Red Hat Advanced Cluster Security for Kubernetes 4.4

安装

安装 Red Hat Advanced Cluster Security for Kubernetes

Last Updated: 2024-06-28

Red Hat Advanced Cluster Security for Kubernetes 4.4 安装

安装 Red Hat Advanced Cluster Security for Kubernetes

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 高级 RHACS 安装概述 3
1.1. 常规安装指南 3
1.2. 不同平台的安装方法 3
1.3. 不同架构的安装方法 4
1.4. OPENSHIFT CONTAINER PLATFORM 上 RHACS 的安装步骤 4
1.5. KUBERNETES 上 RHACS 的安装步骤 6

第 2 章 RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES 的默认资源要求 7
2.1. 常规 RHACS 要求 7
2.2. 中央服务（自助管理） 8
2.3. 安全的集群服务 11

第 3 章 推荐的 RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES 资源要求 14
3.1. 中央服务（自助管理） 14
3.2. 安全的集群服务 15

第 4 章 在 RED HAT OPENSHIFT 上安装 RHACS 17
4.1. 在 RED HAT OPENSHIFT 中为 RHACS 安装 CENTRAL 服务 17
4.2. 使用 OPERATOR 为 RHACS 配置 CENTRAL 配置选项 50
4.3. 为 RED HAT OPENSHIFT 上的 RHACS 生成并应用 INIT 捆绑包 59
4.4. 在 RED HAT OPENSHIFT 中为 RHACS 安装安全集群服务 62
4.5. 使用 OPERATOR 为 RHACS 配置安全集群服务选项 77
4.6. 验证 RED HAT OPENSHIFT 上的 RHACS 安装 85

第 5 章 在其他平台上安装 RHACS 87
5.1. 在其他平台上安装 RHACS 的高级别概述 87
5.2. 在其他平台上为 RHACS 安装 CENTRAL 服务 87
5.3. 在其他平台上为 RHACS 生成并应用 INIT 捆绑包 112
5.4. 在其他平台上为 RHACS 安装安全集群服务 114
5.5. 在其他平台上验证 RHACS 安装 128

第 6 章 卸载 RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES 129
6.1. 删除命名空间 129
6.2. 删除全局资源 129
6.3. 删除标签和注解 130

第 1 章 高级 RHACS 安装概述

Red Hat Advanced Cluster Security for Kubernetes (RHACS)为自我管理的 Red Hat OpenShift Kubernetes 系统或平台（如 OpenShift Container Platform、Amazon Elastic Kubernetes Service (Amazon EKS)、Google Kubernetes Engine (Google GKE)和 Microsoft Azure Kubernetes Service (Microsoft AKS)）提供安全服务。

有关支持的平台和架构的详情，请查看 Red Hat Advanced Cluster Security for Kubernetes 支持列表 。有关 RHACS 的生命周期支持信息，请参阅 Red Hat Advanced Cluster Security for Kubernetes 支持政策 。

1.1. 常规安装指南

要确定最佳安装体验，请遵循以下准则：

1. 了解此模块中描述的安装平台和方法。
2. 了解 Red Hat Advanced Cluster Security for Kubernetes 架构 。
3. 检查默认资源要求。

1.2. 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

不是所有安装方法都支持所有平台。如需更多信息，请参阅 Red Hat Advanced Cluster Security for Kubernetes Support Matrix。

表 1.1. 平台和推荐的安装方法

平台类型	平台	推荐的安装方法	安装步骤
受管服务平台	Red Hat OpenShift Dedicated (OSD) Azure Red Hat OpenShift (ARO) Red Hat OpenShift Service on AWS (ROSA) Red Hat OpenShift on IBM Cloud	operator （推荐）、Helm chart 或 **roxctl** CLI [1]	• 在 Red Hat OpenShift 中为 RHACS 安装 Central 服务 • 在 Red Hat OpenShift 中为 RHACS 安装安全集群服务

平台类型	平台	推荐的安装方法	安装步骤
	Amazon Elastic Kubernetes Service (Amazon EKS)	Helm chart （推荐）或 **roxctl** CLI [1]	• 在其他平台上为 RHACS 安装 Central 服务 • 在其他平台上为 RHACS 安装安全集群服务
	Google Kubernetes Engine (Google GKE)		
	Microsoft Azure Kubernetes Service (Microsoft AKS)		
自我管理的平台	Red Hat OpenShift Container Platform (OCP)	operator （推荐）、Helm chart 或 **roxctl** CLI [1]	• 在 Red Hat OpenShift 中为 RHACS 安装 Central 服务 • 在 Red Hat OpenShift 中为 RHACS 安装安全集群服务
	Red Hat OpenShift Kubernetes Engine (OKE)		

1. 除非有以下这个安装方法的具体要求，否则不要使用 **roxctl** 安装方法。

1.3. 不同架构的安装方法

Red Hat Advanced Cluster Security for Kubernetes (RHACS)支持以下架构。如需有关支持的平台和架构的信息，请参阅 Red Hat Advanced Cluster Security for Kubernetes 支持列表 。另外，下表提供有关每个架构可用的安装方法的信息。

表 1.2. 每个架构的架构和支持的安装方法

支持的构架	支持的安装方法
AMD64	Operator （首选）、Helm chart 或 **roxctl** CLI （不推荐）
ppc64le (IBM Power)	Operator
s390x (IBM Z 和 IBM® LinuxONE)	

1.4. OPENSHIFT CONTAINER PLATFORM 上 RHACS 的安装步骤

1.4.1. 使用 RHACS Operator 在 Red Hat OpenShift 上安装 RHACS

1. 在 Red Hat OpenShift 集群中，将 RHACS Operator 安装到 **rhacs-operator** 项目或命名空间中。

2. 在包含 Central 的 Red Hat OpenShift 集群中，称为 central 集群，使用 RHACS Operator 将 Central 服务安装到 **stackrox** 项目中。一个中央集群可以保护多个集群。

3. 从中央集群登录到 RHACS web 控制台，然后创建一个 init 捆绑包并下载它。然后，init 捆绑包会在您要保护的集群中安装，称为安全集群。

4. 对于安全集群：

   a. 将 RHACS Operator 安装到 **rhacs-operator** 命名空间中。

   b. 在安全集群中，通过执行以下步骤应用您在 RHACS 中创建的 init 捆绑包：

      - 使用 OpenShift Container Platform Web 控制台导入您创建的 init 捆绑包的 YAML 文件。确保您位于 **stackrox** 命名空间中。

      - 在终端窗口中，运行 **oc create -f <init_bundle>.yaml -n <stackrox** > 命令，指定 init 捆绑包下载的 YAML 文件的路径。

   c. 在安全集群中，使用 RHACS Operator 将安全 Cluster 服务安装到 **stackrox** 命名空间中。在创建这些服务时，请确保在 **Central Endpoint** 字段中输入 Central 的地址和端口号，以便安全集群可以与 Central 通信。

1.4.2. 使用 Helm chart 在 Red Hat OpenShift 上安装 RHACS

1. 添加 RHACS Helm chart 仓库。

2. 在包含 Central 的 Red Hat OpenShift 集群上安装 **central-services** Helm chart，称为 central 集群。

3. 登录到 Central 集群上的 RHACS web 控制台并创建 init 捆绑包。

4. 对于您要保护的每个集群，登录到安全集群并执行以下步骤：

   a. 应用您使用 RHACS 创建的 init 捆绑包。要在安全集群中应用 init 捆绑包，请执行以下步骤之一：

      - 使用 OpenShift Container Platform Web 控制台导入您创建的 init 捆绑包的 YAML 文件。确保您位于 **stackrox** 命名空间中。

      - 在终端窗口中，运行 **oc create -f <init_bundle>.yaml -n <stackrox** > 命令，指定 init 捆绑包下载的 YAML 文件的路径。

   b. 在安全集群中安装 **secured-cluster-services** Helm chart，指定您创建的 init 捆绑包的路径。

1.4.3. 使用 roxctl CLI 在 Red Hat OpenShift 上安装 RHACS

此安装方法也称为 *清单安装方法*。

1. 安装 **roxctl** CLI。

2. 在包含 Central 的 Red Hat OpenShift 集群中执行以下步骤：

   a. 在终端窗口中，使用 **roxctl** CLI 运行交互式 install 命令。

   b. 运行 setup shell 脚本。

c. 在终端窗口中，使用 **oc create** 命令创建 Central 资源。

3. 执行以下操作之一：

- 在 RHACS web 控制台中，创建并下载传感器 YAML 文件和密钥。
- 在安全集群中，使用 **roxctl sensor generate openshift** 命令。

4. 在安全集群中，运行传感器安装脚本。

1.5. KUBERNETES 上 RHACS 的安装步骤

1.5.1. 使用 Helm chart 在 Kubernetes 平台上安装 RHACS

1. 添加 RHACS Helm chart 仓库。
2. 在包含 Central 的集群中安装 **central-services** Helm Chart，称为 Central 集群。
3. 从 Central 集群中登录到 RHACS web 控制台，并创建一个要在您要保护的集群中安装的 init 捆绑包，称为安全集群。
4. 对于每个安全集群：

a. 应用您使用 RHACS 创建的 init 捆绑包。登录到安全集群并运行 **kubectl create -f <init_bundle>.yaml -n <stackrox** > 命令，指定 init 捆绑包下载的 YAML 文件的路径。

b. 在安全集群中安装 **secured-cluster-services** Helm chart，指定之前创建的 init 捆绑包的路径。

1.5.2. 使用 roxctl CLI 在 Kubernetes 平台上安装 RHACS

此安装方法也称为 *清单安装方法*。

1. 安装 **roxctl** CLI。
2. 在包含 Central 的 Kubernetes 集群中，执行以下步骤：

a. 在终端窗口中，使用 **roxctl** CLI 运行交互式 install 命令。

b. 运行 setup shell 脚本。

c. 在终端窗口中，使用 **kubectl create** 命令创建 Central 资源。

3. 执行以下操作之一：

- 在 RHACS web 控制台中，创建并下载传感器 YAML 文件和密钥。
- 在您要保护的集群中，称为安全集群，使用 **roxctl sensor generate openshift** 命令。

4. 在安全集群中，运行传感器安装脚本。

第 2 章 RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES 的默认资源要求

2.1. 常规 RHACS 要求

在安装 RHACS 之前，您的系统必须满足几个要求。

警告

您不能在以下位置安装 RHACS：

- 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。

要安装 RHACS，您必须有以下系统之一：

- OpenShift Container Platform 版本 4.11 或更高版本，以及带有 Red Hat Enterprise Linux CoreOS (RHCOS)或 Red Hat Enterprise Linux (RHEL)支持的操作系统的集群节点
- 受支持的受管 Kubernetes 平台，以及具有 Amazon Linux、CentOS、Container-Optimized OS (Google、Red Hat Enterprise Linux CoreOS (RHCOS)、Debian、Red Hat Enterprise Linux (RHEL)或 Ubuntu)支持的受管 Kubernetes 平台和集群节点
  有关支持的平台和架构的详情，请查看 Red Hat Advanced Cluster Security for Kubernetes 支持列表。有关 RHACS 的生命周期支持信息，请参阅 Red Hat Advanced Cluster Security for Kubernetes 支持政策。

以下最低要求和建议适用于集群节点。

架构

amd64,**ppc64le**, 或 **s390x**

注意

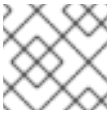
从 RHACS 4.3 开始，在 IBM Power (**ppc64le**)、IBM Z (s390x)和 IBM® LinuxONE (**s390x**)集群中支持 Central 和安全集群服务。

处理器

需要 3 个 CPU 内核。

内存

需要 6 GiB RAM。



注意

请参阅每个组件的默认内存和 CPU 要求，并确保节点大小可以支持它们。

存储

安装 Central 的集群需要持久性卷声明(PVC)。强烈建议您在启用了 Scanner V4 的安全集群中。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。

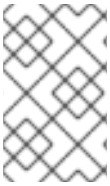
重要

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

如果您计划使用 Helm chart 安装 RHACS，您必须满足以下要求：

- 如果您要使用 Helm chart 安装和配置 RHACS，则必须具有 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。
- 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 Authentication 。

2.2. 中央服务（自助管理）



注意

如果您使用 Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service)，则不需要查看 Central 服务的要求，因为它们由红帽管理。您只需要查看安全集群服务的要求。

Central 服务包含以下组件：

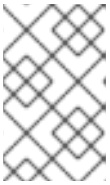
- Central
- 扫描程序

2.2.1. Central

名为 Central 的容器化服务处理 API 交互和 RHACS Web 门户访问，而名为 Central DB (PostgreSQL 13) 的容器化服务处理数据持久性。

Central DB 和 Scanner V4 需要安装 Central 的集群中的持久性存储。

- 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，并收集有关容器中运行的网络流量和进程的数据。如果没有这些更新，如果在集群中添加新节点，或者更新节点的操作系统，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、CPU 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

Central 需要 Central DB 存储数据。下表列出了安装和运行 Central DB 所需的最小内存和存储值。

Central DB	CPU	内存	存储
Request（请求）	4 个核	8 GiB	100 GiB
限制	8 个内核	16 GiB	100 GiB

2.2.2. 扫描程序

从版本 4.4 开始，RHACS 包括两个镜像漏洞扫描程序： StackRox Scanner 和 Scanner V4。计划在以后的发行版本中删除 StackRox Scanner，但版本 4.4 需要执行节点和平台扫描。扫描程序 V4 是首选的镜像扫描程序，因为它通过 StackRox 扫描器提供额外的功能，如扩展语言和操作系统支持以及来自其他漏洞数据库的数据。

内存和 CPU 要求

stackrox Scanner

此表中的要求基于默认的 2 个副本。

stackrox Scanner	CPU	内存
Request（请求）	2 个内核	3000 MiB
限制	4 个核	8000 MiB

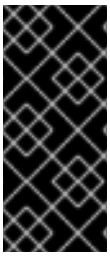
StackRox Scanner-DB

StackRox 扫描程序需要 Scanner-DB 来存储数据。下表列出了安装和运行 Scanner-DB 所需的最小内存和存储值。

Scanner-DB	CPU	内存
Request（请求）	0.2 个内核	512 MiB
限制	2 个内核	4000 MiB

scanner V4 （技术预览）

扫描程序 V4 是可选的。此表中的要求基于默认的 2 个副本。



重要

扫描程序 V4 只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅技术预览功能支持范围。

scanner V4 Indexer	CPU	内存
Request（请求）	2 个内核	3000 MiB
限制	4 个核	6 GiB

此表中的要求基于默认的 2 个副本。

scanner V4 Matcher	CPU	内存
Request（请求）	2 个内核	8 GiB
限制	4 个核	10 GiB

扫描程序 V4 需要 Scanner V4 DB 来存储数据。下表列出了安装和运行 Scanner V4 DB 所需的最小内存和存储值。对于 Scanner V4 DB，需要一个 PVC 来确保最佳性能。这个 PVC 必须是 50 GiB。

扫描程序 V4 DB	CPU	内存
Request（请求）	0.2 个内核	3 GiB
限制	2 个内核	4 GiB

2.3. 安全的集群服务

安全集群服务包含以下组件：

- Sensor
- 准入控制器
- Collector

2.3.1. Sensor

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 CPU 要求
下表列出了在安全集群中安装和运行传感器所需的最小内存和存储值。

Sensor	CPU	内存
Request（请求）	2 个内核	4 GiB
限制	4 个核	8 GiB

2.3.2. 准入控制器

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 CPU 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）	0.05 个内核	100 MiB
限制	0.5 个内核	500 MiB

2.3.3. Collector

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。收集器 Pod 有三个容器。第一个容器是收集器，它实际监控和报告节点上的运行时活动。另外两个是 compliance 和 node-inventory。

集合要求
要使用 **CORE_BPF** 集合方法，基本内核必须支持 BTF，并且 BTF 文件必须可供收集器使用。通常，内核版本必须高于 5.8 （适用于 RHEL 节点的4.18）和 **CONFIG_DEBUG_INFO_BTF** 配置选项必须被设置。

收集器在以下列表中显示的标准位置查找 BTF 文件：

例 2.1. BTF 文件位置

```
/sys/kernel/btf/vmlinux
/boot/vmlinux-<kernel-version>
/lib/modules/<kernel-version>/vmlinux-<kernel-version>
/lib/modules/<kernel-version>/build/vmlinux
/usr/lib/modules/<kernel-version>/kernel/vmlinux
/usr/lib/debug/boot/vmlinux-<kernel-version>
/usr/lib/debug/boot/vmlinux-<kernel-version>.debug
/usr/lib/debug/lib/modules/<kernel-version>/vmlinux
```

如果存在这些文件，则内核可能会支持 BTF，**CORE_BPF** 是可配置的。

内存和 CPU 要求
默认情况下，收集器服务运行 3 个副本。下表列出了每个副本的请求和限值，以及收集器副本的总和。

收集器容器

类型	CPU	内存
Request（请求）	0.06 内核	320 MiB
限制	0.9 个内核	1000 MiB

Compliance 容器

类型	CPU	内存
Request（请求）	0.01 个内核	10 MiB
限制	1 个内核	2000 MiB

node-inventory 容器

类型	CPU	内存
Request（请求）	0.01 个内核	10 MiB
限制	1 个内核	500 MiB

收集器副本要求总数

类型	CPU	内存
Request（请求）	0.07 个内核	340 MiB
限制	2.75 个内核	3500 MiB

2.3.4. scanner V4 （技术预览）

扫描程序 V4 是可选的。如果在安全集群中安装 Scanner V4，则应用以下要求。

重要

扫描程序 V4 只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅技术预览功能支持范围。

此表中的要求基于默认的 2 个副本。

scanner V4 Indexer	CPU	内存
Request（请求）	2 个内核	3000 MiB
限制	4 个核	6 GiB

扫描程序 V4 需要 Scanner V4 DB 来存储数据。下表列出了安装和运行 Scanner V4 DB 所需的最小内存和存储值。对于 Scanner V4 DB，强烈建议使用 PVC，因为它可以确保最佳性能。PVC 应该为 10 GiB。

扫描程序 V4 DB	CPU	内存
Request（请求）	0.2 个内核	3 GiB
限制	2 个内核	4 GiB

第 3 章 推荐的 RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES 资源要求

推荐的资源指南是通过执行集中测试在给定数量命名空间中创建以下对象来实现的：

- 10 个部署，有 3 个 pod 副本处于睡眠状态，挂载 4 个 secret、4 个配置映射
- 10 个服务，每个服务都指向之前部署的 TCP/8080 和 TCP/8443 端口
- 1 个路由指向上一个服务的第一个路由
- 包含 2048 个随机字符串字符的 10 个 secret
- 10 个配置映射包含 2048 个随机字符串字符

在分析结果的过程中，部署数量被识别为增加使用资源的主要因素。并且，我们正在对所需资源的估算使用部署数量。

其他资源

- 默认资源要求

3.1. 中央服务（自助管理）

注意

如果您使用 Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service)，则不需要查看 Central 服务的要求，因为它们由红帽管理。您只需要查看安全集群服务的要求。

Central 服务包含以下组件：

- Central
- 扫描程序

注意

有关扫描程序的默认资源要求，请查看默认资源要求页面。

3.1.1. Central

内存和 CPU 要求

下表列出了为一个安全集群运行 Central 所需的最小内存和 CPU 值。表包括并发 Web 门户用户的数量。

部署	并发 Web 门户用户	CPU	内存
< 25,000	1 个用户	2 个内核	8 GiB
< 25,000	< 5 个用户	6 个内核	12 GiB

部署	并发 Web 门户用户	CPU	内存
< 50,000	1 个用户	2 个内核	12 GiB
< 50,000	< 5 个用户	6 个内核	16 GiB

3.1.2. 扫描程序

stackrox Scanner 内存和 CPU 要求
下表列出了 Central 集群中 StackRox Scanner 部署所需的最小内存和 CPU 值。表包括在所有安全集群中部署的唯一镜像数量。

唯一镜像	Replicas	CPU	内存
< 100	1 个副本	1 个内核	1.5 GiB
< 500	1 个副本	2 个内核	2.5 GiB
< 2000	2 个副本	2 个内核	2.5 GiB
< 5000	3 个副本	2 个内核	2.5 GiB

其他资源

- 默认资源要求

3.2. 安全的集群服务

安全集群服务包含以下组件：

- Sensor
- 准入控制器
- Collector

注意

本页中不包含收集器组件。默认资源要求列出了在默认的资源要求页面中。

3.2.1. Sensor

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 CPU 要求
下表列出了在安全集群中运行的 Sensor 所需的最小内存和 CPU 值。

Deployments	每个部署的 Pod	CPU	内存
< 25,000	3	2 个内核	8 GiB
< 50,000	3	2 个内核	16 GiB

3.2.2. 准入控制器

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 CPU 要求
下表列出了在安全集群中运行的准入控制器所需的最小内存和 CPU 值。

Deployments	每个部署的 Pod	CPU	内存
< 25,000	3	0.5 个内核	600 MiB
< 50,000	3	0.5 个内核	1200 MiB

第 4 章 在 RED HAT OPENSHIFT 上安装 RHACS

4.1. 在 RED HAT OPENSHIFT 中为 RHACS 安装 CENTRAL 服务

Central 是包含 RHACS 应用程序管理界面和服务的资源。它处理数据持久性、API 交互和 RHACS 门户访问。您可以使用同一中实例来保护多个 OpenShift Container Platform 或 Kubernetes 集群。

您可以使用以下方法之一在 OpenShift Container Platform 或 Kubernetes 集群上安装 Central：

- 使用 Operator 安装
- 使用 Helm chart 安装
- 使用 **roxctl** CLI 安装（除非有需要使用它的特定安装需要）

4.1.1. 使用 Operator 安装 Central

4.1.1.1. 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。

前提条件

- 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 4.11 或更高版本。有关支持的平台和架构的详情，请查看 Red Hat Advanced Cluster Security for Kubernetes 支持列表 。有关 RHACS 的生命周期支持信息，请参阅 Red Hat Advanced Cluster Security for Kubernetes 支持政策 。

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 的信息，然后点 **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。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，您必须手动批准更新请求，才能将 Operator 更新至最新版本。

重要

如果选择手动更新，在更新安装了 Central 的集群中的 RHACS Operator 时，您必须更新所有安全集群中的 RHACS Operator。安装 Central 的安全集群和集群必须具有相同的版本，以确保最佳功能。

6. 点 **Install**。

验证

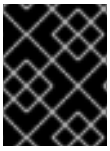
- 安装完成后，进入 **Operators → Installed Operators**，以验证 Red Hat Advanced Cluster Security for Kubernetes Operator 的状态是否为 **Succeeded**。

下一步

- 将 Operator 安装到 **rhacs-operator** 项目中。使用该 Operator，将 **Central** 自定义资源安装到 **stackrox** 项目中。

4.1.1.2. 使用 Operator 方法安装 Central

Red Hat Advanced Cluster Security for Kubernetes 的主要组件被称为 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** 自定义资源安装依赖于 Central 生成的证书。

先决条件

- 您必须使用 OpenShift Container Platform 4.11 或更高版本。有关支持的平台和架构的详情，请查看 Red Hat Advanced Cluster Security for Kubernetes 支持列表 。有关 RHACS 的生命周期支持信息，请参阅 Red Hat Advanced Cluster Security for Kubernetes 支持政策 。

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. 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**。红帽建议您使用 **stackrox** 作为项目名称。
5. 在 **Provided APIs** 部分下，选择 **Central**。点 **Create Central**。
6. 可选：如果您使用声明性配置，在 **Configure via:** 旁边点 **YAML 视图** 并添加声明性配置的信息，如下例所示：

```
...
spec:
  central:
    declarativeConfiguration:
      configMaps:
      - name: "<declarative-configs>" 1
      secrets:
      - name: "<sensitive-declarative-configs>" 2
...
```

1. 将 <declarative-configs> 替换为您要使用的配置映射的名称。
2. 将 <sensitive-declarative-configs> 替换为您要使用的 secret 的名称。

7. 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。
8. 您可以为 Central 配置可用选项：
   - 中央组件设置：

设置	Description
管理员密码	包含管理员密码的 secret。如果您不希望 RHACS 为您生成密码，请使用此字段。
公开	使用路由、负载均衡器或节点端口公开 Central 的设置。请参阅"为 Red Hat OpenShift 上安装 RHACS 的 Central 服务"部分中的 **central.exposure.<parameter** > 信息。
面向用户的 TLS 证书 secret	如果要在 Central 中终止 TLS 并提供自定义服务器证书，请使用此字段。

设置	Description
监控	为 Central 配置监控端点。请参阅"在 Red Hat OpenShift 上安装 RHACS 中的"Public 配置文件"部分中的 **central.exposeMonitoring** 参数。
Persistence	这些字段配置 Central 应如何存储其持久数据。使用持久性卷声明(PVC)获得最佳结果，特别是当您使用 Scanner V4 时。请参阅"为 Red Hat OpenShift 上安装 RHACS 的 Central 服务"部分中的 **central.persistence.<parameter** > 信息。
中央数据库设置	Central DB 的设置，包括数据持久性。请参阅"为 Red Hat OpenShift 上安装 RHACS"中的"Public 配置文件"部分中的 **central.db.<parameter** > 信息。
Resources	如果您需要覆盖内存和 CPU 资源的默认设置，请在咨询文档后使用这些字段。如需更多信息，请参阅"安装"一章中的 "Recommended resource requirements for RHACS" 部分。
容限（Tolerations）	使用此参数将 Central 配置为仅在特定节点上运行。请参阅"为 Red Hat OpenShift 上安装 RHACS 的 Central 服务"部分中的 **central.tolerations** 参数部分。

- **扫描程序组件** 设置：默认扫描程序的设置，也称为 StackRox Scanner。请参阅"为 Red Hat OpenShift 上安装 RHACS 的 Central 服务"部分中的 "Scanner" 表。
- **扫描程序 V4 组件** 设置 ：可选扫描程序 V4 扫描程序的设置，包括在版本 4.4 及更高版本中。它目前没有默认启用。您可以同时启用 StackRox Scanner 和 Scanner V4 以进行并发使用。请参阅"为 Red Hat OpenShift 上安装 RHACS 的 Central 服务"部分中的 "Scanner V4" 表。

重要

扫描程序 V4 只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅技术预览功能支持范围。

启用 Scanner V4 时，您可以配置以下选项：

设置	Description
indexer	对镜像进行索引并创建查找报告的过程。您可以配置副本和自动扩展、资源和容限。在更改默认资源值前，请参阅"对于 RHACS 的默认资源要求"中的"Scanner V4"部分，以及"Installation"章节中的"推荐资源要求"。

设置	Description
matcher	对 Scanner V4 DB 中存储的漏洞数据执行来自索引器报告的漏洞匹配的过程。您可以配置副本和自动扩展、资源和容限。在更改默认资源值前，请参阅"对于 RHACS 的默认资源要求"中的"Scanner V4"部分，以及"Installation"章节中的"推荐资源要求"。
DB	存储 Scanner V4 信息的数据库，包括漏洞数据和索引报告。您可以配置持久性、资源和容限。如果使用 Scanner V4，则 Central 集群中需要一个持久性卷声明(PVC)。强烈建议在安全集群中使用 PVC 来获得最佳性能。在更改默认资源值前，请参阅"对于 RHACS 的默认资源要求"中的"Scanner V4"部分，以及"Installation"章节中的"推荐资源要求"。

- **出口** ：出站网络流量的设置，包括 RHACS 是否应该以在线（连接）或离线（断开连接）模式运行。
- **TLS** ：使用此字段添加额外的可信根证书颁发机构(CA)。
- **高级配置** ：您可以使用这些字段执行以下操作：
  - 指定额外的镜像 pull secret
  - 添加自定义环境变量来为受管 pod 的容器设置
  - 启用 Red Hat OpenShift 监控

9. 点 **Create**。

注意

如果您使用集群范围代理，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。
4. 在您要监控的每个集群中安装安全集群服务。

其他资源

- Red Hat Advanced Cluster Security for Kubernetes 的默认资源要求
- 推荐的 Red Hat Advanced Cluster Security for Kubernetes 资源要求
- 公共配置文件

4.1.1.3. 在 PostgreSQL 实例中置备数据库

此步骤是可选的。您可以使用现有的 PostgreSQL 基础架构为 RHACS 置备数据库。使用本节中的说明来配置 PostgreSQL 数据库环境，创建用户、数据库、架构、角色和授予所需的权限。

流程

1. 创建一个新用户：

```
CREATE USER stackrox WITH PASSWORD <password>;
```

2. 创建数据库：

```
CREATE DATABASE stackrox;
```

3. 连接到数据库：

```
\connect stackrox
```

4. 创建用户模式：

```
CREATE SCHEMA stackrox;
```

5. （可选）撤销公共的权利：

```
REVOKE CREATE ON SCHEMA public FROM PUBLIC;
REVOKE USAGE ON SCHEMA public FROM PUBLIC;
REVOKE ALL ON DATABASE stackrox FROM PUBLIC;
```

6. 创建角色：

```
CREATE ROLE readwrite;
```

7. 为角色授予连接权限：

```
GRANT CONNECT ON DATABASE stackrox TO readwrite;
```

8. 为 **readwrite** 角色添加所需的权限：

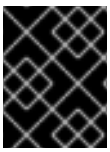
```
GRANT USAGE ON SCHEMA stackrox TO readwrite;
GRANT USAGE, CREATE ON SCHEMA stackrox TO readwrite;
GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA stackrox TO
readwrite;
ALTER DEFAULT PRIVILEGES IN SCHEMA stackrox GRANT SELECT, INSERT, UPDATE,
DELETE ON TABLES TO readwrite;
GRANT USAGE ON ALL SEQUENCES IN SCHEMA stackrox TO readwrite;
ALTER DEFAULT PRIVILEGES IN SCHEMA stackrox GRANT USAGE ON SEQUENCES
TO readwrite;
```

9. 将 **readwrite** 角色分配给 **stackrox** 用户：

```
GRANT readwrite TO stackrox;
```

4.1.1.4. 使用 Operator 安装带有外部数据库的 Central

Red Hat Advanced Cluster Security for Kubernetes 的主要组件被称为 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** 自定义资源安装依赖于 Central 生成的证书。

有关 RHACS 数据库的更多信息，请参阅 数据库覆盖范围。

先决条件

- 您必须使用 OpenShift Container Platform 4.11 或更高版本。如需有关支持的 OpenShift Container Platform 版本的更多信息，请参阅 Red Hat Advanced Cluster Security for Kubernetes 支持列表。
- 您必须在数据库实例中有一个支持 PostgreSQL 13 和具有以下权限的用户的数据库：
  - 对数据库的连接权利。
  - schema 的 **Usage** 和 **Create**。
  - 对 schema 中的所有表的 **Select**, **Insert**, **Update**, 和 **Delete** 权限。
  - 对 schema 中所有序列的 **Usage**。

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. 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**。红帽建议您使用 **stackrox** 作为项目名称。

5. 使用 OpenShift Container Platform Web 控制台或终端在部署的命名空间中创建密码 secret。
   - 在 OpenShift Container Platform web 控制台中进入 **Workloads → Secrets** 页面。使用密钥 **password** 和值创建一个 **Key/Value secret**，作为纯文本文件的路径，其中包含调配数据库的超级用户密码。
   - 或者，在终端中运行以下命令：

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

   1 如果使用 Kubernetes，请输入 **kubectl** 而不是 **oc**。

   2 使用纯文本密码的文件的路径替换 **password.txt**。

6. 返回到 OpenShift Container Platform Web 控制台中的 Red Hat Advanced Cluster Security for Kubernetes operator 页面。在 **Provided APIs** 部分下，选择 **Central**。点 **Create Central**。
7. 可选：如果您使用声明性配置，在 **Configure via:** 旁边点 **YAML 视图**。
8. 添加声明性配置的信息，如下例所示：

```
...
spec:
  central:
    declarativeConfiguration:
      configMaps:
      - name: <declarative-configs> 1
      secrets:
      - name: <sensitive-declarative-configs> 2
...
```

   1 将 <declarative-configs> 替换为您要使用的配置映射的名称。

   2 将 <sensitive-declarative-configs> 替换为您要使用的 secret 的名称。

9. 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。
10. 进入 **Central 组件设置 → Central DB Settings**。
11. 对于 **Administrator Password**，将引用的 secret 指定为 **external-db-password**（或之前创建的密码的 secret 名称）。
12. 对于 **Connection String**，以 **keyword=value** 格式指定连接字符串，例如 host=< **host> port=5432 database=stackrox user=stackrox sslmode=verify-ca**
13. 对于 **Persistence → PersistentVolumeClaim → Claim Name**，请删除 **central-db**。
14. 如果需要，您可以指定证书颁发机构，以便在数据库证书和 Central 之间有信任。要添加此功能，进入 YAML 视图并在顶级 spec 下添加一个 TLS 块，如下例所示：

```
spec:
  tls:
    additionalCAs:
```

```
- name: db-ca
  content: |
    <certificate>
```

15. 点 **Create**。

注意

如果您使用集群范围代理，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。
4. 在您要监控的每个集群中安装安全集群服务。

其他资源

- Central 配置选项
- PostgreSQL 连接字符串 Docs

4.1.1.5. 使用 Operator 方法验证中央安装

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 **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 Red Hat OpenShift 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 Red Hat OpenShift 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 控制台来查找到 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。在完全配置 RHACS 前（例如，您拥有 **Central** 资源，并至少安装并配置一个 **SecuredCluster** 资源），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. 在您要监控的每个集群中安装安全集群服务。

4.1.2. 使用 Helm chart 安装 Central

您可以使用 Helm chart 安装 Central，而无需自定义任何自定义，使用默认值，或使用带有额外自定义配置参数的 Helm chart。

4.1.2.1. 使用 Helm chart 安装 Central，而无需自定义

您可以在没有自定义的情况下在集群中安装 RHACS。您必须添加 Helm Chart 仓库并安装 **central-services** Helm Chart，以安装 Central 和 Scanner 的集中组件。

4.1.2.1.1. 添加 Helm Chart 仓库

流程

- 添加 RHACS chart 存储库。

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

Red Hat Advanced Cluster Security for Kubernetes 的 Helm 仓库包括用于安装不同组件的 Helm chart，包括：

- 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 per-cluster 和 per-node 组件 (Sensor、Admission Controller、Collector 和 Scanner-slim)。

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/
```

4.1.2.1.2. 在不自定义的情况下安装 central-services Helm chart

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

前提条件

- 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 Authentication 。

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.username=<username> \ 1
  --set imagePullSecrets.password=<password> \ 2
  --set central.exposure.route.enabled=true
```

1. 为 Red Hat Container Registry 身份验证包含 pull secret 的用户名。
2. 包括 Red Hat Container Registry 身份验证的 pull secret 密码。

- 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.username=<username> \ 1
  --set imagePullSecrets.password=<password> \ 2
  --set central.exposure.loadBalancer.enabled=true
```

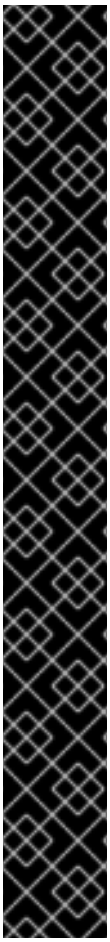
1. 为 Red Hat Container Registry 身份验证包含 pull secret 的用户名。
2. 包括 Red Hat Container Registry 身份验证的 pull secret 密码。

- 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.username=<username> \ 1
  --set imagePullSecrets.password=<password> 2
```

-

1. 为 Red Hat Container Registry 身份验证包含 pull secret 的用户名。
2. 包括 Red Hat Container Registry 身份验证的 pull secret 密码。



重要

- 如果要在需要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
```

- 如果您已在安装的命名空间中创建了一个或多个镜像 pull secret，而不是使用用户名和密码，您可以使用 **--set imagePullSecrets.useExisting="<pull-secret-1;pull-secret-2>"**。
- 不要使用镜像 pull secret：
  - 如果您要从 **quay.io/stackrox-io** 或不需要身份验证的专用网络中的 registry 拉取镜像。使用 **--set imagePullSecrets.allowNone=true**，而不是指定用户名和密码。
  - 如果您已经在安装的命名空间中的默认服务帐户中配置了镜像 pull secret。使用 **--set imagePullSecrets.useFromDefaultServiceAccount=true**，而不是指定用户名和密码。

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。

4.1.2.2. 使用带有自定义的 Helm chart 安装 Central

您可以使用 **helm install** 和 **helm upgrade** 命令的 Helm Chart 配置参数在 Red Hat OpenShift 集群上安装 RHACS。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。确保您安全地存储这个文件。
- 配置文件 **declarative-config-values.yaml** ：如果您使用声明性配置将声明性配置挂载添加到 Central，请创建此文件。

4.1.2.2.1. 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。

4.1.2.2.1.1. 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	Description
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**。

4.1.2.2.1.2. 代理配置

如果要在需要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	Description
env.proxyConfig	您的代理配置。

4.1.2.2.1.3. 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**
- **central.db.serviceTLS.cert**
- **central.db.serviceTLS.key**
- **central.db.password.value**
- 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	Description
central.jwtSigner.key	RHACS 应该用来签名 JSON Web 令牌(JWT)进行身份验证的私钥。
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。

参数	Description
central.defaultTLS.cert	Central 应该使用的用户面向用户的证书。RHACS 将这个证书用于 RHACS 门户。 ● 对于新安装，您必须提供证书，否则 RHACS 使用自签名证书安装 Central。 ● 如果要升级，RHACS 将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 RHACS 使用自签名证书安装 Central。 ● 如果要升级，RHACS 将使用现有证书及其密钥。
central.db.password.value	Central 数据库的连接密码。
central.adminPassword.value	用于登录到 RHACS 的管理员密码。
central.adminPassword.htpasswd	用于登录到 RHACS 的管理员密码。此密码以散列格式存储，使用 bcrypt。
central.db.serviceTLS.cert	Central DB 服务应用于部署 Central DB 的内部证书。
central.db.serviceTLS.key	Central DB 服务应使用的内部证书的私钥。
central.db.password.value	用于连接到 Central DB 的密码。

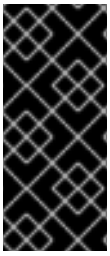
注意

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>
```

4.1.2.2.1.4. 扫描程序

StackRox Scanner 和 Scanner V4 的可配置参数（技术预览）。



重要

扫描程序 V4 只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅技术预览功能支持范围。

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**
- **scannerV4.db.password.value**
- **scannerV4.indexer.serviceTLS.cert**
- **scannerV4.indexer.serviceTLS.key**
- **scannerV4.matcher.serviceTLS.cert**
- **scannerV4.matcher.serviceTLS.key**
- **scannerV4.db.serviceTLS.cert**
- **scannerV4.db.serviceTLS.key**

参数	Description
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 RHACS 自动在内部创建和使用其值。
scanner.serviceTLS.cert	StackRox Scanner 服务用于部署 StackRox 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。

参数	Description
scannerV4.db.password.value	用于通过 Scanner V4 数据库进行身份验证的密码。不要修改此参数，因为 RHACS 自动在内部创建和使用其值。
scannerV4.db.serviceTLS.cert	Scanner V4 DB 服务用于部署 Scanner V4 数据库的内部证书。
scannerV4.db.serviceTLS.key	Scanner V4 DB 服务应使用的内部证书的私钥。
scannerV4.indexer.serviceTLS.cert	Scanner V4 服务用于部署 Scanner V4 Indexer 的内部证书。
scannerV4.indexer.serviceTLS.key	Scanner V4 Indexer 使用的内部证书的私钥。
scannerV4.matcher.serviceTLS.cert	Scanner V4 服务用于部署 Scanner V4 Matcher 的内部证书。
scannerV4.matcher.serviceTLS.key	Scanner V4 Matcher 应该使用的内部证书的私钥。

4.1.2.2.2. 公共配置文件

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

4.1.2.2.2.1. 镜像 pull secret

镜像拉取 secret 是从 registry 中拉取镜像所需的凭证。

参数	Description
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**。

4.1.2.2.2.2. 镜像

镜像声明配置来设置主 registry，Helm Chart 用来解析 **central.image**、**scanner.image**、**scanner.dbImage**、**scannerV4.image** 和 **scannerV4.db.image** 参数的镜像。

参数	Description
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**。

4.1.2.2.2.3. 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	Description
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	要安装 RHACS 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** 在离线模式下使用 RHACS。

4.1.2.2.2.4. 其他可信证书颁发机构

RHACS 自动引用要信任的系统根证书。当 Central 时，StackRox Scanner 或 Scanner V4 必须联系到使用您机构中颁发机构发布的证书或全局可信合作伙伴机构发布的服务的服务，您可以使用以下参数来指定对这些服务的信任：

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

4.1.2.2.2.5. 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。

下表包含外部 PostgreSQL 数据库的设置。

参数	Description
central.declarativeConfiguration.mounts.configMaps	挂载用于声明配置的配置映射。
Central.declarativeConfiguration.mounts.secrets	挂载用于声明配置的 secret。
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。
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。
central.resources.requests.cpu	Central 的 CPU 请求。
central.resources.limits.memory	Central 的内存限值。
central.resources.limits.cpu	Central 的 CPU 限制。
central.persistence.hostPath	RHACS 应该创建数据库卷的节点上的路径。红帽不推荐使用这个选项。

参数	Description
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** 创建新 PVC 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 RHACS，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。
central.db.external	使用 **true** 指定不应部署中央 DB，并且将使用外部数据库。
central.db.source.connectionString	用于连接到数据库的 Central 的连接字符串。这仅在将 **central.db.external** 设置为 true 时使用。连接字符串必须采用 keyword/value 格式，如 PostgreSQL 文档中的 "Additional resources" 所述。 ● 仅支持 PostgreSQL 13。 ● 不支持通过 PgBouncer 连接。 ● 用户必须是超级用户，能够创建和删除数据库。
central.db.source.minConns	与要建立的数据库的最小连接数。
central.db.source.maxConns	与要建立的数据库的连接数上限。
central.db.source.statementTimeoutMs	单个查询或事务的毫秒可以针对数据库处于活跃状态。
central.db.postgresConfig	用于中央 DB 的 postgresql.conf，如 PostgreSQL 文档中的"添加资源"中所述。

参数	Description
central.db.hbaConfig	用于 Central DB 的 pg_hba.conf，如 PostgreSQL 文档中的 "Additional resources" 所述。
central.db.nodeSelector	将节点选择器标签指定为 **label-key: label-value**，以强制 Central DB 仅调度到具有指定标签的节点。
central.db.image.registry	一个自定义 registry，用于覆盖 Central DB 镜像的全局 **image.registry** 参数。
central.db.image.name	覆盖默认中央 DB 镜像名称(**central-db**)的自定义镜像名称。
central.db.image.tag	覆盖 Central DB 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。如果您在自己的 registry 中镜像 Central DB 镜像，请不要修改原始镜像标签。
central.db.image.fullRef	Central DB 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.db.image.registry**、**central.db.image.name** 和 **central.db.image.tag** 参数。
central.db.resources.requests.memory	Central DB 的内存请求。
central.db.resources.requests.cpu	Central DB 的 CPU 请求。
central.db.resources.limits.memory	Central DB 的内存限值。
central.db.resources.limits.cpu	Central DB 的 CPU 限制。
central.db.persistence.hostPath	RHACS 应该创建数据库卷的节点上的路径。红帽不推荐使用这个选项。
central.db.persistence.persistentVolumeClaim.claimName	您要使用的持久性卷声明(PVC)的名称。
central.db.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.db.persistence.persistentVolumeClaim.size	由指定声明管理的持久性卷的大小（以 GiB 为单位）。

4.1.2.2.2.6. stackrox Scanner

下表列出了 StackRox Scanner 的可配置参数。这是用于节点和平台扫描的扫描程序。如果没有启用

下表列出了 StackRox Scanner 的可配置参数。这是用于节点和平台扫描的扫描程序。如果没有启用 Scanner V4，StackRox 扫描程序也会执行镜像扫描。从版本 4.4 开始，可以启用 Scanner V4 以提供镜像扫描。请参阅 Scanner V4 参数的下一表。

参数	Description
scanner.disable	使用 **true** 在没有 StackRox 扫描器的情况下安装 RHACS。当将其与 **helm upgrade** 命令一起使用时，Helm 会移除现有的 StackRox Scanner 部署。
scanner.exposeMonitoring	指定 **true**，以在端口号 **9090** 上为 StackRox Scanner 公开 Prometheus 指标端点。
scanner.replicas	为 StackRox 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	配置 StackRox Scanner 的日志级别。红帽建议不要更改默认日志级别值(**INFO**)。
scanner.nodeSelector	将节点选择器标签指定为 **label-key: label-value**，以强制 StackRox 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 StackRox 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 StackRox 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	StackRox Scanner 的内存请求。
scanner.resources.requests.cpu	StackRox Scanner 的 CPU 请求。
scanner.resources.limits.memory	StackRox Scanner 的内存限值。
scanner.resources.limits.cpu	StackRox 扫描器的 CPU 限制。
scanner.dbResources.requests.memory	StackRox Scanner 数据库部署的内存请求。
scanner.dbResources.requests.cpu	StackRox Scanner 数据库部署的 CPU 请求。
scanner.dbResources.limits.memory	StackRox Scanner 数据库部署的内存限值。

参数	Description
scanner.dbResources.limits.cpu	StackRox Scanner 数据库部署的 CPU 限制。
scanner.image.registry	StackRox Scanner 镜像的自定义 registry。
scanner.image.name	覆盖默认 StackRox Scanner 镜像名称(**扫描程序**)的自定义镜像名称。
scanner.dbImage.registry	StackRox Scanner DB 镜像的自定义 registry。
scanner.dbImage.name	覆盖默认 StackRox 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 StackRox 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 StackRox 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。

4.1.2.2.2.7. scanner V4

下表列出了 Scanner V4 的可配置参数。

重要

扫描程序 V4 只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅技术预览功能支持范围。

参数	Description
scannerV4.db.persistence.persistentVolumeClaim.claimName	用于管理 Scanner V4 持久数据的 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 **scanner-v4-db**。为防止数据丢失，当 Central 被删除时，PVC 不会被自动删除。
scannerV4.disable	使用 **false** 启用 Scanner V4。当设置此参数时，还必须通过设置 **scanner.disable=false** 来启用 StackRox Scanner。在达到 StackRox Scanner 和 Scanner V4 之间的功能奇偶校验前，Scanner V4 只能与 StackRox Scanner 结合使用。不支持在没有启用 StackRox 扫描器的情况下启用 Scanner V4。当使用 **helm upgrade** 命令将此参数设置为 **true** 时，Helm 会移除现有的 Scanner V4 部署。

参数	Description
scannerV4.exposeMonitoring	指定 **true**，以在端口号 **9090** 上为 Scanner V4 公开 Prometheus 指标端点。
scannerV4.indexer.replicas	为 Scanner V4 Indexer 部署创建的副本数。当您将 **scannerV4.indexer.autoscaling** 参数一起使用时，这个值会设置初始副本数。
scannerV4.indexer.logLevel	配置 Scanner V4 Indexer 的日志级别。红帽建议不要更改默认日志级别值(**INFO**)。
scannerV4.indexer.nodeSelector	将节点选择器标签指定为 **label-key: label-value**，以强制 Scanner V4 Indexer 仅调度到具有指定标签的节点。
scannerV4.indexer.tolerations	如果节点选择器选择污点节点，请使用此参数为 Scanner V4 Indexer 指定污点容限键、值和效果。此参数主要用于基础架构节点。
scannerV4.indexer.autoscaling.disable	使用 **true** 为 Scanner V4 Indexer 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。
scannerV4.indexer.autoscaling.minReplicas	自动扩展的最小副本数。
scannerV4.indexer.autoscaling.maxReplicas	自动扩展的最大副本数。
scannerV4.indexer.resources.requests.memory	Scanner V4 Indexer 的内存请求。
scannerV4.indexer.resources.requests.cpu	Scanner V4 Indexer 的 CPU 请求。
scannerV4.indexer.resources.limits.memory	Scanner V4 Indexer 的内存限值。
scannerV4.indexer.resources.limits.cpu	Scanner V4 Indexer 的 CPU 限制。
scannerV4.matcher.replicas	为 Scanner V4 Matcher 部署创建的副本数。当您将 **scannerV4.matcher.autoscaling** 参数一起使用时，这个值会设置初始副本数。
scannerV4.matcher.logLevel	红帽建议不要更改默认日志级别值(**INFO**)。
scannerV4.matcher.nodeSelector	将节点选择器标签指定为 **label-key: label-value**，以强制 Scanner V4 Matcher 仅调度到具有指定标签的节点。

参数	Description
scannerV4.matcher.tolerations	如果节点选择器选择污点节点，请使用此参数为 Scanner V4 Matcher 指定污点容限键、值和效果。此参数主要用于基础架构节点。
scannerV4.matcher.autoscaling.disable	使用 **true** 为 Scanner V4 Matcher 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。
scannerV4.matcher.autoscaling.minReplicas	自动扩展的最小副本数。
scannerV4.matcher.autoscaling.maxReplicas	自动扩展的最大副本数。
scannerV4.matcher.resources.requests.memory	Scanner V4 Matcher 的内存请求。
scannerV4.matcher.resources.requests.cpu	Scanner V4 Matcher 的 CPU 请求。
scannerV4.db.resources.requests.memory	Scanner V4 数据库部署的内存请求。
scannerV4.db.resources.requests.cpu	Scanner V4 数据库部署的 CPU 请求。
scannerV4.db.resources.limits.memory	Scanner V4 数据库部署的内存限值。
scannerV4.db.resources.limits.cpu	Scanner V4 数据库部署的 CPU 限制。
scannerV4.db.nodeSelector	将节点选择器标签指定为 **label-key: label-value**，以强制 Scanner V4 DB 仅调度到具有指定标签的节点。
scannerV4.db.tolerations	如果节点选择器选择污点节点，请使用此参数为 Scanner V4 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。
scannerV4.db.image.registry	Scanner V4 DB 镜像的自定义 registry。
scannerV4.db.image.name	覆盖默认 Scanner V4 DB 镜像名称(**scanner-v4-db**)的自定义镜像名称。
scannerV4.image.registry	Scanner V4 镜像的自定义 registry。
scannerV4.image.name	覆盖默认 Scanner V4 镜像名称(**scanner-v4)的自定义镜像名称**。

4.1.2.2.2.8. 自定义

使用这些参数为 RHACS 创建的所有对象指定附加属性。

参数	Description
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。
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。
customize.scanner-v4-indexer.labels	附加到 Scanner V4 Indexer 创建的所有对象的自定义标签，并附加到属于它们的 pod。

参数	Description
customize.scanner-v4-indexer.annotations	附加到 Scanner V4 Indexer 创建的所有对象的自定义注解，并附加到属于它们的 pod。
customize.scanner-v4-indexer.podLabels	附加到 Scanner V4 Indexer 创建的所有对象的自定义标签，并附加到属于它们的 pod。
customize.scanner-v4-indexer.podAnnotations	附加到 Scanner V4 Indexer 创建的所有对象的自定义注解，并附加到属于它们的 pod。
customize.scanner-4v-indexer.envVars	所有 Scanner V4 Indexer 容器及其属于它们的 pod 的自定义环境变量。
customize.scanner-v4-matcher.labels	附加到 Scanner V4 Matcher 创建的所有对象的自定义标签，并放入它们所属的 pod。
customize.scanner-v4-matcher.annotations	附加到 Scanner V4 Matcher 创建的所有对象的自定义注解，并附加到它们所属的 pod。
customize.scanner-v4-matcher.podLabels	附加到 Scanner V4 Matcher 创建的所有对象的自定义标签，并放入它们所属的 pod。
customize.scanner-v4-matcher.podAnnotations	附加到 Scanner V4 Matcher 创建的所有对象的自定义注解，并附加到它们所属的 pod。
customize.scanner-4v-matcher.envVars	所有 Scanner V4 Matcher 容器及其属于它们的 pod 的自定义环境变量。
customize.scanner-v4-db.labels	附加到 Scanner V4 DB 创建的所有对象的自定义标签，并附加到它们所属的 pod。
customize.scanner-v4-db.annotations	附加到 Scanner V4 DB 创建的所有对象的自定义注解，并附加到它们所属的 pod。
customize.scanner-v4-db.podLabels	附加到 Scanner V4 DB 创建的所有对象的自定义标签，并附加到它们所属的 pod。
customize.scanner-v4-db.podAnnotations	附加到 Scanner V4 DB 创建的所有对象的自定义注解，并附加到它们所属的 pod。
customize.scanner-4v-db.envVars	所有 Scanner V4 DB 容器及其属于它们的 pod 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。

4.1.2.2.2.9. 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命名空间和发行版本名称的 RHACS 实例。

参数	Description
allowNonstandardNamespace	使用 **true** 将 RHACS 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** 以外的发行版本名称部署 RHACS。

4.1.2.2.3. 声明性配置值

要使用声明性配置，您必须创建一个 YAML 文件（在这个示例中，名为 "declarative-config-values.yaml"），以将声明性配置挂载添加到 Central。此文件用于 Helm 安装。

流程

1. 使用以下示例创建 YAML 文件（本例中为 **declarative-config-values.yaml**）：

```
central:
  declarativeConfiguration:
    mounts:
      configMaps:
        - declarative-configs
      secrets:
        - sensitive-declarative-configs
```

2. 安装 Central 服务 Helm chart，如"安装 central-services Helm Chart"中所述，引用 **declarative-config-values.yaml** 文件。

4.1.2.2.4. 安装 central-services Helm chart

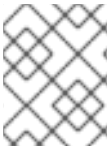
配置 **values-public.yaml** 和 **values-private.yaml** 文件后，安装 **central-services** Helm Chart 来部署集中式组件（Central 和 Scanner）。

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。



注意

可选：如果使用声明性配置，请将 **-f <path_to_declarative-config-values.yaml** 添加到此命令，以便在 Central 中挂载声明性配置文件。

4.1.2.3. 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。

当使用 **helm upgrade** 命令进行修改时，会应用以下准则和要求：

- 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，每当您进行更改时，您必须手动指定所有选项。
- 有些更改（如启用 Scanner V4）需要为组件发布新证书。因此，您必须在进行这些更改时提供 CA。

重要

扫描程序 V4 只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅技术预览功能支持范围。

  - 如果 CA 在初始安装过程中由 Helm chart 生成，则必须从集群中检索这些值，并将其提供给 **helm upgrade** 命令。**central-services** Helm Chart 的安装后备注包括用于检索自动生成的值的命令。
  - 如果 CA 在 Helm Chart 之外生成，并在安装 **central-services** chart 时提供，那么您必须在使用 **helm upgrade** 命令时再次执行该操作，例如在 **helm upgrade** 命令中使用 **--reuse-values** 标志。

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 \
  --reuse-values \ 1
  -f <path_to_init_bundle_file \
  -f <path_to_values_public.yaml> \
  -f <path_to_values_private.yaml>
```

1 如果您修改了没有包括在 **values_public.yaml** 和 **values_private.yaml** 文件中的值，请包含 **--reuse-values** 参数。

4.1.3. 使用 roxctl CLI 安装 Central

警告

对于生产环境，红帽建议使用 Operator 或 Helm chart 来安装 RHACS。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。

4.1.3.1. 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**。

4.1.3.1.1. 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。

注意

用于 Linux 的 **roxctl** CLI 可用于 **amd64**、 **ppc64le** 和 **s390x** 架构。

流程

1. 确定目标操作系统的 **roxctl** 架构：

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

2. 下载 **roxctl** CLI：

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

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
```

4.1.3.1.2. 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。

注意

用于 macOS 的 **roxctl** CLI 可用于 **amd64** 架构。

流程

1. 下载 **roxctl** CLI：

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

4.1.3.1.3. 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。

注意

amd64 架构提供了适用于 Windows 的 **roxctl** CLI。

流程

- 下载 **roxctl** CLI：

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

验证

- 验证您已安装的 **roxctl** 版本：

```
$ roxctl version
```

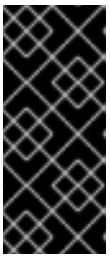
4.1.3.2. 使用交互式安装程序

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
```



重要

使用 **roxctl** CLI 安装 RHACS 会创建 PodSecurityPolicy (PSP)对象，以便向后兼容。如果要在 Kubernetes 版本 1.25 及更新版本上，或在 OpenShift Container Platform version 4.12 和更新版本上安装 RHACS，则必须禁用 PSP 对象的创建。要做到这一点，对于 **roxctl central generate** 和 **roxctl sensor generate** 命令，将 **--enable-pod-security-policies** 选项设置为 **false**。

2. 按 **Enter** 接受提示的默认值或根据需要输入自定义值。以下示例显示了交互式安装程序提示：

```
Enter path to the backup bundle from which to restore keys and certificates (optional):
Enter read templates from local filesystem (default: "false"):
Enter path to helm templates on your local filesystem (default: "/path"):
Enter PEM cert bundle file (optional): 1
Enter Create PodSecurityPolicy resources (for pre-v1.25 Kubernetes) (default: "true"): 2
Enter administrator password (default: autogenerated):
Enter orchestrator (k8s, openshift):
Enter default container images settings (development_build, stackrox.io, rhacs, opensource);
it controls repositories from where to download the images, image names and tags format
(default: "development_build"):
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"):
Enter whether to enable telemetry (default: "false"):
Enter central-db image to use (if unset, a default will be used according to --image-defaults):
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"): 3
Enter main image to use (if unset, a default will be used according to --image-defaults):
Enter whether to run StackRox in offline mode, which avoids reaching out to the Internet
(default: "false"):
Enter list of secrets to add as declarative configuration mounts in central (default: "[]"): 4
Enter list of config maps to add as declarative configuration mounts in central (default: "[]"):
5
Enter the deployment tool to use (kubectl, helm, helm-values) (default: "kubectl"):
Enter scanner-db image to use (if unset, a default will be used according to --image-defaults):
Enter scanner image to use (if unset, a default will be used according to --image-defaults):
Enter Central volume type (hostpath, pvc): 6
Enter external volume name for Central (default: "stackrox-db"):
Enter external volume size in Gi for Central (default: "100"):
Enter storage class name for Central (optional if you have a default StorageClass
configured):
Enter external volume name for Central DB (default: "central-db"):
Enter external volume size in Gi for Central DB (default: "100"):
Enter storage class name for Central DB (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. 如果您正在运行 Kubernetes 版本 1.25 或更高版本，请将此值设置为 **false**。
3. 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。
4. 有关使用声明配置进行身份验证和授权的更多信息，请参阅 "Red Hat Advanced Cluster Security for Kubernetes 中的"管理 RBAC"中的为身份验证和授权资源提供配置。
5. 有关使用声明配置进行身份验证和授权的更多信息，请参阅 "Red Hat Advanced Cluster Security for Kubernetes 中的"管理 RBAC"中的为身份验证和授权资源提供配置。
6. 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，而是在 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）。

4.1.3.3. 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	Example
Route（路由）	**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**
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
```

4.2. 使用 OPERATOR 为 RHACS 配置 CENTRAL 配置选项

当使用 Operator 安装 Central 实例时，您可以配置可选设置。

4.2.1. 使用 Operator 的中央配置选项

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。

下表包含外部 PostgreSQL 数据库的设置。

4.2.1.1. Central 设置

参数	Description
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。

参数	Description
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 Red Hat OpenShift 路由公开 Central。默认值为 **false**。
central.exposure.route.host	指定用于 Central 路由的自定义主机名。保留为不设置，以接受 OpenShift Container Platform 提供的默认值。
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.monitoring.exposeEndpoint	使用 **Enabled** 为 Central 启用监控。当您启用监控时，RHACS 会在端口号 **9090** 上创建新的监控服务。默认值为 **Disabled**。
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，当 Central 被删除时，PVC 不会被自动删除。
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	Description
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。
central.db.passwordSecret.name	在 **password** 数据项中指定一个具有数据库密码的 secret。只有在您要手动指定连接字符串时，才使用此参数。如果省略，Operator 会自动生成密码，并将其存储在 **central-db-password** secret 的 **password** 项中。
central.db.connectionString	设置此参数将不会部署 Central DB，并且 Central 将使用指定的连接字符串进行连接。如果为此参数指定值，还必须为 **central.db.passwordSecret.name** 指定一个值。这个参数有以下限制： ● 连接字符串必须采用关键字/值格式，如 PostgreSQL 文档中所述。如需更多信息，请参阅**额外资源**部分中的链接。 ● 仅支持 PostgreSQL 13。 ● 不支持通过 PGBouncer 连接。 ● 用户必须是能够创建和删除数据库的超级用户。
central.db.tolerations	如果节点选择器选择污点节点，请使用此参数为 Central 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。
central.db.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.db.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 **central-db**。为防止数据丢失，当 Central 被删除时，PVC 不会被自动删除。
central.db.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。
central.db.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。

参数	Description
central.db.resources.limits	使用此参数覆盖 Central DB 的默认资源限值。
central.db.resources.requests	使用此参数覆盖 Central DB 的默认资源请求。

4.2.1.2. stackrox Scanner 设置

参数	Description
scanner.analyzer.nodeSelector	如果您希望此扫描程序只在特定节点上运行，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 StackRox 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 StackRox Scanner 的默认资源限值。
scanner.analyzer.resources.requests	使用此参数覆盖 StackRox Scanner 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	指定分析器自动扩展配置中使用的最大副本
scanner.analyzer.scaling.minReplicas	指定分析器自动扩展配置中使用的最小副本
scanner.analyzer.scaling.replicas	禁用自动扩展时，始终将副本数配置为与这个值匹配。
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。
scanner.db.tolerations	如果节点选择器选择污点节点，请使用此参数为 StackRox 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 StackRox Scanner DB 的默认资源限值。
scanner.db.resources.requests	使用此参数覆盖 StackRox Scanner DB 的默认资源请求。

参数	Description
scanner.monitoring.exposeEndpoint	使用 **Enabled** 为 StackRox Scanner 启用监控。当您启用监控时，RHACS 会在端口号 **9090** 上创建新的监控服务。默认值为 **Disabled**。
scanner.scannerComponent	如果您不想部署 StackRox Scanner，您可以使用此参数禁用它。如果您禁用 StackRox Scanner，本节中的所有其他设置都无效。红帽不推荐为 Kubernetes 禁用 Red Hat Advanced Cluster Security for Kubernetes 的 StackRox 扫描器。如果您启用了 Scanner V4，请不要禁用 StackRox Scanner。扫描程序 V4 要求 StackRox 扫描程序也被启用，以提供必要的扫描功能。

4.2.1.3. scanner V4 设置（技术预览）

重要

扫描程序 V4 只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅技术预览功能支持范围。

参数	Description
scannerV4.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。
scannerV4.db.tolerations	如果节点选择器选择污点节点，请使用此参数为 Scanner V4 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。
scannerV4.db.resources.limits	使用此参数覆盖 Scanner V4 DB 的默认资源限值。
scannerV4.db.resources.requests	使用此参数覆盖 Scanner V4 DB 的默认资源请求。
scannerV4.db.persistence.persistentVolumeClaim.claimName	用于管理 Scanner V4 持久数据的 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 **scanner-v4-db**。为防止数据丢失，当 Central 被删除时，PVC 不会被自动删除。
scannerV4.indexer.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。
scannerV4.indexer.tolerations	如果节点选择器选择污点节点，请使用此参数为 Scanner V4 Indexer 指定污点容限键、值和效果。此参数主要用于基础架构节点。
scannerV4.indexer.resources.limits	使用此参数覆盖 Scanner V4 Indexer 的默认资源限值。

参数	Description
scannerV4.indexer.resources.requests	使用此参数覆盖 Scanner V4 Indexer 的默认资源请求。
scannerV4.indexer.scaling.autoScaling	启用后，Scanner V4 Indexer 副本的数量会根据指定的限值动态管理。
scannerV4.indexer.scaling.maxReplicas	指定 Scanner V4 Indexer 自动扩展配置中使用的最大副本。
scannerV4.indexer.scaling.minReplicas	指定 Scanner V4 Indexer 自动扩展配置中使用的最小副本。
scannerV4.indexer.scaling.replicas	当 Scanner V4 Indexer 禁用了自动扩展时，副本数始终配置为与这个值匹配。
scannerV4.matcher.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。
scannerV4.matcher.tolerations	如果节点选择器选择污点节点，请使用此参数为 Scanner V4 Matcher 指定污点容限键、值和效果。此参数主要用于基础架构节点。
scannerV4.matcher.resources.limits	使用此参数覆盖 Scanner V4 Matcher 的默认资源限值。
scannerV4.matcher.resources.requests	使用此参数覆盖 Scanner V4 Matcher 的默认资源请求。
scannerV4.matcher.scaling.autoScaling	启用后，Scanner V4 Matcher 副本的数量会根据指定的限值动态管理。
scannerV4.matcher.scaling.maxReplicas	指定 Scanner V4 Matcher 自动扩展配置中使用的最大副本。
scannerV4.matcher.scaling.minReplicas	指定 Scanner V4 Matcher 自动扩展配置中使用的最小副本。
scannerV4.matcher.scaling.replicas	当 Scanner V4 Matcher 禁用自动扩展时，副本数始终配置为与这个值匹配。
scannerV4.monitoring.exposeEndpoint	为 Scanner V4 配置监控端点。监控端点允许其他服务从与 Prometheus 兼容的格式提供的 Scanner V4 中收集指标。使用 **Enabled** 来公开监控端点。当您启用监控时，RHACS 会创建一个新的服务，使用端口 9090 **监控**，以及允许到端口入站连接的网络策略。默认情况下，这没有启用。
scannerV4.scannerComponent	启用 Scanner V4。默认值为 **默认值**，它被禁用。要启用 Scanner V4，将此参数设置为 **Enabled**。

4.2.1.4. 常规设置和各种设置

参数	Description
tls.additionalCAs	要信任的安全集群的其他可信 CA 证书。这些证书通常用于使用私有证书颁发机构与服务集成。
misc.createSCCs	指定 **true** 为 Central 创建 **SecurityContextConstraints** (SCC)。设置为 **true** 可能会导致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 RHACS 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。
monitoring.openshift.enabled	如果将此选项设置为 **false**，Red Hat Advanced Cluster Security for Kubernetes 将不会设置 Red Hat OpenShift 监控。在 Red Hat OpenShift 4 上默认为 **true**。
overlays	请参阅使用带有覆盖的 Operator 自定义安装

4.2.2. 使用带有覆盖的 Operator 自定义安装

了解如何通过 overlays 使用 Operator 方法定制 RHACS 安装。

4.2.2.1. overlays

当 **Central** 或 **SecuredCluster** 自定义资源没有以参数的形式公开某些低级别配置选项时，您可以使用 **.spec.overlays** 字段进行调整。使用此字段来修改这些自定义资源生成的 Kubernetes 资源。

.spec.overlays 字段由一系列补丁组成，按其列出的顺序应用。这些补丁由 Kubernetes 资源上的 Operator 在部署到集群前由 Kubernetes 资源处理。

警告

Central 和 **SecuredCluster** 中的 **.spec.overlays** 字段允许用户以任意方式修改低级别 Kubernetes 资源。只有在所需的自定义无法通过 **SecuredCluster** 或 **Central** 自定义资源提供时，才使用此功能。

对 **.spec.overlays** 功能的支持主要是有限的，因为它授予了对 Kubernetes 资源进行 intricate 和高度具体的修改的功能，这可能因一个实施而异。这种自定义级别引入了一个超过标准使用场景的复杂性，这使其难以提供广泛的支持。每个修改都是唯一的，可能在产品的不同版本和配置中以无法预计的方式与 Kubernetes 系统交互。这种差异意味着排除并保证这些自定义的稳定性需要一定程度的专业知识和理解。因此，虽然此功能支持定制 Kubernetes 资源来满足精确需求，但还必须考虑确保配置的兼容性和稳定性，特别是在升级或更改底层产品期间。

以下示例显示了覆盖的结构：

```
overlays:
- apiVersion: v1 1
  kind: ConfigMap 2
  name: my-configmap 3
  patches:
    - path: .data 4
      value: | 5
        key1: data2
        key2: data2
```

1. targeted Kubernetes resource ApiVersion，如 **apps/v1**,**v1**,**networking.k8s.io/v1**
2. 资源类型（如 Deployment、ConfigMap、NetworkPolicy）
3. 资源的名称，如 **my-configmap**
4. 字段的 jsonpath 表达式，如 **spec.template.spec.containers[name:central].env[-1]**
5. 新字段值的 YAML 字符串

4.2.2.1.1. 添加覆盖

对于自定义，您可以在 **Central** 或 **SecuredCluster** 自定义资源中添加覆盖。使用 OpenShift CLI (**oc**)或 OpenShift Container Platform Web 控制台进行修改。

如果覆盖没有按预期生效，请检查 RHACS Operator 日志是否有语法错误或记录的问题。

4.2.2.2. 覆盖示例

4.2.2.2.1. 为 Central ServiceAccount 指定 EKS pod 角色 ARN

在 **中央** ServiceAccount 中添加 Amazon Elastic Kubernetes Service (EKS) pod 角色 Amazon Resource Name (ARN)注解，如下例所示：

```
apiVersion: platform.stackrox.io
kind: Central
metadata:
  name: central
spec:
  # ...
  overlays:
  - apiVersion: v1
    kind: ServiceAccount
    name: central
    patches:
      - path: metadata.annotations.eks\.amazonaws\.com/role-arn
        value: "\"arn:aws:iam:1234:role\""
```

4.2.2.2.2. 将环境变量注入中央部署

将环境变量注入到 **中央** 部署中，如下例所示：

```
apiVersion: platform.stackrox.io
kind: Central
metadata:
  name: central
spec:
  # ...
  overlays:
  - apiVersion: apps/v1
    kind: Deployment
    name: central
    patches:
    - path: spec.template.spec.containers[name:central].env[-1]
      value: |
        name: MY_ENV_VAR
        value: value
```

4.2.2.2.3. 使用入口规则扩展网络策略

在 **allow-ext-to-central** 网络策略中添加一个入口规则，用于端口 999 流量，如下例所示：

```
apiVersion: platform.stackrox.io
kind: Central
metadata:
  name: central
spec:
    # ...
    overlays:
    - apiVersion: networking.k8s.io/v1
      kind: NetworkPolicy
      name: allow-ext-to-central
      patches:
        - path: spec.ingress[-1]
          value: |
            ports:
            - port: 999
              protocol: TCP
```

4.2.2.2.4. 修改 ConfigMap 数据

修改 **central-endpoints** ConfigMap 数据，如下例所示：

```
apiVersion: platform.stackrox.io
kind: Central
metadata:
  name: central
spec:
    # ...
    overlays:
    - apiVersion: v1
      kind: ConfigMap
      name: central-endpoints
      patches:
      - path: data
        value: |
          endpoints.yaml: |
            disableDefault: false
```

4.2.2.2.5. 将容器添加到 中央 部署中

在 **中央** 部署中添加新容器，如下例所示：

```
apiVersion: platform.stackrox.io
kind: Central
metadata:
  name: central
spec:
    # ...
    overlays:
    - apiVersion: apps/v1
      kind: Deployment
      name: central
      patches:
        - path: spec.template.spec.containers[-1]
      value: |
        name: nginx
        image: nginx
        ports:
          - containerPort: 8000
            name: http
            protocol: TCP
```

其他资源

- 连接字符串 - PostgreSQL 文档
- 通过配置文件进行参数交互 - PostgreSQL 文档
- pg_hba.conf 文件 - PostgreSQL 文档

4.3. 为 RED HAT OPENSHIFT 上的 RHACS 生成并应用 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 门户或 **roxctl** CLI 创建 init 捆绑包。然后，您可以使用它应用 init 捆绑包来创建资源。

要为 RHACS 云服务配置 init 捆绑包，请参阅以下资源：

- 为安全集群(Red Hat Cloud)生成 init 捆绑包
- 为安全集群(Red Hat Cloud)应用 init 捆绑包
- 为 Kubernetes 安全集群生成 init 捆绑包
- 为 Kubernetes 安全集群应用 init 捆绑包

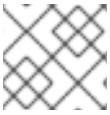
注意

您必须具有 **Admin** 用户角色才能创建 init 捆绑包。

4.3.1. 生成 init 捆绑包

4.3.1.1. 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。

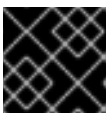


注意

您必须具有 **Admin** 用户角色才能创建 init 捆绑包。

流程

1. 如"使用 Operator 方法验证中央安装"中所述，查找 RHACS 门户的地址。
2. 登录到 RHACS 门户。
3. 如果您没有安全集群，则会出现 **Platform Configuration → Clusters** 页面。
4. 点 **Create init bundle**。
5. 为集群 init 捆绑包输入一个名称。
6. 选择您的平台。
7. 选择您要用于安全集群的安装方法： **Operator** 或 **Helm Chart**。
8. 点 **Download** 生成并下载以 YAML 文件形式创建的 init 捆绑包。如果您使用相同的安装方法，您可以对所有安全集群使用一个 init 捆绑包及其对应的 YAML 文件。



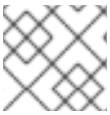
重要

安全地存储此捆绑包，因为它包含 secret。

9. 通过使用它来在安全集群中创建资源来应用 init 捆绑包。
10. 在每个集群中安装安全的集群服务。

4.3.1.2. 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。



注意

您必须具有 **Admin** 用户角色才能创建 init 捆绑包。

先决条件

- 您已配置了 **ROX_API_TOKEN** 和 **ROX_CENTRAL_ADDRESS** 环境变量：
  a. 运行以下命令设置 **ROX_API_TOKEN** ：

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

  b. 运行以下命令设置 **ROX_CENTRAL_ADDRESS** 环境变量：

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

流程

- 要生成包含 Helm 安装 secret 的集群 init 捆绑包，请运行以下命令：

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

- 要生成包含 Operator 安装 secret 的集群 init 捆绑包，请运行以下命令：

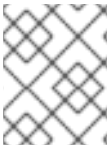
```
$ roxctl -e "$ROX_CENTRAL_ADDRESS" \
  central init-bundles generate <cluster_init_bundle_name> \
  --output-secrets cluster_init_bundle.yaml
```

重要

确保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。

4.3.1.3. 在安全集群中应用 init 捆绑包

在配置安全集群前，您必须使用它来应用 init 捆绑包来在集群中创建所需资源。应用 init 捆绑包可让安全集群中的服务与 Central 通信。



注意

如果使用 Helm chart 安装，请不要执行此步骤。使用 Helm 完成安装 ; 请参阅"使用 Helm chart 在安全集群中安装 RHACS"。

先决条件

- 您必须生成了一个包含 secret 的 init 捆绑包。

- 您必须在安装安全集群服务的集群中创建了 **stackrox** 项目或命名空间。不需要将 **stackrox** 用于项目，而是确保在扫描集群时不会报告 RHACS 进程的漏洞。

流程

要创建资源，请执行以下步骤之一：

- 使用 OpenShift Container Platform Web 控制台创建资源：在 OpenShift Container Platform Web 控制台中，确保您位于 **stackrox** 命名空间中。在顶部菜单中，点 **+** 打开 **Import YAML** 页面。您可以拖动 init 捆绑包文件或将其内容复制并粘贴到编辑器中，然后点 **Create**。命令完成后，显示显示 **collector-tls**、**sensor-tls** 和 admission-control-tls 的资源已创建。
- 使用 Red Hat OpenShift CLI 创建资源：使用 Red Hat OpenShift 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 Central 服务的项目的名称。

4.3.2. 后续步骤

- 在您要监控的所有集群中安装 RHACS 安全集群服务。

4.3.3. 其他资源

- 使用 Helm chart 在安全集群中安装 RHACS

4.4. 在 RED HAT OPENSHIFT 中为 RHACS 安装安全集群服务

您可以使用以下方法之一在安全集群中安装 RHACS：

- 使用 Operator 安装
- 使用 Helm chart 安装
- 使用 **roxctl** CLI 安装（除非有需要使用它的特定安装需要）

4.4.1. 使用 Operator 在安全集群中安装 RHACS

4.4.1.1. 安装安全的集群服务

您可以使用 Operator 在集群中安装 Secured Cluster 服务，这将创建 **SecuredCluster** 自定义资源。您必须在要监控的环境中的每个集群中安装 Secured Cluster 服务。

先决条件

- 如果使用 OpenShift Container Platform，您必须安装版本 4.11 或更高版本。
- 您已在要保护的集群中安装了 RHACS Operator，称为安全集群。

- 您已生成 init 捆绑包并将其应用到集群。

流程

1. 在安全集群的 OpenShift Container Platform Web 控制台中，进入 **Operators → Installed Operators** 页面。
2. 点 RHACS Operator。
3. 从 **Operator 详情**页面的中央导航菜单中点 **Secured Cluster**。
4. 点 **Create SecuredCluster**。
5. 在 **Configure via** 字段中选择以下选项之一：
   - **表单视图** ：如果要使用屏幕字段配置安全集群且不需要更改任何其他字段，则使用这个选项。
   - **YAML 视图** ：使用此视图使用 YAML 文件设置安全集群。YAML 文件显示在窗口中，您可以在其中编辑字段。如果您选择这个选项，请在完成编辑完该文件时，点 **Create**。
6. 如果使用 **Form view**，请通过接受或编辑默认名称来输入新项目名称。默认值为 **stackrox-secured-cluster-services**。
7. 可选：为集群添加任何标签。
8. 输入您的 **SecuredCluster** 自定义资源的唯一名称。
9. 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**。
   - *只有在安装了* Central 的同一集群中安装安全集群服务时，才使用 **central.stackrox.svc:443** 的默认值。
   - 在配置多个集群时，不要使用默认值。反之，在为每个集群配置 **Central Endpoint** 值时使用主机名。
10. 对于剩余的字段，接受默认值，或者根据需要配置自定义值。例如，如果您使用自定义证书或不受信任的 CA，您可能需要配置 TLS。如需更多信息，请参阅"使用 Operator 为 RHACS 配置安全集群服务选项"。
11. 点 **Create**。
12. 在短暂暂停后，**Secured** Clusters 页面会显示 **stackrox-secured-cluster-services** 的状态。您可能会看到以下条件：
    - **conditions: Deployed, Initialized** 已安装安全集群服务，安全集群与 Central 通信。
    - **conditions: Initialized, Irreconcilable**: 安全集群没有与 Central 通信。确保将您在 RHACS web 门户中创建的 init 捆绑包应用到安全集群。

后续步骤

1. 配置额外的安全集群设置（可选）。
2. 验证安装。

4.4.2. 使用 Helm chart 在安全集群中安装 RHACS

您可以使用没有自定义的 Helm chart、使用默认值或配置参数自定义的 Helm chart 在安全集群中安装 RHACS。

4.4.2.1. 使用 Helm chart 在安全集群中安装 RHACS

4.4.2.1.1. 添加 Helm Chart 仓库

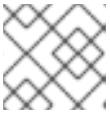
流程

- 添加 RHACS chart 存储库。

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

Red Hat Advanced Cluster Security for Kubernetes 的 Helm 仓库包括用于安装不同组件的 Helm chart，包括：

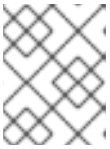
- 用于安装集中组件（Central 和 Scanner）的中央服务Helm Chart（**central-services**）。



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 per-cluster 和 per-node 组件 (Sensor、Admission Controller、Collector 和 Scanner-slim)。



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/
```

4.4.2.1.2. 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、Collector 和 Scanner-slim)。

先决条件

- 您必须已为集群生成 RHACS init 捆绑包。
- 您必须有权访问 Red Hat Container Registry 和一个 pull secret 进行身份验证。有关从 **registry.redhat.io** 下载镜像的详情，请参考 Red Hat Container Registry Authentication 。
- 您必须有用于公开 Central 服务的地址和端口号。

流程

- 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
  -f <path_to_pull_secret.yaml> \ 2
  --set clusterName=<name_of_the_secured_cluster> \
  --set centralEndpoint=<endpoint_of_central_service> 3
  --set scanner.disable=false 4
```

1. 使用 **-f** 选项指定 init 捆绑包的路径。
2. 使用 -f 选项指定 Red Hat Container Registry 身份验证的 pull secret 的路径。
3. 指定 Central 的地址和端口号。例如，**acs.domain.com:443**。
4. 将 **scanner.disable** 参数的值设置为 **false**，这意味着在安装过程中将启用 Scanner-slim。在 Kubernetes 中，安全集群服务现在包括 Scanner-slim 作为可选组件。

其他资源

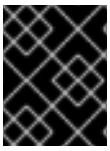
- 为 Red Hat OpenShift 上的 RHACS 生成并应用 init 捆绑包

4.4.2.2. 使用自定义配置 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。确保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。

4.4.2.2.1. 配置参数

参数	Description
clusterName	集群的名称。
centralEndpoint	Central 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。在配置多个集群时，使用地址的主机名（如 **central.example.com:443**）。

参数	Description
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。
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	**CORE_BPF**、**EBPF**（已弃用）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。

参数	Description
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。
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，使用 **AdmissionReview** 请求进行工作负载创建事件。
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 Kubernetes**exec** 和 **portforward** 事件。RHACS 不支持 OpenShift Container Platform 3.11 的此功能。

参数	Description
admissionControl.dynamic.enforceOnCreates	此设置控制 Red Hat Advanced Cluster Security for Kubernetes 是否评估策略 ; 如果被禁用，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。
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 控制器。
admissionControl.dynamic.timeout	在评估准入审核请求时，Red Hat Advanced Cluster Security for Kubernetes 应该等待的时间（以秒为单位）。使用它来设置启用镜像扫描时的请求超时。如果镜像扫描运行的时间比指定的时间长，Red Hat Advanced Cluster Security for Kubernetes 接受请求。
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。

参数	Description
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。
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 控制器的编配 secret 创建。
collector.slimMode	如果要使用 slim Collector 镜像部署 Collector，请指定 **true**。使用带有 EBPF 集合方法的 slim Collector 镜像需要 Central 提供匹配的 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 控制器的资源规格。
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 控制器的端口号 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。

参数	Description
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一个 Scanner-slim 和 Scanner DB，以允许扫描 OpenShift Container Registry 上的镜像。OpenShift Container Platform 和 Kubernetes 安全集群中支持启用 Scanner-slim。默认值为 **true**。
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 部署中的自动扩展。
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。

参数	Description
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。
scanner.dbResources.limits.cpu	Scanner DB 容器的 CPU 限制。使用此参数覆盖默认值。
monitoring.openshift.enabled	如果将此选项设置为 **false**，Red Hat Advanced Cluster Security for Kubernetes 将不会设置 Red Hat OpenShift 监控。在 Red Hat OpenShift 4 上默认为 **true**。

4.4.2.2.1.1. 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）。

4.4.2.2.2. 使用自定义安装 secured-cluster-services Helm chart

配置 **values-public.yaml** 和 **values-private.yaml** 文件后，安装 **secure-cluster-services** Helm chart 以部署以下 per-cluster 和 per-node 组件：

- Sensor
- 准入控制器
- Collector
- scanner：安装 StackRox Scanner 时为安全集群可选
- 扫描程序 DB：安装 StackRox Scanner 时为安全集群可选
- 安装 Scanner V4 Indexer 和 Scanner V4 DB 时，扫描程序 V4 Indexer 和 Scanner V4 DB: 可选

重要

扫描程序 V4 只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅技术预览功能支持范围。

先决条件

- 您必须已为集群生成 RHACS init 捆绑包。
- 您必须有权访问 Red Hat Container Registry 和一个 pull secret 进行身份验证。有关从 **registry.redhat.io** 下载镜像的详情，请参考 Red Hat Container Registry Authentication 。
- 您必须有用于公开 Central 服务的地址和端口号。

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
  --set imagePullSecrets.username=<username> \ 2
  --set imagePullSecrets.password=<password> 3
```

1. 使用 **-f** 选项指定 YAML 配置文件的路径。
2. 为 Red Hat Container Registry 身份验证包含 pull secret 的用户名。
3. 包括 Red Hat Container Registry 身份验证的 pull secret 密码。

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。

其他资源

- 为 Red Hat OpenShift 上的 RHACS 生成并应用 init 捆绑包

4.4.2.3. 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。

当使用 **helm upgrade** 命令进行修改时，会应用以下准则和要求：

- 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，每当您进行更改时，您必须手动指定所有选项。
- 有些更改（如启用 Scanner V4）需要为组件发布新证书。因此，您必须在进行这些更改时提供 CA。

重要

扫描程序 V4 只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅技术预览功能支持范围。

  - 如果 CA 在初始安装过程中由 Helm chart 生成，则必须从集群中检索这些值，并将其提供给 **helm upgrade** 命令。**central-services** Helm Chart 的安装后备注包括用于检索自动生成的值的命令。
  - 如果 CA 在 Helm Chart 之外生成，并在安装 **central-services** chart 时提供，那么您必须在使用 **helm upgrade** 命令时再次执行该操作，例如在 **helm upgrade** 命令中使用 **--reuse-values** 标志。

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 如果您修改了没有包括在 **values_public.yaml** 和 **values_private.yaml** 文件中的值，请包含 **--reuse-values** 参数。

4.4.3. 使用 roxctl CLI 在安全集群中安装 RHACS

此方法也称为清单安装方法。

先决条件

- 如果您计划使用 **roxctl** CLI 命令生成传感器安装脚本使用的文件，则已安装 **roxctl** CLI。
- 您已生成供传感器安装脚本使用的文件。

流程

- 在 OpenShift Container Platform 安全集群中，通过运行传感器安装脚本来部署 Sensor 组件。

4.4.3.1. 安装 roxctl CLI

您必须首先下载二进制文件。您可以在 Linux、Windows 或 macOS 上安装 **roxctl**。

4.4.3.1.1. 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。

注意

用于 Linux 的 **roxctl** CLI 可用于 **amd64**、 **ppc64le** 和 **s390x** 架构。

流程

1. 确定目标操作系统的 **roxctl** 架构：

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

2. 下载 **roxctl** CLI：

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

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
```

4.4.3.1.2. 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。

注意

用于 macOS 的 **roxctl** CLI 可用于 **amd64** 架构。

流程

1. 下载 **roxctl** CLI：

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

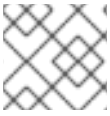
验证

- 验证您已安装的 **roxctl** 版本：

```
$ roxctl version
```

4.4.3.1.3. 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。



注意

amd64 架构提供了适用于 Windows 的 **roxctl** CLI。

流程

- 下载 **roxctl** CLI：

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

验证

- 验证您已安装的 **roxctl** 版本：

```
$ roxctl version
```

4.4.3.2. 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。此安装方法也称为清单安装方法。

要使用清单安装方法执行安装，*请仅遵循以下流程之一*：

- 使用 RHACS web 门户下载集群捆绑包，然后提取并运行传感器脚本。
- 使用 **roxctl** CLI 为 OpenShift Container Platform 集群生成所需的传感器配置，并将其与您的 Central 实例关联。

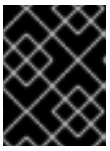
先决条件

- 您必须已安装了 Central 服务，也可以在 Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service)上选择 **ACS 实例** 来访问 Central 服务。

4.4.3.2.1. 使用 Web 门户的清单安装方法

流程

1. 在安全集群中，在 RHACS 门户中，进入 **Platform Configuration → Clusters**。
2. 选择 **Secure a cluster → Legacy 安装方法**。
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. 在可以访问被监控的集群的系统中，从集群捆绑包中提取并运行 **sensor** 脚本：

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

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

如果您收到没有部署 Sensor 所需的权限的警告，请按照屏幕说明操作，或联系集群管理员寻求帮助。

部署 Sensor 后，它会联系中心并提供集群信息。

4.4.3.2.2. 使用 roxctl CLI 安装清单

流程

1. 运行以下命令，为 OpenShift Container Platform 集群生成所需的传感器配置，并将其与您的 Central 实例关联：

   -

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

1 对于 **--openshift-version** 选项，请指定集群的主 OpenShift Container Platform 版本号。例如，为 OpenShift Container Platform 版本 **3.x** 指定 **3**，为 OpenShift Container Platform 版本 **4.x** 指定 **4**。

2. 在可以访问被监控的集群的系统中，从集群捆绑包中提取并运行 **sensor** 脚本：

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

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

如果您收到没有部署 Sensor 所需的权限的警告，请按照屏幕说明操作，或联系集群管理员寻求帮助。

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，当在 **Platform Configuration → Clusters** 中查看集群列表时，集群状态会显示一个绿色勾号和 **Healthy** 状态。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 中输入以下命令：

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

   - 在 Kubernetes 上，输入以下命令：

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

2. 点 **Finish** 关闭窗口。

安装后，Sensor 开始向 RHACS 报告安全信息，RHACS 门户仪表板开始显示部署、镜像和策略违反情况。

4.5. 使用 OPERATOR 为 RHACS 配置安全集群服务选项

当使用 Operator 安装安全集群服务时，您可以配置可选设置。

4.5.1. 安全集群服务配置选项

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。

4.5.1.1. 所需的配置设置

参数	Description

参数	Description
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。如果没有为此参数指定值，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，您无法再次更改它。要更改名称，您必须删除并重新创建对象。

4.5.1.2. 准入控制器设置

参数	Description
admissionControl.listenOnCreates	指定 **true** 以启用创建对象的防止策略强制。默认值为 **true**。
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 **true**。
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**。

参数	Description
admissionControl.timeoutSeconds	在将 Red Hat Advanced Cluster Security for Kubernetes 标记为失败前，使用此参数指定 Red Hat Advanced Cluster Security for Kubernetes 的最大秒数。

4.5.1.3. 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	Description
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。

参数	Description
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。
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**。

4.5.1.4. 镜像配置

在使用自定义 registry 时使用镜像配置设置。

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

4.5.1.5. 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	Description
perNode.collector.collection	系统级数据收集的方法。默认值为 **CORE_BPF**。红帽建议将 **CORE_BPF** 用于数据收集。如果您选择 **NoCollection**，Collector 不会报告任何有关网络活动的信息，以及进程执行。可用选项包括 **NoCollection**、**EBPF** 和 **CORE_BPF**。 **注意** 红帽弃用了 **EBPF** 选项，并将在以后的版本中删除。使用 **CORE_BPF** 替代。
perNode.collector.imageFlavor	用于 Collector 的镜像类型。您可以将它指定为 **Regular** 或 **Slim**。**常规** 镜像更大，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**。

参数	Description
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。

4.5.1.6. 污点容限设置

参数	Description
taintToleration	为确保对集群活动进行全面监控，Red Hat Advanced Cluster Security for Kubernetes 在集群中的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**。

4.5.1.7. 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	Description
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。

4.5.1.8. 常规设置和各种设置

参数	Description
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。

参数	Description
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。
overlays	请参阅使用带有覆盖的 Operator 自定义安装

4.5.2. 使用带有覆盖的 Operator 自定义安装

了解如何通过 overlays 使用 Operator 方法定制 RHACS 安装。

4.5.2.1. overlays

当 **Central** 或 **SecuredCluster** 自定义资源没有以参数的形式公开某些低级别配置选项时，您可以使用 **.spec.overlays** 字段进行调整。使用此字段来修改这些自定义资源生成的 Kubernetes 资源。

.spec.overlays 字段由一系列补丁组成，按其列出的顺序应用。这些补丁由 Kubernetes 资源上的 Operator 在部署到集群前由 Kubernetes 资源处理。

警告

Central 和 **SecuredCluster** 中的 **.spec.overlays** 字段允许用户以任意方式修改低级别 Kubernetes 资源。只有在所需的自定义无法通过 **SecuredCluster** 或 **Central** 自定义资源提供时，才使用此功能。

对 **.spec.overlays** 功能的支持主要是有限的，因为它授予了对 Kubernetes 资源进行 intricate 和高度具体的修改的功能，这可能因一个实施而异。这种自定义级别引入了一个超过标准使用场景的复杂性，这使其难以提供广泛的支持。每个修改都是唯一的，可能在产品的不同版本和配置中以无法预计的方式与 Kubernetes 系统交互。这种差异意味着排除并保证这些自定义的稳定性需要一定程度的专业知识和理解。因此，虽然此功能支持定制 Kubernetes 资源来满足精确需求，但还必须考虑确保配置的兼容性和稳定性，特别是在升级或更改底层产品期间。

以下示例显示了覆盖的结构：

```
overlays:
- apiVersion: v1 1
  kind: ConfigMap 2
  name: my-configmap 3
```

```
patches:
  - path: .data      4
    value: |         5
      key1: data2
      key2: data2
```

1. targeted Kubernetes resource ApiVersion，如 **apps/v1**,**v1**,**networking.k8s.io/v1**
2. 资源类型（如 Deployment、ConfigMap、NetworkPolicy）
3. 资源的名称，如 **my-configmap**
4. 字段的 jsonpath 表达式，如 **spec.template.spec.containers[name:central].env[-1]**
5. 新字段值的 YAML 字符串

4.5.2.1.1. 添加覆盖

对于自定义，您可以在 **Central** 或 **SecuredCluster** 自定义资源中添加覆盖。使用 OpenShift CLI (**oc**)或 OpenShift Container Platform Web 控制台进行修改。

如果覆盖没有按预期生效，请检查 RHACS Operator 日志是否有语法错误或记录的问题。

4.5.2.2. 覆盖示例

4.5.2.2.1. 为 Central ServiceAccount 指定 EKS pod 角色 ARN

在 **中央** ServiceAccount 中添加 Amazon Elastic Kubernetes Service (EKS) pod 角色 Amazon Resource Name (ARN)注解，如下例所示：

```
apiVersion: platform.stackrox.io
kind: Central
metadata:
  name: central
spec:
  # ...
  overlays:
  - apiVersion: v1
    kind: ServiceAccount
    name: central
    patches:
      - path: metadata.annotations.eks\.amazonaws\.com/role-arn
        value: "\"arn:aws:iam:1234:role\""
```

4.5.2.2.2. 将环境变量注入中央部署

将环境变量注入到 **中央** 部署中，如下例所示：

```
apiVersion: platform.stackrox.io
kind: Central
metadata:
  name: central
spec:
```

```
  # ...
  overlays:
  - apiVersion: apps/v1
    kind: Deployment
    name: central
    patches:
    - path: spec.template.spec.containers[name:central].env[-1]
      value: |
        name: MY_ENV_VAR
        value: value
```

4.5.2.2.3. 使用入口规则扩展网络策略

在 **allow-ext-to-central** 网络策略中添加一个入口规则，用于端口 999 流量，如下例所示：

```
apiVersion: platform.stackrox.io
kind: Central
metadata:
  name: central
spec:
    # ...
    overlays:
    - apiVersion: networking.k8s.io/v1
      kind: NetworkPolicy
      name: allow-ext-to-central
      patches:
        - path: spec.ingress[-1]
          value: |
            ports:
            - port: 999
              protocol: TCP
```

4.5.2.2.4. 修改 ConfigMap 数据

修改 **central-endpoints** ConfigMap 数据，如下例所示：

```
apiVersion: platform.stackrox.io
kind: Central
metadata:
  name: central
spec:
    # ...
    overlays:
    - apiVersion: v1
      kind: ConfigMap
      name: central-endpoints
      patches:
      - path: data
        value: |
          endpoints.yaml: |
            disableDefault: false
```

4.5.2.2.5. 将容器添加到 中央 部署中

在 **中央** 部署中添加新容器，如下例所示：

```
apiVersion: platform.stackrox.io
kind: Central
metadata:
  name: central
spec:
    # ...
    overlays:
    - apiVersion: apps/v1
      kind: Deployment
      name: central
      patches:
        - path: spec.template.spec.containers[-1]
      value: |
        name: nginx
        image: nginx
        ports:
          - containerPort: 8000
            name: http
            protocol: TCP
```

4.6. 验证 RED HAT OPENSHIFT 上的 RHACS 安装

提供验证 RHACS 是否已正确安装的步骤。

4.6.1. 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情况。

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. 转至 **https://localhost:18443/**。

2. 使用 Red Hat OpenShift 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=quay.io/stackrox-io/docs:example-vulnerables-cve-2014-6271 -n test
$ oc run samba --labels=app=rce \
  --image=quay.io/stackrox-io/docs:example-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和策略违反情况。进入 RHACS 门户来查看违反情况。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 在其他平台上安装 RHACS

5.1. 在其他平台上安装 RHACS 的高级别概述

Red Hat Advanced Cluster Security for Kubernetes (RHACS)在 Amazon Elastic Kubernetes Service (Amazon EKS)、Google Kubernetes Engine (Google GKE)和 Microsoft Azure Kubernetes Service (Microsoft AKS)上为自我管理的 RHACS 提供安全服务。

安装前：

- 了解 不同平台的安装方法。
- 了解 Red Hat Advanced Cluster Security for Kubernetes 架构 。
- 检查默认资源要求页面。

以下列表提供了安装步骤的高级概述：

1. 使用 Helm chart 或 **roxctl** CLI 在集群中安装 Central 服务。
2. 生成并应用 init 捆绑包。
3. 在每个安全集群中安装安全集群资源。

5.2. 在其他平台上为 RHACS 安装 CENTRAL 服务

Central 是包含 RHACS 应用程序管理界面和服务的资源。它处理数据持久性、API 交互和 RHACS 门户访问。您可以使用同一中实例来保护多个 OpenShift Container Platform 或 Kubernetes 集群。

您可以使用以下方法之一安装 Central：

- 使用 Helm chart 安装
- 使用 **roxctl** CLI 安装（除非有需要使用它的特定安装需要）

5.2.1. 使用 Helm chart 安装 Central

您可以使用 Helm chart 安装 Central，而无需自定义任何自定义，使用默认值，或使用带有额外自定义配置参数的 Helm chart。

5.2.1.1. 使用 Helm chart 安装 Central，而无需自定义

您可以在 Red Hat OpenShift 集群上安装 RHACS。您必须添加 Helm Chart 仓库并安装 **central-services** Helm Chart，以安装 Central 和 Scanner 的集中组件。

5.2.1.1.1. 添加 Helm Chart 仓库

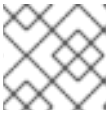
流程

- 添加 RHACS chart 存储库。

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

Red Hat Advanced Cluster Security for Kubernetes 的 Helm 仓库包括用于安装不同组件的 Helm chart，包括：

- 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 per-cluster 和 per-node 组件 (Sensor、Admission Controller、Collector 和 Scanner-slim)。

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/
```

5.2.1.1.2. 在不自定义的情况下安装 central-services Helm chart

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

前提条件

- 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 Authentication 。

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.username=<username> \ 1
  --set imagePullSecrets.password=<password> \ 2
  --set central.exposure.route.enabled=true
```

1. 为 Red Hat Container Registry 身份验证包含 pull secret 的用户名。
2. 包括 Red Hat Container Registry 身份验证的 pull secret 密码。

- 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.username=<username> \ 1
```

```
--set imagePullSecrets.password=<password> \ 2
--set central.exposure.loadBalancer.enabled=true
```

1 为 Red Hat Container Registry 身份验证包含 pull secret 的用户名。

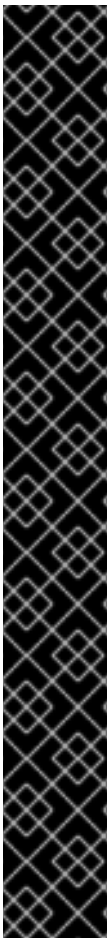
2 包括 Red Hat Container Registry 身份验证的 pull secret 密码。

- 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.username=<username> \ 1
  --set imagePullSecrets.password=<password>  2
```

1 为 Red Hat Container Registry 身份验证包含 pull secret 的用户名。

2 包括 Red Hat Container Registry 身份验证的 pull secret 密码。



重要

- 如果要在需要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
```

- 如果您已在安装的命名空间中创建了一个或多个镜像 pull secret，而不是使用用户名和密码，您可以使用 **--set imagePullSecrets.useExisting="<pull-secret-1;pull-secret-2>"**。
- 不要使用镜像 pull secret：
  - 如果您要从 **quay.io/stackrox-io** 或不需要身份验证的专用网络中的 registry 拉取镜像。使用 **--set imagePullSecrets.allowNone=true**，而不是指定用户名和密码。
  - 如果您已经在安装的命名空间中的默认服务帐户中配置了镜像 pull secret。使用 **--set imagePullSecrets.useFromDefaultServiceAccount=true**，而不是指定用户名和密码。

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。

5.2.1.2. 使用带有自定义的 Helm chart 安装 Central

您可以使用 **helm install** 和 **helm upgrade** 命令的 Helm Chart 配置参数在 Red Hat OpenShift 集群上安装 RHACS。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。确保您安全地存储这个文件。
- 配置文件 **declarative-config-values.yaml** ：如果您使用声明性配置将声明性配置挂载添加到 Central，请创建此文件。

5.2.1.2.1. 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。

5.2.1.2.1.1. 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	Description
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。

参数	Description
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**。

5.2.1.2.1.2. 代理配置

如果要在需要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	Description
env.proxyConfig	您的代理配置。

5.2.1.2.1.3. 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**
- **central.db.serviceTLS.cert**
- **central.db.serviceTLS.key**
- **central.db.password.value**
- 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	Description
central.jwtSigner.key	RHACS 应该用来签名 JSON Web 令牌(JWT)进行身份验证的私钥。
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。RHACS 将这个证书用于 RHACS 门户。 ● 对于新安装，您必须提供证书，否则 RHACS 使用自签名证书安装 Central。 ● 如果要升级，RHACS 将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 RHACS 使用自签名证书安装 Central。 ● 如果要升级，RHACS 将使用现有证书及其密钥。
central.db.password.value	Central 数据库的连接密码。
central.adminPassword.value	用于登录到 RHACS 的管理员密码。
central.adminPassword.htpasswd	用于登录到 RHACS 的管理员密码。此密码以散列格式存储，使用 bcrypt。
central.db.serviceTLS.cert	Central DB 服务应用于部署 Central DB 的内部证书。
central.db.serviceTLS.key	Central DB 服务应使用的内部证书的私钥。
central.db.password.value	用于连接到 Central DB 的密码。

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>
```

5.2.1.2.1.4. 扫描程序

StackRox Scanner 和 Scanner V4 的可配置参数（技术预览）。

重要

扫描程序 V4 只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅技术预览功能支持范围。

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**
- **scannerV4.db.password.value**
- **scannerV4.indexer.serviceTLS.cert**
- **scannerV4.indexer.serviceTLS.key**
- **scannerV4.matcher.serviceTLS.cert**
- **scannerV4.matcher.serviceTLS.key**
- **scannerV4.db.serviceTLS.cert**
- **scannerV4.db.serviceTLS.key**

参数	Description
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 RHACS 自动在内部创建和使用其值。
scanner.serviceTLS.cert	StackRox Scanner 服务用于部署 StackRox 扫描器的内部证书。

参数	Description
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。
scannerV4.db.password.value	用于通过 Scanner V4 数据库进行身份验证的密码。不要修改此参数，因为 RHACS 自动在内部创建和使用其值。
scannerV4.db.serviceTLS.cert	Scanner V4 DB 服务用于部署 Scanner V4 数据库的内部证书。
scannerV4.db.serviceTLS.key	Scanner V4 DB 服务应使用的内部证书的私钥。
scannerV4.indexer.serviceTLS.cert	Scanner V4 服务用于部署 Scanner V4 Indexer 的内部证书。
scannerV4.indexer.serviceTLS.key	Scanner V4 Indexer 使用的内部证书的私钥。
scannerV4.matcher.serviceTLS.cert	Scanner V4 服务用于部署 Scanner V4 Matcher 的内部证书。
scannerV4.matcher.serviceTLS.key	Scanner V4 Matcher 应该使用的内部证书的私钥。

5.2.1.2.2. 公共配置文件

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

5.2.1.2.2.1. 镜像 pull secret

镜像拉取 secret 是从 registry 中拉取镜像所需的凭证。

参数	Description
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。

参数	Description
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**。

5.2.1.2.2.2. 镜像

镜像声明配置来设置主 registry，Helm Chart 用来解析 **central.image**、**scanner.image**、**scanner.dbImage**、**scannerV4.image** 和 **scannerV4.db.image** 参数的镜像。

参数	Description
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**。

5.2.1.2.2.3. 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	Description
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	要安装 RHACS 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** 在离线模式下使用 RHACS。

5.2.1.2.2.4. 其他可信证书颁发机构

RHACS 自动引用要信任的系统根证书。当 Central 时，StackRox Scanner 或 Scanner V4 必须联系到使用您机构中颁发机构发布的证书或全局可信合作伙伴机构发布的服务的服务，您可以使用以下参数来指定对这些服务的信任：

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

5.2.1.2.2.5. 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。

下表包含外部 PostgreSQL 数据库的设置。

参数	Description
central.declarativeConfiguration.mounts.configMaps	挂载用于声明配置的配置映射。
Central.declarativeConfiguration.mounts.secrets	挂载用于声明配置的 secret。
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。
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。

参数	Description
central.resources.requests.cpu	Central 的 CPU 请求。
central.resources.limits.memory	Central 的内存限值。
central.resources.limits.cpu	Central 的 CPU 限制。
central.persistence.hostPath	RHACS 应该创建数据库卷的节点上的路径。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** 创建新 PVC 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 RHACS，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。
central.db.external	使用 **true** 指定不应部署中央 DB，并且将使用外部数据库。
central.db.source.connectionString	用于连接到数据库的 Central 的连接字符串。这仅在将 **central.db.external** 设置为 true 时使用。连接字符串必须采用 keyword/value 格式，如 PostgreSQL 文档中的 "Additional resources" 所述。 ● 仅支持 PostgreSQL 13。 ● 不支持通过 PgBouncer 连接。 ● 用户必须是超级用户，能够创建和删除数据库。

参数	Description
central.db.source.minConns	与要建立的数据库的最小连接数。
central.db.source.maxConns	与要建立的数据库的连接数上限。
central.db.source.statementTimeoutMs	单个查询或事务的毫秒可以针对数据库处于活跃状态。
central.db.postgresConfig	用于中央 DB 的 postgresql.conf，如 PostgreSQL 文档中的"添加资源"中所述。
central.db.hbaConfig	用于 Central DB 的 pg_hba.conf，如 PostgreSQL 文档中的 "Additional resources" 所述。
central.db.nodeSelector	将节点选择器标签指定为 **label-key: label-value**，以强制 Central DB 仅调度到具有指定标签的节点。
central.db.image.registry	一个自定义 registry，用于覆盖 Central DB 镜像的全局 **image.registry** 参数。
central.db.image.name	覆盖默认中央 DB 镜像名称(**central-db**)的自定义镜像名称。
central.db.image.tag	覆盖 Central DB 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。如果您在自己的 registry 中镜像 Central DB 镜像，请不要修改原始镜像标签。
central.db.image.fullRef	Central DB 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.db.image.registry**、**central.db.image.name** 和 **central.db.image.tag** 参数。
central.db.resources.requests.memory	Central DB 的内存请求。
central.db.resources.requests.cpu	Central DB 的 CPU 请求。
central.db.resources.limits.memory	Central DB 的内存限值。
central.db.resources.limits.cpu	Central DB 的 CPU 限制。
central.db.persistence.hostPath	RHACS 应该创建数据库卷的节点上的路径。红帽不推荐使用这个选项。
central.db.persistence.persistentVolumeClaim.claimName	您要使用的持久性卷声明(PVC)的名称。

参数	Description
central.db.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.db.persistence.persistentVolumeClaim.size	由指定声明管理的持久性卷的大小（以 GiB 为单位）。

5.2.1.2.2.6. stackrox Scanner

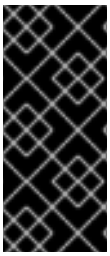
下表列出了 StackRox Scanner 的可配置参数。这是用于节点和平台扫描的扫描程序。如果没有启用 Scanner V4，StackRox 扫描程序也会执行镜像扫描。从版本 4.4 开始，可以启用 Scanner V4 以提供镜像扫描。请参阅 Scanner V4 参数的下一表。

参数	Description
scanner.disable	使用 **true** 在没有 StackRox 扫描器的情况下安装 RHACS。当将其与 **helm upgrade** 命令一起使用时，Helm 会移除现有的 StackRox Scanner 部署。
scanner.exposeMonitoring	指定 **true**，以在端口号 **9090** 上为 StackRox Scanner 公开 Prometheus 指标端点。
scanner.replicas	为 StackRox 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	配置 StackRox Scanner 的日志级别。红帽建议不要更改默认日志级别值(**INFO**)。
scanner.nodeSelector	将节点选择器标签指定为 **label-key: label-value**，以强制 StackRox 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 StackRox 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 StackRox 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	StackRox Scanner 的内存请求。
scanner.resources.requests.cpu	StackRox Scanner 的 CPU 请求。

参数	Description
scanner.resources.limits.memory	StackRox Scanner 的内存限值。
scanner.resources.limits.cpu	StackRox 扫描器的 CPU 限制。
scanner.dbResources.requests.memory	StackRox Scanner 数据库部署的内存请求。
scanner.dbResources.requests.cpu	StackRox Scanner 数据库部署的 CPU 请求。
scanner.dbResources.limits.memory	StackRox Scanner 数据库部署的内存限值。
scanner.dbResources.limits.cpu	StackRox Scanner 数据库部署的 CPU 限制。
scanner.image.registry	StackRox Scanner 镜像的自定义 registry。
scanner.image.name	覆盖默认 StackRox Scanner 镜像名称(**扫描程序**)的自定义镜像名称。
scanner.dbImage.registry	StackRox Scanner DB 镜像的自定义 registry。
scanner.dbImage.name	覆盖默认 StackRox 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 StackRox 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 StackRox 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。

5.2.1.2.2.7. scanner V4

下表列出了 Scanner V4 的可配置参数。



重要

扫描程序 V4 只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅技术预览功能支持范围。

参数	Description

参数	Description
scannerV4.db.persistence.persistentVolumeClaim.claimName	用于管理 Scanner V4 持久数据的 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 **scanner-v4-db**。为防止数据丢失，当 Central 被删除时，PVC 不会被自动删除。
scannerV4.disable	使用 **false** 启用 Scanner V4。当设置此参数时，还必须通过设置 **scanner.disable=false** 来启用 StackRox Scanner。在达到 StackRox Scanner 和 Scanner V4 之间的功能奇偶校验前，Scanner V4 只能与 StackRox Scanner 结合使用。不支持在没有启用 StackRox 扫描器的情况下启用 Scanner V4。当使用 **helm upgrade** 命令将此参数设置为 **true** 时，Helm 会移除现有的 Scanner V4 部署。
scannerV4.exposeMonitoring	指定 **true**，以在端口号 **9090** 上为 Scanner V4 公开 Prometheus 指标端点。
scannerV4.indexer.replicas	为 Scanner V4 Indexer 部署创建的副本数。当您将 **scannerV4.indexer.autoscaling** 参数一起使用时，这个值会设置初始副本数。
scannerV4.indexer.logLevel	配置 Scanner V4 Indexer 的日志级别。红帽建议不要更改默认日志级别值(**INFO**)。
scannerV4.indexer.nodeSelector	将节点选择器标签指定为 **label-key: label-value**，以强制 Scanner V4 Indexer 仅调度到具有指定标签的节点。
scannerV4.indexer.tolerations	如果节点选择器选择污点节点，请使用此参数为 Scanner V4 Indexer 指定污点容限键、值和效果。此参数主要用于基础架构节点。
scannerV4.indexer.autoscaling.disable	使用 **true** 为 Scanner V4 Indexer 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。
scannerV4.indexer.autoscaling.minReplicas	自动扩展的最小副本数。
scannerV4.indexer.autoscaling.maxReplicas	自动扩展的最大副本数。
scannerV4.indexer.resources.requests.memory	Scanner V4 Indexer 的内存请求。
scannerV4.indexer.resources.requests.cpu	Scanner V4 Indexer 的 CPU 请求。
scannerV4.indexer.resources.limits.memory	Scanner V4 Indexer 的内存限值。
scannerV4.indexer.resources.limits.cpu	Scanner V4 Indexer 的 CPU 限制。

参数	Description
scannerV4.matcher.replicas	为 Scanner V4 Matcher 部署创建的副本数。当您将 **scannerV4.matcher.autoscaling** 参数一起使用时，这个值会设置初始副本数。
scannerV4.matcher.logLevel	红帽建议不要更改默认日志级别值(**INFO**)。
scannerV4.matcher.nodeSelector	将节点选择器标签指定为 **label-key: label-value**，以强制 Scanner V4 Matcher 仅调度到具有指定标签的节点。
scannerV4.matcher.tolerations	如果节点选择器选择污点节点，请使用此参数为 Scanner V4 Matcher 指定污点容限键、值和效果。此参数主要用于基础架构节点。
scannerV4.matcher.autoscaling.disable	使用 **true** 为 Scanner V4 Matcher 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。
scannerV4.matcher.autoscaling.minReplicas	自动扩展的最小副本数。
scannerV4.matcher.autoscaling.maxReplicas	自动扩展的最大副本数。
scannerV4.matcher.resources.requests.memory	Scanner V4 Matcher 的内存请求。
scannerV4.matcher.resources.requests.cpu	Scanner V4 Matcher 的 CPU 请求。
scannerV4.db.resources.requests.memory	Scanner V4 数据库部署的内存请求。
scannerV4.db.resources.requests.cpu	Scanner V4 数据库部署的 CPU 请求。
scannerV4.db.resources.limits.memory	Scanner V4 数据库部署的内存限值。
scannerV4.db.resources.limits.cpu	Scanner V4 数据库部署的 CPU 限制。
scannerV4.db.nodeSelector	将节点选择器标签指定为 **label-key: label-value**，以强制 Scanner V4 DB 仅调度到具有指定标签的节点。
scannerV4.db.tolerations	如果节点选择器选择污点节点，请使用此参数为 Scanner V4 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。
scannerV4.db.image.registry	Scanner V4 DB 镜像的自定义 registry。
scannerV4.db.image.name	覆盖默认 Scanner V4 DB 镜像名称(**scanner-v4-db**)的自定义镜像名称。

参数	Description
scannerV4.image.registry	Scanner V4 镜像的自定义 registry。
scannerV4.image.name	覆盖默认 Scanner V4 镜像名称(**scanner-v4)的自定义镜像名称**。

5.2.1.2.2.8. 自定义

使用这些参数为 RHACS 创建的所有对象指定附加属性。

参数	Description
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。
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。

参数	Description
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。
customize.scanner-v4-indexer.labels	附加到 Scanner V4 Indexer 创建的所有对象的自定义标签，并附加到属于它们的 pod。
customize.scanner-v4-indexer.annotations	附加到 Scanner V4 Indexer 创建的所有对象的自定义注解，并附加到属于它们的 pod。
customize.scanner-v4-indexer.podLabels	附加到 Scanner V4 Indexer 创建的所有对象的自定义标签，并附加到属于它们的 pod。
customize.scanner-v4-indexer.podAnnotations	附加到 Scanner V4 Indexer 创建的所有对象的自定义注解，并附加到属于它们的 pod。
customize.scanner-4v-indexer.envVars	所有 Scanner V4 Indexer 容器及其属于它们的 pod 的自定义环境变量。
customize.scanner-v4-matcher.labels	附加到 Scanner V4 Matcher 创建的所有对象的自定义标签，并放入它们所属的 pod。
customize.scanner-v4-matcher.annotations	附加到 Scanner V4 Matcher 创建的所有对象的自定义注解，并附加到它们所属的 pod。
customize.scanner-v4-matcher.podLabels	附加到 Scanner V4 Matcher 创建的所有对象的自定义标签，并放入它们所属的 pod。
customize.scanner-v4-matcher.podAnnotations	附加到 Scanner V4 Matcher 创建的所有对象的自定义注解，并附加到它们所属的 pod。
customize.scanner-4v-matcher.envVars	所有 Scanner V4 Matcher 容器及其属于它们的 pod 的自定义环境变量。
customize.scanner-v4-db.labels	附加到 Scanner V4 DB 创建的所有对象的自定义标签，并附加到它们所属的 pod。
customize.scanner-v4-db.annotations	附加到 Scanner V4 DB 创建的所有对象的自定义注解，并附加到它们所属的 pod。
customize.scanner-v4-db.podLabels	附加到 Scanner V4 DB 创建的所有对象的自定义标签，并附加到它们所属的 pod。

参数	Description
customize.scanner-v4-db.podAnnotations	附加到 Scanner V4 DB 创建的所有对象的自定义注解，并附加到它们所属的 pod。
customize.scanner-4v-db.envVars	所有 Scanner V4 DB 容器及其属于它们的 pod 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。

5.2.1.2.2.9. 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命名空间和发行版本名称的 RHACS 实例。

参数	Description
allowNonstandardNamespace	使用 **true** 将 RHACS 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** 以外的发行版本名称部署 RHACS。

5.2.1.2.3. 声明性配置值

要使用声明性配置，您必须创建一个 YAML 文件（在这个示例中，名为 "declarative-config-values.yaml"），以将声明性配置挂载添加到 Central。此文件用于 Helm 安装。

流程

1. 使用以下示例创建 YAML 文件（本例中为 **declarative-config-values.yaml**）：

```
central:
  declarativeConfiguration:
    mounts:
      configMaps:
        - declarative-configs
      secrets:
        - sensitive-declarative-configs
```

2. 安装 Central 服务 Helm chart，如"安装 central-services Helm Chart"中所述，引用 **declarative-config-values.yaml** 文件。

其他资源

- 连接字符串 - PostgreSQL 文档
- 通过配置文件进行参数交互 - PostgreSQL 文档
- pg_hba.conf 文件 - PostgreSQL 文档

5.2.1.2.4. 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。

注意

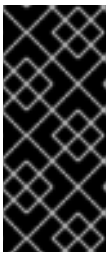
可选：如果使用声明性配置，请将 **-f <path_to_declarative-config-values.yaml** 添加到此命令，以便在 Central 中挂载声明性配置文件。

5.2.1.3. 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。

当使用 **helm upgrade** 命令进行修改时，会应用以下准则和要求：

- 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，每当您进行更改时，您必须手动指定所有选项。
- 有些更改（如启用 Scanner V4）需要为组件发布新证书。因此，您必须在进行这些更改时提供 CA。



重要

扫描程序 V4 只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅技术预览功能支持范围。

  - 如果 CA 在初始安装过程中由 Helm chart 生成，则必须从集群中检索这些值，并将其提供给 **helm upgrade** 命令。**central-services** Helm Chart 的安装后备注包括用于检索自动生成的值的命令。

- 如果 CA 在 Helm Chart 之外生成，并在安装 **central-services** chart 时提供，那么您必须在使用 **helm upgrade** 命令时再次执行该操作，例如在 **helm upgrade** 命令中使用 **--reuse-values** 标志。

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 \
  --reuse-values \ 1
  -f <path_to_init_bundle_file \
  -f <path_to_values_public.yaml> \
  -f <path_to_values_private.yaml>
```

1 如果您修改了没有包括在 **values_public.yaml** 和 **values_private.yaml** 文件中的值，请包含 **--reuse-values** 参数。

5.2.2. 使用 roxctl CLI 安装 Central

警告

对于生产环境，红帽建议使用 Operator 或 Helm chart 来安装 RHACS。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。

5.2.2.1. 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**。

5.2.2.1.1. 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。

注意

用于 Linux 的 **roxctl** CLI 可用于 **amd64**、 **ppc64le** 和 **s390x** 架构。

流程

1. 确定目标操作系统的 **roxctl** 架构：

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

2. 下载 **roxctl** CLI：

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

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
```

5.2.2.1.2. 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。

注意

用于 macOS 的 **roxctl** CLI 可用于 **amd64** 架构。

流程

1. 下载 **roxctl** CLI：

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

5.2.2.1.3. 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。

注意

amd64 架构提供了适用于 Windows 的 **roxctl** CLI。

流程

- 下载 **roxctl** CLI：

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

验证

- 验证您已安装的 **roxctl** 版本：

```
$ roxctl version
```

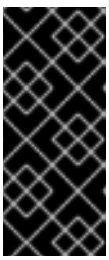
5.2.2.2. 使用交互式安装程序

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
```



重要

使用 **roxctl** CLI 安装 RHACS 会创建 PodSecurityPolicy (PSP)对象，以便向后兼容。如果要在 Kubernetes 版本 1.25 及更新版本上，或在 OpenShift Container Platform version 4.12 和更新版本上安装 RHACS，则必须禁用 PSP 对象的创建。要做到这一点，对于 **roxctl central generate** 和 **roxctl sensor generate** 命令，将 **--enable-pod-security-policies** 选项设置为 **false**。

2. 按 **Enter** 接受提示的默认值或根据需要输入自定义值。以下示例显示了交互式安装程序提示：

```
Enter path to the backup bundle from which to restore keys and certificates (optional):
Enter read templates from local filesystem (default: "false"):
Enter path to helm templates on your local filesystem (default: "/path"):
Enter PEM cert bundle file (optional): 1
Enter Create PodSecurityPolicy resources (for pre-v1.25 Kubernetes) (default: "true"): 2
Enter administrator password (default: autogenerated):
Enter orchestrator (k8s, openshift):
Enter default container images settings (development_build, stackrox.io, rhacs, opensource);
it controls repositories from where to download the images, image names and tags format
(default: "development_build"):
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"):
```

```
Enter whether to enable telemetry (default: "false"):
Enter central-db image to use (if unset, a default will be used according to --image-defaults):
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"): 3
Enter main image to use (if unset, a default will be used according to --image-defaults):
Enter whether to run StackRox in offline mode, which avoids reaching out to the Internet
(default: "false"):
Enter list of secrets to add as declarative configuration mounts in central (default: "[]"): 4
Enter list of config maps to add as declarative configuration mounts in central (default: "[]"):
5
Enter the deployment tool to use (kubectl, helm, helm-values) (default: "kubectl"):
Enter scanner-db image to use (if unset, a default will be used according to --image-defaults):
Enter scanner image to use (if unset, a default will be used according to --image-defaults):
Enter Central volume type (hostpath, pvc): 6
Enter external volume name for Central (default: "stackrox-db"):
Enter external volume size in Gi for Central (default: "100"):
Enter storage class name for Central (optional if you have a default StorageClass
configured):
Enter external volume name for Central DB (default: "central-db"):
Enter external volume size in Gi for Central DB (default: "100"):
Enter storage class name for Central DB (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. 如果您正在运行 Kubernetes 版本 1.25 或更高版本，请将此值设置为 **false**。
3. 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。
4. 有关使用声明配置进行身份验证和授权的更多信息，请参阅 "Red Hat Advanced Cluster Security for Kubernetes 中的"管理 RBAC"中的为身份验证和授权资源提供配置。
5. 有关使用声明配置进行身份验证和授权的更多信息，请参阅 "Red Hat Advanced Cluster Security for Kubernetes 中的"管理 RBAC"中的为身份验证和授权资源提供配置。
6. 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，而是在 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）。

5.2.2.3. 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	Example
Route（路由）	**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**
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**

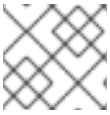
注意

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
```

5.3. 在其他平台上为 RHACS 生成并应用 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 门户或 **roxctl** CLI 创建 init 捆绑包。然后，您可以使用它应用 init 捆绑包来创建资源。



注意

您必须具有 **Admin** 用户角色才能创建 init 捆绑包。

5.3.1. 生成 init 捆绑包

5.3.1.1. 使用 RHACS 门户生成 init 捆绑包

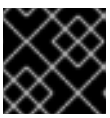
您可以使用 RHACS 门户创建包含 secret 的 init 捆绑包。

注意

您必须具有 **Admin** 用户角色才能创建 init 捆绑包。

流程

1. 如"使用 Operator 方法验证中央安装"中所述，查找 RHACS 门户的地址。
2. 登录到 RHACS 门户。
3. 如果您没有安全集群，则会出现 **Platform Configuration → Clusters** 页面。
4. 点 **Create init bundle**。
5. 为集群 init 捆绑包输入一个名称。
6. 选择您的平台。
7. 选择您要用于安全集群的安装方法： **Operator** 或 **Helm Chart**。
8. 点 **Download** 生成并下载以 YAML 文件形式创建的 init 捆绑包。如果您使用相同的安装方法，您可以对所有安全集群使用一个 init 捆绑包及其对应的 YAML 文件。

   重要

   安全地存储此捆绑包，因为它包含 secret。

9. 通过使用它来在安全集群中创建资源来应用 init 捆绑包。
10. 在每个集群中安装安全的集群服务。

5.3.1.2. 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。

注意

您必须具有 **Admin** 用户角色才能创建 init 捆绑包。

先决条件

- 您已配置了 **ROX_API_TOKEN** 和 **ROX_CENTRAL_ADDRESS** 环境变量：
  a. 运行以下命令设置 **ROX_API_TOKEN** ：

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

  b. 运行以下命令设置 **ROX_CENTRAL_ADDRESS** 环境变量：

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

流程

- 要生成包含 Helm 安装 secret 的集群 init 捆绑包，请运行以下命令：

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

- 要生成包含 Operator 安装 secret 的集群 init 捆绑包，请运行以下命令：

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

重要

确保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。

5.3.1.3. 在安全集群中应用 init 捆绑包

在配置安全集群前，您必须使用它来应用 init 捆绑包来在集群中创建所需资源。应用 init 捆绑包可让安全集群中的服务与 Central 通信。

注意

如果使用 Helm chart 安装，请不要执行此步骤。使用 Helm 完成安装 ; 请参阅"使用 Helm chart 在安全集群中安装 RHACS"。

先决条件

- 您必须生成了一个包含 secret 的 init 捆绑包。
- 您必须在安装安全集群服务的集群中创建了 **stackrox** 项目或命名空间。不需要将 **stackrox** 用于项目，而是确保在扫描集群时不会报告 RHACS 进程的漏洞。

流程

要创建资源，请执行以下步骤之一：

- 使用 OpenShift Container Platform Web 控制台创建资源：在 OpenShift Container Platform Web 控制台中，确保您位于 **stackrox** 命名空间中。在顶部菜单中，点 + 打开 **Import YAML** 页面。您可以拖动 init 捆绑包文件或将其内容复制并粘贴到编辑器中，然后点 **Create**。命令完成后，显示显示 **collector-tls**、**sensor-tls** 和 admission-control-tls 的资源已创建。
- 使用 Red Hat OpenShift CLI 创建资源：使用 Red Hat OpenShift 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 Central 服务的项目的名称。

- 使用 **kubectl** CLI，运行以下命令来创建资源：

```
$ kubectl create namespace stackrox 1
$ kubectl create -f <init_bundle>.yaml \ 2
  -n <stackrox> 3
```

1. 创建安装安全集群资源的项目。这个示例使用 **stackrox**。
2. 指定包含 secret 的 init 捆绑包的文件名。
3. 指定您创建的项目名称。这个示例使用 **stackrox**。

5.3.2. 后续步骤

- 在您要监控的所有集群中安装 RHACS 安全集群服务。

5.4. 在其他平台上为 RHACS 安装安全集群服务

您可以为 Amazon Elastic Kubernetes Service (Amazon EKS)、Google Kubernetes Engine (Google GKE) 和 Microsoft Azure Kubernetes Service (Microsoft AKS)等平台在安全集群中安装 RHACS。

5.4.1. 使用 Helm chart 在安全集群中安装 RHACS

您可以使用没有自定义的 Helm chart、使用默认值或配置参数自定义的 Helm chart 在安全集群中安装 RHACS。

5.4.1.1. 使用 Helm chart 在安全集群中安装 RHACS

5.4.1.1.1. 添加 Helm Chart 仓库

流程

- 添加 RHACS chart 存储库。

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

Red Hat Advanced Cluster Security for Kubernetes 的 Helm 仓库包括用于安装不同组件的 Helm chart，包括：

- 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 per-cluster 和 per-node 组件 (Sensor、Admission Controller、Collector 和 Scanner-slim)。

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/
```

5.4.1.1.2. 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、Collector 和 Scanner-slim)。

先决条件

- 您必须已为集群生成 RHACS init 捆绑包。
- 您必须有权访问 Red Hat Container Registry 和一个 pull secret 进行身份验证。有关从 **registry.redhat.io** 下载镜像的详情，请参考 Red Hat Container Registry Authentication 。
- 您必须有用于公开 Central 服务的地址和端口号。

其他资源

- 在其他平台上为 RHACS 生成并应用 init 捆绑包

5.4.1.2. 使用自定义配置 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。确保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。

5.4.1.2.1. 配置参数

参数	Description
clusterName	集群的名称。
centralEndpoint	Central 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。在配置多个集群时，使用地址的主机名（如 **central.example.com:443**）。
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。
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。

参数	Description
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	**CORE_BPF**、**EBPF**（已弃用）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。
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。

参数	Description
admissionControl.listenOnCreates	此设置控制 Kubernetes 是否配置为联系 Red Hat Advanced Cluster Security for Kubernetes，使用 **AdmissionReview** 请求进行工作负载创建事件。
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 Kubernetes**exec** 和 **portforward** 事件。RHACS 不支持 OpenShift Container Platform 3.11 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 ; 如果被禁用，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。
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 控制器。
admissionControl.dynamic.timeout	在评估准入审核请求时，Red Hat Advanced Cluster Security for Kubernetes 应该等待的时间（以秒为单位）。使用它来设置启用镜像扫描时的请求超时。如果镜像扫描运行的时间比指定的时间长，Red Hat Advanced Cluster Security for Kubernetes 接受请求。
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。

参数	Description
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。
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 控制器的编配 secret 创建。

参数	Description
collector.slimMode	如果要使用 slim Collector 镜像部署 Collector，请指定 **true**。使用带有 EBPF 集合方法的 slim Collector 镜像需要 Central 提供匹配的 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 控制器的资源规格。
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 控制器的端口号 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一个 Scanner-slim 和 Scanner DB，以允许扫描 OpenShift Container Registry 上的镜像。OpenShift Container Platform 和 Kubernetes 安全集群中支持启用 Scanner-slim。默认值为 **true**。
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 部署中的自动扩展。

参数	Description
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。
scanner.dbResources.limits.cpu	Scanner DB 容器的 CPU 限制。使用此参数覆盖默认值。
monitoring.openshift.enabled	如果将此选项设置为 **false**，Red Hat Advanced Cluster Security for Kubernetes 将不会设置 Red Hat OpenShift 监控。在 Red Hat OpenShift 4 上默认为 **true**。

5.4.1.2.1.1. 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）。

5.4.1.2.2. 使用自定义安装 secured-cluster-services Helm chart

配置 **values-public.yaml** 和 **values-private.yaml** 文件后，安装 **secure-cluster-services** Helm chart 以部署以下 per-cluster 和 per-node 组件：

- Sensor
- 准入控制器
- Collector
- scanner：安装 StackRox Scanner 时为安全集群可选
- 扫描程序 DB：安装 StackRox Scanner 时为安全集群可选
- 安装 Scanner V4 Indexer 和 Scanner V4 DB 时，扫描程序 V4 Indexer 和 Scanner V4 DB: 可选

重要

扫描程序 V4 只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅技术预览功能支持范围。

先决条件

- 您必须已为集群生成 RHACS init 捆绑包。
- 您必须有权访问 Red Hat Container Registry 和一个 pull secret 进行身份验证。有关从 **registry.redhat.io** 下载镜像的详情，请参考 Red Hat Container Registry Authentication 。
- 您必须有用于公开 Central 服务的地址和端口号。

流程

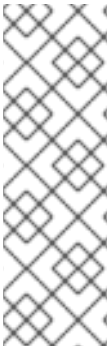
- 运行以下命令：

```
$ 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
  --set imagePullSecrets.username=<username> \ 2
  --set imagePullSecrets.password=<password> 3
```

1 使用 **-f** 选项指定 YAML 配置文件的路径。

2 为 Red Hat Container Registry 身份验证包含 pull secret 的用户名。

3 包括 Red Hat Container Registry 身份验证的 pull secret 密码。



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。

其他资源

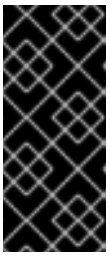
- 在其他平台上为 RHACS 生成并应用 init 捆绑包

5.4.1.3. 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。

当使用 **helm upgrade** 命令进行修改时，会应用以下准则和要求：

- 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，每当您进行更改时，您必须手动指定所有选项。
- 有些更改（如启用 Scanner V4）需要为组件发布新证书。因此，您必须在进行这些更改时提供 CA。



重要

扫描程序 V4 只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅技术预览功能支持范围。

  - 如果 CA 在初始安装过程中由 Helm chart 生成，则必须从集群中检索这些值，并将其提供给 **helm upgrade** 命令。**central-services** Helm Chart 的安装后备注包括用于检索自动生成的值的命令。
  - 如果 CA 在 Helm Chart 之外生成，并在安装 **central-services** chart 时提供，那么您必须在使用 **helm upgrade** 命令时再次执行该操作，例如在 **helm upgrade** 命令中使用 **--reuse-values** 标志。

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 如果您修改了没有包括在 **values_public.yaml** 和 **values_private.yaml** 文件中的值，请包含 **--reuse-values** 参数。

5.4.2. 使用 roxctl CLI 在安全集群中安装 RHACS

要使用 CLI 在安全集群中安装 RHACS，请执行以下步骤：

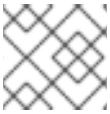
1. 安装 **roxctl** CLI
2. 安装 Sensor。

5.4.2.1. 安装 roxctl CLI

您必须首先下载二进制文件。您可以在 Linux、Windows 或 macOS 上安装 **roxctl**。

5.4.2.1.1. 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。



注意

用于 Linux 的 **roxctl** CLI 可用于 **amd64**、 **ppc64le** 和 **s390x** 架构。

流程

1. 确定目标操作系统的 **roxctl** 架构：

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

2. 下载 **roxctl** CLI：

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

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
```

5.4.2.1.2. 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。

注意

用于 macOS 的 **roxctl** CLI 可用于 **amd64** 架构。

流程

1. 下载 **roxctl** CLI：

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

5.4.2.1.3. 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。

注意

amd64 架构提供了适用于 Windows 的 **roxctl** CLI。

流程

- 下载 **roxctl** CLI：

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

验证

- 验证您已安装的 **roxctl** 版本：

```
$ roxctl version
```

5.4.2.2. 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。此安装方法也称为清单安装方法。

要使用清单安装方法执行安装，*请仅遵循以下流程之一*：

- 使用 RHACS web 门户下载集群捆绑包，然后提取并运行传感器脚本。
- 使用 **roxctl** CLI 为 OpenShift Container Platform 集群生成所需的传感器配置，并将其与您的 Central 实例关联。

先决条件

- 您必须已安装了 Central 服务，也可以在 Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service)上选择 **ACS 实例** 来访问 Central 服务。

5.4.2.2.1. 使用 Web 门户的清单安装方法

流程

1. 在安全集群中，在 RHACS 门户中，进入 **Platform Configuration → Clusters**。
2. 选择 **Secure a cluster → Legacy 安装方法**。
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 **wss://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. 在可以访问被监控的集群的系统中，从集群捆绑包中提取并运行 **sensor** 脚本：

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

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

如果您收到没有部署 Sensor 所需的权限的警告，请按照屏幕说明操作，或联系集群管理员寻求帮助。

部署 Sensor 后，它会联系中心并提供集群信息。

5.4.2.2.2. 使用 roxctl CLI 安装清单

流程

1. 运行以下命令，为 OpenShift Container Platform 集群生成所需的传感器配置，并将其与您的 Central 实例关联：

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

1 对于 **--openshift-version** 选项，请指定集群的主 OpenShift Container Platform 版本号。例如，为 OpenShift Container Platform 版本 **3.x** 指定 **3**，为 OpenShift Container Platform 版本 **4.x** 指定 **4**。

2. 在可以访问被监控的集群的系统中，从集群捆绑包中提取并运行 **sensor** 脚本：

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

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

如果您收到没有部署 Sensor 所需的权限的警告，请按照屏幕说明操作，或联系集群管理员寻求帮助。

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，当在 **Platform Configuration → Clusters** 中查看集群列表时，集群状态会显示一个绿色勾号和 **Healthy** 状态。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 中输入以下命令：

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

- 在 Kubernetes 上，输入以下命令：

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

2. 点 **Finish** 关闭窗口。

安装后，Sensor 开始向 RHACS 报告安全信息，RHACS 门户仪表板开始显示部署、镜像和策略违反情况。

5.5. 在其他平台上验证 RHACS 安装

提供验证 RHACS 是否已正确安装的步骤。

5.5.1. 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情况。

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. 对于负载均衡器：

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

   b. 对于端口转发：
      i. 运行以下命令：

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

      ii. 转至 **https://localhost:18443/**。
2. 新建命名空间：

```
$ kubectl create namespace test
```

3. 使用关键漏洞启动一些应用程序：

```
$ kubectl run shell --labels=app=shellshock,team=test-team \
  --image=quay.io/stackrox-io/docs:example-vulnerables-cve-2014-6271 -n test
$ kubectl run samba --labels=app=rce \
  --image=quay.io/stackrox-io/docs:example-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和策略违反情况。进入 RHACS 门户来查看违反情况。您可以使用默认用户名 **admin** 和生成的密码登录到 RHACS 门户。

第 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 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
```