Red Hat Advanced Cluster Security for Kubernetes 4.4

RHACS 云服务

关于 RHACS 云服务

Last Updated: 2024-06-28

Red Hat Advanced Cluster Security for Kubernetes 4.4 RHACS 云服务

关于 RHACS 云服务

法律通告

摘要

关于了解 RHACS 云服务的指导。

目录

第 1 章 RHACS 云服务描述

1.1. RHACS 简介

Red Hat Advanced Cluster Security for Kubernetes (RHACS)是一个企业级的 Kubernetes 原生容器安全解决方案，可帮助您更安全地构建、部署和运行云原生应用程序。

Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service)提供 Kubernetes 原生安全性作为服务。使用 RHACS 云服务，红帽维护、升级和管理您的中央服务。

中央服务包括用户界面 (UI)、数据存储、RHACS 应用程序编程接口 (API) 和镜像扫描功能。您可以通过 Red Hat Hybrid Cloud Console 部署 Central 服务。 当您创建新的 ACS 实例时，红帽会为 RHACS 创建单独的 control plane。

RHACS 云服务允许您保护与 Central 实例通信的自我管理集群。您安全（称为 Secured Clusters）的集群由您管理，而不是由红帽管理。安全的集群服务包括可选的漏洞扫描服务、准入控制服务以及用于运行时监控和合规性的数据收集服务。您可以在您要保护的任何 OpenShift 或 Kubernetes 集群上安装安全集群服务。

1.2. 账单

客户可以在 Amazon Web Services (AWS) 市场购买 RHACS 云服务订阅。服务成本按安全内核每小时收取，或者属于安全集群的节点的 vCPU。

> **例 1.1. 订阅成本示例**
>
> 如果您已建立到两个安全集群的连接，每个节点都有 5 个与 8 个 vCPU 相同的节点（如 Amazon EC2 m7g.2xlarge），则安全内核的总数为 80 (2 x 5 x 8 = 80)。

1.3. 安全和合规性

Central 实例中的所有 Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service)数据都会在传输中加密。数据与定期调度的备份一起存储在安全存储中，并具有完整复制和高可用性。RHACS 云服务可通过云数据中心获得，以确保最佳性能和满足数据驻留要求的能力。

1.3.1. 身份验证供应商

当使用 Red Hat Hybrid Cloud Console 创建 Central 实例时，集群管理员的身份验证将配置为过程的一部分。客户必须管理对 Central 实例的所有访问，作为其集成解决方案的一部分。有关可用验证方法的更多信息，请参阅了解身份验证供应商。

RHACS 云服务中的默认身份提供程序是 Red Hat Single Sign-On (SSO)。有关使用 Red Hat SSO 进行身份验证的更多信息，请参阅 对 ACS 控制台的默认访问。

1.3.2. 法规合规性

有关最新的规范合规性信息，请参阅了解 OpenShift Dedicated 的进程和安全性。

1.4. 指标和日志记录

1.4.1. 服务指标

服务指标仅供内部使用。红帽在商定的级别提供和维护服务。服务指标只能被授权的红帽人员访问。如需更多信息，请参阅 PRODUCT APPENDIX 4 Red Hat ONLINE SERVICES 。

1.4.2. 客户指标

核心使用容量指标可以通过 Subscription Watch 或 Subscriptions 页面 获得。

1.4.3. 服务日志记录

Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service)的所有组件的系统日志是内部的，且仅适用于红帽员工。红帽不提供对组件日志的用户访问权限。如需更多信息，请参阅 PRODUCT APPENDIX 4 Red Hat ONLINE SERVICES。

1.5. 可扩展性和服务级别

Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service)对它可以保护的内核数或集群设置了限制。限制基于安全集群中可用的资源，以及对可用性的限制。

1.5.1. 服务级别目标和协议

有关服务级别目标(SLO)和服务等级协议(SLA)的更多信息，请参阅 PRODUCT APPENDIX 4 RED HAT ONLINE SERVICES。

1.6. 更新和升级

红帽在对影响服务的更新和升级之前，红帽以合理的努力通知客户。有关对 Central 实例的服务更新所需的决定，其计时是 Red Hat 的唯一责任。

客户无法控制何时对 Central 服务进行升级。如需更多信息，请参阅 PRODUCT APPENDIX 4 Red Hat ONLINE SERVICES。升级到 Red Hat Advanced Cluster Security Cloud Service (RHACS 云服务)的版本被视为服务更新的一部分。

客户负责及时进行 RHACS 安全集群服务升级，需要与 RHACS 云服务保持兼容性。

红帽建议为连接到 RHACS 云服务的安全集群启用自动升级。

有关升级版本的更多信息，请参阅 Red Hat Advanced Cluster Security for Kubernetes Support Matrix 。

1.7. 可用性

可用性和避免出现灾难对于任何安全平台至关重要。Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service)对多个级别的故障提供了大量保护。为了考虑可能的云供应商失败，红帽建立了多个可用区。

1.7.1. 备份和恢复

所有 RHACS 云服务集群都使用数据库备份进行备份。这也适用于存储在 Central 数据库中的客户数据。

所有快照都使用适当的云供应商快照 API 创建，然后上传到安全对象存储，用于 Amazon Web Services (AWS)是一个 S3 存储桶。

- 红帽不提交恢复点目标(RPO)或恢复时间目标(RTO)。如需更多信息，请参阅 PRODUCT APPENDIX 4 Red Hat ONLINE SERVICES。

- 站点可靠性工程仅作为预防措施执行备份。它们存储在与集群相同的区域中。
- 客户应该部署带有 Kubernetes 最佳实践的多个可用区保护集群，以确保区域内的高可用性。

1.7.2. 获得支持

RHACS 云服务包括红帽标准(Standard)和高级(Premium)支持，您可以使用 红帽客户门户网站 访问它们。您可以为产品"Red Hat Advanced Cluster Security Cloud Service"打开支持问题单。

红帽支持响应有限可用性客户提交的支持票据，而 Red Hat Site Reliability 工程师(SRE)会主动监控 Red Hat Advanced Cluster Security Cloud Service (RHACS 云服务)的健康状况。

此外，红帽事业部解决方案架构师(BU SA)在客户专家、红帽支持和红帽 SRE 间充当实践技术，支持用户有限可用性。

- 有关 RHACS 云服务支持涵盖的内容的更多信息，请参阅覆盖范围详情。
- 有关产品支持服务条款的更多信息，请参阅 产品支持条款。

1.7.3. 服务删除

您可以使用 Red Hat Hybrid Cloud Console 的默认删除操作来删除 RHACS 云服务。删除 RHACS Cloud Service Central 实例会自动删除所有 RHACS 组件。删除不可逆。

1.8. 定价

红帽在有限可用性期间为 Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service)收取订阅费用。如需更多信息，请参阅 PRODUCT APPENDIX 4 Red Hat ONLINE SERVICES 。

1.9. 服务等级协议

有关为 Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service)提供的服务级别协议(SLA)的更多信息，请参阅 PRODUCT APPENDIX 4 RED HAT ONLINE SERVICES 。

1.10. 时间表

有限可用性
对 Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service)的产品支持提供有限的客户。如需更多信息，请参阅 PRODUCT APPENDIX 4 Red Hat ONLINE SERVICES 。

公开发行
对于所有 RHACS 云服务客户，都提供了对 RHACS 云服务的产品支持。如需更多信息，请参阅 PRODUCT APPENDIX 4 Red Hat ONLINE SERVICES 。

第 2 章 RED HAT ADVANCED CLUSTER SECURITY CLOUD SERVICE 架构

发现 Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service)架构和概念。

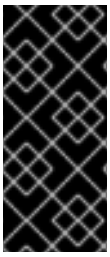
2.1. RED HAT ADVANCED CLUSTER SECURITY CLOUD SERVICE 架构概述

Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service)是一个红帽管理的软件即服务(SaaS)平台，可让您在构建、部署和运行时生命周期中保护 Kubernetes 和 OpenShift Container Platform 集群和应用程序。

RHACS 云服务包括许多内置的 DevOps 实施控制和以安全为中心的最佳实践，具体取决于行业标准，如互联网安全中心(CIS)基准和国家标准技术(NIST)指南。您还可以将其与现有 DevOps 工具和工作流集成，以提高安全性和合规性。

RHACS 云服务架构

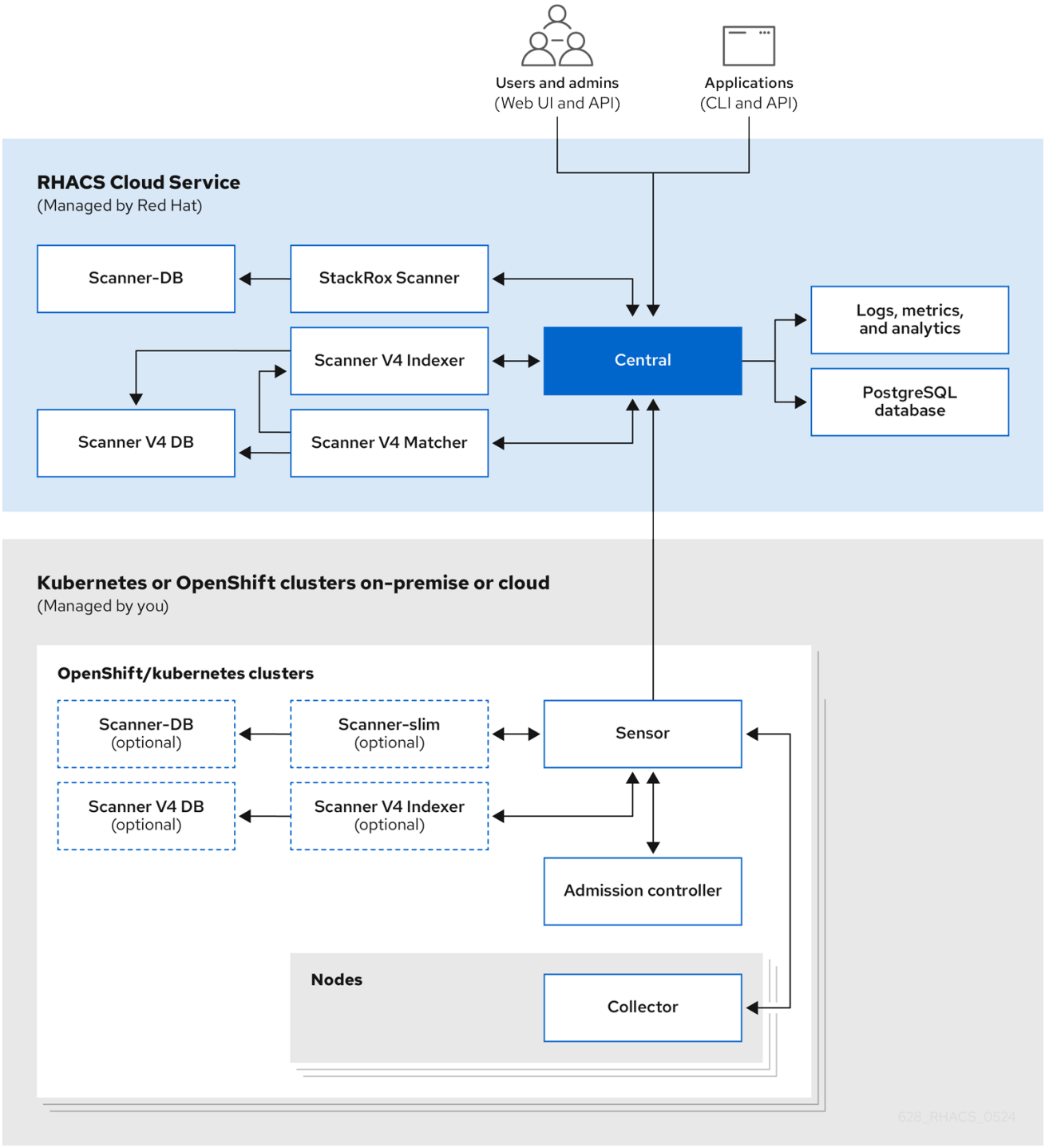
下图显示了带有 StackRox Scanner 和 Scanner V4 的架构，对于版本 4.4 是技术预览。Scanner V4 的安装是可选的，但提供了额外的优点。



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。



中央服务包括用户界面 (UI)、数据存储、RHACS 应用程序编程接口 (API) 和镜像扫描功能。您可以通过 Red Hat Hybrid Cloud Console 部署 Central 服务。 当您创建新的 ACS 实例时，红帽会为 RHACS 创建单独的 control plane。

RHACS 云服务允许您保护与 Central 实例通信的自我管理集群。您安全（称为 Secured Clusters）的集群由您管理，而不是由红帽管理。安全的集群服务包括可选的漏洞扫描服务、准入控制服务以及用于运行时监控和合规性的数据收集服务。您可以在您要保护的任何 OpenShift 或 Kubernetes 集群上安装安全集群服务。

2.2. CENTRAL

红帽管理 Central，即 RHACS 云服务的 control plane。这些服务包括以下组件：

- **Central** ： Central 是 RHACS 应用程序管理界面和服务。它处理 API 交互和用户界面 (RHACS Portal) 访问。
- **Central DB** ： Central DB 是 RHACS 的数据库，并处理所有数据持久性。它目前基于 PostgreSQL 13。
- scanner **V4** （技术预览）：从版本 4.4 开始，RHACS 包含扫描程序 V4 漏洞扫描程序来扫描容器镜像。扫描程序 V4 基于 ClairCore 构建，同时还支持 Clair 扫描程序。scanner V4 包括 Indexer、Matcher 和 Scanner V4 DB 组件，它们用于扫描。
- **stackrox Scanner**: StackRox Scanner 是 RHACS 中的默认扫描程序。StackRox 扫描程序源自 Clair v2 开源扫描程序的分叉。
- **scanner-DB** ：此数据库包含 StackRox Scanner 的数据。

RHACS 扫描程序会分析每个镜像层，以确定基础操作系统，并确定操作系统软件包管理器安装的编程语言软件包和软件包。它们与来自各种漏洞来源的已知漏洞匹配。另外，StackRox Scanner 会识别节点的操作系统和平台中的漏洞。这些功能计划在以后的版本中为 Scanner V4。

2.2.1. 漏洞源

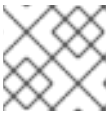
RHACS 使用以下漏洞源：

- alpine 安全数据库
- Amazon Linux 安全中心跟踪的数据
- Debian 安全跟踪器
- Oracle OVAL
- Photon OVAL
- Red Hat OVAL
- Red Hat CVE Map ：这用于 Red Hat Container Catalog 中显示的镜像。
- SUSE OVAL
- Ubuntu OVAL
- OSV ：这用于与语言相关的漏洞，如 Go、Java、Node.js (JavaScript)、Python 和 Ruby。这个源可能会为漏洞提供 GitHub 安全公告(GHSA) ID 而不是 CVE 号。

注意

RHACS Scanner V4 使用此许可证的 OSV.dev 上可用的 OSV 数据库。https://github.com/google/osv.dev/blob/master/LICENSE

- NVD ：这用于各种目的，如在供应商不提供信息时填补信息差距。例如，Alpine 不提供描述、CVSS 分数、严重性或发布日期。



注意

此产品使用 NVD API，但不由 NVD 结束或认证。

- stackrox: 上游 StackRox 项目维护一组漏洞，这些漏洞可能会因为来自其他源的数据格式或数据不存在而被发现。

Scanner V4 Indexer 使用以下源：

- repository-to-cpe.json ：将 RPM 存储库映射到其相关的 cps，这是匹配基于 RHEL 的镜像的漏洞所必需的。
- container-name-repos-map.json ：这与提供它们的存储库匹配。

2.3. 安全的集群服务

您可以使用 RHACS Cloud Service 在您要保护的每个集群中安装安全集群服务。安全的集群服务包括以下组件：

- **Sensor** ：传感器是负责分析和监控集群的服务。Sensor 侦听 OpenShift Container Platform 或 Kubernetes API 和 Collector 事件来报告集群的当前状态。Sensor 还根据 RHACS 云服务策略触发部署时间和运行时违反情况。另外，Sensor 负责所有集群交互，如应用网络策略、启动 RHACS 云服务策略的重新处理以及与 Admission 控制器交互。
- **准入控制器** ： Admission 控制器可防止用户创建在 RHACS 云服务中违反安全策略的工作负载。
- **Collector** ：收集器分析和监控集群节点上的容器活动。它收集容器运行时和网络活动信息，并将收集的数据发送到 Sensor。
- **stackrox Scanner 和 Scanner V4** （技术预览）：在 Kubernetes 中，安全集群服务包括 Scanner-slim 作为可选组件。但是，在 OpenShift Container Platform 上，RHACS 云服务在每个安全集群中安装 Scanner-slim 版本，以便在 OpenShift Container Platform 集成 registry 和其他 registry 中扫描镜像。

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**: Scanner V4 Indexer 执行镜像索引，之前被称为镜像分析。根据镜像和 registry 凭证，索引程序会从 registry 中拉取镜像。它找到基础操作系统（如果存在），并查找软件包。它存储和输出索引报告，其中包含给定镜像的查找。
  - **扫描程序 V4 DB** ：如果启用了 Scanner V4，则安装此组件。此数据库存储扫描程序 V4 的信息，包括索引报告。为获得最佳性能，请为 Scanner V4 DB 配置持久性卷声明(PVC)。
  - **scanner-DB** ：此数据库包含 StackRox Scanner 的数据。

注意

当在与 **central -services 相同的集群中安装 secured** -cluster-services 并在同一个命名空间中安装时，**secured-cluster-services** 不会部署 Scanner V4 组件。相反，假设 **central-services** 已包含 Scanner V4 的部署。

其他资源

- 外部组件

2.4. 数据访问和权限

红帽无法访问在其中安装安全集群服务的集群。实际上，RHACS 云服务并不需要访问安全集群的权限。例如，您不需要创建一个新的 IAM 策略、访问角色或 API 令牌。

但是，RHACS 云服务会存储安全集群服务发送的数据。所有数据都在 RHACS 云服务中加密。在 RHACS 云服务平台中加密数据有助于确保数据的保密性和完整性。

当您在集群中安装安全集群服务时，它会生成数据并将其传送到 RHACS 云服务。此数据在 RHACS 云服务平台内保持安全，只有授权的 SRE 团队成员和系统才能访问这些数据。RHACS 云服务使用此数据来监控集群和应用程序的安全性和合规性，并提供有用的分析和分析，以帮助您优化部署。

第 3 章 RHACS 云服务入门

Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service)为您的 Red Hat OpenShift 和 Kubernetes 集群提供安全服务。有关安全集群支持的平台的更多信息，请参阅 Red Hat Advanced Cluster Security for Kubernetes Support Matrix。

先决条件

- 确保您可以从 Red Hat Hybrid Cloud Console 访问 **Advanced Cluster Security** 菜单选项。

注意

要访问 RHACS Cloud Service 控制台，您需要 Red Hat Single Sign-On (SSO)凭证，如果配置了其他身份提供程序的凭证。请参阅 对 ACS 控制台的默认访问权限。

3.1. 安装步骤的高级概述

以下小节概述了安装步骤和相关文档的链接。

3.1.1. 保护 Red Hat OpenShift 集群

要使用 Operator 保护 Red Hat OpenShift 集群，请执行以下步骤：

1. 验证您要保护的集群 是否满足要求。
2. 在 Red Hat Hybrid Cloud 控制台中，创建一个 **ACS 实例**。
3. 在您要保护的每个 Red Hat OpenShift 集群中，创建一个名为 **stackrox**的项目。此项目将包含 RHACS 云服务安全集群的资源。
4. 在 ACS 控制台中，创建一个 init 捆绑包。init 捆绑包包含允许 RHACS 云服务安全集群和 ACS 控制台通信的 secret。
5. 在每个 Red Hat OpenShift 集群中，使用它来创建资源 来应用 init 捆绑包。
6. 在每个 Red Hat OpenShift 集群上，安装 RHACS Operator。
7. 在每个 Red Hat OpenShift 集群中，使用 Operator 在 **stackrox** 项目中创建安全 集群资源。
8. 通过确保安全集群可以与 ACS 实例通信来验证安装。???

要使用 Helm chart 或 **roxctl** CLI 保护 Red Hat OpenShift 集群，请执行以下步骤：

1. 验证您要保护的集群 是否满足要求。
2. 在 Red Hat Hybrid Cloud 控制台中，创建一个 **ACS 实例**。
3. 在您要保护的每个 Red Hat OpenShift 集群中，创建一个名为 **stackrox**的项目。此项目将包含 RHACS 云服务安全集群的资源。
4. 在 ACS 控制台中，创建一个 init 捆绑包。init 捆绑包包含允许 RHACS 云服务安全集群和 ACS 控制台通信的 secret。
5. 在每个 Red Hat OpenShift 集群中，使用它来创建资源 来应用 init 捆绑包。

6. 在每个 Red Hat OpenShift 集群中，使用 Helm chart 或 **roxctl** CLI 在 **stackrox** 集群中安装安全集群资源。

7. 通过确保安全集群可以与 ACS 实例通信来验证安装。???

3.1.2. 保护 Kubernetes 集群

要保护 Kubernetes 集群，请执行以下步骤：

1. 验证您要保护的集群 是否满足要求。

2. 在 Red Hat Hybrid Cloud 控制台中，创建一个 **ACS 实例**。

3. 在 ACS 控制台中，创建一个 init 捆绑包。init 捆绑包包含允许 RHACS 云服务安全集群和 ACS 控制台通信的 secret。

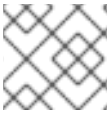
4. 在每个 Kubernetes 集群中，使用它创建资源 应用 init 捆绑包。

5. 在每个 Kubernetes 集群中，使用 Helm chart 或 **roxctl** CLI 安装安全 集群资源。

6. 通过确保安全集群可以与 ACS 实例通信来验证安装。???

3.2. 对 ACS 控制台的默认访问权限

默认情况下，用户可用的身份验证机制是使用 Red Hat Single Sign-On (SSO)进行身份验证。您不能删除或更改 Red Hat SSO 身份验证提供程序。但是，您可以更改最小访问角色并添加额外规则，或者添加其他身份提供程序。



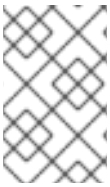
注意

要了解身份验证供应商如何在 ACS 中工作，请参阅了解身份验证供应商。

为每个 ACS 控制台创建一个 **sso.redhat.com** 的专用 OIDC 客户端。所有 OIDC 客户端共享相同的 **sso.redhat.com** 域。**sso.redhat.com** 发布的令牌中的声明映射到 ACS 发布的令牌，如下所示：

- **realm_access.roles** 到 **groups**
- **org_id** to **rh_org_id**
- **is_org_admin** to **rh_is_org_admin**
- **sub** 到 **userid**

内置的 Red Hat SSO 身份验证提供程序将所需的属性 **rh_org_id** 设置为分配给创建 RHACS 云服务实例的用户的机构 ID。这是用户所属的机构帐户的 ID。这可以被视为用户所具备的"租户"，并且归其所有。只有具有相同组织帐户的用户才能使用 Red Hat SSO 身份验证提供程序访问 ACS 控制台。



注意

要更好地控制对 ACS 控制台的访问权限，请配置另一个身份提供程序，而不依赖于 Red Hat SSO 身份验证提供程序。如需更多信息，请参阅了解身份验证供应商。要将其他身份验证提供程序配置为登录页面上的第一个身份验证选项，其名称应小于 **Red Hat SSO**。

最小访问角色设置为 **None**。为此字段分配不同的值，可以将 RHACS Cloud Service 实例访问到具有相同机构帐户的所有用户。

在内置 Red Hat SSO 身份验证提供程序中设置的其他规则包括：

- 将 **userid** 映射到 **Admin**的规则
- 将机构的管理员映射到 **Admin**的规则

您可以添加更多规则，将 ACS 控制台的访问权限授予具有相同机构帐户的人员。例如，您可以使用 **email** 作为密钥。

第 4 章 RED HAT ADVANCED CLUSTER SECURITY CLOUD SERVICE 的默认资源要求

4.1. RHACS 云服务的常规要求

在安装 Red Hat Advanced Cluster Security Cloud Service 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 云服务：

- 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 云服务，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。

以下最低要求和建议适用于集群节点。

架构

支持的构架有 **amd64**、**ppc64 le**、或 **s390x**。

注意

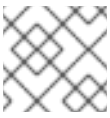
IBM Power (**ppc64le**)、IBM Z (**s390x**)和 IBM® LinuxONE (**s390x**)集群支持安全集群服务。

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。

Storage

对于 RHACS 云服务，不需要持久性卷声明(PVC)。但是，如果您启用了 Scanner V4 的安全集群，则强烈建议使用 PVC。使用固态驱动器(SSD)以获得最佳性能。但是，如果您没有 SSD，也可以使用另一个存储类型。

重要

您不能将 Ceph FS 存储与 RHACS 云服务一起使用。红帽建议将 RBD 块模式 PVC 用于 RHACS 云服务。

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

- 如果您要使用 Helm chart 安装和配置 RHACS 云服务，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 。

4.2. 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

4.2.1. 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

4.2.2. 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
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

4.2.3. 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 4.1. 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

类型	CPU	内存
限制	1 个内核	2000 MiB

node-inventory 容器

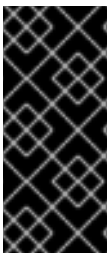
类型	CPU	内存
Request（请求）	0.01 个内核	10 MiB
限制	1 个内核	500 MiB

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

4.2.4. 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

扫描程序 V4 DB	CPU	内存
限制	2 个内核	4 GiB

第 5 章 推荐的 RED HAT ADVANCED CLUSTER SECURITY CLOUD SERVICE 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

在分析结果期间，部署数量被识别为增加使用资源的主要因素。部署数量用于估算所需资源。

其他资源

- 默认资源要求

5.1. 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。

5.1.1. 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

5.1.2. 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 6 章 使用 RED HAT OPENSHIFT 安全集群设置 RHACS 云服务

6.1. 在 RED HAT CLOUD 上创建 RHACS 云实例

通过在 Red Hat Hybrid Cloud Console 中选择一个实例来访问 Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service)。**ACS 实例** 包含红帽为您配置和管理的 RHACS 云服务管理界面和服务。管理界面连接到您的安全集群，其中包含扫描的服务并收集有关漏洞的信息。一个实例可以连接到并监控多个集群。

6.1.1. 在控制台中创建实例

在 Red Hat Hybrid Cloud 控制台中，创建一个 **ACS 实例** 以连接到您的安全集群。

流程

创建 **ACS 实例** ：

1. 登录到 Red Hat Hybrid Cloud 控制台。
2. 在导航菜单中选择 **Advanced Cluster Security → ACS Instances**。
3. 选择 **Create ACS 实例** 并在显示字段中输入信息，或者从下拉列表中选择适当的选项：
   - **名称** ：输入 **ACS 实例的名称**。**ACS 实例** 包含 RHACS Central 组件，也称为 "Central"，其中包括由红帽配置和管理的 RHACS 云服务管理界面和服务。您管理与 Central 通信的安全集群。您可以将多个安全集群连接到一个实例。
   - **云供应商** ： Central 所在的云供应商。选择 **AWS**。
   - **Cloud region** ： Central 所在的云供应商的区域。选择以下区域之一：
     - US-East, N. Virginia
     - 欧洲、爱尔兰
   - **可用区** ： 使用默认值(**多**)。
4. 单击 **Create instance**。

6.1.2. 后续步骤

- 在您要保护的每个 Red Hat OpenShift 集群中，创建一个名为 **stackrox**的项目。此项目将包含 RHACS 云服务安全集群的资源。

6.2. 在 RED HAT OPENSHIFT 安全集群中创建项目

在您要保护的每个 Red Hat OpenShift 集群上创建一个项目。然后，您可以使用此项目使用 Operator 或 Helm chart 安装 RHACS Cloud Service 资源。

6.2.1. 在集群中创建项目

流程

- 在 OpenShift Container Platform 集群中，进入 **Home → Projects** 并为 RHACS Cloud Service 创建一个项目。使用 **stackrox** 作为项目名称。

6.2.2. 后续步骤

- 在 ACS 控制台中，创建一个 init 捆绑包。init 捆绑包包含允许 RHACS 云服务安全集群和 ACS 控制台通信的 secret。

6.3. 为安全集群生成 INIT 捆绑包

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

6.3.1. 生成 init 捆绑包

6.3.1.1. 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。

6.3.1.2. 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。

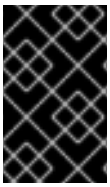
先决条件

- 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>
```



重要

在 RHACS Cloud Service 中，当使用需要 Central 地址的 **roxctl** 命令时，请使用 Red Hat Hybrid Cloud Console 的 **Instance Details** 部分显示的 **Central 实例地址**。例如，使用 **acs-ABCD12345.acs.rhcloud.com** 而不是 **acs-data-ABCD12345.acs.rhcloud.com**。

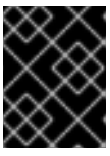
流程

- 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。

6.3.2. 后续步骤

- 使用 init 捆绑包创建资源

6.4. 为安全集群应用 INIT 捆绑包

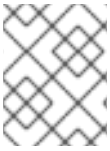
使用它应用 init 捆绑包来创建资源。

注意

您必须具有 **Admin** 用户角色才能应用 init 捆绑包。

6.4.1. 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 捆绑包以便在安全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 RHACS 云服务通信。



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。

验证

- 重启 Sensor 以获取新证书。
  有关如何重启 Sensor 的更多信息，请参阅"添加资源"部分中的"重启 Sensor 容器"。

6.4.2. 后续步骤

- 在每个 Red Hat OpenShift 集群上，安装 RHACS Operator。
- 在您要监控的所有集群中安装 RHACS 安全集群服务。

6.4.3. 其他资源

- 重启 Sensor 容器

6.5. 安装 OPERATOR

在安全集群中安装 RHACS Operator。

6.5.1. 为 RHACS 云服务安装 RHACS Operator

使用 OpenShift Container Platform 提供的 OperatorHub 是安装 RHACS Operator 的最简单方法。

先决条件

- 您可以使用具有 Operator 安装权限的账户访问 OpenShift Container Platform 集群。
- 您必须使用 OpenShift Container Platform 4.11 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 。

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，则当有新版 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。

     红帽建议在 RHACS 云服务中为 Operator 启用自动升级。如需更多信息，请参阅 Red Hat Advanced Cluster Security for Kubernetes Support Matrix。
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**。

6.5.2. 后续步骤

- 在每个 Red Hat OpenShift 集群中，在 **stackrox** 项目中安装安全集群资源。

6.6. 从 RHACS 云服务安装安全集群资源

您可以使用 Operator 或 Helm chart 在安全集群中安装 RHACS 云服务。您还可以使用 **roxctl** CLI 安装它，但不要使用这个方法，除非您有需要使用它的特定安装需要。

先决条件

- 您已创建了 Red Hat OpenShift 集群，并在其上安装 Operator。
- 在 RHACS 云服务中的 ACS 控制台中，您已创建并下载 init 捆绑包。
- 您可以使用 **oc create** 命令应用 init 捆绑包。
- 在安装过程中，您记下 **了 Central API 端点**，包括地址和端口号。您可以从云控制台导航菜单中选择 **Advanced Cluster Security → ACS Instances** 来查看此信息，然后点您创建的 ACS 实例。

6.6.1. 使用 Operator 在安全集群中安装 RHACS

6.6.1.1. 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**。

   - 对于 RHACS 云服务，请使用 **中央 API 端点**，包括地址和端口号。您可以从云控制台导航菜单中选择 **Advanced Cluster Security → ACS Instances** 来查看此信息，然后点您创建的 ACS 实例。

   - *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。

6.6.2. 使用 Helm chart 在安全集群中安装 RHACS 云服务

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。

首先，确保添加 Helm Chart 仓库。

6.6.2.1. 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/
```

6.6.2.2. 使用 Helm chart 在安全集群中安装 RHACS 云服务

6.6.2.2.1. 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

- 在基于 Kubernetes 的集群上运行以下命令：

```
$ helm install -n stackrox --create-namespace \
  stackrox-secured-cluster-services rhacs/secured-cluster-services \
  -f <path_to_cluster_init_bundle.yaml> \ 1
  -f <path_to_pull_secret.yaml> \ 2
  --set clusterName=<name_of_the_secured_cluster> \
  --set centralEndpoint=<endpoint_of_central_service> 3
  --set imagePullSecrets.username=<your redhat.com username> \ 4
  --set imagePullSecrets.password=<your redhat.com password> 5
```

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

- 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

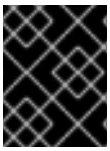
- 为安全集群生成 init 捆绑包
- 为安全集群应用 init 捆绑包

6.6.2.3. 使用自定义配置 secured-cluster-services Helm chart

您可以将 Helm Chart 配置参数与 **helm install** 和 **helm upgrade** 命令一起使用。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

使用 **secured-cluster-services** Helm Chart 时，请不要更改属于 chart 的 **values.yaml** 文件。

6.6.2.3.1. 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。
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。

参数	Description
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。
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。

参数	Description
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。
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。

参数	Description
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。
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。

参数	Description
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。
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。

参数	Description
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**。

6.6.2.3.1.1. 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。

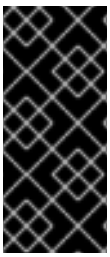
配置是分层的，在更通用范围（例如，所有对象）中定义的元数据被覆盖为更通用范围的元数据（例如，仅适用于 Sensor 部署）。

6.6.2.3.2. 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

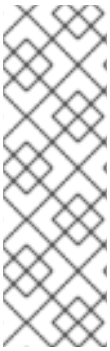
- 您必须已为集群生成 RHACS init 捆绑包。
- 您必须有权访问 Red Hat Container Registry 和一个 pull secret 进行身份验证。有关从 **registry.redhat.io** 下载镜像的详情，请参考 Red Hat Container Registry Authentication 。
- 您必须有用于公开 Central 服务的地址和端口号。

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

- 为安全集群生成 init 捆绑包

- 为安全集群应用 init 捆绑包

6.6.2.4. 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。

6.6.3. 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。

6.6.3.1. 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**。

6.6.3.1.1. 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
```

6.6.3.1.2. 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
```

6.6.3.1.3. 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
```

6.6.3.2. 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。

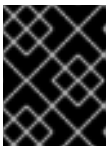
先决条件

- 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。

6.6.3.2.1. 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。

6.6.3.2.2. 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。

6.6.4. 后续步骤

- 通过确保安全集群可以与 ACS 实例通信来验证安装。???

6.7. 在 RHACS 云服务中为安全集群服务配置代理

您必须在 Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service)环境中为安全集群服务配置代理设置，以便在安全集群和指定代理服务器之间建立连接。这样可确保可靠的数据收集和传输。

6.7.1. 在 SecuredCluster CR 中指定环境变量

要配置出口代理，您可以使用集群范围的 Red Hat OpenShift 代理，或者在 SecuredCluster 自定义资源 (CR)配置文件中指定 **HTTP_PROXY**、**HTTPS_PROXY** 和 **NO_PROXY** 环境变量，以确保代理正确使用代理并绕过指定域中的内部请求。

代理配置适用于所有运行的服务：Sensor、Collector、Admission Controller 和 Scanner。

流程

- 在 SecuredCluster CR 配置文件中的自定义规格中指定 **HTTP_PROXY**、**HTTPS_PROXY** 和 **NO_PROXY** 环境变量：
  例如：

```
# proxy collector
customize:
  envVars:
    - name: HTTP_PROXY
      value: http://egress-proxy.stackrox.svc:xxxx 1
    - name: HTTPS_PROXY
      value: http://egress-proxy.stackrox.svc:xxxx 2
    - name: NO_PROXY
      value: .stackrox.svc 3
```

1. 变量 **HTTP_PROXY** 设置为值 **http://egress-proxy.stackrox.svc:xxxx**。这是用于 HTTP 连接的代理服务器。
2. 变量 **HTTPS_PROXY** 设置为值 **http://egress-proxy.stackrox.svc:xxxx**。这是用于 HTTPS 连接的代理服务器。
3. 变量 **NO _PROXY** 设置为 **.stackrox.svc**。此变量用于定义不应通过代理服务器访问的主机名或 IP 地址。

6.8. 验证安全集群的安装

安装 RHACS 云服务后，您可以执行一些步骤来验证安装是否成功。

要验证安装，请从 Red Hat Hybrid Cloud 控制台访问 ACS 控制台。仪表板显示 RHACS 云服务监控的集群数量，以及节点、部署、镜像和违反情况的信息。

如果没有数据出现在 ACS 控制台中：

- 确保至少有一个安全集群连接到 RHACS 云服务实例。如需更多信息，请参阅从 RHACS 云服务安装安全集群资源。
- 检查 Sensor pod 日志，以确保与 RHACS 云服务实例的连接成功。
- 在 Red Hat OpenShift 集群中，进入 **Platform Configuration → Clusters** 来验证组件是否健康并查看额外的操作信息。
- 检查本地集群的 Operator 中的 **SecuredCluster** API 中的值，以确保正确输入了 **Central API 端点**。这个值应该与 Red Hat Hybrid Cloud Console 中的 **ACS 实例** 详情中所示的值相同。

第 7 章 使用 KUBERNETES 安全集群设置 RHACS 云服务

7.1. 为 KUBERNETES 集群创建 RHACS 云服务实例

通过在 Red Hat Hybrid Cloud Console 中选择一个实例来访问 Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service)。**ACS 实例** 包含红帽为您配置和管理的 RHACS 云服务管理界面和服务。管理界面连接到您的安全集群，其中包含扫描的服务并收集有关漏洞的信息。一个实例可以连接到并监控多个集群。

7.1.1. 在控制台中创建实例

在 Red Hat Hybrid Cloud 控制台中，创建一个 **ACS 实例** 以连接到您的安全集群。

流程

创建 **ACS 实例** ：

1. 登录到 Red Hat Hybrid Cloud 控制台。
2. 在导航菜单中选择 **Advanced Cluster Security → ACS Instances**。
3. 选择 **Create ACS 实例** 并在显示字段中输入信息，或者从下拉列表中选择适当的选项：
   - **名称** ：输入 **ACS 实例的名称**。**ACS 实例** 包含 RHACS Central 组件，也称为 "Central"，其中包括由红帽配置和管理的 RHACS 云服务管理界面和服务。您管理与 Central 通信的安全集群。您可以将多个安全集群连接到一个实例。
   - **云供应商** ： Central 所在的云供应商。选择 **AWS**。
   - **Cloud region** ： Central 所在的云供应商的区域。选择以下区域之一：
     - US-East, N. Virginia
     - 欧洲、爱尔兰
   - **可用区** ： 使用默认值(**多**)。
4. 单击 **Create instance**。

7.1.2. 后续步骤

- 在您要保护的每个 Kubernetes 集群中，使用 Helm chart 或 **roxctl** CLI 安装安全 集群资源。

7.2. 为 KUBERNETES 安全集群生成 INIT 捆绑包

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

7.2.1. 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。

7.2.2. 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>
```

重要

在 RHACS Cloud Service 中，当使用需要 Central 地址的 **roxctl** 命令时，请使用 Red Hat Hybrid Cloud Console 的 **Instance Details** 部分显示的 **Central 实例地址**。例如，使用 **acs-ABCD12345.acs.rhcloud.com** 而不是 **acs-data-ABCD12345.acs.rhcloud.com**。

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。

7.2.3. 后续步骤

- 使用 init 捆绑包创建资源

7.3. 为 KUBERNETES 安全集群应用 INIT 捆绑包

使用它应用 init 捆绑包来创建资源。

7.3.1. 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 捆绑包以便在安全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 RHACS 云服务通信。

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

- 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**。

验证

- 重启 Sensor 以获取新证书。
  有关如何重启 Sensor 的更多信息，请参阅"添加资源"部分中的"重启 Sensor 容器"。

7.3.2. 后续步骤

- 在您要监控的所有集群中安装 RHACS 安全集群服务。

7.3.3. 其他资源

- 重启 Sensor 容器

7.4. 从 KUBERNETES 集群上的 RHACS 云服务安装安全集群服务

您可以使用以下方法之一在安全集群中安装 RHACS 云服务：

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

7.4.1. 使用 Helm chart 在安全集群中安装 RHACS 云服务

您可以使用没有自定义的 Helm chart 在安全集群中安装 RHACS，方法是使用带有默认值的 Helm chart，或使用带有自定义配置参数的 Helm chart。

首先，确保添加 Helm Chart 仓库。

7.4.1.1. 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：

- 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/
```

7.4.1.2. 使用 Helm chart 在安全集群中安装 RHACS 云服务

7.4.1.2.1. 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 。
- 您必须具有 **Central API 端点**，包括地址和端口号。您可以从云控制台导航菜单中选择 **Advanced Cluster Security → ACS Instances** 来查看此信息，然后点您创建的 ACS 实例。

流程

- 在基于 Kubernetes 的集群上运行以下命令：

```
$ helm install -n stackrox --create-namespace \
  stackrox-secured-cluster-services rhacs/secured-cluster-services \
  -f <path_to_cluster_init_bundle.yaml> \ 1
  -f <path_to_pull_secret.yaml> \ 2
  --set clusterName=<name_of_the_secured_cluster> \
  --set centralEndpoint=<endpoint_of_central_service> 3
  --set imagePullSecrets.username=<your redhat.com username> \ 4
  --set imagePullSecrets.password=<your redhat.com password> 5
```

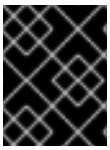
1. 使用 **-f** 选项指定 init 捆绑包的路径。
2. 使用 -f 选项指定 Red Hat Container Registry 身份验证的 pull secret 的路径。
3. 输入 Central API 端点，包括地址和端口号。您可以通过选择 Advanced Cluster Security → ACS 实例，然后点您创建的 ACS 实例，在 Red Hat Hybrid Cloud Console 控制台中再次查看此信息。
4. 为 Red Hat Container Registry 身份验证包含 pull secret 的用户名。
5. 包括 Red Hat Container Registry 身份验证的 pull secret 密码。

7.4.1.3. 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。

7.4.1.3.1. 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。

参数	Description
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。
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。

参数	Description
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。
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。

参数	Description
admissionControl.resources.requests.memory	Admission Control 容器的内存请求。使用此参数覆盖默认值。
admissionControl.resources.requests.cpu	Admission Control 容器的 CPU 请求。使用此参数覆盖默认值。
admissionControl.resources.limits.memory	Admission Control 容器的内存限值。使用此参数覆盖默认值。
admissionControl.resources.limits.cpu	Admission Control 容器的 CPU 限制。使用此参数覆盖默认值。
admissionControl.nodeSelector	将节点选择器标签指定为 **label-key: label-value**，以强制 Admission Control 仅调度到具有指定标签的节点。
admissionControl.tolerations	如果节点选择器选择污点节点，请使用此参数指定污点容限键、值以及 Admission Control 的效果。此参数主要用于基础架构节点。
admissionControl.serviceTLS.cert	Admission Control 使用的内部服务到服务的 TLS 证书。
admissionControl.serviceTLS.key	Admission Control 使用的内部服务对服务的 TLS 证书密钥。
registryOverride	使用此参数覆盖默认的 **docker.io** registry。如果使用其他 registry，请指定 registry 的名称。
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。
scanner.autoscaling.minReplicas	自动扩展的最小副本数。默认值为 2。

参数	Description
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**。

7.4.1.3.1.1. 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"
```

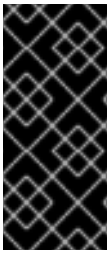
通过 **customize** 设置，您可以为此 Helm Chart 创建的所有对象指定自定义 Kubernetes 元数据（标签和注解）以及工作负载的其他 pod 标签、Pod 注解和容器环境变量。

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）。

7.4.1.3.2. 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 。
- 您必须具有 **Central API 端点**，包括地址和端口号。您可以从云控制台导航菜单中选择 **Advanced Cluster Security → ACS Instances** 来查看此信息，然后点您创建的 ACS 实例。

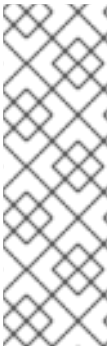
流程

- 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
```

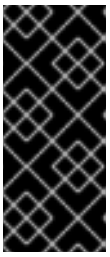
1 如果您使用 base64 编码变量，请使用 **helm install ... -f <(echo "$INIT_BUNDLE_YAML_SECRET" | base64 --decode)** 命令。

7.4.1.4. 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。

7.4.2. 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。

7.4.2.1. 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**。

7.4.2.1.1. 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
```

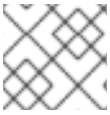
验证

- 验证您已安装的 **roxctl** 版本：

```
$ roxctl version
```

7.4.2.1.2. 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
```

7.4.2.1.3. 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
```

7.4.2.2. 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。

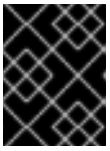
先决条件

- 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。

7.4.2.2.1. 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。
   - 输入 Central API 端点，包括地址和端口号。您可以通过选择 **Advanced Cluster Security → ACS Instances**，然后点您创建的 ACS 实例，在 Red Hat Hybrid Cloud Console 中再次查看此信息。
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。

7.4.2.2.2. 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 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。

7.5. 验证安全集群的安装

安装 RHACS 云服务后，您可以执行一些步骤来验证安装是否成功。

要验证安装，请从 Red Hat Hybrid Cloud 控制台访问 ACS 控制台。仪表板显示 RHACS 云服务监控的集群数量，以及节点、部署、镜像和违反情况的信息。

如果没有数据出现在 ACS 控制台中：

- 确保至少有一个安全集群连接到 RHACS 云服务实例。如需更多信息，请参阅使用 Helm chart 或 **roxctl** CLI 安装的说明。
- 检查 Sensor pod 日志，以确保与 RHACS 云服务实例的连接成功。
- 检查本地集群的 Operator 中的 **SecuredCluster** API 中的值，以确保正确输入了 **Central API 端点**。这个值应该与 Red Hat Hybrid Cloud Console 中的 **ACS 实例** 详情中所示的值相同。