



# Red Hat Advanced Cluster Security for Kubernetes 3.73

## Upgrading

Upgrading Red Hat Advanced Cluster Security for Kubernetes



# Red Hat Advanced Cluster Security for Kubernetes 3.73 Upgrading

---

Upgrading Red Hat Advanced Cluster Security for Kubernetes

## Legal Notice

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.

## Abstract

This section provides instructions on upgrading Red Hat Advanced Cluster Security for Kubernetes by using Helm charts or the roxctl command-line interface.

## Table of Contents

<b>CHAPTER 1. UPGRADING BY USING THE OPERATOR</b>	<b>3</b>
1.1. ROLLING BACK AN OPERATOR UPGRADE BY USING THE CLI	3
1.2. ROLLING BACK AN OPERATOR UPGRADE BY USING THE WEB CONSOLE	5
1.3. ADDITIONAL RESOURCES	7
<b>CHAPTER 2. UPGRADING USING HELM CHARTS</b>	<b>8</b>
2.1. UPDATING THE HELM CHART REPOSITORY	8
2.2. ADDITIONAL RESOURCES	8
<b>CHAPTER 3. MANUALLY UPGRADING USING THE ROXCTL CLI</b>	<b>9</b>
3.1. SET UP THE ROX_SCANNER_DB_INIT ENVIRONMENT VARIABLE	9
3.2. BACKING UP THE CENTRAL DATABASE	9
3.3. UPGRADING THE CENTRAL CLUSTER	10
3.3.1. Upgrading Central	10
3.3.1.1. Upgrading Central on OpenShift Container Platform	10
3.3.1.2. Upgrading Central on Kubernetes	12
3.3.2. Upgrading the roxctl CLI	12
3.3.2.1. Uninstalling the roxctl CLI	13
3.3.2.2. Installing the roxctl CLI on Linux	13
3.3.2.3. Installing the roxctl CLI on macOS	13
3.3.2.4. Installing the roxctl CLI on Windows	14
3.3.3. Upgrading Scanner	14
3.3.3.1. Upgrading to RHACS version 3.71	16
3.3.4. Verifying the Central cluster upgrade	17
3.4. UPGRADING ALL SECURED CLUSTERS	17
3.4.1. Update ValidatingWebhookConfiguration	18
3.4.2. Updating other images	18
3.4.3. Verifying secured cluster upgrade	19
3.5. ROLLING BACK CENTRAL	19
3.5.1. Rolling back Central normally	19
3.5.2. Rolling back Central forcefully	20
3.6. VERIFYING UPGRADES	21
3.7. REVOKING THE API TOKEN	21



# CHAPTER 1. UPGRADING BY USING THE OPERATOR

Upgrades through the Red Hat Advanced Cluster Security for Kubernetes (RHACS) Operator are performed automatically or manually, depending on the **Update approval** option you chose at installation.

If you installed RHACS using the Operator and selected **Automatic** in the **Update approval** field, RHACS is automatically updated when a new software version is released. If you selected **Manual**, you must approve subsequent Operator updates by using Operator Lifecycle Manager (OLM). For more information, see [Manually approving a pending Operator update](#).

To roll back an Operator upgrade, you must perform the steps described in one of the following sections. You can roll back an Operator upgrade by using the CLI or the OpenShift Container Platform web console.

## 1.1. ROLLING BACK AN OPERATOR UPGRADE BY USING THE CLI

You can roll back the Operator version by using CLI commands.

### Procedure

1. Delete the OLM subscription by running the following command:

- For OpenShift Container Platform, run the following command:

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

- For Kubernetes, run the following command:

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

2. Delete the cluster service version (CSV) by running the following command:

- For OpenShift Container Platform, run the following command:

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

- For Kubernetes, run the following command:

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

3. Determine the previous version you want to roll back to by choosing one of the following options:

- If the current Central instance is running, query the RHACS API to get the rollback version by running the following command:

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

- If the current Central instance is not running, perform the following steps:

**NOTE**

This procedure can only be used for RHACS release 3.74 and earlier when the **rocksdb** database is installed.

- a. Ensure the Central deployment is scaled down by running the following command:

- For OpenShift Container Platform, run the following command:

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

- For Kubernetes, run the following command:

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

- b. Save the following pod spec as a YAML file:

```
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. Create a pod in your Central namespace by running the following command using the YAML file that you saved:

- For OpenShift Container Platform, run the following command:

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

- For Kubernetes, run the following command:

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

- d. After pod creation is complete, get the version by running the following command:

- For OpenShift Container Platform, run the following command:

```
-
```



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

- For Kubernetes, run the following command:

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

- Edit the **central-config.yaml ConfigMap** to set the **maintenance.forceRollBackVersion: <version>** parameter by running the following command:

- For OpenShift Container Platform, run the following command:

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

- For Kubernetes, run the following command:

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

- Set the image for the Central deployment using the version string shown in Step 3 as the image tag. For example, run the following command:

- For OpenShift Container Platform, run the following command:

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

- For Kubernetes, run the following command:

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

## Verification

- Ensure that the Central pod starts and has a **ready** status. If the pod crashes, check the logs to see if the backup was restored. A successful log message appears similar to the following example:

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

- Reinstall the Operator on the rolled back channel. For example, **3.71.3** is installed on the **rhacs-3.71** channel.

## 1.2. ROLLING BACK AN OPERATOR UPGRADE BY USING THE WEB CONSOLE

You can roll back the Operator version by using the OpenShift Container Platform web console.

### Prerequisites

- You have access to an OpenShift Container Platform cluster web console using an account with **cluster-admin** permissions.

## Procedure

1. Navigate to the **Operators → Installed Operators** page.
2. Locate the RHACS Operator and click on it.
3. On the **Operator Details** page, select **Uninstall Operator** from the **Actions** list. Following this action, the Operator stops running and no longer receives updates.
4. Determine the previous version you want to roll back to by choosing one of the following options:
  - If the current Central instance is running, you can query the RHACS API to get the rollback version by running the following command from a terminal window:

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

- You can create a pod and extract the previous version by performing the following steps:



### NOTE

This procedure can only be used for RHACS release 3.74 and earlier when the **rocksdb** database is installed.

- a. Navigate to **Workloads → Deployments → central**.
- b. Under **Deployment details**, click the down arrow next to the pod count to scale down the pod.
- c. Navigate to **Workloads → Pods → Create Pod** and paste the contents of the pod spec as shown in the following example into the editor:

```
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. Click **Create**.
  - e. After the pod is created, click the **Logs** tab to get the version string.
5. Update the rollback configuration by performing the following steps:
    - a. Navigate to **Workloads → ConfigMaps → central-config** and select **Edit ConfigMap** from the **Actions** list.
    - b. Find the **forceRollbackVersion** line in the value of the **central-config.yaml** key.
    - c. Replace **none** with **3.73.3**, and then save the file.
  6. Update Central to the earlier version by performing the following steps:
    - a. Navigate to **Workloads → Deployments → central** and select **Edit Deployment** from the **Actions** list.
    - b. Update the image name, and then save the changes.

## Verification

1. Ensure that the Central pod starts and has a **ready** status. If the pod crashes, check the logs to see if the backup was restored. A successful log message appears similar to the following example:

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

2. Reinstall the Operator on the rolled back channel. For example, **3.71.3** is installed on the **rhacs-3.71** channel.

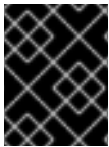
## 1.3. ADDITIONAL RESOURCES

- [Installing Central using the Operator method](#)
- [Operator Lifecycle Manager workflow](#)
- [Manually approving a pending Operator update](#)

## CHAPTER 2. UPGRADING USING HELM CHARTS

If you have installed Red Hat Advanced Cluster Security for Kubernetes by using Helm charts, to upgrade to the latest version of Red Hat Advanced Cluster Security for Kubernetes you must perform the following:

- Update the Helm chart.
- Update configuration files for the central-services Helm chart.
- Upgrade the central-services Helm chart.
- Update configuration files for the secured-cluster-services Helm chart.
- Upgrade the secured-cluster-services Helm chart.



### IMPORTANT

To ensure optimal functionality, use the same version for your secured-cluster-services Helm chart and central-services Helm chart.

### 2.1. UPDATING THE HELM CHART REPOSITORY

You must always update Helm charts before upgrading to a new version of Red Hat Advanced Cluster Security for Kubernetes.

#### Prerequisites

- You must have already added the Red Hat Advanced Cluster Security for Kubernetes Helm chart repository.

#### Procedure

- Update Red Hat Advanced Cluster Security for Kubernetes charts repository.

```
$ helm repo update
```

#### Verification

- Run the following command to verify the added chart repository:

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

### 2.2. ADDITIONAL RESOURCES

- [Installing Central using Helm charts](#)
- [Installing RHACS on secured clusters by using Helm charts](#)

## CHAPTER 3. MANUALLY UPGRADING USING THE ROXCTL CLI

You can upgrade to the latest version of Red Hat Advanced Cluster Security for Kubernetes (RHACS) from a supported older version.



### NOTE

You need to perform the manual upgrade procedure only if you used the **roxctl** CLI to deploy RHACS.

To upgrade RHACS to the latest version, you must perform the following:

- Set the **ROX\_SCANNER\_DB\_INIT** environment variable
- Backup the Central database
- Upgrade Central
- Upgrade the **roxctl** CLI
- Upgrade Scanner
- Verify that all secured clusters are upgraded

### 3.1. SET UP THE ROX\_SCANNER\_DB\_INIT ENVIRONMENT VARIABLE

ScannerDB's **initContainer** requires a new environment variable called **ROX\_SCANNER\_DB\_INIT**. You must set its value to **true** before you upgrade.

#### Procedure

- For OpenShift Container Platform, run the following command:

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

- For Kubernetes, run the following command:

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

### 3.2. BACKING UP THE CENTRAL DATABASE

You can back up the Central database and use that backup for rolling back from a failed upgrade or data restoration in the case of an infrastructure disaster.

#### Prerequisites

- You must have an API token with **read** permission for all resources of Red Hat Advanced Cluster Security for Kubernetes. The **Analyst** system role has **read** permissions for all resources.
- You have installed the **roxctl** CLI.

- You have configured the **ROX\_API\_TOKEN** and the **ROX\_CENTRAL\_ADDRESS** environment variables.

### Procedure

- Run the backup command:
  - For Red Hat Advanced Cluster Security for Kubernetes 3.0.55 and newer:

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

- For Red Hat Advanced Cluster Security for Kubernetes 3.0.54 and older:

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

### Additional resources

- [Authenticating using the \*\*roxctl\*\* CLI](#)

## 3.3. UPGRADING THE CENTRAL CLUSTER

After you have backed up the Central database, the next step is to upgrade the Central cluster. This step includes upgrading Central, the **roxctl** CLI, and the Scanner.

### 3.3.1. Upgrading Central

You can update Central to the latest version by downloading and deploying the updated images.

#### 3.3.1.1. Upgrading Central on OpenShift Container Platform

If you installed Red Hat Advanced Cluster Security for Kubernetes on OpenShift Container Platform, use the following procedure to upgrade.

### Procedure

1. Patch the local role:

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

2. Cleanup existing roles and role bindings:

```
$ 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. Set **sensor** and **admission-control** to **restricted[-v2]** security context constraints by removing the hard-coded security context:

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

- 1 Red Hat Advanced Cluster Security for Kubernetes recreates the pods automatically, however, **sensor** can take some time to restart.

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

4. Run the following commands to upgrade Central:

```
$ 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 If you deploy images from a private image registry, push the new image into your private registry, and replace the image registry address here.

## IMPORTANT

If you have not installed Red Hat Advanced Cluster Security for Kubernetes by using Helm or Operator, and you want to enable authentication using the OpenShift OAuth server, you must run the following additional commands:

```
$ 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"}}"  
    }  
  }  
'
```

## Verification

- Verify that the new pods have deployed:

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

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

### 3.3.1.2. Upgrading Central on Kubernetes

If you installed Red Hat Advanced Cluster Security for Kubernetes on Kubernetes, use the following procedure to upgrade.

#### Prerequisites

- If you deploy images from a private image registry, first push the new image into your private registry, and then replace your image registry in the following commands.

#### Procedure

1. Patch the local role:

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

2. Run the following commands to upgrade Central:

```
$ 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** If you deploy images from a private image registry, push the new image into your private registry, and replace the image registry address here.

#### Verification

- Verify that the new pods have deployed:

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

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

### 3.3.2. Upgrading the roxctl CLI

To upgrade the **roxctl** CLI to the latest version you must uninstall the existing version of **roxctl** CLI and then install the latest version of the **roxctl** CLI.



### 3.3.2.1. Uninstalling the roxctl CLI

You can uninstall the **roxctl** CLI binary on Linux by using the following procedure.

#### Procedure

- Find and delete the **roxctl** binary:

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

- 1** Depending on your environment, you might need administrator rights to delete the **roxctl** binary.

### 3.3.2.2. Installing the roxctl CLI on Linux

You can install the **roxctl** CLI binary on Linux by using the following procedure.

#### Procedure

- Download the latest version of the **roxctl** CLI:

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

- Make the **roxctl** binary executable:

```
$ chmod +x roxctl
```

- Place the **roxctl** binary in a directory that is on your **PATH**:  
To check your **PATH**, execute the following command:

```
$ echo $PATH
```

#### Verification

- Verify the **roxctl** version you have installed:

```
$ roxctl version
```

### 3.3.2.3. Installing the roxctl CLI on macOS

You can install the **roxctl** CLI binary on macOS by using the following procedure.

#### Procedure

- Download the latest version of the **roxctl** CLI:

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

- Remove all extended attributes from the binary:

```
$ xattr -c roxctl
```

3. Make the **roxctl** binary executable:

```
$ chmod +x roxctl
```

4. Place the **roxctl** binary in a directory that is on your **PATH**:  
To check your **PATH**, execute the following command:

```
$ echo $PATH
```

### Verification

- Verify the **roxctl** version you have installed:

```
$ roxctl version
```

### 3.3.2.4. Installing the roxctl CLI on Windows

You can install the **roxctl** CLI binary on Windows by using the following procedure.

#### Procedure

- Download the latest version of the **roxctl** CLI:

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

### Verification

- Verify the **roxctl** version you have installed:

```
$ roxctl version
```

After you upgrade the **roxctl** CLI you can upgrade Scanner.

### 3.3.3. Upgrading Scanner

You can update Scanner to the latest version by using the **roxctl** CLI.

#### Prerequisites

- If you deploy images from a private image registry, you must first push the new image into your private registry, then edit the commands in the following section to use the name of your private image registry.

#### Procedure

1. If you have created custom scanner configurations, you must apply those changes before updating the scanner configuration file.
  - a. Generate Scanner using the following **roxctl** command:

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

## b. Apply the TLS secrets YAML file:

- If you use OpenShift Container Platform, enter the following command:

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

- If you use Kubernetes, enter the following command:

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

## c. Apply the Scanner configuration YAML file:

- If you use OpenShift Container Platform, enter the following command:

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

- If you use Kubernetes, enter the following command:

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

## 2. Update the Scanner image:

- If you use OpenShift Container Platform, enter the following command:

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

- If you use Kubernetes, enter the following command:

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

## 3. Update the Scanner database image:

- If you use OpenShift Container Platform, enter the following command:

```
$ 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
```

- If you use Kubernetes, enter the following command:

```
$ 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
```

**Verification**

- Check that the new pods have deployed successfully:
  - If you use OpenShift Container Platform, enter the following command:

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

- If you use Kubernetes, enter the following command:

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

### 3.3.3.1. Upgrading to RHACS version 3.71

If you are upgrading to RHACS 3.71 using the **roxctl** CLI and YAML files, you need to perform some additional steps. The Scanner DB image no longer mounts the **scanner-db-password** Kubernetes Secret into the **db** Scanner DB container. Instead, **scanner-db-password** is only used in the init container, **init-db**. Therefore, you must add the **POSTGRES\_PASSWORD\_FILE** environment variable to the init container configuration. The init container must also mount the **scanner-db-tls-volume** and **scanner-db-password** volumes. The following section provides the upgrade steps for RHACS if you are using OpenShift Container Platform or Kubernetes. For more information about init containers, see the [Kubernetes documentation](#).

#### Prerequisites

- This procedure assumes the **db** container in the Scanner DB configuration is at **index 0**, which is the first entry in the **containers** list; and the **scanner-db-password** volume mount is at **index 2**, which is the third entry.

While this scenario applies to most deployments, check the configuration for Scanner DB before entering these commands. If your values differ, you must adjust the **.../containers/x/volumeMounts/y** value in the following commands.

#### Procedure

1. Apply the patch:

- If you use OpenShift Container Platform, enter the following command:

```
$ 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}}}}}}'
```

- If you use Kubernetes, enter the following command:

```
$ 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. Remove the path:

- If you use OpenShift Container Platform, enter the following command:

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

- If you use Kubernetes, enter the following command:

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

### 3.3.4. Verifying the Central cluster upgrade

After you have upgraded both Central and Scanner, verify that the Central cluster upgrade is complete.

#### Procedure

- Check the Central logs:

If you are using OpenShift Container Platform, enter the following command:

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

If you are using Kubernetes, enter the following command:

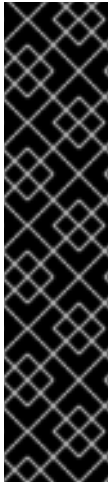
```
$ kubectl logs -n stackrox deploy/central -c central
```

#### Sample output of a successful upgrade

```
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. UPGRADING ALL SECURED CLUSTERS

After upgrading Central services, you must upgrade all secured clusters.



## IMPORTANT

- If you are using automatic upgrades:
  - Update all your secured clusters by using automatic upgrades.
  - Skip the instructions in this section and follow the instructions in the [Verify upgrades](#) and [Revoking the API token](#) sections.
- If you are not using automatic upgrades, you must run the instructions in this section on all secured clusters including the Central cluster.
  - To ensure optimal functionality, use the same RHACS version for your secured clusters and the cluster on which Central is installed.

To complete manual upgrades of each secured cluster running Sensor, Collector, and Admission controller, follow the instructions in this section.

### 3.4.1. Update ValidatingWebhookConfiguration

Earlier RHACS versions included a wrong entry in the ValidatingWebhookConfiguration. To fix it, you must update the ValidatingWebhookConfiguration.

#### Procedure

- If you have enabled **listenOnEvents** in your Admission controller, you must run the following command:

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

- 1** If you use Kubernetes, enter **kubectl** instead of **oc**.

### 3.4.2. Updating other images

You must update the sensor, collector and compliance images on each secured cluster when not using automatic upgrades.



## NOTE

If you are using Kubernetes, use **kubectl** instead of **oc** for the commands listed in this procedure.

#### Procedure

1. Update the Sensor image:

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

- 1** If you use Kubernetes, enter **kubectl** instead of **oc**.

## 2. Update the Compliance image:

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

**1** If you use Kubernetes, enter **kubectl** instead of **oc**.

## 3. Update the Collector image:

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

**1** If you use Kubernetes, enter **kubectl** instead of **oc**.

**NOTE**

If you are using the collector slim image, run the following command instead:

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

## 4. Update the admission control image:

```
$ 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. Verifying secured cluster upgrade**

After you have upgraded secured clusters, verify that the updated pods are working.

**Procedure**

- Check that the new pods have deployed:

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

**1** If you use Kubernetes, enter **kubectl** instead of **oc**.

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

**1** If you use Kubernetes, enter **kubectl** instead of **oc**.

**3.5. ROLLING BACK CENTRAL**

You can roll back to a previous version of Central if the upgrade to a new version is unsuccessful.

**3.5.1. Rolling back Central normally**

You can roll back to a previous version of Central if upgrading Red Hat Advanced Cluster Security for Kubernetes fails.

### Prerequisites

- You must be using Red Hat Advanced Cluster Security for Kubernetes 3.0.57.0 or higher.
- Before you can perform a rollback, you must have free disk space available on your persistent storage. Red Hat Advanced Cluster Security for Kubernetes uses disk space to keep a copy of databases during the upgrade. If the disk space is not enough to store a copy and the upgrade fails, you will not be able to roll back to an earlier version.

### Procedure

- Run the following command to roll back to a previous version when an upgrade fails (before the Central service starts):

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

- 1 If you use Kubernetes, enter **kubectl** instead of **oc**.

### 3.5.2. Rolling back Central forcefully

You can use forced rollback to roll back to an earlier version of Central (after the Central service starts).



#### IMPORTANT

Using forced rollback to switch back to a previous version might result in loss of data and functionality.

### Prerequisites

- You must be using Red Hat Advanced Cluster Security for Kubernetes 3.0.58.0 or higher.
- Before you can perform a rollback, you must have free disk space available on your persistent storage. Red Hat Advanced Cluster Security for Kubernetes uses disk space to keep a copy of databases during the upgrade. If the disk space is not enough to store a copy and the upgrade fails, you will not be able to roll back to an earlier version.

### Procedure

- Run the following commands to perform a forced rollback:
  - To forcefully rollback to the previously installed version:

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

- 1 If you use Kubernetes, enter **kubectl** instead of **oc**.

- To forcefully rollback to a specific version:

1. Edit Central's **ConfigMap**:

■



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

1 If you use Kubernetes, enter **kubectl** instead of **oc**.

2. Update the value of the **maintenance.forceRollbackVersion** key:

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

1 Specify the version that you want to roll back to.

3. Update the Central image version:

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

1 If you use Kubernetes, enter **kubectl** instead of **oc**.

2 Specify the version that you want to roll back to. It must be the same version that you specified for the **maintenance.forceRollbackVersion** key in the **central-config** config map.

## 3.6. VERIFYING UPGRADES

The updated Sensors and Collectors continue to report the latest data from each secured cluster.

The last time Sensor contacted Central is visible in the RHACS portal.

### Procedure

1. On the RHACS portal, navigate to **Platform Configuration → System Health**.
2. Check to ensure that Sensor Upgrade shows clusters up to date with Central.

## 3.7. REVOKING THE API TOKEN

For security reasons, Red Hat recommends that you revoke the API token that you have used to complete Central database backup.

### Prerequisites

- After the upgrade, you must reload the RHACS portal page and re-accept the certificate to continue using the RHACS portal.

## Procedure

1. On the RHACS portal, navigate to **Platform Configuration → Integrations**.
2. Scroll down to the **Authentication Tokens** category, and click **API Token**.
3. Select the checkbox in front of the token name that you want to revoke.
4. Click **Revoke**.
5. On the confirmation dialog box, click **Confirm**.