



# Red Hat Advanced Cluster Security for Kubernetes 3.73

## アップグレード

Red Hat Advanced Cluster Security for Kubernetes のアップグレード



# Red Hat Advanced Cluster Security for Kubernetes 3.73 アップグレード

---

Red Hat Advanced Cluster Security for Kubernetes のアップグレード

## 法律上の通知

Copyright © 2023 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## 概要

このセクションでは、Helm チャートまたは `roxctl` コマンドラインインターフェイスを使用して、Red Hat Advanced Cluster Security for Kubernetes をアップグレードする手順を説明します。

---

## 目次

<b>第1章 OPERATOR を使用したアップグレード</b> .....	<b>3</b>
1.1. CLI を使用した OPERATOR アップグレードのロールバック	3
1.2. WEB コンソールを使用した OPERATOR アップグレードのロールバック	5
1.3. 関連情報	7
<b>第2章 HELM チャートを使用したアップグレード</b> .....	<b>8</b>
2.1. HELM チャートリポジトリの更新	8
2.2. 関連情報	8
<b>第3章 ROXCTL CLI を使用して手動でアップグレード</b> .....	<b>9</b>
3.1. ROX_SCANNER_DB_INIT 環境変数を設定する	9
3.2. CENTRAL データベースのバックアップ	9
3.3. CENTRAL クラスターのアップグレード	10
3.4. すべてのセキュアなクラスターのアップグレード	17
3.5. CENTRAL のロールバック	19
3.6. アップグレードの確認	21
3.7. API トークンの取り消し	22



## 第1章 OPERATOR を使用したアップグレード

Red Hat Advanced Cluster Security for Kubernetes (RHACS) Operator を介したアップグレードは、インストール時に選択した **Update approval** オプションに応じて、自動または手動で実行されます。

Operator を使用して RHACS をインストールし、**Update approval** フィールドで **Automatic** を選択した場合は、新しいソフトウェアバージョンがリリースされると、RHACS は自動的に更新されます。**Manual** を選択した場合は、Operator Lifecycle Manager (OLM) を使用して後続の Operator 更新を承認する必要があります。詳細は、[保留中の Operator 更新を手動で承認する](#) を参照してください。

Operator アップグレードをロールバックするには、次のセクションのいずれかで説明されている手順を実行する必要があります。CLI または OpenShift Container Platform Web コンソールを使用して、Operator アップグレードをロールバックできます。

### 1.1. CLI を使用した OPERATOR アップグレードのロールバック

CLI コマンドを使用して Operator バージョンをロールバックできます。

#### 手順

1. 次のコマンドを実行して、OLM サブスクリプションを削除します。

- OpenShift Container Platform の場合、以下のコマンドを実行します。

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

- Kubernetes の場合、次のコマンドを実行します。

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

2. 次のコマンドを実行して、クラスターサービスバージョン (CSV) を削除します。

- OpenShift Container Platform の場合、以下のコマンドを実行します。

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

- Kubernetes の場合、次のコマンドを実行します。

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

3. 次のオプションのいずれかを選択して、ロールバックする前のバージョンを決定します。

- 現在の Central インスタンスが実行中の場合は、次のコマンドを実行して RHACS API にクエリーを実行し、ロールバックバージョンを取得します。

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

- 現在の Central インスタンスが実行されていない場合は、次の手順を実行します。



## 注記

この手順は、**rocksdb** データベースがインストールされている RHACS リリース 3.74 以前でのみ使用できます。

- a. 次のコマンドを実行して、Central デプロイメントがスケールダウンされていることを確認します。

- OpenShift Container Platform の場合、以下のコマンドを実行します。

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

- Kubernetes の場合、次のコマンドを実行します。

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

- b. 次の Pod 仕様を YAML ファイルとして保存します。

```
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. 保存した YAML ファイルを使用し、次のコマンドを実行して、Central namespace に Pod を作成します。

- OpenShift Container Platform の場合、以下のコマンドを実行します。

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

- Kubernetes の場合、次のコマンドを実行します。

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

- d. Pod の作成が完了したら、次のコマンドを実行してバージョンを取得します。



- OpenShift Container Platform の場合、以下のコマンドを実行します。

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

- Kubernetes の場合、次のコマンドを実行します。

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

4. 次のコマンドを実行して、**central-config.yaml ConfigMap** を編集して、**maintenance.forceRollBackVersion:<version>** パラメーターを設定します。

- OpenShift Container Platform の場合、以下のコマンドを実行します。

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

- Kubernetes の場合、次のコマンドを実行します。

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

5. ステップ 3 で示されたバージョン文字列をイメージタグとして使用して、Central デプロイメントのイメージを設定します。たとえば、以下のコマンドを実行します。

- OpenShift Container Platform の場合、以下のコマンドを実行します。

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

- Kubernetes の場合、次のコマンドを実行します。

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

## 検証

1. Central Pod が起動し、**ready** ステータスになっていることを確認します。Pod がクラッシュした場合は、ログをチェックして、バックアップが復元されたかどうかを確認します。成功した場合のログメッセージは、次の例のように表示されます。

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

2. ロールバックされたチャンネルに Operator を再インストールします。たとえば、**3.71.3** は **rhacs-3.71** チャンネルにインストールされます。

## 1.2. WEB コンソールを使用した OPERATOR アップグレードのロールバック

OpenShift Container Platform Web コンソールを使用して Operator バージョンをロールバックできます。

## 前提条件

- **cluster-admin** パーミッションを持つアカウントを使用して OpenShift Container Platform クラスタ Web コンソールにアクセスできる。

## 手順

1. **Operators** → **Installed Operators** ページに移動します。
2. RHACS Operator を検索およびクリックします。
3. **Operator Details** ページで、**Actions** リストから **Uninstall Operator** を選択します。このアクションの後には、Operator は実行を停止し、更新を受信しなくなります。
4. 次のオプションのいずれかを選択して、ロールバックする前のバージョンを決定します。
  - 現在の Central インスタンスが実行中の場合は、ターミナルウィンドウから次のコマンドを実行して、RHACS API をクエリーしてロールバックバージョンを取得できます。

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

- 次の手順を実行して、Pod を作成し、以前のバージョンを解凍できます。



### 注記

この手順は、**rocksdb** データベースがインストールされている RHACS リリース 3.74 以前でのみ使用できます。

- a. **Workloads** → **Deployments** → **central** に移動します。
- b. **Deployment details** で、Pod 数の横にある下矢印をクリックして Pod をスケールダウンします。
- c. **Workloads** → **Pods** → **Create Pod** に移動し、次の例に示すように Pod 仕様の内容をエディターに貼り付けます。

```
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. **Create** をクリックします。
  - e. Pod が作成されたら、**Logs** タブをクリックしてバージョン文字列を取得します。
5. 次の手順を実行して、ロールバック設定を更新します。
    - a. **Workloads** → **ConfigMaps** → **central-config** に移動し、**Actions** リストから **Edit ConfigMap** を選択します。
    - b. **central-config.yaml** キーの値で、**forceRollbackVersion** 行を見つけます。
    - c. **none** を **3.73.3** に置き換えて、ファイルを保存します。
  6. 次の手順を実行して、Central を以前のバージョンに更新します。
    - a. **Workloads** → **Deployments** → **central** に移動し、**Actions** リストから **Edit Deployment** を選択します。
    - b. イメージ名を更新し、変更を保存します。

## 検証

1. Central Pod が起動し、**ready** ステータスになっていることを確認します。Pod がクラッシュした場合は、ログをチェックして、バックアップが復元されたかどうかを確認します。成功した場合のログメッセージは、次の例のように表示されます。

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

2. ロールバックされたチャンネルに Operator を再インストールします。たとえば、**3.71.3** は **rhacs-3.71** チャンネルにインストールされます。

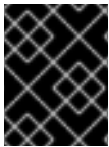
## 1.3. 関連情報

- [Operator メソッドを使用した Central のインストール](#)
- [Operator Lifecycle Manager ワークフロー](#)
- [保留中の Operator 更新の手動による承認](#)

## 第2章 HELM チャートを使用したアップグレード

Helm チャートを使用して Red Hat Advanced Cluster Security for Kubernetes をインストールしていて、Red Hat Advanced Cluster Security for Kubernetes の最新バージョンにアップグレードするには、次の手順を実行する必要があります。

- Helm チャートを更新します。
- central-services Helm チャートの設定ファイルを更新します。
- central-services Helm チャートをアップグレードします。
- secured-cluster-services Helm チャートの設定ファイルを更新します。
- secured-cluster-services Helm チャートをアップグレードします。



### 重要

最適な機能を確保するには、secure-cluster-services Helm チャートと central-services Helm チャートに同じバージョンを使用してください。

### 2.1. HELM チャートリポジトリの更新

Red Hat Advanced Cluster Security for Kubernetes の新しいバージョンにアップグレードする前に、常に Helm チャートを更新する必要があります。

#### 前提条件

- Red Hat Advanced Cluster Security for Kubernetes の Helm チャートリポジトリをすでに追加している必要がある。

#### 手順

- Red Hat Advanced Cluster Security for Kubernetes チャートリポジトリを追加します。

```
$ helm repo update
```

#### 検証

- 次のコマンドを実行して、追加されたチャートリポジトリを確認します。

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

### 2.2. 関連情報

- [Helm チャートを使用した Central のインストール](#)
- [Helm チャートを使用したセキュアなクラスターへの RHACS のインストール](#)

## 第3章 ROXCTL CLI を使用して手動でアップグレード

Red Hat Advanced Cluster Security for Kubernetes (RHACS) のサポートされている古いバージョンから最新バージョンにアップグレードできます。



### 注記

手動アップグレード手順を実行する必要があるのは、**roxctl** CLI を使用して RHACS をデプロイした場合のみです。

RHACS を最新バージョンにアップグレードするには、以下を実行する必要があります。

- **ROX\_SCANNER\_DB\_INIT** 環境変数を設定する
- Central データベースをバックアップする
- Central をアップグレードする
- **roxctl** CLI をアップグレードする
- スキャナーをアップグレードする
- 保護されたすべてのクラスターがアップグレードされていることを確認する

### 3.1. ROX\_SCANNER\_DB\_INIT 環境変数を設定する

ScannerDB の **initContainer** には、**ROX\_SCANNER\_DB\_INIT** という新しい環境変数が必要です。アップグレードする前に、その値を **true** に設定する必要があります。

#### 手順

- OpenShift Container Platform の場合、以下のコマンドを実行します。

```
$ oc -n stackrox set env deploy/scanner-db -c init-db ROX_SCANNER_DB_INIT=true
```

- Kubernetes の場合、次のコマンドを実行します。

```
$ kubectl -n stackrox set env deploy/scanner-db -c init-db ROX_SCANNER_DB_INIT=true
```

### 3.2. CENTRAL データベースのバックアップ

Central データベースをバックアップし、そのバックアップを使用して、インフラストラクチャーの障害が発生した場合に、失敗したアップグレードまたはデータの復元からロールバックすることができます。

#### 前提条件

- Red Hat Advanced Cluster Security for Kubernetes のすべてのリソースに対する **read** 権限を持つ API トークンがある。**Analyst** システムロールには、すべてのリソースに対する **read** 権限がある。
- **roxctl** CLI をインストールした。

- **ROX\_API\_TOKEN** および **ROX\_CENTRAL\_ADDRESS** 環境変数が設定されている。

## 手順

- backup コマンドを実行します。
  - Red Hat Advanced Cluster Security for Kubernetes 3.0.55 以降の場合:

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

- Red Hat Advanced Cluster Security for Kubernetes 3.0.54 以前の場合:

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

## 関連情報

- [roxctl CLI を使用した認証](#)

## 3.3. CENTRAL クラスターのアップグレード

Central データベースをバックアップしたら、次のステップは Central クラスターをアップグレードすることです。この手順には、Central、**roxctl** CLI、および Scanner のアップグレードが含まれます。

### 3.3.1. Central のアップグレード

更新されたイメージをダウンロードしてデプロイすることにより、Central を最新バージョンに更新できます。

#### 3.3.1.1. OpenShift Container Platform での Central のアップグレード

OpenShift Container Platform に Red Hat Advanced Cluster Security for Kubernetes をインストールした場合は、以下の手順を使用してアップグレードします。

## 手順

1. ローカルロールにパッチを適用します。

```
$ oc -n stackrox patch role edit -p '{"rules":[{"apiGroups":["*"],"resources":["*"],"verbs":["create","get","list","watch","update","patch","delete","deletecollection"]}]}'
```

2. 既存のロールとロールバインディングをクリーンアップします。

```
$ oc -n stackrox delete RoleBinding admission-control-use-scc || true
```

```
$ oc -n stackrox delete RoleBinding sensor-use-scc || true
```

```
$ oc -n stackrox delete Role use-anyuid-scc || true
```

3. ハードコーディングされたセキュリティコンテキストを削除して、**sensor**、**admission-control**、**restricted[v2]** セキュリティコンテキスト制約に設定します。

```
$ oc -n stackrox patch deploy sensor -p '{"spec":{"template":{"spec":{"securityContext":null}}}}' 1
```

- 1 Red Hat Advanced Cluster Security for Kubernetes は Pod を自動的に再作成しますが、**sensor** の再起動には時間がかかる場合があります。

```
$ oc -n stackrox patch deploy admission-control -p '{"spec":{"template":{"spec":{"securityContext":null}}}}'
```

4. 次のコマンドを実行して、Central をアップグレードします。

```
$ oc -n stackrox patch deploy/central -p '{"spec":{"template":{"spec":{"containers":[{"name":"central","env":[{"name":"ROX_NAMESPACE","valueFrom":{"fieldRef":{"fieldPath":"metadata.namespace"}}}]}}}}}'
```

```
$ oc -n stackrox patch deployment/scanner -p '{"spec":{"template":{"spec":{"containers":[{"name":"scanner","securityContext":{"runAsUser":65534}}]}}}}'
```

```
$ oc -n stackrox set image deploy/central central=registry.redhat.io/advanced-cluster-security/rhacs-main-rhel8:3.73.5 1
```

- 1 プライベートイメージレジストリーからイメージをデプロイする場合は、新しいイメージをプライベートレジストリーにプッシュし、ここでイメージレジストリーアドレスを置き換えます。

## 重要

Helm または Operator を使用して Red Hat Advanced Cluster Security for Kubernetes をインストールしておらず、OpenShift OAuth サーバーを使用して認証を有効にする場合は、次の追加コマンドを実行する必要があります。

```
$ oc -n stackrox set env deploy/central  
ROX_ENABLE_OPENSHIFT_AUTH=true
```

```
$ oc -n stackrox patch serviceaccount/central -p '  
{  
  "metadata": {  
    "annotations": {  
      "serviceaccounts.openshift.io/oauth-redirecturi.main":  
        "sso/providers/openshift/callback",  
      "serviceaccounts.openshift.io/oauth-redirectreference.main": "  
        {"kind":"OAuthRedirectReference","apiVersion":"v1","reference":  
        {"kind":"Route","name":"central"}}"  
    }  
  }  
}'
```

## 検証

- 新しい Pod がデプロイされたことを確認します。

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

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

### 3.3.1.2. Central on Kubernetes のアップグレード

Red Hat Advanced Cluster Security for Kubernetes を Kubernetes にインストールした場合は、以下の手順を使用してアップグレードします。

#### 前提条件

- プライベートイメージレジストリーからイメージをデプロイする場合は、最初に新しいイメージをプライベートレジストリーにプッシュしてから、次のコマンドでイメージレジストリーを置き換えます。

#### 手順

- ローカルロールにパッチを適用します。

```
$ kubectl -n stackrox patch role edit -p '{"rules":[{"apiGroups":["*"],"resources":["*"],"verbs":["create","get","list","watch","update","patch","delete","deletecollection"]}']}'
```

- 次のコマンドを実行して、Central をアップグレードします。

```
$ kubectl -n stackrox patch deploy/central -p '{"spec":{"template":{"spec":{"containers":[{"name":"central","env":[{"name":"ROX_NAMESPACE","valueFrom":{"fieldRef":{"fieldPath":"metadata.namespace"}}}]}}}}}'
```

```
$ kubectl -n stackrox patch deployment/scanner -p '{"spec":{"template":{"spec":{"containers":[{"name":"scanner","securityContext":{"runAsUser":65534}}]}}}'
```

```
$ kubectl -n stackrox set image deploy/central central=registry.redhat.io/advanced-cluster-security/rhacs-main-rhel8:3.73.5 1
```

- 1** プライベートイメージレジストリーからイメージをデプロイする場合は、新しいイメージをプライベートレジストリーにプッシュし、ここでイメージレジストリーアドレスを置き換えます。

#### 検証

- 新しい Pod がデプロイされたことを確認します。

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

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

### 3.3.2. roxctl CLI のアップグレード

**roxctl** CLI を最新バージョンにアップグレードするには、既存のバージョンの **roxctl** CLI をアンインストールしてから、最新バージョンの **roxctl** CLI をインストールする必要があります。



### 3.3.2.1. roxctl CLI のアンインストール

次の手順を使用して、Linux に **roxctl** CLI バイナリーをアンインストールできます。

#### 手順

- **roxctl** バイナリーを見つけて削除します。

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

- 1 環境によっては、**roxctl** バイナリーを削除するために管理者権限が必要になる場合があります。

### 3.3.2.2. Linux への roxctl CLI のインストール

次の手順を使用して、Linux に **roxctl** CLI バイナリーをインストールできます。

#### 手順

1. **roxctl** CLI の最新バージョンをダウンロードします。

```
$ curl -O https://mirror.openshift.com/pub/rhacs/assets/3.73.5/bin/Linux/roxctl
```

2. **roxctl** バイナリーを実行可能にします。

```
$ chmod +x roxctl
```

3. **PATH** 上にあるディレクトリーに **roxctl** バイナリーを配置します。  
**PATH** を確認するには、以下のコマンドを実行します。

```
$ echo $PATH
```

#### 検証

- インストールした **roxctl** のバージョンを確認します。

```
$ roxctl version
```

### 3.3.2.3. macOS への roxctl CLI のインストール

次の手順を使用して、**roxctl** CLI バイナリーを macOS にインストールできます。

#### 手順

1. **roxctl** CLI の最新バージョンをダウンロードします。

```
$ curl -O https://mirror.openshift.com/pub/rhacs/assets/3.73.5/bin/Darwin/roxctl
```

2. バイナリーからすべての拡張属性を削除します。

```
$ xattr -c roxctl
```

3. **roxctl** バイナリーを実行可能にします。

```
$ chmod +x roxctl
```

4. **PATH** 上にあるディレクトリーに **roxctl** バイナリーを配置します。  
**PATH** を確認するには、以下のコマンドを実行します。

```
$ echo $PATH
```

#### 検証

- インストールした **roxctl** のバージョンを確認します。

```
$ roxctl version
```

#### 3.3.2.4. Windows への roxctl CLI のインストール

次の手順を使用して、**roxctl** CLI バイナリーを Windows にインストールできます。

#### 手順

- **roxctl** CLI の最新バージョンをダウンロードします。

```
$ curl -O https://mirror.openshift.com/pub/rhacs/assets/3.73.5/bin/Windows/roxctl.exe
```

#### 検証

- インストールした **roxctl** のバージョンを確認します。

```
$ roxctl version
```

**roxctl** CLI をアップグレードした後、Scanner をアップグレードできます。

#### 3.3.3. Scanner のアップグレード

**roxctl** CLI を使用して、Scanner を最新バージョンに更新できます。

#### 前提条件

- プライベートイメージレジストリーからイメージをデプロイする場合は、最初に新しいイメージをプライベートレジストリーにプッシュしてから、次のセクションのコマンドを編集して、プライベートイメージレジストリーの名前を使用する必要があります。

#### 手順

1. カスタムスキャナー設定を作成した場合は、スキャナー設定ファイルを更新する前に、これらの変更を適用する必要があります。
  - a. 次の **roxctl** コマンドを使用してスキャナーを生成します。

```
$ roxctl -e "$ROX_CENTRAL_ADDRESS" scanner generate
```

## b. TLS シークレット YAML ファイルを適用します。

- OpenShift Container Platform を使用する場合は、以下のコマンドを入力します。

```
$ oc apply -f scanner-bundle/scanner/02-scanner-03-tls-secret.yaml
```

- Kubernetes を使用する場合は、次のコマンドを入力します。

```
$ kubectl apply -f scanner-bundle/scanner/02-scanner-03-tls-secret.yaml
```

## c. スキャナー設定 YAML ファイルを適用します。

- OpenShift Container Platform を使用する場合は、以下のコマンドを入力します。

```
$ oc apply -f scanner-bundle/scanner/02-scanner-04-scanner-config.yaml
```

- Kubernetes を使用する場合は、次のコマンドを入力します。

```
$ kubectl apply -f scanner-bundle/scanner/02-scanner-04-scanner-config.yaml
```

## 2. Scanner イメージを更新します。

- OpenShift Container Platform を使用する場合は、以下のコマンドを入力します。

```
$ oc -n stackrox set image deploy/scanner scanner=registry.redhat.io/advanced-cluster-security/rhacs-scanner-rhel8:3.73.5
```

- Kubernetes を使用する場合は、次のコマンドを入力します。

```
$ kubectl -n stackrox set image deploy/scanner scanner=registry.redhat.io/advanced-cluster-security/rhacs-scanner-rhel8:3.73.5
```

## 3. Scanner データベースイメージを更新します。

- OpenShift Container Platform を使用する場合は、以下のコマンドを入力します。

```
$ oc -n stackrox set image deploy/scanner-db db=registry.redhat.io/advanced-cluster-security/rhacs-scanner-db-rhel8:3.73.5 init-db=registry.redhat.io/advanced-cluster-security/rhacs-scanner-db-rhel8:3.73.5
```

- Kubernetes を使用する場合は、次のコマンドを入力します。

```
$ kubectl -n stackrox set image deploy/scanner-db db=registry.redhat.io/advanced-cluster-security/rhacs-scanner-db-rhel8:3.73.5 init-db=registry.redhat.io/advanced-cluster-security/rhacs-scanner-db-rhel8:3.73.5
```

## 検証

- 新しい Pod が正常にデプロイされたことを確認します。
  - OpenShift Container Platform を使用する場合は、以下のコマンドを入力します。

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

- Kubernetes を使用する場合は、次のコマンドを入力します。

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

### 3.3.3.1. RHACS バージョン 3.71 へのアップグレード

**roxctl** CLI および YAML ファイルを使用して RHACS 3.71 にアップグレードする場合は、いくつかの追加手順を実行する必要があります。Scanner DB イメージは、**scanner-db-password** Kubernetes Secret を **db** Scanner DB コンテナにマウントしなくなりました。代わりに、**scanner-db-password** は init コンテナ **init-db** でのみ使用されます。したがって、**POSTGRES\_PASSWORD\_FILE** 環境変数を init コンテナ設定に追加する必要があります。init コンテナは、**scanner-db-tls-volume** および **scanner-db-password** ボリュームもマウントする必要があります。以下のセクションでは、OpenShift Container Platform または Kubernetes を使用している場合の RHACS のアップグレード手順を説明します。init コンテナの詳細については、[Kubernetes のドキュメント](#) を参照してください。

#### 前提条件

- この手順では、Scanner DB 設定の **db** コンテナが **index 0** にあると想定している。これは、**containers** リストの最初のエントリーである。また、**scanner-db-password** ボリュームマウントは、3 番目のエントリーである **index 2** にある。

このシナリオはほとんどの展開に当てはまりますが、これらのコマンドを入力する前に Scanner DB の設定を確認してください。値が異なる場合は、次のコマンドで **.../containers/x/volumeMounts/y** の値を調整する必要があります。

#### 手順

1. パッチを適用します。

- OpenShift Container Platform を使用する場合は、以下のコマンドを入力します。

```
$ oc -n stackrox patch deployment.apps/scanner-db --patch '{"spec":{"template":{"spec":{"initContainers":[{"name":"init-db","env":[{"name":"POSTGRES_PASSWORD_FILE","value":"/run/secrets/stackrox.io/secrets/password"}],"command":["/usr/local/bin/docker-entrypoint.sh","postgres","-c","config_file=/etc/postgresql.conf"],"volumeMounts":[{"name":"db-data","mountPath":"/var/lib/postgresql/data"}, {"name":"scanner-db-tls-volume","mountPath":"/run/secrets/stackrox.io/certs","readOnly":true}, {"name":"scanner-db-password","mountPath":"/run/secrets/stackrox.io/secrets","readOnly":true}],"securityContext":{"runAsGroup":70,"runAsNonRoot":true,"runAsUser":70}}}}}}'
```

- Kubernetes を使用する場合は、次のコマンドを入力します。

```
$ kubectl -n stackrox patch deployment.apps/scanner-db --patch '{"spec":{"template":{"spec":{"initContainers":[{"name":"init-db","env":[{"name":"POSTGRES_PASSWORD_FILE","value":"/run/secrets/stackrox.io/secrets/password"}],"command":["/usr/local/bin/docker-entrypoint.sh","postgres","-c","config_file=/etc/postgresql.conf"],"volumeMounts":[{"name":"db-data","mountPath":"/var/lib/postgresql/data"}, {"name":"scanner-db-tls-volume","mountPath":"/run/secrets/stackrox.io/certs","readOnly":true}, {"name":"scanner-db-password","mountPath":"/run/secrets/stackrox.io/secrets","readOnly":true}],"securityContext":{"runAsGroup":70,"runAsNonRoot":true,"runAsUser":70}}}}}}'
```

## 2. パスを削除します。

- OpenShift Container Platform を使用する場合は、以下のコマンドを入力します。

```
$ oc -n stackrox patch deployment.apps/scanner-db --type json --patch
' [{"op": "remove", "path": "/spec/template/spec/containers/0/volumeMounts/2"} ]'
```

- Kubernetes を使用する場合は、次のコマンドを入力します。

```
$ kubectl -n stackrox patch deployment.apps/scanner-db --type json --patch
' [{"op": "remove", "path": "/spec/template/spec/containers/0/volumeMounts/2"} ]'
```

### 3.3.4. Central クラスターのアップグレードの確認

Central と Scanner の両方をアップグレードした後、Central クラスターのアップグレードが完了していることを確認します。

#### 手順

- Central ログを確認します。  
OpenShift Container Platform を使用している場合は、次のコマンドを入力します。

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

Kubernetes を使用している場合は、次のコマンドを入力します。

```
$ kubectl logs -n stackrox deploy/central -c central
```

#### 正常なアップグレードのサンプル出力

```
No database restore directory found (this is not an error).
Migrator: 2019/10/25 17:58:54: starting DB compaction
Migrator: 2019/10/25 17:58:54: Free fraction of 0.0391 (40960/1048576) is < 0.7500. Will not compact
badger 2019/10/25 17:58:54 INFO: All 1 tables opened in 2ms
badger 2019/10/25 17:58:55 INFO: Replaying file id: 0 at offset: 846357
badger 2019/10/25 17:58:55 INFO: Replay took: 50.324µs
badger 2019/10/25 17:58:55 DEBUG: Value log discard stats empty
Migrator: 2019/10/25 17:58:55: DB is up to date. Nothing to do here.
badger 2019/10/25 17:58:55 INFO: Got compaction priority: {level:0 score:1.73 dropPrefix:[]}
version: 2019/10/25 17:58:55.189866 ensure.go:49: Info: Version found in the DB was current. We're good to go!
```

## 3.4. すべてのセキュアなクラスターのアップグレード

Central サービスをアップグレードした後、すべてのセキュアなクラスターをアップグレードする必要があります。



## 重要

- 自動アップグレードを使用している場合は、以下を行います。
  - 自動アップグレードを使用して、保護されたすべてのクラスターを更新します。
  - このセクションの手順をスキップして、[アップグレードの確認](#) および [API トークンの取り消し](#) セクションの手順に従ってください。
- 自動アップグレードを使用していない場合は、Central クラスターを含むすべてのセキュアなクラスターでこのセクションの手順を実行する必要があります。
  - 最適な機能を確保するには、セキュアなクラスターと Central がインストールされているクラスターに同じ RHACS バージョンを使用してください。

Sensor、Collector、および Admission コントローラーを実行しているセキュリティーで保護された各クラスターの手動アップグレードを完了するには、このセクションの手順に従ってください。

### 3.4.1. ValidatingWebhookConfiguration の更新

以前の RHACS バージョンでは、ValidatingWebhookConfiguration に間違ったエントリーが含まれていました。これを修正するには、ValidatingWebhookConfiguration を更新する必要があります。

#### 手順

- アドミッションコントローラーで **listenOnEvents** を有効にしている場合は、次のコマンドを実行する必要があります。

```
$ oc patch validatingwebhookconfiguration stackrox -p '{"webhooks":[{"name": "k8sevents.stackrox.io", "rules": [{"apiGroups": ["*"], "apiVersions": ["*"], "operations": ["CONNECT"], "resources": ["pods", "pods/exec", "pods/portforward"]}]}]}' 1
```

- 1 Kubernetes を使用する場合は、**oc** の代わりに **kubectl** と入力します。

### 3.4.2. その他のイメージの更新

自動アップグレードを使用しない場合は、それぞれのセキュアなクラスターのセンサー、コレクター、コンプライアンスイメージを更新する必要があります。



## 注記

Kubernetes を使用している場合は、この手順にリストされているコマンドに **oc** の代わりに **kubectl** を使用してください。

#### 手順

1. Sensor イメージを更新します。

```
$ oc -n stackrox set image deploy/sensor sensor=registry.redhat.io/advanced-cluster-security/rhacs-main-rhel8:3.73.5 1
```

- 1 Kubernetes を使用する場合は、**oc** の代わりに **kubectl** と入力します。

## 2. Compliance イメージを更新します。

```
$ oc -n stackrox set image ds/collector compliance=registry.redhat.io/advanced-cluster-security/rhacs-main-rhel8:3.73.5 ❶
```

❶ Kubernetes を使用する場合は、**oc** の代わりに **kubectl** と入力します。

## 3. Collector イメージを更新します。

```
$ oc -n stackrox set image ds/collector collector=registry.redhat.io/advanced-cluster-security/rhacs-collector-rhel8:3.73.5 ❶
```

❶ Kubernetes を使用する場合は、**oc** の代わりに **kubectl** と入力します。



## 注記

コレクタースリムイメージを使用している場合は、代わりに次のコマンドを実行します。

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

## 4. アドミSSIONコントロールイメージを更新します。

```
$ oc -n stackrox set image deploy/admission-control admission-control=registry.redhat.io/advanced-cluster-security/rhacs-main-rhel8:3.73.5
```

## 3.4.3. セキュアなクラスターのアップグレードの確認

セキュアなクラスターをアップグレードしたら、更新された Pod が機能していることを確認します。

## 手順

- 新しい Pod がデプロイされていることを確認します。

```
$ oc get deploy,ds -n stackrox -o wide ❶
```

❶ Kubernetes を使用する場合は、**oc** の代わりに **kubectl** と入力します。

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

❶ Kubernetes を使用する場合は、**oc** の代わりに **kubectl** と入力します。

## 3.5. CENTRAL のロールバック

新しいバージョンへのアップグレードが失敗した場合は、以前のバージョンの Central にロールバックできます。

### 3.5.1. Central を通常どおりロールバック

Red Hat Advanced Cluster Security for Kubernetes のアップグレードが失敗した場合は、以前のバージョンの Central にロールバックできます。

#### 前提条件

- Red Hat Advanced Cluster Security for Kubernetes 3.0.57.0 以降を使用している必要がある。
- ロールバックを実行する前に、永続ストレージで使用可能な空きディスク容量が必要である。Red Hat Advanced Cluster Security for Kubernetes は、ディスク領域を使用して、アップグレード中にデータベースのコピーを保持する。ディスク容量がコピーを保存するのに十分でなく、アップグレードが失敗した場合は、以前のバージョンにロールバックすることはできない。

#### 手順

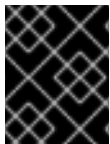
- アップグレードが失敗した場合 (Central サービスが開始する前) に、次のコマンドを実行して前のバージョンにロールバックします。

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

- 1** Kubernetes を使用する場合は、**oc** の代わりに **kubectl** と入力します。

### 3.5.2. Central を強制的にロールバックする

強制ロールバックを使用して、以前のバージョンの Central にロールバックできます (Central サービスの開始後)。



#### 重要

強制ロールバックを使用して以前のバージョンに戻すと、データと機能が失われる可能性があります。

#### 前提条件

- Red Hat Advanced Cluster Security for Kubernetes 3.0.58.0 以降を使用している必要がある。
- ロールバックを実行する前に、永続ストレージで使用可能な空きディスク容量が必要である。Red Hat Advanced Cluster Security for Kubernetes は、ディスク領域を使用して、アップグレード中にデータベースのコピーを保持する。ディスク容量がコピーを保存するのに十分でなく、アップグレードが失敗した場合は、以前のバージョンにロールバックすることはできない。

#### 手順

- 次のコマンドを実行して、強制ロールバックを実行します。
  - 以前にインストールしたバージョンに強制的にロールバックするには、以下のコマンドを実行します。

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



1 Kubernetes を使用する場合は、**oc** の代わりに **kubectl** と入力します。

○ 特定のバージョンに強制的にロールバックするには、以下を行います。

1. Central の **ConfigMap** を編集します。

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

1 Kubernetes を使用する場合は、**oc** の代わりに **kubectl** と入力します。

2. **maintenance.forceRollbackVersion** キーの値を更新します。

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

1 ロールバックするバージョンを指定します。

3. Central イメージのバージョンを更新します。

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

1 Kubernetes を使用する場合は、**oc** の代わりに **kubectl** と入力します。

2 ロールバックするバージョンを指定します。これは、**central-config** 設定マップで **maintenance.forceRollbackVersion** キーに指定したものと同一バージョンである必要があります。

### 3.6. アップグレードの確認

更新された Sensor と Collector は、それぞれのセキュアなクラスターからの最新データを引き続き報告します。

Sensor が Central に最後に接続した時刻は、RHACS ポータルに表示されます。

#### 手順

1. RHACS ポータルで、**Platform Configuration** → **System Health** に移動します。
2. Sensor Upgrade で、Central で最新のクラスターが表示されることを確認してください。

## 3.7. API トークンの取り消し

セキュリティ上の理由から、Red Hat では、Central データベースのバックアップを完了するために使用した API トークンを取り消すことが推奨されます。

### 前提条件

- アップグレード後、RHACS ポータルページをリロードし、証明書を再承認して、RHACS ポータルを引き続き使用する必要がある。

### 手順

1. RHACS ポータルで、**Platform Configuration → Integrations** に移動します。
2. **Authentication Tokens** カテゴリーまで下にスクロールし、**API Token** をクリックします。
3. 取り消すトークン名の前にあるチェックボックスを選択します。
4. **Revoke** をクリックします。
5. 確認ダイアログボックスで、**Confirm** をクリックします。