Red Hat Advanced Cluster Security for Kubernetes 4.4

操作

操作 Red Hat Advanced Cluster Security for Kubernetes

Last Updated: 2024-06-28

Red Hat Advanced Cluster Security for Kubernetes 4.4 操作

操作 Red Hat Advanced Cluster Security for Kubernetes

法律通告

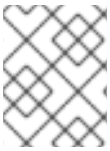
摘要

本文档论述了如何在 Red Hat Advanced Cluster Security for Kubernetes 中执行常见操作任务，包括使用仪表板、管理合规、评估安全风险、管理安全策略和网络策略、检查镜像是否有漏洞并响应违反情况。

目录

第 1 章 它提供了一些额外的易于过滤和定制的导航快捷方式和可操作的小部件，以便您可以专注于重要的数据。

Red Hat Advanced Cluster Security for Kubernetes (RHACS)仪表板可让您快速访问您需要的数据。它提供了一些额外的易于过滤和定制的导航快捷方式和可操作的小部件，以便您可以专注于重要的数据。您可以查看环境中的风险级别、合规状态、策略违反以及镜像中常见漏洞和暴露 (CVE) 的信息。



注意

当您第一次打开 RHACS 门户时，仪表板可能会为空。在至少一个集群中部署 Sensor 后，仪表板会反映您的环境的状态。

以下小节描述了 Dashboard 组件。

1.1. 状态栏

Status Bar 为关键资源提供一览数字计数器。计数器反映了您当前由与用户配置文件关联的角色定义的当前访问范围可见的内容。您可以点这些计数器，快速访问所需列表视图页面，如下所示：

计数	目的地
集群	Platform Configuration → Clusters
节点	Configuration Management → Application & Infrastructure → Nodes
违反情况	违反主菜单
部署	Configuration Management → Application & Infrastructure → Deployments
镜像	Vulnerability Management → Dashboard → Images
Secrets	Configuration Management → Application & Infrastructure → Secrets

1.2. 仪表板过滤器

控制面板包含一个顶级过滤器，可同时应用到所有小部件。您可以选择一个或多个集群，以及所选集群中的一个或多个命名空间。如果没有选择集群或命名空间，则视图会自动切换到 **All**。对过滤器的任何更改都会立即反映到所有小部件，限制它们存在的数据到所选范围。Dashboard 过滤器不会影响 **Status Bar**。

1.3. 小部件选项

可以自定义一些小部件，以帮助您专注于特定数据。小部件提供不同的控制，可用于更改数据的排序方式、过滤数据并自定义小部件的输出。

小部件提供了两种自定义不同方面的方法：

- **Options** 菜单在存在时提供适用于该小部件的特定选项。
- 当存在时，**动态图例** 提供了一种通过隐藏一个或多个 axis 类别来过滤数据的方法。例如，在 **Policy violations by category** widget 中，您可以点严重性来包含或排除数据中所选严重性的违反情况。

注意

各个小部件自定义设置是短期的，在离开仪表板时将其重置为系统默认设置。

1.4. 可操作的小部件

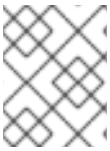
以下小节描述了 Dashboard 中可用的可操作小部件。

1.4.1. 策略违反情况的严重性

此小部件显示 Dashboard-filtered 范围的严重性级别分布。点 chart 中的 **严重性级别** 进入 **Violations** 页面，针对这个严重性和范围过滤。它还列出了您在 Dashboard 过滤器中定义的范围内三个最新的 **关键** 级别策略违反情况。点特定违反情况会直接进入该违反情况的 **Violations** 详情页面。

1.4.2. 镜像最有风险

此小部件列出了 Dashboard-filtered 范围中的前 6 个存在安全漏洞的镜像，它们按照计算的风险优先级排序，以及它们包含的重要 CVE 的数量。点击镜像名称直接进入 **Vulnerability Management** 下的 **Image Findings** 页面。使用 **Options** 菜单来专注于可修复的 CVE，或者进一步关注活跃镜像。



注意

当在 Dashboard 过滤器中选择了集群或命名空间时，显示的数据将被过滤为活跃镜像，或者被过滤范围内部署使用的镜像。

1.4.3. 部署面临大多数风险

此小部件提供有关环境中顶级部署风险的信息。它显示其他信息，如资源位置（集群和命名空间）和风险优先级分数。另外，您可以点部署来查看与部署相关的风险信息；例如，其策略违反和漏洞。

1.4.4. 旧镜像

较旧的镜像会带来更高的安全风险，因为它们可以包含已解决的漏洞。如果旧的镜像处于活动状态，它们可能会公开部署被利用。您可以使用此小部件来快速评估您的安全状况，并确定相关的镜像。您可以使用默认范围，或使用您自己的值自定义年龄间隔。您可以查看不活跃和活动的镜像，或使用 Dashboard 过滤器来专注于活动镜像的特定区域。然后，您可以点此小部件中的 age 组，在 **Vulnerability Management → Images** 页面中只查看这些镜像。

1.4.5. 按类别划分的策略

此小部件可帮助您深入了解您的组织在遵守安全策略时面临的挑战，方法是分析违反了哪些策略类型。该小部件显示值得关注的五个策略类别。探索 **Options** 菜单，了解数据片段的不同方法。您可以过滤数据，专门用于部署或运行时违反情况。

您还可以更改排序模式。默认情况下，数据首先根据最高严重性中的违反数量进行排序。因此，所有具有关键策略的类别都会在没有关键策略的类别之前显示。其他排序模式考虑违反总数，而不考虑严重性。因为有些类别不包含关键策略（例如"Docker CIS"），因此两种排序模式可以提供显著不同的视图，从而提

供额外的见解。

点击图形底部的严重性级别来包含或排除数据中的该级别。选择不同的严重性级别可能会生成不同前 5 个选择列表或其顺序。数据过滤到 Dashboard 过滤器选择的范围。

1.4.6. 按标准合规性

您可以将 **Compliance by 标准** 小部件与 Dashboard 过滤器一起使用，以专注于您最重要的区域。根据排序的顺序，这个小部件会列出前 6 个或后 6 个合规性基准。选择 **Options** 以按照覆盖范围百分比排序。点其中一个基准标签或图形直接进入 **Compliance Controls** 页面，按仪表板范围和所选基准过滤。

注意

Compliance widget 仅在运行 合规性扫描 后显示详情。

第 2 章 使用 COMPLIANCE OPERATOR

2.1. 使用带有 RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES 的 COMPLIANCE OPERATOR

您可以将 RHACS 配置为使用 Compliance Operator 进行 OpenShift Container Platform 集群的合规性报告和补救。Compliance Operator 的结果在 RHACS Compliance Dashboard 中报告。

Compliance Operator 自动审查许多技术实施，并将其与行业标准、基准和基准的某些方面进行比较。

Compliance Operator 不是一个审核员(auditor)。为了遵守这些各种标准或认证，您必须与授权的审计员参与，如合格的安全评估者(QSA)、联合授权局(JAB)或其他行业认可监管机构来评估您的环境。

Compliance Operator 根据与此类标准相关的通用信息和实践提出建议，并协助补救，但实际合规是您的责任。您需要与授权的审核员合作，以达到符合标准的要求。

有关最新更新，请参阅 Compliance Operator 发行注记。

2.1.1. 安装 Compliance Operator

使用 Operator Hub 安装 Compliance Operator。

重要

如果在 Sensor 完全正常工作后安装 Compliance Operator，您必须在安全集群中重启 Sensor。

有关重启 Sensor 的更多信息，请参阅"添加资源"部分中的"重启 Sensor"。

流程

1. 在 Web 控制台中，进入 **Operators → OperatorHub** 页面。
2. 在 **Filter by keyword** 框中输入 Compliance **operator** 以查找 Compliance Operator。
3. 选择 **Compliance Operator** 查看详情页面。
4. 阅读 Operator 的信息，然后点 **Install**。

后续步骤

- 配置 ScanSettingBinding 对象

其他资源

- 在安全集群中重启 Sensor

2.1.2. 在安全集群中重启 Sensor

如果在安装 RHACS 后安装 Compliance Operator，则需要使用命令行界面(CLI)或用户界面(UI)在安全集群中重启 Sensor。

流程

- 要从 CLI 重启 Sensor，请运行以下命令：

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

- 要从 UI 重启 Sensor，请执行以下步骤：

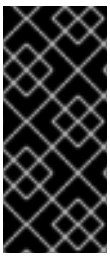
1. 将活动项目更改为 **stackrox**。
2. 进入 **Workloads → Pods**。
3. 找到名称以 **sensor-** 开头的 pod，然后单击 **Actions → Delete Pod**。

2.1.3. 配置 ScanSettingBinding 对象

在 **openshift-compliance** 命名空间中创建 **ScanSettingBinding** 对象，以使用 **cis** 和 **cis-node** 配置集扫描集群。

重要

- 如果使用合规性 2.0 功能，您可以使用 RHACS 创建合规性扫描计划而不是在 Compliance Operator 上创建 **ScanSettingBinding** 来调度扫描。
有关使用合规性 2.0 功能调度合规性扫描的更多信息，请参阅"添加资源"部分中的"自定义和自动化合规性扫描"。
- 本例使用 **ocp4-cis** 和 **ocp4-cis-node** 配置集，但 OpenShift Container Platform 提供了额外的配置集。如需更多信息，请参阅"添加资源"部分中的"了解 Compliance Operator"。



重要

Compliance 2.0 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。

流程

选择以下选项之一：

- 使用 CLI 创建 YAML 文件和对象。例如：

a. 使用以下文本创建名为 **sscan.yaml** 的文件：

```
apiVersion: compliance.openshift.io/v1alpha1
kind: ScanSettingBinding
metadata:
  name: cis-compliance
profiles:
  - name: ocp4-cis-node
    kind: Profile
    apiGroup: compliance.openshift.io/v1alpha1
  - name: ocp4-cis
    kind: Profile
    apiGroup: compliance.openshift.io/v1alpha1
```

```
settingsRef:
  name: default
  kind: ScanSetting
  apiGroup: compliance.openshift.io/v1alpha1
```

b. 运行以下命令来创建 **ScanSettingBinding** 对象：

```
$ oc create -f sscan.yaml -n openshift-compliance
```

如果成功，会显示以下信息：

```
$ scansettingbinding.compliance.openshift.io/cis-compliance created
```

- 通过执行以下步骤来创建对象：

a. 将活动项目更改为 **openshift-compliance**。

b. 点 + 打开 **Import YAML** 页面。

c. 粘贴上例中的 YAML，然后点 **Create**。

验证

1. 在 RHACS 中运行合规性扫描。
有关使用 compliance 1.0 功能运行合规性扫描的更多信息，请参阅"添加资源"部分中的"运行合规性扫描"。

2. 确保显示 **ocp4-cis** 和 **ocp4-cis-node** 结果。

其他资源

- 了解 Compliance Operator
- Compliance Operator 扫描
- 在 RHACS 中运行合规性扫描
- 自定义和自动化合规性扫描

第 3 章 管理合规性

3.1. 管理合规性 1.0 功能

通过使用 Red Hat Advanced Cluster Security for Kubernetes，您可以评估、检查并报告容器化基础架构的合规性状态。您可以根据行业标准运行开箱即用的合规性扫描，包括：

- 用于 **Docker** 和 **Kubernetes**的 **CIS Benchmarks** （互联网安全中心）
- **HIPAA** （健康可移植性和责任法案）
- **NIST** 特殊发布 800-190 和 800-53 （标准与技术研究院）
- **PCI DSS**（支付卡行业数据安全标准）
- **OpenSCAP** (Open Security Content Automation Protocol)：当安装 Compliance Operator 并配置为为 RHACS 提供结果时，OpenShift Container Platform 集群中的 RHACS 中可用

通过根据这些标准扫描您的环境，您可以：

- 评估您的基础架构是否符合法规合规性。
- 强化 Docker Engine 和 Kubernetes 编配器。
- 了解并管理环境的整体安全状态。
- 获取集群、命名空间和节点的合规性状态的详细视图。

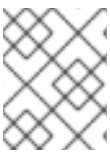
3.1.1. 查看合规性仪表板

合规仪表板提供环境中所有集群、命名空间和节点的合规性标准的高级视图。

合规仪表板包括图表，并提供用于调查与合规性要求的潜在问题的选项。您可以进入单个集群、命名空间或节点的合规性扫描结果。此外，您还可以在容器化环境中生成有关合规性状态的报告。

流程

- 在 RHACS 门户中，从导航菜单中选择 **Compliance (1.0)**。



注意

第一次打开 Compliance 仪表板时，您将看到一个空白仪表板。您必须运行合规性扫描来填充仪表板。

3.1.2. 运行合规性扫描

运行合规性扫描会在所有合规标准中检查整个基础架构的合规性状态。当您运行合规性扫描时，Red Hat Advanced Cluster Security for Kubernetes 会获取您的环境的数据快照。数据快照包括警报、镜像、网络策略、部署和相关基于主机的数据。Central 从集群中运行的 Sensors 收集基于主机的数据。之后，Central 从每个收集器 Pod 中运行的合规性容器收集更多数据。Compliance 容器收集有关环境的以下数据：

- Docker 守护进程、Docker 镜像和 Docker 容器的配置。
- 有关 Docker 网络的信息。

- Docker、Kubernetes 和 OpenShift Container Platform 的命令行参数和流程。
- 特定文件路径的权限。
- 核心 Kubernetes 和 OpenShift Container Platform 服务的配置文件。

数据收集完成后，Central 对数据执行检查以确定结果。您可以从合规仪表板查看结果，并根据结果生成合规性报告。

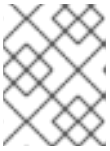
注意

在合规性扫描中：

- **Control** 描述了一个行业或监管合规标准中的单行项目，审核员会评估信息系统以遵守所述标准。Red Hat Advanced Cluster Security for Kubernetes 通过完成一个或多个检查来检查符合单个控制的证据。
- 检查是在单个控制评估期间执行的单个测试。
- 有些控制关联有多个检查。如果有任何关联的检查无法进行控制，则整个控制状态将标记为 **Fail**。

流程

1. 进入 RHACS 门户，并通过从导航菜单中选择 **Compliance (1.0)** 来打开合规仪表板。
2. 可选：默认情况下，所有标准下的信息都会在合规性结果中显示。要只查看特定标准的信息，请执行以下步骤：
   a. 点 **Manage Standard**。
   b. 默认情况下，选择所有标准。清除您不想显示的任何特定标准的复选框，然后单击 **Save**。没有选择的标准不会显示仪表板显示（包括小部件）、从仪表板访问的合规性结果表，以及使用 **Export** 按钮创建的 PDF 文件。但是，当以 CSV 文件导出结果时，会包括所有默认标准。
3. 点 **扫描环境**。



注意

扫描整个环境需要大约 2 分钟才能完成。根据环境中的集群和节点数量，这个时间可能会有所不同。

验证

1. 在 RHACS 门户中，进入 **Configuration Management**。
2. 在 **CIS Kubernetes v1.5** widget 中，点 **Scan**。
3. RHACS 显示一条消息，表示合规性扫描正在进行中。

3.1.3. 查看合规性扫描结果

运行合规性扫描后，合规仪表板会显示结果作为环境的合规性状态。您可以直接从仪表板查看合规违反情况，过滤详情视图并深入分析合规标准，以了解您的环境是否符合特定基准。本节介绍如何查看和过滤合规性扫描结果。

您可以使用快捷方式检查集群、命名空间和节点的合规性状态。在合规仪表板的顶部查找这些快捷方式。点击这些快捷方式，您可以查看合规快照，并根据集群、命名空间或节点的整体合规性生成报告。

合规性状态

状态	Description
Fail	合规性检查失败。
Pass	合规性检查通过。
N/A	Red Hat Advanced Cluster Security for Kubernetes 会跳过检查，因为它不适用。
info	合规检查收集的数据，但 Red Hat Advanced Cluster Security for Kubernetes 无法进行 **Pass** 或 **Fail** determination。
Error	由于技术问题，合规检查失败。

3.1.3.1. 查看集群的合规性状态

您可以从合规仪表板查看所有集群或单个集群的合规性状态。

流程

- 查看环境中所有集群的合规性状态：
  a. 进入 RHACS 门户，并通过从导航菜单中选择 **Compliance (1.0)** 来打开合规仪表板。
  b. 在合规仪表板上点 **Clusters**。
- 查看环境中特定集群的合规性状态：
  a. 进入 RHACS 门户，并通过从导航菜单中选择 **Compliance (1.0)** 来打开合规仪表板。
  b. 在合规仪表板上，**按照集群小部件查找护标准**。
  c. 在此小部件中，点集群名称查看其合规状态。

3.1.3.2. 查看命名空间的合规性状态

您可以从合规仪表板查看所有命名空间或单个命名空间的合规性状态。

流程

- 查看环境中所有命名空间的合规性状态：
  1. 进入 RHACS 门户，并通过从导航菜单中选择 **Compliance (1.0)** 来打开合规仪表板。
  2. 点合规仪表板上的 **Namespaces**。
- 查看环境中的特定命名空间的合规性状态：

1. 进入 RHACS 门户，并通过从导航菜单中选择 **Compliance (1.0)** 来打开合规仪表板。
2. 点 **Namespaces** 打开命名空间详情页面。
3. 在 **Namespaces** 表中点一个命名空间。在右侧打开一个侧面板。
4. 在侧面面板中，点命名空间的名称来查看其合规状态。

3.1.3.3. 查看特定标准的合规性状态

Red Hat Advanced Cluster Security for Kubernetes 支持 NIST、PCI DSS、NIST、HIPAA、Kubernetes 和 CIS for Docker 合规性标准的 CIS。您可以查看单个合规标准的所有合规控制。

流程

1. 进入 RHACS 门户，并通过从导航菜单中选择 **Compliance (1.0)** 来打开合规仪表板。
2. 在合规仪表板上，查找 **集群小部件之间的传递标准**。
3. 在此小部件中，点标准以查看与该标准关联的所有控件的信息。

注意

CIS Docker 中的许多控制都引用每个 Kubernetes 节点上 Docker 引擎的配置。许多 CIS Docker 控制也是构建和使用容器的最佳实践，RHACS 具有强制实施其用途的策略。如需更多信息，请参阅"添加资源"中的"管理安全策略"。

其他资源

- 管理安全策略

3.1.3.4. 查看特定控制的合规性状态

您可以查看所选标准的特定控制的合规性状态。

流程

1. 在 RHACS 门户中，进入 **Compliance (1.0)**。
2. 在合规仪表板上，**按照集群小部件查找护标准**。
3. 在此小部件中，点标准以查看与该标准关联的所有控件的信息。
4. 在 **Controls** 表中点控制。在右侧打开一个侧面板。
5. 在侧面面板中，点击控件的名称来查看其详情。

3.1.4. 过滤合规性状态

Red Hat Advanced Cluster Security for Kubernetes 搜索可让您从合规仪表板过滤不同的数据组合。要专注于集群、行业标准、传递或失败控制的子集，您可以缩小合规仪表板上可见的数据范围。

流程

1. 进入 RHACS 门户，并通过从导航菜单中选择 **Compliance (1.0)** 来打开合规仪表板。

2. 在合规仪表板上，选择 **Clusters**、或 **Namespaces** 或 **Nodes** 以打开详情页面。
3. 在搜索栏中输入过滤条件，然后按 **Enter** 键。

3.1.5. 生成合规性报告

Red Hat Advanced Cluster Security for Kubernetes 可让您生成报告来跟踪环境的合规性状态。您可以使用这些报告将各种行业的合规性状态传达给其他利益相关者。

您可以生成：

- **执行报告**，侧重于业务方面，并以 PDF 格式包括合规状态的图表和摘要。
- **证据报告**，侧重于技术方面，并以 CSV 格式包括详细信息。

流程

1. 进入 RHACS 门户，并通过从导航菜单中选择 **Compliance (1.0)** 来打开合规仪表板。
2. 在合规仪表板上，单击 **Export**。
   - 要生成执行报告，请选择 **Download page 作为 PDF**。
   - 要生成证据报告，请选择 **Download Evidence 作为 CSV**。

提示

Export 选项会出现在所有合规页面和过滤的视图中。

3.1.5.1. 证据报告

您可以以 CSV 格式从 Red Hat Advanced Cluster Security for Kubernetes 导出全面的合规相关数据作为证据报告。此证据报告包含有关合规评估的详细信息，并针对技术角色（如合规审核员、DevOps 工程师或安全专家）量身定制。

证据报告包含以下信息：

CSV 字段	Description
Standard（标准）	合规性标准，如 CIS Kubernetes。
集群	评估的集群的名称。
命名空间	部署所在的命名空间或项目的名称。
对象类型	对象的 Kubernetes 实体类型。例如，**节点**、**集群**、**DaemonSet**、**Deployment** 或 **StaticPod**。
对象名称	对象的名称，它是 Kubernetes 系统生成的字符串，用于唯一标识对象。例如，**gke-setup-dev21380-default-pool-8e086a77-1jfq**。

CSV 字段	Description
控制	控制号，它出现在合规标准中。
控制描述	有关合规性的描述，检查控制是否已执行。
状态	合规检查通过或失败。例如，**Pass** 或 **Fail**。
证据	有关特定合规检查失败或传递的原因的说明。
评估时间	运行合规性扫描的时间和日期。

3.1.6. 支持的基准版本

Red Hat Advanced Cluster Security for Kubernetes 支持以下行业标准和规范框架的合规性检查：

benchmark	支持的版本
用于 Docker 和 Kubernetes 的 CIS Benchmarks （互联网安全中心）	CIS Kubernetes v1.5.0 和 CIS Docker v1.2.0
HIPAA （健康可移植性和责任法案）	HIPAA 164
NIST （标准与技术研究院）	NIST 特殊发布 800-190 和 800-53 Rev. 4
PCI DSS （支付卡行业数据安全标准）	PCI DSS 3.2.1

3.2. 管理合规性 2.0 功能（技术预览）

重要

Compliance 2.0 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。

您可以使用 Red Hat Advanced Cluster Security for Kubernetes (RHACS)门户中的合规性 2.0 功能查看与集群关联的合规性结果。该功能将 Compliance Operator 收集的合规性信息收集到一个接口中。

有关使用 Compliance Operator 的更多信息，请参阅在 Red Hat Advanced Cluster Security for Kubernetes 中使用 Compliance Operator。

注意

目前，合规性 2.0 功能和 Compliance Operator 仅评估基础架构和平台合规性。

3.2.1. 查看集群的合规性状态

通过查看集群合规页面，您可以全面了解集群的合规状态。

流程

- 在 RHACS 门户中，进入 **Compliance (2.0)→ Cluster Compliance → Coverage** 选项卡。

3.2.2. 集群合规页面概述

集群合规页面在以下组中组织信息：

- **集群** ：提供集群的详情，并提供其当前状态和配置的快照。
- **Operator 状态** ：评估集群中 Compliance Operator 实例的健康状态和操作状态，并确保 Operator 在最佳运行可无缝运行。
- **Compliance** ：显示已针对扫描的配置集传递的检查百分比。

3.2.3. 自定义和自动化合规性扫描

通过创建合规性扫描调度，您可以自定义和自动化合规性扫描，使其与操作要求保持一致。

流程

1. 在 RHACS 门户中，进入 **Compliance (2.0)→ Cluster Compliance → Schedules** 选项卡。
2. 单击 **Create scan schedule**。
3. 在 **Configuration options** 页面中，提供以下信息：
   - **名称** ：输入名称来标识不同的合规性扫描。
   - **描述** ：指定每个合规性扫描的原因。
   - **配置调度** ：调整扫描调度以适合您的所需调度：
     - **频率** ：从下拉列表中选择您要执行扫描的频率。
       支持以下值：
       - **每日**
       - **每周**
       - **monthly**
     - **日期：** 从列表中，选择您要在其上执行扫描的一周中的一个或多个天数。
       支持以下值：
       - **Monday**
       - **Tuesday**
       - **Wednesday**
       - **周四**

- **周五**
- **Saturday**
- **Sunday**
- **month 的第一个**
- **月份的中间**

注意

只有在将扫描频率指定为 **Weekly** 或 **Monthly** 时，这些值才适用。

- **Time**: 开始键入要运行扫描的时间(**hh:mm**)。从显示的列表中，选择一个时间。

4. 点击 **Next**。
5. 在 **Clusters** 页面中，选择要包含在扫描中的一个或多个集群。
6. 点击 **Next**。
7. 在 **Profiles** 页面中，选择要包含在扫描中的一个或多个配置集。
8. 点击 **Next**。
9. 检查扫描配置，然后单击 **Create**。

验证

1. 在 RHACS 门户中，进入 **Compliance (2.0)→ Cluster Compliance → Schedules** 选项卡。
2. 选择您创建的合规性扫描。
3. 在 **Clusters** 部分中，验证 Operator 状态是否健康。
4. 可选： 要编辑扫描调度，点 **Edit scan schedule**，进行更改，然后点 **Save**。

3.2.4. 监控和分析集群的健康状况

通过查看合规性扫描的状态，您可以高效地监控和分析集群的健康状况。

重要

等待 Compliance Operator 返回扫描结果。它可能需要几分钟时间。

流程

1. 在 RHACS 门户中，进入 **Compliance (2.0)→ Cluster Compliance → Coverage** 选项卡。
2. 选择一个集群来查看单个扫描的详情。
3. 可选：在 **Filter by keyword 框中输入** 合规复选框的名称来查看状态。
4. 可选： 在 **Compliance status** 下拉列表中，使用您要过滤扫描详情来选择一个或多个状态。

支持以下值：

- **Pass**
- **Fail**
- **错误**
- **info**
- **Manual**
- **不适用**
- **Inconsistent**

3.2.5. 合规性扫描状态概述

通过了解合规性扫描状态，您可以管理环境的整体安全状态。

Status	Description
Fail	合规性检查失败。
Pass	合规性检查通过。
不适用	跳过合规检查，因为它不适用。
info	合规性检查收集的数据，但 RHACS 无法进行传递或未确定。
错误	由于技术问题，合规检查失败。
Manual	需要人工干预才能确保合规性。
Inconsistent	合规性扫描数据不一致，需要更接近检查和目标解析。

第 4 章 评估安全风险

Red Hat Advanced Cluster Security for Kubernetes 会评估整个环境中的风险，并根据其安全风险对运行的部署进行评级。它还详细介绍了需要立即关注的漏洞、配置和运行时活动。

4.1. 风险视图

Risk 视图列出所有来自所有集群的部署，根据策略违反、镜像内容、部署配置和其他类似因素，根据多因素风险指标进行排序。列表顶部的部署会带来最大风险。

Risk 视图显示每行具有以下属性的部署列表：

- **名称** ：部署的名称。
- **创建** ：部署的创建时间。
- **Cluster** ：运行部署的集群名称。
- **命名空间**：部署所在的命名空间。
- **优先级** ：优先级根据严重性和风险指标进行排名。

在 **Risk** 视图中，您可以：

- 选择一个列标题，以升序或降序排列违反情况。
- 使用过滤器栏过滤违反情况。
- 根据过滤的条件创建新策略。

要查看有关部署风险的更多详情，请在**风险**视图中选择部署。

4.1.1. 打开 risk 视图

您可以分析风险视图中的所有 **风险**，并采取纠正措施。

流程

- 进入 RHACS 门户并从导航菜单中选择 **风险**。

4.2. 从 RISK 视图创建安全策略

在**风险**视图中评估部署的风险时，当您应用本地页面过滤时，您可以根据您使用的过滤标准创建新的安全策略。

流程

1. 进入 RHACS 门户并从导航菜单中选择 **风险**。
2. 应用您要为其创建策略的本地页面过滤条件。
3. 选择 **New Policy** 并填写所需字段以创建新策略。

4.2.1. 了解 Red Hat Advanced Cluster Security for Kubernetes 如何将过滤条件转换为策略标准

当您根据您使用的过滤条件从 **Risk** 视图中创建新的安全策略时，并非所有条件都直接应用到新策略。

- Red Hat Advanced Cluster Security for Kubernetes 将 **Cluster**、**Namespace** 和 **Deployment** 过滤器转换为对等策略范围。
  - **风险** 视图中的本地页面过滤组合了搜索术语：
    - 将同一类别中的搜索词与 **OR** 运算符相结合。例如，如果搜索查询是 **Cluster:A,B**，则过滤器与**集群 A** 或**集群 B** 中的部署匹配。
    - 将不同类别的搜索术语与 **AND** 运算符相结合。例如，如果搜索查询是 **Cluster:A+Namespace:Z**，则过滤器 **与集群 A 和命名空间 Z** 中的部署匹配。
  - 当您向策略添加多个范围时，策略会匹配任何范围中的违反情况。
    - 例如，如果您搜索 **(Cluster A OR Cluster B) AND (Namespace Z)** 会导致两个策略范围： **(Cluster=A AND Namespace=Z)** OR **(Cluster=B AND Namespace=Z)**。
- Red Hat Advanced Cluster Security for Kubernetes 丢弃或修改没有直接映射到策略条件并报告丢弃的过滤器。

下表列出了过滤搜索属性如何映射到策略条件：

搜索属性	策略标准
添加功能	添加功能
注解	不允许注解
CPU 内核限制	容器 CPU 限制
CPU 内核请求	容器 CPU 请求
CVE	CVE
CVE 发布日期	dropped
CVE Snoozed	dropped
CVSS	CVSS
集群	mvapich 转换为范围
组件	镜像组件（名称）
组件版本	镜像组件（版本）
Deployment	mvapich 转换为范围

搜索属性	策略标准
部署类型	dropped
Dockerfile 指令关键字	Dockerfile 行（密钥）
Dockerfile 指令值	Dockerfile 行（值）
drop Capabilities	dropped
环境密钥	环境变量（密钥）
环境值	环境变量（值）
环境变量源	环境变量（源）
公开的节点端口	dropped
公开服务	dropped
公开服务端口	dropped
公开级别	端口公开
外部主机名	dropped
外部 IP	dropped
镜像	dropped
image 命令	dropped
创建的镜像	自镜像创建以来的天数
镜像条目点	dropped
镜像标签	不允许的镜像标签
镜像操作系统	镜像操作系统
Image Pull Secret	dropped
镜像 Registry	镜像 Registry
镜像远程	镜像远程
镜像扫描时间	自镜像上次扫描以来的天数

搜索属性	策略标准
镜像标签	镜像标签
镜像 Top CVSS	dropped
镜像用户	dropped
镜像卷	dropped
标签	mvapich 转换为范围
最大公开级别	dropped
内存限制(MB)	容器内存限制
内存请求(MB)	容器内存请求
命名空间	mvapich 转换为范围
命名空间 ID	dropped
Pod 标签	dropped
端口	端口
端口协议	协议
优先级	dropped
Privileged	Privileged
Process Ancestor	Process Ancestor
进程参数	进程参数
进程名称	进程名称
进程路径	dropped
进程标签	dropped
Process UID	Process UID
只读 Root 文件系统	只读 Root 文件系统
Secret	dropped

搜索属性	策略标准
Secret 路径	dropped
服务帐户	dropped
服务帐户权限级别	最低 RBAC 权限级别
容限键	dropped
容限值	dropped
卷目的地	卷目的地
卷名称	卷名称
卷 ReadOnly	可写卷
卷源	卷源
卷类型	卷类型

4.3. 查看风险详情

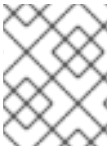
当您在 **Risk** 视图中选择一个部署时，**风险详情** 会在右侧的面板中打开。**Risk Details** 面板显示按多个标签页分组的详细信息。

4.3.1. risk Indicators 标签页

Risk Details 中的 **Risk Indicators** 标签页解释了发现的风险。

Risk Indicators 选项卡包括以下部分：

- **策略冲突** ：为所选部署违反的策略名称。
- **可疑的进程执行** ：可疑进程运行的进程、参数和容器名称。
- **镜像漏洞** ：镜像包括其 CVSS 分数的总 CVE。
- **服务配置** ：通常有问题的配置，如读写(RW)功能、是否丢弃能力以及特权容器是否存在。
- **服务不稳定** ：在集群内部或外部公开的容器端口。
- **组件对 攻击者使用**：发现软件工具被攻击者使用。
- **镜像中的组件数量** ：每个镜像中找到的软件包数量。
- **Image Freshness** ：镜像名称和年龄，如 **285 days old**。
- **RBAC 配置** ：授予 Kubernetes 基于角色的访问控制(RBAC)中的部署的权限级别。



注意

并非所有部分都出现在 **Risk Indicators** 选项卡中。Red Hat Advanced Cluster Security for Kubernetes 只会显示影响所选部署的相关部分。

4.4. DEPLOYMENT DETAILS 标签页

Deployment Risk 面板的 **Deployment Details** 选项卡中的部分提供了更多信息，以便您能够就如何解决发现的风险做出适当的决定。

4.4.1. 概述部分

Overview 部分显示以下内容的详情：

- **部署 ID** ：部署的字母数字字符。
- **命名空间** ：部署所在的 Kubernetes 或 OpenShift Container Platform 命名空间。
- **Updated**: 部署被更新时的时间戳。
- **部署类型** ：部署类型，如 **Deployment** 或 **DaemonSet**。
- **Replicas**：为此部署部署的 pod 数量。
- **标签** ：附加到 Kubernetes 或 OpenShift Container Platform 应用程序的键值标签。
- **Cluster** ：运行部署的集群名称。
- **Annotations** ：部署的 Kubernetes 注解。
- **Service Account** 代表 pod 中运行的进程的身份。当进程通过服务帐户进行身份验证时，它可以联系 Kubernetes API 服务器并访问集群资源。如果 pod 没有分配的服务帐户，它会获取 default 服务帐户。

4.4.2. 容器配置部分

容器配置部分显示以下内容：

- **Image Name** ：要部署的镜像的名称。
- **Resources**
  - **CPU 请求（内核）**：容器请求的 CPU 数量。
  - **CPU 限制（内核）** ：容器可以使用的最大 CPU 数量。
  - **内存请求(MB)** ：容器请求的内存大小。
  - **内存限制(MB)** ：容器可以使用的最大内存量，而不被终止。
- **mounts**
  - **名称** ：挂载的名称。
  - **源** ：挂载数据来自的路径。
  - **目标** ：挂载数据到的路径。

- 类型 ：挂载的类型。

- Secrets：部署中使用的 Kubernetes secret 的名称，以及用于 X.509 证书的 secret 值的基本详情。

4.4.3. 安全上下文部分

安全上下文 部分显示以下内容：

- Privileged：如果容器是特权，则列为 **true**。

4.5. 进程发现标签页

Process Discovery 选项卡提供环境中每个容器执行的所有二进制文件的完整列表，具体由部署概述。

process discovery 标签页显示以下内容：

- **二进制名称** ：执行的二进制名称。

- **容器** ：执行进程的部署中的容器。

- **参数** ：通过二进制文件传递的特定参数。

- **时间** ：在给定容器中执行二进制文件的最长时间的日期和时间。

- **Pod ID** ：容器所在的 pod 的标识符。

- **UID** ：进程执行的 Linux 用户身份。

使用过滤器栏中的 **Process Name:<name** > 查询来查找特定的进程。

4.5.1. 事件时间表部分

Process Discovery 选项卡中的 Event Timeline 部分提供了所选部署的事件概述。它显示策略违反、进程活动和容器终止或重启事件的数量。

您可以选择 Event Timeline 查看更多详细信息。

Event Timeline 模态框显示所选部署的所有 pod 的事件。

时间表上的事件归类为：

- 进程活动

- 策略违反情况

- 容器重启

- 容器终止

事件以图标形式显示在时间表上。要查看有关事件的更多详细信息，请将鼠标指针悬停在事件图标上。详情会出现在工具提示中。

- 点 Show Legend 来查看与哪个事件类型对应的图标。

- 选择 **Export → Download PDF** 或 **Export →** Download **→ Download CSV** 以下载事件时间表信息。
- 选择 **Show All** 下拉菜单来过滤在时间表上可见的事件类型。
- 点展开图标，以单独查看所选 pod 中每个容器的事件。

时间表中的所有事件也可以在底部的 minimap 控制中可见。minimap 控制事件时间表中可见的事件数量。您可以通过修改 minimap 上突出显示的区域来更改时间表中显示的事件。为此，可从左或右边（或两者）减小突出显示的区域，然后拖动突出显示的区域。

注意

- 当容器重启时，Red Hat Advanced Cluster Security for Kubernetes:
  - 显示 pod 中每个容器最多 10 个不活跃容器实例的容器终止和重启事件。例如，对于有两个容器 **app** 和 **sidecar** 的 pod，Red Hat Advanced Cluster Security for Kubernetes 会保持活动最多 10 个应用程序实例，最多保留 10 个 **sidecar** 实例。
  - 不跟踪与容器之前实例关联的进程活动。
- Red Hat Advanced Cluster Security for Kubernetes 仅显示每个 pod 的每个（进程名称、进程参数、UID）元组的最新执行。
- Red Hat Advanced Cluster Security for Kubernetes 仅显示活跃 pod 的事件。
- Red Hat Advanced Cluster Security for Kubernetes 根据 Kubernetes 和 Collector 报告的时间调整报告的时间戳。Kubernetes 时间戳使用基于第二个精度，并将时间舍入到最接近的秒。但是，Collector 使用更精确的时间戳。例如，如果 Kubernetes 将容器启动时间报告为 **10:54:48**，并且 Collector 报告了从 **10:54:47.5349823** 启动的容器中的一个进程，Red Hat Advanced Cluster Security for Kubernetes 会将容器启动时间调整为 **10:54:47.5349823**。

4.6. 使用进程基准

您可以使用精简基础架构的安全性流程来最小化风险。使用这个方法，Red Hat Advanced Cluster Security for Kubernetes 首先发现现有的进程并创建基准。然后，它以默认的 deny-all 模式运行，只允许基准中列出的进程运行。

进程基准
安装 Red Hat Advanced Cluster Security for Kubernetes 时，没有默认的进程基准。当 Red Hat Advanced Cluster Security for Kubernetes 发现部署时，它会为部署中的每个容器类型创建一个进程基准。然后，它会将所有发现的进程添加到自己的进程基线中。

进程基准状态
在进程发现阶段，所有基准都处于解锁的状态。

处于 **解锁** 的状态：

- 当 Red Hat Advanced Cluster Security for Kubernetes 发现新进程时，它会将该进程添加到流程基线中。
- 进程没有以风险的形式显示，且不会触发任何违反情况。

当 Red Hat Advanced Cluster Security for Kubernetes 收到部署中容器的第一个进程指示符后，它会完成进程发现阶段。此时：

- Red Hat Advanced Cluster Security for Kubernetes 会停止在进程基准中添加进程。
- 进程基准中没有的新进程显示为风险，但它们不会触发任何违反情况。

要生成违反情况，您必须手动锁定进程基准。

处于 **锁定状态** ：

- Red Hat Advanced Cluster Security for Kubernetes 会停止在进程基准中添加进程。
- 不在进程基准中的新进程会触发违反情况。

独立于锁定或解锁的基准状态，您可以始终从基准中添加或删除进程。

注意

对于部署，如果每个 pod 有多个容器，Red Hat Advanced Cluster Security for Kubernetes 会为每个容器类型创建一个进程基准。对于这样的部署，如果一些基准被锁定，并且有些基准被解锁，则该部署的基准状态会显示为 **Mixed**。

4.6.1. 查看进程基准

您可以从 **风险** 视图中查看进程基准。

流程

1. 在 RHACS 门户中，从导航菜单中选择 **Risk**。
2. 在默认 **风险** 视图中从部署列表中选择部署。部署详情在右侧的面板中打开。
3. 在 **Deployment** details 面板中，选择 **Process Discovery** 选项卡。
4. 进程基准在 **Spec Container Baselines** 部分可见。

4.6.2. 在基准中添加进程

您可以将进程添加到基准中。

流程

1. 在 RHACS 门户中，从导航菜单中选择 **Risk**。
2. 在默认 **风险** 视图中从部署列表中选择部署。部署详情在右侧的面板中打开。
3. 在 **Deployment** details 面板中，选择 **Process Discovery** 选项卡。
4. 在 **Running Processes** 部分下，点您要添加到进程基数的进程的 **Add** 图标。

注意

Add 图标仅适用于不在进程基准中的进程。

4.6.3. 从基准中删除进程

您可以从基准中删除进程。

流程

1. 在 RHACS 门户中，从导航菜单中选择 **Risk**。
2. 在默认 **风险** 视图中从部署列表中选择部署。部署详情在右侧的面板中打开。
3. 在 **Deployment** details 面板中，选择 **Process Discovery** 选项卡。
4. 在 **Spec Container baselines** 部分下，点您要从进程基准中删除的进程的 **Remove** 图标。

4.6.4. 锁定和解锁进程基准

您可以**锁定**基线，以为所有没有列在基线中的进程触发违规，也可以**取消锁定**基线来停止触发违规。

流程

1. 在 RHACS 门户中，从导航菜单中选择 **Risk**。
2. 在默认 **风险** 视图中从部署列表中选择部署。部署详情在右侧的面板中打开。
3. 在 **Deployment** details 面板中，选择 **Process Discovery** 选项卡。
4. 在 **Spec Container baselines** 部分下：
   - 点 **Lock** 图标为不在基准中的进程触发违反情况。
   - 点 **Unlock** 图标停止为不在基准中的进程触发违反情况。

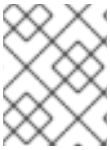
第 5 章 使用准入控制器强制

Red Hat Advanced Cluster Security for Kubernetes 可以与 Kubernetes 准入控制器 和 OpenShift Container Platform 准入插件 一起工作，允许您在 Kubernetes 或 OpenShift Container Platform 创建工作负载前强制实施安全策略，如部署、守护进程集或作业。

RHACS 准入控制器可防止用户创建违反您在 RHACS 中配置策略的工作负载。从 RHACS 版本 3.0.41 开始，您还可以配置准入控制器以防止对违反策略的工作负载进行更新。

RHACS 使用 **ValidatingAdmissionWebhook** 控制器来验证正在置备的资源是否符合指定的安全策略。为了解决这个问题，RHACS 会创建一个 **ValidatingWebhookConfiguration**，其中包含多个 Webhook 规则。

当 Kubernetes 或 OpenShift Container Platform API 服务器收到与其中一个 webhook 规则匹配的请求时，API 服务器会向 RHACS 发送 **AdmissionReview** 请求。然后，RHACS 根据配置的安全策略接受或拒绝请求。



注意

要在 OpenShift Container Platform 上使用准入控制器强制，您需要 Red Hat Advanced Cluster Security for Kubernetes 版本 3.0.49 或更新版本。

5.1. 了解准入控制器强制

如果要使用准入控制器强制，请考虑以下几点：

- **API 延迟** ：使用准入控制器强制会增加 Kubernetes 或 OpenShift Container Platform API 延迟，因为它涉及额外的 API 验证请求。许多标准 Kubernetes 库（如 fabric8）默认具有简短的 Kubernetes 或 OpenShift Container Platform API 超时。另外，请考虑您可能使用的任何自定义自动化中的 API 超时。
- **镜像扫描** ：您可以通过在集群配置面板中设置 **Contact Image Scanners** 选项来选择准入控制器在查看请求时是否扫描镜像。
  - 如果启用此设置，如果扫描或镜像签名验证结果不可用，Red Hat Advanced Cluster Security for Kubernetes 会联系镜像扫描程序，这会增加显著的延迟。
  - 如果您禁用了此设置，则强制决定仅在缓存的扫描和签名验证结果可用时考虑镜像扫描条件。
- 您可以使用准入控制器强制进行：
  - pod **securityContext** 中的选项。
  - 部署配置。
  - 镜像组件和漏洞。
- 您不能为以下目的使用准入控制器强制：
  - 任何运行时行为，如进程。
  - 基于端口暴露的任何策略。

- 如果 Kubernetes 或 OpenShift Container Platform API 服务器和 RHACS Sensor 之间存在连接问题，则准入控制器可能会失败。要解决这个问题，删除 **ValidatingWebhookConfiguration** 对象，如禁用准入控制器强制部分所述。

- 如果您为策略启用了部署时间强制，且启用了准入控制器，RHACS 会尝试阻止违反策略的部署。如果准入控制器没有拒绝不合规的部署，例如，在超时时，RHACS 仍然会应用其他部署时间强制机制，如扩展到零副本。

5.2. 启用准入控制器强制

在安装 Sensor 或编辑现有集群配置时，您可以从 **Clusters** 视图中启用准入控制器强制。

流程

1. 在 RHACS 门户中，进入 **Platform Configuration → Clusters**。
2. 选择 **Secure a cluster → Legacy 安装方法** 从列表中选择现有集群或保护新集群。
3. 如果要保护新集群，在 **集群配置** 面板的静态配置部分中，输入集群的详情。
4. 如果您计划使用准入控制器在 **对象创建事件上强制使用准入控制器，则红帽建议您只打开 Configure Admission Controller Webhook 来侦听** 对象创建切换。
5. 如果您计划使用准入控制器在更新事件时强制使用准入控制器，则红帽建议您只打开 **Configure Admission Controller Webhook 来侦听对象更新** 切换。
6. 如果您计划使用准入控制器强制 pod 执行 **和 pod 端口转发事件，红帽建议只打开 Enable Admission Controller Webhook 来侦听 exec 和 port-forward** 事件切换。
7. 在 **Dynamic Configuration** 部分中配置以下选项：
   - **在对象创建时强制** ：此切换控制准入控制服务的行为。您必须具有 **Configure Admission Controller Webhook 来侦听 Object Creates** 切换，才能使它正常工作。
   - **在对象更新上强制** ：此切换控制准入控制服务的行为。您必须具有 **Configure Admission Controller Webhook 来侦听 Object Updates** 切换，才能使它正常工作。
8. 选择 **Next**。
9. 在 **Download files** 部分中，选择 **Download YAML 文件和密钥**。

注意

当为现有集群启用准入控制器时，请按照以下指导操作：

- 如果您在 **Static Configuration** 部分中进行任何更改，您必须下载 YAML 文件并重新部署 Sensor。
- 如果您在 **Dynamic Configuration** 部分中进行任何更改，您可以跳过下载文件和部署，因为 RHACS 会自动同步 Sensor 并应用更改。

10. 选择 **Finish**。

验证

- 使用生成的 YAML 置备新集群后，运行以下命令验证准入控制器强制是否正确配置：

```
$ oc get ValidatingWebhookConfiguration 1
```

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

输出示例

```
NAME       CREATED AT
stackrox   2019-09-24T06:07:34Z
```

5.3. 绕过准入控制器强制

要绕过准入控制器，请在配置 YAML 中添加 **admission.stackrox.io/break-glass** 注解。绕过准入控制器会触发策略违反情况，其中包括部署详情。红帽建议提供一个 issue-tracker 链接或作为此注解值的其他引用，以便其他人可以了解您绕过准入控制器的原因。

5.4. 禁用准入控制器强制

您可以从 Red Hat Advanced Cluster Security for Kubernetes (RHACS)门户上的 **Clusters** 视图禁用准入控制器强制。

流程

1. 在 RHACS 门户中，选择 **Platform Configuration → Clusters**。
2. 从列表中选择现有集群。
3. 在 **Dynamic Configuration** 项中，关闭T **Enforce on Object Creates** 和 **Enforce on Object Updates**。
4. 选择 **Next**。
5. 选择 **Finish**。

5.4.1. 禁用关联的策略

您可以在相关策略上关闭强制，该策略会指示准入控制器跳过强制。

流程

1. 在 RHACS 门户中，进入 **Platform Configuration → Policy Management**。
2. 在默认策略中禁用强制：
   - 在 policies 视图中，找到 **Kubernetes Actions: Exec into Pod** 策略。点 overflow 菜单 ，然后选择 **Disable policy**。
   - 在 policies 视图中，找到 **Kubernetes Actions: Port Forward to Pod** 策略。点 overflow 菜单 ，然后选择 **Disable policy**。

3. 使用来自默认 **Kubernetes Actions: Port Forward to Pod** 和 **Kubernetes Actions: Exec into Pod** 策略中的条件来在禁用您所创建的任何自定义策略的强制。

5.4.2. 禁用 Webhook

您可以从 RHACS 门户中的 **Clusters** 视图禁用准入控制器强制。

重要

如果通过关闭 webhook 来禁用准入控制器，您必须重新部署 Sensor 捆绑包。

流程

1. 在 RHACS 门户中，进入 **Platform Configuration → Clusters**。
2. 从列表中选择现有集群。
3. 在 **Static Configuration** 项中关闭 **Enable Admission Controller Webhook to listen on exec and port-forward events**。
4. 选择 **Next** 以继续 Sensor 设置。
5. 点 **Download YAML 文件和密钥**。
6. 在可以访问被监控的集群的系统中，提取并运行 **传感器** 脚本：

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

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

注意

如果您收到没有部署传感器所需的权限的警告，请按照屏幕上的说明操作，或联系您的集群管理员寻求帮助。

部署传感器后，它会联系中心并提供集群信息。

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

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

   - 对于 Kubernetes：

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

8. 选择 **Finish**。

注意

当您禁用准入控制器时，RHACS 不会删除 **ValidatingWebhookConfiguration** 参数。但是，它不接受所有 **AdmissionReview** 请求，而不是检查违反情况的请求。

要删除 **ValidatingWebhookConfiguration** 对象，请在安全集群中运行以下命令：

- 在 OpenShift Container Platform 中：

```
$ oc delete ValidatingWebhookConfiguration/stackrox
```

- 对于 Kubernetes：

```
$ kubectl delete ValidatingWebhookConfiguration/stackrox
```

5.5. VALIDATINGWEBHOOKCONFIGURATION YAML 文件更改

使用 Red Hat Advanced Cluster Security for Kubernetes，您可以强制使用以下安全策略：

- 对象创建
- 对象更新
- Pod 执行
- Pod 端口转发

如果 Central 或 Sensor 不可用

准入控制器需要从 Sensor 进行初始配置才能工作。Kubernetes 或 OpenShift Container Platform 会保存此配置，即使所有准入控制服务副本重新调度到其他节点上，它仍然可以访问。如果存在此初始配置，准入控制器会强制执行所有配置的部署时策略。

如果 Sensor 或 Central 稍后不可用：

- 您将无法运行镜像扫描，或者查询缓存的镜像扫描的信息。但是，准入控制器根据超时过期前收集的可用信息进行仍然可以正常工作，即使收集的信息不完整。
- 您将无法从 RHACS 门户禁用准入控制器，或修改现有策略的强制，因为更改不会传播到准入控制服务。

注意

如果需要禁用准入控制强制功能，您可以通过运行以下命令来删除验证 Webhook 配置：

- 在 OpenShift Container Platform 中：

```
$ oc delete ValidatingWebhookConfiguration/stackrox
```

- 对于 Kubernetes：

```
$ kubectl delete ValidatingWebhookConfiguration/stackrox
```

使准入控制器更可靠

红帽建议在 control plane 上调度准入控制服务，而不是在 worker 节点上调度。部署 YAML 文件包含在 control plane 上运行的软首选项，但它不会被强制使用。

默认情况下，准入控制服务运行 3 个副本。要提高可靠性，您可以运行以下命令来增加副本：

```
$ oc -n stackrox scale deploy/admission-control --replicas=<number_of_replicas> 1
```

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

使用 roxctl CLI

您可以在生成 Sensor 部署 YAML 文件时使用以下选项：

- **--admission-controller-listen-on-updates** ：如果您使用这个选项，Red Hat Advanced Cluster Security for Kubernetes 会生成一个 Sensor 捆绑包，并带有 **ValidatingWebhookConfiguration**，来从 Kubernetes 或 OpenShift Container Platform API 服务器接收更新事件。

- **--admission-controller-enforce-on-updates** ：如果您使用这个选项，Red Hat Advanced Cluster Security for Kubernetes 配置 Central，以便准入控制器还会强制实施安全策略对象更新。

这两个选项都是可选的，默认为 **false**。

第 6 章 管理安全策略

Red Hat Advanced Cluster Security for Kubernetes 允许您使用开箱即用的安全策略，并为容器环境定义自定义多因素策略。通过配置这些策略，您可以自动防止环境中的高风险服务部署，并响应运行时安全事件。

6.1. 使用默认的安全策略

Red Hat Advanced Cluster Security for Kubernetes 包括了一组默认策略，它们提供广泛的覆盖范围来识别安全问题，并确保您的环境中的安全性最佳实践。

查看默认策略：

- 在 RHACS 门户中，进入 **Platform Configuration → Policy Management**。

Policies 视图列出了默认策略，并为每个策略包括以下参数：

- **策略** ：策略的名称。
- **Description**: A longer, more detailed description for the policy.
- **状态** ：策略的当前状态，可以是 **Enabled** 或 **Disabled**。
- **Notifiers**：为策略配置的通知程序列表。
- **严重性** ：在需要注意的注意程度上，策略(critical、高、中等)的排名。
- **生命周期** ：此策略应用到的容器生命周期（构建、部署或运行时）的阶段，以及启用策略时强制执行的阶段。

默认策略预先配置了参数，并且属于类别，例如：

- 异常活动
- Cryptocurrency Mining
- DevOps 最佳实践
- Kubernetes
- 网络工具
- 软件包管理
- 权限
- 安全性最佳实践
- 系统修改
- 漏洞管理

您可以编辑这些类别并创建自己的类别。

注意

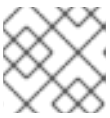
您不能删除默认策略或编辑默认策略标准。

6.2. 修改现有安全策略

您可以编辑您创建的策略以及 Red Hat Advanced Cluster Security for Kubernetes 提供的现有默认策略。

流程

1. 在 RHACS 门户中，进入 **Platform Configuration → Policy Management**。
2. 在 **Policies** 页面中，选择您要编辑的策略。
3. 选择 **Actions → Edit policy**。
4. 修改**策略详细**。您可以修改策略名称、严重性、类别、描述、比例和指导。您还可以通过从 **Attach notifiers** 部分下的可用的 **Notifiers** 选择将通知程序附加到策略。
5. 点击 **Next**。
6. 在 **Policy behavior** 部分中，为策略选择 **Lifecycle stages** 和 **Event sources**。
7. 选择 **Response 方法** 来解决策略违反情况。
8. 点击 **Next**。
9. 在 **Policy criteria** 部分中，展开 Drag **out 策略字段** 部分下的类别。使用 drag-and-drop 策略字段指定策略条件的逻辑条件。



注意

您无法编辑默认策略的策略标准。

10. 点击 **Next**。
11. 在 **Policy Scope** 部分中，修改 **Restrict by scope**, **Exclude by scope**, 和 **Exclude images** 设置。
12. 点击 **Next**。
13. 在 **Review policy** 部分中，预览策略违反情况。
14. 点击 **Save**。

其他资源

- 从系统策略视图创建安全策略

6.3. 创建和管理策略类别

6.3.1. 使用 Policy categories 选项卡创建策略类别

从版本 3.74 开始，RHACS 提供了在 Red Hat Advanced Cluster Security Cloud Service 或 RHACS 中创建和管理策略类别的新方法，如果您启用了 PostgreSQL 数据库。使用此功能时，策略创建以外的所有策略工作流都保持不变。

您还可以使用 **PolicyCategoryService** API 对象来配置策略类别。如需更多信息，请参阅 RHACS 门户中的 **Help → API 参考**。

流程

1. 在 RHACS 门户中，进入 **Platform Configuration → Policy Management**。
2. 点 **Policy categories** 选项卡。此选项卡提供了现有类别的列表，允许您根据类别名称过滤列表。您还可以单击 **Show all categories**，然后选择要从显示列表中删除默认或自定义类别的复选框。
3. 点 **Create category**。
4. 输入类别名称并点 **Create**。

6.3.2. 使用 Policy categories 选项卡修改策略类别

从版本 3.74 开始，RHACS 提供了在 Red Hat Advanced Cluster Security Cloud Service 或 RHACS 中创建和管理策略类别的新方法，如果您启用了 PostgreSQL 数据库。使用此功能时，策略创建以外的所有策略工作流都保持不变。

您还可以使用 **PolicyCategoryService** API 对象来配置策略类别。如需更多信息，请参阅 RHACS 门户中的 **Help → API 参考**。

流程

1. 在 RHACS 门户中，进入 **Platform Configuration → Policy Management**。
2. 点 **Policy categories** 选项卡。此选项卡提供了现有类别的列表，允许您根据类别名称过滤列表。您还可以单击 **Show all categories**，然后选择要从显示列表中删除默认或自定义类别的复选框。
3. 点策略名称编辑或删除它。无法选择、编辑或删除默认策略类别。

其他资源

- 从系统策略视图创建安全策略

6.4. 创建自定义策略

除了使用默认策略外，您还可以在 Red Hat Advanced Cluster Security for Kubernetes 中创建自定义策略。

要构建新策略，您可以克隆现有策略或从头开始创建一个新策略。

- 您还可以根据 RHACS 门户中的 **Risk** 视图中的过滤器标准创建策略。
- 您还可以在策略条件中使用 **AND**, **OR**, 和 **NOT** 逻辑运算符来创建高级策略。

6.4.1. 从系统策略视图创建安全策略

您可以从系统策略视图创建新的安全策略。

流程

1. 在 RHACS 门户中，进入 **Platform Configuration → Policy Management**。
2. 点击 **Create policy**。
3. 在 **Policy details** 部分中，输入以下有关您的策略的详细信息：
   - 输入策略的 **Name**。
   - 可选：通过从 **Attach notifiers** 部分下的可用 **Notifiers** 选择，将通知程序附加到策略。

     注意

     在转发警报前，您必须将 RHACS 与通知供应商（如 Webhook、JIRA、Pageruty、Spluty、Splunk 等）集成。
   - 选择 **此策略的严重性级别**，可以是 **Critical**、**High**、**Medium** 或 **Low**。
   - 选择您要 **应用到** 此策略的策略类别。有关创建类别的详情，请参考本文档后面的"创建和管理策略类别"。
   - 在 **Description** 字段中输入策略详情。
   - 在 **Rationale** 字段中输入有关为什么策略存在的解释。
   - 在 **Guidance** 字段中输入步骤来解决此策略的违反情况。
   - 可选：在 **MITRE ATT&CK** 部分中，选择您需要为策略指定的 tactics and the techniques。
     a. 单击 **Add tactic**，然后从下拉列表中选择 tactic。
     b. 点 **Add Technology**，为所选 tactic 添加技术。您可以为 tactic 指定多种技术。
4. 点击 **Next**。
5. 在 **Policy behavior** 部分中，执行以下步骤：
   a. 选择适用于 **您的策略** 的生命周期阶段： **Build**、deploy**或 Runtime**。您可以选择多个阶段。
      - 构建时策略适用于镜像字段，如 CVE 和 Dockerfile 指令。
      - deploy-time 策略可包括所有构建时策略标准，但它们也可以包含集群配置中的数据，如以特权模式运行或挂载 Docker 套接字。
      - 运行时策略可以包含所有构建时间和部署时间策略标准，但它们也可以包含运行时进程执行的数据。
   b. 可选：如果您选择了 **Runtime** 生命周期阶段，请选择以下 **事件源** 之一：
      - Deployment ：当事件源包括进程和网络活动、pod exec 和 pod 端口转发时，RHACS 会触发策略违反情况。
      - 当事件源与 Kubernetes 审计日志记录匹配时，RHACS 会触发策略违反情况。
6. 对于 **Response 方法**，请选择以下选项之一：
   a. **inform** ：在违反列表中包括违反情况。

b. **inform 和 enforce** ：强制操作。

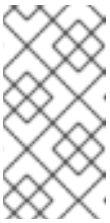
7. 可选：如果您选择了 **Inform and enforce**，在 **Configure enforcement behavior** 中，使用每个生命周期选择策略的强制行为。它仅适用于您在配置生命周期阶段时所选择的 **阶段**。每个生命周期阶段的强制行为都有所不同。

   - **构建** ：当镜像与策略条件匹配时，RHACS 无法构建您的持续集成(CI)。
   - **部署** ：对于 **Deploy** 阶段，RHACS 会阻止在 RHACS 准入控制器配置并运行时与策略条件匹配的部署。
     - 在带有准入控制器强制的集群中，Kubernetes 或 OpenShift Container Platform API 服务器会阻止所有不合规的部署。在其他集群中，RHACS 编辑不合规部署，以防止调度 pod。
     - 对于现有部署，策略更改仅在发生 Kubernetes 事件时在下次检测条件时导致强制。有关强制的更多信息，请参阅"部署阶段的安全策略强制"。
   - **Runtime** - 当 pod 中的事件与策略条件匹配时，RHACS 会删除所有 pod。

警告

策略实施可能会影响运行应用程序或开发流程。在启用强制选项前，请通知所有利益相关者，并计划如何响应自动执行操作。

8. 点击 **Next**。
9. 在 **Policy Criteria** 部分中，配置您要触发该策略的属性。
   a. 单击策略字段并将它拖到 **Policy Section** 中，以添加条件。



注意

可用的策略字段取决于您为策略选择的生命周期阶段。例如，在为运行时生命周期创建策略时，**Kubernetes 访问策略** 或 **Networking** 下的条件可用，但为构建生命周期创建策略时不可用。有关策略条件的更多信息，请参阅"附加资源"部分中的"策略标准"部分，包括有关条件及其可用的生命周期阶段的信息。

   b. 可选：点 **Add condition** 来添加包括会触发策略的额外条件的策略部分（例如，为了触发一个旧的、稳定的镜像，您可以配置 **image tag 不是 latest 或 image age** 并指定自一个镜像构建后的需要经过的最少天数）。

10. 点击 **Next**。
11. 在 **Policy scope** 部分中，配置以下内容：
    - 点 **Add inclusion scope** 使用 **Restrict by scope** 仅对特定集群、命名空间或标签启用此策略。您可以添加多个范围，并为命名空间和标签在 RE2 语法中使用正则表达式。

- 点 **Add excluded scope** 来使用 **Exclude by scope** 来排除您指定的部署、集群、命名空间和标签。该策略不适用于您选择的实体。您可以添加多个范围，并为命名空间和标签在 RE2 语法中使用正则表达式。但是，您不能使用正则表达式来选择部署。

- 对于 **Excluded Images （仅限构建生命周期）**，请选择您不想触发违反的所有镜像。

注意

Excluded Images 设置仅在使用 **Build** 生命周期阶段检查持续集成系统中的镜像时适用。如果您使用此策略检查 **Deploy** 生命周期阶段中运行的部署，或检查在 **Runtime** 生命周期阶段中的运行时活动时，则不会生效。

12. 点击 **Next**。
13. 在 **Review policy** 部分中，预览策略违反情况。
14. 点击 **Save**。

6.4.1.1. 部署阶段的安全策略强制

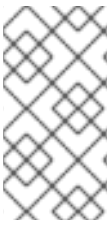
Red Hat Advanced Cluster Security for Kubernetes 支持两种类型的安全策略强制进行部署时间策略强制：通过准入控制器和 RHACS Sensor 的软强制进行硬强制。准入控制器会阻止创建或更新违反策略的部署。如果准入控制器被禁用或不可用，则 Sensor 可以通过将违反策略部署到 **0** 的部署来缩减副本来执行强制。

警告

策略实施可能会影响运行应用程序或开发流程。在启用强制选项前，请通知所有利益相关者，并计划如何响应自动强制操作。

6.4.1.1.1. 硬强制

硬强制由 RHACS 准入控制器执行。在带有准入控制器强制的集群中，Kubernetes 或 OpenShift Container Platform API 服务器会阻止所有不合规的部署。准入控制器会阻止 **CREATE** 和 **UPDATE** 操作。任何满足启用了 deploy-time 强制配置的策略的 pod 创建或更新请求都将失败。



注意

Kubernetes 准入 webhook 仅支持 **CREATE**、**UPDATE**、**DELETE** 或 **CONNECT** 操作。RHACS 准入控制器只支持 **CREATE** 和 **UPDATE** 操作。**kubectl patch**、**kubectl set** 和 **kubectl scale** 等操作是 PATCH 操作，而不是 UPDATE 操作。因为 Kubernetes 不支持 PATCH 操作，所以 RHACS 无法对 PATCH 操作执行强制。

要进行阻塞，您必须在 RHACS 中为集群启用以下设置：

- **在 Object Creates 上强制**：此切换在 **Dynamic Configuration** 部分中，控制准入控制服务的行为。您必须在打开的 **Static Configuration** 部分中具有 **Configure Admission Controller Webhook 来侦听 Object Creates** 开关才能正常工作。

- **在对象更新上强制** ：此切换在 **Dynamic Configuration** 部分中，控制准入控制服务的行为。您必须在打开的 **Static Configuration** 部分中具有 **Configure Admission Controller Webhook 来侦听 Object Updates** 切换。

如果您对 **Static Configuration** 设置进行了更改，您必须重新部署安全集群才能使这些更改生效。

6.4.1.1.2. 软强制

软强制由 RHACS Sensor 执行。这个强制可防止启动操作。使用软强制时，Sensor 将副本扩展到 0，并阻止调度 pod。在这个强制中，集群中提供了非就绪的部署。

如果配置了软强制，且 Sensor 停机，则 RHACS 无法执行强制。

6.4.1.1.3. 命名空间排除

默认情况下，RHACS 从强制阻止中排除某些管理命名空间，如 **stackrox**、**kube-system** 和 **istio-system** 命名空间。这样做的原因是，必须部署这些命名空间中的一些项目才能使 RHACS 正常工作。

6.4.1.1.4. 对现有部署的强制

对于现有部署，策略更改仅在发生 Kubernetes 事件时在下次检测条件时导致强制。如果对策略进行更改，您必须通过选择 **Policy Management** 并点 **Reassess All** 来重新评估策略。此操作会在所有现有部署中应用部署策略，无论是否有新的传入的 Kubernetes 事件。如果违反了策略，则 RHACS 执行强制。

其他资源

- 策略标准
- 使用准入控制器强制

6.4.2. 从 risk 视图创建安全策略

在风险视图中评估部署的风险时，当您应用本地页面过滤时，您可以根据您使用的过滤标准创建新的安全策略。

流程

1. 进入 RHACS 门户并从导航菜单中选择 **风险**。
2. 应用您要为其创建策略的本地页面过滤条件。
3. 选择 **New Policy** 并填写所需字段以创建新策略。

其他资源

- 使用本地页面过滤
- 从系统策略视图创建安全策略

6.4.3. 策略标准

在 **Policy Criteria** 部分中，您可以配置要触发策略的数据。

您可以根据下表中列出的属性来配置策略。

在这个表中：

- **Regular expressions**, **AND, OR**, 和 **NOT** 列指示您可以使用正则表达式和其他逻辑运算符以及特定的属性。
  - **!** 用于 **Regex** （正则表达式）表示您只能对列出的字段使用正则表达式。
  - **!** 对于 **AND**，或 **OR** 表示您只能将上述逻辑运算符用于属性。
  - **Regex** / **NOT** / **AND, OR** column 指示属性不支持任何这些(regex、Negation、logical operators)。
- **RHACS 版本** 列指示必须使用该属性的 Red Hat Advanced Cluster Security for Kubernetes 版本。
- 对于满足以下条件的属性，不能使用运算符 **AND** 和 **OR** 的组合：
  - 布尔值 **true** 和 **false**
  - 最低值语义，例如：
    - **最低 RBAC 权限**
    - **自镜像创建以来的天数**
- 您不能将 **NOT** 逻辑运算符用于具有以下属性：
  - 布尔值 **true** 和 **false**
  - 已使用进行比较（如 **<**, **>**, **<=**, **>=** 操作符）的数字值。
  - compound 条件可以有多个值，例如：
    - **Dockerfile 行**，其中包含指令和参数。
    - **环境变量**，由名称和值组成。
  - 其他含义，包括 **Add Capabilities**、**Drop Capabilities**、**自镜像创建后的第一天**，以及 **自镜像上次扫描以来的日期**。

属性	Description	JSON 属性	允许的值	regex,NOT,AND, OR	阶段
部分：镜像 registry					
镜像 Registry	镜像 registry 的名称。	镜像 Registry	字符串	regex, NOT, AND, OR	**构建**, **部署**, **运行时** （与运行时条件一起使用）
镜像名称	registry 中镜像的全名，如 **library/nginx**。	镜像远程	字符串	regex, NOT, AND, OR	**构建**, **部署**, **运行时** （与运行时条件一起使用）

属性	Description	JSON 属性	允许的值	regex,NOT,AND, OR	阶段
镜像标签	镜像的标识符。	镜像标签	字符串	regex, NOT, AND, OR	**构建，部署，运行时** （与运行时条件一起使用）
镜像签名	可用于验证镜像签名的签名集成列表。在没有签名或其签名的镜像上创建警报，至少可以由其中一个提供的签名集成来验证。	镜像签名验证者	已配置镜像签名集成的有效 ID	!仅限 **or** only	**构建，部署，运行时** （与运行时条件一起使用）
部分：镜像内容					
CVE 可修复	只有您在评估的部署中的镜像具有可修复的 CVE 时，此条件才会导致违反情况。	可修复	布尔值	×	**构建，部署，运行时** （与运行时条件一起使用）
自 CVE Was First Discovered In images 中的 daysSince	只有当 RHACS 在特定镜像中发现 CVE 后超过指定天数时，此条件才会导致违反情况。	自 CVE Was First Discovered In images 中的 daysSince	整数	×	**构建，部署，运行时** （与运行时条件一起使用）
因 CVE Was First Discovered in System 的 day Since CVE Was	只有由于 RHACS 在 RHACS 监控的所有部署的镜像中发现 CVE，所以这个条件才会超过指定天数。	因 CVE Was First Discovered in System 的 day Since CVE Was	整数	×	**构建，部署，运行时** （与运行时条件一起使用）
镜像年龄	镜像创建日期起的最小天数。	镜像期限	整数	×	**构建，部署，运行时** （与运行时条件一起使用）

属性	Description	JSON 属性	允许的值	regex,NOT,AND, OR	阶段
镜像扫描年龄	镜像上次扫描后的最小天数。	镜像扫描期限	整数	×	**构建**, **部署**, **运行时** （与运行时条件一起使用）
镜像用户	匹配 Dockerfile 中的 USER 指令。详情请查看 https://docs.docker.com/engine/reference/builder/#user 。	镜像用户	字符串	regex, NOT, AND, OR	**构建**, **部署**, **运行时** （与运行时条件一起使用）
Dockerfile 行	Dockerfile 中的特定行，包括指令和参数。	Dockerfile 行	其中一个：LABEL, RUN, CMD, EXPOSE, ENV, ADD, COPY, ENTRYPOINT, VOLUME, USER, WORKDIR, ONBUILD	!正则表达式只适用于值 AND, OR	**构建**, **部署**, **运行时** （与运行时条件一起使用）
镜像扫描状态	检查镜像是否已扫描。	未扫描的镜像	布尔值	×	**构建**, **部署**, **运行时** （与运行时条件一起使用）
CVSS	通用漏洞评分系统，使用它来匹配分数大于 **>**、小于 **<**，或等于 **=** 指定的 CVSS 的漏洞的镜像。	CVSS	<, >, <=, >= 或 nothing (represents equal to) criu-numpyandnumpy-setuptools 是一个十进制（带有可选部分值的数字）。 示例：>=5, 或 9.5	AND, OR	**构建**, **部署**, **运行时** （与运行时条件一起使用）

属性	Description	JSON 属性	允许的值	regex,NOT,AND, OR	阶段
重要性	基于 CVSS 或供应商的严重性。可以是 Low, Moderate, Important 或 Critical 之一。	重要性	<, >, criu, >= or nothing (represents equal to) criu-criuand iwl-wagon One of: UNKNOWN LOW MODERATE IMPORTANT CRITICAL 示例：>=IMPORTANT, 或 CRITICAL	AND, OR	**构建，部署，运行时** （与运行时条件一起使用）
修复人	修复镜像中标记的漏洞的软件包版本字符串。除了识别漏洞的其他条件外，也可以使用此条件，例如使用 CVE 条件。	修复人	字符串	regex, NOT, AND, OR	**构建，部署，运行时** （与运行时条件一起使用）
CVE	常见的漏洞和风险，将其与特定 CVE 编号一起使用。	CVE	字符串	regex, NOT, AND, OR	**构建，部署，运行时** （与运行时条件一起使用）
镜像组件	镜像中存在的特定软件组件的名称和版本号。	镜像组件	key=value 值是可选的。 如果缺少值，则必须采用 "key=".	regex, AND, OR	**构建，部署，运行时** （与运行时条件一起使用）
镜像操作系统	镜像基础操作系统的名称和版本号。例如，**alpine:3.17.3**	镜像操作系统	字符串	regex, NOT, AND, OR	**构建，部署，运行时** （与运行时条件一起使用）

属性	Description	JSON 属性	允许的值	regex,NOT,AND, OR	阶段
需要镜像标签	确保存在 Docker 镜像标签。如果部署中的任何镜像没有指定标签，则策略会触发。您可以对 key 和 value 字段使用正则表达式来匹配标签。**Require Image Label** 策略条件仅在与 Docker registry 集成时才有效。有关 Docker 标签的详情，请参阅 Docker 文档 https://docs.docker.com/config/labels-custom-metadata/。	所需的镜像标签	key=value 值是可选的。 如果缺少值，则必须采用 "key=".	regex, AND, OR	**构建**，**部署**，**运行时** （与运行时条件一起使用）
不允许镜像标签	确保不使用特定的 Docker 镜像标签。如果部署中的任何镜像具有指定标签，则策略会触发。您可以对 key 和 value 字段使用正则表达式来匹配标签。只有与 Docker registry 集成时，"Disallow Image Label 策略"条件才有效。有关 Docker 标签的详情，请参阅 Docker 文档 https://docs.docker.com/config/labels-custom-metadata/。	不允许的镜像标签	key=value 值是可选的。 如果缺少值，则必须采用 "key=".	regex, AND, OR	**构建**，**部署**，**运行时** （与运行时条件一起使用）

属性	Description	JSON 属性	允许的值	regex,NOT,AND, OR	阶段
部分：容器配置					
环境变量	根据名称或值检查环境变量。	环境变量	RAW=key=value，将部署配置中直接指定的环境变量与特定的键和值匹配。值可以被省略来仅匹配键。 如果配置中没有直接定义环境变量，则可以使用 SOURCE=KEY 格式，其中 SOURCE 是 SECRET_KEY、CONFIG_MAP_KEY、FIELD 或 RESOURCE_FIELD 之一。在这种情况下，条件只能匹配键而不是值。	!正则表达式只适用于键和值（如果使用 RAW） AND, OR	**部署**, **运行时** （与运行时条件一起使用）
容器 CPU 请求	检查为给定资源保留的内核数。	容器 CPU 请求	<, >, numpy, >= 或 nothing （代表等于） criu-numpyandnumpy A decimal A decimal, a optional fractional value） 示例： >=5, 或 9.5	AND, OR	**部署**, **运行时** （与运行时条件一起使用）
容器 CPU 限制	检查允许资源使用的最大内核数。	容器 CPU 限制	（与容器 CPU 请求相同）	AND, OR	**部署**, **运行时** （与运行时条件一起使用）

属性	Description	JSON 属性	允许的值	regex,NOT,AND, OR	阶段
容器内存请求	检查为给定资源保留的内存量。	容器内存请求	（与容器 CPU 请求相同）	AND, OR	**部署**, **运行时** （与运行时条件一起使用）
容器内存限制	检查允许资源使用的最大内存量。	容器内存限制	（与容器 CPU 请求相同）	AND, OR	**部署**, **运行时** （与运行时条件一起使用）
特权容器	特权运行部署。	特权容器	布尔值	×	**部署**, **运行时** （与运行时条件一起使用）
根文件系统写入性	使用 root 文件系统运行的容器配置为只读。	只读 Root 文件系统	布尔值	×	**部署**, **运行时** （与运行时条件一起使用）
seccomp 配置集类型	容器允许的 seccomp 配置集类型。	seccomp 配置集类型	其中之一： UNCONFINED RUNTIME_DEFAULT LOCALHOST	×	**部署**, **运行时** （与运行时条件一起使用）
权限升级	在配置了开发时提供警报，以允许容器进程获得比父进程更多的特权。	允许权限升级	布尔值	×	**部署**, **运行时** （与运行时条件一起使用）

属性	Description	JSON 属性	允许的值	regex,NOT,AND, OR	阶段
drop Capabilities	必须从容器中丢弃的 Linux 功能。在不丢弃指定功能时提供警报。例如，如果配置了 **SYS_ADMIN** 和 **SYS_BOOT**，且部署只丢掉了这两个能力中的一*个或没有丢掉*，警告会发生。	drop Capabilities	其中之一： ALL AUDIT_CONTROL AUDIT_READ AUDIT_WRITE BLOCK_SUSPEND CHOWN DAC_OVERRIDE DAC_READ_SEARCH FSETID IPC_LOCK IPC_OWNER KILL LEASE LINUX_IMMUTABLE LINUX_IMMUTABLE MAC_ADMIN MAC_OVERRIDE MKNOD NET_ADMIN NET_BIND_SERVICE NET_BROADCAST NET_RAW SETGID SETFCAP SETPCAP SETUID SYS_ADMIN SYS_BOOT SYS_CHROOT SYS_MODULE SYS_NICE GIGABYTE SYS_PACCT SYS_PTRACE SYS_RAWIO SYS_RESOURCE SYS_TIME SYS_TTY_CON	和	**部署**，**运行时**（与运行时条件一起使用）

属性	Description	JSON 属性	允许的值	regex,NOT,AND, OR	阶段
添加功能	不得添加到容器中的 Linux 功能，例如发送原始数据包或覆盖文件权限的能力。在添加指定功能时提供警报。例如，如果使用 **NET_ADMIN** 或 **NET_RAW** 配置，且部署清单 YAML 文件至少包含这两个功能之一，则会出现警报。	添加功能	FIG SYSLOG WAKE_ALARM AUDIT_CONTROL AUDIT_READ AUDIT_WRITE BLOCK_SUSPEND CHOWN DAC_OVERRIDE DAC_READ_SEARCH FOWNER FSETID IPC_LOCK IPC_OWNER KILL LEASE LINUX_IMMUTABLE MAC_ADMIN MAC_OVERRIDE MKNOD NET_ADMIN NET_BIND_SERVICE NET_BROADCAST NET_RAW SETGID SETFCAP SETPCAP SETUID SYS_ADMIN SYS_BOOT SYS_CHROOT SYS_MODULE SYS_PACCT CAMELAWSS SYS_PTRACE SYS_RAWIO SYS_RESOURCE SYS_TIME SYS_TTY_CONFIG SYSLOG WAKE_ALARM	或者	**部署**，**运行时**（与运行时条件一起使用）

属性	Description	JSON 属性	允许的值	regex,NOT,AND, OR	阶段
容器名称	容器的名称。	容器名称	字符串	regex, NOT, AND, OR	**部署**, **运行时** （与运行时条件一起使用）
Apparmor Profile	容器中使用的应用程序 Armor ("AppArmor") 配置集。	Apparmor Profile	字符串	regex, NOT, AND, OR	**部署**, **运行时** （与运行时条件一起使用）
存活度（Liveness）探测	容器是否定义了存活度探测。	存活度（Liveness）探测	布尔值	×	**部署**, **运行时** （与运行时条件一起使用）
就绪度（Readiness）探测	容器是否定义了就绪度探测。	就绪度（Readiness）探测	布尔值	×	**部署**, **运行时** （与运行时条件一起使用）
部分：部署元数据					
不允许注解	不允许在指定环境中的 Kubernetes 资源中存在的注解。	不允许注解	key=value 值是可选的。 如果缺少值，则必须采用 "key=".	regex, AND, OR	**部署**, **运行时** （与运行时条件一起使用）
所需的标签	检查 Kubernetes 中是否存在所需的标签。	所需的标签	key=value 值是可选的。 如果缺少值，则必须采用 "key=".	regex, AND, OR	**部署**, **运行时** （与运行时条件一起使用）
必需注解	检查 Kubernetes 中是否存在所需的注解。	必需注解	key=value 值是可选的。 如果缺少值，则必须采用 "key=".	regex, AND, OR	**部署**, **运行时** （与运行时条件一起使用）

属性	Description	JSON 属性	允许的值	regex,NOT,AND, OR	阶段
运行时类	部署的 **RuntimeClass**。	运行时类	字符串	regex, NOT, AND, OR	**部署**, **运行时** （与运行时条件一起使用）
主机网络	检查 **HostNetwork** 是否已启用，这意味着容器没有放置在单独的网络堆栈中（例如，容器的网络没有容器化）。这意味着容器可以完全访问主机的网络接口。	主机网络	布尔值	×	**部署**, **运行时** （与运行时条件一起使用）
主机 PID	检查在容器和主机间是否隔离了进程 ID (PID)命名空间。这允许不同 PID 命名空间中的进程具有相同的 PID。	主机 PID	布尔值	×	**部署**, **运行时** （与运行时条件一起使用）
主机 IPC	检查主机上的 IPC (POSIX/SysV IPC)命名空间（提供命名共享内存片段、semaphores 和消息队列）的隔离是否与容器共享。	主机 IPC	布尔值	×	**部署**, **运行时** （与运行时条件一起使用）
命名空间	部署所属的命名空间的名称。	命名空间	字符串	regex, NOT, AND, OR	**部署**, **运行时** （与运行时条件一起使用）

属性	Description	JSON 属性	允许的值	regex,NOT,AND, OR	阶段
Replicas	部署副本数量。如果使用 **oc scale** 将部署副本从 0 扩展到数字，则准入控制器会在部署违反策略时阻止此操作。	Replicas	<, >, criu, >= 或 nothing （代表等于） criu-numpyandnumpy 一个十进制（带有可选部分值的数字）。 示例： >=5, 或 9.5	NOT, AND, OR	**部署**, **运行时** （与运行时条件一起使用）
部分：存储					
卷名称	存储的名称。	卷名称	字符串	regex, NOT, AND, OR	**部署**, **运行时** （与运行时条件一起使用）
卷源	指明置备卷的表单。例如：**persistentVolumeClaim** 或 **hostPath**。	卷源	字符串	regex, NOT, AND, OR	**部署**, **运行时** （与运行时条件一起使用）
卷目的地	挂载卷的路径。	卷目的地	字符串	regex, NOT, AND, OR	**部署**, **运行时** （与运行时条件一起使用）
卷类型	卷的类型。	卷类型	字符串	regex, NOT, AND, OR	**部署**, **运行时** （与运行时条件一起使用）
挂载的卷不稳定	挂载为可写的卷。	可写挂载的卷	布尔值	×	**部署**, **运行时** （与运行时条件一起使用）

属性	Description	JSON 属性	允许的值	regex,NOT,AND, OR	阶段
挂载传播	检查容器是否在双向中 **挂载卷、主机到容器** 或无 **模式**。	挂载传播	其中之一： NONE HOSTTOCONTAINER 双向	NOT, AND, OR	**部署**, **运行时** （与运行时条件一起使用）
主机挂载不稳定	资源已在主机上挂载了具有写入权限的路径。	可写主机挂载	布尔值	×	**部署**, **运行时** （与运行时条件一起使用）
部分： Networking					
协议	公开端口使用的协议，如 TCP 或 UDP。	公开端口协议	字符串	regex, NOT, AND, OR	**部署**, **运行时** （与运行时条件一起使用）
端口	部署公开的端口号。	公开端口	<, >, criu, >= 或 nothing （代表等于） criu-numpyandcriu-PROFILE 一个整数。 示例： >=1024, 或 22	NOT, AND, OR	**部署**, **运行时** （与运行时条件一起使用）
公开的节点端口	部署外部公开的端口号。	公开的节点端口	（与公开端口相同）	NOT, AND, OR	**部署**, **运行时** （与运行时条件一起使用）
端口公开	服务的暴露方法，如负载均衡器或节点端口。	端口公开方法	其中之一： UNSET EXTERNAL NODE HOST INTERNAL ROUTE	NOT, AND, OR	**部署**, **运行时** （与运行时条件一起使用）

属性	Description	JSON 属性	允许的值	regex,NOT,AND, OR	阶段
期望的意外网络流	检查检测到的网络流量是否是部署的网络基准的一部分。	期望的意外网络流	布尔值	×	仅 **运行时** - 网络
Ingress 网络策略	检查入口 Kubernetes 网络策略是否存在。	具有 Ingress 网络策略	布尔值	regex, AND, OR	**部署**, **运行时** （与运行时条件一起使用）
出口网络策略	检查出口 Kubernetes 网络策略是否存在。	具有 Egress 网络策略	布尔值	regex, AND, OR	**部署**, **运行时** （与运行时条件一起使用）
部分：进程活动					
进程名称	部署中执行的进程的名称。	进程名称	字符串	regex, NOT, AND, OR	仅 **运行时** - 进程
Process Ancestor	部署中执行进程的任何父进程的名称。	Process Ancestor	字符串	regex, NOT, AND, OR	仅 **运行时** - 进程
进程参数	部署中执行进程的命令参数。	进程参数	字符串	regex, NOT, AND, OR	仅 **运行时** - 进程
Process UID	部署期间执行的进程的 UNIX 用户 ID。	Process UID	整数	NOT, AND, OR	仅 **运行时** - 进程
执行意外的进程	检查进程执行没有在部署锁定的进程基准中列出的部署。	执行意外的进程	布尔值	×	仅 **运行时** - 进程
部分：Kubernetes 访问					
服务帐户	服务帐户的名称。	服务帐户	字符串	regex, NOT, AND, OR	**部署**, **运行时** （与运行时条件一起使用）

属性	Description	JSON 属性	允许的值	regex,NOT,AND, OR	阶段
自动挂载服务帐户令牌	检查部署配置是否自动挂载服务帐户令牌。	自动挂载服务帐户令牌	布尔值	×	**部署**, **运行时** （与运行时条件一起使用）
最低 RBAC 权限	如果部署的 Kubernetes 服务帐户等于 **=** 或高于**>**指定的级别，则匹配。	最低 RBAC 权限	其中之一： DEFAULT ELEVATED_IN_NAMESPACE ELEVATED_CLUSTER_WIDE CLUSTER_ADMIN	非	**部署**, **运行时** （与运行时条件一起使用）
部分：Kubernetes 事件					
Kubernetes 操作	Kubernetes 操作的名称，如 **Pod Exec**。	Kubernetes 资源	其中之一： PODS_EXEC PODS_PORTFORWARD	!仅限 **or** only	仅 **运行时** - Kubernetes 事件
Kubernetes 用户名	访问资源的用户的名称。	Kubernetes 用户名	仅限连字符(-)和冒号(:)的字母数字字符	regex, NOT, !仅限 **or** only	仅 **运行时** - Kubernetes 事件
Kubernetes 用户组	访问资源所属用户的组名称。	Kubernetes 用户组	仅限连字符(-)和冒号(:)的字母数字字符	regex, !仅限 **or** only	仅 **运行时** - Kubernetes 事件
Kubernetes 资源	访问的 Kubernetes 资源的类型。	Kubernetes 资源	其中之一： 配置映射 Secret ClusterRole ClusterRoleBindings NetworkPolicies SecurityContextConstraints EgressFirewalls	!仅限 **or** only	仅 **运行时** - 审计日志

属性	Description	JSON 属性	允许的值	regex,NOT,AND, OR	阶段
Kubernetes API Verb	用于访问资源的 Kubernetes API 动词，如 **GET** 或 **POST**。	Kubernetes API Verb	其中之一： CREATE DELETE GET PATCH UPDATE	!仅限 **or** only	仅 **运行时** - 审计日志
Kubernetes 资源名称	访问的 Kubernetes 资源的名称。	Kubernetes 资源名称	仅限连字符(-)和冒号(:)的字母数字字符	regex, NOT, !仅限 **or** only	仅 **运行时** - 审计日志
用户代理	用于访问资源的用户代理。例如，**oc**，或 **kubectl**。	用户代理	字符串	regex, NOT, !仅限 **or** only	仅 **运行时** - 审计日志
源 IP 地址	用户从中访问资源的 IP 地址。	源 IP 地址	IPV4 或 IPV6 地址	regex, NOT, !仅限 **or** only	仅 **运行时** - 审计日志
是 Impersonated User	检查请求是否由服务帐户或某些其他帐户模拟。	是 Impersonated User	布尔值	×	仅 **运行时** - 审计日志

6.4.3.1. 为策略条件添加逻辑条件

您可以使用 drag-and-drop 策略字段面板指定策略标准的逻辑条件。

先决条件

- 您必须使用 Red Hat Advanced Cluster Security for Kubernetes 版本 3.0.45 或更新版本。

流程

1. 在 **Policy Criteria** 部分中，选择 **Add a new condition** 来添加新策略部分。
   - 您可以点 **Edit** 图标重命名 policy 部分。
   - Drag **out a policy 字段** 部分列出了多个类别中的可用策略标准。您可以扩展和折叠这些类别，以查看策略标准属性。
2. 将属性拖到 **Drop a policy 字段** 到 policy 部分的区域。
3. 根据您选择的属性的类型，您可以获得不同的选项来配置所选属性的条件。例如：
   - 如果您选择了带有布尔值 **Read-Only Root Filesystem** 的属性，您会看到 **READ-ONLY** 和 **WRITABLE** 选项。

- 如果您选择了带有复合值 环境变量的属性，您会看到输入 **Key**、**Value** 和 **Value From** 字段的值的选项，以及一个图标，以为可用选项添加更多值。

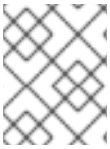
  a. 要组合一个属性的多个值，请点 **Add** 图标。

  b. 您也可以点策略部分中列出的逻辑运算符 **AND** 或 **OR**，以在 **AND** 和 **OR** 运算符之间进行切换。在 Operator 间的切换只在策略部分内工作，而不是在两个不同的策略部分之间工作。

4. 您可以通过重复这些步骤来指定多个 **AND** 和 **OR** 条件。为添加的属性配置条件后，点 **Next** 继续创建策略。

6.5. 共享安全策略

从 Red Hat Advanced Cluster Security for Kubernetes 版本 3.0.44 开始，您可以通过导出和导入策略在不同的 Central 实例间共享您的安全策略。它可帮助您为所有集群强制执行相同的标准。要共享策略，您可以将它们导出为 JSON 文件，然后将其导入回另一个 Central 实例。



注意

目前，您无法使用 RHACS 门户一次导出多个安全策略。但是，您可以使用 API 导出多个安全策略。在 RHACS 门户中，进入 **Help → API 参考** 来查看 API 参考。

6.5.1. 导出安全策略

当您导出策略时，它包括所有策略内容，还包括集群范围、集群排除以及所有配置的通知。

流程

1. 在 RHACS 门户中，进入 **Platform Configuration → Policy Management**。
2. 在 **Policies** 页面中，选择您要编辑的策略。
3. 选择 **Actions → Export policy to JSON**。

6.5.2. 导入安全策略

您可以从 RHACS 门户上 **的系统策略** 视图导入安全策略。

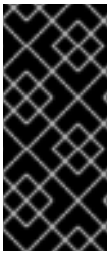
流程

1. 在 RHACS 门户中，进入 **Platform Configuration → Policy Management**。
2. 单击 **Import policy**。
3. 在 **Import policy JSON** 对话框中，点 **Upload** 并选择您要上传的 JSON 文件。
4. 点 **Begin import**。

RHACS 中的每个安全策略都有唯一的 ID (UID)和唯一名称。当您导入策略时，RHACS 会按如下方式处理上传的策略：

- 如果导入的策略 UID 和名称与任何现有策略不匹配，RHACS 会创建一个新策略。
- 如果导入的策略具有与现有策略相同的 UID，但使用不同的名称，您可以：

    - 保留这两个策略。RHACS 使用新的 UID 保存导入的策略。
    - 将现有的策略替换为导入的策略。
- 如果导入的策略具有与现有策略相同的名称，但不同的 UID，您可以：
    - 通过为导入的策略提供新名称来保留这两个策略。
    - 将现有的策略替换为导入的策略。
- 如果导入的策略具有与现有策略相同的名称和 UID，Red Hat Advanced Cluster Security for Kubernetes 会检查策略条件是否符合现有策略。如果策略条件匹配，RHACS 保留现有策略并显示成功消息。如果策略条件不匹配，您可以：
    - 通过为导入的策略提供新名称来保留这两个策略。
    - 将现有的策略替换为导入的策略。

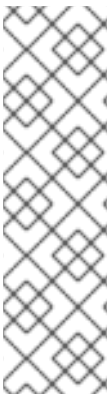


重要

- 如果您导入到同一 Central 实例中，RHACS 将使用所有导出的字段。
- 如果您导入到不同的 Central 实例中，RHACS 会省略某些字段，如集群范围、集群排除和通知。RHACS 在消息中显示这些省略的字段。这些字段因每个安装而异，您不能将它们从一个 Central 实例迁移到另一个安装。

第 7 章 默认安全策略

Red Hat Advanced Cluster Security for Kubernetes 中的默认安全策略提供了广泛的覆盖范围来识别安全问题，并确保您的环境中的安全性最佳实践。通过配置这些策略，您可以自动防止环境中的高风险服务部署，并响应运行时安全事件。



注意

Red Hat Advanced Cluster Security for Kubernetes 中策略的严重性级别与红帽产品安全团队分配的严重性级别不同。

Red Hat Advanced Cluster Security for Kubernetes 策略严重性级别为 Critical, High, Medium, 和 Low。红帽产品安全团队将漏洞严重性等级评级为 Critical、Important、Moderate 和 Low。

虽然策略的严重性级别和红帽产品安全团队可能会进行交互，但必须区分它们。有关红帽产品安全严重性等级的更多信息，请参阅 严重性等级。

7.1. 关键严重性安全策略

下表列出了 Red Hat Advanced Cluster Security for Kubernetes 中具有关键严重性的默认安全策略。策略按照生命周期阶段进行组织。

表 7.1. 关键严重性安全策略

生命周期阶段	名称	Description	状态
构建或部署	Apache Struts: CVE-2017-5638	当部署包含 CVE-2017-5638 Apache Struts 漏洞的镜像时发出警报。	Enabled
构建或部署	Log4Shell: log4j Remote Code Execution vulnerability	当部署包含 CVE-2021-44228 和 CVE-2021-45046 Log4Shell 漏洞的镜像时发出警报。版本 2.0-beta9 - 2.15.0 的 Apache Log4j Java 日志库中存在漏洞，不包括 2.12.2。	Enabled
构建或部署	快速重置：HTTP/2 协议中拒绝服务漏洞	在带有镜像（包含与 HTTP/2 服务器的 Denial Service (DoS)漏洞相关的组件）的部署中的警报。这个问题解决了在 HTTP/2 中处理多路流的缺陷。客户端可以快速创建请求并立即重置请求，从而为服务器创建额外的工作，同时避免达到任何服务器端的限制，从而导致拒绝服务攻击。	Enabled

生命周期阶段	名称	Description	状态
构建或部署	Spring4Shell (Spring Framework Remote Code Execution)和 Spring Cloud Function 漏洞	当部署包含 CVE-2022-22965 漏洞的镜像时发出警报，这会影响 Spring MVC，以及影响 Spring Cloud 的 CVE-2022-22963 漏洞。在版本 3.16、3.2.2 和不受支持的版本中，Spring Cloud 包含漏洞。Spring Framework 版本为 5.3.0 - 5.3.17、版本 5.2.0 - 5.2.19 以及旧的不支持的版本中存在漏洞。	Enabled
Runtime	Privileged Container 中执行的 iptables	特权 pod 运行 iptables 时的警报。	Enabled

7.2. 高严重性安全策略

下表列出了 Red Hat Advanced Cluster Security for Kubernetes 中具有高严重性的默认安全策略。策略按照生命周期阶段进行组织。

表 7.2. 高严重性安全策略

生命周期阶段	名称	Description	状态
构建或部署	可修复 CVSS >= 7	当部署具有可修复的、CVSS 最少为 7 的安全漏洞时发出警报。	Disabled
构建或部署	可修复的严重性至少为重要	当部署具有可修复漏洞的部署时，警报的严重性等级至少为 Important （重要）。	Enabled
构建或部署	在镜像中公开安全 Shell (ssh)端口	当部署公开端口 22 时发出警报，这通常为 SSH 访问保留。	Enabled
部署	紧急部署注解	当部署使用紧急注解时，如 "admission.stackrox.io/break-glass":"ticket-1234" to circumvent StackRox Admission 控制器检查。	Enabled

生命周期阶段	名称	Description	状态
部署	环境变量包含 Secret	当部署具有包含 'SECRET' 的环境变量时发出警报。	Enabled
部署	可修复 CVSS >= 6 和特权	当部署以特权模式运行，带有至少 6 CVSS 的可修复漏洞时发出警报。	在版本 3.72.0 及更高版本中默认禁用
部署	带有重要和关键修复的 CVE 的特权容器	当以特权模式运行的容器具有重要或关键修复漏洞时，会发出警报。	Enabled
部署	Secret 挂载为环境变量	当部署具有作为环境变量挂载的 Kubernetes secret 时发出警报。	Disabled
部署	Secure Shell (ssh)端口公开	当部署公开端口 22 时发出警报，这通常为 SSH 访问保留。	Enabled
Runtime	Cryptocurrency Mining Process Execution	生成 crypto-curcy mining 进程。	Enabled
Runtime	iptables 执行	检测某个人运行 iptables，这是在容器中管理网络状态的已弃用方法。	Enabled
Runtime	Kubernetes Actions: Exec into Pod	当 Kubernetes API 收到一个容器中运行命令的请求时发出警报。	Enabled
Runtime	Linux 组添加执行	检测某人运行 addgroup 或 groupadd 二进制文件来添加 Linux 组。	Enabled
Runtime	Linux 用户添加执行	检测某人运行 useradd 或 adduser 二进制文件来添加 Linux 用户。	Enabled
Runtime	Login Binaries	指明某人尝试登录时。	Disabled
Runtime	网络管理执行	检测某人运行可操作网络配置和管理的二进制文件。	Enabled

生命周期阶段	名称	Description	状态
Runtime	nmap Execution	当某个人在运行时启动容器中的 nmap 进程时发出警报。	Enabled
Runtime	OpenShift: Kubeadmin Secret Accessed	当某人访问 kubeadmin 机密时发出警报。	Enabled
Runtime	密码 Binaries	指明某人尝试更改密码的时间。	Disabled
Runtime	以集群 Kubelet 端点为目标的进程	检测 healthz、kubelet API 或 heapster 端点的滥用。	Enabled
Runtime	以集群 Kubernetes Docker Stats 端点为目标的进程	检测 Kubernetes docker stats 端点的滥用。	Enabled
Runtime	以 Kubernetes 服务端点为目标的进程	检测 Kubernetes Service API 端点的滥用。	Enabled
Runtime	UID 为 0 的进程	当部署包含 UID 0 运行的进程时发出警报。	Disabled
Runtime	Secure Shell Server (sshd)执行	检测运行 SSH 守护进程的容器。	Enabled
Runtime	setuid 进程	使用 setuid 二进制文件，允许人们使用升级的特权运行某些程序。	Disabled
Runtime	影子文件修改	指明某人试图修改影子文件。	Disabled
Runtime	Java 应用程序 shell Spawned	检测何时将 shell （如 bash、csh、sh 或 zsh）作为 Java 应用程序的子进程运行。	Enabled
Runtime	未授权的网络流	为任何位于"alert on anomal anomalous violations"设置基准之外的网络流生成违反情况。	Enabled
Runtime	未授权的进程执行	为 Kubernetes 部署中容器规格未明确允许的任何进程执行生成违反情况。	Enabled

7.3. 中性安全策略

下表列出了 Red Hat Advanced Cluster Security for Kubernetes 中具有中等严重性的默认安全策略。策略按照生命周期阶段进行组织。

表 7.3. 中性安全策略

生命周期阶段	名称	Description	状态
Build	Docker CIS 4.4：确保镜像被扫描并重新构建，使其包含安全补丁	当镜像没有被扫描并重新构建来包括安全补丁时的警报。扫描镜像通常要查找漏洞，重新构建镜像使其包含安全补丁，然后实例化镜像的容器。	Disabled
部署	30 天扫描期限	当部署在 30 天内没有扫描时发出警报。	Enabled
部署	添加了 CAP_SYS_ADMIN 功能	当部署包含使用 CAP_SYS_ADMIN 扩展的容器时发出警报。	Enabled
部署	使用读写根文件系统的容器	当部署包含具有读写根文件系统的容器时发出警报。	Disabled
部署	允许权限升级的容器	当容器可能会以意外的特权运行时发出警报，从而造成安全风险。当容器进程具有超过其父进程超过其父进程时，可能会发生这种情况。容器可以使用意外的特权运行。	Enabled
部署	部署应该至少有一个 Ingress 网络策略	如果部署缺少 Ingress 网络策略，则发出警报。	Disabled
部署	使用外部公开端点部署	检测部署是否有任何通过任何方法进行外部公开的服务。具有集群外公开的服务的部署会带来更高遭受入侵的风险，因为它们可以在集群外访问。此策略提供了一个警报，以便您可以验证在集群外的服务暴露。如果服务只需要集群内通信，请使用 service type ClusterIP。	Disabled

生命周期阶段	名称	Description	状态
部署	Docker CIS 5.1：确保如果适用，启用了 AppArmor 配置集	使用 AppArmor 通过强制一个称为 AppArmor 配置集的安全策略来保护 Linux 操作系统和应用程序。Apparmor 是一个 Linux 应用程序安全系统，默认在某些 Linux 发行版上提供，如 Debian 和 Ubuntu。	Enabled
部署	Docker CIS 5.15：确保主机的进程命名空间没有共享	在容器和主机之间创建进程级别的隔离。进程 ID (PID)命名空间隔离进程 ID 空间，这意味着不同 PID 命名空间中的进程可以具有相同的 PID。	Enabled
部署	Docker CIS 5.16：确保主机的 IPC 命名空间没有共享	当主机上的 IPC 命名空间与容器共享时发出警报。IPC (POSIX/SysV IPC)命名空间分隔命名的共享内存段、semaphores 和消息队列。	Enabled
部署	Docker CIS 5.19 ：确保挂载传播模式没有启用	启用挂载传播模式时的警报。启用挂载传播模式时，您可以使用双向、主机到容器以及 None 模式挂载容器卷。除非明确需要，否则不要使用双向挂载传播模式。	Enabled
部署	Docker CIS 5.21 ：确保默认 seccomp 配置集没有被禁用	当 seccomp 配置集被禁用时发出警报。seccomp 配置集使用允许列表来允许常见的系统调用和阻止所有其他系统调用。	Disabled
部署	Docker CIS 5.7：确保特权端口没有在容器中映射	当特权端口在容器中映射时发出警报。低于 1024 的 TCP/IP 端口号是特权端口。出于安全原因，普通用户和进程无法使用它们，但容器可能会将其端口映射到特权端口。	Enabled

生命周期阶段	名称	Description	状态
部署	Docker CIS 5.9 和 5.20：确保主机的网络命名空间没有共享	共享主机的网络命名空间时的警报。启用 HostNetwork 时，容器不会放置在单独的网络堆栈中，容器的网络不会被容器化。因此，容器可以完全访问主机的网络接口，并启用了共享 UTS 命名空间。UTS 命名空间提供主机名和 NIS 域名之间的隔离，并且它设置主机名和域，这些主机名和域对在该命名空间中运行进程可见。在容器内运行的进程通常需要知道主机名或域名，因此 UTS 命名空间不应与主机共享。	Enabled
部署	没有扫描的镜像	当部署包含未扫描的镜像时发出警报。	Disabled
Runtime	Kubernetes Actions: 端口转发到 Pod	Kubernetes API 收到端口转发请求时的警报。	Enabled
部署	挂载容器运行时套接字	当部署在容器运行时套接字上挂载了卷挂载时，会发出警报。	Enabled
部署	挂载敏感主机目录	当部署挂载敏感主机目录时发出警报。	Enabled
部署	没有指定资源请求或限制	当部署包含没有资源请求和限值的容器时发出警报。	Enabled
部署	Pod 服务帐户令牌自动挂载	通过将默认服务帐户令牌挂载到应用需要与 Kubernetes API 交互的 pod，以保护 pod 默认服务帐户令牌受到攻击。	Enabled
部署	特权容器	当部署包含以特权模式运行的容器时发出警报。	Enabled
Runtime	crontab 执行	检测 crontab 调度的作业编辑器的使用。	Enabled

生命周期阶段	名称	Description	状态
Runtime	检测到 netcat 执行	检测 netcat 在容器中运行的时间。	Enabled
Runtime	OpenShift: Advanced Cluster Security Central Admin Secret Accessed	当某人访问 Red Hat Advanced Cluster Security Central secret 时发出警报。	Enabled
Runtime	OpenShift：由 Impersonated User 访问的 Kubernetes Secret	当某人模拟用户访问集群中的 secret 时发出警报。	Enabled
Runtime	远程文件复制二进制执行	当部署运行远程文件复制工具时发出警报。	Enabled

7.4. 低严重性安全策略

下表列出了 Red Hat Advanced Cluster Security for Kubernetes 中严重性较低的默认安全策略。策略按照生命周期阶段进行组织。

表 7.4. 低严重性安全策略

生命周期阶段	名称	Description	状态
构建或部署	90 天镜像期限	当部署在 90 天内没有更新时发出警报。	Enabled
构建或部署	使用 ADD 命令而不是 COPY	当部署使用 ADD 命令时发出警报。	Disabled
构建或部署	Anlpine Linux Package Manager (apk) in Image	当部署包含 Alpine Linux 软件包管理器(apk)时发出警报。	Enabled
构建或部署	Image 中的 curl	当部署包含 curl 时发出警报。	Disabled
构建或部署	Docker CIS 4.1：确保创建了容器 Has Been 的用户	确保容器以非 root 用户身份运行。	Enabled
构建或部署	Docker CIS 4.7: Alert on Update instructions	确保在 Dockerfile 中不单独使用更新指令。	Enabled
构建或部署	CMD 中指定的不安全	当部署在命令中使用 "insecure" 时发出警报。	Enabled

生命周期阶段	名称	Description	状态
构建或部署	latest 标签	当部署包含使用 'latest' 标签的镜像时发出警报。	Enabled
构建或部署	Red Hat Package Manager in Image	当部署包含红帽、Fedora 或 CentOS 软件包管理系统的组件时发出警报。	Enabled
构建或部署	所需的镜像标签	当部署包含缺少指定标签的镜像时发出警报。	Disabled
构建或部署	Ubuntu Package Manager 执行	检测 Ubuntu 软件包管理系统的使用。	Enabled
构建或部署	镜像中的 Ubuntu Package Manager	当部署包含镜像中 Debian 或 Ubuntu 软件包管理系统的组件时发出警报。	Enabled
构建或部署	Image中的 wget	当部署包含 wget 时的警报.	Disabled
部署	丢弃所有功能	当部署不丢弃所有功能时发出警报。	Disabled
部署	编配器 Secret 卷的使用不正确	当部署使用带有 'VOLUME /run/secrets' 的 Dockerfile 时发出警报。	Enabled
部署	已部署的 Kubernetes 仪表板	当检测到 Kubernetes 仪表板服务时发出警报。	Enabled
部署	必需注解：电子邮件	当部署缺少 'email' 注解时发出警报。	Disabled
部署	必需注解：Owner/Team	当部署缺少 'owner' 或 'team' 注解时发出警报。	Disabled
部署	所需标签：Owner/Team	当部署缺少 'owner' 或 'team' 标签时发出警报。	Disabled
Runtime	alpine Linux Package Manager Execution	当 Alpine Linux 软件包管理器(apk)在运行时运行时发出警报。	Enabled

生命周期阶段	名称	Description	状态
Runtime	chkconfig Execution	检测 ckconfig 服务管理器的使用，该服务通常不会在容器中使用。	Enabled
Runtime	编译器工具执行	在运行时运行编译软件的二进制文件时发出警报。	Enabled
Runtime	Red Hat Package Manager 执行	当红帽、Fedora 或 CentOS 软件包管理器程序在运行时运行时发出警报。	Enabled
Runtime	shell 管理	运行命令以添加或删除 shell 时的警报。	Disabled
Runtime	systemctl Execution	检测 systemctl 服务管理器的使用。	Enabled
Runtime	systemd 执行	检测 systemd 服务管理器的使用。	Enabled

第 8 章 管理网络策略

Kubernetes 网络策略 是一种规范，如何允许 pod 组相互通信和其他网络端点。这些网络策略配置为 YAML 文件。通过只查看这些文件，通常很难确定应用的网络策略是否达到所需的网络拓扑。

Red Hat Advanced Cluster Security for Kubernetes (RHACS)从编配器收集所有定义的网络策略，并提供用来使这些策略更易于使用的工具。

为了支持网络策略强制，RHACS 提供以下工具：

- 网络图
- 网络策略生成器
- 网络策略模拟器
- 构建时网络策略生成器

8.1. 网络图

8.1.1. 关于网络图

网络图提供有关环境中部署、网络流和网络策略的高级信息。

RHACS 处理每个安全集群中的所有网络策略，以显示哪些部署可以相互联系，并可以访问外部网络。它还监控运行部署并跟踪它们之间的流量。您可以在网络图中查看以下项目：

内部实体

它们代表了在部署和属于专用地址空间(RFC 1918 中定义的)之间的连接。如需更多信息，请参阅"涉及内部实体的无效"。

外部实体

这些代表部署与不属于私有地址空间的 IP 地址之间的连接，如 RFC 1918 中定义的。如需更多信息，请参阅网络图中"外部实体和连接"。

网络组件

在顶部菜单中，您可以选择命名空间（由 **NS** 标签表示）和部署（由 **D** 标签表示）以显示所选集群的图形上（由 **CL** 标签表示）。您可以使用下拉列表并选择要过滤的条件来进一步过滤部署，如常见漏洞和暴露(CVE)、标签和镜像。

网络流

您可以为图形选择以下流之一：

活跃流量

选择此默认选项会显示观察到的流量，专注于您选择的命名空间或特定部署。您可以选择显示信息的时间段。

不活跃流

选择这个选项会显示网络策略允许的潜在流，帮助您识别实现更紧密隔离所需的缺少网络策略。您可以选择显示信息的时间段。

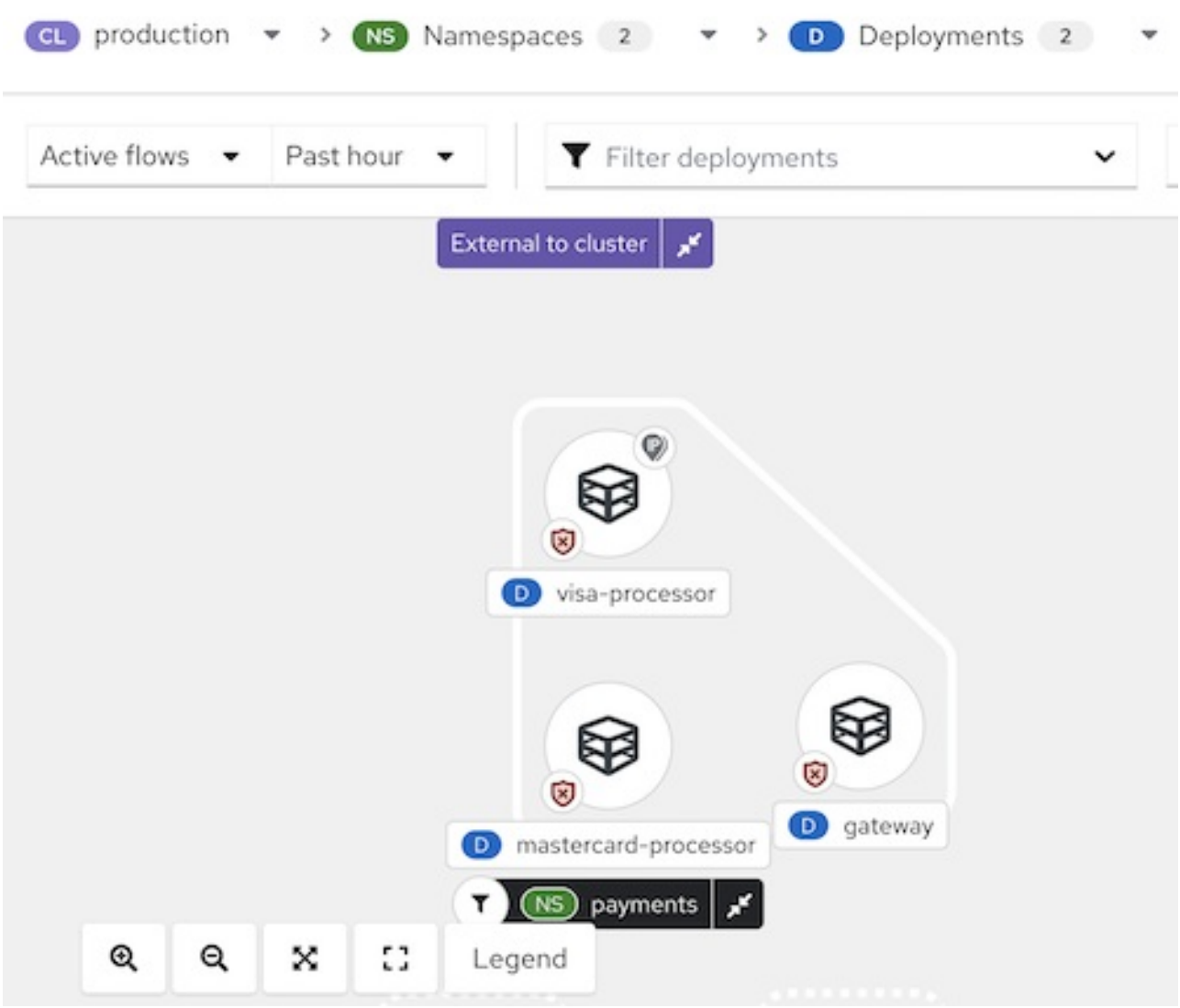
网络策略

您可以查看所选组件的现有策略，或查看没有策略的组件。您还可以从网络图形视图模拟网络策略。如需更多信息，请参阅"从网络图模拟网络策略"。

8.1.1.1. 在网络图中显示、导航和用户界面

您可以使用下图中显示的网络图，单击项目并查看有关它们的额外信息。您还可以在图形中执行操作，如将网络流添加到您的基准中。

图 8.1. 网络图示例



以下提示可帮助您使用网络图：

- 打开图例提供了有关使用中的符号及其含义的信息。图例显示表示网络图中命名空间、部署和连接的符号的说明文本。
- 从下拉列表中选择额外的显示选项，控制图形是否显示图标，如网络策略状态徽标、活动外部流量徽标以及用于边缘连接的端口和协议标签。
- RHACS 检测到网络流量的变化，如加入或离开节点。如果检测到更改，网络图会显示可用更新数量的通知。为了避免中断您的关注，图不会自动更新。点通知来更新图形。

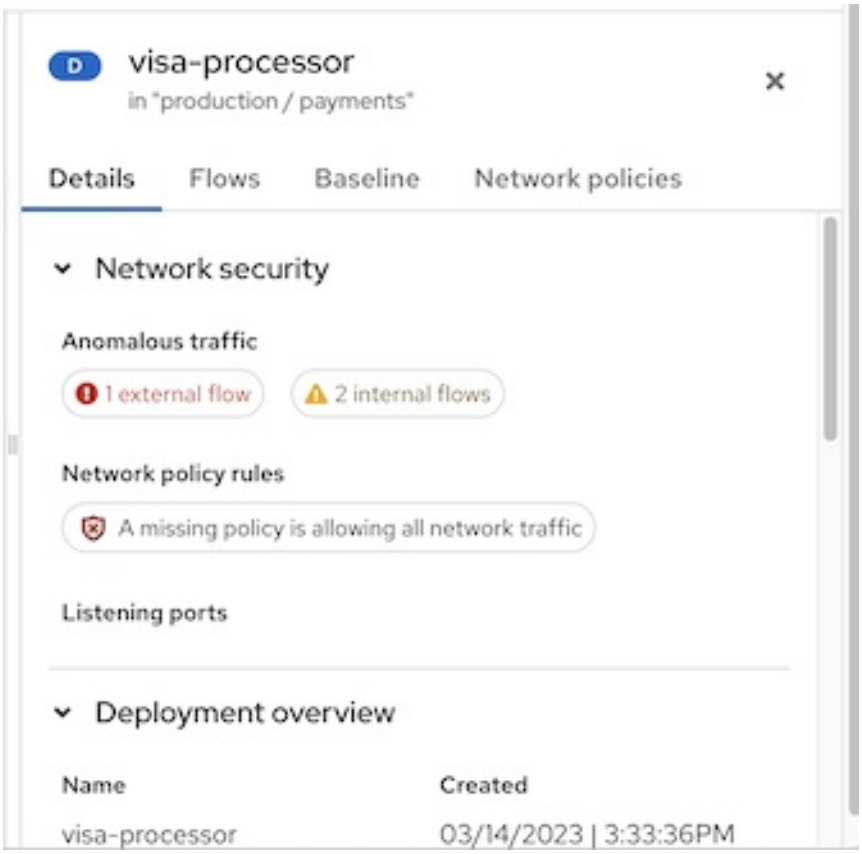
当您点击图形中的项目时，带有 collapsible 部分的 rearranged 侧面板会显示该项目的信息。您可以点以下项目：

- 部署
- 命名空间
- 外部实体

- CIDR 块
- 外部组

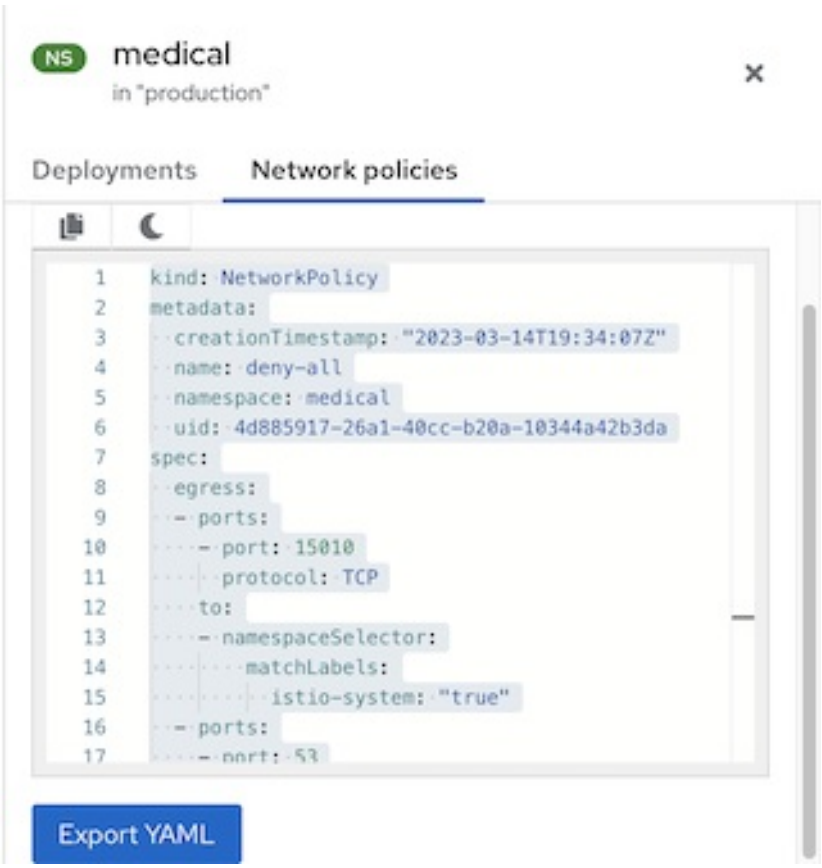
侧面板根据您选择的图形中的项目显示相关信息。标头中项目名称旁边的 **D** 或 **NS** 标签（本例中为"visa-processor"）表示它是部署还是命名空间。以下示例演示了部署的侧面板。

图 8.2. 部署示例的侧面板



查看命名空间时，侧面板包含一个搜索栏和一个部署列表。您可以单击部署来查看其信息。侧面板还包括 **Network policies** 选项卡。在此选项卡中，您可以查看、复制到剪贴板，或导出该命名空间中定义的任何网络策略，如下例所示。

图 8.3. 一个命名空间的侧面板示例



8.1.1.2. 网络图中的外部实体和连接

网络图视图显示受管集群和外部源之间的网络连接。另外，RHACS 会自动发现并突出显示公共无类别域间路由(CIDR)地址块，如 Google Cloud、AWS、Microsoft Azure、Oracle Cloud 和 Cloudflare。使用此信息，您可以识别具有活跃外部连接的部署，并决定是否从网络外部创建或接收未授权连接。

默认情况下，外部连接指向一个通用的 **外部实体** 图标和网络图中的不同 CIDR 地址块。但是，您可以通过点 **Manage CIDR 块并取消选择自动发现的 CIDR 块** 来选择不会显示 **自动发现的 CIDR** 块。

RHACS 包括以下云供应商的 IP 范围：

- Google Cloud
- AWS
- Microsoft Azure
- Oracle Cloud
- Cloudflare

RHACS 每 7 天获取和更新云供应商的 IP 范围，并每天更新 CIDR 块。如果您使用离线模式，可以通过安装新的支持软件包来更新这些范围。

下图提供了一个网络图示例。在本例中，根据用户选择的选项，图形描述了所选命名空间中的部署。在点部署等项目前，不会显示流量流。该图使用红色徽标来指示缺少策略的部署，从而允许所有网络流量。

8.1.1.3. 涉及内部实体的连接

网络图可用于识别与不属于任何已知部署或 CIDR 块的实体的活跃连接的部署。其中一些连接永远不会连接到集群外，并在集群的专用网络内进行。网络图表示它们作为与内部实体的连接或来自 *内部实体*。

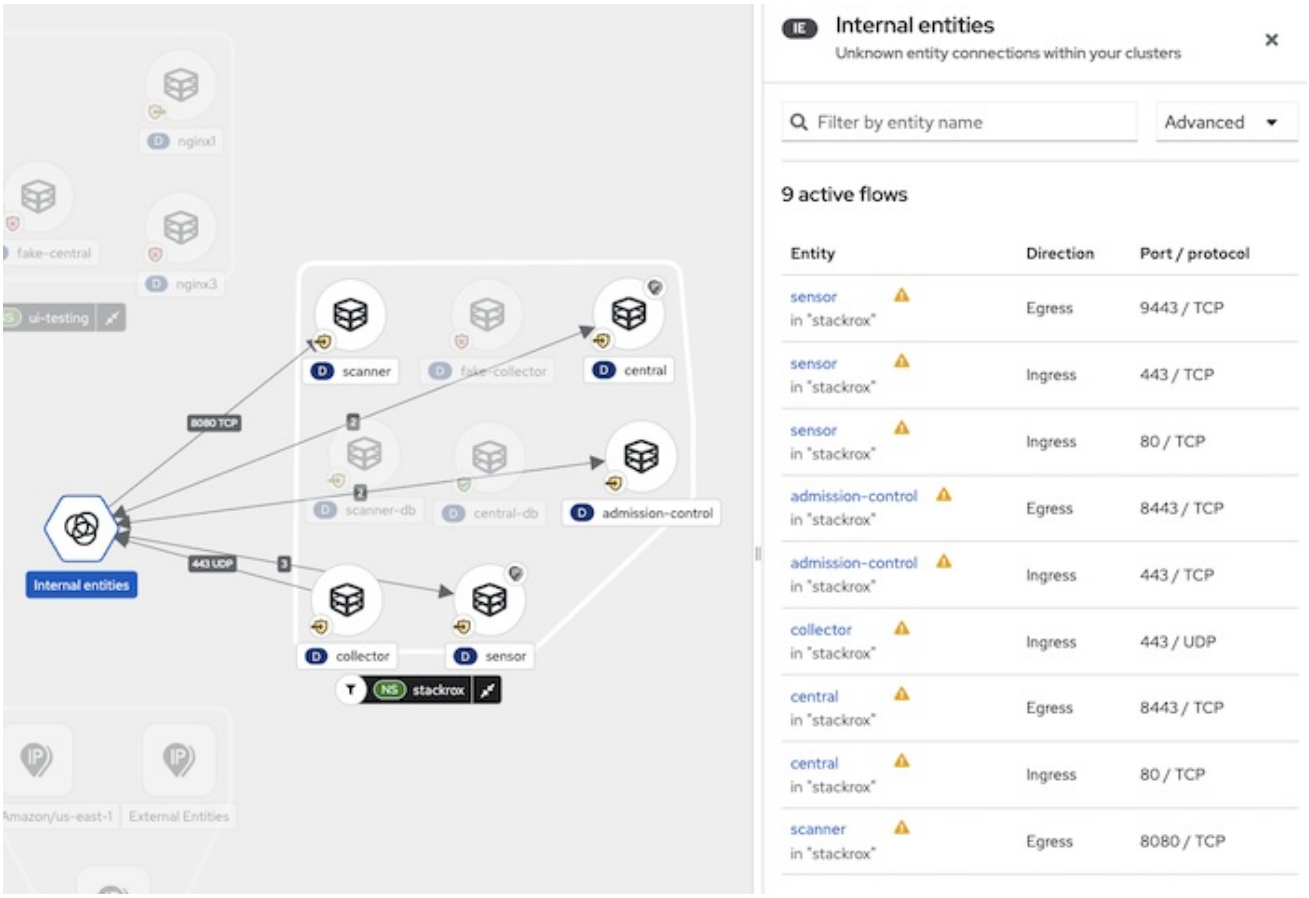
与内部实体的连接代表部署和属于私有地址空间的 IP 地址的连接，如 RFC 1918 中定义的。在某些情况下，Sensor 无法识别连接中涉及的一个或多个部署。在这种情况下，系统会分析 IP 地址，并确定连接是内部还是外部。

以下场景可能会导致连接被归类为涉及内部实体之一：

- IP 地址更改或删除接受连接（服务器）的方启动连接（客户端）仍然会尝试到达它
- 与编配器 API 通信的部署
- 使用网络 CNI 插件进行通信的部署，如 Calico
- 重启 Sensor，从而导致将 IP 地址映射到过去部署，例如当 Sensor 无法识别过去实体的 IP 地址或现有实体的过去 IP 地址时，

内部实体通过图标表示，如下图所示。点 **Internal entities** 显示这些实体的流。

图 8.4. 内部实体示例



8.1.2. 访问控制和权限

要查看网络图形，用户必须至少具有授予 **Network Graph Viewer**默认权限集的权限。

为 **Network Graph Viewer**权限集授予以下权限：

- 读取 **部署**
- read **NetworkGraph**

- 读取 **NetworkPolicy**

如需更多信息，请参阅"添加资源"部分中的"系统权限集"。

其他资源

- 系统权限集

8.1.3. 查看部署信息

网络图提供了 RHACS 发现的部署、命名空间和连接的可视化映射。点击图形中的部署，您可以查看部署的信息，包括以下详情：

- 网络安全性，如流数、现有或缺少网络策略规则以及监听端口
- 标签和注解
- 端口配置
- 容器信息
- 入口和出口连接的异常和基准流，包括协议和端口号
- 网络策略

流程

查看命名空间中的部署详情：

1. 在 RHACS 门户中，进入 **Network Graph** 并从下拉列表中选择您的集群。
2. 点 **Namespaces** 列表，并使用搜索字段找到命名空间，或选择单独的命名空间。
3. 单击 **Deployments** 列表，并使用搜索字段查找部署，或者选择要在网络图中显示的单个部署。
4. 在网络图中，点部署来查看信息面板。
5. 点 **Details**,**Flows**,**Baseline**, 或 **Network policies** 标签页查看对应的信息。

8.1.4. 在网络图中查看网络策略

网络策略指定允许 pod 组相互通信和其他网络端点。Kubernetes **NetworkPolicy** 资源使用标签来选择 pod，并定义允许哪些流量从所选 pod 或所选 pod 的流量的规则。RHACS 在网络图中发现并显示所有 Kubernetes 集群、命名空间、部署和 pod 的网络策略信息。

流程

1. 在 RHACS 门户中，进入 **Network Graph** 并从下拉列表中选择您的集群。
2. 点 **Namespaces** 列表并选择单独的命名空间，或使用搜索字段找到命名空间。
3. 点 **Deployments** 列表并选择单个部署，或使用 search 字段来定位部署。
4. 在网络图中，点部署来查看信息面板。
5. 在 **Details** 选项卡中，在 **Network security** 部分中，您可以查看提供以下信息的网络策略规则的摘要信息：

- 管理入口或出口流量的网络中是否存在策略
- 如果您的网络缺少策略，因此允许所有入口或出口流量

6. 要查看网络策略的 YAML 文件，您可以点策略规则，或者点击 **Network policies** 选项卡。

8.1.5. 在网络图中配置 CIDR 块

您可以指定自定义 CIDR 块，或者在网络图中配置自动发现的 CIDR 块的显示。

流程

1. 在 RHACS 门户中，进入 **Network Graph**，然后选择 **Manage CIDR Blocks**。您可以执行以下操作：
   - 切换 **自动发现的 CIDR 块**，以便在网络图中隐藏自动发现的 CIDR 块。

     注意

     当您隐藏自动发现的 CIDR 块时，所有集群都会隐藏自动发现的 CIDR 块，而不仅适用于网络图中所选集群。

   - 通过执行以下步骤在图形中添加自定义 CIDR 块：
     a. 在字段中输入 CIDR 名称和 CIDR 地址。要添加额外的 CIDR 块，点 **Add CIDR 块** 并为每个块输入信息。
     b. 单击 **Update Configuration** 以保存更改。

8.2. 使用网络图生成和模拟网络策略

8.2.1. 关于从网络图形生成策略

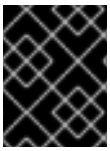
Kubernetes 网络策略控制哪些 pod 接收传入的网络流量，以及哪些 pod 可以发送传出流量。通过使用网络策略来启用和禁用 pod 的流量，您可以限制网络攻击面。

这些网络策略是 YAML 配置文件。通常很难深入了解网络流，并手动创建这些文件。您可以使用 RHACS 生成这些文件。当您自动生成网络策略时，RHACS 遵循以下准则：

- RHACS 为命名空间中的每个部署生成一个网络策略。策略的 pod 选择器是部署的 pod 选择器。
  - 如果部署已有网络策略，RHACS 不会生成新策略或删除现有策略。
    生成的策略只将流量限制到现有部署。
  - 您稍后创建的部署不会有任何限制，除非您为它们创建或生成新的网络策略。
  - 如果新部署需要与具有网络策略的部署联系，您可能需要编辑网络策略以允许访问。
- 每个策略的名称与部署名称相同，前缀为 **stackrox-generated-**。例如，生成的网络策略中的部署 **depABC** 的策略名称为 **stackrox-generated-depABC**。所有生成的策略也都有一个标识标签。
- 如果满足以下条件之一，RHACS 会生成一条规则，允许从任何 IP 地址的流量：
  - 部署在所选时间内具有来自集群外部的传入连接

    - 部署通过节点端口或负载均衡器服务公开

- RHACS 为每个部署生成一个 **入口** 规则，从中有传入连接。
    - 对于同一命名空间中的部署，此规则使用来自其他部署的 pod 选择器标签。
    - 对于不同命名空间中的部署，此规则使用命名空间选择器。要实现此目的，RHACS 会自动将标签 **namespace.metadata.stackrox.io/name** 添加到每个命名空间中。



重要

在个别情况下，如果独立 pod 没有任何标签，则生成的策略允许来自或到 pod 的整个命名空间的流量。

8.2.2. 在网络图中生成网络策略

RHACS 可让您根据环境中实际观察到的网络通信流自动生成网络策略。

您可以根据您在网络图形中选择的集群、命名空间和部署生成策略。为当前网络 Graph 范围中包含的任何部署生成策略。例如，当前范围可能包括整个集群、集群和命名空间，或者在所选命名空间中单独选择的部署。您还可以通过应用 **Filter deployments** 字段中的其中一个过滤器，使用集群、命名空间和部署选择的任意组合来进一步减少范围。例如，您可以将特定集群和命名空间中的部署范围缩小到受特定 CVE 影响。策略由基准发现期间观察到的流量生成。

1. 在 RHACS 门户中，进入 **网络图**。
2. 选择一个集群，然后选择一个或多个命名空间。
3. 可选： 选择单独的部署来限制生成的策略仅那些部署。您还可以使用 **Filter 部署功能** 进一步缩小范围。
4. 在网络图标头中，选择 **Network 策略生成器**。
5. 可选：在打开的信息面板中，选择 **Exclude 端口 & 协议**，以便在从基准生成网络策略时删除端口/协议限制。
   例如，**nginx3** 部署与 **nginx4** 形成端口 80 连接，这包含在 **nginx4** 的基准中。如果生成了策略，并且未选择此复选框（默认行为），生成的策略会将允许从 **nginx3** 到 **nginx4** 的连接限制为仅端口 80。如果选择了此选项生成策略，生成的策略将允许从 **nginx3** 连接到 **nginx4** 连接中的任何端口。
6. 点 **Generate and simulate network policies**。RHACS 为您选择的范围生成策略。此范围显示在 **Generate network policies** 面板的顶部。

注意

单击范围内的部署信息，可显示包含的部署列表。

7. 可选：将生成的网络策略配置 YAML 文件复制到剪贴板，或通过点击面板中的下载图标下载它。
8. 可选： 要将生成的网络策略与现有网络策略进行比较，请点击 **Compare**。现有和生成的网络策略的 YAML 文件显示在并排的视图中。

注意

有些项目没有生成策略，如具有特定保护命名空间中的现有入口策略或部署的命名空间，如 **stackrox** 或 **acs**。

9. 可选：点击 **Actions** 菜单执行以下操作：

   - 使用通知程序共享 YAML 文件：将 YAML 文件发送到您配置的系统通知程序之一，如 Slack、ServiceNow 或使用通用 Webhook 的应用程序。这些通知程序通过导航到 **Platform Configuration → Integrations** 进行配置。如需更多信息，请参阅"附加资源"部分中的文档。
   - 从活跃流量重建规则：刷新生成的策略。
   - 将规则恢复到之前应用的 YAML：删除模拟策略并恢复到最后一个网络策略。

8.2.3. 在网络图中保存生成的策略

您可以从 RHACS 下载并保存生成的网络策略。使用此选项下载策略，以便您可以将策略提交到 Git 等版本控制系统中。

流程

- 生成网络策略后，点 **Network Policy Simulator** 面板中的 **Download YAML** 图标。

8.2.4. 在网络图中测试生成的策略

下载 RHACS 生成的网络策略后，您可以使用 CLI 或自动部署过程将其应用到集群来测试它们。您不能直接在网络图中应用生成的网络策略。

流程

1. 要使用保存的 YAML 文件创建策略，请运行以下命令：

```
$ oc create -f "<generated_file>.yml" 1
```

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

2. 如果生成的策略造成问题，您可以通过运行以下命令来删除它们：

```
$ oc delete -f "<generated_file>.yml" 1
```

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

警告

直接应用网络策略可能会导致运行应用程序出现问题。在将网络策略应用到生产工作负载之前，始终在开发环境或测试集群中下载并测试网络策略。

8.2.5. 在网络图中恢复到之前应用的策略

您可以删除策略并恢复到之前应用的策略。

流程

1. 在 RHACS 门户中，进入 **网络图**。
2. 从顶栏的菜单中选择集群名称。
3. 选择一个或多个命名空间和部署。
4. 选择 **Simulate 网络策略**。
5. 选择 **View active YAMLS**。
6. 在 **Actions** 菜单中，选择 **Revert 规则到之前应用的 YAML**。

警告

直接应用网络策略可能会导致运行应用程序出现问题。在将网络策略应用到生产工作负载之前，始终在开发环境或测试集群中下载并测试网络策略。

8.2.6. 删除网络图中自动生成的所有策略

您可以使用 RHACS 从您创建的集群中删除所有自动生成的策略。

流程

- 运行以下命令：

```
$ oc get ns -o jsonpath='{.items[*].metadata.name}' | \
xargs -n 1 oc delete networkpolicies -l \
'network-policy-generator.stackrox.io/generated=true' -n 1
```

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

8.2.7. 从网络图模拟网络策略

您当前的网络策略可能允许不需要的网络通信。您可以使用网络策略生成器创建网络策略，将入口流量限制为一组部署的计算基准。

注意

Network Graph 不会在视觉化中显示生成的策略。生成的策略仅适用于限制出口流量的入口流量和策略。

流程

1. 在 RHACS 门户中，进入 **网络图**。
2. 选择一个集群，然后选择一个或多个命名空间。
3. 在网络图标头中选择 **Network 策略生成器**。
4. 可选：要生成带有要在模拟中使用的网络策略的 YAML 文件，请单击 **Generate and simulate network policies**。如需更多信息，请参阅"在网络图中生成网络策略"。
5. 上传要在模拟中使用的网络策略的 YAML 文件。网络图视图显示您提议的网络策略将达到什么。执行以下步骤：
   a. 点 **Upload YAML**，然后选择 文件。
   b. 点 **Open**。系统会显示一条消息，以指示上传的策略的处理状态。
6. 您可以通过点 **View active YAMLS** 选项卡来查看与当前网络策略对应的活跃 YAML 文件，然后从下拉列表中选择策略。您还可以执行以下操作：
   - 点适当的按钮复制或下载显示的 YAML 文件。
   - 使用 **Actions** 菜单从活跃流量重建规则，或将规则恢复到之前应用的 YAML。如需更多信息，请参阅"在网络图中生成网络策略"。

其他资源

- 以离线模式更新内核支持软件包
- 使用通用 Webhook 集成

8.3. 关于网络图中的网络基础

在 RHACS 中，您可以使用网络基础来最小化风险。这是保持基础架构安全的主动方法。RHACS 首先发现现有的网络流并创建基准，然后将这个基准之外的网络流视为异常。

安装 RHACS 时，没有默认网络基准。当 RHACS 发现网络流时，它会创建一个基准，然后它会将所有发现的网络流添加到其中，遵循以下准则：

- 当 RHACS 发现新的网络活动时，它会将该网络流添加到网络基准中。
- 网络流没有显示为异常流，且不会触发任何违反情况。

在发现阶段后，会执行以下操作：

- RHACS 停止在网络基准中添加网络流。
- 不在网络基准中的新网络流显示为异常流，但它们不触发任何违反情况。

8.3.1. 从网络图查看网络基准

您可以从网络图形视图查看网络基准。

流程

1. 点 **Namespaces** 列表，并使用搜索字段找到命名空间，或选择单独的命名空间。

2. 单击 **Deployments** 列表，并使用搜索字段查找部署，或者选择要在网络图中显示的单个部署。
3. 在网络图中，点部署来查看信息面板。
4. 选择 **Baseline** 选项卡。使用 **根据实体名称字段** 的过滤器来进一步限制显示的流。
5. 可选：您可以通过执行以下操作之一将基准流标记为异常：
   - 选择单个实体。点 overflow 菜单  ，然后选择 **Mark as anomalous**。
   - 选择多个实体，然后单击 **Bulk 操作**，然后选择 **Mark as anomalous**。
6. 可选：选中用于排除端口和协议的框。
7. 可选： 要将基准保存为网络策略 YAML 文件，请点击 **Download baseline 作为网络策略**。

8.3.2. 从网络图下载网络基准

您可以从网络图形视图下载网络基准作为 YAML 文件。

流程

1. 在 RHACS 门户中，进入 **网络图**。
2. 点 **Namespaces** 列表，并使用搜索字段找到命名空间，或选择单独的命名空间。
3. 单击 **Deployments** 列表，并使用搜索字段查找部署，或者选择要在网络图中显示的单个部署。
4. 在网络图中，点部署来查看信息面板。
5. **Baseline** 选项卡列出了基准流。使用 **根据实体名称字段** 的过滤器来进一步限制流列表。
6. 可选：选中用于排除端口和协议的框。
7. 点 **Download baseline 作为网络策略**。

8.3.3. 配置网络基础时间线

您可以使用 **ROX_NETWORK_BASELINE_OBSERVATION_PERIOD** 和 **ROX_BASELINE_GENERATION_DURATION** 环境变量来配置观察周期和网络基准生成持续时间。

流程

1. 运行以下命令，设置 **ROX_NETWORK_BASELINE_OBSERVATION_PERIOD** 环境变量：

```
$ oc -n stackrox set env deploy/central \ (1)
  ROX_NETWORK_BASELINE_OBSERVATION_PERIOD=<value> (2)
```

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

(2) 值必须是时间单位，例如 **：300ms**、**-1.5h** 或 **2h45m**。有效的时间单位是 **ns**、**us** 或 crius **,ms,s** ,**m**,**h**。

2. 运行以下命令，设置 **ROX_BASELINE_GENERATION_DURATION** 环境变量：

```
$ oc -n stackrox set env deploy/central \ 1
  ROX_BASELINE_GENERATION_DURATION=<value> 2
```

1. 如果使用 Kubernetes，请输入 **kubectl** 而不是 **oc**。
2. 值必须是时间单位，例如 **：300ms**、**-1.5h** 或 **2h45m**。有效的时间单位是 **ns**、**us** 或 crius **,ms,s** ,**m**,**h**。

8.3.4. 在网络图中启用基准违反情况的警报

您可以配置 RHACS 来检测异常网络流，并为不在基准中的流量触发违反情况。这有助于您在使用网络策略阻止流量前确定网络是否包含不需要的流量。

流程

1. 点 **Namespaces** 列表，并使用搜索字段找到命名空间，或选择单独的命名空间。
2. 单击 **Deployments** 列表，并使用搜索字段查找部署，或者选择要在网络图中显示的单个部署。
3. 在网络图中，点部署来查看信息面板。
4. 在 **Baseline** 选项卡中，您可以查看基准流。使用 **根据实体名称字段** 的过滤器来进一步限制显示的流。
5. 在 **基准违反选项上切换 Alert**。
   - 在切换 **Alert on baseline violations** 选项后，异常网络流会触发违反情况。
   - 您可以再次切换 **Alert on baseline violations** 选项，以停止接收异常网络流的违反情况。

第 9 章 构建时网络策略工具

build-time 网络策略工具可让您使用 **roxctl** CLI 在开发和操作工作流中自动创建和验证 Kubernetes 网络策略。这些工具可用于包含项目工作负载和网络策略清单的指定文件目录，且不需要 RHACS 身份验证。

表 9.1. 网络策略工具

命令	描述
roxctl netpol generate	通过分析指定目录中的项目的 YAML 清单来生成 Kubernetes 网络策略。如需更多信息，请参阅使用构建时网络策略生成器。
roxctl netpol 连接映射	通过检查工作负载和 Kubernetes 网络策略清单，列出项目目录中工作负载间允许的连接。您可以使用各种文本格式或图形 **.dot** 格式生成输出。如需更多信息，请参阅使用 roxctl netpol connectivity map 命令的连接映射。
roxctl netpol 连接 diff	在允许两个项目版本之间的连接中创建一系列变化。这由每个版本的 目录中的工作负载和 Kubernetes 网络策略清单决定。此功能显示执行源代码(syntactic) **diff** 时的语义差异。如需更多信息，请参阅 识别项目版本之间的允许连接的不同。

9.1. 使用构建时网络策略生成器

构建时网络策略生成器可以根据应用程序 YAML 清单自动生成 Kubernetes 网络策略。在在集群中部署应用程序前，您可以使用它来开发网络策略，作为持续集成/持续部署(CI/CD)管道的一部分。

红帽开发了此功能，与 NP-Guard 项目 开发人员兼容。首先，build-time 网络策略生成器分析本地文件夹中的 Kubernetes 清单，包括服务清单、配置映射和工作负载清单，如 **Pod**、**Deployment**、**ReplicaSet**、**Job**、**DaemonSet** 和 **StatefulSet**。然后，它会发现所需的连接，并创建 Kubernetes 网络策略来实现 pod 隔离。这些策略不允许超过所需的入口和出口流量。

9.1.1. 生成构建时网络策略

build-time 网络策略生成器包含在 **roxctl** CLI 中。对于构建网络策略生成功能，**roxctl** CLI 不需要与 RHACS Central 通信，因此您可以在任何开发环境中使用它。

先决条件

1. build-time 网络策略生成器递归扫描您在运行命令时指定的目录。因此，在运行该命令前，您必须已具有服务清单、配置映射和工作负载清单，如 **Pod**、**Deployment**、**ReplicaSet**、**Job**、**DaemonSet** 和 **StatefulSet** 作为指定目录中的 YAML 文件。
2. 使用 **kubectl apply -f** 命令验证这些 YAML 文件是否按原样应用。build-time 网络策略生成器不适用于使用 Helm 样式模板的文件。
3. 验证服务网络地址没有硬编码。需要连接到服务的每个工作负载都必须将服务网络地址指定为变量。您可以使用工作负载的资源环境变量或配置映射来指定此变量。

- 示例 1：使用环境变量
- 示例 2：使用配置映射
- 示例 3：使用配置映射

4. 服务网络地址必须与以下官方正则表达式模式匹配：

```
(http(s)?://)?<svc>(.<ns>(.svc.cluster.local)?)?(:<portNum>)? 1
```

1 在这种模式中，

- <svc> 是服务名称。
- <ns> 是定义该服务的命名空间。
- <portNum> 是公开的服务端口号。

以下是与模式匹配的一些示例：

- **wordpress-mysql:3306**
- **redis-follower.redis.svc.cluster.local:6379**
- **redis-leader.redis**
- **http://rating-service.**

流程

1. 运行 help 命令验证构建网络策略生成功能是否可用：

```
$ roxctl netpol generate -h
```

2. 使用 **netpol generate** 命令生成策略：

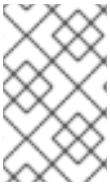
```
$ roxctl netpol generate <folder_path> [flags] 1
```

1 指定文件夹的路径，其中可以包含用于分析的 YAML 资源的子目录。命令扫描整个子文件夹树。另外，您还可以指定参数来修改命令的行为。

有关可选参数的更多信息，请参阅 roxctl netpol generate 命令选项。

后续步骤

- 生成策略后，您必须检查它们以确保完整性和准确性，以防 YAML 文件中未按预期指定任何相关网络地址。
- 最重要的是，验证所需的连接是否没有被隔离策略阻止。为了帮助进行这个检查，您可以使用 **roxctl netpol connectivity map** 工具。



注意

作为工作负载部署的一部分，使用自动化将网络策略应用到集群可节省时间并确保准确性。您可以使用拉取请求提交生成的策略来遵循 GitOps 方法，为团队提供在部署作为管道的一部分前查看策略的机会。

9.1.2. roxctl netpol generate 命令选项

roxctl netpol generate 命令支持以下选项：

选项	描述
-h, --help	查看 **netpol** 命令的帮助文本。
-d, --output-dir <dir>	将生成的策略保存到目标文件夹中。每个策略有一个文件。
-f, --output-file <filename>	将生成的策略保存并合并到单个 YAML 文件中。
--fail	在第一次遇到的错误时失败。默认值为 **false**。
--remove	删除输出路径（如果已存在）。
--strict	将警告视为错误。默认值为 **false**。

9.2. 使用 ROXCTL NETPOL CONNECTIVITY MAP 命令的连接映射

连接映射根据 Kubernetes 清单中定义的网络策略提供不同工作负载间允许的连接的详情。您可以视觉化并了解 Kubernetes 环境中的不同工作负载如何允许根据您设置的网络策略相互通信。

要检索连接映射信息，**roxctl netpol connectivity map** 命令需要一个包含 Kubernetes 工作负载和网络策略清单的目录路径。输出详细介绍了 Kubernetes 资源分析中的连接详情。

9.2.1. 从 Kubernetes 清单目录检索连接映射信息

流程

- 运行以下命令来检索连接映射信息：

```
$ roxctl netpol connectivity map <folder_path> [flags] 1
```

1 指定文件夹的路径，其中可以包含用于分析的 YAML 资源和网络策略的子文件夹，如 **netpol-analysis-example-minimal/**。命令扫描整个子文件夹树。另外，您还可以指定参数来修改命令的行为。

有关可选参数的更多信息，请参阅 roxctl netpol connectivity map 命令选项。

例 9.1. 输出示例

src	dst	conn
0.0.0.0-255.255.255.255	default/frontend[Deployment]	TCP 8080
default/frontend[Deployment]	0.0.0.0-255.255.255.255	UDP 53
default/frontend[Deployment]	default/backend[Deployment]	TCP 9090

输出显示包含允许连接行列表的表。每个连接行由三个部分组成：source (**src**)、destination (**dst**)和允许的连接属性(**conn**)。

您可以将 **src** 解释为源端点，**dst** 作为目标端点，**conn** 作为允许的连接属性。端点具有 **namespace/name[Kind]** 格式，如 **default/backend[Deployment]**。

9.2.2. 连接映射输出格式和视觉化

您可以使用各种输出格式，包括 **txt**、**md**、**csv**、**json** 和 **dot**。**点** 格式是以连接图形方式显示输出的理想选择。它可以使用图形视觉化软件（如 Graphviz 工具 ）查看，并 扩展到 VSCode。您可以使用 Graphviz 将 **点** 输出转换为格式，如 **svg**、**jpeg** 或 **png**，无论是本地安装还是通过在线查看器安装。

9.2.3. 使用 Graphviz 从点输出中生成 svg 图形

按照以下步骤，从 **点** 输出中以 **svg** 格式创建图形。

先决条件

- Graphviz 安装在本地系统中。

流程

- 运行以下命令，以 **svg** 格式创建图形：

```
$ dot -Tsvg connlist_output.dot > connlist_output_graph.svg
```

以下是点输出示例以及由 Graphviz 生成的图形：

  - 示例 1：点输出
  - 示例 2： Graphviz 生成的 Graph

9.2.4. roxctl netpol connectivity map 命令选项

roxctl netpol connectivity map 命令支持以下选项：

选项	描述
--fail	在第一次遇到的错误时失败。默认值为 **false**。
--focus-workload string	专注于输出中指定工作负载名称的连接。
-h，--help	查看 **roxctl netpol connectivity map** 命令的帮助文本。
-f,--output-file string	将连接列表输出保存到特定文件中。
-o,--output-format string	配置输出格式。支持的格式有 **txt**、**json**、**md**、**点** 和 **csv**。默认值为 **txt**。
--remove	删除输出路径（如果已存在）。默认值为 **false**。
--save-to-file	将连接列表输出保存到默认文件中。默认值为 **false**。
--strict	将警告视为错误。默认值为 **false**。

9.3. 识别项目版本之间允许的连接的不同

此命令帮助您了解两个项目版本之间的允许连接差异。它分析了每个版本目录中的工作负载和 Kubernetes 网络策略清单，并以文本格式创建区别。

您可以使用各种输出格式查看连接差异报告，包括 **文本**、**md** 和 **csv**。

9.3.1. 使用 roxctl netpol connection diff 命令生成连接差异报告

要生成连接差异报告，**roxctl netpol connectivity diff** 命令需要两个文件夹 **dir1** 和 **dir2**，每个包含 Kubernetes 清单，包括网络策略。

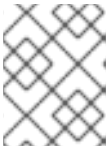
流程

- 运行以下命令，以确定指定目录中的 Kubernetes 清单之间的连接差异：

```
$ roxctl netpol connectivity diff --dir1=<folder_path_1> --dir2=<folder_path_2> [flags] 1
```

1 指定文件夹的路径，其中可以包含用于分析的 YAML 资源和网络策略的子文件夹。命令可扫描两个目录的整个子文件夹树。例如，&lt **;folder_path_1>** 是 **netpol-analysis-example-minimal/**，& **lt;folder_path_2&gt**; 是 **netpol-diff-example-minimal/**。另外，您还可以指定参数来修改命令的行为。

有关可选参数的更多信息，请参阅 roxctl netpol connectivity diff 命令选项 。



注意

该命令考虑您可以使用 **kubectl apply -f** 接受的所有 YAML 文件，然后它们成为 **roxctl netpol connectivity diff** 命令的有效输入。

例 9.2. 输出示例

diff-type	source	目的地	dir 1	dir 2	workloads-diff-info
changed	default/frontend[Deployment]	default/backend[Deployment]	TCP 9090	TCP 9090,UDP 53	
添加	0.0.0.0-255.255.255.255	default/backend[Deployment]	无连接	TCP 9090	

与 **dir1** 中允许的连接相比，语义差异报告为您提供了在 **dir2** 中更改、添加或删除的连接的概述。当您查看输出时，每行都代表了与 **dir1** 相比，在 **dir2** 中添加、删除或更改的允许的连接。

以下是 **roxctl netpol connectivity diff** 命令以各种格式生成的输出示例：

- 示例 1：文本格式
- 示例 2：md 格式
- 示例 3: csv 格式

如果适用，**Workload-diff-info** 提供了有关添加或删除与添加或删除连接相关的工作负载的更多详情。

例如，如果因为删除了工作负载 **B**，从工作负载 **A** 到工作负载 **B** 的连接已被删除，则 workload **-diff-info** 表示工作负载 **B** 已被删除。但是，如果因为网络策略更改以及工作负载 **A** 和 **B** 都删除了这样的连接，则 workload **-diff-info** 为空。

9.3.2. roxctl netpol connectivity diff 命令选项

roxctl netpol connectivity diff 命令支持以下选项：

选项	描述
--dir1 string	输入资源的第一个目录路径。这是强制选项。
--dir2 string	要与第一个目录路径进行比较的输入资源的第二个目录路径。这是强制选项。
--fail	在第一次遇到的错误时失败。默认值为 **false**。

选项	描述
-h，--help	查看 **roxctl netpol connectivity diff** 命令的帮助文本。
-f,--output-file string	将连接差异输出保存到特定文件中。
-o,--output-format string	配置输出格式。支持的格式有 **txt**、**md** 和 **csv**。默认值为 **txt**。
--remove	删除输出路径（如果已存在）。默认值为 **false**。
--save-to-file	将连接差异输出保存到默认文件中。默认值为 **false**。
--strict	将警告视为错误。默认值为 **false**。

9.3.3. 区分语法和语义差异输出

在以下示例中，**dir1** 是 **netpol-analysis-example-minimal/**，**dir2** 是 **netpol-diff-example-minimal/**。目录之间的区别在于网络策略 **backend-netpol** 中的小变化。

dir1 中的策略示例：

```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  creationTimestamp: null
  name: backend-netpol
spec:
  ingress:
  - from:
    - podSelector:
        matchLabels:
          app: frontend
    ports:
    - port: 9090
      protocol: TCP
  podSelector:
    matchLabels:
      app: backendservice
  policyTypes:
  - Ingress
  - Egress
status: {}
```

dir2 中的更改是在 ports 属性前添加的 **-**，这会产生不同的输出。

9.3.3.1. 语法不同输出

流程

- 运行以下命令比较两个指定目录中的 **netpols.yaml** 文件的内容：

```
$ diff netpol-diff-example-minimal/netpols.yaml netpol-analysis-example-
minimal/netpols.yaml
```

输出示例

```
12c12
<   - ports:
---
>     ports:
```

9.3.3.2. 语义差别输出

流程

- 运行以下命令分析两个指定目录中的 Kubernetes 清单和网络策略之间的连接差异：

```
$ roxctl netpol connectivity diff --dir1=roxctl/netpol/connectivity/diff/testdata/netpol-analysis-
example-minimal/ --dir2=roxctl/netpol/connectivity/diff/testdata/netpol-diff-example-minimal
```

输出示例

```
Connectivity diff:
diff-type: changed, source: default/frontend[Deployment], destination:
default/backend[Deployment], dir1:  TCP 9090, dir2: TCP 9090,UDP 53
diff-type: added, source: 0.0.0.0-255.255.255.255, destination: default/backend[Deployment],
dir1:  No Connections, dir2: TCP 9090
```

第 10 章 审计侦听端点

Red Hat Advanced Cluster Security for Kubernetes (RHACS)提供审核在安全集群中侦听端口的进程，并根据部署、命名空间或集群过滤此数据。

您可以使用以下方法查看有关它们监听的进程和端口的信息：

- 在 RHACS web 门户中，进入 **Network → Listening Endpoints**。
- 连接到 API 中的 **ListeningEndpointsService** 对象。如需有关 API 的更多信息，请参阅 RHACS web 门户中的 **Help → API 参考**。

该页面按部署提供进程列表，其中包含列表中每个进程的以下信息：

- 部署名称
- 集群
- Namespace
- 计算或侦听部署中端口的进程数量

您可以使用 filter 字段并输入单个部署、命名空间和集群来进一步过滤页面中显示的信息。

单击列表顶部的展开图标，以展开列出的所有部署的所有部分，或者单击单个部署行上的展开图标来查看该部署的附加信息。提供以下信息：

- exec 文件路径：进程的位置
- PID：进程的系统 ID
- 端口：进程正在侦听的端口
- 协议：进程中使用的协议
- Pod ID：包含进程的 pod 的名称
- 容器名称：正在侦听的进程的容器的名称

点部署名称可进入 RHACS web 门户中的 **Risk** 页面，您可以在其中查看有关部署的信息，包括策略违反和其他部署详情等风险指标。

第 11 章 查看集群配置

了解如何使用 **配置管理** 视图并了解集群中不同实体之间的关联，以有效地管理集群配置。

每个 OpenShift Container Platform 集群都包含集群中分发的许多不同实体，这有助于理解和操作可用信息。

Red Hat Advanced Cluster Security for Kubernetes (RHACS)提供有效的配置管理，将所有这些分布式实体整合到一个页面上。它在一个单一的 **Configuration Management** 视图中集成了有关所有集群、命名空间、节点、部署、镜像、secret、用户、组、服务帐户和角色的信息，可帮助您视觉化不同的实体和它们之间的连接。

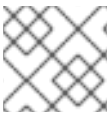
11.1. 使用配置管理视图

要打开 **Configuration Management** 视图，请从导航菜单中选择 **Configuration Management**。与仪表板类似，它会显示一些有用的小部件。

这些小部件是交互式的，显示以下信息：

- 安全策略按严重性违规
- CIS （用于信息安全）Docker 和 Kubernetes 基准控制的状态
- 在大多数集群中具有管理员权限的用户
- 集群中最广泛使用的 secret

Configuration Management 视图中的标头显示集群中的策略和 CIS 控制的数量。



注意

只有 Deploy 生命周期阶段中的策略才会包含在策略计数和策略列表视图中。

标头包含下拉菜单，允许您在实体之间切换。例如，您可以：

- 点 **Policies** 查看所有策略及其严重性，或者选择 **CIS Controls** 查看所有控制的详细信息。
- 点 **Application and Infrastructure** 并选择集群、命名空间、节点、部署、镜像和 secret 来查看详细信息。
- 点 **RBAC Visibility and Configuration**，然后选择 users and groups, service account, 和 roles 来查看详细信息。

11.2. 识别 KUBERNETES 角色中的错误配置

您可以使用 **Configuration Management** 视图来识别潜在的错误配置，如被授予 **cluster-admin** 角色的用户、组或服务帐户，或者未向任何人授予的角色。

11.2.1. 查找 Kubernetes 角色及其分配

使用 **配置管理** 视图获取有关分配给特定用户和组的 Kubernetes 角色的信息。

流程

1. 进入 RHACS 门户并点 **Configuration Management**。
2. 从 **Configuration Management** 视图的标头中选择 **Role-Based Access Control → Users and Groups**。**Users and Groups** 视图显示 Kubernetes 用户和组列表、它们分配的角色，以及是否为每个角色都启用了 **cluster-admin** 角色。
3. 选择用户或组来查看关联的集群和命名空间权限的更多详情。

11.2.2. 查找服务帐户及其权限

使用 **Configuration Management** 视图来查找服务帐户正在使用的位置及其权限。

流程

1. 在 RHACS 门户中，进入 **Configuration Management**。
2. 在 **Configuration Management** 视图的标头中选择 **RBAC Visibility and Configuration → Service Accounts**。**Service Accounts** 视图显示集群中的 Kubernetes 服务帐户列表、其分配的角色，以及是否启用 **cluster-admin** 角色，以及部署使用它们的部署。
3. 选择一个行或下划线链接来查看更多详情，包括为所选服务帐户授予哪个集群和命名空间权限。

11.2.3. 查找未使用的 Kubernetes 角色

使用 **Configuration Management** 视图来获取有关 Kubernetes 角色的更多信息，并查找未使用的角色。

流程

1. 在 RHACS 门户中，进入 **Configuration Management**。
2. 在 **Configuration Management** 视图的头中选择 **RBAC Visibility and Configuration → Roles**。**Roles** 视图显示集群中的 Kubernetes 角色列表、它们授予的权限以及使用位置。
3. 选择一个行或下划线链接来查看角色的更多详情。
4. 要查找未授予任何用户、组或服务帐户的角色，请选择 **Users & Groups** 列标头。然后在保存 **Shift** 键时选择 **Service Account** 列标头。列表中显示没有授予任何用户、组或服务帐户的角色。

11.3. 查看 KUBERNETES SECRET

查看环境中使用的 Kubernetes secret，并使用这些 secret 识别部署。

流程

1. 在 RHACS 门户中，进入 **Configuration Management**。
2. 在大多数不同 **Deployment 小部件的 Secret** 上，选择 **View All**。**Secrets** 视图显示 Kubernetes secret 的列表。
3. 选择一个行来查看更多详情。

使用可用的信息来识别 secret 是否在不需要的部署中使用。

11.4. 查找策略违反情况

Configuration Management 视图中 **的严重性小部件中的 Policy Violations** 在 sunburst 图表中显示策略违反情况。图表的每个级别都由一个环或圆圈表示。

- 内部圆圈代表违反情况的总数。
- 下一个环代表 **Low**, **Medium**, **High**, 和 **Critical** 策略类别。
- 外部环代表特定类别中的单个策略。

Configuration Management 视图仅显示将 **Lifecycle Stage** 设置为 **Deploy** 的策略的信息。它不包括解决运行时的行为或为**构建**阶段评估而配置的策略。

流程

1. 在 RHACS 门户中，进入 **Configuration Management**。
2. 在 **Policy Violations by Severity** 小部件中，将鼠标移到 sunburst chart 中，以查看策略违反情况的详情。
3. 选择 ***n* 评级为高**（其中 *n* 是一个数字），以查看高优先级策略违反情况的详细信息。**Policies** 视图显示根据所选类别过滤的策略违反列表。
4. 选择一个行来查看更多详情，包括策略描述、补救、带有违反情况的部署等。详情可在面板中显示。
5. 信息面板中的 **Policy Findings** 部分列出了发生这些违反情况的部署。
6. 在 **Policy Findings** 部分下选择一个部署，以查看相关的详情，包括 Kubernetes 标签、注解和服务帐户。

您可以使用详细信息来计划违反情况的补救。

11.5. 查找失败的 CIS 控制

与**配置管理**视图中的**策略冲突**类似，**CIS 控制**小部件提供了有关故障中心的信息，用于信息安全 (CIS) 控制。

图表的每个级别都由一个环或圆圈表示。

- 内部圆圈代表失败控制的百分比。
- 下一个环代表控制类别。
- 最顶层的环代表特定类别中的各个控制。

流程

1. 从 **CIS controls** widget 的标头中选择 **CIS Docker v1.2.0**。使用它来在 CIS Docker 和 Kubernetes 控制间切换。
2. 将鼠标悬停在 sunburst chart 上，以查看失败控制的详情。
3. 选择 ***n* 控制失败**，其中 *n* 是一个数字，以查看失败控制的详细信息。**Controls** 视图显示根据合规状态过滤的失败控制列表。
4. 选择一个行来查看更多详情，包括控制描述和控制失败的节点。

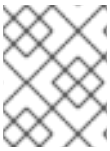
5. 信息面板中的 **Control Findings** 部分列出了控制失败的节点。选择一个行来查看更多详情，包括 Kubernetes 标签、注解和其他元数据。

您可以使用详细信息来专注于节点、行业标准或失败控制的子集。您还可以评估、检查和报告容器化基础架构的合规性状态。

第 12 章 检查漏洞的镜像

使用 Red Hat Advanced Cluster Security for Kubernetes，您可以使用 RHACS 扫描程序分析镜像以了解漏洞，或者您可以将集成 配置为使用其他受支持的扫描程序。

RHACS 中的扫描程序通过分析每个镜像层来查找软件包并将其与已知漏洞匹配，方法是将它们与来自不同源填充的漏洞数据库进行比较。根据使用的扫描程序，源包括国家漏洞数据库(NVD)、开源漏洞(OSV)数据库和操作系统漏洞源。



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

RHACS 包含两个扫描程序： StackRox Scanner 和 Scanner V4。

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。

StackRox 扫描程序源自 Clair v2 开源扫描程序的分叉，是默认的扫描程序。在版本 4.4 中，RHACS 引入了 Scanner V4，基于 ClairCore 构建，它提供额外的镜像扫描功能。

注意

本文档使用术语 "RHACS scanner" 或 "Scanner" 来参考两个扫描程序提供的组合扫描功能： StackRox Scanner 和 Scanner V4。当引用特定扫描程序的功能时，会使用特定扫描程序的名称。

当 RHACS 扫描程序发现任何漏洞时，它会执行以下操作：

- 在 Vulnerability Management 视图中显示它们以了解详细分析
- 根据风险对漏洞进行评级，并在 RHACS 门户中突出显示它们进行风险评估
- 根据启用 的安全策略检查它们

RHACS 扫描程序检查镜像，并根据镜像中的文件识别已安装的组件。如果修改了最终镜像来删除以下文件，则可能无法识别已安装的组件或漏洞：

组件	文件

组件	文件
软件包管理器	• **/etc/alpine-release** • **/etc/apt/sources.list** • **/etc/lsb-release** • **/etc/os-release** 或 **/usr/lib/os-release** • **/etc/oracle-release**,**/etc/centos-release**,**/etc/redhat-release**, 或 **/etc/system-release** • 其他类似的系统文件。
语言级依赖项	• **package.json** 用于 JavaScript. • 用于 Python 的 **dist-info** 或 **egg-info**。 • Java 存档(JAR)中的 **MANIFEST.MF** 用于 Java。
应用程序级别的依赖项	• **dotnet/shared/Microsoft.AspNetCore.App/** • **dotnet/shared/Microsoft.NETCore.App/**

12.1. 关于 RHACS SCANNER V4 （技术预览）

RHACS 提供自己的扫描程序，或者您可以将集成配置为将 RHACS 与其他漏洞扫描程序一起使用。

从版本 4.4 开始，Scanner V4 基于 ClairCore 构建，为语言和特定于操作系统的镜像组件提供扫描。对于版本 4.4，RHACS 还使用 StackRox Scanner 提供一些扫描功能，直到功能在以后的发行版本中实现为止。

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。

其他资源

- 使用 Operator 为 OpenShift Container Platform 安装 RHACS 的扫描程序 V4 设置
- 使用 Helm 为 OpenShift Container Platform 安装 RHACS 的扫描程序 V4 设置
- 使用 Helm 为 Kubernetes 安装 RHACS 的扫描程序 V4 设置

12.2. 扫描镜像

对于版本 4.4，RHACS 提供两个扫描程序： StackRox Scanner 和 Scanner V4。两个扫描程序都可以检查网络中连接的安全集群中的镜像。在使用 Operator 或使用委派扫描时，部署的 Red Hat OpenShift 环境中默认启用安全集群扫描。如需更多信息，请参阅"访问委派的镜像扫描"。

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。

使用 StackRox Scanner 时，RHACS 执行以下操作：

- Central 将镜像扫描请求提交到 StackRox Scanner。
- 在收到这些请求时，StackRox Scanner 从相关 registry 中拉取镜像层，检查镜像，并识别每个层中安装的软件包。然后，它会将确定的软件包和特定于编程语言的依赖项与漏洞列表进行比较，并将信息发回到 Central
- StackRox Scanner 标识以下区域中的漏洞：
  - 基础镜像操作系统
  - 软件包管理器安装的软件包
  - 特定于编程语言的依赖项
  - 编程运行时和框架

使用 Scanner V4 时，RHACS 执行以下操作：

- Central 请求 Scanner V4 Indexer 下载和索引(analyze)给定镜像。
- 扫描程序 V4 Indexer 从 registry 中拉取镜像元数据，以确定镜像的层，并下载之前未索引的层。
- 扫描程序 V4 Indexer 从 Central 请求映射文件，以帮助索引过程。扫描程序 V4 Indexer 在索引报告中生成。
- Central 请求 Scanner V4 Matcher 与给定镜像匹配到已知漏洞。这个过程会产生最终扫描结果：漏洞报告。扫描程序 V4 Matcher 从 Central 请求最新的漏洞。
- 扫描程序 V4 Matcher 从 Scanner V4 Indexer 请求镜像索引（索引报告）的结果。然后，它会使用报告来确定相关的漏洞。只有在 Central 集群中索引镜像时，才会发生此交互。当 Scanner V4 与安全集群中索引的镜像的漏洞匹配时，不会发生此交互。
- Indexer 在与索引结果相关的 Scanner V4 DB 中存储数据，以确保镜像层仅下载并索引一次。这可以防止不必要的网络流量和其他资源利用率。
- 启用安全集群扫描后，Sensor 会请求 Scanner V4 来索引镜像。扫描程序 V4 Indexer 从 Sensor 请求映射文件，以帮助索引过程，除非在同一命名空间中存在 Central。在这种情况下，会联系 Central。

12.2.1. 了解并解决常见的 Scanner 警告信息

当使用 Red Hat Advanced Cluster Security for Kubernetes (RHACS)扫描镜像时，您可能会看到 **CVE DATA MAY BE INACCURATE** 警告信息。当扫描程序无法检索有关操作系统或其他镜像中其他软件包的完整信息时，扫描程序会显示此消息。

下表显示了一些常见的 Scanner 警告信息：

表 12.1. 警告信息

消息	描述
无法检索 OS CVE 数据，只有语言 CVE 数据可用	表示 Scanner 不支持镜像的基本操作系统，因此无法检索操作系统级别的软件包的 CVE 数据。
过时的操作系统 CVE 数据	表示镜像的基本操作系统已达到生命周期结束，这意味着漏洞数据已过时。例如，Debian 8 和 9。 有关识别镜像中组件所需的文件的更多信息，请参阅检查漏洞的镜像。
获取基础操作系统信息失败	表示 Scanner 会扫描镜像，但无法决定用于镜像的基本操作系统。
从 registry 检索元数据失败	表示目标 registry 在网络上无法访问。原因可能是防火墙阻止 **docker.io** 或阻止访问的身份验证问题。 要分析根本原因，请为私有 registry 或存储库创建一个特殊的 registry 集成，以获取 RHACS Central 的 pod 日志。有关如何进行此操作的说明，请参阅 与镜像 registry 集成。
红帽漏洞扫描程序认证范围之外的镜像	表示 Scanner 扫描了镜像，但镜像旧且没有在 Red Hat Scanner 认证范围内。如需更多信息，请参阅 红帽漏洞扫描程序认证合作伙伴指南。  **重要** 如果您使用红帽容器镜像，请考虑使用比 2020 年 6 月更新 的基础镜像。

12.2.2. 支持的软件包格式

扫描程序可以检查镜像中使用以下软件包格式的漏洞：

- apt
- apk
- dpkg
- rpm

12.2.3. 支持的编程语言

扫描程序可以检查依赖项中的漏洞，以了解以下编程语言：

- Go （仅扫描 V4）
  - 二进制文件：用于构建二进制文件的标准库版本已被分析。如果使用模块支持(go.mod)构建二进制文件，则也会分析依赖项。
- Java
  - JAR
  - WAR
  - EAR
- JavaScript
  - Node.js
  - npm package.json
- Python
  - egg 和 wheel 格式
- Ruby
  - gem

12.2.4. 支持的运行时和框架

从 Red Hat Advanced Cluster Security for Kubernetes 3.0.50 (Scanner 版本 2.5.0)开始，StackRox Scanner 识别以下开发人员平台中的漏洞：

- .NET Core
- ASP.NET Core

Scanner V4 不支持它们。

12.2.5. 支持的操作系统

本节中列出的支持的平台是 Scanner 识别漏洞的发行版本，它与您可以安装 Red Hat Advanced Cluster Security for Kubernetes 的支持的平台不同。

扫描程序识别包含以下 Linux 发行版本的镜像中的漏洞。有关使用的漏洞数据库的更多信息，请参阅"RHACS 架构"中的"Vulnerability sources"。

分发	版本

分发	版本
alpine Linux	**alpine:3.2**[1],**alpine:3.3**,**alpine:3.4**,**alpine:3.5**,**alpine:3.6**,**alpine:3.7**,**alpine:3.8**,**alpine:3.9**,**alpine:3.10**, **alpine:3.11**,**alpine:3.12**,**alpine:3.13**,**alpine:3.14**, **alpine:3.15**,**alpine:3.16**,**alpine:3.17**,**alpine:3.18**, **alpine:3.19**[2], **alpine:edge**
Amazon Linux	**amzn:2018.03**, **amzn:2**, **amzn:2023**[2]
CentOS	**centos:6**[1], **centos:7**[1], **centos:8**[1]
Debian	**Debian:10**,**debian:11**,**debian:12**,**debian:unstable**,**distroless**
Oracle Linux	版本 5-9[2]
Photon OS	1.0[2], 2.0[2], 3.0 [2]
Red Hat Enterprise Linux (RHEL)	**rhel:6**[3], **rhel:7**[3], **rhel:8**[3], **rhel:9**[3]
SUSE	SLES 11, 12, 15[2]; openSUSE Leap 42.3, 15.0, 15.1[2]; SUSE Linux[2]
Ubuntu	**Ubuntu:14.04**,**iwl:16.04**,**iwl:18.04**,**iwl:20.04**,**ubuntu:21.04**,**ubuntu:21.10**,**ubuntu:22.04**,**ubuntu:22.10**,**需要需要：23.04**,**ubuntu:23.10** 供应商不会更新以下漏洞源： **Ubuntu:12.04**,**ubuntu:12.10 , iwl:13.04 , iwl:14.10**,**ubuntu:15.04**,**ubuntu::15.10,** 需要 **Ubuntu:17.04**,**ubuntu:17.10**, iwl**:18.10 , iwl:18.10 , iwl:19.04**,**ubuntu:19.10**,**iwl:20.10**,

1. 仅在 StackRox Scanner 中支持。
2. 仅在扫描器 V4 中支持。
3. Scanner V4 不支持比 2020 年 6 月旧的镜像。

注意

- 扫描程序不支持 Fedora 操作系统，因为 Fedora 不维护漏洞数据库。但是，Scanner 仍然检测基于 Fedora 的镜像中的特定语言漏洞。

其他资源

- 漏洞源

- 与镜像漏洞策略集成

12.3. 访问委派的镜像扫描

您可以隔离只能从安全集群访问的容器镜像 registry。委派的镜像扫描功能可让您从安全集群中的任何 registry 中扫描镜像。

12.3.1. 通过访问委派的镜像扫描来增强镜像扫描

目前，默认情况下，Central Services Scanner 为安全集群中观察到的镜像执行索引（识别组件）和漏洞匹配（带有漏洞数据的丰富功能），但 OpenShift Container Platform 集成 registry 中的镜像除外。

对于 OpenShift Container Platform 集成 registry 中的镜像，在安全集群中安装 Scanner-slim 会执行索引，Central Services Scanner 会执行漏洞匹配。

委派的镜像扫描功能通过允许 Scanner-slim 从任何 registry 索引镜像拉取镜像来扩展扫描功能，然后将其发送到 Central 进行漏洞匹配。要使用这个功能，请确保在安全集群中安装 Scanner-slim。如果没有 Scanner-slim，则扫描请求将直接发送到 Central。

12.3.2. 配置委派的镜像扫描

新的委托 registry 配置指定要从中委派镜像扫描的 registry。对于 Sensor 观察的镜像，此配置允许您从没有 registry、所有 registry 或特定 registry 中委派扫描。要使用 **roxctl** CLI、Jenkins 插件或 API 启用扫描委托，还必须指定目标集群和源 registry。

先决条件

- scanner-slim 必须安装在安全集群中，才能扫描镜像。

注意

OpenShift Container Platform 和 Kubernetes 安全集群中支持启用 Scanner-slim。

流程

1. 在 RHACS 门户中，进入 **Platform Configuration → Clusters**。
2. 在 **Clusters** 视图标头中，点 **Manage delegated scan**。
3. 在 **委派的** 镜像扫描页面中，提供以下信息：
   - **委派扫描** ：通过选择以下选项之一来选择镜像委派的范围：
     - none ：默认选项。这个选项指定安全集群不会扫描任何镜像，但 OpenShift Container Platform 集成 registry 中的镜像除外。
     - 所有 registry：此选项表示所有镜像都由安全集群扫描。
     - 指定 registry：此选项指定应由安全集群根据 registry 列表扫描哪些镜像。
   - **选择默认集群来委派至** ：从下拉列表中选择默认集群名称，该集群将处理来自命令行界面 (CLI)和 API 的扫描请求。这是可选的，如果需要，您可以选择 **None**。

- 可选：点 **Add registry** 并指定源 registry 和目标集群详情。如果扫描请求没有来自 CLI 和 API，您可以将目标集群选为 **None**。如果需要，您可以添加多个源 registry 和目标集群。

4. 点击 **Save**。

镜像集成现在在 Central 和 Sensor 之间同步，Sensor 从每个命名空间中捕获 pull-secrets。然后，Sensor 使用这些凭证向镜像 registry 进行身份验证。

12.3.3. 在安全集群中安装和配置 Scanner-slim

12.3.3.1. 使用 Operator

RHACS Operator 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。

如需更多信息，请参阅使用 Operator 在安全集群中安装 RHACS。

12.3.3.2. 使用 Helm

安全集群服务 Helm Chart (**secured-cluster-services**)在每个安全集群中安装 Scanner-slim 版本。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 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。

- 对于 OpenShift Container Platform 安装，请参阅在没有自定义的情况下安装 secure-cluster-services Helm chart。
- 对于非 OpenShift Container Platform 安装，如 Amazon Elastic Kubernetes Service (Amazon EKS)、Google Kubernetes Engine (Google GKE)和 Microsoft Azure Kubernetes Service (Microsoft AKS)，请参阅在没有自定义的情况下安装 secure-cluster-services Helm chart。

12.3.3.3. 安装后验证

流程

- 验证安全集群的状态表示 Scanner 是否存在并健康：
  a. 在 RHACS 门户中，进入 **Platform Configuration → Clusters**。
  b. 在 **Clusters** 视图中，选择一个集群来查看其详情。
  c. 在 **Health Status** 卡中，确保 **Scanner** 存在并标记为 **Healthy**。

12.3.3.4. 使用镜像扫描

您可以使用 **roxctl** CLI、Jenkins 和 API 扫描存储在集群特定 OpenShift Container Platform 集成镜像 registry 中的镜像。您可以在委派的扫描配置中指定适当的集群，或使用 **roxctl** CLI、Jenkins 和 API 中提供的 cluster 参数。

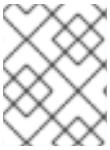
有关如何使用 roxctl CLI 扫描镜像的更多信息，请参阅使用 **roxctl** CLI 扫描镜像。

12.4. 设置扫描

您可以配置用于扫描的设置，如自动扫描活动和不活跃的镜像。

12.4.1. 自动扫描活动镜像

Red Hat Advanced Cluster Security for Kubernetes 会定期扫描所有活跃的镜像并更新镜像扫描结果，以反映最新的漏洞定义。活动镜像是您在您的环境中部署的镜像。



注意

在 Red Hat Advanced Cluster Security for Kubernetes 3.0.57 中，您可以通过为镜像配置 **Watch** 设置来启用不活跃镜像的自动扫描。

Central 从扫描程序或其他集成镜像扫描程序中获取所有活跃镜像的镜像扫描结果，每 4 小时更新结果。

您还可以使用 **roxctl** CLI 按需检查镜像扫描结果。

12.4.2. 扫描不活跃的镜像

Red Hat Advanced Cluster Security for Kubernetes (RHACS)每 4 小时扫描所有活跃的（部署）镜像，并更新镜像扫描结果以反映最新的漏洞定义。

您还可以将 RHACS 配置为自动扫描不活跃（未部署）镜像。

流程

1. 在 RHACS 门户中，进入 **Vulnerability Management (2.0)→ Workload CVEs （技术预览）**。
2. 点 **<number> Images** 显示镜像列表，并找到您要监视的镜像。
3. 点溢出菜单 ，然后选择 **Watch image**。然后，RHACS 会扫描镜像并显示错误或成功信息。
4. （可选） 要删除监视的镜像，请点溢出菜单 ，然后选择 **Unwatch image**。
5. （可选）您可以在页面标头中点 **Manage watched images** 来查看所有被监视的镜像列表，并添加附加镜像。

重要

在 RHACS 门户中，点 **Platform Configuration → System Configuration** 查看数据保留配置。

对于 **System Configuration** 页面中提到的天数，所有与从监视的镜像列表中删除的镜像相关的数据都会继续出现在 RHACS 门户中，仅在该周期过后删除。

6. 点 **Close** 返回 **Workload CVEs** 页面。

其他资源

- 扫描不活跃的镜像
- 安装 roxctl CLI

12.5. 关于漏洞

RHACS 从多个漏洞源获取漏洞定义和更新。这些源是常规的性质，如 NVD 或特定于发行版，如 Alpine、Debian 和 Ubuntu。有关查看和解决发现的漏洞的更多信息，请参阅 漏洞管理。

12.5.1. 获取漏洞定义

在在线模式中，Central 每 5 分钟从单个源获取漏洞定义。此源将来自上游源的漏洞定义合并，每 3 小时刷新一次。

- 源的地址是 **https://definitions.stackrox.io**。
- 您可以通过设置 **ROX_SCANNER_VULN_UPDATE_INTERVAL** 环境变量来更改 Central 和 StackRox Scanner 的默认查询频率：

```
$ oc -n stackrox set env deploy/central ROX_SCANNER_VULN_UPDATE_INTERVAL=<value> 1
```

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

请注意以下指导：

- StackRox Scanner 的配置映射仍然有一个 **updater.interval** 参数，用于配置扫描程序的更新频率，但它不再包含 **fetchFromCentral** 参数。
- Scanner V4 不支持设置此环境变量。

有关 RHACS 使用的漏洞源的更多信息，请参阅 "Red Hat Advanced Cluster Security for Kubernetes 架构"中的 "Vulnerability sources"。

其他资源

- 漏洞源

12.5.2. 了解漏洞分数

Red Hat Advanced Cluster Security for Kubernetes 门户为每个漏洞显示一个通用漏洞评分系统(CVSS)基本分数。RHACS 根据以下条件显示 CVSS 分数：

- 如果 CVSS v3 分数可用，RHACS 会显示分数并列出 **v3**。例如： **6.5 (v3)**。

注意

只有在您使用 StackRox Scanner 版本 1.3.5 及之后的版本或扫描器 V4 时，CVSS v3 分数才可用。

- 如果 CVSS v3 分数不可用，RHACS 可能只显示 CVSS v2 分数。例如： **6.5**。

您可以使用 API 获取 CVSS 分数。如果 CVSS v3 信息可用于漏洞，响应可能包括 CVSS v3 和 CVSS v2 信息。

对于红帽安全公告(RHSA)，CVSS 分数设置为所有相关 CVE 中最高 CVSS 分数。一个 RHSA 可以包含多个 CVE，红帽有时会根据漏洞对其他红帽产品的影响来分配不同的分数。

12.6. 禁用特定于语言的漏洞扫描

扫描程序默认识别编程语言特定依赖项中的漏洞。您可以禁用特定于语言的依赖项扫描。

流程

- 要禁用特定于语言的漏洞扫描，请运行以下命令：

```
$ oc -n stackrox set env deploy/scanner \ 1
  ROX_LANGUAGE_VULNS=false 2
```

1. 如果使用 Kubernetes，请输入 **kubectl** 而不是 **oc**。
2. 如果您使用 Red Hat Advanced Cluster Security for Kubernetes 版本 3.0.47 或更早版本，请将环境变量名称 **ROX_LANGUAGE_VULNS** 替换为 **LANGUAGE_VULNS**。

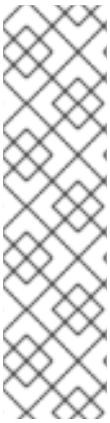
12.7. 其他资源

- Red Hat CVE 数据库

第 13 章 验证镜像签名

您可以使用 Red Hat Advanced Cluster Security for Kubernetes (RHACS)通过针对预先配置的密钥验证镜像签名来确保集群中容器镜像的完整性。

您可以创建策略来阻止未签名的镜像和没有验证签名的镜像。您还可以使用 RHACS 准入控制器停止未授权部署创建来强制实施策略。



注意

- RHACS 3.70 只支持 Cosign 签名和 Cosign 公钥签名验证。有关 Cosign 的更多信息，请参阅 Cosign 概述。
- 您必须至少使用 1 Cosign 公钥配置签名集成，以进行签名验证。
- 对于所有部署和监视的镜像：
  - RHACS 每 4 小时获取并验证签名。
  - 每当您更改或更新签名集成公钥时，RHACS 会验证签名。

13.1. 配置签名集成

在执行镜像签名验证前，您必须首先在 RHACS 中添加 Cosign 公钥。

先决条件

- 您必须已有一个 PEM 编码的 Cosign 公钥。有关 Cosign 的更多信息，请参阅 Cosign 概述。

流程

1. 在 RHACS 门户中，选择 **Platform Configuration → Integrations**。
2. 向下滚动到 **Signature Integrations** 部分，然后点 **Signature**。
3. 点 **New integration**。
4. 输入 **集成名称**。
5. 点 **Cosign → Add a new public key**。
6. 输入 **公钥** 名称。
7. 对于 **Public key value** 字段，输入 PEM 编码的公钥。
8. （可选）您可以通过点 **Add a new public key** 并输入详情来添加多个密钥。
9. 点击 **Save**。

13.2. 在策略中使用签名验证

在创建自定义安全策略时，您可以使用 **受信任的镜像签名策略** 标准来验证镜像签名。

先决条件

- 您必须已经配置了与至少 1 Cosign 公钥的签名集成。

流程

1. 在创建或编辑策略时，对于 **Policy criteria** 部分，将 **Not verified by trusted image signers**策略条件拖放到策略字段放置区。
2. 单击 **Select**。
3. 从列表中选择可信镜像签名，然后单击 **Save**。

其他资源

- 从系统策略视图创建安全策略
- 策略标准

13.3. 强制签名验证

要防止用户使用未签名镜像，您可以使用 RHACS 准入控制器强制签名验证。您必须首先在集群配置设置中启用 **Contact Image Scanners** 功能。然后，在创建安全策略以强制签名验证时，您可以使用 **Inform 和 enforce** 选项。

如需更多信息，请参阅启用准入控制器强制。

其他资源

- 从系统策略视图创建安全策略

第 14 章 管理漏洞

14.1. 漏洞管理

攻击者可能会利用您的环境中的安全漏洞来执行未经授权的操作，如拒绝服务、远程代码执行或未授权对敏感数据的访问。因此，对漏洞的管理是成功 Kubernetes 安全计划的基础步骤。

14.1.1. 漏洞管理流程

漏洞管理是识别和修复漏洞的持续流程。Red Hat Advanced Cluster Security for Kubernetes 可帮助您促进漏洞管理流程。

成功的漏洞管理计划通常包括以下关键任务：

- 执行资产评估
- 对漏洞进行优先级排序
- 评估暴露信息
- 采取行动
- 持续恢复资产

Red Hat Advanced Cluster Security for Kubernetes 可帮助机构在其 OpenShift Container Platform 和 Kubernetes 集群上执行持续评估。它为组织提供了所需的上下文信息，以便更有效地对环境中的漏洞进行优先级排序和操作。

14.1.1.1. 执行资产评估

对机构资产进行评估涉及以下操作：

- 识别环境中的资产
- 扫描这些资产以识别已知漏洞
- 报告您环境中的漏洞，以影响利益相关者

当您在 Kubernetes 或 OpenShift Container Platform 集群上安装 Red Hat Advanced Cluster Security for Kubernetes 时，它会首先聚合集群中运行的资产，以帮助您识别这些资产。RHACS 允许机构在其 OpenShift Container Platform 和 Kubernetes 集群上执行持续评估。RHACS 为机构提供上下文信息，以便更有效地对环境中的漏洞进行优先级和操作。

应该由使用 RHACS 的机构漏洞管理流程监控的重要资产包括：

- **组件** ：组件是可用作镜像一部分的软件包或节点上运行的软件包。组件是存在漏洞的最低级别。因此，组织必须升级、修改或删除软件组件才能修复漏洞。
- **镜像** ：创建环境以运行可执行代码的软件组件和代码集合。镜像是您升级组件以修复漏洞的位置。
- **节点** ：用于管理和运行使用 OpenShift 或 Kubernetes 应用程序的服务器，以及组成 OpenShift Container Platform 或 Kubernetes 服务的组件。

Red Hat Advanced Cluster Security for Kubernetes 将这些资产分组到以下结构中：

- **部署** ：Kubernetes 中的应用程序的定义，它可能根据一个或多个镜像运行带有容器的 pod。
- **命名空间** ：一组资源，如支持并隔离应用程序的 Deployment。
- **集群** ：用于运行使用 OpenShift 或 Kubernetes 的应用的一组节点。

Red Hat Advanced Cluster Security for Kubernetes 会扫描资产中的已知漏洞，并使用常见漏洞和暴露(CVE)数据来评估已知漏洞的影响。

14.1.2. 查看漏洞

RHACS 提供了以下方法来查看系统中发现的漏洞：

- 要根据命名空间或部署查看应用程序漏洞，或在 RHACS web 门户中查看镜像中的漏洞，请访问 **Vulnerability Management (1.0)** → **Dashboard**。
- 要查看系统中集群中运行的应用程序漏洞，请转至 **Vulnerability Management (2.0)** → **Workload CVEs**。您可以根据镜像、部署、命名空间和集群过滤漏洞。

14.1.2.1. 查看应用程序漏洞

您可以查看 Red Hat Advanced Cluster Security for Kubernetes 中的应用程序漏洞。

流程

1. 在 RHACS 门户中，进入 **Vulnerability Management 1.0** → **Dashboard**。
2. 在 **Dashboard** 视图标头中，选择 **Application & Infrastructure** → **Namespaces** 或 **Deployments**。
3. 在列表中，搜索并选择 **您要查看的** 命名空间或 **Deployment**。
4. 要获取有关应用程序的更多信息，请从右侧的**相关实体**中选择一个实体。

14.1.2.2. 查看镜像漏洞

您可以查看 Red Hat Advanced Cluster Security for Kubernetes 中的镜像漏洞。

流程

1. 在 RHACS 门户中，进入 **Vulnerability Management 1.0** → **Dashboard**。
2. 在 **Dashboard** 视图标头中，选择 **Images**。
3. 从镜像列表中，选择您要调查的镜像。您还可以通过执行以下步骤之一过滤列表：
   a. 在搜索栏中输入 **Image**，然后选择 **Image** 属性。
   b. 在搜索栏中输入镜像名称。
4. 在镜像详情视图中，查看列出的 CVE，并优先执行相应的操作来解决受影响的组件。
5. 从右侧的 **Related entities** 中选择 **Components**，以获取有关受所选镜像影响的所有组件的更多信息。或者，在 **Image findings** 部分的 **Affected components** 列中选择**组件**，以了解受特定 CVE 影响的组件列表。

其他资源

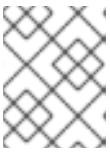
- 使用本地页面过滤

14.1.2.3. 查看漏洞管理(2.0)中的工作负载 CVE

您可以在镜像和部署间查看 RHACS 中的漏洞或 CVE 的完整列表。您可以使用搜索栏来选择特定的 CVE、镜像、部署、命名空间或集群。

流程

1. 在 RHACS 门户中，进入 **Vulnerability Management (2.0)→ Workload CVEs**。
2. 从下拉列表中选择您要使用的搜索条件。您可以从列表中选择一个项目类型，如集群，然后选择项目的特定名称。您可以通过从列表中选择其他项目并选择新项目的特定名称来向过滤器添加其他项目。例如，您可以选择特定的镜像和特定集群来限制这些选择的结果。您可以对以下项目进行过滤：
   - CVE
   - 镜像
   - Deployment
   - 命名空间
   - Cluster
   - 组件
   - 组件源
3. 可选：使用 **CVE 严重性** 列表来选择您要显示的 CVE 的严重性。
4. 点相关的按钮查看系统中的漏洞、镜像或部署列表。



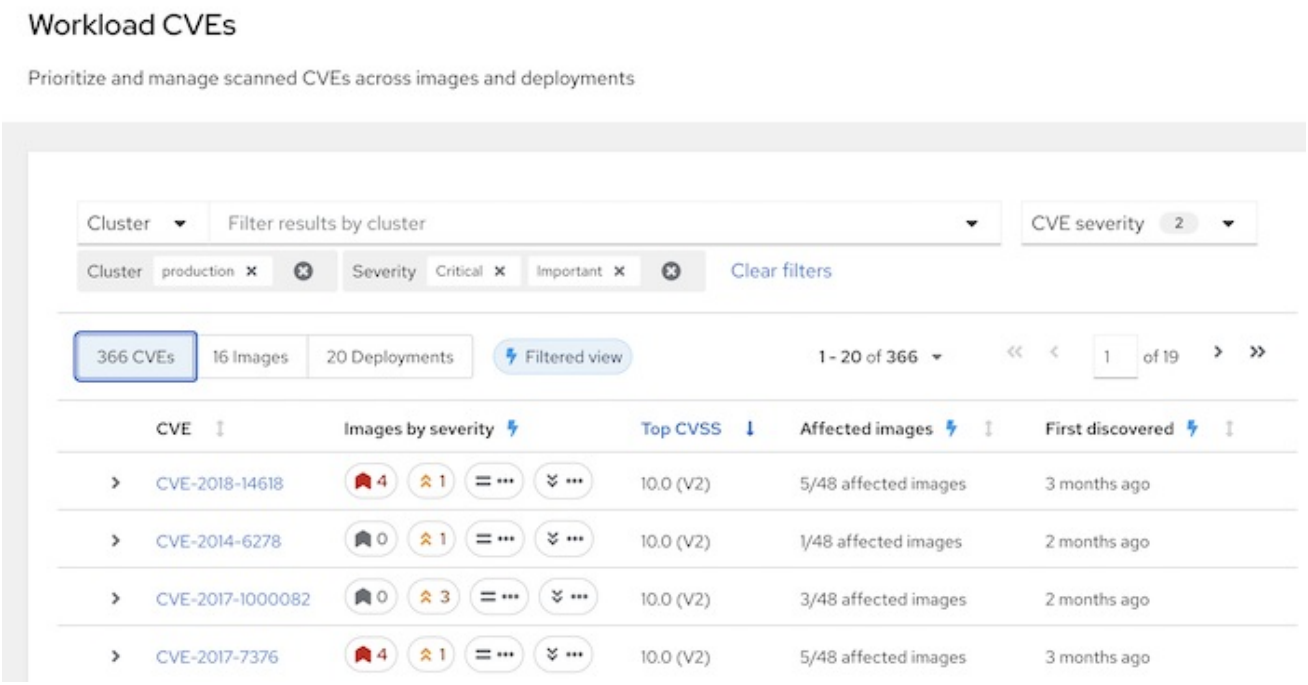
注意

Filtered 视图 图标显示根据您选择的条件过滤显示的结果。您可以单击 **Clear filters** 来删除所有过滤器，或通过单击各个过滤器来删除它们。

5. 在结果列表中，点 CVE、镜像名称或部署名称查看有关项目的更多信息。例如，根据项目类型，您可以查看以下信息：
   - CVE 是否可以被修复
   - 镜像是否活跃
   - 包含 CVE 的镜像中的 Dockerfile 行
   - 有关红帽和其他 CVE 数据库中 CVE 的外部链接

搜索示例

下图显示了名为"production"的集群的搜索条件示例，以查看该集群中关键和重要严重性的 CVE。



14.1.2.3.1. 查看基础架构漏洞

您可以使用 Red Hat Advanced Cluster Security for Kubernetes 查看节点中的漏洞。

流程

1. 在 RHACS 门户中，进入 **Vulnerability Management 1.0 → Dashboard**。
2. 在 **Dashboard** 视图标头中选择 **Application & Infrastructure → Cluster**。
3. 从集群列表中选择您要调查的集群。
4. 查看集群漏洞，并优先执行集群上受影响节点的操作。

14.1.2.3.2. 查看节点漏洞

您可以使用 Red Hat Advanced Cluster Security for Kubernetes 查看特定节点中的漏洞。

流程

1. 在 RHACS 门户中，进入 **Vulnerability Management 1.0 → Dashboard**。
2. 在 **Dashboard** 视图标头中选择 **Nodes**。
3. 从节点列表中，选择您要调查的节点。
4. 检查所选节点的漏洞以及采取优先级的操作。
5. 要获取有关节点上受影响组件的更多信息，请从右侧的 **相关实体** 中选择 **组件**。

14.1.2.4. 对漏洞进行优先级排序

回答以下问题以优先选择您环境中的漏洞进行行动和调查：

- 对于您的组织而言，受影响的资产非常重要？

- 在调查漏洞时，需要如何严重？
- 漏洞是否可以由受影响软件组件的补丁修复？
- 存在的漏洞是否违反了任何机构的安全策略？

这些问题的回答可帮助安全性和开发团队确定他们是否希望对漏洞的暴露进行量化。

Red Hat Advanced Cluster Security for Kubernetes 为您提供了促进应用程序和组件中漏洞的优先级的方法。

14.1.2.5. 评估暴露信息

要评估您对漏洞的风险，请回答以下问题：

- 您的应用程序是否受到漏洞的影响？
- 漏洞是否被其他因素缓解？
- 是否存在可能导致利用此漏洞的已知威胁？
- 您正在使用软件包具有该漏洞？
- 是否将时间花费在特定漏洞上，并且软件包是否值得考虑？

根据您的评估执行以下操作：

- 如果您确定没有暴露，或者您的环境中没有应用漏洞，请考虑将漏洞标记为假的正状态。
- 如果您愿意修复、缓解或接受风险，请考虑是否希望修复、缓解或接受风险。
- 考虑是否要删除或更改软件包以减少您的攻击面。

14.1.2.6. 采取行动

决定对漏洞采取行动后，您可以执行以下操作之一：

- 修复漏洞
- 缓解并接受风险
- 接受风险
- 将漏洞标记为假正

您可以通过执行以下操作之一修复漏洞：

- 删除软件包
- 将软件包更新为一个不可安全的版本。

其他资源

- 查看假的正或延迟的 CVE

14.1.2.6.1. 查找新的组件版本

以下流程找到要升级到的新组件版本。

流程

1. 在 RHACS 门户中，进入 **Vulnerability Management 1.0 → Dashboard**。
2. 在 **Dashboard** 视图标头中，选择 **Images**。
3. 从镜像列表中选择您已评估的镜像。
4. 在 **Image findings** 部分下，选择 CVE。
5. 选择您要采取的 CVE 受影响组件。
6. 查看 CVE 已修复的组件版本并更新您的镜像。

14.1.2.7. 接受风险

按照本节中的说明接受 Red Hat Advanced Cluster Security for Kubernetes 中的风险。

先决条件

- 您必须具有 **VulnerabilityManagementRequests** 资源的**写入**权限。

使用或没有缓解措施接受风险：

流程

1. 在 RHACS 门户中，进入 **Vulnerability Management 1.0 → Dashboard**。
2. 在 **Dashboard** 视图标头中，选择 **Images**。
3. 从镜像列表中选择您已评估的镜像。
4. 找到列出您要采取行动的 CVE 的行。
5. 对于您确定的 CVE，点 overflow 菜单  。
6. 点 **Defer CVE**。
7. 选择您要延迟 CVE 的日期和时间。
8. 如果要延迟所选镜像标签的 CVE 或此镜像的所有标签，请选择。
9. 输入延迟的原因。
10. 点 **Request approval**。选择 CVE 右侧的蓝色信息图标，并复制批准链接，以与您的机构延迟批准者共享。

14.1.2.7.1. 将漏洞标记为假正

以下流程将漏洞标记为假正。

先决条件

- 您必须具有 **VulnerabilityManagementRequests** 资源的**写入**权限。

流程

1. 在 RHACS 门户中，进入 **Vulnerability Management 1.0 → Dashboard**。
2. 在 **Dashboard** 视图标头中，选择 **Images**。
3. 从镜像列表中选择您已评估的镜像。
4. 找到列出您要采取行动的 CVE 的行。
5. 点击您确定的 CVE 右侧的  并点 **Defer CVE**。
6. 选择您要延迟 CVE 的日期和时间。
7. 如果要延迟所选镜像标签的 CVE 或此镜像的所有标签，请选择。
8. 输入延迟的原因。
9. 点 **Request approval**。
10. 选择 CVE 右侧的蓝色信息图标，并复制批准链接，以与您的机构延迟批准者共享。

14.1.2.7.2. 查看假的正或延迟的 CVE

使用以下步骤检查假的正或延迟的 CVE。

先决条件

- 您必须具有 **VulnerabilityManagementApprovals** 资源的**写入**权限。

您可以查看"假正"或延迟的 CVE：

流程

1. 在浏览器中或 RHACS 门户中打开批准链接。
2. 进入 **Vulnerability Management → Risk Acceptance** 并搜索 CVE。
3. 查看漏洞范围和操作，以确定您是否要批准它。
4. 点击 CVE 最右侧的  ，并批准或拒绝批准请求。

14.1.2.8. 向团队报告漏洞

当组织必须不断重新评估并报告其漏洞时，一些组织会发现，有与关键利益相关者的通信会很有帮助，以帮助进行漏洞管理流程。

您可以使用 Red Hat Advanced Cluster Security for Kubernetes 通过电子邮件调度这些周期性通信。这些通信应限定到关键利益相关者需要的最相关信息。

要发送这些通信，您必须考虑以下问题：

- 与利益相关者通信时，哪些计划会有最大的影响？
- 谁是受众？
- 是否只在报告中发送特定的严重性漏洞？
- 是否只在报告中发送可修复的漏洞？

14.1.3. 漏洞报告

您可以从 RHACS web 门户中的 **Vulnerability Management (2.0)** 菜单创建并下载按需镜像漏洞报告。此报告包括镜像和部署间的通用漏洞和风险的完整列表，在 RHACS 中称为工作负载 CVE。您可以通过在 RHACS 中调度电子邮件或使用其它方法共享此报告与审核员或内部利益相关者共享。

14.1.3.1. 创建漏洞管理报告配置

RHACS 指导您完成创建漏洞管理报告配置的过程。此配置决定了将在计划的时间或按需求运行的报告作业中包含的信息。

流程

1. 在 RHACS 门户中，进入 **漏洞管理(2.0) → 漏洞报告**。
2. 点 **Create report**。
3. 在 **Report name** 字段中输入报告配置的名称。
4. 可选：在 **Description** 字段中输入描述报告配置的文本。
5. 在 **CVE severity** 字段中，选择您要包含在报告配置中的通用漏洞和暴露(CVE)的严重性。
6. 选择 **CVE 状态**。您可以选择 **Fixable**、**Unfixable** 或两者。
7. 在 **Image type** 字段中，选择是否要包含来自部署的镜像的 CVE、监视的镜像或两者。
8. 在 **自字段起发现的** CVE 中，选择您希望 CVE 包含在报告配置中的时间周期。
9. 在 **Configure report scope** 字段中，您可以执行以下操作：
   - 选择现有集合并点 **View** 查看集合信息，编辑集合，并获取集合结果预览。在查看集合时，在字段中输入文本搜索与该文本字符串匹配的集合。
   - 点 **Create collection** 以创建新集合。

注意

有关集合的更多信息，请参阅"添加资源"部分中的"创建和使用部署集合"。

10. 点 **Next** 来配置交付目的地，并选择性地设置交付计划。

14.1.3.1.1. 配置交付目的地和调度

除非在上一页上选择了选项，否则为漏洞报告配置目的地和交付计划是可选的，否则您可以选择包含自上次调度的报告后发现的 CVE 的选项。如果您选择了这个选项，则需要为漏洞报告配置目的地和交付计划。

流程

1. 要配置用于交付的目的地，请在 **Configure delivery destinations** 部分中，添加发送目的地并设置报告时间表。

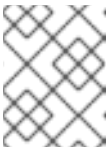
2. 要电子邮件报告，必须至少配置一个电子邮件通知。选择现有通知程序或创建一个新的电子邮件通知，以通过电子邮件发送您的报告。有关创建电子邮件通知的更多信息，请参阅"添加资源"部分中的"配置电子邮件插件"。
   当您选择通知程序时，在通知程序中配置的作为**默认接收者**的电子邮件地址会出现在 **Distribution list** 字段中。您可以添加以逗号分开的其他电子邮件地址。

3. 默认电子邮件模板会自动应用。要编辑此默认模板，请执行以下步骤：

   a. 点编辑图标，然后在 **Edit** 选项卡中输入自定义主题和电子邮件正文。

   b. 点 **Preview** 选项卡查看您建议的模板。

   c. 点 **Apply** 将您的更改保存到模板。

   注意

   在查看特定报告的报告作业时，您可以看到在创建报告时是否使用默认模板或自定义模板。

4. 在 **Configure schedule** 部分中，为报告选择星期的频率和日期。

5. 点 **Next** 查看您的漏洞报告配置并完成它创建。

14.1.3.1.2. 检查并创建报告配置

您可以在创建漏洞报告配置前查看漏洞报告配置的详情。

流程

1. 在 **Review and create** 部分中，您可以查看报告配置参数、交付目标、电子邮件模板（如果您选择了电子邮件交付、交付计划和报告格式）。要进行任何更改，请点击 **Back** to go to the previous 部分，并编辑您要更改的字段。

2. 单击 **Create** 以创建报告配置并保存。

14.1.3.2. 漏洞报告权限

为您的用户帐户创建、查看和下载报告的能力取决于访问控制设置或角色和权限集。

例如，您只能查看、创建和下载用户帐户有权访问的数据报告。另外，还有以下限制：

- 您只能下载您生成的报告，您无法下载其他用户生成的报告。
- 报告权限会根据用户帐户的访问设置进行限制。如果帐户的访问设置发生了变化，旧的报告不会反映更改。例如，如果您给出了新权限，并希望查看这些权限允许的漏洞数据，则必须创建新的漏洞报告。

14.1.3.3. 编辑漏洞报告配置

您可以从报告配置列表中编辑现有漏洞报告配置，或者首先选择单独的报告配置。

流程

1. 要编辑现有漏洞报告配置，在 RHACS web 门户中，进入 **Vulnerability Management (2.0)** → **Vulnerability Reporting** 并选择以下方法之一：

   - 在报告配置列表中找到您要编辑的报告配置。点溢出菜单 ，然后选择 **Edit report**。
   - 点报告配置列表中的报告配置名称。然后，单击 **Actions** 并选择 **Edit report**。

2. 更改报告配置并保存。

14.1.3.4. 下载漏洞报告

您可以生成按需漏洞报告，然后下载它。

注意

您只能下载您生成的报告，您无法下载其他用户生成的报告。

流程

1. 在 RHACS web 门户中，进入 **Vulnerability Management (2.0)** → **Vulnerability Reporting**，在报告配置列表中找到您要用来创建可下载报告的报告配置。

2. 使用以下方法之一生成漏洞报告：

   - 从列表中生成报告：

     a. 点溢出菜单 ，然后选择 **Generate download**。**My active job status** 列显示报告创建的状态。**处理** 状态退出后，您可以下载报告。

   - 从报告窗口中生成报告：

     a. 点报告配置名称打开配置详情窗口。

     b. 点 **Actions** 并选择 **Generate download**。

3. 要下载报告，如果您要查看报告配置列表，请点击报告配置名称来打开报告。

4. 点 **All report jobs**。

5. 如果报告已完成，点 **Status** 列中的 **Ready for download** 链接。该报告采用 **.csv** 格式，压缩至 **.zip** 文件中以便下载。

14.1.3.5. 根据需要发送漏洞报告

您可以立即发送漏洞报告，而不是等待调度的发送时间。

流程

1. 在 RHACS web 门户中，进入 **Vulnerability Management (2.0)** → **Vulnerability Reporting**，在报告配置列表中找到您要发送的报告配置。

2. 点溢出菜单  ，然后选择 **Send report now**。

14.1.3.6. 克隆漏洞报告配置

您可以通过克隆漏洞报告配置来制作漏洞报告配置的副本。当您要使用次要更改重复使用报告配置时（如报告不同部署或命名空间中的漏洞）时，这非常有用。

流程

1. 在 RHACS web 门户中，进入 **Vulnerability Management (2.0)** → **Vulnerability Reporting**，并在报告配置列表中找到您要克隆的报告配置。
2. 单击 **Clone report**。
3. 进行您要报告参数和交付目的地的任何更改。
4. 点 **Create**。

14.1.3.7. 删除漏洞报告配置

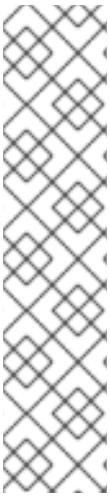
删除报告配置会删除配置以及之前使用此配置运行的任何报告。

流程

1. 在 RHACS web 门户中，进入 **Vulnerability Management (2.0)** → **Vulnerability Reporting**，并在报告列表中找到您要删除的报告配置。
2. 点溢出菜单  ，然后选择 **Delete report**。

14.1.3.8. 配置漏洞管理报告作业保留设置

您可以配置设置，以确定漏洞报告作业请求何时过期，以及报告作业的其他保留设置。



注意

这些设置不会影响以下漏洞报告作业：

- 处于 **WAITING** 或 **PREPARING** 状态的作业（未完成的作业）
- 最后一次成功调度的报告作业
- 最后一次成功发送了报告作业
- 最后成功下载报告作业
- 通过手动删除或配置可下载报告修剪设置，可下载报告文件没有被删除的可下载报告作业

流程

1. 在 RHACS web 门户中，进入 **Platform Configuration** → **System Configuration**。您可以为漏洞报告作业配置以下设置：

- 漏洞报告 运行历史记录保留 ：记录保留的天数，该日期将保留已运行的漏洞报告作业。此设置控制报告作业在 Vulnerability Management (2.0)→ Vulnerability Management (2.0)→ Vulnerability Reporting 下的 All report jobs 标签页中列出的天数。除截止日期以外的所有报告历史记录都会被修剪，但以下作业除外：
  - 未完成的作业。
  - 系统中仍然存在准备可下载报告的作业。
  - 每个作业类型的最后一次成功报告作业（计划的电子邮件、按需电子邮件或下载）。这样可确保用户具有每种类型最后一次运行的作业的信息。
- 准备好的可下载漏洞报告保留天数 ：在选择报告配置时，可以在 All report Jobs 选项卡中下载 on-demand 下载的漏洞报告作业的数量。
- 可下载的漏洞报告限制：分配给可下载的漏洞报告 作业的空间（以 MB 为单位）。达到限制后，下载队列中最旧的报告作业将被删除。

2. 要更改这些值，请单击 Edit，进行更改，然后单击 Save。

14.1.3.9. 升级到 RHACS 版本 4.3 及更新的版本时报告漏洞迁移

Red Hat Advanced Cluster Security for Kubernetes (RHACS)版本 4.3 包括在 Vulnerability Management 1.0 → Reporting 页面中在以前版本的 RHACS 中自动迁移漏洞报告配置。您可以通过点 Vulnerability Management (2.0)→ Vulnerability Reporting 来访问迁移的报告配置。在 RHACS web 门户或使用 API 中不再提供早期版本的报告配置。

RHACS 在迁移过程中执行以下操作：

- 报告配置会被复制，以创建报告的新版本，您可以通过点 Vulnerability Management (2.0)→ Vulnerability Reporting 来访问。
- 报告的原始名称用于在将报告迁移到新位置时使用。
- 在 Vulnerability Management 2.0 （技术预览）→ Reporting 页面创建的报告配置不受升级到 RHACS 版本 4.3 或更高版本的影响。用于访问这些报告配置的菜单项已重命名为 Vulnerability Management (2.0)，页面被重命名为 Vulnerability Reporting。
- 如果没有迁移之前使用 Vulnerability Management 1.0 页面创建的报告配置，因为通知程序已不存在，则该配置的详情将添加到 Central pod 生成的日志中。您可以使用日志中的详细信息，通过点 Vulnerability Management (2.0)→ Vulnerability Reporting 并添加新报告来重新创建报告配置。
- 对于之前使用 Vulnerability Management 1.0 页面创建的每个报告配置，最新的成功调度的报告作业会被迁移到报告配置的 All Report jobs 部分。要查看报告配置，请点 Vulnerability Management (2.0) → Vulnerability Reporting，然后点报告配置。

如果您需要从更新的版本中回滚到 RHACS 4.2，则会出现以下操作：

- 现在，随着迁移失效的报告配置会再次变为可正常运行的报告配置，点 Vulnerability Management 1.0 → Reporting 来获得。
- 迁移创建的报告配置可以正常工作，可通过点 Vulnerability Reporting 2.0 （技术预览）提供。您可以手动删除在 1.0 或 2.0 报告版本中创建的不需要的报告配置。
- 如果在回滚到 RHACS 4.2 或更早版本后更新了 Vulnerability Management 1.0 → Reporting 页面中的报告配置，则当系统再次升级时，这些更新可能不适用于迁移的报告配置。如果发生这种

情况，则报告配置的详情将添加到 Central pod 生成的日志中。您可以通过点 Vulnerability Management (2.0) → Vulnerability Reporting 并使用日志中的详情手动更新报告配置。

- 当您再次升级到 RHACS 版本 4.3 或更高版本时，在 Vulnerability Management 1.0 → Reporting 页面中创建的任何新报告配置都会被迁移。

14.1.4. 其他资源

- 创建和使用部署集合
- 将访问范围迁移到集合
- 配置电子邮件插件

14.2. 常见漏洞管理任务

常见的漏洞管理任务包括识别和排列漏洞的优先级，修复漏洞，以及对新威胁进行监控。以下是您可以在 Vulnerability Management → Dashboard 视图中执行的一些常见任务。

14.2.1. 查找影响您基础架构的关键 CVE

使用 漏洞管理 视图来识别影响您的平台最多的 CVE。

流程

1. 进入 RHACS 门户，从导航菜单中点 Vulnerability Management。
2. 在 Vulnerability Management 视图标头中选择 CVE。
3. 在 CVEs 视图中，选择 Env Impact 列标头，以根据环境影响降序排列 CVE。

14.2.2. 查找最易受攻击的镜像组件

使用 漏洞管理 视图来识别存在安全漏洞的镜像组件。

流程

1. 进入 RHACS 门户，从导航菜单中点 Vulnerability Management。
2. 在 Vulnerability Management 视图标头中选择 Application & Infrastructure → Components。
3. 在 Components 视图中，选择 CVEs 列标题，根据 CVE 的数量按降序排列组件。

14.2.3. 识别引入漏洞的容器镜像层

使用 Vulnerability Management 视图来识别存在安全漏洞的组件及其出现在的镜像层。

流程

1. 进入 RHACS 门户，从导航菜单中点 Vulnerability Management。
2. 从 Top Riskiest Images 小部件中选择 镜像，或者点击 Dashboard 顶部的 Images 按钮并选择镜像。

3. 在镜像详情视图中，选择 Dockerfile 旁边的展开图标来查看镜像组件摘要。

4. 选择特定组件的展开图标，以获取有关影响所选组件的 CVE 的更多详细信息。

您还可以通过进入到 Vulnerability Management (2.0) → Workload CVE 来查看此信息。如需更多信息，请参阅"添加资源"部分中的"查看漏洞管理(2.0)中的工作负载 CVE"。

14.2.4. 识别镜像中引入 CVE 的组件的 Dockerfile 行

您可以在使用 CVE 引入组件的镜像中识别特定的 Dockerfile 行。

流程

查看有问题的行：

1. 进入 RHACS 门户，从导航菜单中点 Vulnerability Management。

2. 从 Top Riskiest Images 小部件中选择 镜像，或者点击 Dashboard 顶部的 Images 按钮并选择镜像。

3. 在 Image Findings 中的 Image 详情视图中，CVE 列在 Observed CVEs, Deferred CVEs, 和 False positive CVEs 标签页中。

4. 找到您要进一步检查的 CVE。在 Affected Components 列中，点 <number> Components 链接来查看受 CVE 影响的组件列表。您可以在此窗口中执行以下操作：

   - 单击特定组件旁边的展开图标，以查看引入 CVE 的镜像中 Dockerfile 行。要解决 CVE，您需要在 Dockerfile 中更改这一行；例如，您可以升级组件。

   - 点组件的名称进入 组件概述页面，并查看组件的更多信息。

您还可以通过进入到 Vulnerability Management (2.0) → Workload CVE 来查看此信息。如需更多信息，请参阅"添加资源"部分中的"查看漏洞管理(2.0)中的工作负载 CVE"。

14.2.5. 仅查看可修复的 CVE 的详情

使用 漏洞管理 视图过滤和仅显示可修复的 CVE。

流程

> In the {product-title-short} portal, go to *Vulnerability Management*.
> . From the *Vulnerability Management* view header, select *Filter CVEs* -> *Fixable*.

14.2.6. 识别基础镜像的操作系统

使用 Vulnerability Management 视图来识别基础镜像的操作系统。

流程

1. 进入 RHACS 门户，从导航菜单中点 Vulnerability Management。

2. 在 Vulnerability Management 视图标头中选择 Images。

3. 查看 Image OS 列下所有镜像的基础操作系统(OS)和 OS 版本。

4. 选择一个镜像来查看其详情。基础操作系统也可以在 Image Summary → Details 和 Metadata 部分下提供。

注意

Red Hat Advanced Cluster Security for Kubernetes 在以下情况下 将镜像操作系统 列为 未知 信息：

- 操作系统信息不可用，或者
- 如果使用的镜像扫描程序不提供此信息。

Docker Trusted Registry、Google Container Registry 和 Anchore 不提供此信息。

您还可以通过进入到 Vulnerability Management (2.0) → Workload CVE 来查看此信息。如需更多信息，请参阅"添加资源"部分中的"查看漏洞管理(2.0)中的工作负载 CVE"。

14.2.7. 识别最佳风险对象

使用 漏洞管理 视图来识别环境中的主要风险对象。Top Risky 小部件显示有关环境中顶级风险镜像、部署、集群和命名空间的信息。风险取决于漏洞的数量及其 CVSS 分数。

流程

1. 进入 RHACS 门户，从导航菜单中点 Vulnerability Management。
2. 选择 Top Risky widget 标头，以选择风险镜像、部署、集群和命名空间。
   图表上的小圆圈代表所选对象（镜像、部署、集群、命名空间）。将鼠标悬停在圆圈上，以查看它们所代表的对象的概述。并选择圆圈来查看所选对象、相关实体以及它们之间的连接的详细信息。

   例如，如果您要 查看由 CVE Count 和 CVSS 分数划分的顶级风险 Deployment，则图表中的每个圆圈代表一个部署。

   - 将鼠标悬停在部署上时，您会看到部署概述，其中包括部署名称、集群和命名空间的名称、严重性、风险优先级、CVSS 和 CVE 计数（包括可修复）。
   - 当您选择部署时，会为所选部署打开 Deployment 视图。Deployment 视图显示部署的深入详情，并包含有关该部署的策略违反情况、常见漏洞、CVE 和风险镜像的信息。
3. 选择 View All on the widget 标头来查看所选类型的所有对象。例如，如果您根据 CVE Count 和 CVSS 分数选择了 Top Risky Deployments，您可以选择 View All 来查看基础架构中所有部署的详细信息。

14.2.8. 识别镜像和组件的主要风险

与 Top Risky 类似，顶级风险 小部件列出了主要风险和组件的名称。此小部件还包括列出的镜像中的 CVE 总数和可修复的 CVE 的数量。

流程

1. 进入 RHACS 门户，从导航菜单中点 Vulnerability Management。
2. 选择 Top Riskiest Images 小部件标头，以选择风险的镜像和组件。如果您要查看 Top Riskiest 镜像 ：

- 当您将鼠标悬停在列表中的镜像上时，您会看到镜像概述，其中包括镜像名称、扫描时间和 CVE 数量以及严重性(critical、高、中型和低)。
- 当您选择镜像时，会为所选镜像打开 Image 视图。Image 视图显示镜像的深入详情，包括 CVSS 分数、主要风险组件、可修复的 CVE 和镜像的 Dockerfile 的信息。

3. 选择 View All on the widget 标头来查看所选类型的所有对象。例如，如果您选择了Top Riskiest 组件，您可以选择 View All 来查看基础架构中所有组件的详细信息。

14.2.9. 查看镜像的 Dockerfile

使用 漏洞管理 视图查找镜像中漏洞的根本原因。您可以查看 Dockerfile，并准确查找 Dockerfile 中引入了漏洞以及与该单个命令关联的所有组件。

Dockerfile 部分显示以下信息：

- Dockerfile 中的所有层
- 每个层的说明及其值
- 每个层中包含的组件
- 每个层的组件中的 CVE 数量

当特定层引入的组件时，您可以选择展开图标来查看其组件的摘要。如果这些组件中存在 CVE，您可以选择单个组件的展开图标，以获取有关影响该组件的 CVE 的详情。

流程

1. 在 RHACS 门户中，进入 漏洞管理。
2. 从 Top Riskiest Images 小部件中选择 镜像，或者点击 Dashboard 顶部的 Images 按钮并选择镜像。
3. 在镜像详情视图中，选择 Dockerfile 旁边的展开图标来查看指令、值、创建日期和组件的摘要。
4. 选择单个组件的展开图标来查看更多信息。

您还可以通过进入到 Vulnerability Management (2.0)→Workload CVE 来查看此信息。如需更多信息，请参阅"添加资源"部分中的"查看漏洞管理(2.0)中的工作负载 CVE"。

14.2.10. 禁用识别节点中的漏洞

默认启用节点中的漏洞。您可以从 RHACS 门户禁用它。

流程

1. 在 RHACS 门户中，进入 Platform Configuration →Integrations。
2. 在 Image Integrations 下，选择 StackRox Scanner。
3. 从扫描程序列表中，选择 StackRox Scanner 来查看其详情。
4. 从 Types 中删除 Node Scanner 选项。
5. 选择 Save。

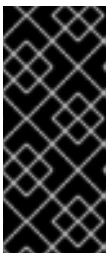
14.2.11. 扫描不活跃的镜像

Red Hat Advanced Cluster Security for Kubernetes (RHACS)每 4 小时扫描所有活跃的（部署）镜像，并更新镜像扫描结果以反映最新的漏洞定义。

您还可以将 RHACS 配置为自动扫描不活跃（未部署）镜像。

流程

1. 在 RHACS 门户中，进入 Vulnerability Management (2.0)→ Workload CVEs （技术预览）。
2. 点 <number> Images 显示镜像列表，并找到您要监视的镜像。
3. 点溢出菜单 ，然后选择 Watch image。然后，RHACS 会扫描镜像并显示错误或成功信息。
4. （可选） 要删除监视的镜像，请点溢出菜单 ，然后选择 Unwatch image。
5. （可选）您可以在页面标头中点 Manage watched images 来查看所有被监视的镜像列表，并添加附加镜像。



重要

在 RHACS 门户中，点 Platform Configuration → System Configuration 查看数据保留配置。

对于 System Configuration 页面中提到的天数，所有与从监视的镜像列表中删除的镜像相关的数据都会继续出现在 RHACS 门户中，仅在该周期过后删除。

6. 点 Close 返回 Workload CVEs 页面。

14.2.12. 创建用于阻止特定 CVE 的策略

您可以从 Vulnerability Management 视图创建新策略，或将特定的 CVE 添加到现有策略中。

流程

1. 从 Vulnerability Management 视图标头中点 CVE。
2. 您可以选择一个或多个 CVE 的复选框，然后单击 Add selected CVE to Policy(添加 图标)，或者将鼠标移到列表中的 CVE 上，然后选择 Add 图标。
3. 对于策略名称 ：
   - 要将 CVE 添加到现有策略中，请从下拉列表中选择现有策略。
   - 要创建新策略，请为新策略输入名称，然后选择 Create <policy_name>。
4. 为 Severity 选择一个值，可以是 Critical、High、 Medium 或 Low。
5. 选择适用于策略的生命周期阶段，从 Build, 或 Deploy。您还可以选择这两个生命周期阶段。

6. 在 Description 框中输入策略详情。
7. 如果要创建策略，请关闭 Enable Policy 切换功能，但在以后启用它。默认情况下 Enable Policy 是开启的。
8. 验证此策略中所含的 CVE。
9. 单击 Save Policy。

14.2.13. 查看最近检测到的漏洞

Vulnerability Management 视图中的 Recently Detected Vulnerabilities 会根据扫描时间和 CVSS 分数显示当前在扫描镜像时发现的安全漏洞列表。它还包含有关受 CVE 影响的镜像数量及其对环境的影响（百分比）的信息。

- 当您将鼠标悬停在列表中的 CVE 上时，您会看到 CVE 的概述，其中包括扫描时间、CVSS 分数、描述、影响，以及是否使用 CVSS v2 还是 v3 分数。
- 当您选择 CVE 时，为所选 CVE 打开 CVE 详情视图。CVE 详情视图中显示 CVE 的深入详情，以及它出现的组件、镜像和部署。
- 在 Recently Detected Vulnerabilities widget 标头中选择 View All，以查看基础架构中所有 CVE 的列表。您还可以过滤 CVE 列表。

14.2.14. 查看最常见的漏洞

漏洞 管理视图中的最常见漏洞 小部件显示影响由 CVSS 分数安排的最大部署和镜像数量的漏洞列表。

- 当您将鼠标悬停在列表中的 CVE 上时，您会看到 CVE 概述，其中包括、扫描时间、CVSS 分数、描述、影响，以及是否使用 CVSS v2 还是 v3 评分。
- 当您选择 CVE 时，为所选 CVE 打开 CVE 详情视图。CVE 详情视图中显示 CVE 的深入详情，以及它出现的组件、镜像和部署。
- 选择 View All on the most Common Vulnerabilities widget 标头来查看您的基础架构中的所有 CVE 列表。您还可以过滤 CVE 列表。要将 CVE 导出为 CSV 文件，请选择 Export → Download CVES 作为 CSV。

14.2.15. 识别具有最严重策略违反情况的部署

在 Vulnerability Management 视图中 带有最严重的策略违反小部件的 Deployment 会显示影响该部署的部署和严重性的漏洞列表。

- 当将鼠标悬停在列表中的部署上时，您会看到部署概述，其中包括部署名称、集群的名称以及部署所在的命名空间，以及失败的策略数量及其严重性。
- 当您选择部署时，会为所选部署打开 Deployment 视图。Deployment 视图显示部署的深入详情，并包含有关该部署的策略违反情况、常见漏洞、CVE 和风险镜像的信息。
- 选择 View All on the most Common Vulnerabilities widget 标头来查看您的基础架构中的所有 CVE 列表。您还可以过滤 CVE 列表。要将 CVE 导出为 CSV 文件，请选择 Export → Download CVES 作为 CSV。

14.2.16. 使用大多数 Kubernetes 和 Istio 漏洞查找集群

使用 Vulnerability Management (1.0)视图来识别环境中具有大多数 Kubernetes、 Red Hat OpenShift 和 Istio 漏洞（已弃用）的集群。

具有最编配器和 Istio 漏洞 小部件的集群显示集群列表，按 Kubernetes、 Red Hat OpenShift 和 Istio 漏洞（已弃用）的数量排名。列表顶部的集群是具有最高漏洞的集群。

流程

1. 点列表中的一个集群查看集群的详情。Cluster 视图包括：
   - Cluster Summary 部分，显示集群详情和元数据、顶级风险对象（部署、命名空间和镜像）、最近检测到的漏洞、风险镜像以及具有最严重策略违反情况的部署。
   - Cluster Findings 部分，其中包括失败策略列表以及可修复的 CVE 列表。
   - 相关的实体 部分，其中显示了集群包含的命名空间、部署、策略、镜像、组件和 CVE 的数量。您可以选择这些实体来查看更多详细信息。
2. 点小部件标头中的 View All，以查看所有集群的列表。

14.2.17. 识别节点中的漏洞

您可以使用 Vulnerability Management 视图来识别节点中的漏洞。识别的漏洞包括：

- Kubernetes 核心组件。
- 容器运行时(Docker、 CRI-O、 runC 和 containerd)。

注意

- Red Hat Advanced Cluster Security for Kubernetes 可以识别以下操作系统中的漏洞：
  - Amazon Linux 2
  - CentOS
  - Debian
  - Garden Linux (Debian 11)
  - Red Hat Enterprise Linux CoreOS (RHCOS)
  - Red Hat Enterprise Linux (RHEL)
  - Ubuntu (AWS、 Microsoft Azure、 GCP 和 GKE 特定版本)

流程

1. 在 RHACS 门户中，进入 Vulnerability Management → Dashboard。
2. 在 Dashboard 视图标头中选择 Nodes 来查看影响节点的所有 CVE 列表。
3. 从列表中选择节点，以查看影响该节点的所有 CVE 的详细信息。
   a. 当您选择节点时，为所选节点打开 Node details 面板。Node 视图显示节点的深入详情，并包含由 CVSS 分数以及该节点可修复的 CVE 的信息。

b. 选择 View All on the CVE by CVSS score widget 标头来查看所选节点上所有 CVE 的列表。您还可以过滤 CVE 列表。

c. 要将可修复的 CVE 导出为 CSV 文件，请在 Node Findings 部分选择 Export as CSV。

14.3. 扫描 RHCOS 节点主机

对于 OpenShift Container Platform，Red Hat Enterprise Linux CoreOS (RHCOS)是 control plane 唯一支持的操作系统。虽然对于节点主机，OpenShift Container Platform 支持 RHCOS 和 Red Hat Enterprise Linux (RHEL)。使用 Red Hat Advanced Cluster Security for Kubernetes (RHACS)，您可以扫描 RHCOS 节点漏洞并检测潜在的安全威胁。

RHACS 扫描在节点主机上安装的 RHCOS RPM，作为 RHCOS 安装的一部分，以了解任何已知的漏洞。

首先，RHACS 会分析并检测 RHCOS 组件。然后，它使用 RHEL 和 OpenShift 4.X Open Vulnerability 和评估语言(OVAL) v2 安全数据流来匹配识别的组件的漏洞。

注意

- 如果使用 roxctl CLI 安装 RHACS，您必须手动启用 RHCOS 节点扫描功能。当您在 OpenShift Container Platform 上使用 Helm 或 Operator 安装方法时，这个功能会被默认启用。

其他资源

- RHEL CoreOS 和 OCP 使用的 RHEL 版本

14.3.1. 启用 RHCOS 节点扫描

如果使用 OpenShift Container Platform，您可以使用 Red Hat Advanced Cluster Security for Kubernetes (RHACS)启用对 Red Hat Enterprise Linux CoreOS (RHCOS)节点的扫描。

先决条件

- 要扫描安全集群的 RHCOS 节点主机，您必须在 OpenShift Container Platform 4.11 或更高版本上安装了安全集群。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. 运行以下命令来更新合规性容器之一。

   - 对于禁用了指标的默认合规容器，请运行以下命令：

```
$ oc -n stackrox patch daemonset/collector -p '{"spec":{"template":{"spec":
{"containers":[{"name":"compliance","env":
[{"name":"ROX_METRICS_PORT","value":"disabled"},
{"name":"ROX_NODE_SCANNING_ENDPOINT","value":"127.0.0.1:8444"},
{"name":"ROX_NODE_SCANNING_INTERVAL","value":"4h"},
{"name":"ROX_NODE_SCANNING_INTERVAL_DEVIATION","value":"24m"},
{"name":"ROX_NODE_SCANNING_MAX_INITIAL_WAIT","value":"5m"},
{"name":"ROX_RHCOS_NODE_SCANNING","value":"true"},
{"name":"ROX_CALL_NODE_INVENTORY_ENABLED","value":"true"}]}]}}}}'
```

- 对于启用了 Prometheus 指标的合规性容器，请运行以下命令：

```
$ oc -n stackrox patch daemonset/collector -p '{"spec":{"template":{"spec":
{"containers":[{"name":"compliance","env":
[{"name":"ROX_METRICS_PORT","value":":9091"},
{"name":"ROX_NODE_SCANNING_ENDPOINT","value":"127.0.0.1:8444"},
{"name":"ROX_NODE_SCANNING_INTERVAL","value":"4h"},
{"name":"ROX_NODE_SCANNING_INTERVAL_DEVIATION","value":"24m"},
{"name":"ROX_NODE_SCANNING_MAX_INITIAL_WAIT","value":"5m"},
{"name":"ROX_RHCOS_NODE_SCANNING","value":"true"},
{"name":"ROX_CALL_NODE_INVENTORY_ENABLED","value":"true"}]}]}}}}'
```

2. 通过执行以下步骤更新 Collector DaemonSet (DS)：

   a. 运行以下命令，将新卷挂载添加到 Collector DS 中：

```
$ oc -n stackrox patch daemonset/collector -p '{"spec":{"template":{"spec":
{"volumes":[{"name":"tmp-volume","emptyDir":{}},{"name":"cache-
volume","emptyDir":{"sizeLimit":"200Mi"}}]}}}}'
```

   b. 运行以下命令添加新 NodeScanner 容器：

```
$ oc -n stackrox patch daemonset/collector -p '{"spec":{"template":{"spec":
{"containers":[{"command":["/scanner","--nodeinventory","--config=",""],"env":
[{"name":"ROX_NODE_NAME","valueFrom":{"fieldRef":
{"apiVersion":"v1","fieldPath":"spec.nodeName"}}},
{"name":"ROX_CLAIR_V4_SCANNING","value":"true"},
{"name":"ROX_COMPLIANCE_OPERATOR_INTEGRATION","value":"true"},
{"name":"ROX_CSV_EXPORT","value":"false"},
{"name":"ROX_DECLARATIVE_CONFIGURATION","value":"false"},
{"name":"ROX_INTEGRATIONS_AS_CONFIG","value":"false"},
{"name":"ROX_NETPOL_FIELDS","value":"true"},
{"name":"ROX_NETWORK_DETECTION_BASELINE_SIMULATION","value":"true"},
{"name":"ROX_NETWORK_GRAPH_PATTERNFLY","value":"true"},
{"name":"ROX_NODE_SCANNING_CACHE_TIME","value":"3h36m"},
{"name":"ROX_NODE_SCANNING_INITIAL_BACKOFF","value":"30s"},
{"name":"ROX_NODE_SCANNING_MAX_BACKOFF","value":"5m"},
{"name":"ROX_PROCESSES_LISTENING_ON_PORT","value":"false"},
{"name":"ROX_QUAY_ROBOT_ACCOUNTS","value":"true"},
{"name":"ROX_ROXCTL_NETPOL_GENERATE","value":"true"},
{"name":"ROX_SOURCED_AUTOGENERATED_INTEGRATIONS","value":"false"},
{"name":"ROX_SYSLOG_EXTRA_FIELDS","value":"true"},
{"name":"ROX_SYSTEM_HEALTH_PF","value":"false"},
{"name":"ROX_VULN_MGMT_WORKLOAD_CVES","value":"false"}],"image":"regis
try.redhat.io/advanced-cluster-security/rhacs-scanner-slim-
rhel8:4.4.3","imagePullPolicy":"IfNotPresent","name":"node-inventory","ports":
[{"containerPort":8444,"name":"grpc","protocol":"TCP"}],"volumeMounts":
[{"mountPath":"/host","name":"host-root-ro","readOnly":true},
{"mountPath":"/tmp/","name":"tmp-volume"},
{"mountPath":"/cache","name":"cache-volume"}]}]}}}}'
```

14.3.2. 分析和检测

当您将 RHACS 与 OpenShift Container Platform 搭配使用时，RHACS 会创建两个协调容器来分析和检测，Compliance 容器和 Node-inventory 容器。Compliance 容器已经是早期 RHACS 版本的一部分。但是，Node-inventory 容器带有 RHACS 4.0，仅适用于 OpenShift Container Platform 集群节点。

启动后，Compliance 和 Node-inventory 容器在五分钟内开始对 Red Hat Enterprise Linux CoreOS (RHCOS)软件组件的第一个清单扫描。接下来，Node-inventory 容器会扫描节点的文件系统来识别已安装的 RPM 软件包并报告 RHCOS 软件组件。之后，清单扫描会定期进行，通常每四个小时进行。您可以通过为 Compliance 容器配置 ROX_NODE_SCANNING_INTERVAL 环境变量来自定义默认间隔。

14.3.3. 漏洞匹配

Central 服务（包括 Central 和 Scanner）执行漏洞匹配。扫描程序使用红帽的开放漏洞和评估语言 (OVAL) v2 安全数据流来匹配 Red Hat Enterprise Linux CoreOS (RHCOS)软件组件上的漏洞。

与早期版本不同，RHACS 4.0 不再使用 Kubernetes 节点元数据来查找内核和容器运行时版本。相反，它使用已安装的 RHCOS RPM 来评估该信息。

14.3.4. 相关环境变量

您可以使用以下环境变量在 RHACS 上配置 RHCOS 节点扫描。

表 14.1. node-inventory 配置

环境变量	描述
ROX_NODE_SCANNING_CACHE_TIME	缓存清单被视为过时的时间。默认为 **ROX_NODE_SCANNING_INTERVAL** 的 90%，即 **3h36m**。
ROX_NODE_SCANNING_INITIAL_BACKOFF	如果找到 backoff 文件，则节点扫描的初始时间（以秒为单位）。默认值为 **30s**。
ROX_NODE_SCANNING_MAX_BACKOFF	backoff 的上限。默认值为 5m，是 Kubernetes 重启策略稳定性计时器的 50%。

表 14.2. 合规性配置

环境变量	描述
ROX_NODE_SCANNING_INTERVAL	节点扫描之间的间隔持续时间的基本值。deafult 值为 **4h**。
ROX_NODE_SCANNING_INTERVAL_DEVIATION	节点扫描持续时间可能与基本间隔时间不同。但是，最大值受 **ROX_NODE_SCANNING_INTERVAL** 限制。
ROX_NODE_SCANNING_MAX_INITIAL_WAIT	第一次节点扫描前等待的最长时间，这是随机生成的。您可以将此值设置为 **0,** 以禁用初始节点扫描等待时间。默认值为 **5m**。

14.3.5. 识别节点中的漏洞

您可以使用 Vulnerability Management 视图来识别节点中的漏洞。识别的漏洞包括：

- Kubernetes 核心组件。
- 容器运行时(Docker、CRI-O、runC 和 containerd)。

注意

- Red Hat Advanced Cluster Security for Kubernetes 可以识别以下操作系统中的漏洞：
  - Amazon Linux 2
  - CentOS
  - Debian
  - Garden Linux (Debian 11)
  - Red Hat Enterprise Linux CoreOS (RHCOS)
  - Red Hat Enterprise Linux (RHEL)
  - Ubuntu (AWS、Microsoft Azure、GCP 和 GKE 特定版本)

流程

1. 在 RHACS 门户中，进入 Vulnerability Management → Dashboard。
2. 在 Dashboard 视图标头中选择 Nodes 来查看影响节点的所有 CVE 列表。
3. 从列表中选择节点，以查看影响该节点的所有 CVE 的详细信息。
   a. 当您选择节点时，为所选节点打开 Node details 面板。Node 视图显示节点的深入详情，并包含由 CVSS 分数以及该节点可修复的 CVE 的信息。
   b. 选择 View All on the CVE by CVSS score widget 标头来查看所选节点上所有 CVE 的列表。您还可以过滤 CVE 列表。
   c. 要将可修复的 CVE 导出为 CSV 文件，请在 Node Findings 部分选择 Export as CSV。

第 15 章 响应违反情况

使用 Red Hat Advanced Cluster Security for Kubernetes (RHACS)，您可以查看策略违反情况，深入到违反情况的实际原因，并采取纠正措施。

RHACS 的内置策略识别各种安全发现，包括漏洞(CVE)、DevOps 最佳实践、高风险构建和部署实践以及可疑运行时行为。无论您使用默认开箱即用的安全策略，还是使用您自己的自定义策略，RHACS 会在启用的策略失败时报告违反情况。

15.1. VIOLATIONS 视图

您可以分析 Violations 视图中的所有违反情况，并采取正确的操作。

在 RHACS 门户中，进入 Violations 来查看发现的违反情况。

Violations 视图显示每行具有以下属性的违反情况列表：

- policy ：违反策略的名称。
- Entity: 发生违反情况的实体。
- 类型 ：实体类型，如部署、命名空间或集群。
- 强制 ：指示在发生违反时是否强制执行策略。
- 严重性 ：代表严重性 Low, Medium, High, 或 Critical。
- 类别 ：策略类别。策略类别在 Policy categories 选项卡中的 Platform Configuration → Policy Management 中列出。
- Lifecycle: 策略将应用到的阶段：Build, Deploy, 或 Runtime。
- Time: 发生违反时的日期和时间。

与其他视图类似，您可以执行以下操作：

- 选择一个列标题，以升序或降序排列违反情况。
- 使用过滤器栏过滤违反情况。如需更多信息，请参阅搜索和过滤部分。
- 在 Violations 视图中选择违反情况，以查看违反情况的详情。

15.1.1. 将违反情况标记为已解决

如果删除了具有运行时违反情况的策略，则不会从 Violations 页面删除违反情况。您可以通过将违反情况标记为 resolved 来手动删除违反情况。

流程

1. 选择 Violations 并在违反情况列表中找到违反情况。
2. 点 overflow 菜单 ，然后选择以下选项之一：

- **解决并添加到流程基线 ：解决违反情况，并将相关的进程添加到进程基准中。如果再次执行进程，则会显示新的违反情况。**
- **将标记为已解析 ：解决违反情况。**

15.2. 查看违反详情

当您在 Violations 视图中选择违反情况时，会打开一个窗口，其中包含有关违反情况的更多信息。它提供了按多个选项卡分组的详细信息。

15.2.1. 违反标签页

Violation Details 的 Violation 标签页解释了如何违反了策略。如果策略目标 deploy-phase 属性，您可以查看违反策略的特定值，如违反名称。如果策略目标运行时活动，您可以查看违反策略的进程的详细信息，包括其参数以及创建它的上级进程。

15.2.2. Deployment 标签页

Details 面板的 Deployment 选项卡显示违反情况的部署详情。

概述部分
部署概述 部分列出了以下信息：

- **部署 ID ：部署的字母数字标识符。**
- **部署名称 ：部署的名称。**
- **部署类型 ：部署的类型。**
- **Cluster ：部署容器的集群名称。**
- **Namespace ：部署的集群的唯一标识符。**
- **副本 ：复制部署的数量。**
- **Created ：创建部署的时间和日期。**
- **Updated ：更新部署的时间和日期。**
- **Labels ：应用到所选部署的标签。**
- **Annotations：应用到所选部署。**
- **Service Account ：所选部署的服务帐户的名称。**

容器配置部分
容器配置 部分列出以下信息：

- **容器 ：对于每个容器，提供以下信息：**
  - **镜像名称 ：所选部署的镜像名称。点名称查看有关镜像的更多信息。**
  - **资源 ：本节提供以下字段的信息：**
    - **CPU 请求（内核）：容器请求的内核数。**
    - **CPU 限制（内核）：容器可请求的最大内核数。**

- 内存请求(MB) ：容器请求的内存大小。
- 内存限制(MB) ：容器可请求的最大内存。

- 卷 ：挂载到容器中的卷（若有）。
- 机密 ：与所选部署关联的 Secret。对于每个 secret，提供以下字段的信息：
  - 名称 ：机密的名称。
  - 容器路径 ：存储 secret 的位置。
- 名称 ：要挂载该服务的位置的名称。
- 源 ：数据源路径。
- 目标 ：存储数据的路径。
- 类型 ：卷的类型。

端口配置部分
Port configuration 部分提供有关部署中端口的信息，包括以下字段：

- 端口 ：由部署公开的所有端口，以及与此部署和端口关联的任何 Kubernetes 服务（如果存在）。对于每个端口，会列出以下字段：
  - containerPort ：部署公开的端口号。
  - 协议 ：端口使用的协议，如 TCP 或 UDP。
  - exposure: 服务公开方法，如负载均衡器或节点端口。
  - exposureInfo ：本节提供了以下字段的信息：
    - 级别 ：指示服务在内部或外部公开端口。
    - serviceName ：Kubernetes 服务的名称。
    - serviceID ：存储在 RHACS 中的 Kubernetes 服务的 ID。
    - serviceClusterIp ： *集群中* 另一个部署或服务的 IP 地址，可用于访问该服务。这不是外部 IP 地址。
    - Service Port: 服务使用的端口。
    - NodePort ：外部流量进入节点的节点上的端口。
    - externalIPs ：可用于从集群外部访问服务的 IP 地址（如果存在）。此字段不适用于内部服务。

安全上下文部分
Security context 部分列出了容器是否作为特权容器运行。

- 特权 ：
  - 如果为 特权，则为 true。
  - 如果不是特权，则为 false。

网络策略部分
Network policy部分列出了包含违反情况的命名空间中的命名空间和所有网络策略。点网络策略名称查看网络策略的完整 YAML 文件。

15.2.3. 策略标签页

Details 面板的 Policy 选项卡显示导致违反情况的策略详情。

策略概述部分
Policy overview部分列出了以下信息：

- 严重性 ：对政策（关键、高、中等或低）的等级。
- 类别 ：策略的策略类别。策略类别在 Policy categories 选项卡中的 Platform Configuration → Policy Management 中列出。
- 类型 ：策略是生成用户（由用户创建的策略）还是系统策略（默认为内置在 RHACS 中）。
- 描述 ：有关策略警报的详细说明。
- Rationale：有关策略建立原因的信息及其重要原因。
- 指导 ：对如何解决违反情况的效果。
- MITRE ATT&CK：指示是否有适用于此策略的 MITRE战术和技术。

策略行为
Policy behavior部分提供以下信息：

- Lifecycle Stage: 策略所属的生命周期阶段：Build, Deploy, 或 Runtime。
- 事件源 ：此字段仅在生命周期阶段为 Runtime 时才适用。它可以是以下之一：
  - 部署 ：当事件源包括进程和网络活动、pod exec 和 pod 端口转发时，RHACS 会触发策略违反情况。
  - 审计日志 ：当事件源与 Kubernetes 审计日志记录匹配时，RHACS 会触发策略违反情况。
- 响应 ：响应可以是以下之一：
  - inform ：策略违反情况会在违反情况列表中生成违反情况。
  - inform 和 enforce ：强制使用违反情况。
- 强制 ：如果响应被设置为 Inform and enforce，列出了为以下阶段设置的强制类型：
  - 构建 ：当镜像与策略条件匹配时，RHACS 无法构建您的持续集成(CI)。
  - 部署 ：对于 Deploy 阶段，RHACS 会阻止在 RHACS 准入控制器配置并运行时与策略条件匹配的部署。
    - 在带有准入控制器强制的集群中，Kubernetes 或 OpenShift Container Platform API 服务器会阻止所有不合规的部署。在其他集群中，RHACS 编辑不合规部署，以防止调度 pod。
    - 对于现有部署，策略更改仅在发生 Kubernetes 事件时在下次检测条件时导致强制。有关强制的更多信息，请参阅"部署阶段的安全策略强制"。

- 运行时 ：当 pod 中的事件与策略条件匹配时，RHACS 会删除所有 pod。

策略条件部分
Policy criteria 部分列出了策略的策略标准。

15.2.3.1. 部署阶段的安全策略强制

Red Hat Advanced Cluster Security for Kubernetes 支持两种类型的安全策略强制进行部署时间策略强制：通过准入控制器和 RHACS Sensor 的软强制进行硬强制。准入控制器会阻止创建或更新违反策略的部署。如果准入控制器被禁用或不可用，则 Sensor 可以通过将违反策略部署到 0 的部署来缩减副本来执行强制。

警告

策略实施可能会影响运行应用程序或开发流程。在启用强制选项前，请通知所有利益相关者，并计划如何响应自动强制操作。

15.2.3.1.1. 硬强制

硬强制由 RHACS 准入控制器执行。在带有准入控制器强制的集群中，Kubernetes 或 OpenShift Container Platform API 服务器会阻止所有不合规的部署。准入控制器会阻止 CREATE 和 UPDATE 操作。任何满足启用了 deploy-time 强制配置的策略的 pod 创建或更新请求都将失败。

注意

Kubernetes 准入 webhook 仅支持 CREATE、UPDATE、DELETE 或 CONNECT 操作。RHACS 准入控制器只支持 CREATE 和 UPDATE 操作。kubectl patch、kubectl set 和 kubectl scale 等操作是 PATCH 操作，而不是 UPDATE 操作。因为 Kubernetes 不支持 PATCH 操作，所以 RHACS 无法对 PATCH 操作执行强制。

要进行阻塞，您必须在 RHACS 中为集群启用以下设置：

- 在 Object Creates 上强制 ：此切换在 Dynamic Configuration 部分中，控制准入控制服务的行为。您必须在打开的 Static Configuration 部分中具有 Configure Admission Controller Webhook 来侦听 Object Creates 开关才能正常工作。
- 在对象更新上强制 ：此切换在 Dynamic Configuration 部分中，控制准入控制服务的行为。您必须在打开的 Static Configuration 部分中具有 Configure Admission Controller Webhook 来侦听 Object Updates 切换。

如果您对 Static Configuration 设置进行了更改，您必须重新部署安全集群才能使这些更改生效。

15.2.3.1.2. 软强制

软强制由 RHACS Sensor 执行。这个强制可防止启动操作。使用软强制时，Sensor 将副本扩展到 0，并阻止调度 pod。在这个强制中，集群中提供了非就绪的部署。

如果配置了软强制，且 Sensor 停机，则 RHACS 无法执行强制。

15.2.3.1.3. 命名空间排除

默认情况下，RHACS 从强制阻止中排除某些管理命名空间，如 stackrox、kube-system 和 istio-system 命名空间。这样做的原因是，必须部署这些命名空间中的一些项目才能使 RHACS 正常工作。

15.2.3.1.4. 对现有部署的强制

对于现有部署，策略更改仅在发生 Kubernetes 事件时在下次检测条件时导致强制。如果对策略进行更改，您必须通过选择 Policy Management 并点 Reassess All 来重新评估策略。此操作会在所有现有部署中应用部署策略，无论是否有新的传入的 Kubernetes 事件。如果违反了策略，则 RHACS 执行强制。

其他资源

- 使用准入控制器强制

第 16 章 创建和使用部署集合

您可以使用 RHACS 中的集合来定义和命名一组资源，使用匹配的模式。然后，您可以将系统进程配置为使用这些集合。

目前，集合仅在以下条件下可用：

- 集合仅适用于部署。
- 您只能使用带有漏洞报告的集合。如需更多信息，请参阅附加资源部分中的 "Vulnerability reporting"。
- 只有在使用 PostgreSQL 数据库时，部署集合仅适用于 RHACS 客户。

注意

默认情况下，RHACS 云服务使用 PostgreSQL 数据库，在安装 RHACS 版本 4.0 及更新的版本中也会默认使用它。使用早于 3.74 的 RHACS 客户可以在红帽的帮助下迁移到 PostgreSQL 数据库。

16.1. 先决条件

用户帐户必须具有以下权限才能使用 Collections 功能：

- **WorkflowAdministration**：您必须具有 Read 访问权限来查看集合，Write 权限以添加、更改或删除集合。
- 部署 ：您需要 Read Access 或 Read 和 Write Access，以了解配置的规则如何与部署匹配。

这些权限包含在 **Admin** 系统角色中。有关角色和权限的更多信息，请参阅"添加资源"中的"管理 RHACS 中的 RBAC"。

16.2. 了解部署集合

部署集合仅适用于使用 PostgreSQL 数据库的 RHACS 客户。默认情况下，RHACS 云服务使用 PostgreSQL 数据库，在安装 RHACS 版本 4.0 及更新的版本中也会默认使用它。使用早于 3.74 的 RHACS 客户可以在红帽的帮助下迁移到 PostgreSQL 数据库。

RHACS 集合是一个用户定义的，名为 reference。它通过使用选择规则定义逻辑分组。这些规则可以与部署、命名空间或集群名称或标签匹配。您可以使用完全匹配或正则表达式指定规则。集合在运行时解析，并可以引用集合定义时不存在的对象。集合可以通过使用其他集合进行构建以描述复杂的层次结构。

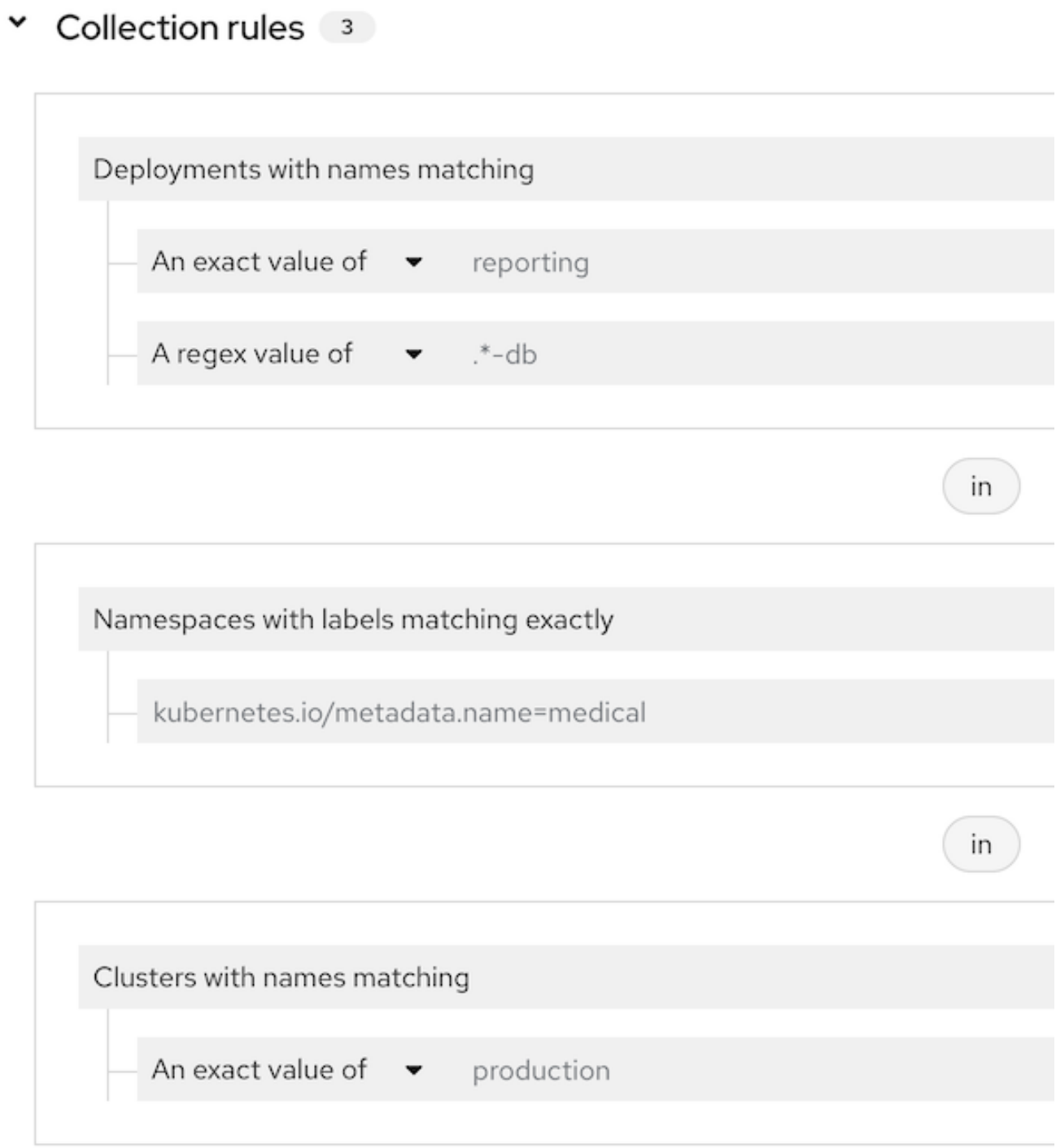
集合为您提供了一个语言来描述如何组织动态基础架构，无需克隆和重复编辑 RHACS 属性，如包含和排除范围。

您可以使用集合来标识系统中的任何一组部署，例如：

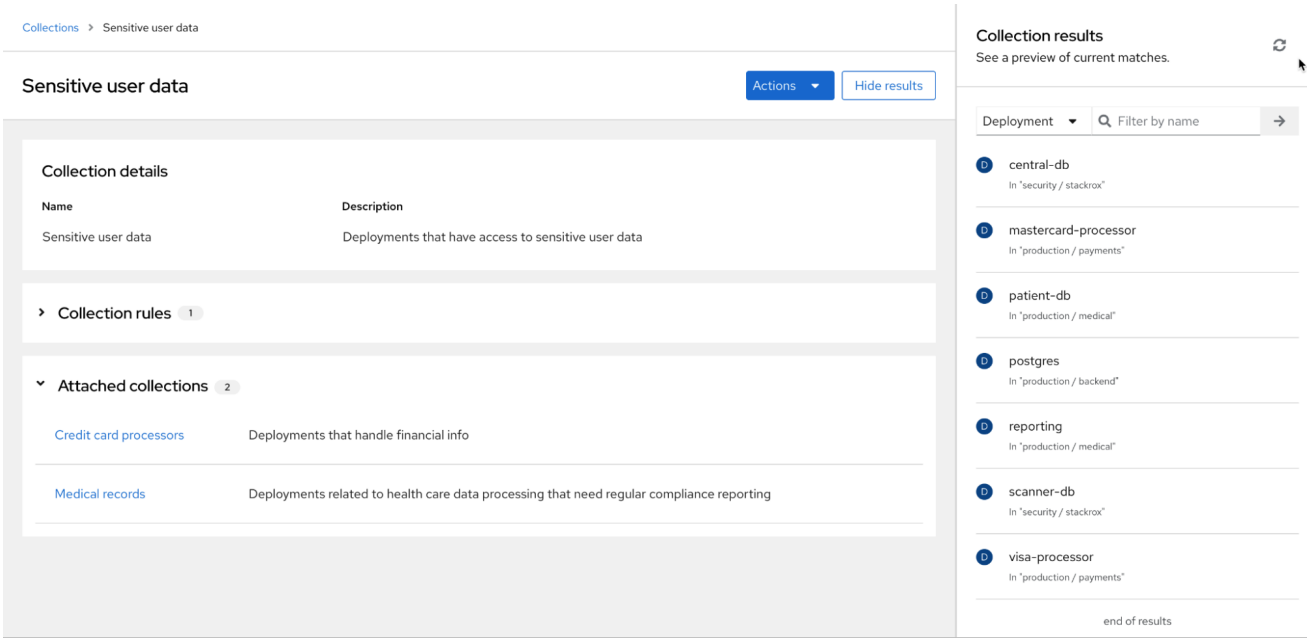
- 由特定开发团队拥有的基础架构区域
- 在开发或生产环境中运行时需要不同策略例外的应用程序
- 一个分布式应用程序，用于跨越多个命名空间，使用通用部署标签定义
- 整个生产环境或测试环境

可使用 RHACS 门户创建和管理集合。集合编辑器可帮助您在部署、命名空间和集群级别应用选择规则。您可以使用简单和复杂的规则，包括正则表达式。

您可以通过选择一个或多个部署、命名空间或集群来定义集合，如下图所示。此图显示了一个集合，其中包含名称 reporting 或名称中包含 db 的部署。该集合包括与命名空间中的名称与 kubernetes.io/metadata.name=medical 标签匹配的部署，以及在名为 production 的集群中。



集合编辑器还帮助您通过附加或嵌套其他集合来描述复杂的层次结构。编辑器提供了一个实时预览侧面板，可帮助您了解您要通过显示生成的与您配置的规则匹配的规则。下图提供了一个来自名为"Sensitive User Data"的集合的结果示例，其中包含一组集合规则（未显示）。"敏感用户数据"集合有两个附加的集合，即"Credit 卡处理器"和"Medical records"，每个集合都有自己的集合规则。侧面面板中显示的结果包括与为所有三个集合配置的规则匹配的项目。



16.3. 访问部署集合

要使用集合，请点击 Platform Configuration → Collections。该页面显示当前配置的集合列表。您可以执行以下操作：

- 通过在 Search by name 字段中输入文本来搜索集合，然后按 →。
- 点集合列表中的集合，以只读模式查看集合。
- 对于现有集合，点 ⋮ 来编辑、克隆或删除它。

注意

您不能删除在 RHACS 中活跃使用的集合。

- 点 Create collection 以创建新部署集合。

16.4. 创建部署集合

在创建集合时，您必须命名它并为集合定义规则。

流程

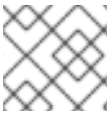
1. 在 Collections 页面中，点 Create collection。
2. 输入集合的名称和描述。
3. 在 Collection rules 部分中，必须至少执行以下操作之一：
   - 为集合定义规则：如需更多信息，请参阅"创建集合规则"部分。
   - 将现有集合附加到集合：如需更多信息，请参阅"添加附加集合"部分。

4. 您的规则配置的结果或选择附加的集合在 Collection 结果 实时预览面板中提供。点 Hide 结果 从显示中删除此面板。

5. 点 Save。

16.4.1. 创建集合规则

在创建集合时，必须至少配置一个规则，或将另一个集合附加到您要创建的新集合中。



注意

目前，集合仅适用于部署。

配置规则以选择要包含在集合中的资源。使用 preview 面板查看集合规则的结果（在配置它们时）。您可以以任何顺序配置规则。

流程

1. 在 Deployments 部分中，从下拉列表中选择以下选项之一：

   - 所有部署 ：包括集合中的所有部署。如果选择这个选项，则必须使用命名空间或集群或附加另一个集合来过滤集合。
   - 带有与名称匹配的部署，点此选项按名称选择，然后点击以下选项之一：
     - 选择 An exact value of，并输入部署的确切名称。
     - 选择 A regex value of来使用正则表达式来搜索部署。如果您不知道部署的确切名称，则此选项很有用。正则表达式是定义模式的字母、数字和符号的字符串。RHACS 使用此模式匹配字符或字符组并返回结果。有关正则表达式的更多信息，请参阅"添加资源"部分中的"常规-Expressions.info"。
   - 带有完全匹配标签的 部署：点击这个选项来选择与您输入的确切文本匹配的标签部署。标签必须是有效的 Kubernetes 标签，格式为 key=value。

2. 可选： 要使用与包含其他条件匹配的名称或标签添加更多部署，请点 OR 并配置另一个准确或正则表达式值。

以下示例提供了为医疗应用程序配置集合的步骤。在本例中，您希望集合包含 报告 部署、名为 patient-db 的数据库，您想要使用标签(key = kubernetes.io/metadata.name 和 value = health)选择命名空间。在本例中，执行以下步骤：

1. 在 Collection rules 中，选择 Deployments with name match。
2. 单击 An exact value of，再输入 reporting。
3. 单击 OR。
4. 点 A regex value of并输入 adtrust -db 以选择环境中名称以 db 结尾的所有部署。regex value 选项使用正则表达式进行模式匹配；有关正则表达式的更多信息，请参阅 Additional resources 部分中的 " regular-Expressions.info"。右侧的面板可能会显示您不想包含的数据库。您可以使用其他过滤器排除这些数据库。例如：
   a. 点 Namespaces with labels matching exactly并输入 kubernetes.io/metadata.name=medical 来只包括被标记为 medical 命名空间中的部署来根据命名空间进行过滤。

b. 如果您知道命名空间的名称，点 Namespaces with name match，并输入名称。

16.4.2. 添加附加的集合

如果要基于部署创建小集合，对集合进行分组，并将它们添加到其他集合中会很有用。您可以重复使用这些较小的集合并将其合并到更大的分级集合中。在您要创建的集合中添加额外的集合：

1. 执行以下操作之一：
   - 在 Filter by name 字段中输入 text，然后按 → 查看匹配的结果。
   - 点 Available collections 列表中的集合名称，以查看有关集合的信息，如集合的名称和规则，以及与该集合匹配的部署。
2. 查看集合信息后，关闭窗口以返回到 Attached collections 页面。
3. 单击 +Attach。Attached collections 部分列出了您附加的集合。

注意

当您添加附加的集合时，附加的集合会根据配置的选择规则包含结果。例如，如果附加的集合包含根据父集合中使用的规则过滤的资源，则这些项目仍然会因为附加的集合中的规则而添加到父集合中。附加的集合使用 OR 运算符扩展原始集合。

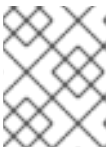
4. 点 Save。

16.5. 将访问范围迁移到集合

RHACS 中的数据库更改从 rocksdb 升级到 PostgreSQL 作为技术预览提供，从版本 3.74 开始，通常在 4.0 版本中提供。当数据库从 rocksdb 迁移到 PostgreSQL 时，漏洞报告中使用的现有访问范围将迁移到集合。您可以通过进入到 Vulnerability Management → Reporting 并查看报告信息来验证迁移是否会导致现有报告正确配置。

迁移过程为报告配置中使用的访问范围创建集合对象。RHACS 根据访问范围的复杂性，为单个访问范围生成两个或多个集合。为给定访问范围生成的集合包括以下类型：

- 嵌入式集合 ：为了模拟原始访问范围的确切选择逻辑，RHACS 生成一个或多个集合，与原始访问范围相同。集合的名称的格式是 System-generated embedded collection *number* for the scope，其中 *number* 是从 0 开始的值。



注意

这些嵌入式集合没有任何附加的集合。它们具有集群和命名空间选择规则，但没有部署规则，因为原始访问范围在部署中没有过滤。

- 访问范围的根集合：此集合添加到报告配置中。集合名称采用 System-generated root 集合的格式，范围为。此集合不定义任何规则，而是附加一个或多个嵌入式集合。这些嵌入式集合的组合会产生与原始访问范围相同的集群和命名空间选择。

对于定义集群或命名空间标签选择器的访问范围，RHACS 只能迁移在键和值之间具有"IN" Operator 的范围。使用 RHACS 门户创建的标签选择器访问范围默认使用 'IN' operator。不支持迁移使用 'NOT_IN', 'EXISTS' 和 'NOT_EXISTS' 运算符的范围。如果无法为访问范围创建集合，则会在迁移过程中创建日志消息。日志消息的格式如下：

> Failed to create collections for scope _scope-name_: Unsupported operator NOT_IN in scope's label selectors. Only operator 'IN' is supported.
> The scope is attached to the following report configurations: [list of report configs]; Please manually create an equivalent collection and edit the listed report configurations to use this collection. Note that reports will not function correctly until a collection is attached.

您还可以点 Vulnerability Management → Reporting 中的报告来查看报告信息页面。如果报告需要附加集合，则此页面将包含一条消息。

注意

迁移过程中不会删除原始访问范围。如果您只创建了在过滤漏洞管理报告中使用的访问范围，您可以手动删除访问范围。

16.6. 使用 API 管理集合

您可以使用 CollectionService API 对象配置集合。例如，您可以使用 CollectionService_DryRunCollection 返回与 RHACS 门户中实时预览面板的结果列表。如需更多信息，请参阅 RHACS 门户中的 Help → API 参考。

其他资源

- 在 RHACS 中管理 RBAC
- 漏洞报告
- 使用正则表达式： regular-Expressions.info

第 17 章 搜索和过滤

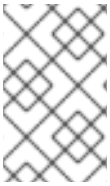
即时查找资源的功能对于保护您的集群非常重要。使用 Red Hat Advanced Cluster Security for Kubernetes 搜索功能更快地查找相关资源。例如，您可以使用它来查找公开给新发布的 CVE 的部署，或查找具有外部网络暴露的所有部署。

17.1. 搜索语法

搜索查询由两个部分组成：

- 标识您要搜索的资源类型的属性。
- 找到匹配资源的搜索词。

例如，若要在 visa-processor 部署中查找所有违反情况，搜索查询为 Deployment:visa-processor。在此搜索查询中，Deployment 是属性，visa-processor 是搜索词。



注意

您必须先选择一个属性，然后才能使用搜索术语。但是，在某些视图中，如 风险 视图和 冲突 视图，Red Hat Advanced Cluster Security for Kubernetes 会自动根据您输入的搜索词应用相关属性。

- 您可以在查询中使用多个属性。当您使用多个属性时，结果仅包含与所有属性匹配的项目。

  示例

  当您搜索 Namespace:frontend CVE:CVE-2018-11776 时，它只会返回在 frontend 命名空间中违反 CVE-2018-11776 的资源。

- 您可以将多个搜索词与每个属性一起使用。当您使用多个搜索词时，结果包括与任何搜索术语匹配的所有项目。

  示例

  如果您使用搜索查询 Namespace: frontend backend，它会从命名空间 frontend 或 backend 返回匹配结果。

- 您可以组合多个属性和搜索词对。

  示例

  搜索查询 Cluster:production Namespace:frontend CVE:CVE-2018-11776 会返回所有违反了 production 集群中的 frontend 命名空间中的 CVE-2018-11776 的资源。

- 搜索术语可以是单词的一部分，在这种情况下，Red Hat Advanced Cluster Security for Kubernetes 返回所有匹配的结果。

  示例

  如果您搜索 Deployment:def，则结果包括以 def 开始的所有部署。

- 要显式搜索特定术语，请使用引号中的搜索词。

  示例

  当您搜索 Deployment:"def" 时，结果仅包含部署 定义。

- 您还可以在搜索词前使用 r/ 来使用正则表达式。

  示例

  当您搜索 Namespace:r/stgemx 时，结果包括与命名空间 stackrox 和 stix 匹配。

- 使用 ! 表示您不希望结果的搜索词。

  示例

  如果您搜索 Namespace:!stackrox，则结果包括与 stackrox 命名空间以外的所有命名空间匹配。

- 使用比较操作 >, <, =, >=, 或 <= 来匹配一个特定的值或一个值的范围。

  示例

  如果您搜索 CVSS:>=6，则结果包括通用漏洞评分系统(CVSS)分数 6 或更高版本的所有漏洞。

17.2. SEARCH AUTOCOMPLETE

当您输入查询时，Red Hat Advanced Cluster Security for Kubernetes 会自动显示属性和搜索术语的相关建议。

17.3. 使用全局搜索

通过使用全局搜索，您可以在环境中搜索所有资源。根据您在搜索查询中使用的资源类型，结果按以下类别分组：

- 所有结果（列出所有类别的匹配结果）
- 集群
- Deployments
- 镜像
- 命名空间
- 节点
- 策略（policy）
- 策略类别 [1]
- 角色
- 角色绑定
- Secrets
- 服务帐户
- 用户和组
- 违反情况

1. 只有在您使用以下方法时，Policy 类别 选项才可用：
    - PostgreSQL 在 Red Hat Advanced Cluster Security for Kubernetes (RHACS)中作为后端数据库。
    - Red Hat Advanced Cluster Security Cloud Service (RHACS Cloud Service)。

这些类别被列为 RHACS 门户全局搜索页面上的表，您可以点类别名称来识别属于所选类别的结果。

要执行全局搜索，请在 RHACS 门户中选择 Search。

17.4. 使用本地页面过滤

您可以在 RHACS 门户的所有视图中使用本地页面过滤。本地页面过滤的工作方式与全局搜索类似，但只有相关属性可用。您可以选择搜索栏来显示特定视图的所有可用属性。

17.5. 常见搜索查询

以下是您可以使用 Red Hat Advanced Cluster Security for Kubernetes 运行的一些常见搜索查询。

查找受特定 CVE 影响的部署

查询	示例
CVE:<CVE_number>	CVE:CVE-2018-11776

查找特权运行部署

查询	示例
privileged:<true_or_false>	privileged:true

查找具有外部网络暴露的部署

查询	示例
exposure Level:<level>	公开级别：External

查找运行特定进程的部署

查询	示例
Process Name:<process_name>	Process Name:bash

查找具有严重但可修复的漏洞的部署

查询	示例
CVSS:<expression_and_score>	CVSS:>=6 Fixable:gem

查找使用通过环境变量公开的密码的部署

查询	示例
环境 Key:<query>	环境密钥：r/fusepassfuse

查找在其中有特定软件组件的运行部署

查询	示例
component:<component_name>	component:libgpg-error 或 component:sudo

查找用户或组

使用 Kubernetes Labels 和 Selectors, 以及 Annotations 来为您的部署附加元数据。然后，您可以根据应用的注解和标签查询以标识个人或组。

查找拥有特定部署的人员

查询	示例
deployment :<deployment_name> Label:<key_value> 或 Deployment:<deployment_name> Annotation:<key_value>	Deployment:app-server Label:team=backend

从公共 registry 查找部署镜像的人员

查询	示例
Image Registry:<registry_name> Label:<key_value> 或 Image Registry:<registry_name> Annotation:<key_value>	Image Registry:docker.io Label:team=backend

查找要部署到 default 命名空间的人员

查询	示例
namespace:default Label:<key_value> 或 Namespace:default Annotation:<key_value>	Namespace:default Label:team=backend

17.6. 搜索属性

以下是在 Red Hat Advanced Cluster Security for Kubernetes 中搜索和过滤时可以使用的搜索属性列表。

属性	描述
添加功能	为容器提供额外的 Linux 功能，例如，修改文件或执行网络操作的能力。

属性	描述
注解	任意的、不标识附加到编配器对象的元数据。
CPU 内核限制	允许资源使用的最大内核数。
CPU 内核请求	为给定资源保留的最小内核数。
CVE	常见的漏洞和风险，将其与特定 CVE 编号一起使用。
CVSS	通用漏洞评分系统，将其与 CVSS 分数一起使用并大于(>)、小于(<)或等于(=)符号。
类别	策略类别包括 DevOps 最佳实践、安全最佳实践、特权、漏洞管理、多个，以及您创建的任何自定义策略类别。
证书过期	证书到期日期.
集群	Kubernetes 或 OpenShift Container Platform 集群的名称。
集群 ID	Kubernetes 或 OpenShift Container Platform 集群的唯一 ID。
集群角色	使用 **true** 为命名空间范围的角色搜索集群范围的角色和 **false**。
组件	软件(daemond、docker)、对象（镜像、容器、服务）、registry (Docker 镜像的存储库)。
组件计数	镜像中的组件数量。
组件版本	软件、对象或注册表的版本。
创建时间	secret 对象的时间和日期。
Deployment	部署的名称。
部署类型	部署所基于的 Kubernetes 控制器的类型。
描述	部署的描述。
Dockerfile 指令关键字	镜像中的 Dockerfile 指令中的关键字。
Dockerfile 指令值	镜像中的 Dockerfile 指令中的值。
drop Capabilities	已从容器中丢弃的 Linux 功能。例如 **CAP_SETUID** 或 **CAP_NET_RAW**。
强制	分配给部署的强制类型。例如，**无**、**Scale 到 Zero Replicas** 或 **Add an Unsatisfiable Node Constraint**。

属性	描述
环境密钥	标签键值字符串的关键部分，这是元数据，用于进一步识别和整理容器环境。
环境值	标签键值字符串的值部分，这是元数据，用于进一步识别和整理容器环境。
公开的节点端口	公开节点端口的端口号。
公开服务	公开的服务的名称。
公开服务端口	公开服务的端口号。
公开级别	部署端口的暴露类型，如 **external** 或 **node**。
外部主机名	部署的外部端口暴露的主机名。
外部 IP	部署的外部端口暴露的 IP 地址。
可修复的 CVE 数量	镜像上可修复的 CVE 数量。
修复人	修复镜像中标记的漏洞的软件包版本字符串。
镜像	镜像的名称。
image 命令	镜像中指定的命令。
创建的镜像	创建镜像的时间和日期。
镜像条目点	镜像中指定的 entrypoint 命令。
Image Pull Secret	拉取镜像时使用的 secret 名称，如部署中指定的。
Image Pull Secret Registry	镜像 pull secret 的 registry 名称。
镜像 Registry	镜像 registry 的名称。
镜像远程	指明可远程访问的镜像。
镜像扫描时间	镜像最后一次扫描的时间和日期。
镜像标签	镜像的标识符。
镜像用户	容器镜像在运行时要使用的用户或组的名称。
镜像卷	容器镜像中配置的卷的名称。

属性	描述
不活跃部署	使用 **true** 搜索不活跃部署，为活跃部署搜索 **false**。
标签	标签键值字符串的 key-value 字符串，用于进一步识别和整理镜像、容器、守护进程、卷、网络和其他资源。
生命周期阶段	触发此策略的生命周期阶段的类型。
最大公开级别	对于部署，所有给定端口/服务的最大网络暴露级别。
内存限制(MB)	资源允许使用的最大内存量。
内存请求(MB)	为给定资源保留的最小内存量。
命名空间	命名空间的名称。
命名空间 ID	部署中包含命名空间对象的唯一 ID。
节点	节点的名称。
节点 ID	节点的唯一 ID。
Pod 标签	识别附加到单个 pod 的元数据的单个部分。
策略	安全策略的名称。
端口	部署公开的端口号。
端口协议	公开端口使用的 TCP 或 UDP 等 IP 协议。
优先级	部署的风险优先级。（仅在 **风险** 视图中提供。）
Privileged	使用 **true** 搜索特权运行部署，否则为 **false**。
Process Ancestor	部署中进程指示器的任何父进程的名称。
进程参数	部署中进程指示符的命令参数。
进程名称	部署中进程指示符的进程名称。
进程路径	指向容器中的二进制路径，用于部署中的进程指示符。
Process UID	部署中进程指示符的 UNIX 用户 ID。
只读 Root 文件系统	使用 **true** 搜索使用配置为只读的根文件系统运行的容器。

属性	描述
角色	Kubernetes RBAC 角色的名称。
角色绑定	Kubernetes RBAC 角色绑定的名称。
角色 ID	将 Kubernetes RBAC 角色绑定绑定到的角色 ID。
Secret	包含敏感信息的 secret 对象的名称。
Secret 路径	文件系统中 secret 对象的路径。
Secret 类型	secret 的类型，如证书或 RSA 公钥。
服务帐户	服务帐户或部署的服务帐户名称。
重要性	指明违反情况的重要性级别：Critical, High, Medium, Low。
主题	Kubernetes RBAC 中的主题的名称。
主题 Kind	Kubernetes RBAC 中的主题类型，如 **SERVICE_ACCOUNT**、**USER** 或 **GROUP**。
taint Effect	当前应用到节点的污点类型。
污点键	当前应用到节点的污点的键。
污点值	当前应用到节点的污点允许的值。
容限键	应用到部署的容限的关键。
容限值	应用到部署的容限的值。
违反	当由一个策略指定的条件尚未满足时，在 **Volations** 页面中显示的通知。
违反状态	使用它来搜索已解析的违反情况。
违反时间	首次发生违反的时间和日期。
卷目的地	数据卷的挂载路径。
卷名称	存储的名称。
卷 ReadOnly	使用 **true** 搜索挂载为只读的卷。
卷源	指明置备卷的表单（例如，**persistentVolumeClaim** 或 **hostPath**）。

属性	描述
卷类型	卷的类型。

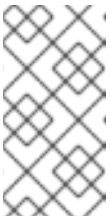
第 18 章 管理用户访问权限

18.1. 在 RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES 中管理 RBAC

Red Hat Advanced Cluster Security for Kubernetes (RHACS)附带基于角色的访问控制(RBAC)，可用于配置角色，并授予不同用户的 Red Hat Advanced Cluster Security for Kubernetes 的不同级别访问权限。

从 3.63 版本开始，RHACS 包含一个有范围的访问控制功能，可让您配置精细的和特定权限集，以定义给定 RHACS 用户或一组用户如何与 RHACS 交互，哪些资源可以访问哪些操作。

- *角色是*权限集和访问范围的集合。您可以通过指定规则将角色分配给用户和组。您可以在配置身份验证供应商时配置这些规则。Red Hat Advanced Cluster Security for Kubernetes 中有两种角色：
  - 由红帽创建且无法更改的系统角色。
  - 自定义角色，Red Hat Advanced Cluster Security for Kubernetes 管理员可以随时创建并更改。



注意

- 如果为用户分配多个角色，则它们可以访问所分配角色的组合权限。
- 如果您的用户分配了自定义角色，并且删除该角色，则所有关联的用户都转移到您配置的最小访问角色。

- *权限集*是一组权限，用于定义角色对给定资源可以执行的操作。*资源是* Red Hat Advanced Cluster Security for Kubernetes 的功能，您可以设置 view (读取)和修改(写入)权限。Red Hat Advanced Cluster Security for Kubernetes 中有两种权限集：
  - 系统权限集，由红帽创建且无法更改。
  - 自定义权限集，Red Hat Advanced Cluster Security for Kubernetes 管理员可以随时创建并更改。
- *访问*范围是一组用户可以访问的 Kubernetes 和 OpenShift Container Platform 资源。例如，您可以定义一个访问权限范围，仅允许用户访问给定项目中 pod 的信息。Red Hat Advanced Cluster Security for Kubernetes 中有两种访问范围：
  - 系统访问范围，由红帽创建且无法更改。
  - 自定义访问范围，Red Hat Advanced Cluster Security for Kubernetes 管理员可以随时创建并更改。

18.1.1. 系统角色

Red Hat Advanced Cluster Security for Kubernetes (RHACS)包括一些默认系统角色，您可以在创建规则时应用到用户。您还可以根据需要创建自定义角色。

系统角色	描述
Admin	此角色面向管理员。使用它提供对所有资源的读写访问权限。
分析	此角色适用于无法进行任何更改但可以查看所有内容的用户。使用它为所有资源提供只读访问权限。
持续集成	此角色适用于 CI （持续集成）系统，包括强制执行部署策略所需的权限集。
网络 Graph Viewer	此角色适用于需要查看网络图形的用户。
无	此角色对任何资源没有读写访问权限。您可以将此角色设置为所有用户的最低访问角色。
Sensor Creator	RHACS 使用此角色自动执行新的集群设置。它包括在安全集群中创建 Sensors 的权限。
漏洞管理批准器	此角色允许您提供对批准漏洞延迟或假正请求的访问。
漏洞管理请求器	此角色允许您提供对请求漏洞延迟或假正的访问。
漏洞报告 Creator	此角色允许您创建和管理调度漏洞报告的漏洞报告配置。

18.1.1.1. 查看系统角色的权限集和访问范围

您可以查看默认系统角色的权限集和访问权限范围。

流程

1. **在 RHACS 门户中，进入 Platform Configuration → Access control。**
2. **选择 Roles。**
3. **点其中一个角色查看其详情。详情页面显示 slected 角色的权限集和访问范围。**

注意

您无法修改默认系统角色的权限集和访问权限范围。

18.1.1.2. 创建自定义角色

您可以从 Access Control 视图创建新角色。

先决条件

- **您必须具有 Admin 角色，或具有为 AuthProvider 和 Role 资源设置权限的权限的 Admin 角色，以创建、修改和删除自定义角色。**
- **在创建角色前，您必须为自定义角色创建权限集和访问权限范围。**

流程

1. 在 RHACS 门户中，进入 Platform Configuration → Access Control。
2. 选择 Roles。
3. 单击 Create role。
4. 为新角色输入 Name 和 Description。
5. 为角色选择 Permission set。
6. 为角色选择一个 Access 范围。
7. 点击 Save。

其他资源

- 创建自定义权限集
- 创建自定义访问范围

18.1.1.3. 为用户或组分配角色

您可以使用 RHACS 门户将角色分配给用户或组。

流程

1. 在 RHACS 门户中，进入 Platform Configuration → Access Control。
2. 从身份验证提供程序列表中，选择身份验证提供程序。
3. 单击 Edit minimum role 和 rules。
4. 在 Rules 部分下，点 Add new rule。
5. 对于 Key，请从 userid, name, email 或 group 中选择一个值。
6. 对于 Value，根据您选择的键输入用户 ID、名称、电子邮件地址或组的值。
7. 点 Role 下拉菜单，再选择您要分配的角色。
8. 点击 Save。

您可以为每个用户或组重复这些指令，并分配不同的角色。

18.1.2. 系统权限集

Red Hat Advanced Cluster Security for Kubernetes 包括了一些可应用到角色的默认系统权限集。您还可以根据需要创建自定义权限集。

权限集	描述
Admin	提供对所有资源的读写访问权限。

权限集	描述
分析	为所有资源提供只读访问权限。
持续集成	此权限集针对 CI （持续集成）系统，并包含强制执行部署策略所需的权限。
网络 Graph Viewer	提供查看网络图形的最小权限。
None	任何资源都不允许读和写权限。
Sensor Creator	为在安全集群中创建 Sensors 所需的资源提供权限。

18.1.2.1. 查看系统权限集的权限

您可以查看 RHACS 门户中设置的系统权限。

流程

1. **在 RHACS 门户中，进入 Platform Configuration → Access control。**
2. **选择 Permission set。**
3. **点其中一个权限集来查看其详情。详情页面显示资源列表及其所选权限集的权限。**

注意

您无法修改系统权限集的权限。

18.1.2.2. 创建自定义权限集

您可以从 Access Control 视图创建新权限集。

先决条件

- **您必须具有 Admin 角色，或具有为 AuthProvider 和 Role 资源设置权限的权限的 Admin 角色，以创建、修改和删除权限集。**

流程

1. **在 RHACS 门户中，进入 Platform Configuration → Access Control。**
2. **选择 Permission set。**
3. **点 Create permissions set。**
4. **为新权限集输入 Name 和 Description。**
5. **对于每个资源，在 Access level 列下，从 No access,Read access, 或 Read and Write access 中选择其中一个权限。**

警告

- 如果要为用户配置权限集，您必须为以下资源授予只读权限：
  - 警报
  - 集群
  - Deployment
  - 镜像
  - NetworkPolicy
  - NetworkGraph
  - 工作流管理
  - Secret
- 在创建新权限集时，这些权限会预先选中。
- 如果您没有授予这些权限，用户将遇到 RHACS 门户中查看页面的问题。

6. 点击 Save。

18.1.3. 系统访问范围

Red Hat Advanced Cluster Security for Kubernetes 包括了一些您可以应用到角色的默认系统访问范围。您还可以根据需要创建自定义访问范围。

Acces 范围	描述
不受限制	提供对 Red Hat Advanced Cluster Security for Kubernetes 监视器的所有集群和命名空间的访问权限。
拒绝所有	不提供任何 Kubernetes 和 OpenShift Container Platform 资源的访问权限。

18.1.3.1. 查看系统访问范围的详情

您可以查看允许且不允许在 RHACS 门户中的 Kubernetes 和 OpenShift Container Platform 资源。

流程

1. 在 RHACS 门户中，进入 Platform Configuration → Access control。
2. 选择 Access scopes。

3. 点一个访问范围来查看其详情。详情页面显示集群和命名空间列表，以及所选访问范围允许哪些列表。

注意

您无法修改系统访问范围允许的资源。

18.1.3.2. 创建自定义访问范围

您可以从 Access Control 视图中创建新的访问范围。

先决条件

- 您必须具有 Admin 角色，或具有为 **AuthProvider** 和 **Role** 资源设置权限的权限的 Admin 角色，以创建、修改和删除权限集。

流程

1. 在 RHACS 门户中，进入 Platform Configuration → Access control。
2. 选择 Access scopes。
3. 单击 Create access scope。
4. 为新访问范围输入 Name 和 Description。
5. 在 Allowed resources 部分下：
   - 使用 Cluster filter 和 Namespace 过滤器字段过滤列表中可见的集群和命名空间列表。
   - 展开 Cluster name 以查看该集群中的命名空间列表。
   - 要允许访问集群中的所有命名空间，请切换 Manual 选择 列中的开关。

注意

通过访问特定集群，用户可以访问集群范围内的以下资源：

- OpenShift Container Platform 或 Kubernetes 集群元数据和安全信息
- 授权集群的合规性信息
- 节点元数据和安全信息
- 访问该集群中的所有命名空间及其关联的安全信息

   - 要允许访问命名空间，请切换命名空间的 Manual 选择栏中的开关。

注意

通过访问特定命名空间，可以访问命名空间范围内以下信息：

- 部署的警报和违反情况
- 镜像的漏洞数据
- 部署元数据和安全信息
- 角色和用户信息
- 部署的网络图、策略和基准信息
- 处理信息和流程基准配置
- 对每个部署有优先级的风险信息

6. 如果要允许根据标签访问集群和命名空间，请点击 Label selection rules 部分下的 Add label selector。然后单击 Add rule，为标签选择器指定 Key 和 Value 对。您可以为集群和命名空间指定标签。
7. 点击 Save。

18.1.4. 资源定义

Red Hat Advanced Cluster Security for Kubernetes 包括了许多资源。下表列出了 Red Hat Advanced Cluster Security for Kubernetes 资源，并描述了用户可以使用 读取或写入权限 执行的操作。

注意

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

资源	读取权限	写入权限
权限	查看与 Red Hat Advanced Cluster Security for Kubernetes 实例匹配的用户元数据的单点登录(SSO)和基于角色的访问控制(RBAC)规则的配置，以及访问 Red Hat Advanced Cluster Security for Kubernetes 实例的用户，包括身份验证供应商为其提供的元数据。	创建、修改或删除 SSO 配置及配置的 RBAC 规则。

资源	读取权限	写入权限
管理	查看以下项目： • 数据保留、安全通知和其他相关配置的选项 • Red Hat Advanced Cluster Security for Kubernetes 组件中的当前日志记录详细程度 • 上传的探测文件的清单内容 • 现有镜像扫描程序集成 • 自动升级的状态 • 有关 Red Hat Advanced Cluster Security for Kubernetes 服务到服务身份验证的元数据 • 扫描程序捆绑包的内容 (download)	编辑以下项目： • 数据保留、安全通知和相关配置 • 日志记录级别 • Central 中的支持软件包（上传） • 镜像扫描程序集成 (create/modify/delete) • 安全集群的自动升级（启用/禁用） • 服务到服务身份验证凭据 (revoke/re-issue)
警报	查看现有的策略违反情况。	解析或编辑策略违反情况。
CVE	*只限内部使用*	*只限内部使用*
集群	查看现有的安全集群。	添加新的安全集群并修改或删除现有集群。
Compliance	查看合规性标准和结果、最近运行合规性以及相关的完成状态。	触发合规性运行。
Deployment	查看安全集群中的部署（工作负载）。	N/A
DeploymentExtension	查看以下项目： • 进程基准 • 部署中的进程活动 • 风险结果	修改以下项目： • 进程基准（添加或删除进程）
检测	针对镜像或部署 YAML 检查构建时间策略。	N/A
Image	查看镜像、它们的组件及其漏洞。	N/A
集成	查看集成及其配置，包括备份、registry、镜像签名、通知系统和 API 令牌。	添加、修改和删除集成及其配置以及 API 令牌。

资源	读取权限	写入权限
K8sRole	在安全集群中查看 Kubernetes RBAC 的角色。	N/A
K8sRoleBinding	查看安全集群中 Kubernetes RBAC 的角色绑定。	N/A
K8sSubject	在安全集群中查看 Kubernetes RBAC 的用户和组。	N/A
命名空间	查看安全集群中的现有 Kubernetes 命名空间。	N/A
NetworkGraph	查看安全集群中的活跃和允许的网络连接。	N/A
NetworkPolicy	查看安全集群中的现有网络策略并模拟更改。	在安全集群中应用网络策略更改。
节点	查看安全集群中的现有 Kubernetes 节点。	N/A
工作流管理	查看所有资源集合。	添加、修改或删除资源集合。
角色	查看现有的 Red Hat Advanced Cluster Security for Kubernetes RBAC 角色及其权限。	添加、修改或删除角色及其权限。
Secret	查看安全集群中有关 secret 的元数据。	N/A
ServiceAccount	列出安全集群中的 Kubernetes 服务帐户。	N/A
VulnerabilityManagementApprovals	查看所有待处理的延迟或对漏洞的假请求。	批准或拒绝任何待处理的延迟或假的正请求，并将任何之前批准的请求移到观察到。
VulnerabilityManagementRequests	查看所有待处理的延迟或对漏洞的假请求。	在漏洞上请求延迟，将其标记为假正，或者将同一用户发出的待处理或之前批准的请求返回给观察。
WatchedImage	查看取消部署和监控的镜像。	配置监视的镜像。
工作流管理	查看所有资源集合。	创建、修改或删除资源集合。

18.1.5. 身份验证和授权资源的声明配置

您可以使用声明性配置进行身份验证和授权资源，如身份验证供应商、角色、权限集和访问范围。有关如何使用声明性配置的说明，请参阅"添加资源"部分中的"使用声明性配置"。

其他资源

- 使用声明配置

18.2. 启用 PKI 身份验证

如果使用企业证书颁发机构(CA)进行身份验证，您可以配置 Red Hat Advanced Cluster Security for Kubernetes (RHACS)来使用其个人证书验证用户。

配置 PKI 身份验证后，用户和 API 客户端可以使用其个人证书登录。没有证书的用户仍然可以使用其他身份验证选项，包括 API 令牌、本地管理员密码或其他身份验证提供程序。PKI 身份验证在与 Web UI、gRPC 和 REST API 相同的端口号上提供。

当您配置 PKI 身份验证时，Red Hat Advanced Cluster Security for Kubernetes 默认使用与 PKI、Web UI、gRPC、其他单点登录(SSO)提供程序和 REST API 相同的端口。您还可以使用 YAML 配置文件配置和公开端点，为 PKI 身份验证配置单独的端口。

18.2.1. 使用 RHACS 门户配置 PKI 身份验证

您可以使用 RHACS 门户配置公钥基础架构(PKI)身份验证。

流程

1. 在 RHACS 门户中，进入 Platform Configuration → Access Control。
2. 单击 Create Auth Provider，然后从下拉列表中选择 User Certificates。
3. 在 Name 字段中，指定此身份验证提供程序的名称。
4. 在 CA 证书(PEM) 字段中，以 PEM 格式粘贴 root CA 证书。
5. 为使用 PKI 身份验证 访问 RHACS 的用户分配最小访问角色。用户必须具有授予此角色的权限或具有较高权限的角色才能登录到 RHACS。

   提示

   为安全起见，红帽建议在完成设置时首先将 最小访问角色设置为 None。之后，您可以返回 Access Control 页面，根据您的身份提供程序中的用户元数据设置更多定制的访问规则。

6. 要为访问 RHACS 的用户和组添加访问规则，请点击 Rules 部分中的 Add new rule。例如，要为名为 administrator 的用户提供 Admin 角色，您可以使用以下键值对创建访问规则：

键	值
名称	Administrator
角色	Admin

7. 点 Save。

18.2.2. 使用 roxctl CLI 配置 PKI 身份验证

您可以使用 roxctl CLI 配置 PKI 身份验证。

流程

- 运行以下命令：

```
$ roxctl -e <hostname>:<port_number> central userpki create -c <ca_certificate_file> -r
<default_role_name> <provider_name>
```

18.2.3. 更新身份验证密钥和证书

您可以使用 RHACS 门户更新身份验证密钥和证书。

流程

1. 创建新的身份验证提供程序。
2. 将角色映射从旧身份验证供应商复制到新的身份验证供应商。
3. 使用旧的 root CA 密钥重命名或删除旧的身份验证供应商。

18.2.4. 使用客户端证书登录

配置 PKI 身份验证后，用户在 RHACS 门户登录页面中看到证书提示。只有用户系统上安装了由配置的 root CA 信任的客户端证书时，提示符才会显示。

使用本节中介绍的流程使用客户端证书登录。

流程

1. 打开 RHACS 门户。
2. 在浏览器提示符处选择证书。
3. 在登录页面中，选择要使用证书登录的身份验证提供程序名称选项。如果您不想使用证书登录，也可以使用管理员密码或其他登录方法登录。

注意

使用客户端证书登录到 RHACS 门户后，除非重启浏览器，否则无法使用不同的证书登录。

18.3. 了解身份验证供应商

身份验证供应商连接到用户身份的第三方源（如身份提供程序或 IDP），获取用户身份，根据该身份发出令牌，并将令牌返回到 Red Hat Advanced Cluster Security for Kubernetes (RHACS)。此令牌允许 RHACS 授权用户。RHACS 在用户界面和 API 调用中使用令牌。

安装 RHACS 后，您必须设置 IDP 来授权用户。

注意

如果您使用 OpenID Connect (OIDC)作为 IDP，RHACS 依赖于映射规则来检查来自用户 ID 令牌或 UserInfo 端点响应中特定声明的值，如 组、电子邮件、userid 和名称，以授权用户。如果没有这些详细信息，则映射无法成功，用户也不会获得对所需资源的访问权限。因此，您需要确保从 IDP 授权用户（如 组 ）所需的声明包含在 IDP 的身份验证响应中，以启用成功映射。

其他资源

- 将 Okta Identity Cloud 配置为 SAML 2.0 身份提供程序
- 将 Google Workspace 配置为 OIDC 身份提供程序
- 将 OpenShift Container Platform OAuth 服务器配置为身份提供程序
- 使用 SSO 配置将 Azure AD 连接到 RHACS

18.3.1. 声明映射

声明是身份提供程序包含令牌内用户的数据。

使用声明映射，您可以指定 RHACS 是否应该将其从 IDP 接收到的 claim 属性自定义到 RHACS-issued 令牌中的另一个属性。如果不使用声明映射，RHACS 不会在 RHACS-issued 令牌中包含 claim 属性。

例如，您可以使用声明映射，从用户身份中的角色映射到 RHACS-issued 令牌中的组。

RHACS 为每个身份验证供应商使用不同的默认声明映射。

18.3.1.1. OIDC 默认声明映射

以下列表提供默认的 OIDC 声明映射：

- sub 到 userid
- name 到 name
- email 到 email
- 组 到 组

18.3.1.2. Auth0 默认声明映射

Auth0 默认声明映射与 OIDC 默认声明映射相同。

18.3.1.3. SAML 2.0 默认声明映射

以下列表适用于 SAML 2.0 默认声明映射：

- subject.NameID 映射到 userid
- 来自响应的每个 SAML AttributeStatement.Attribute 都会被映射到其名称

18.3.1.4. Google IAP 默认声明映射

以下列表提供了 Google IAP 默认声明映射：

- sub 到 userid
- email 到 email
- hd 到 hd
- google.access_levels 到 access_levels

18.3.1.5. 用户证书默认声明映射

用户证书与所有其他身份验证提供程序不同，因为与第三方 IDP 通信，它们从用户所使用的证书获取用户信息。

用户证书的默认声明映射包括：

- CertFingerprint to userid
- subject → Common Name 到 name
- EmailAddresses 到 email
- subject → Organizational Unit to groups

18.3.1.6. OpenShift Auth 默认声明映射

以下列表提供了 OpenShift Auth 默认声明映射：

- 组 到 组
- UID 到 userid
- name 到 name

18.3.2. 规则

要授权用户，RHACS 依赖于映射规则来检查特定声明的值，如来自用户身份的组、电子邮件、userid 和名称。规则允许映射具有特定值的属性的用户到特定角色。例如，规则可以包括以下内容：'key' 是 电子邮件，值为 john@redhat.com，role 为 Admin。

如果缺少声明，映射将无法成功，用户也不会访问所需资源。因此，要启用成功映射，您必须确保来自 IDP 的身份验证响应包含授权用户所需的声明，如 组。

18.3.3. 最低访问角色

RHACS 使用特定身份验证供应商发布的 RHACS 令牌为每个调用者分配最小访问角色。默认将最小访问角色设置为 None。

例如，假设有一个具有最低访问的角色 Analyst 的身份验证提供程序。在这种情况下，使用此提供程序登录的所有用户都将为其分配 Analyst 角色。

18.3.4. 所需属性

必要属性可以根据用户身份是否具有带有特定值的属性来限制发出 RHACS 令牌。

例如，您只能将 RHACS 配置为仅在带有键 is_internal 的属性具有属性值 true 时发出令牌。将属性 is_internal 设置为 false 或未设置的用户不会获得令牌。

18.4. 配置身份提供程序

18.4.1. 将 Okta Identity Cloud 配置为 SAML 2.0 身份提供程序

您可以将 Okta 用作 Red Hat Advanced Cluster Security for Kubernetes (RHACS)的单点登录(SSO)供应商。

18.4.1.1. 创建一个 Okta 应用程序

在将 Okta 用作 Red Hat Advanced Cluster Security for Kubernetes 的 SAML 2.0 身份提供程序前，您必须创建一个 Okta app。

警告

Okta 的 Developer 控制台不支持创建自定义 SAML 2.0 应用程序。如果使用 Developer 控制台，您必须首先切换到 管理控制台 (Classic UI)。要切换，请单击页面左上角的 Developer Console 并选择 Classic UI。

先决条件

- 您必须拥有一个对 Okta 门户具有管理权限的帐户。

流程

1. 在 Okta 门户上，从菜单栏中选择 Applications。
2. 单击 Add Application，然后选择 Create New App。
3. 在 Create a New Application Integration对话框中，将 Web 保留为平台，然后选择 SAML 2.0 作为您要登录用户的协议。
4. 点 Create。
5. 在 General Settings 页面中，在 App name 字段中输入应用程序的名称。
6. 点击 Next。
7. 在 SAML Settings 页面中，为以下字段设置值：
    a. 单点登录 URL
        - 将它指定为 https://<RHACS_portal_hostname>/sso/providers/saml/acs。
        - 请选中 Use this for Recipient URL 和 Destination URL选项。
        - 如果您的 RHACS 门户可以通过不同的 URL 访问，您可以通过选中 Allow this application 来请求其他 SSO URL 选项，并使用指定格式添加替代 URL。

b. 受众 URI (SP 实体 ID)

- 将值设为 RHACS 或者您选择的另一个值。
- 请记住，在配置 Red Hat Advanced Cluster Security for Kubernetes 时，需要这个值。

c. 属性声明

- 您必须至少添加一个 attribute 语句。
- 红帽建议使用 email 属性：
  - Name: email
  - 格式： 未指定
  - 值 ： user.email

8. 在继续操作前，验证您是否至少配置了一个 Attribute 语句。
9. 点击 Next。
10. 在 Feedback 页面中，选择一个适用于您的选项。
11. 选择一个合适 的应用程序类型。
12. 点 Finish。

配置完成后，您将重定向到新应用的 Sign On 设置页面。黄色框包含配置 Red Hat Advanced Cluster Security for Kubernetes 所需的信息的链接。

创建应用程序后，将 Okta 用户分配给这个应用程序。进入 Assignments 选项卡，再分配可以访问 Red Hat Advanced Cluster Security for Kubernetes 的独立用户或组集合。例如，分配组 Everyone，以允许机构中的所有用户访问 Red Hat Advanced Cluster Security for Kubernetes。

18.4.1.2. 配置 SAML 2.0 身份提供程序

使用本节中的说明，将安全断言标记语言(SAML) 2.0 身份提供程序与 Red Hat Advanced Cluster Security for Kubernetes (RHACS)集成。

先决条件

- 您必须具有在 RHACS 中配置身份提供程序的权限。
- 对于 Okta 身份提供程序，您必须为 RHACS 配置 Okta 应用程序。

流程

1. 在 RHACS 门户中，进入 Platform Configuration → Access Control。
2. 点 Create auth provider，并从下拉列表中选择 SAML 2.0。
3. 在 Name 字段中输入用于标识此身份验证提供程序的名称；例如，Okta 或 Google。集成名称显示在登录页面上，以帮助用户选择正确的登录选项。

4. 在 ServiceProvider issuer 字段中输入您用作 Okta 中的Audience URI 或 SP Entity ID 的值，或者在其他供应商中输入类似的值。

5. 选择 配置 类型：

   - 选项 1：动态配置 ：如果您选择了这个选项，请输入 IdP 元数据 URL 或 身份提供程序控制台中提供的身份提供程序元数据的 URL。配置值从 URL 获取。

   - 选项 2：静态配置 ：从 Okta 控制台中的View Setup instructionss链接复制所需的静态字段，或者其它供应商的类似位置：

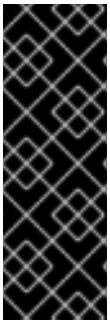
     - IdP Issuer
     - IdP SSO URL
     - 名称/ID 格式
     - IdP 证书(PEM)

6. 为使用 SAML 访问 RHACS 的用户分配最小访问角色。

   提示

   完成设置时，将最小访问角色设置为 Admin。之后，您可以返回 Access Control 页面，根据您的身份提供程序中的用户元数据设置更多定制的访问规则。

7. 点击 Save。



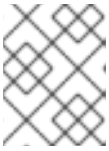
重要

如果您的 SAML 身份提供程序的身份验证响应满足以下条件：

- 包括一个 NotValidAfter assertion: 用户会话保持有效，直到NotValidAfter 字段中指定的时间已过。用户会话过期后，用户必须重新进行身份验证。
- 不包括 NotValidAfter assertion：用户会话在 30 天内保持有效，然后用户必须重新进行身份验证。

验证

1. 在 RHACS 门户中，进入 Platform Configuration → Access Control。
2. 选择 Auth Providers 选项卡。
3. 点击您要验证配置的身份验证供应商。
4. 从 Auth Provider 部分标头中选择 Test login。Test 登录页面 将在新的浏览器标签页中打开。
5. 使用您的凭证登录。
   - 如果您成功登录，RHACS 会显示用于登录到该系统的凭证的身份提供程序的用户 ID 和 用户属性。
   - 如果您的登录尝试失败，RHACS 会显示描述无法处理身份提供程序的响应信息。
6. 关闭 Test login 浏览器标签页。



注意

即使响应指示身份验证成功，您可能需要根据身份提供程序中的用户元数据创建额外的访问规则。

18.4.2. 将 Google Workspace 配置为 OIDC 身份提供程序

您可以使用 Google Workspace 作为 Red Hat Advanced Cluster Security for Kubernetes 的单点登录(SSO)供应商。

18.4.2.1. 为您的 GCP 项目设置 OAuth 2.0 凭证

要将 Google Workspace 配置为 Red Hat Advanced Cluster Security for Kubernetes 的身份供应商，您必须首先为 GCP 项目配置 OAuth 2.0 凭证。

先决条件

- 您必须具有对机构的 Google Workspace 帐户的管理员级别访问权限，才能创建新项目，或为现有项目创建和配置 OAuth 2.0 凭证的权限。红帽建议您创建一个新项目来管理对 Red Hat Advanced Cluster Security for Kubernetes 的访问。

流程

1. 创建新的 Google Cloud Platform (GCP)项目，请参阅 Google 文档主题 创建和管理项目。
2. 创建项目后，打开 Google API 控制台中的 Credentials 页面。
3. 验证左上角列出的项目名称，以确保您正在使用正确的项目。
4. 要创建新凭据，请转至 Create Credentials → OAuth 客户端 ID。
5. 选择 Web application 作为 Application type。
6. 在 Name 框中，输入应用程序的名称，如 RHACS。
7. 在 Authorized 重定向 URIs 框中，输入 **https://<stackrox_hostname>:<port_number>/sso/providers/oidc/callback**。
   - 将 <stackrox_hostname > 替换为您公开 Central 实例的主机名。
   - 将 <port_number > 替换为您公开 Central 的端口号。如果您使用标准 HTTPS 端口443，您可以省略端口号。
8. 点 Create。这会创建一个应用程序和凭证，并将您重新重定向到凭据页面。
9. 此时会打开一个信息框，显示新创建的应用程序的详细信息。关闭信息框。
10. 复制并保存以 .apps.googleusercontent.com 结尾的客户端 ID。您可以使用 Google API 控制台来检查此客户端 ID。
11. 从左侧的导航菜单中选择 OAuth consent 屏幕。

注意

OAuth consent 屏幕配置对整个 GCP 项目有效，而不仅仅是您在前面的步骤中创建的应用程序。如果您已经在这个项目中配置了 OAuth 同意屏幕，并希望为 Red Hat Advanced Cluster Security for Kubernetes 登录应用不同的设置，请创建一个新的 GCP 项目。

12. 在 OAuth consent 屏幕页面中：

    a. 选择 Application type 作为 Internal。如果您选择 Public，则具有 Google 帐户的任何人都可以登录。

    b. 输入描述性 应用程序名称。当用户登录后，此名称显示在同意屏幕上的用户。例如，使用 RHACS 或 <organization_name> SSO for Red Hat Advanced Cluster Security for Kubernetes。

    c. 验证 Google API 的范围是否 仅列出 电子邮件、配置集和 openid 范围。单点登录只需要这些范围。如果您授予其他范围，它会增加公开敏感数据的风险。

18.4.2.2. 指定客户端 secret

Red Hat Advanced Cluster Security for Kubernetes 版本 3.0.39 及更新版本在指定客户端 secret 时支持 OAuth 2.0 授权代码授予 身份验证流。当使用此身份验证流时，Red Hat Advanced Cluster Security for Kubernetes 使用刷新令牌来保持用户登录，超过 OIDC 身份提供程序中配置的令牌过期时间。

当用户登出时，Red Hat Advanced Cluster Security for Kubernetes 从客户端中删除刷新令牌。另外，如果您的身份提供程序 API 支持刷新令牌撤销，Red Hat Advanced Cluster Security for Kubernetes 也会向身份提供程序发送请求以撤销刷新令牌。

在将 Red Hat Advanced Cluster Security for Kubernetes 配置为与 OIDC 身份提供程序集成时，您可以指定客户端 secret。

注意

- 您不能使用带有 *Fragment* Callback mode 的 Client Secret。
- 您不能编辑现有身份验证供应商的配置。
- 如果要使用 客户端 Secret，则必须在 Red Hat Advanced Cluster Security for Kubernetes 中创建新的 OIDC 集成。

在将 Red Hat Advanced Cluster Security for Kubernetes 与 OIDC 身份提供程序连接时，红帽建议使用客户端 secret。如果您不想使用 Client Secret，您必须选择 Do not use Client Secret （不推荐）选项。

18.4.2.3. 配置 OIDC 身份提供程序

您可以将 Red Hat Advanced Cluster Security for Kubernetes (RHACS)配置为使用 OpenID Connect (OIDC)身份提供程序。

先决条件

- 您必须已在身份提供程序中配置了应用程序，如 Google Workspace。
- 您必须具有在 RHACS 中配置身份提供程序的权限。

流程

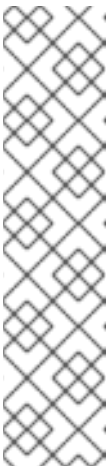
1. 在 RHACS 门户中，进入 Platform Configuration → Access Control。
2. 点 Create auth provider，并从下拉列表中选择 OpenID Connect。
3. 在以下字段中输入信息：
   - 名称 ：用于标识身份验证提供程序的名称，如 Google Workspace。集成名称显示在登录页面上，以帮助用户选择正确的登录选项。
   - 回调模式 ：选择 Auto-select （推荐），这是默认值，除非身份提供程序需要其他模式。

     注意

     片段 模式围绕单页应用程序(SPAs)的限制而设计。红帽只支持早期集成的 Fragment 模式，我们不推荐将其用于后续集成。

   - 签发者 ：身份提供程序的根 URL；例如，Google Workspace 的 https://accounts.google.com。如需更多信息，请参阅您的身份提供程序文档。

     注意

     如果您使用 RHACS 版本 3.0.49 及更新的版本，对于 Issuer，您可以执行以下操作：

     - 为您的 root URL 为 https+insecure:// 前缀，以跳过 TLS 验证。此配置不安全，我们不推荐这样做。仅将其用于测试目的。
     - 指定查询字符串；例如，?key1=value1&key2=value2 和 root URL。当您将其输入到授权端点时，RHACS 将 Issuer 的值附加到授权端点。您可以使用它来自定义供应商的登录屏幕。例如，您可以使用 hd 参数将 Google Workspace 登录屏幕优化到特定的托管域，或使用 pfidpadapterid 参数在 PingFederate 中预选一个验证方法。

   - 客户端 ID ：您配置的项目的 OIDC 客户端 ID。
   - Client Secret ：输入身份提供程序(IdP)提供的客户端 secret。如果您不使用客户端 secret （不推荐），请选择 Do not use Client Secret。
4. 为使用所选 身份提供程序访问 RHACS 的用户分配最小 访问角色。

   提示

   完成设置时，将最小访问角色设置为 Admin。之后，您可以返回 Access Control 页面，根据您的身份提供程序中的用户元数据设置更多定制的访问规则。

5. 要为访问 RHACS 的用户和组添加访问规则，请点击 Rules 部分中的 Add new rule。例如，要为名为 administrator 的用户提供 Admin 角色，您可以使用以下键值对创建访问规则：

键	值
名称	Administrator

角色	Admin

6. 点 Save。

验证

1. 在 RHACS 门户中，进入 Platform Configuration → Access Control。
2. 选择 Auth provider 选项卡。
3. 选择您要验证配置的身份验证供应商。
4. 从 Auth Provider 部分标头中选择 Test login。Test 登录页面 将在新的浏览器标签页中打开。
5. 使用您的凭证登录。
   - 如果您成功登录，RHACS 会显示用于登录到该系统的凭证的身份提供程序的用户 ID 和 用户属性。
   - 如果您的登录尝试失败，RHACS 会显示描述无法处理身份提供程序的响应信息。
6. 关闭 Test Login 浏览器标签页。

18.4.3. 将 OpenShift Container Platform OAuth 服务器配置为身份提供程序

OpenShift Container Platform 包括一个内置的 OAuth 服务器，可用作 Red Hat Advanced Cluster Security for Kubernetes (RHACS)的身份验证供应商。

18.4.3.1. 将 OpenShift Container Platform OAuth 服务器配置为身份提供程序

要将内置的 OpenShift Container Platform OAuth 服务器集成为 RHACS 的身份供应商，请使用本节中的说明。

先决条件

- 您必须具有 AuthProvider 权限，才能在 RHACS 中配置身份提供程序。
- 您必须已通过身份提供程序在 OpenShift Container Platform OAuth 服务器中配置了用户和组。有关身份提供程序要求的详情，请参阅 了解身份提供程序配置。

注意

以下流程只为 OpenShift Container Platform OAuth 服务器配置一个名为 central 的主路由。

流程

1. 在 RHACS 门户中，进入 Platform Configuration → Access Control。
2. 点 Create auth provider，然后从下拉列表中选择 OpenShift Auth。
3. 在 Name 字段中输入身份验证提供程序的名称。

4. 为使用所选 身份提供程序访问 RHACS 的用户分配最小 访问角色。用户必须具有授予此角色的权限或具有较高权限的角色才能登录到 RHACS。

   提示

   为安全起见，红帽建议在完成设置时首先将 最小访问角色设置为 None。之后，您可以返回 Access Control 页面，根据您的身份提供程序中的用户元数据设置更多定制的访问规则。

5. 可选： 要为用户和组访问 RHACS 添加访问规则，请点击 Rules 部分中的 Add new rule，然后输入规则信息并点 Save。您需要用户或组的属性，以便您可以配置访问权限。

   提示

   组映射功能更为强大，因为组通常与团队或权限集关联，且需要比用户少修改。

   要在 OpenShift Container Platform 中获取用户信息，您可以使用以下方法之一：

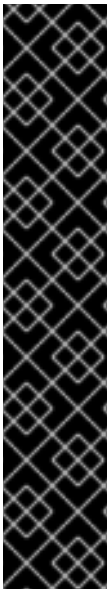
   - 点 User Management → Users → <username > → YAML。
   - 访问 k8s/cluster/user.openshift.io~v1~User/<username>/yaml 文件，并记录 名称、uid (RHACS 中的userid) 和组 的值。
   - 使用 *OpenShift Container Platform API 参考*中所述。

   以下配置示例描述了如何使用以下属性为 Admin 角色配置规则：

   - 名称 ：管理员
   - groups: ["system:authenticated", "system:authenticated:oauth", "myAdministratorsGroup"]
   - uid: 12345-00aa-1234-123b-123fcdef1234

   您可以使用以下方法之一为这个管理员角色添加规则：

   - 要为名称配置规则，请从 Key 下拉列表中选择 name，在 Value 字段中输入 administrator，然后在 Role 下选择 Administrator。
   - 要为组配置规则，请从 Key 下拉列表中选择 groups，在 Value 字段中输入 myAdministratorsGroup，然后在 Role 下选择 Admin。
   - 要为用户名配置规则，请从 Key 下拉列表中选择 userid，在 Value 字段中输入 12345-00aa-1234-123b-123fcdef1234，然后在 Role 下选择 Admin。



重要

- 如果将自定义 TLS 证书用于 OpenShift Container Platform OAuth 服务器，您必须将 CA 的 root 证书作为可信 root CA 添加到 Red Hat Advanced Cluster Security for Kubernetes 中。否则，Central 无法连接到 OpenShift Container Platform OAuth 服务器。
- 要使用 roxctl CLI 安装 Red Hat Advanced Cluster Security for Kubernetes 时启用 OpenShift Container Platform OAuth 服务器集成，请在 Central 中将 ROX_ENABLE_OPENSHIFT_AUTH 环境变量设置为 true ：

```
$ oc -n stackrox set env deploy/central
ROX_ENABLE_OPENSHIFT_AUTH=true
```

- 对于访问规则，OpenShift Container Platform OAuth 服务器不会返回密钥 电子邮件。

其他资源

- 配置 LDAP 身份提供程序
- 添加可信证书颁发机构

18.4.3.2. 为 OpenShift Container Platform OAuth 服务器创建额外路由

当使用 Red Hat Advanced Cluster Security for Kubernetes 门户将 OpenShift Container Platform OAuth 服务器配置为身份提供程序时，RHACS 仅为 OAuth 服务器配置单一路由。但是，您可以通过在 Central 自定义资源中将注解指定为注解来创建其他路由。

先决条件

- 您必须已将服务帐户配置为 OpenShift Container Platform OAuth 服务器的 OAuth 客户端。

流程

- 如果使用 RHACS Operator 安装 RHACS：
  1. 创建一个 CENTRAL_ADDITIONAL_ROUTES 环境变量，其中包含 Central 自定义资源的补丁：

```
$ CENTRAL_ADDITIONAL_ROUTES='
spec:
  central:
    exposure:
      loadBalancer:
        enabled: false
        port: 443
      nodePort:
        enabled: false
      route:
        enabled: true
    persistence:
      persistentVolumeClaim:
        claimName: stackrox-db
  customize:
```

```
    annotations:
      serviceaccounts.openshift.io/oauth-redirecturi.main:
sso/providers/openshift/callback (1)
      serviceaccounts.openshift.io/oauth-redirectreference.main: "
{\"kind\":\"OAuthRedirectReference\",\"apiVersion\":\"v1\",\"reference\":
{\"kind\":\"Route\",\"name\":\"central\"}}" (2)
      serviceaccounts.openshift.io/oauth-redirecturi.second:
sso/providers/openshift/callback (3)
      serviceaccounts.openshift.io/oauth-redirectreference.second: "
{\"kind\":\"OAuthRedirectReference\",\"apiVersion\":\"v1\",\"reference\":
{\"kind\":\"Route\",\"name\":\"second-central\"}}" (4)
'
```

(1) 用于设置主路由的重定向 URI。

(2) 主路由的重定向 URI 引用。

(3) 设置第二个路由的重定向。

(4) 第二个路由的重定向引用。

2. 将 CENTRAL_ADDITIONAL_ROUTES 补丁应用到 Central 自定义资源：

```
$ oc patch centrals.platform.stackrox.io \
  -n <namespace> \ (1)
  <custom-resource> \ (2)
  --patch "$CENTRAL_ADDITIONAL_ROUTES" \
  --type=merge
```

(1) 将 <namespace> 替换为包含 Central 自定义资源的项目的名称。

(2) 将 <custom-resource> 替换为 Central 自定义资源的名称。

- 或者，如果您使用 Helm 安装 RHACS：

1. 在 values-public.yaml 文件中添加以下注解：

```
customize:
  central:
    annotations:
      serviceaccounts.openshift.io/oauth-redirecturi.main:
sso/providers/openshift/callback (1)
      serviceaccounts.openshift.io/oauth-redirectreference.main: "
{\"kind\":\"OAuthRedirectReference\",\"apiVersion\":\"v1\",\"reference\":
{\"kind\":\"Route\",\"name\":\"central\"}}" (2)
      serviceaccounts.openshift.io/oauth-redirecturi.second:
sso/providers/openshift/callback (3)
      serviceaccounts.openshift.io/oauth-redirectreference.second: "
{\"kind\":\"OAuthRedirectReference\",\"apiVersion\":\"v1\",\"reference\":
{\"kind\":\"Route\",\"name\":\"second-central\"}}" (4)
```

(1) 设置主路由的重定向。

2 主路由的重定向引用。

3 设置第二个路由的重定向。

4 第二个路由的重定向引用。

2. 使用 helm upgrade 将自定义注解应用到 Central 自定义资源：

```
$ helm upgrade -n stackrox \
  stackrox-central-services rhacs/central-services \
  -f <path_to_values_public.yaml> 1
```

1 使用 -f 选项指定 values-public.yaml 配置文件的路径。

其他资源

- 服务帐户作为 OAuth 客户端
- 重定向作为 OAuth 客户端的服务帐户的 URI

18.4.4. 使用 SSO 配置将 Azure AD 连接到 RHACS

要使用 Sign-On (SSO)配置将 Azure Active Directory (AD)连接到 RHACS，您需要向令牌添加特定的声明（例如，组 声明到令牌），并将用户、组或两者都分配给企业级应用程序。

18.4.4.1. 使用 SSO 配置将组声明添加到 SAML 应用的令牌

在 Azure AD 中配置应用程序注册，以在令牌中包含 组 声明。具体步骤请参阅 使用 SSO 配置将组声明添加到 SAML 应用的 令牌中。

重要

验证您是否正在使用最新版本的 Azure AD。有关如何将 Azure AD 升级到最新版本的更多信息，请参阅 Azure AD Connect: 从上一版本升级到最新版本。

18.5. 配置短期访问

Red Hat Advanced Cluster Security for Kubernetes (RHACS)提供了配置对用户界面和 API 调用的短期访问的功能。

您可以通过为 RHACS 发布的令牌交换 OpenID Connect (OIDC)身份令牌进行配置。

我们建议使用它，特别是用于持续集成(CI)的使用，因为与长期 API 令牌相比，短期访问是首选使用。

下列步骤概述了如何配置对用户界面和 API 调用的短期访问的高级工作流：

1. 配置 RHACS 以信任 OIDC 身份提供程序签发者，以交换简短的 RHACS 发布的令牌。
2. 通过调用 API 为简短的 RHACS 发布的令牌交换 OIDC 身份令牌。

注意

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

其他资源

- 使用身份验证供应商与 roxctl 进行身份验证
- 配置 API 令牌

18.5.1. 为 OIDC 身份令牌签发者配置短期访问

开始为 OpenID Connect (OIDC)身份令牌签发者配置简短访问。

流程

1. 在 RHACS 门户中，进入 Platform Configuration → Integrations。
2. 滚动到 Authentication Tokens 类别，然后单击 Machine access configuration。
3. 点 Create configuration。
4. 选择 配置类型，选择以下之一：
   - 如果您使用任意 OIDC 身份令牌签发者，则 通用。
   - 如果您计划从 GitHub Actions 访问 RHACS，则 GitHub Actions。
5. 输入 OIDC 身份令牌签发者。
6. 为配置发布的令牌输入令牌 生命周期。

注意

令牌生命周期的格式为 XhYmZs，您无法将其设置为 24 小时。

7. 在配置中添加规则：
   - Key 是要使用的 OIDC 令牌声明。
   - Value 是预期的 OIDC 令牌声明值。
   - 如果存在 OIDC 令牌声明和值，则 Role 是分配给令牌的角色。

注意

规则类似于身份验证提供程序规则，以根据声明值分配角色。

作为常规规则，红帽建议在规则中使用唯一的不可变声明。常规建议是使用 OIDC 身份令牌中 的子 声明。有关 OIDC 令牌声明的更多信息，请参阅标准 OIDC 声明列表。

8. 点击 Save。

18.5.2. 交换身份令牌

先决条件

- 您有一个有效的 OpenID Connect (OIDC)令牌。
- 您为您要访问的 RHACS 实例添加了机器访问 配置。

流程

1. 准备 POST 请求的 JSON 数据：

```
{
  "idToken": "<id_token>"
}
```

2. 发送 POST 请求到 API /v1/auth/m2m/exchange。
3. 等待 API 响应：

```
{
  "accessToken": "<access_token>"
}
```

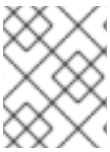
4. 使用返回的访问令牌访问 RHACS 实例。

注意

如果使用 GitHub Actions，您可以使用 stackrox/central-login GitHub Action。

第 19 章 使用系统健康仪表板

Red Hat Advanced Cluster Security for Kubernetes 系统健康仪表板提供了一个单一界面，用于查看有关 Red Hat Advanced Cluster Security for Kubernetes 组件的健康相关信息。



注意

系统健康仪表板仅适用于 Red Hat Advanced Cluster Security for Kubernetes 3.0.53 及更新版本。

19.1. 系统健康仪表板详情

访问健康仪表板：

- 在 RHACS 门户中，进入 Platform Configuration → System Health。

健康仪表板在以下组中组织信息：

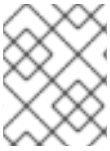
- Cluster Health - 显示 Red Hat Advanced Cluster Security for Kubernetes 集群的整体状态。
- 漏洞定义 - 显示漏洞定义的最后更新时间。
- Image Integrations - 显示您集成的所有 registry 的健康状况。
- 通知程序 集成 - 显示您集成的通知程序(Slack、电子邮件、JIRA 或其他类似的集成)的健康状况。
- 备份 集成 - 显示您集成的任何备份供应商的健康状况。

仪表板列出了不同组件的以下状态：

- 健康 - 组件可以正常工作。
- degraded - 组件不健康。这个状态意味着集群可以正常工作，但有些组件不健康，需要注意。
- unhealthy - 此组件处于健康状态，需要立即关注。
- Uninitialized - 组件尚未报告给 Central，以便其运行状况评估。未初始化状态有时可能需要注意，但通常会在几分钟后或集成被使用时报告返回健康状态。

集群健康部分
Cluster Overview 显示有关 Red Hat Advanced Cluster Security for Kubernetes 集群健康状况的信息。它报告以下健康信息：

- Collector Status - 它显示 Red Hat Advanced Cluster Security for Kubernetes 使用的 Collector pod 是否报告健康。
- Sensor Status - 它显示 Red Hat Advanced Cluster Security for Kubernetes 使用的 Sensor pod 是否报告健康。
- 传感器升级 - 它表明，与 Central 相比，传感器是否运行了正确的版本。
- 凭证过期 - 它显示 Red Hat Advanced Cluster Security for Kubernetes 的凭证是否接近过期。

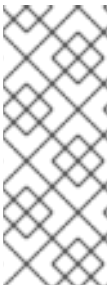


注意

处于 **Uninitialized** 状态的集群不会在 Red Hat Advanced Cluster Security for Kubernetes 保护的集群数量中报告，直到它们签入为止。

漏洞定义部分
Vulnerabilities Definition 部分显示更新最后的时间漏洞定义，以及定义是否是最新的。

integrations 部分
有 3 个集成部分 Image Integrations, Notifier Integrations, 和 Backup Integrations。与 Cluster Health 部分类似，这些部分列出了不健康的集成数量（如果存在）。否则，所有集成报告都处于健康状态。



注意

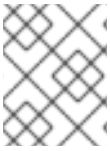
如果满足以下条件，则 Integrations 部分将健康集成列为 0。

- 您没有将 Red Hat Advanced Cluster Security for Kubernetes 与任何第三方工具集成。
- 您已与一些工具集成，但禁用了集成，或者尚未设置任何策略违反情况。

19.2. 查看产品使用数据

RHACS 根据从 RHACS 传感器收集的指标，为安全 Kubernetes 节点数量提供产品使用数据，以及安全集群的 CPU 单元。这些信息有助于估算 RHACS 消耗数据进行报告。

有关如何在 Kubernetes 中定义 CPU 单元的更多信息，请参阅 CPU 资源单元。



注意

OpenShift Container Platform 提供自己的使用情况报告；此信息用于自我管理的 Kubernetes 系统。

RHACS 在 web 门户和 API 中提供以下使用数据：

- 目前安全 CPU 单元： RHACS 安全集群使用的 Kubernetes CPU 单元数量，从最新的指标集合开始。
- 目前安全节点数： RHACS 保护的 Kubernetes 节点数量，从最新的指标集合开始。
- 最大安全 CPU 单元： RHACS 安全集群使用的最大 CPU 单元数，如每小时测量，并在 Start date 和 End date 定义的时间周期内聚合。
- 最大安全节点数：由 RHACS 保护的最大 Kubernetes 节点数量，如每小时，并在 Start date 和 End date 定义的时间周期内聚合。
- CPU 单元观察日期：收集最大安全 CPU 单元数据的日期。
- 节点数观察日期：收集最大安全节点数数据的日期。

传感器每 5 分钟收集一次数据，因此显示当前数据可能会有短暂的延迟。要查看历史数据，您必须配置 Start date 和 End date 并下载数据文件。日期范围包含，取决于您的时区。

提供的最大值会根据请求的周期的每小时最大值计算。每小时最大值可用于以 CSV 格式下载。

注意

显示的数据不会发送到红帽或显示为 Prometheus 指标。

流程

1. 在 RHACS 门户中，进入 Platform Configuration → System Health。
2. 单击 Show product usage。
3. 在 Start date 和 End date 字段中，选择您要显示数据的日期。这个范围包含，取决于您的时区。
4. 可选：要下载详细数据，请点击 Download CSV。

您还可以使用 **ProductUsageService** API 对象来获取此数据。如需更多信息，请参阅 RHACS 门户中的 Help → API 参考。

19.3. 使用 RHACS 门户生成诊断捆绑包

您可以使用 RHACS 门户中的系统健康仪表板生成诊断捆绑包。

先决条件

- 要生成诊断捆绑包，您需要 **DebugLogs** 资源的 **read** 权限。

流程

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

19.3.1. 其他资源

- 生成诊断捆绑包

第 20 章 使用管理事件页面

您可以在带有 Red Hat Advanced Cluster Security for Kubernetes (RHACS)的单一接口中查看管理事件信息。您可以使用此界面帮助您理解和解释重要事件详情。

20.1. 访问不同域中的事件日志

通过查看管理事件页面，您可以访问不同域中的各种事件日志。

流程

- 在 RHACS 平台中，进入 Platform Configuration → Administration Events。

20.2. 管理事件页面概述

管理事件页面在以下组中组织信息：

- 域 ：根据发生事件的 RHACS 中的特定区域或域来划分事件。此分类有助于组织并了解事件的上下文。
  以下域包括：
  - 身份验证
  - General
  - 镜像扫描
  - 集成
- 资源类型 ：根据涉及的资源或组件类型分类事件。
  包括以下资源类型：
  - API 令牌
  - 集群
  - 镜像
  - 节点
  - 通知程序
- 级别 ：指示事件的严重性或重要程度。
  包括以下级别：
  - 错误
  - Warning
  - 成功
  - info
  - Unknown

- 最后发生的事件：提供 发生事件时的时间戳和日期的信息。它有助于跟踪事件的时间，这对于诊断问题并了解操作或事件序列至关重要。
- 数量 ：指示特定事件发生的次数。这个数字可用于评估问题的频率。多次发生的事件表示您需要修复的持久性问题。

每个事件还为您提供修复错误所需的操作。

20.3. 获取有关特定域中事件的信息

通过查看管理事件的详情，您可以获取有关该特定域中事件的更多信息。这可让您更好地了解事件的上下文和详情。

流程

- 在 Administration Events 页面中，单击域以查看其详细信息。

20.4. 管理事件详情概述

管理事件提供描述错误或事件的日志信息。

日志提供以下信息：

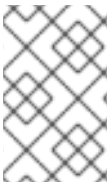
- 事件上下文
- 修复错误的步骤

管理事件页面在以下组中组织信息：

- 资源类型 ：根据涉及的资源或组件类型分类事件。
  包括以下资源类型：
  - API 令牌
  - 集群
  - 镜像
  - 节点
  - 通知程序
- 资源名称 ：指定事件引用的资源或组件的名称。它标识发生事件的域中的特定实例。
- Event type ：指定事件源。Central 生成与从日志语句中创建的管理事件对应的日志事件。
- 事件 ID ：由分配给每个事件的字母数字字符组成的唯一标识符。事件 ID 在识别、跟踪和管理一段时间内的事件 ID 非常有用。
- 创建于 ：指示事件最初创建或记录的时间戳和日期。
- 最后发生于 ：指定事件最后一次发生的时间戳和日期。这会跟踪事件的时间，这对于诊断和修复重复问题至关重要。
- 数量 ：指示特定事件发生的次数。这个数字可用于评估问题的频率。多次发生的事件表示您需要修复的持久性问题。

20.5. 设置管理事件的过期

通过指定天数，您可以控制管理事件何时过期。这对管理事件非常重要，并确保保留了所需期间的信息。



注意

默认情况下，管理事件会保留 4 天。这些事件的保留周期由最后一次发生的时间决定，而不是由创建的时间决定。这意味着，只有在最后一次发生的时间超过指定的保留周期时，才会删除事件。

流程

1. 在 RHACS 门户中，进入 Platform Configuration → System Configuration。您可以为管理事件配置以下设置：
   - 管理事件保留天数 ：保留管理事件的天数。
2. 若要更改此值，请单击 Edit，进行更改，然后单击保存。