式 install 命令：

```
$ roxctl central generate interactive
```

2. 按 **Enter** 接受提示的默认值或根据需要输入自定义值。

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

1. 如果要添加自定义 TLS 证书，请提供 PEM 编码证书的文件路径。当您指定自定义证书时，交互式安装程序还会提示您为您要使用的自定义证书提供 PEM 私钥。
2. 要使用 RHACS 门户，您必须使用路由（负载均衡器或节点端口）公开中。
3. 如果您计划在带有 hostPath 卷的 OpenShift Container Platform 上安装 Red Hat Advanced Cluster Security for Kubernetes，您必须修改 SELinux 策略。

警告

在 OpenShift Container Platform 中，对于 hostPath 卷，您必须修改 SELinux 策略以允许访问主机和容器共享的目录。这是因为 SELinux 默认阻止目录共享。要修改 SELinux 策略，请运行以下命令：

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

但是，红帽不推荐修改 SELinux 策略，在 OpenShift Container Platform 上安装时使用 PVC。

在完成时，安装程序会创建一个名为 central-bundle 的文件夹，其中包含用于部署 Central 所需的 YAML 清单和脚本。另外，它显示了您需要运行的脚本的屏幕说明，以部署其他可信证书颁发机构、中部和扫描器，以及登录 RHACS 门户的身份验证说明（如果您回答提示时未提供密码）。

5.3.2. 运行中央安装脚本

运行交互式安装程序后，您可以运行 **setup.sh** 脚本来安装 Central。

流程

1. 运行 **setup.sh** 脚本来配置镜像 registry 访问：

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

2. 创建所需资源：

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

3. 检查部署进度：

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

4. 在 Central 运行后，找到 RHACS 门户 IP 地址并在浏览器中打开。根据您在回答提示时选择的风险，请使用以下方法之一获取 IP 地址。

公开方法	命令	地址	示例
Route	**oc -n stackrox get route central**	在输出中 **HOST/PORT** 列下的地址	**https://central-stackrox.example.route**
节点端口	**oc get node -owide && oc -n stackrox get svc central-loadbalancer**	任何节点的 IP 或主机名，在服务显示的端口中	**https://198.51.100.0:31489**

公开方法	命令	地址	示例
Load Balancer	**oc -n stackrox get svc central-loadbalancer**	在端口 443 上为服务显示 EXTERNAL-IP 或主机名	**https://192.0.2.0**
无	**central-bundle/central/scripts/port-forward.sh 8443**	**https://localhost:8443**	**https://localhost:8443**

注意

如果您在互动安装过程中选择了自动生成的密码，您可以运行以下命令将其记录到 Central：

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

5.4. 安装扫描器（SCANNER）

您可以配置 Red Hat Advanced Cluster Security for Kubernetes，以从各种开源和商业镜像扫描程序中获取镜像数据。

但是，Red Hat Advanced Cluster Security for Kubernetes 还提供一个镜像漏洞扫描程序组件，称为 Scanner。它增强了带有镜像漏洞信息的部署。

红帽建议部署 Scanner，以便它可以扫描所有镜像，包括公共 registry 中的镜像，以了解漏洞。您可以使用 Central 在同一集群中部署 Scanner。

先决条件

- 您必须配置镜像 registry，以允许 Scanner 来下载和扫描镜像。通常，镜像 registry 集成由 Red Hat Advanced Cluster Security for Kubernetes 自动创建。

流程

1. 运行以下命令来配置镜像 registry 访问：

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

2. 脚本完成后，运行以下命令以创建扫描程序服务：

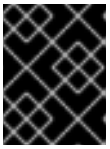
```
$ oc create -R -f central-bundle/scanner
```

5.5. 安装传感器（SENSOR）

要监控集群，您必须部署 Sensor。您必须将 Sensor 部署到要监控的每个集群中。以下步骤描述使用 RHACS 门户添加传感器。

流程

1. 在 RHACS 门户，进入到 **Platform Configuration → Clusters**。
2. 选择 **+ New Cluster**。
3. 为集群指定一个名称。
4. 根据您要部署 Sensor 的位置，为字段提供适当的值。
   - 如果您要在同一集群中部署 Sensor，请接受所有字段的默认值。
   - 如果您要部署到不同的集群中，请将 **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，可以被其他集群访问。
   - 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** ：
     - 使用 **wss://** 为地址加上前缀。
     - 在地址后添加端口号，例如 **ws://stackrox-central.example.com:443**。
5. 点 **Next** 以继续 Sensor 设置。
6. 点 **Download YAML File and Keys** 下载集群捆绑包（zip 归档）。



重要

集群捆绑包 zip 存档包括每个集群的唯一配置和密钥。不要在另一个集群中重复使用相同的文件。

7. 在可以访问受监控的集群的系统中，解压缩并从集群捆绑包中运行 **sensor** 脚本：

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

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

如果您收到部署 Sensor 的所需权限的警告，请按照屏幕说明操作，或与集群管理员联系以获取帮助。

部署 Sensor 后，它会联系中心并提供集群信息。

验证

1. 返回 RHACS 门户并检查部署是否成功。如果成功，则在 #2 部分中会出现一个绿色勾号。如果您没有看到绿色勾选标记，请使用以下命令检查问题：
   - 在 OpenShift Container Platform 中：

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

   - 对于 Kubernetes：

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

2. 点 **Finish** 关闭窗口。

安装后，Sensor 开始向 Red Hat Advanced Cluster Security for Kubernetes 报告安全信息，而 RHACS 门户仪表板开始显示部署、镜像和策略违反情况。

5.6. 验证安装

完成安装后，运行几个存在安全漏洞的应用程序并进入 RHACS 门户来评估安全评估结果和策略违反结果。

注意

以下部分中列出的示例应用程序包含关键漏洞，它们旨在验证 Red Hat Advanced Cluster Security for Kubernetes 的构建和部署时间评估功能。

验证安装：

1. 根据您的暴露的方法查找 RHACS 门户地址：
   a. 对于路由：

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

   b. 对于负载均衡器：

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

   c. 对于端口转发：
      i. 运行以下命令：

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

      ii. 进入到 **https://localhost:18443/**。
2. 使用 OpenShift Container Platform CLI 创建新项目：

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

3. 使用关键漏洞启动一些应用程序：

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

Red Hat Advanced Cluster Security for Kubernetes 会在向集群提交后自动扫描这些部署以了解安全风险以及策略违反情况。进入 RHACS 门户以查看违反情况。您可以使用默认用户名 **admin** 和生成的密码登录到 RHACS 门户。

5.7. 其他资源

- 通过 Helm chart 使用自定义安装 Red Hat Advanced Cluster Security for Kubernetes

第 6 章 卸载 RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES

安装 Red Hat Advanced Cluster Security for Kubernetes 时，它会创建：

- 如果选择了 Operator 安装方法，一个名为 **rhacs-operator** 的命名空间，Operator 将在这个命名空间中安装
- 名为 **stackrox** 的命名空间，或者您创建的 Central 和 SecuredCluster 自定义资源的另外一个命名空间
- 所有组件的 **PodSecurityPolicy** 和 Kubernetes 基于角色的访问控制 (RBAC) 对象
- 命名空间上的额外标签，用于生成的网络策略
- 一个应用程序自定义资源定义 (CRD)，如果它不存在

卸载 Red Hat Advanced Cluster Security for Kubernetes 涉及删除所有这些项目。

6.1. 删除命名空间

您可以使用 OpenShift Container Platform 或 Kubernetes 命令行界面删除 Red Hat Advanced Cluster Security for Kubernetes 创建的命名空间。

流程

- 删除 **stackrox** 命名空间：
  - 在 OpenShift Container Platform 中：

```
$ oc delete namespace stackrox
```

  - 对于 Kubernetes：

```
$ kubectl delete namespace stackrox
```

注意

如果您在不同的命名空间中安装了 RHACS，请在 **delete** 命令中使用该命名空间的名称。

6.2. 删除全局资源

您可以使用 OpenShift Container Platform 或 Kubernetes 命令行界面删除 Red Hat Advanced Cluster Security for Kubernetes 创建的全局资源。

流程

- 删除全局资源：
  - 在 OpenShift Container Platform 中：

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

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

```
$ oc delete ValidatingWebhookConfiguration stackrox
```

- 对于 Kubernetes：

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

```
$ kubectl delete ValidatingWebhookConfiguration stackrox
```

6.3. 删除标签和注解

您可以使用 OpenShift Container Platform 或 Kubernetes 命令行界面删除 Red Hat Advanced Cluster Security for Kubernetes 所创建的标签和注解。

流程

- 删除标签和注解：
  - 在 OpenShift Container Platform 中：

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

  - 对于 Kubernetes：

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