Red Hat Advanced Cluster Security for Kubernetes 4.4

升级

升级 Red Hat Advanced Cluster Security for Kubernetes

Last Updated: 2024-06-28

Red Hat Advanced Cluster Security for Kubernetes 4.4 升级

升级 Red Hat Advanced Cluster Security for Kubernetes

法律通告

摘要

本节介绍了使用 Helm chart 或 roxctl 命令行界面升级 Red Hat Advanced Cluster Security for Kubernetes。

目录

第 1 章 使用 OPERATOR 升级

根据安装时选择的 **Update approval** 选项，通过 Red Hat Advanced Cluster Security for Kubernetes (RHACS) Operator 自动执行或手动升级。

升级时遵循以下准则：

- 如果 Central 的版本早于 3.74，则必须在升级到 4.x 版本前升级到 3.74。有关将 Central 升级到 3.74 版本，请参阅升级文档 3.74。
- 当从 3.74 升级基于 Operator 的 Central 部署时，首先确保 Operator 升级模式设置为 **Manual**。然后，按照版本 4.0 的升级文档中的步骤将 Operator 升级到 4.0 版本，并确保 Central 在线。升级到 4.0 版本后，红帽建议将 Central 升级到最新版本以获得完整功能。

1.1. 准备升级

在升级 Red Hat Advanced Cluster Security for Kubernetes (RHACS)版本前，您必须执行以下步骤：

- 如果您要从 3.74 版本升级，请验证您是否正在运行 RHACS Operator 3.74 的最新补丁版本。
- 备份现有的 Central 数据库。
- 如果要升级的集群包含 **SecuredCluster** 自定义资源(CR)，请将集合方法改为 EBPF 或 CORE_BPF。如需更多信息，请参阅"更改集合方法"。

1.1.1. 更改集合方法

如果要升级的集群包含 **SecuredCluster** CR，您必须确保在升级前，每个节点集合设置被设置为 **CORE_BPF**，如果从 4.1 或更高版本升级。否则，将集合方法设置为 **EBPF**。要将集合方法设置为 **EBPF**，您必须在升级后将 **forceCollection** 参数设置为 **true**，并确保集合方法为 **EBPF**。

流程

1. 在 OpenShift Container Platform web 控制台中进入 RHACS Operator 页面。
2. 在顶部导航菜单中，选择 **Secured Cluster**。
3. 点实例名称，如 **stackrox-secured-cluster-services**。
4. 使用以下方法之一更改设置：
   - 在 **Form view** 中，在 **Per Node Settings → Collector Settings → Collection** 下，选择 **CORE_BPF**。
   - 点 **YAML** 打开 YAML 编辑器，并找到 **spec.perNode.collector.collection** 属性。如果值为 **KernelModule**，请将其改为 **CORE_BPF**。

注意

只有在从 4.1 之前的版本升级，或者有特定原因来使用它时，才使用 **EBPF**。

5. 点 **Save**。

1.1.2. 设置 forceCollection 参数

当升级安全集群时，如果您将集合方法设置为 **EBPF**，则必须在升级后将 **forceCollection** 参数设置为 **true**。然后，确保在 YAML 编辑器中将 **spec.perNode.collector.collection** 仍然设置为 **EBPF**。

流程

1. 在 OpenShift Container Platform web 控制台中进入 RHACS Operator 页面。
2. 在顶部导航菜单中，选择 **Secured Cluster**。
3. 点实例名称，如 **stackrox-secured-cluster-services**。
4. 点 **YAML** 打开 YAML 编辑器。
5. 找到 **spec.perNode.collector.forceCollection** 参数，并将其设置为 **true**。
6. 点 **Save**。

其他资源

- 更新安装的 Operator
- 备份 Red Hat Advanced Cluster Security for Kubernetes

1.2. 修改 CENTRAL 自定义资源

Central DB 服务需要持久性存储。如果您还没有为作为 SSD 或高性能的 Central 集群配置默认存储类，您必须更新 **Central** 自定义资源来为 Central DB 持久性卷声明 (PVC) 配置存储类。

注意

如果您已经为 Central 配置了默认存储类，请跳过此部分。

流程

- 使用以下配置更新中央自定义资源：

```
spec:
  central:
    db:
      isEnabled: Default (1)
      persistence:
        persistentVolumeClaim: (2)
          claimName: central-db
          size: 100Gi
          storageClassName: <storage-class-name>
```

1. 您不能将 **IsEnabled** 的值改为 **Enabled**。
2. 如果存在此声明，您的集群会使用现有的声明，否则它会创建一个新的声明。

1.3. 为外部数据库修改 CENTRAL 自定义资源

先决条件

- 您必须在数据库实例中有一个支持 PostgreSQL 13 和具有以下权限的用户的数据库：
  - 对数据库的连接权利。
  - schema 的 **Usage** 和 **Create**。
  - 对 schema 中的所有表的 **Select**, **Insert**, **Update**, 和 **Delete** 权限。
  - 对 schema 中所有序列的 **Usage**。

流程

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

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

   1. 如果使用 Kubernetes，请输入 **kubectl** 而不是 **oc**。
   2. 使用纯文本密码的文件的路径替换 **password.txt**。

2. 进入 OpenShift Container Platform Web 控制台中的 Red Hat Advanced Cluster Security for Kubernetes operator 页面。在顶部导航栏中选择 **Central**，再选择您要连接到数据库的实例。
3. 进入 **YAML 编辑器**视图。
4. 对于 **db.passwordSecret.name**，请指定您在前面的步骤中创建的引用的 secret。例如，**external-db-password**。
5. 对于 **db.connectionString**，使用 **keyword=value** 格式指定连接字符串，例如 host=< **host> port=5432 database=stackrox user=stackrox sslmode=verify-ca**
6. 对于 **db.persistence**，请删除整个块。
7. 如果需要，您可以通过在顶层 spec 中添加 TLS 块来为 Central 指定证书颁发机构来信任数据库证书，如下例所示：
   - 使用以下配置更新中央自定义资源：

```
spec:
  tls:
    additionalCAs:
    - name: db-ca
      content: |
        <certificate>
  central:
    db:
      isEnabled: Default 1
```

```
connectionString: "host=<host> port=5432 user=<user> sslmode=verify-ca"
passwordSecret:
  name: external-db-password
```

1 您不能将 **IsEnabled** 的值改为 **Enabled**。

8. 点击 **Save**。

其他资源

- 在 PostgreSQL 实例中置备数据库

1.4. 更改订阅频道

您可以使用 OpenShift Container Platform Web 控制台或使用命令行更改 RHACS Operator 的更新频道。要从 RHACS 3.74 升级到 RHACS 4.0，您必须更改更新频道。

重要

您必须为安装 RHACS Operator 的所有集群更改订阅频道，包括 Central 和所有安全集群。

先决条件

- 您必须验证您是否正在使用最新的 RHACS 3.74 Operator，且没有待处理的手动 Operator 升级。
- 您必须验证您是否已备份了现有的 Central 数据库。
- 您可以使用具有 **cluster-admin** 权限的账户访问 OpenShift Container Platform 集群 Web 控制台。

使用 Web 控制台更改订阅频道

使用以下 Web 控制台更改订阅频道的说明：

流程

1. 在 OpenShift Container Platform Web 控制台的 **Administrator** 视角中，进入 **Operators → Installed Operators**。
2. 找到 RHACS Operator 并点它。
3. 点 **Subscription** 标签页。
4. 点 **Update Channel** 下的更新频道的名称。
5. 选择 **stable**，然后点 **Save**。
6. 对于带有 **自动批准策略** 的订阅，更新会自动开始。返回到 **Operators → Installed Operators** 页面，以监控更新的进度。完成后，状态会变为 **Succeeded** 和 **Up to date**。
   对于采用**手动**批准策略的订阅，您可以从 **Subscription** 选项卡中手动批准更新。

使用命令行更改订阅频道

使用命令行更改订阅频道的说明：

流程

- 运行以下命令，将订阅频道改为 **stable** ：

```
$ oc -n rhacs-operator \ 1
  patch subscriptions.operators.coreos.com rhacs-operator \
  --type=merge --patch='{ "spec": { "channel": "stable" }}'
```

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

在更新 RHACS Operator 过程中，会置备一个名为 **central-db** 的新部署，数据开始迁移。它大约需要 30 分钟，且仅在升级时进行一次。

1.5. 升级到 4.1 及之后的版本后，删除 CENTRAL 附加 PV

Kubernetes 和 OpenShift Container Platform 不自动删除持久性卷(PV)。当您从早期版本升级 RHACS 时，名为 **stackrox-db** 的 Central PV 会保留挂载。但是，在 RHACS 4.1 中，Central 不再需要之前附加的 PV。

PV 具有之前 RHACS 版本使用的数据和持久性文件。您可以使用 PV 在 RHACS 4.1 之前回滚到更早的版本。或者，如果您有一个用于 Central 的大型 RocksDB 备份捆绑包，您可以使用 PV 恢复这些数据。

完成升级到 4.1 后，您可以删除 Central 附加的持久性卷声明(PVC)来释放存储。仅当没有计划从之前的 RocksDB 备份回滚或恢复时，才删除 PVC。

警告

删除 PVC 后，您无法在 RHACS 4.1 之前将 Central 回滚到早期版本，或恢复使用 RocksDB 创建的大型 RocksDB 备份。

1.5.1. 对于 RHACS 版本 4.1 及之后的版本，使用 RHACS Operator 删除 Central 附加 PV

删除 Central 附加持久性卷声明(PVC) **stackrox-db** 以释放存储空间。

流程

- 在 Central 中添加以下注解：

```
annotations:
  platform.stackrox.io/obsolete-central-pvc: "true"
```

验证

- 运行以下命令：

```
$ oc -n stackrox describe pvc stackrox-db | grep -i 'Used By'
Used By: <none> 1
```

1 等待您看到过的 By: **<none>**。它可能需要几分钟时间。

1.6. 回滚 OPERATOR 升级

要回滚 Operator 升级，您必须执行以下部分中描述的步骤。您可以使用 CLI 或 OpenShift Container Platform Web 控制台回滚 Operator 升级。

注意

如果您要从 RHACS 4.0 回滚，则只能回滚到 RHACS 3.74 的最新补丁版本。

1.6.1. 使用 CLI 回滚 Operator 升级

您可以使用 CLI 命令回滚 Operator 版本。

流程

1. 运行以下命令来删除 OLM 订阅：
   - 对于 OpenShift Container Platform，运行以下命令：

     ```
     $ oc -n rhacs-operator delete subscription rhacs-operator
     ```

   - 对于 Kubernetes，运行以下命令：

     ```
     $ kubectl -n rhacs-operator delete subscription rhacs-operator
     ```

2. 运行以下命令来删除集群服务版本 (CSV)：
   - 对于 OpenShift Container Platform，运行以下命令：

     ```
     $ oc -n rhacs-operator delete csv -l operators.coreos.com/rhacs-operator.rhacs-operator
     ```

   - 对于 Kubernetes，运行以下命令：

     ```
     $ kubectl -n rhacs-operator delete csv -l operators.coreos.com/rhacs-operator.rhacs-operator
     ```

3. 通过选择以下选项之一来确定您要回滚到的早期版本：
   - 如果当前 Central 实例正在运行，请运行以下命令查询 RHACS API 以获取回滚版本：

     ```
     $ curl -k -s -u <user>:<password> https://<central hostname>/v1/centralhealth/upgradestatus | jq -r .upgradeStatus.forceRollbackTo
     ```

   - 如果当前 Central 实例没有运行，请执行以下步骤：

     注意

     这个过程只能在安装 **rocksdb** 数据库时用于 RHACS 版本 3.74 及更早版本。

a. 运行以下命令，确保 Central 部署已缩减：

- 对于 OpenShift Container Platform，运行以下命令：

```
$ oc scale -n <central namespace> –replicas=0 deploy/central
```

- 对于 Kubernetes，运行以下命令：

```
$ kubectl scale -n <central namespace> –replicas=0 deploy/central
```

b. 将以下 pod 规格保存为 YAML 文件：

```
apiVersion: v1
kind: Pod
metadata:
  name: get-previous-db-version
spec:
  containers:
  - name: get-previous-db-version
    image: registry.redhat.io/advanced-cluster-security/rhacs-main-rhel8:<rollback
version>
    command:
    - sh
    args:
    - '-c'
    - "cat /var/lib/stackrox/.previous/migration_version.yaml | grep '^image:' | cut -f 2 -d
: | tr -d ' '"
    volumeMounts:
    - name: stackrox-db
      mountPath: /var/lib/stackrox
  volumes:
  - name: stackrox-db
    persistentVolumeClaim:
      claimName: stackrox-db
```

c. 运行以下命令，使用您保存的 YAML 文件在 Central 命名空间中创建 pod：

- 对于 OpenShift Container Platform，运行以下命令：

```
$ oc create -n <central namespace> -f pod.yaml
```

- 对于 Kubernetes，运行以下命令：

```
$ kubectl create -n <central namespace> -f pod.yaml
```

d. 创建 pod 后，运行以下命令来获取版本：

- 对于 OpenShift Container Platform，运行以下命令：

```
$ oc logs -n <central namespace> get-previous-db-version
```

- 对于 Kubernetes，运行以下命令：

```
$ kubectl logs -n <central namespace> get-previous-db-version
```

4. 运行以下命令，编辑 **central-config.yaml ConfigMap** 以设置 **maintenance.forceRollBackVersion:<version>** 参数：

   - 对于 OpenShift Container Platform，运行以下命令：

     ```
     $ oc get configmap -n <central namespace> central-config -o yaml | sed -e "s/forceRollbackVersion: none/forceRollbackVersion: <version>/" | oc -n <central namespace> apply -f -
     ```

   - 对于 Kubernetes，运行以下命令：

     ```
     $ kubectl get configmap -n <central namespace> central-config -o yaml | sed -e "s/forceRollbackVersion: none/forceRollbackVersion: <version>/" | kubectl -n <central namespace> apply -f -
     ```

5. 使用 Step 3 中显示的版本字符串为 Central 部署设置镜像，作为镜像标签。例如，运行以下命令：

   - 对于 OpenShift Container Platform，运行以下命令：

     ```
     $ oc set image -n <central namespace> deploy/central central=registry.redhat.io/advanced-cluster-security/rhacs-main-rhel8:<version>
     ```

   - 对于 Kubernetes，运行以下命令：

     ```
     $ kubectl set image -n <central namespace> deploy/central central=registry.redhat.io/advanced-cluster-security/rhacs-main-rhel8:<version>
     ```

验证

1. 确保 Central pod 启动并处于 **ready** 状态。如果 pod 崩溃，请检查日志以查看备份是否已恢复。一个成功的日志消息类似以下示例：

   ```
   Clone to Migrate ".previous", ""
   ```

2. 在回滚频道上重新安装 Operator。例如，**3.74.2** 安装在 **rhacs-3.74** 频道中。

1.6.2. 使用 Web 控制台回滚 Operator 升级

您可以使用 OpenShift Container Platform Web 控制台回滚 Operator 版本。

先决条件

- 您可以使用具有 **cluster-admin** 权限的账户访问 OpenShift Container Platform 集群 Web 控制台。

流程

1. 进入 **Operators → Installed Operators** 页面。
2. 找到 RHACS Operator 并点它。
3. 在 **Operator Details** 页面中，从 **Actions** 列表中选择 **Uninstall Operator**。按照此操作，Operator 将停止运行，不再接收更新。

4. 通过选择以下选项之一来确定您要回滚到的早期版本：

   - 如果当前 Central 实例正在运行，您可以通过从终端窗口中运行以下命令来查询 RHACS API 来获取回滚版本：

     ```
     $ curl -k -s -u <user>:<password> https://<central
     hostname>/v1/centralhealth/upgradestatus | jq -r .upgradeStatus.forceRollbackTo
     ```

   - 您可以通过执行以下步骤来创建 pod 并提取之前的版本：

     注意

     这个过程只能在安装 **rocksdb** 数据库时用于 RHACS 版本 3.74 及更早版本。

     a. 进入 **Workloads → Deployments → central**。

     b. 在 **Deployment details** 下，点 pod 数旁边的向下箭头，以缩减 pod。

     c. 进入 **Workloads → Pods → Create Pod**，将 pod 规格的内容粘贴到编辑器中：

     ```
     apiVersion: v1
     kind: Pod
     metadata:
       name: get-previous-db-version
     spec:
       containers:
       - name: get-previous-db-version
         image: registry.redhat.io/advanced-cluster-security/rhacs-main-rhel8:<rollback
     version>
         command:
         - sh
         args:
         - '-c'
         - "cat /var/lib/stackrox/.previous/migration_version.yaml | grep '^image:' | cut -f 2 -d
     : | tr -d ' '"
         volumeMounts:
         - name: stackrox-db
           mountPath: /var/lib/stackrox
       volumes:
       - name: stackrox-db
         persistentVolumeClaim:
           claimName: stackrox-db
     ```

     d. 点 **Create**。

     e. 创建 pod 后，点 **Logs** 选项卡来获取版本字符串。

5. 通过执行以下步骤更新回滚配置：

   a. 进入 **Workloads → ConfigMaps → central-config**，然后从 **Actions** 列表中选择 **Edit ConfigMap**。

   b. 在 **central-config.yaml** 键的值中找到 **forceRollbackVersion** 行。

   c. 将 **none** 替换为 **3.73.3**，然后保存文件。

6. 通过执行以下步骤将 Central 更新至早期版本：

   a. 进入 **Workloads → Deployments → central**，然后从 Actions 列表中选择 **Edit Deployment**。

   b. 更新镜像名称，然后保存更改。

验证

1. 确保 Central pod 启动并处于 **ready** 状态。如果 pod 崩溃，请检查日志以查看备份是否已恢复。一个成功的日志消息类似以下示例：

```
Clone to Migrate ".previous", ""
```

2. 在回滚频道上重新安装 Operator。例如，**3.74.2** 安装在 **rhacs-3.74** 频道中。

其他资源

- 使用 Operator 方法安装 Central
- Operator Lifecycle Manager 工作流
- 手动批准待处理的 Operator 更新

1.7. OPERATOR 升级问题故障排除

按照以下步骤调查并解决 RHACS Operator 的与升级相关的问题。

1.7.1. 无法调度 Central 数据库

在升级过程中，按照以下说明对 Central DB pod 进行故障排除：

1. 检查 **central-db** pod 的状态：

```
$ oc -n <namespace> get pod -l app=central-db (1)
```

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

2. 如果 pod 的状态为 **Pending**，请使用 describe 命令获取更多详细信息：

```
$ oc -n <namespace> describe po/<central-db-pod-name> (1)
```

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

3. 您可能会看到 **FailedScheduling** 警告信息：

```
Type     Reason            Age   From               Message
----     ------            ----  ----               -------
Warning  FailedScheduling  54s   default-scheduler  0/7 nodes are available: 1 Insufficient memory, 3 node(s) had untolerated taint {node-role.kubernetes.io/master: }, 4 Insufficient cpu. preemption: 0/7 nodes are available: 3 Preemption is not helpful for scheduling, 4 No preemption victims found for incoming pod.
```

4. 此警告信息建议调度的节点没有足够的内存来满足 pod 的资源要求。如果您有一个小的环境，请考虑在节点上增加资源，或添加一个可以支持数据库的更大的节点。
   否则，请考虑在 **central → db → resources** 下的自定义资源中减少 **central-db** pod 的资源要求。但是，运行比推荐最小值少的资源的中心可能会导致 RHACS 的性能降低。

1.7.2. Central 或 Secured 集群无法部署

当 RHACS Operator 时：

- 无法部署 Central 或 Secured 集群。
- 将 CR 更改应用到实际资源失败。

您必须检查自定义资源条件以查找问题。

- 对于 Central，运行以下命令检查条件：

```
$ oc -n rhacs-operator describe centrals.platform.stackrox.io
```
(1)

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

- 对于安全集群，运行以下命令检查条件：

```
$ oc -n rhacs-operator describe securedclusters.platform.stackrox.io
```
(1)

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

您可以识别条件输出中的配置错误：

输出示例

```
 Conditions:
    Last Transition Time:  2023-04-19T10:49:57Z
    Status:                False
    Type:                  Deployed
    Last Transition Time:  2023-04-19T10:49:57Z
    Status:                True
    Type:                  Initialized
    Last Transition Time:  2023-04-19T10:59:10Z
    Message:               Deployment.apps "central" is invalid:
spec.template.spec.containers[0].resources.requests: Invalid value: "50": must be less than or equal
to cpu limit
    Reason:                ReconcileError
    Status:                True
    Type:                  Irreconcilable
    Last Transition Time:  2023-04-19T10:49:57Z
    Message:               No proxy configuration is desired
    Reason:                NoProxyConfig
    Status:                False
    Type:                  ProxyConfigFailed
    Last Transition Time:  2023-04-19T10:49:57Z
    Message:               Deployment.apps "central" is invalid:
spec.template.spec.containers[0].resources.requests: Invalid value: "50": must be less than or equal
```

```
to cpu limit
    Reason:              InstallError
    Status:              True
    Type:                ReleaseFailed
```

另外，您可以查看 RHACS pod 日志以查找有关此问题的更多信息。运行以下命令来查看日志：

```
oc -n rhacs-operator logs deploy/rhacs-operator-controller-manager manager
```
(1)

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

第 2 章 使用 HELM CHART 升级

您必须根据您正在运行的 RHACS 发行版本遵循 RHACS 的特定升级路径。在更新 Helm chart 并更正升级前，还必须备份您的 Central 数据库。

2.1. 从 RHACS 版本 3.74 及更早版本升级序列

从早期版本升级时，请遵循以下指导：

- 如果 Central 的发行版本早于 3.74，则必须在升级到 4.x 版本前升级到最新的 3.74 补丁。有关从早期版本升级到 3.74 的信息，请参阅版本 3.74 的升级文档。
- 当从版本 3.74 升级基于 Helm 的安装时，您可以通过 4.4 升级到 RHACS 版本 4.0 的任何最新版本。但是，为了获得完整的功能，请升级到 4.4 版本。

如果您使用 Helm chart 安装 RHACS，升级到 RHACS 的最新版本，请执行以下步骤：

1. 备份 Central 数据库。
2. （可选）优化 Central 的数据库和持久性卷声明(PVC)。
3. 另外，还可为 central-services Helm Chart 生成包含 root 证书的 **values-private.yaml** 配置文件。
4. 更新 Helm Chart。
5. 运行 **helm upgrade** 命令。

重要

为确保最佳功能，您的 secure-cluster-services Helm Chart 和 central-services Helm Chart 需要使用相同的版本。

2.2. 备份 CENTRAL 数据库

在基础架构灾难的情况下，您可以备份中心数据库，并使用该备份从失败的升级或数据恢复中回滚。

先决条件

- 您必须具有一个 API 令牌，其具有对 Red Hat Advanced Cluster Security for Kubernetes 的所有资源的 **read** 权限。**analysts** 系统角色具有所有资源的 **read** 权限。
- 已安装了 **roxctl** CLI。
- 您已配置了 **ROX_API_TOKEN** 和 **ROX_CENTRAL_ADDRESS** 环境变量。

流程

- 运行备份命令：

```
$ roxctl -e "$ROX_CENTRAL_ADDRESS" central backup
```

其他资源

- 使用 roxctl CLI 进行按需备份
- 安装 roxctl CLI

2.3. 优化 CENTRAL 数据库和 PVC

当您升级到 Red Hat Advanced Cluster Security for Kubernetes (RHACS) 4.0 时，RHACS 会创建一个名为 **central-db** 的 PostgreSQL 实例，并带有默认持久性卷声明(PVC)。另外，您可以自定义 **central-db** 或 PVC 配置。

红帽建议以下最小内存和 CPU 请求：

```
central:
  db:
    resources:
      requests:
        memory: 16Gi
        cpu: 8
      limits:
        memory: 16Gi
        cpu: 8
```

2.4. 生成 ROOT 证书文件

如果您无法访问用于安装 Red Hat Advanced Cluster Security for Kubernetes (RHACS)的 **values-private.yaml** 配置文件，请使用以下指令来生成包含 root 证书的 **values-private.yaml** 配置文件。

如果可以访问 **values-private.yaml** 配置文件，请在此处跳过该指令。

重要

生成的 **values-private.yaml** 文件具有敏感配置选项。确保您安全地存储这个文件。

流程

1. 下载 **create_certificate_values_file.sh** 脚本。
2. 使 **create_certificate_values_file.sh** 脚本可执行：

```
$ chmod +x create_certificate_values_file.sh
```

3. 运行 **create_certificate_values_file.sh** 脚本文件：

```
$ create_certificate_values_file.sh values-private.yaml
```

2.5. 更新 HELM CHART 仓库

在升级到 Red Hat Advanced Cluster Security for Kubernetes 的新版本前，您必须始终更新 Helm chart。

先决条件

- 您必须已经添加了 Red Hat Advanced Cluster Security for Kubernetes Helm Chart 仓库。

- 您必须使用 Helm 版本 3.8.3 或更新版本。

流程

- 更新 Red Hat Advanced Cluster Security for Kubernetes chart 软件仓库。

```
$ helm repo update
```

验证

- 运行以下命令来验证添加的 chart 存储库：

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

2.6. 其他资源

- 使用 Helm chart 安装 Central
- 使用 Helm chart 在安全集群中安装 RHACS

2.7. 运行 HELM 升级命令

您可以使用 **helm upgrade** 命令更新 Red Hat Advanced Cluster Security for Kubernetes (RHACS)。

先决条件

- 您必须有权访问用于安装 Red Hat Advanced Cluster Security for Kubernetes (RHACS)的 **values-private.yaml** 配置文件。否则，您必须先生成包含 root 证书的 **values-private.yaml** 配置文件，然后才能继续这些命令。

流程

- 运行 helm upgrade 命令并使用 **-f** 选项指定配置文件：

```
$ helm upgrade -n stackrox stackrox-central-services \
  rhacs/central-services --version <current-rhacs-version> \ 1
  -f values-private.yaml \
  --set central.db.password.generate=true \
  --set central.db.serviceTLS.generate=true \
  --set central.db.persistence.persistentVolumeClaim.createClaim=true
```

注意

您可以使用 **--reuse-values** 选项在升级过程中保留之前配置的 Helm 值。如果这样做，在升级到下一个版本前，您必须关闭 **central-db** 创建。请参阅以下命令示例：

```
$ helm upgrade -n stackrox stackrox-central-services \
  rhacs/central-services --version <current-rhacs-version> --reuse-values \
  -f values-private.yaml \
  --set central.db.password.generate=false \
  --set central.db.serviceTLS.generate=false \
  --set central.db.persistence.persistentVolumeClaim.createClaim=false
```

2.8. 升级到 4.1 及之后的版本后，删除 CENTRAL 附加 PV

Kubernetes 和 OpenShift Container Platform 不自动删除持久性卷(PV)。当您从早期版本升级 RHACS 时，名为 **stackrox-db** 的 Central PV 会保留挂载。但是，在 RHACS 4.1 中，Central 不再需要之前附加的 PV。

PV 具有之前 RHACS 版本使用的数据和持久性文件。您可以使用 PV 在 RHACS 4.1 之前回滚到更早的版本。或者，如果您有一个用于 Central 的大型 RocksDB 备份捆绑包，您可以使用 PV 恢复这些数据。

完成升级到 4.1 后，您可以删除 Central 附加的持久性卷声明(PVC)来释放存储。仅当没有计划从之前的 RocksDB 备份回滚或恢复时，才删除 PVC。

警告

删除 PVC 后，您无法在 RHACS 4.1 之前将 Central 回滚到早期版本，或恢复使用 RocksDB 创建的大型 RocksDB 备份。

2.8.1. 使用 Helm 删除中央附加 PV

删除 Central 附加持久性卷声明(PVC) **stackrox-db** 以释放存储空间。

流程

- 运行以下命令：

```
$ helm upgrade -n stackrox stackrox-central-services \
    rhacs/central-services --version <current-rhacs-version> \
    --set central.persistence.none=true
```

验证

- 运行以下命令：

```
$ oc -n stackrox describe pvc stackrox-db | grep -i 'Used By'
Used By: <none> 1
```

1 1 等待您看到过的 By: **<none>**。它可能需要几分钟时间。

2.9. 回滚 HELM 升级

如果升级到新版本失败，您可以回滚到以前版本的 Central。

流程

1. 运行以下 **helm upgrade** 命令：

```
$ helm upgrade -n stackrox \
  stackrox-central-services rhacs/central-services \
  --version <previous_rhacs_74_version> \ 1
  --set central.db.enabled=false
```

1 将 **<previous_rhacs_74_version>** 替换为之前安装的 RHACS 版本。

2. 删除 **central-db** 持久性卷声明 (PVC)：

```
$ oc -n stackrox delete pvc central-db 1
```

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

第 3 章 使用 ROXCTL CLI 手动升级

您可以从受支持的旧版本升级到 Red Hat Advanced Cluster Security for Kubernetes (RHACS)的最新版本。

重要

- 只有在使用 **roxctl** CLI 安装 RHACS 时，才需要执行手动升级步骤。
- 每个版本升级都需要遵循手动步骤，例如从 3.74 升级到 4.0 版本，以及从 4.0 升级到 4.1 版本。因此，红帽建议先从 3.74 升级到 4.0，然后从 4.0 升级到 4.1，然后从 4.1 升级到 4.2，直到安装了所选版本。如需完整的功能，红帽建议升级到最新版本。

要将 RHACS 升级到最新版本，请执行以下步骤：

1. 备份中央数据库
2. 升级 **roxctl** CLI
3. 生成 Central 数据库置备捆绑包
4. 使用 Central DB 置备捆绑包创建资源
5. 升级 Central 集群
6. 升级所有安全集群

3.1. 备份 CENTRAL 数据库

在基础架构灾难的情况下，您可以备份中心数据库，并使用该备份从失败的升级或数据恢复中回滚。

先决条件

- 您必须具有一个 API 令牌，其具有对 Red Hat Advanced Cluster Security for Kubernetes 的所有资源的 **read** 权限。**analysts** 系统角色具有所有资源的 **read** 权限。
- 已安装了 **roxctl** CLI。
- 您已配置了 **ROX_API_TOKEN** 和 **ROX_CENTRAL_ADDRESS** 环境变量。

流程

- 运行备份命令：

```
$ roxctl -e "$ROX_CENTRAL_ADDRESS" central backup
```

其他资源

- 使用 roxctl CLI 进行身份验证

3.2. 升级 ROXCTL CLI

要将 **roxctl** CLI 升级到最新版本，您必须卸载 **roxctl** CLI 的现有版本，然后安装 **roxctl** CLI 的最新版本。

3.2.1. 卸载 roxctl CLI

您可以按照以下流程卸载 Linux 上的 **roxctl** CLI 二进制文件。

流程

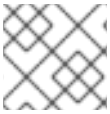
- 查找并删除 **roxctl** 二进制文件：

```
$ ROXPATH=$(which roxctl) && rm -f $ROXPATH 1
```

1 根据您的环境，您可能需要管理员删除 **roxctl** 二进制文件。

3.2.2. 在 Linux 中安装 roxctl CLI

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



注意

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

流程

1. 确定目标操作系统的 **roxctl** 架构：

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

2. 下载 **roxctl** CLI：

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

3. 使 **roxctl** 二进制文件可执行：

```
$ chmod +x roxctl
```

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

```
$ echo $PATH
```

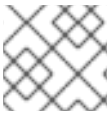
验证

- 验证您已安装的 **roxctl** 版本：

```
$ roxctl version
```

3.2.3. 在 macOS 上安装 roxctl CLI

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



注意

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

流程

1. 下载 **roxctl** CLI：

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

2. 从二进制文件中删除所有扩展属性：

```
$ xattr -c roxctl
```

3. 使 **roxctl** 二进制文件可执行：

```
$ chmod +x roxctl
```

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

```
$ echo $PATH
```

验证

- 验证您已安装的 **roxctl** 版本：

```
$ roxctl version
```

3.2.4. 在 Windows 上安装 roxctl CLI

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

注意

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

流程

- 下载 **roxctl** CLI：

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

验证

- 验证您已安装的 **roxctl** 版本：

```
$ roxctl version
```

3.3. 生成 CENTRAL 数据库置备捆绑包

在升级 Central 前，您必须首先生成数据库置备捆绑包。这个捆绑包是一个 **tar** 归档，它有一个 README 文件、几个 YAML 配置文件，以及在安装过程中帮助的一些脚本。

先决条件

- 您必须有带有 **Admin** 角色的 API 令牌。
- 您必须已安装了 **roxctl** CLI。

流程

1. 设置 **ROX_API_TOKEN** 和 **ROX_CENTRAL_ADDRESS** 环境变量：

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

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

2. 运行 **central db generate** 命令：

```
$ roxctl -e $ROX_CENTRAL_ADDRESS central db generate \
  <cluster_type> \ 1
  <storage> \ 2
  --output-dir <bundle_dir> \ 3
  --central-db-image registry.redhat.io/advanced-cluster-security/rhacs-central-db-rhel8:4.4.3
```

1. **cluster-type** 是集群的类型，为 Kubernetes 指定 **k8s**，为 OpenShift Container Platform 指定 **openshift**。
2. 对于 **storage**，指定 **hostpath** 或 **pvc**。如果使用 **pvc**，您可以使用附加选项来指定卷名称、大小和存储类。运行 **$ roxctl central db generate openshift pvc -h** 以了解更多详细信息。
3. 对于 **bundle-dir**，请指定您要保存生成的置备捆绑包的路径。

下一步

- 使用 Central DB 置备捆绑包来创建其他资源。

3.4. 使用 CENTRAL DB 置备捆绑包创建资源

在升级 Central 集群前，您必须使用 Central DB 置备捆绑包来创建 Central 集群所需的其他资源。这个捆绑包是一个 **tar** 归档，它有一个 README 文件、几个 YAML 配置文件，以及在安装过程中帮助的一些脚本。

先决条件

- 您必须已生成了一个 Central DB 置备捆绑包。
- 您必须已提取 **tar** 归档捆绑包。

流程

1. 打开提取的捆绑包目录并运行 **setup** 脚本 ：

```
$ ./scripts/setup.sh
```

2. 运行 **deploy-central-db** 脚本：

```
$ ./deploy-central-db.sh
```

3.5. 升级中央集群

在创建了 Central 数据库的备份并使用置备捆绑包生成必要的资源后，下一步是升级 Central 集群。这个过程涉及升级 Central 和 Scanner。

3.5.1. 升级 Central

您可以通过下载和部署更新的镜像，将 Central 更新到最新版本。

流程

- 运行以下命令以更新 Central 镜像：

```
$ oc -n stackrox set image deploy/central central=registry.redhat.io/advanced-cluster-security/rhacs-main-rhel8:4.4.3 1
```

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

验证

- 验证新 pod 是否已部署：

```
$ oc get deploy -n stackrox -o wide
```

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

3.5.1.1. 编辑 Central 部署的 GOMEMLIMIT 环境变量

升级到 4.4 版本需要您手动将 **GOMEMLIMIT** 环境变量替换为 **ROX_MEMLIMIT** 环境变量。您必须为每个部署编辑此变量。

流程

1. 运行以下命令来编辑 Central 部署的变量：

```
$ oc -n stackrox edit deploy/central 1
```

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

2. 将 **GOMEMLIMIT** 变量替换为 **ROX_MEMLIMIT**。
3. 保存该文件。

3.5.2. 升级扫描器

您可以通过下载和部署更新的镜像将 Scanner 更新至最新版本。

流程

- 运行以下命令以更新 Scanner 镜像：

```
$ oc -n stackrox set image deploy/scanner scanner=registry.redhat.io/advanced-cluster-security/rhacs-scanner-rhel8:4.4.3 1
```

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

验证

- 验证新 pod 是否已部署：

```
$ oc get deploy -n stackrox -o wide
```

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

3.5.2.1. 编辑 Scanner 部署的 GOMEMLIMIT 环境变量

升级到 4.4 版本需要您手动将 **GOMEMLIMIT** 环境变量替换为 **ROX_MEMLIMIT** 环境变量。您必须为每个部署编辑此变量。

流程

1. 运行以下命令来编辑 Scanner 部署的变量：

```
$ oc -n stackrox edit deploy/scanner 1
```

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

2. 将 **GOMEMLIMIT** 变量替换为 **ROX_MEMLIMIT**。
3. 保存该文件。

3.5.3. 验证 Central 集群升级

在升级了 Central 和 Scanner 后，验证该中央集群升级已完成。

流程

- 运行以下命令检查 Central 日志：

```
$ oc logs -n stackrox deploy/central -c central 1
```

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

成功升级的输出示例

```
No database restore directory found (this is not an error).
Migrator: 2023/04/19 17:58:54: starting DB compaction
Migrator: 2023/04/19 17:58:54: Free fraction of 0.0391 (40960/1048576) is < 0.7500. Will not
compact
badger 2023/04/19 17:58:54 INFO: All 1 tables opened in 2ms
badger 2023/04/19 17:58:55 INFO: Replaying file id: 0 at offset: 846357
badger 2023/04/19 17:58:55 INFO: Replay took: 50.324µs
badger 2023/04/19 17:58:55 DEBUG: Value log discard stats empty
Migrator: 2023/04/19 17:58:55: DB is up to date. Nothing to do here.
badger 2023/04/19 17:58:55 INFO: Got compaction priority: {level:0 score:1.73 dropPrefix:[]}
version: 2023/04/19 17:58:55.189866 ensure.go:49: Info: Version found in the DB was current. We're
good to go!
```

3.6. 升级所有安全集群

升级中部服务后，您必须升级所有安全的集群。

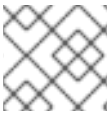
重要

- 如果您使用自动升级：
  - 使用自动升级来更新所有受保护的集群。
  - 跳过本节中的说明，并按照验证升级和撤销 API 令牌部分中的说明进行操作。
- 如果您不使用自动升级，则必须在包括 Central 集群在内的所有安全集群中运行本节中的说明。
  - 为确保最佳功能，请为您的安全集群和安装 Central 的集群使用相同的 RHACS 版本。

要完成每个运行 Sensor、Collector 和 Admission Controller 的安全集群的手动升级，请按照本节中的说明操作。

3.6.1. 更新其他镜像

在不使用自动升级时，您必须更新每个安全集群中的 sensor, collector 和 compliance 镜像。



注意

如果使用 Kubernetes，请在此流程中列出的命令中使用 **kubectl** 而不是 **oc**。

流程

1. 更新 Sensor 镜像：

```
$ oc -n stackrox set image deploy/sensor sensor=registry.redhat.io/advanced-cluster-
security/rhacs-main-rhel8:4.4.3 (1)
```

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

2. 更新 Compliance 镜像：

```
$ oc -n stackrox set image ds/collector compliance=registry.redhat.io/advanced-cluster-security/rhacs-main-rhel8:4.4.3 1
```

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

3. 更新 Collector 镜像：

```
$ oc -n stackrox set image ds/collector collector=registry.redhat.io/advanced-cluster-security/rhacs-collector-rhel8:4.4.3 1
```

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

注意

如果使用 collector slim 镜像，请运行以下命令：

```
$ oc -n stackrox set image ds/collector collector=registry.redhat.io/advanced-cluster-security/rhacs-collector-slim-rhel8:{rhacs-version}
```

4. 更新准入控制镜像：

```
$ oc -n stackrox set image deploy/admission-control admission-control=registry.redhat.io/advanced-cluster-security/rhacs-main-rhel8:4.4.3
```

重要

如果使用 **roxctl** CLI 在 Red Hat OpenShift 上安装 RHACS，则需要迁移安全性上下文约束(SCC)。

如需更多信息，请参阅"添加资源"部分中的"手动升级过程中调整 SCC"。

后续步骤

- 验证安全集群升级

其他资源

- 在手动升级过程中迁移 SCC

3.6.2. 在手动升级过程中迁移 SCC

通过使用 **roxctl** CLI 在手动升级过程中迁移安全性上下文约束(SCC)，您可以无缝转换 Red Hat Advanced Cluster Security for Kubernetes (RHACS)服务以使用 Red Hat OpenShift SCC，确保 Central 和所有安全集群间的兼容性和最佳安全配置。

流程

1. 列出在 Central 和所有安全集群中部署的所有 RHACS 服务：

```
$ oc -n stackrox describe pods | grep 'openshift.io/scc\|^Name:'
```

输出示例

```
Name:      admission-control-6f4dcc6b4c-2phwd
           openshift.io/scc: stackrox-admission-control
#...
Name:      central-575487bfcb-sjdx8
           openshift.io/scc: stackrox-central
Name:      central-db-7c7885bb-6bgbd
           openshift.io/scc: stackrox-central-db
Name:      collector-56nkr
           openshift.io/scc: stackrox-collector
#...
Name:      scanner-68fc55b599-f2wm6
           openshift.io/scc: stackrox-scanner
Name:      scanner-68fc55b599-fztlh
#...
Name:      sensor-84545f86b7-xgdwf
           openshift.io/scc: stackrox-sensor
#...
```

在本例中，您可以看到每个 pod 都有自己的自定义 SCC，这通过 **openshift.io/scc** 字段指定。

2. 添加所需的角色和角色绑定，以使用 Red Hat OpenShift SCC 而不是 RHACS 自定义 SCC。
要添加所需的角色和角色绑定，以将 Red Hat OpenShift SCC 用于 Central 集群，请执行以下步骤：

   a. 创建名为 **update-central.yaml** 的文件，该文件使用以下内容定义角色和角色绑定资源：

   例 3.1. YAML 文件示例

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role (1)
metadata:
  annotations:
    email: support@stackrox.com
    owner: stackrox
  labels:
    app.kubernetes.io/component: central
    app.kubernetes.io/instance: stackrox-central-services
    app.kubernetes.io/name: stackrox
    app.kubernetes.io/part-of: stackrox-central-services
    app.kubernetes.io/version: 4.4.0
  name: use-central-db-scc (2)
  namespace: stackrox (3)
Rules: (4)
- apiGroups:
  - security.openshift.io
  resourceNames:
  - nonroot-v2
  resources:
  - securitycontextconstraints
  verbs:
  - use
```

```
- - -
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  annotations:
    email: support@stackrox.com
    owner: stackrox
  labels:
    app.kubernetes.io/component: central
    app.kubernetes.io/instance: stackrox-central-services
    app.kubernetes.io/managed-by: Helm
    app.kubernetes.io/name: stackrox
    app.kubernetes.io/part-of: stackrox-central-services
    app.kubernetes.io/version: 4.4.0
  name: use-central-scc
  namespace: stackrox
rules:
- apiGroups:
  - security.openshift.io
  resourceNames:
  - nonroot-v2
  resources:
  - securitycontextconstraints
  verbs:
  - use
- - -
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  annotations:
    email: support@stackrox.com
    owner: stackrox
  labels:
    app.kubernetes.io/component: scanner
    app.kubernetes.io/instance: stackrox-central-services
    app.kubernetes.io/name: stackrox
    app.kubernetes.io/part-of: stackrox-central-services
    app.kubernetes.io/version: 4.4.0
  name: use-scanner-scc
  namespace: stackrox
rules:
- apiGroups:
  - security.openshift.io
  resourceNames:
  - nonroot-v2
  resources:
  - securitycontextconstraints
  verbs:
  - use
- - -
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding (5)
metadata:
  annotations:
    email: support@stackrox.com
    owner: stackrox
```

```
  labels:
    app.kubernetes.io/component: central
    app.kubernetes.io/instance: stackrox-central-services
    app.kubernetes.io/name: stackrox
    app.k ubernetes.io/part-of: stackrox-central-services
    app.kubernetes.io/version: 4.4.0
  name: central-db-use-scc 6
  namespace: stackrox
roleRef: 7
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: use-central-db-scc
subjects: 8
- kind: ServiceAccount
  name: central-db
  namespace: stackrox
- - -
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  annotations:
    email: support@stackrox.com
    owner: stackrox
  labels:
    app.kubernetes.io/component: central
    app.kubernetes.io/instance: stackrox-central-services
    app.kubernetes.io/name: stackrox
    app.kubernetes.io/part-of: stackrox-central-services
    app.kubernetes.io/version: 4.4.0
  name: central-use-scc
  namespace: stackrox
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: use-central-scc
subjects:
- kind: ServiceAccount
  name: central
  namespace: stackrox
- - -
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  annotations:
    email: support@stackrox.com
    owner: stackrox
  labels:
    app.kubernetes.io/component: scanner
    app.kubernetes.io/instance: stackrox-central-services
    app.kubernetes.io/name: stackrox
    app.kubernetes.io/part-of: stackrox-central-services
    app.kubernetes.io/version: 4.4.0
  name: scanner-use-scc
  namespace: stackrox
roleRef:
  apiGroup: rbac.authorization.k8s.io
```

```
  kind: Role
  name: use-scanner-scc
subjects:
- kind: ServiceAccount
  name: scanner
  namespace: stackrox
- - -
```

1. Kubernetes 资源的类型，本例中为 **Role**。
2. 角色资源的名称。
3. 创建角色的命名空间。
4. 描述角色资源授予的权限。
5. Kubernetes 资源的类型，在本例中为 **RoleBinding**。
6. 角色绑定资源的名称。
7. 指定要在同一命名空间中绑定的角色。
8. 指定绑定到角色的主题。

b. 运行以下命令，创建 **update-central.yaml** 文件中指定的角色和角色绑定资源：

```
$ oc -n stackrox create -f ./update-central.yaml
```

3. 要添加所需的角色和角色绑定，以便对所有安全集群使用 Red Hat OpenShift SCC，请执行以下步骤：

a. 创建名为 **upgrade-scs.yaml** 的文件，该文件使用以下内容定义角色和角色绑定资源：

例 3.2. YAML 文件示例

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role 1
metadata:
  annotations:
    email: support@stackrox.com
    owner: stackrox
  labels:
    app.kubernetes.io/component: collector
    app.kubernetes.io/instance: stackrox-secured-cluster-services
    app.kubernetes.io/name: stackrox
    app.kubernetes.io/part-of: stackrox-secured-cluster-services
    app.kubernetes.io/version: 4.4.0
    auto-upgrade.stackrox.io/component: sensor
  name: use-privileged-scc 2
  namespace: stackrox 3
rules: 4
- apiGroups:
  - security.openshift.io
  resourceNames:
```

```
  - privileged
  resources:
  - securitycontextconstraints
  verbs:
  - use
- - -
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding (5)
metadata:
  annotations:
    email: support@stackrox.com
    owner: stackrox
  labels:
    app.kubernetes.io/component: collector
    app.kubernetes.io/instance: stackrox-secured-cluster-services
    app.kubernetes.io/name: stackrox
    app.kubernetes.io/part-of: stackrox-secured-cluster-services
    app.kubernetes.io/version: 4.4.0
    auto-upgrade.stackrox.io/component: sensor
  name: collector-use-scc (6)
  namespace: stackrox
roleRef: (7)
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: use-privileged-scc
subjects: (8)
- kind: ServiceAccount
  name: collector
  namespace: stackrox
- - -
```

(1) Kubernetes 资源的类型，本例中为 **Role**。

(2) 角色资源的名称。

(3) 创建角色的命名空间。

(4) 描述角色资源授予的权限。

(5) Kubernetes 资源的类型，在本例中为 **RoleBinding**。

(6) 角色绑定资源的名称。

(7) 指定要在同一命名空间中绑定的角色。

(8) 指定绑定到角色的主题。

b. 运行以下命令，创建 **upgrade-scs.yaml** 文件中指定的角色和角色绑定资源：

```
$ oc -n stackrox create -f ./update-scs.yaml
```

重要

您必须在每个安全集群中运行此命令，以创建 **upgrade-scs.yaml** 文件中指定的角色和角色绑定。

4. 删除特定于 RHACS 的 SCC：

   a. 要删除特定于 Central 集群的 SCC，请运行以下命令：

   ```
   $ oc delete scc/stackrox-central scc/stackrox-central-db scc/stackrox-scanner
   ```

   b. 要删除特定于所有安全集群的 SCC，请运行以下命令：

   ```
   $ oc delete scc/stackrox-admission-control scc/stackrox-collector scc/stackrox-sensor
   ```

重要

您必须在每个安全集群中运行此命令，以删除特定于每个安全集群的 SCC。

验证

- 运行以下命令，确保所有 Pod 都使用正确的 SCC：

  ```
  $ oc -n stackrox describe pods | grep 'openshift.io/scc\|^Name:'
  ```

  将输出与下表进行比较：

组件	以前的自定义 SCC	新的 Red Hat OpenShift 4 SCC
Central	**stackrox-central**	**nonroot-v2**
Central-db	**stackrox-central-db**	**nonroot-v2**
扫描程序	**stackrox-scanner**	**nonroot-v2**
Scanner-db	**stackrox-scanner**	**nonroot-v2**
Admission Controller	**stackrox-admission-control**	**restricted-v2**
Collector	**stackrox-collector**	**privileged**
Sensor	**stackrox-sensor**	**restricted-v2**

3.6.2.1. 编辑 Sensor 部署的 GOMEMLIMIT 环境变量

升级到 4.4 版本需要您手动将 **GOMEMLIMIT** 环境变量替换为 **ROX_MEMLIMIT** 环境变量。您必须为每个部署编辑此变量。

流程

1. 运行以下命令来编辑 Sensor 部署的变量：

```
$ oc -n stackrox edit deploy/sensor (1)
```

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

2. 将 **GOMEMLIMIT** 变量替换为 **ROX_MEMLIMIT**。
3. 保存该文件。

3.6.2.2. 编辑 Collector 部署的 GOMEMLIMIT 环境变量

升级到 4.4 版本需要您手动将 **GOMEMLIMIT** 环境变量替换为 **ROX_MEMLIMIT** 环境变量。您必须为每个部署编辑此变量。

流程

1. 运行以下命令来编辑 Collector 部署的变量：

```
$ oc -n stackrox edit deploy/collector (1)
```

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

2. 将 **GOMEMLIMIT** 变量替换为 **ROX_MEMLIMIT**。
3. 保存该文件。

3.6.2.3. 编辑 Admission Controller 部署的 GOMEMLIMIT 环境变量

升级到 4.4 版本需要您手动将 **GOMEMLIMIT** 环境变量替换为 **ROX_MEMLIMIT** 环境变量。您必须为每个部署编辑此变量。

流程

1. 运行以下命令来编辑 Admission Controller 部署的变量：

```
$ oc -n stackrox edit deploy/admission-control (1)
```

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

2. 将 **GOMEMLIMIT** 变量替换为 **ROX_MEMLIMIT**。
3. 保存该文件。

3.6.2.4. 验证安全集群升级

在升级了安全集群后，验证更新的 pod 是否正常工作。

流程

- 检查新 pod 是否已部署：

```
$ oc get deploy,ds -n stackrox -o wide
```
(1)

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

```
$ oc get pod -n stackrox --watch
```
(1)

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

3.7. 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":"registry.red
hat.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"}]}]}}}}'
```

其他资源

- 扫描 RHCOS 节点主机

3.8. 升级到 4.1 及之后的版本后，删除 CENTRAL 附加 PV

Kubernetes 和 OpenShift Container Platform 不自动删除持久性卷(PV)。当您从早期版本升级 RHACS 时，名为 **stackrox-db** 的 Central PV 会保留挂载。但是，在 RHACS 4.1 中，Central 不再需要之前附加的 PV。

PV 具有之前 RHACS 版本使用的数据和持久性文件。您可以使用 PV 在 RHACS 4.1 之前回滚到更早的版本。或者，如果您有一个用于 Central 的大型 RocksDB 备份捆绑包，您可以使用 PV 恢复这些数据。

完成升级到 4.1 后，您可以删除 Central 附加的持久性卷声明(PVC)来释放存储。仅当没有计划从之前的 RocksDB 备份回滚或恢复时，才删除 PVC。

警告

删除 PVC 后，您无法在 RHACS 4.1 之前将 Central 回滚到早期版本，或恢复使用 RocksDB 创建的大型 RocksDB 备份。

3.8.1. 使用 roxctl CLI 删除 Central 附加 PV

删除 Central 附加持久性卷声明(PVC) **stackrox-db** 以释放存储空间。

流程

- 运行以下命令：

```
$ oc get deployment central -n stackrox -o json | jq '(.spec.template.spec.volumes[] |
select(.name=="stackrox-db"))={"name": "stackrox-db", "emptyDir": {}}' | oc apply -f -
```

它将 **spec.template.spec.volumes** 中的 **stackrox-db'** 条目替换为本地 emptyDir。

验证

- 运行以下命令：

```
$ oc -n stackrox describe pvc stackrox-db | grep -i 'Used By'
Used By: <none> 1
```

1 等待您看到过的 By: **<none>**。它可能需要几分钟时间。

3.9. 回滚 CENTRAL

如果升级到新版本失败，您可以回滚到以前版本的 Central。

3.9.1. 正常回滚 Central

如果升级 Red Hat Advanced Cluster Security for Kubernetes 失败，您可以回滚到以前版本的 Central。

先决条件

- 执行回滚前，持久性存储必须具有可用磁盘空间。Red Hat Advanced Cluster Security for Kubernetes 使用磁盘空间在升级过程中保留数据库副本。如果磁盘空间不足以存储副本，升级失败，您将无法回滚到较早的版本。

流程

- 当升级失败时（Central 服务启动前），运行以下命令回滚到以前的版本：

```
$ oc -n stackrox rollout undo deploy/central 1
```

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

3.9.2. 强制回滚 Central

您可以使用强制回滚回滚到较早版本的 Central （在 Central 服务启动后）。

重要

使用强制回滚切换到以前的版本可能会导致数据丢失和功能。

先决条件

- 执行回滚前，持久性存储必须具有可用磁盘空间。Red Hat Advanced Cluster Security for Kubernetes 使用磁盘空间在升级过程中保留数据库副本。如果磁盘空间不足以存储副本，升级失败，您将无法回滚到较早的版本。

流程

- 运行以下命令来执行强制回滚：
  - 要强制回滚到以前安装的版本：

```
$ oc -n stackrox rollout undo deploy/central 1
```

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

  - 强制回滚到特定版本：

1. 编辑 Central 的 **ConfigMap** ：

```
$ oc -n stackrox edit configmap/central-config 1
```

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

2. 更新 **maintenance.forceRollbackVersion** 键的值：

```
data:
  central-config.yaml: |
    maintenance:
      safeMode: false
      compaction:
        enabled: true
        bucketFillFraction: .5
        freeFractionThreshold: 0.75
      forceRollbackVersion: <x.x.x.x> 1
    ...
```

1 指定要回滚到的版本。

3. 更新 Central 镜像版本：

```
$ oc -n stackrox \ 1
  set image deploy/central central=registry.redhat.io/advanced-cluster-security/rhacs-main-rhel8:<x.x.x.x> 2
```

1. 如果使用 Kubernetes，请输入 **kubectl** 而不是 **oc**。
2. 指定要回滚到的版本。它必须与您在 **central-config** 配置映射中为 **maintenance.forceRollbackVersion** 键指定的版本相同。

3.10. 验证升级

更新的 Sensors 和 Collector 将继续从每个安全集群报告最新的数据。

在 RHACS 门户中可以看到 Sensor 最后联系 Central 的时间。

流程

1. 在 RHACS 门户中，进入 **Platform Configuration → System Health**。
2. 检查以确保 Sensor Upgrade 显示使用 Central 的集群最新。

3.11. 撤销 API 令牌

为了安全起见，红帽建议您撤销用于完成 Central 数据库备份的 API 令牌。

先决条件

- 升级后，您必须重新加载 RHACS 门户页面并重新接受证书，以便继续使用 RHACS 门户。

流程

1. 在 RHACS 门户中，进入 **Platform Configuration → Integrations**。
2. 向下滚动到 **Authentication Tokens** 类别，然后点 **API Token**。
3. 选中您要撤销的令牌名称前面的复选框。
4. 点 **Revoke**。
5. 在确认对话框中，点 **Confirm**。