Red Hat Advanced Cluster Security for Kubernetes 3.71

Upgrading

Upgrading Red Hat Advanced Cluster Security for Kubernetes

Last Updated: 2023-02-22

Red Hat Advanced Cluster Security for Kubernetes 3.71 Upgrading

Upgrading Red Hat Advanced Cluster Security for Kubernetes

Legal Notice

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

CHAPTER 1. 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.

1.1. 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/
```

1.2. ADDITIONAL RESOURCES

- Configuring the central-services Helm chart

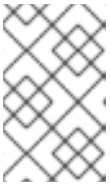
1.3. CHANGING CONFIGURATION OPTIONS AFTER DEPLOYING THE CENTRAL-SERVICES HELM CHART

You can make changes to any configuration options after you have deployed the **central-services** Helm chart.

Procedure

1. Update the **values-public.yaml** and **values-private.yaml** configuration files with new values.
2. Run the **helm upgrade** command and specify the configuration files using the **-f** option:

```
$ helm upgrade -n stackrox \
  stackrox-central-services rhacs/central-services \
  -f <path_to_values_public.yaml> \
  -f <path_to_values_private.yaml>
```



NOTE

You can also specify configuration values using the **--set** or **--set-file** parameters. However, these options are not saved, and it requires you to manually specify all the options again whenever you make changes.

1.4. CHANGING CONFIGURATION OPTIONS AFTER DEPLOYING THE SECURED-CLUSTER-SERVICES HELM CHART

You can make changes to any configuration options after you have deployed the **secured-cluster-services** Helm chart.

Procedure

1. Update the **values-public.yaml** and **values-private.yaml** configuration files with new values.
2. Run the **helm upgrade** command and specify the configuration files using the **-f** option:

```
$ helm upgrade -n stackrox \
  stackrox-secured-cluster-services rhacs/secured-cluster-services \
  --reuse-values \ 1
  -f <path_to_values_public.yaml> \
  -f <path_to_values_private.yaml>
```

1. You must specify the **--reuse-values** parameter, otherwise the Helm upgrade command resets all previously configured settings.

NOTE

You can also specify configuration values using the **--set** or **--set-file** parameters. However, these options are not saved, and it requires you to manually specify all the options again whenever you make changes.

CHAPTER 2. MANUALLY UPGRADING USING THE ROXCTL CLI

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

To upgrade Red Hat Advanced Cluster Security for Kubernetes to the latest version, you must perform the following:

- Backup the Central database
- Upgrade Central
- Upgrade the **roxctl** CLI
- Upgrade Scanner
- Verify that all secured clusters are upgraded

2.1. 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

2.2. 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.

2.2.1. Upgrading Central

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

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 for the commands in this section.

Procedure

- 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"}}}]}]}}}}' 1
```

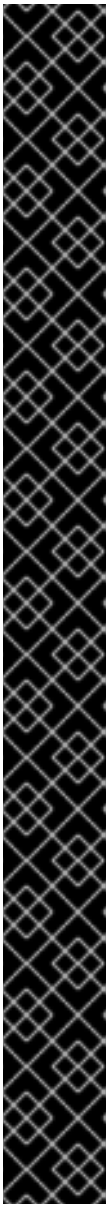
1 If you use Kubernetes, enter **kubectl** instead of **oc**.

```
$ oc -n stackrox patch deployment/scanner -p '{"spec":{"template":{"spec":{"containers":
[{"name":"scanner","securityContext":{"runAsUser":65534}}]}}}}' 1
```

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

```
$ oc -n stackrox set image deploy/central central=registry.redhat.io/advanced-cluster-
security/rhacs-main-rhel8:3.71.3 1
```

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



IMPORTANT

- If you are upgrading from Red Hat Advanced Cluster Security for Kubernetes 3.65.0, you must run the following additional command to create the **stackrox-central-diagnostics** role:

```
$ oc -n stackrox patch role stackrox-central-diagnostics -p '{"rules":
[{"apiGroups":["*"],"resources":
["deployments","daemonsets","replicasets","configmaps","services"],"verbs":
["get","list"]}]}' 1
```

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

- 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 command:

```
$ 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

- Check that the new pods have deployed:

```
$ oc get deploy -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**.

2.2.2. 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.

2.2.2.1. 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.

2.2.2.2. Installing the roxctl CLI on Linux

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

Procedure

1. Download the latest version of the **roxctl** CLI:

```
$ curl -O https://mirror.openshift.com/pub/rhacs/assets/3.71.3/bin/Linux/roxctl
```

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

```
$ chmod +x roxctl
```

3. 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
```

2.2.2.3. Installing the roxctl CLI on macOS

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

Procedure

1. Download the latest version of the **roxctl** CLI:

```
$ curl -O https://mirror.openshift.com/pub/rhacs/assets/3.71.3/bin/Darwin/roxctl
```

2. 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
```

2.2.2.4. 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/3.71.3/bin/Windows/roxctl.exe
```

Verification

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

```
$ roxctl version
```

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

2.2.3. 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-rhel8:3.71.3
```

- If you use Kubernetes, enter the following command:

```
$ kubectl -n stackrox set image deploy/scanner scanner=registry.redhat.io/advanced-cluster-security/rhacs-scanner-rhel8:3.71.3
```

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-rhel8:3.71.3 init-db=registry.redhat.io/advanced-cluster-security/rhacs-scanner-db-rhel8:3.71.3
```

- 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-rhel8:3.71.3 init-db=registry.redhat.io/advanced-cluster-security/rhacs-scanner-db-rhel8:3.71.3
```

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
```

2.2.3.1. 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/pass
word"}],"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}],"securityContex
t":{"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/pass
word"}],"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}],"securityContex
t":{"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"}]'
     ```

2.2.4. 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:

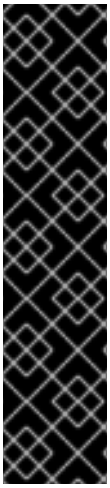
  ```
  $ kubectl logs -n stackrox deploy/central -c central
  ```

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!
```

2.3. 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.

2.3.1. Update readiness probes

If you are upgrading from a version below Red Hat Advanced Cluster Security for Kubernetes 3.65.0, you must run the following additional command to update the readiness probe path. If you are running a higher version than 3.65, skip this step.

Procedure

- Update the readiness probe path:

```
$ oc -n stackrox patch deploy/sensor -p '{"spec":{"template":{"spec":{"containers":
[{"name":"sensor","readinessProbe":{"httpGet":{"path":"/ready"}}}]}}}}' 1
```

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

2.3.2. Updating OpenShift security context constraints

Depending on the version of Red Hat Advanced Cluster Security for Kubernetes you are upgrading to, you must update certain OpenShift Container Platform security context constraints (SCCs).

WARNING

Run the commands in this section only if you are using Red Hat Advanced Cluster Security for Kubernetes with OpenShift Container Platform. Otherwise, skip the instructions in this section.

Procedure

- Red Hat Advanced Cluster Security for Kubernetes 3.64.0 renames the SCCs. If you are upgrading from a version below Red Hat Advanced Cluster Security for Kubernetes 3.64.0, you must delete and reapply the SCCs, otherwise, skip this step:

a. Run the following commands to update Central:

```
$ oc apply -f - <<EOF
kind: SecurityContextConstraints
apiVersion: security.openshift.io/v1
metadata:
  name: stackrox-central
  labels:
    app.kubernetes.io/name: stackrox
  annotations:
    kubernetes.io/description: stackrox-central is the security constraint for the central
server
    email: support@stackrox.com
    owner: stackrox
allowHostDirVolumePlugin: false
allowedCapabilities: []
allowHostIPC: false
allowHostNetwork: false
allowHostPID: false
allowHostPorts: false
allowPrivilegeEscalation: false
allowPrivilegedContainer: false
defaultAddCapabilities: []
fsGroup:
  type: MustRunAs
  ranges:
    - max: 4000
      min: 4000
priority: 0
readOnlyRootFilesystem: true
requiredDropCapabilities: []
runAsUser:
  type: MustRunAs
  uid: 4000
seLinuxContext:
  type: MustRunAs
seccompProfiles:
  - '*'
users:
  - system:serviceaccount:stackrox:central
volumes:
  - '*'
EOF
```

```
$ oc delete scc central
```

b. Run the following commands to update Scanner:

```
$ oc apply -f - <<EOF
kind: SecurityContextConstraints
apiVersion: security.openshift.io/v1
metadata:
  name: stackrox-scanner
  labels:
    app.kubernetes.io/name: stackrox
```

```
  annotations:
    email: support@stackrox.com
    owner: stackrox
    kubernetes.io/description: stackrox-scanner is the security constraint for the Scanner
container
priority: 0
runAsUser:
  type: RunAsAny
seLinuxContext:
  type: RunAsAny
seccompProfiles:
  - '*'
users:
  - system:serviceaccount:stackrox:scanner
volumes:
  - '*'
allowHostDirVolumePlugin: false
allowedCapabilities: []
allowHostIPC: false
allowHostNetwork: false
allowHostPID: false
allowHostPorts: false
allowPrivilegeEscalation: false
allowPrivilegedContainer: false
defaultAddCapabilities: []
fsGroup:
  type: RunAsAny
readOnlyRootFilesystem: false
requiredDropCapabilities: []
EOF
```

```
$ oc delete scc scanner
```

c. Run the following commands on each OpenShift Secured Cluster:

```
$ oc apply -f - <<EOF
apiVersion: security.openshift.io/v1
kind: SecurityContextConstraints
metadata:
  name: stackrox-admission-control
  labels:
    app.kubernetes.io/name: stackrox
    auto-upgrade.stackrox.io/component: "sensor"
  annotations:
    email: support@stackrox.com
    owner: stackrox
    kubernetes.io/description: stackrox-admission-control is the security constraint for the
admission controller
users:
  - system:serviceaccount:stackrox:admission-control
priority: 0
runAsUser:
  type: RunAsAny
seLinuxContext:
  type: RunAsAny
```

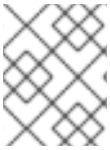
```
seccompProfiles:
  - '*'
supplementalGroups:
  type: RunAsAny
fsGroup:
  type: RunAsAny
groups: []
readOnlyRootFilesystem: true
allowHostDirVolumePlugin: false
allowHostIPC: false
allowHostNetwork: false
allowHostPID: false
allowHostPorts: false
allowPrivilegeEscalation: false
allowPrivilegedContainer: false
allowedCapabilities: []
defaultAddCapabilities: []
requiredDropCapabilities: []
volumes:
  - configMap
  - downwardAPI
  - emptyDir
  - secret
---
apiVersion: security.openshift.io/v1
kind: SecurityContextConstraints
metadata:
  name: stackrox-collector
  labels:
    app.kubernetes.io/name: stackrox
    auto-upgrade.stackrox.io/component: "sensor"
  annotations:
    email: support@stackrox.com
    owner: stackrox
    kubernetes.io/description: This SCC is based on privileged, hostaccess, and
hostmount-anyuid
users:
  - system:serviceaccount:stackrox:collector
allowHostDirVolumePlugin: true
allowPrivilegedContainer: true
fsGroup:
  type: RunAsAny
groups: []
priority: 0
readOnlyRootFilesystem: true
runAsUser:
  type: RunAsAny
seLinuxContext:
  type: RunAsAny
seccompProfiles:
  - '*'
supplementalGroups:
  type: RunAsAny
allowHostIPC: false
allowHostNetwork: false
allowHostPID: false
```

```
allowHostPorts: false
allowPrivilegeEscalation: true
allowedCapabilities: []
defaultAddCapabilities: []
requiredDropCapabilities: []
volumes:
  - configMap
  - downwardAPI
  - emptyDir
  - hostPath
  - secret
---
apiVersion: security.openshift.io/v1
kind: SecurityContextConstraints
metadata:
  name: stackrox-sensor
  labels:
    app.kubernetes.io/name: stackrox
    auto-upgrade.stackrox.io/component: "sensor"
  annotations:
    email: support@stackrox.com
    owner: stackrox
    kubernetes.io/description: stackrox-sensor is the security constraint for the sensor
users:
  - system:serviceaccount:stackrox:sensor
  - system:serviceaccount:stackrox:sensor-upgrader
priority: 0
runAsUser:
  type: RunAsAny
seLinuxContext:
  type: RunAsAny
seccompProfiles:
  - '*'
supplementalGroups:
  type: RunAsAny
fsGroup:
  type: RunAsAny
groups: []
readOnlyRootFilesystem: true
allowHostDirVolumePlugin: false
allowHostIPC: false
allowHostNetwork: false
allowHostPID: false
allowHostPorts: false
allowPrivilegeEscalation: true
allowPrivilegedContainer: false
allowedCapabilities: []
defaultAddCapabilities: []
requiredDropCapabilities: []
volumes:
  - configMap
  - downwardAPI
  - emptyDir
  - secret
EOF
```

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

2.3.3. 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-rhel8:3.71.3 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-rhel8:3.71.3 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-rhel8:3.71.3 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-rhel8:3.71.3
```

2.3.4. 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**.

2.4. ROLLING BACK CENTRAL

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

2.4.1. 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**.

2.4.2. 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.

using config map.

2.5. 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.

2.6. 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**.