Red Hat Advanced Cluster Security for Kubernetes 4.4

Release notes

Highlights what is new and what has changed with Red Hat Advanced Cluster Security for Kubernetes releases

Last Updated: 2024-04-22

Red Hat Advanced Cluster Security for Kubernetes 4.4 Release notes

Highlights what is new and what has changed with Red Hat Advanced Cluster Security for Kubernetes releases

Legal Notice

Abstract

The release notes for Red Hat Advanced Cluster Security for Kubernetes summarize all new features and enhancements, notable technical changes, deprecated and removed features, bug fixes, and any known bugs upon general availability.

Table of Contents

CHAPTER 1. RED HAT ADVANCED CLUSTER SECURITY FOR KUBERNETES 4.4

Red Hat Advanced Cluster Security for Kubernetes (RHACS) is an enterprise-ready, Kubernetes-native container security solution that protects your vital applications across the build, deploy, and runtime stages of the application lifecycle. Red Hat Advanced Cluster Security for Kubernetes deploys into your infrastructure and integrates with your DevOps tools and workflows. This integration provides better security and compliance, enabling DevOps and InfoSec teams to operationalize security.

Table 1.1. Release dates

RHACS version	Released on
4.4.0	28 March 2024

1.1. ABOUT THIS RELEASE

RHACS 4.4 includes the following new features, improvements, and updates:

Compliance

- New Compliance capabilities (Technology Preview)

Network

- Network graph enhancements for internal entities
- Build-time network policy tools is now generally available

Platform

- Init-bundle graphical user interface improvements
- eBPF CO-RE collection method enabled by default
- Bring your own database for RHACS Central is now generally available
- Support RHACS on ROSA hosted control plane
- Life cycle updates
- Integration with Red Hat OpenShift Cluster Manager and Paladin Cloud to discover unsecured clusters
- Migration to stock Red Hat OpenShift SCCs during manual upgrade by using roxctl CLI
- Cluster discovery by using cloud source integrations
- Short-lived API tokens for Central

Policy

- Enhanced roxctl deployment check command

- Authentication of AWS and GCP integrations by using short-lived tokens (Technology Preview)

Vulnerability Management

- Scanner V4 that uses upstream ClairCore (Technology Preview)
- Filter workload CVEs by using component and component source

1.2. NEW FEATURES

This release adds improvements related to the following components and concepts:

1.2.1. Scanner V4 that uses upstream ClairCore (Technology Preview)

IMPORTANT

Scanner V4 is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

In RHACS 4.4, the new Scanner V4 integrates the best features of the existing StackRox Scanner and the upstream Clair V4 Scanner that ships with Red Hat Quay, providing enhanced functionality.

The following are the key highlights of the new Scanner V4:

- **Consistent and accurate scanning**: Reliable vulnerability scan results across the entire Red Hat product ecosystem, Red Hat RHACS, and Red Hat Quay.
- **Expanded language and operating system support** Expanded support for Golang in language vulnerability scanning and inclusion of Oracle Linux, SUSE Linux Enterprise, and Photon OS in operating system scanning.
- **Comprehensive vulnerability database source**: Adoption of **OSV.dev** as the vulnerability database source for all supported programming language packages. The RHACS Scanner V4 uses the OSV database available at OSV.dev under this license.

NOTE

- In RHACS 4.4, both the StackRox Scanner and the new Scanner V4 are available for scanning workloads, and the existing StackRox Scanner reports node and platform vulnerabilities.
- RHACS upgrades and new installations use the StackRox Scanner by default.
- You now have the option to enable the new Scanner V4 in addition to the default StackRox Scanner. Scanner V4 is specifically designed for scanning images, while StackRox Scanner is still required for scanning nodes and platforms. This offers you additional benefits and an extended scope for securing your environment.
- Red Hat plans to make the new Scanner V4 the default Scanner in a future release.

For general information about Scanner V4, see About RHACS Scanner V4.

For more information about enabling Scanner V4 when installing RHACS, see:

- "StackRox Scanner settings" in Installing RHACS on Red Hat OpenShift
- "Scanner V4" in Installing RHACS on other platforms

Scanner V4 has additional memory and storage requirements in addition to the memory and storage needed to run the default StackRox Scanner.

For more information about the resource requirements for the StackRox Scanner and Scanner V4, see:

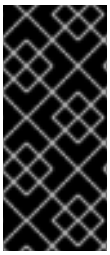
- "General requirements" in Default resource requirements for Red Hat Advanced Cluster Security for Kubernetes
- Recommended resource requirements for Red Hat Advanced Cluster Security for Kubernetes

1.2.2. New Compliance capabilities (Technology Preview)

IMPORTANT

Compliance 2.0 is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .



IMPORTANT

To use the new Compliance capabilities, you must have installed the Compliance Operator.

For more information, see Using the Compliance Operator with Red Hat Advanced Cluster Security for Kubernetes.

RHACS 4.4 offers a seamless experience that combines Compliance Operator and RHACS. You can now configure, schedule, and run infrastructure scans and review the results from RHACS.

In future releases, Red Hat plans to offer the following additional capabilities:

- Remediate deficiencies and export the results of compliance scans from the RHACS dashboard.
- Create custom profiles.
- Support workload compliance.

1.2.3. Build-time network policy tools is now generally available

RHACS 4.4 offers the following tools to help you develop Kubernetes network policies in build-time:

- **roxctl netpol generate** - Generates Kubernetes network policies that meet your application's requirements by analyzing your project's YAML manifests in a local directory.
- **roxctl netpol connectivity map** - Lists the connections allowed by the network policies in your project.
- **roxctl netpol connectivity diff** - Lists the differences in allowed connections between two project versions.

You can access the build-time network policy tools by using the **roxctl** CLI.

For more information, see Build-time network policy tools.

1.2.4. Network graph enhancements for internal entities

In RHACS 4.4, selected connections to private IP addresses within the cluster, which were previously incorrectly identified as external entities, are now shown as internal entities in the Network graph.

This fixes connections that are incorrectly identified as external, for example, when the following scenarios occur:

- The type of a service has been changed to or from a cluster IP address.
- A deployment is restarted and receives a new pod IP address, causing the other communicating party to attempt to reach the old IP address.
- A container attempts to communicate with a locally linked IP address.

For more information, see Network graph.

1.2.5. Init-bundle graphical user interface improvements

RHACS 4.4 now provides an easier way to add secured clusters and manage them in one place in the RHACS portal. This release also offers new guidance to help you when creating init bundles.

For more information, see the following updated documentation:

RHACS Cloud Service

- Generating and applying init bundles on Red Hat OpenShift
- Generating and applying init bundles on Kubernetes

RHACS

- Generating and applying an init bundle for RHACS on Red Hat OpenShift
- Generating and applying an init bundle for RHACS on other platforms

1.2.6. Integration with Red Hat OpenShift Cluster Manager and Paladin Cloud to discover unsecured clusters

In RHACS 4.4, you can now discover new clusters not protected by RHACS through integration with Red Hat OpenShift Cluster Manager and Paladin Cloud. This feature provides a list of clusters in your OpenShift environment or on cloud platforms such as Amazon Elastic Kubernetes Service (Amazon EKS), Google Kubernetes Engine (Google GKE) and Microsoft Azure Kubernetes Service (Microsoft AKS) and provides a discovery mechanism to improve your organization's security.

Paladin Cloud currently offers a free trial to help you discover how Paladin Cloud works with RHACS to secure your Red Hat OpenShift and Kubernetes deployments in the cloud.

For more information about the free trial version, visit Paladin Cloud Free Trial.

1.2.7. eBPF CO-RE collection method enabled by default

RHACS 4.4 introduces a default runtime collection method based on eBPF compile once-run everywhere (CO-RE).

This method becomes the default starting with this release unless explicitly overridden in your secured cluster's configuration.

The CO-RE approach is a modern method for creating portable eBPF applications that ensure compatibility across kernel versions and configurations without requiring changes or compilation of runtime source code on the target machine. When you upgrade to the new version, the migration from eBPF to eBPF CO-RE is automatic.

The following are some of the advantages of this new collection method:

- You can run RHACS-secured clusters on a larger number of Linux operating systems that are not currently included in the support package.
- You no longer need to update the Collector support packages if you are offline. By avoiding issues related to eBPF probe retrieval, such as the risk of losing the network connection or if the probe is not present, you can update a cluster smoothly.

1.2.8. The CORE BPF collection method is now generally available for ppc64le architecture

RHACS 4.2 included the runtime collection method based on BPF CO-RE (Compile Once-Run Everywhere), which is available on the **x86_64** and **s390x** architectures. Starting with RHACS 4.4, BPF CO-RE is now generally available on the **ppc64le** architecture. To enable it, set the value of your secured cluster's **collector.collectionMethod** parameter to **CORE_BPF**.

1.2.9. Bring your own database for RHACS Central is now generally available

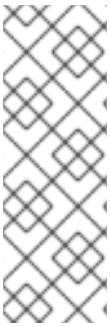
With RHACS 4.4, you can now use your own PostgreSQL-compatible database with Central, enabling you to deploy it either within or outside the cluster. Whether on bare metal, virtual machines, or as a

cloud-hosted service, you can tailor the deployment to your specific requirements. To ensure optimal performance, you must run the database in close proximity to the RHACS Central services.

For more information, see Installing Central with an external database using the Operator method .

1.2.10. Support RHACS on ROSA hosted control plane

In RHACS 4.4, you can install and run RHACS on a Red Hat OpenShift Service on AWS (ROSA) with a hosted control plane (HCP) cluster. Both Central and secured cluster services are supported on an HCP cluster.



NOTE

If you use HCP clusters, access to the primary nodes where the Kubernetes API server audit logs reside is restricted. Therefore, RHACS runtime policies that rely on events from the Kubernetes audit log are not supported on HCP clusters.

It means that RHACS cannot inspect how the API is being used, for example, to modify sensitive resources such as **ConfigMaps**, **Secrets**, **SecurityContextConstraints** (SCCs), **ClusterRoles**, and others.

1.2.11. Enhanced roxctl deployment check command

RHACS 4.4 offers an enhanced **roxctl deployment check** command with the introduction of the **--cluster** and **--namespace** options. These commands can be used with a new **--verbose** flag. By enabling the **--verbose** flag, you receive additional information for each deployment during the policy check. The extended information includes the role-based access control (RBAC) permission level and a comprehensive list of network policies that are applied.

You can now specify the YAML file with one or more deployments to send to Central for policy evaluation by using the **--file** flag. You can also specify multiple YAML files to send to Central for policy evaluation separated by spaces, for example, **--file=<yaml_filename1>**, **--file=<yaml_filename2>**, and so on.

For more information, see Checking deployment YAML files.

1.2.12. Filter workload CVEs by using component and component source

With RHACS 4.4, you can refine your vulnerability views based on components and their sources by using Vulnerability Management 2.0.

You can use the following filter options:

- **Filter by component**: Uses the component name to narrow down the vulnerabilities. For example, if the Ruby Action Pack contains the vulnerability, the component is **actionpack**.
- **Filter by component source**: Identifies the type of software element that contains the vulnerability, such as **NODEJS** or **OS**. You can tailor your views to your specific needs.

1.2.13. Life cycle updates

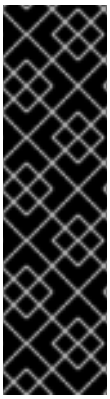
RHACS 4.4 offers an extension of the RHACS release lifecycle and adds full and maintenance support phases.

For more information, see Red Hat Advanced Cluster Security for Kubernetes Support Policy .

For more information about the OpenShift Operator maintenance lifecycles, see the Red Hat Knowledgebase solution OpenShift Operator Life Cycles.

For more information about compatibility and supportability with Red Hat OpenShift releases, see the Red Hat Knowledgebase solution Red Hat Advanced Cluster Security for Kubernetes Support Matrix .

1.2.14. Authentication of AWS and GCP integrations by using short-lived tokens (Technology Preview)



IMPORTANT

Authentication of AWS and GCP integrations by using short-lived tokens is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

RHACS 4.4 introduces support for authentication of Amazon Web Service (AWS) and Google Cloud Platform (GCP) integrations by using short-lived tokens.

For Amazon ECR, AWS Security Hub, and AWS S3 integrations, support for authentication through AssumeRole by using the Secure Token Service (STS) has been added.

In the RHACS portal, go to **Platform Configuration → Integrations**, and then select the integration type, for example, **Amazon ECR**. You can select an existing integration or click **New integration** to create a new one.

In the integration form, select **Use container IAM role** to enable STS AssumeRole that applies to Amazon EKS and Red Hat OpenShift clusters.

For Google Artifact Registry, Google Container Registry, Google Security Command Center and Google Cloud Storage integrations, support for authentication through workload identity federation has been added.

In the RHACS portal, go to **Platform Configuration → Integrations**, and then select the integration type, for example, **Google Artifact Registry**. You can select an existing integration or click **New integration** to create a new one.

In the integration form, select **Use workload identity** to enable token federation for Google GKE and Red Hat OpenShift clusters.

For more information, see Integrating RHACS using short-lived tokens.

1.2.15. Cluster discovery by using cloud source integrations

RHACS 4.4 introduces a new cloud source integration type, which includes integration types for Paladin Cloud and Red Hat OpenShift Cluster Manager. Integrated cloud sources enable RHACS to discover cluster assets from the connected accounts.RHACS matches the discovered clusters against clusters already secured by Central.

In the RHACS portal, go to **Platform Configuration → Clusters → Discovered clusters**to view the current status of the discovered clusters.

For more information, see Integrating with cloud management platforms.

1.2.16. Short-lived API tokens for Central

RHACS 4.4 introduces the option of issuing short-lived API tokens for interaction with the Central API.

You can exchange the short-lived API tokens for an OpenID Connect (OIDC) identity token. This is useful for authenticating and authorizing machines in CI environments, for example.

In the RHACS portal, go to **Platform Configuration → Integrations → Machine access configuration** to create a configuration to enable exchanging OIDC identity tokens for short-lived RHACS-issued tokens.

1.2.17. Migration to stock Red Hat OpenShift SCCs during manual upgrade by using roxctl CLI

With RHACS 4.4, the platform moves away from using custom security context constraints (SCCs). Instead, RHACS services use Red Hat OpenShift SCCs, which ensures future-proof operation and a consistent security posture for RHACS.

For more information, see Migrating SCCs during the manual upgrade.

1.3. NOTABLE TECHNICAL CHANGES

- The default memory requirement for **scanner-db** has been increased from 200MiB to 512MiB to prevent out-of-memory (OOM) errors during database initialization when the memory pressure reaches the node.
- Scanner-slim can now read additional certificate authorities (CAs) correctly from the **additional-ca-sensor** secret.
- Administrator access is not required to delegate ad hoc scan requests to secured clusters.
- Existing integrations that are automatically generated are now deleted on Central startup if **ROX_DISABLE_AUTOGENERATED_REGISTRIES** is set to **true**.
- The **/v1/administration/usage** API endpoint is now considered stable.
- By requesting the **central-monitoring-tls** \ **sensor-monitoring-tls** secrets at startup, you can ensure the presence of the Red Hat OpenShift monitoring **/metrics** server certificate. This requirement only applies if Red Hat OpenShift Monitoring is enabled.
- Use the **ROX_MEMLIMIT** environment variable, which replaces the **GOMEMLIMIT** variable in configuration files. Although you can still use the standard Go environment variable **GOMEMLIMIT**, you should use **ROX_MEMLIMIT** instead to capture the memory limits for your deployment more effectively. **ROX_MEMLIMIT** sets the soft memory limit of the Go process to 95% of the configured amount. Define **ROX_MEMLIMIT** as an integer without units representing the number of bytes.
  If you upgrade RHACS by using the roxctl CLI, you need to manually edit the deployments to use the new variable.

  For more information, see Manually upgrading using the roxctl CLI.

- Publish open source instead of **stackrox.io** helm charts.
- Sensor captures runtime events even when not connected to Central.
- You can now edit the endpoint from an unauthenticated email notifier without any issues. However, if the endpoint is not unauthenticated, credentials are still required to make changes.
- When deleting a collection that is referenced by other objects, such as report configurations, the error message now includes the names of both the collection being deleted and the referencing object.
- The **ROX_SCAN_TIMEOUT** environment variable in Central and Sensor is now set to 10 minutes by default instead of 6 minutes.
- As announced in RHACS 4.2, authenticated access is now required for the **/v1/resources** endpoint.
- The default policy **systemctl Execution** is not triggered when you use the **--version** process argument. This change does not lead to a security issue, as the printed information relates to the capabilities supported by **systemd** at the time of creation, and not the capabilities of your host operating system.
- The default policy **No resource requests or limits specified** has been renamed to **No CPU request or memory limit specified**. It no longer checks the CPU limit or memory request; instead, it specifically recognizes whether the CPU request and memory limit are specified.
- Configure the claim mappings of the authentication provider and required attributes from the UI.
- Configure the API token expiration date. If it is not specified, the API token expires in 1 year.
- The information available on the **Network → Listening Endpoints**page has been improved and updated, including new information about endpoints such as the pod UID and namespace, and the removal of endpoints from the deleted pods.

1.4. DOCUMENTATION ADDITIONS

- A new **roxctl** CLI command reference guide that provides comprehensive reference information for using the **roxctl** CLI commands.
  For more information, see "roxctl" in roxctl CLI command reference.

1.5. DEPRECATED AND REMOVED FEATURES

Some features available in earlier releases have been deprecated or removed.

Deprecated functionality is still included in RHACS and continues to be supported; however, it will be removed in a future release of this product and is not recommended for new deployments. For the most recent list of major functionality deprecated and removed, see the following table. Additional information about some removed or deprecated functionality is available after the table.

In the table, features are marked with the following statuses:

- GA: General Availability
- TP: Technology Preview

- DEP: Deprecated
- REM: Removed
- NA: Not applicable

Table 1.2. Deprecated and removed features tracker

Feature	RHACS 4.2	RHACS 4.3	RHACS 4.4
definitions.stackrox.io	GA	GA	DEP
roxctl connectivity-map	DEP	DEP	DEP
roxctl generate netpol	DEP	DEP	DEP
/v1/clustercves/suppress APIs	GA	DEP	DEP
/v1/clustercves/unsuppress APIs	GA	DEP	DEP
/v1/cve/requests APIs	GA	DEP	DEP
/v1/nodecves/suppress APIs	GA	DEP	DEP
/v1/nodecves/unsuppress APIs	GA	DEP	DEP
Vulnerability Management (1.0) menu item	GA	DEP	DEP
Vulnerability Report Creator permission	DEP	DEP	DEP
/v1/availableAuthProviders endpoint	GA	GA	DEP
/v1/tls-challenge endpoint	GA	GA	DEP
Reporting of Istio vulnerabilities	GA	GA	DEP
Custom Security Context Constraints (SCCs): • **stackrox-collector** • **stackrox-admission-control** • **stackrox-sensor**	DEP	DEP	REM
CIS Docker v1.2.0 Compliance standard	DEP	DEP	REM
PCI DSS 3.2.1 Compliance standard	DEP	DEP	REM

Feature	RHACS 4.2	RHACS 4.3	RHACS 4.4
NIST SP 800-53 Compliance standard	DEP	DEP	REM
NIST SP 800-190 Compliance standard	DEP	DEP	REM
HIPAA 164 Compliance standard	DEP	DEP	REM
CIS Kubernetes v1.5 Compliance standard	DEP	DEP	REM
Reference image pull secret names for the Central components: • **stackrox** • **stackrox-scanner**	GA	DEP	REM
Reference image pull secret names for the secured cluster components: • **stackrox** • **stackrox-scanner** • **secured-cluster-services-main** • **secured-cluster-services-collector** • **collector-stackrox**	GA	DEP	REM

1.5.1. Deprecated features

The following section provides information about deprecated features listed in the preceding table and other additional changes:

- The **/v1/availableAuthProviders** endpoint is deprecated and in future releases, ensure that you authenticate and have at least **READ** permission on the **Access** resource when interacting with the **/v1/availableAuthProviders** endpoint.
- The **/v1/tls-challenge** endpoint is deprecated and in future releases, ensure that you have included proper authentication in all interactions with the **/v1/tls-challenge** endpoint.
- The reporting of Istio vulnerabilities is deprecated and is planned to be removed in a future release.

1.5.2. Removed features

The following section provides information about removed features listed in the preceding table and other additional changes:

- The sunburst widgets in the *Compliance* section have been removed.
- The Docker CIS benchmark has been removed.

- Custom **stackrox-*** security context constraints (SCCs) have been replaced with default SCCs.

- In the Helm and Operator installation modes, references to image pull secrets with specific names are no longer automatically added to service accounts. References are added for compatibility reasons if the secrets exist during installation or upgrade.
  The names of these special secrets are for the Central components such as **stackrox** and **stackrox-scanner**, and for secured cluster components such as **stackrox**, **stackrox-scanner**, **secured-cluster-services-main**, **secured-cluster-services-collector**, and **collector-stackrox**.

  You must explicitly list the required image pull secrets, specifically for Helm-based installations by using the **imagePullSecrets.useExisting** Helm value or for operator-based installations by using the **spec.imagePullSecrets** field in the StackRox custom resources (CRs). This is critical in environments without cluster lookup availability, such as CD pipelines like ArgoCD.

1.6. NOTICE OF UPCOMING CHANGES

- The following search terms will be deprecated and are planned to be removed from the deployment context in a future release:
  - Environment Key, Environment Value, and Environment Variable Source
    - By setting **ROX_DEPLOYMENT_ENVVAR_SEARCH** to **false**, you can remove these environment variable terms.
  - Volume Destination, Volume Name, Volume ReadOnly, Volume Source, and Volume Type
    - By setting **ROX_DEPLOYMENT_VOLUME_SEARCH** to **false**, you can remove these volume terms.
  - Secret and Secret Path
    - By setting **ROX_DEPLOYMENT_SECRET_SEARCH** to **false**, you can remove these secret terms.

- The following search terms will be deprecated and are planned to be removed from the secret context in a future release:
  - Secret Type, Cert Expiration, and Image Pull Secret Registry
    - By setting **ROX_SECRET_FILE_SEARCH** to **false**, you can remove these search terms.

- In RHACS 4.4, the current init bundle process is changed. As a result, the **Integrations → Init Bundle** page is planned to be removed in a future release.

- In RHACS 4.4, **definitions.stackrox.io** is deprecated and is planned to be removed in a future release.

- If you are currently using RHACS 3.74.x or an earlier version, you must stop at RHACS 4.4.x before proceeding with an upgrade to RHACS 4.5 or later versions, as RHACS has switched its underlying data store to PostgreSQL as of RHACS 4.0.0. During the upgrade, the data is automatically migrated to PostgreSQL.
  However, in the upcoming RHACS 4.5.0 release, it is anticipated that the previous data store will no longer be available. Skipping from RHACS 4.0.0 to 4.4.x could result in the data not being migrated properly.

- StackRox Scanner will not receive any new features and will go into maintenance mode. All ongoing development efforts are now focused on the new Scanner V4.

1.7. BUG FIXES

1.7.1. Resolved in version 4.4.0

Release date: 28 March 2024

- Fixed an issue where Sensor reduced the number of allocations required when evaluating process indicators, resulting in improved memory utilization in scenarios with a high volume of runtime events received simultaneously. In addition, the default maximum gRPC payload size has been increased from 12 MB to 24 MB.

- Fixed an issue where the RHACS integration with Jira Cloud could cause issues during creation. With this update, Jira issues that RHACS creates are prioritized correctly and the default priority mappings on the integration creation page in the RHACS portal have been updated to match the default priorities of Jira.
  Checks have been added during integration creation to reduce the risk of issue creation failing after saving. A checkbox has also been introduced to give you the option to disable the priority setting.

- Fixed an issue where an RHACS policy with inform and enforce on deployment did not work when the deployment was scaled up. RHACS has added **deployments/scale** in the resources and actions section of the validating webhook controller in the admission controller.
  With this update, if you use the **oc scale** command to scale the deployment from 0 to a number, the admission controller blocks it if the deployment violates a policy.

- Previously, a hanging network caused a random flake issue when you run the test scenario to open the user profile. With this update, this issue has now been fixed to improve the reliability of the test execution.

- Previously, there were issues with compliance trigger scan calls aborting after 60 seconds due to synchronous execution in the backend. Your current scan progress could be lost if you left and returned. This issue could affect you if you have numerous clusters and large data. The cause was the synchronous loading of data and the dependency on pending IDs, which are only determined during trigger scans.
  With this update, backend calls become asynchronous and an API parameter is introduced to retrieve the latest runs, enabling the user interface (UI) to recognize scans in progress. Improvements include loading running scans on page load, displaying progress and the ability to trigger additional scans during running scans.

- Previously, editing a custom vulnerability report associated with a specific report scope in RHACS 4.1 resulted in the loss of the report scope reference and would trigger **A report scope is required** message.
  The issue occurred when more than 10 collections were created and a vulnerability report edit was associated with 10 or more collections in RHACS 4.2 and 4.3. The UI retrieved the first 10 collections from the API, possibly resulting in a missing reference to the report area. This behavior might persist in RHACS 4.1 and 4.2.

  This update ensures that the linked collection is retained, avoiding the error message.

1.8. IMAGE VERSIONS

Image	Description	Current version
Main	Includes Central, Sensor, Admission controller, and Compliance. Also includes **roxctl** for use in continuous integration (CI) systems.	**registry.redhat.io/advanced-cluster-security/rhacs-main-rhel8:4.4.0**
Scanner	Scans images and nodes.	**registry.redhat.io/advanced-cluster-security/rhacs-scanner-rhel8:4.4.0**
Scanner DB	Stores image scan results and vulnerability definitions.	**registry.redhat.io/advanced-cluster-security/rhacs-scanner-db-rhel8:4.4.0**
Scanner V4	Scans images.	**registry.redhat.io/advanced-cluster-security/rhacs-scanner-v4-rhel8:4.4.0**
Scanner V4 DB	Stores image scan results and vulnerability definitions for Scanner V4.	**registry.redhat.io/advanced-cluster-security/rhacs-scanner-v4-db-rhel8:4.4.0**
Collector	Collects runtime activity in Kubernetes or OpenShift Container Platform clusters.	• **registry.redhat.io/advanced-cluster-security/rhacs-collector-rhel8:4.4.0** • **registry.redhat.io/advanced-cluster-security/rhacs-collector-slim-rhel8:4.4.0**
Central DB	PostgreSQL instance that provides the database storage for Central.	**registry.redhat.io/advanced-cluster-security/rhacs-central-db-rhel8:4.4.0**