



Red Hat build of Apicurio Registry 2.3

Release Notes for Apicurio Registry 2.3

What's new in Red Hat build of Apicurio Registry

Red Hat build of Apicurio Registry 2.3 Release Notes for Apicurio Registry 2.3

What's new in Red Hat build of Apicurio Registry

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[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

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

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] 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[®] 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

Describes the Red Hat build of Apicurio Registry product and provides the latest details on what's new in this release.

Table of Contents

CHAPTER 1. APICURIO REGISTRY RELEASE NOTES	3
1.1. APICURIO REGISTRY INSTALLATION OPTIONS	3
1.2. APICURIO REGISTRY SUPPORTED PLATFORMS	3
1.3. APICURIO REGISTRY NEW FEATURES	3
Apicurio Registry user documentation and examples	5
1.4. APICURIO REGISTRY DEPRECATED FEATURES	5
1.5. UPGRADING AND MIGRATING APICURIO REGISTRY DEPLOYMENTS	5
1.5.1. Upgrading a Apicurio Registry 2.0 deployment on OpenShift	6
1.5.2. Migrating a Apicurio Registry 1.1 deployment on OpenShift	6
1.6. APICURIO REGISTRY RESOLVED ISSUES	6
1.7. APICURIO REGISTRY RESOLVED CVES	8
1.8. APICURIO REGISTRY KNOWN ISSUES	9
Apicurio Registry core known issues	9
Apicurio Registry Operator known issues	9

CHAPTER 1. APICURIO REGISTRY RELEASE NOTES

Red Hat build of Apicurio Registry 2.3 is provided as a General Availability release. Apicurio Registry is a datastore for standard event schemas and API designs, and is based on the [Apicurio Registry](#) open source community project.

You can use Apicurio Registry to manage and share the structure of your data using a web console, REST API, Maven plug-in, or Java client. For example, client applications can dynamically push or pull the latest schema updates to or from Apicurio Registry without needing to redeploy. You can also create optional rules to govern how Apicurio Registry content evolves over time. These rules include content validation and backwards or forwards compatibility of schema or API versions.

1.1. APICURIO REGISTRY INSTALLATION OPTIONS

You can install Apicurio Registry on OpenShift with either of the following data storage options:

- PostgreSQL database
- Red Hat AMQ Streams

For more details, see [Installing and deploying Red Hat build of Apicurio Registry on OpenShift](#) .

1.2. APICURIO REGISTRY SUPPORTED PLATFORMS

Apicurio Registry 2.3 supports the following platform component versions:

- Red Hat OpenShift Container Platform 4.8 - 4.12
- Red Hat OpenShift Service on AWS
- Microsoft Azure Red Hat OpenShift
- PostgreSQL 12 - 15
- Red Hat AMQ Streams 2.1 - 2.3
- Red Hat Single Sign-On (RH-SSO) 7.6
- OpenJDK 11

1.3. APICURIO REGISTRY NEW FEATURES

Apicurio Registry 2.3 includes the following new features:

Apicurio Registry authentication and authorization

- *Expanded role-based authorization* - you can now configure role-based authorization in Apicurio Registry, as well as in RH-SSO as previously. If role-based authorization is enabled in the Apicurio Registry application, you can use the web console or REST API to control access.
- *Expanded owner-based authorization* - you can now enable the owner-based authorization option at the artifact-group level, as well as at the artifact level as previously.

- *Anonymous read access* - when the anonymous read access option is enabled, unauthenticated (anonymous) users have read-only access to all artifacts.
- *Authenticated read access* - when the authenticated read access option is enabled, any authenticated user has read-only access to all artifacts, even if the user has not been granted any Apicurio Registry roles.
- *HTTP basic authentication* - when this option is enabled, users or client applications can use HTTP basic authentication to access Apicurio Registry.
- *Custom TLS certificate for Kafka storage* - when using Kafka for storage, users can now securely connect to Kafka using a custom TLS certificate.
- *Change artifact owner* - administrators or artifact owners can change the owner of a specific schema or API artifact by using the REST API or web console.

Operational and monitoring improvements

- *Audit logging* - any changes to Apicurio Registry data result in an audit log entry.
- *Prometheus metrics* - metrics are exposed in Prometheus format for use in monitoring.
- *Sentry integration* - optional integration with Sentry 1.x.

Operator improvements

- *Custom environment variables* - you can now set arbitrary environment variables in the **ApicurioRegistry** custom resource. These variables are applied to Apicurio Registry using the **Deployment** resource.
- *Support for PodDisruptionBudget* - This resource is automatically created to ensure that at most one replica is unavailable.
- *Support for NetworkPolicy* - the Apicurio Registry Operator creates an ingress network policy for port 8080.

Artifact references

Artifacts can now reference other artifacts in Apicurio Registry. Many supported artifact types allow references from one file to another. For example, an OpenAPI file might have a data type with a property that references a JSON schema defined in another file. Typically, these references have a syntax specific to the artifact type. You can now use the REST API to create mappings so that type-specific references can be resolved to artifacts registered in Apicurio Registry.

Dynamic global configuration of Apicurio Registry instances

Apicurio Registry has many global configuration options that are typically set at deployment time. A subset of these options are now also configurable at runtime for a Apicurio Registry instance. You can manage these options at runtime by using the REST API or web console. For example, these options include owner-based authorization, anonymous read access, and authenticated read access.

Upload artifact from URL

You can now upload a schema or API artifact from a URL, in addition to the already supported upload from a file. You can upload by using the Apicurio Registry web console or the REST API.

Web console improvements

- *Import and export of Apicurio Registry data* - admin users can now use the web console to export all Apicurio Registry data in a **.zip** file, as well as using the REST API as previously. They can then import this **.zip** file into a different Apicurio Registry deployment.
- *Full support for artifact properties* - artifacts in Apicurio Registry can have user-defined and editable metadata such as name, description, labels (simple keyword list), and properties (name/value pairs). The web console has been enhanced to support displaying and editing properties, in addition to using the REST API as previously.
- *Documentation generation for AsyncAPI artifacts* - AsyncAPI artifacts now support the **Documentation** tab on the artifact details page. This tab displays human-readable documentation generated from the AsyncAPI content. This feature was previously available only for OpenAPI artifacts.
- *Option to display JSON as YAML* - for artifact types that are JSON formatted, the **Content** tab on the artifact details page now supports switching between JSON and YAML formats.

REST API improvements

- *Improved /users/me endpoint* - the Apicurio Registry core REST API has a **/users/me** endpoint that returns information about the current authenticated user. You can use this endpoint to inspect a user's assigned role and determine their capabilities.
- *Updated support for Confluent Compatibility API* - Apicurio Registry now supports the Confluent Schema Registry API version 6.

Apicurio Registry user documentation and examples

The documentation library has been updated with the new features available in version 2.3:

- [Installing and deploying Red Hat build of Apicurio Registry on OpenShift](#)
- [Migrating Red Hat build of Apicurio Registry deployments](#)
- [Red Hat build of Apicurio Registry User Guide](#)
- [Apicurio Registry v2 core REST API documentation](#)

The open source demonstration applications have also been updated:

- <https://github.com/Apicurio/apicurio-registry-examples>

1.4. APICURIO REGISTRY DEPRECATED FEATURES

Apicurio Registry version 1.x

Apicurio Registry version 1.x was deprecated in version 2.0 and is no longer fully supported. For more details, see the [Red Hat Application Services Product Update and Support Policy](#) .

1.5. UPGRADING AND MIGRATING APICURIO REGISTRY DEPLOYMENTS

You can upgrade automatically from Apicurio Registry 2.0 to Apicurio Registry 2.3 on OpenShift. There is no automatic upgrade from Apicurio Registry 1.x to Apicurio Registry 2.x, and a migration process is required.

1.5.1. Upgrading a Apicurio Registry 2.0 deployment on OpenShift

You can upgrade from Apicurio Registry 2.0.3 on OpenShift 4.9 to Apicurio Registry 2.3.x on OpenShift 4.11 or later. You must upgrade both your Apicurio Registry and your OpenShift versions, and upgrade OpenShift one minor version at a time.

Prerequisites

- You already have Apicurio Registry 2.0.3 installed on OpenShift 4.9.

Procedure

- In the OpenShift Container Platform web console, click **Administration** and then **Cluster Settings**.
- Click the pencil icon next to the **Channel** field, and select the next minor **candidate** version (for example, change from **stable-4.9** to **candidate-4.10**).
- Click **Save** and then **Update**, and wait until the upgrade is complete.
- If the OpenShift version is less than 4.11, repeat steps 2 and 3, and select **candidate-4.11** or later.
- Click **Operators** > **Installed Operators** > **Red Hat Integration - Service Registry**
- Ensure that the **Update channel** is set to **2.x**.
- If the **Update approval** is set to **Automatic**, the upgrade should be approved and installed immediately after the **2.x** channel is set.
- If the **Update approval** is set to **Manual**, click **Install**.
- Wait until the Operator is deployed and the Apicurio Registry pod is deployed.
- Verify that your Apicurio Registry system is up and running.

Additional resources

- For more details on how to set the Operator update channel in the OpenShift Container Platform web console, see [Changing the update channel for an Operator](#).

1.5.2. Migrating a Apicurio Registry 1.1 deployment on OpenShift

For details on migrating a Apicurio Registry 1.1 deployment to Apicurio Registry 2.x, see [Migrating Red Hat build of Apicurio Registry deployments](#).

1.6. APICURIO REGISTRY RESOLVED ISSUES

Table 1.1. Apicurio Registry resolved issues in version 2.3.0

Issue	Description
Registry-2394	REST API endpoint for core v1 compatibility not properly protected by authentication.

Issue	Description
Registry-1959	Web console incorrectly redirects to HTTP instead of HTTPS.
Registry-1926	Apicurio Registry throws io.apicurio.registry.storage.ArtifactNotFoundException while uploading a new artifact.
Registry-1905	Confluent compatibility layer not working with JSON Schema artifacts.
Registry-1873	kafkasql registry storage option throws Expected one element, but found none exception when querying contentIdFromHash .
Registry-1660	Confluent compatibility layer's schema DTO is not fully compatible.
Registry-1610	Web console does not properly obey the disable roles feature.
Registry-1593	Confluent compatibility API v6 does not return artifact.
Registry-733	Passing sasl.jaas.config property does not work with JAVA_OPTIONS environment variable.
Registry-651	Web console displays inconsistent modifiedOn date.
Registry-358	Global compatibility rule execution broken for RuleApplicationType.CREATE .
Registry-342	Transitive compatibility rules might give false positives.

Table 1.2. Apicurio Registry resolved issues in version 2.3.3

Issue	Description
IPT-858	Avro compatibility check does not work correctly for enum types.
Registry-3128	Add option to minify Avro with Apicurio Registry Maven plug-in.
Registry-3121	Make max subjects configurable in Confluent compatibility API.
Registry-3080	Throw exception when an empty schema is provided in the Confluent compatibility API.
Registry-3014	Fix handling of default JSON value in the Confluent compatibility API.
Registry-2991	On slow machines, kafkasql storage is not ready for existing messages.
Registry-2952	Fix version ordering in Apicurio Registry compatibility rules.
Registry-2919	Support application/json in registry export API operation.

Issue	Description
Registry-2913	Configuring Apicurio Registry event sourcing gives HTTP error.
Registry-2877	Protobuf schema version upload failing with NullPointerException .

1.7. APICURIO REGISTRY RESOLVED CVEs

Table 1.3. Apicurio Registry resolved Common Vulnerabilities and Exposures (CVEs) in version 2.3.x

Issue	Description
IPT-789	CVE-2022-25858 terser: Insecure use of regular expressions leads to ReDoS.
IPT-788	CVE-2022-37734 graphql-java: DoS by malicious query.
IPT-787	CVE-2022-25857 snakeyaml: DoS due to missing nested depth limitation for collections.
IPT-764, IPT-763	CVE-2022-31129 moment: Inefficient parsing algorithm resulting in DoS.
IPT-760	CVE-2022-25647 com.google.code.gson-gson: Deserialization of untrusted data.
IPT-739	CVE-2022-24773 node-forge: Signature verification leniency in checking DigestInfo structure.
IPT-738	CVE-2022-24772 node-forge: Signature verification failing to check trailing garbage bytes can lead to signature forgery.
IPT-737	CVE-2022-24771 node-forge: Signature verification leniency in checking digestAlgorithm structure can lead to signature forgery.
IPT-734	CVE-2022-26520 jdbc-postgresql: Arbitrary file write vulnerability.
IPT-733	CVE-2022-0536 follow-redirects: Exposure of sensitive information by Authorization Header leak.
IPT-732	CVE-2022-0235 node-fetch: Exposure of sensitive information to an unauthorized actor.
IPT-731	CVE-2022-23647 prismjs: Improperly escaped output allows an XSS vulnerability.
IPT-728	CVE-2022-0981 quarkus: Privilege escalation vulnerability with RestEasy Reactive scope leakage in Quarkus.
IPT-723	CVE-2022-21724 quarkus-jdbc-postgresql-deployment: Unchecked class instantiation when providing plug-in classes.

Issue	Description
IPT-705	CVE-2021-22569 protobuf-java: Potential DoS in the parsing procedure for binary data.
IPT-652	CVE-2021-41269 cron-utils: Template injection leading to unauthenticated Remote Code Execution vulnerability.
IPT-566	CVE-2021-37136 netty-codec: Bzip2Decoder doesn't allow setting size restrictions for decompressed data.
IPT-564	CVE-2021-37137 netty-codec: SnappyFrameDecoder doesn't restrict chunk length and might buffer skippable chunks in an unnecessary way.

1.8. APICURIO REGISTRY KNOWN ISSUES

The following known issues apply in Apicurio Registry 2.3.3:

Apicurio Registry core known issues

[IPT-814](#) - Apicurio Registry logout feature incompatible with RH-SSO 7.6

In RH-SSO 7.6, the **redirect_uri** parameter used with the logout endpoint is deprecated. For more details, see the [RH-SSO 7.6 Upgrading Guide](#). Because of this deprecation, when Apicurio Registry is secured by using the RH-SSO Operator, clicking the **Logout** button displays the **Invalid parameter: redirect_uri** error.

For a workaround, see <https://access.redhat.com/solutions/6980926>.

[IPT-701](#) - CVE-2022-23221 H2 allows loading custom classes from remote servers through JNDI

When Apicurio Registry data is stored in AMQ Streams, the H2 database console allows remote attackers to execute arbitrary code by using the JDBC URL. Apicurio Registry is not vulnerable by default and a malicious configuration change is required.

Apicurio Registry Operator known issues

[Operator-42](#) - Autogeneration of OpenShift route might use wrong base host value

If multiple **routerCanonicalHostname** values are specified, autogeneration of the Apicurio Registry OpenShift route might use a wrong base host value.