Red Hat build of Apicurio Registry 2.5

Release Notes for Apicurio Registry 2.5

What's new in Red Hat build of Apicurio Registry

Last Updated: 2024-03-22

Red Hat build of Apicurio Registry 2.5 Release Notes for Apicurio Registry 2.5

What's new in Red Hat build of Apicurio Registry

Legal Notice

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

PREFACE

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright's message .

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your feedback on our documentation.

To propose improvements, open a Jira issue and describe your suggested changes. Provide as much detail as possible to enable us to address your request quickly.

Prerequisite

- You have a Red Hat Customer Portal account. This account enables you to log in to the Red Hat Jira Software instance.
  If you do not have an account, you will be prompted to create one.

Procedure

1. Click the following link: Create issue.
2. In the **Summary** text box, enter a brief description of the issue.
3. In the **Description** text box, provide the following information:
   - The URL of the page where you found the issue.
   - A detailed description of the issue.
     You can leave the information in any other fields at their default values.
4. Click **Create** to submit the Jira issue to the documentation team.

Thank you for taking the time to provide feedback.

CHAPTER 1. APICURIO REGISTRY 2.5 RELEASE NOTES

Red Hat build of Apicurio Registry is a data store for standard event schemas and API designs, and is based on the Apicurio Registry open source community project.

NOTE

Red Hat build of Apicurio Registry is the new product name for Red Hat Integration Service Registry. Red Hat build of Apicurio Registry 2.x and Red Hat Integration Service Registry 2.x are functionally identical.

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 validation of content, integrity of artifact references, 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 2.5 supports the following core platforms:

- Red Hat OpenShift Container Platform: 4.15, 4.14, 4.13, 4.12
- Red Hat OpenShift Service on AWS: 4.13
- Microsoft Azure Red Hat OpenShift: 4.13
- PostgreSQL: 15, 14, 13, 12
- Red Hat AMQ Streams: 2.6, 2.5, 2.2
- OpenJDK: 17, 11

For more details, see the following article:

- Supported Configurations for Red Hat build of Apicurio Registry .

1.2.1. Supported integration with other products

Apicurio Registry 2.5 also supports integration with the following products:

- Red Hat Single Sign-On (RH-SSO) 7.6

- Red Hat build of Debezium 2.3

1.2.2. Operator metadata versions

For details on the corresponding Service Registry Operator metadata versions used to install and deploy Apicurio Registry, see the following article:

- Red Hat Integration - Service Registry Operator metadata versions .

1.3. APICURIO REGISTRY NEW FEATURES

Apicurio Registry 2.5 includes the following new features:

Apicurio Registry core new features

Upgrade to Quarkus 3.x

- The Apicurio Registry server runtime has been upgraded from Quarkus 2.x to Quarkus 3.x. This upgrade provides improved security, performance, and maintenance. For more details, see https://quarkus.io/quarkus3/. Apicurio Registry 2.5 is built on Quarkus 3.2.

Avro SerDes improvements

- Support for generation of schemas with null fields when using Apache Avro serializers/deserializers. For more details, see Registry-3862.

Schema cache fault tolerance

- Added the option to use an existing schema cache entry instead of throwing an error if schema cache loading fails. For more details, see Registry-3807.

Dereference artifact content

- There are some cases where returning artifact content with referenced content inline might be helpful. For these cases, the Core Registry API v2 adds support for the **dereference** query parameter in certain operations. For more details, see the Apicurio Registry v2 core REST API documentation.

- This support is currently implemented only for Avro and Protobuf artifacts when the **dereference** parameter is specified in the API operation. This parameter is not supported for any other artifact types. For more details, see Registry-2865.

  NOTE

  For Protobuf artifacts, dereferencing content is supported only when all of the schemas belong to the same package.

Apicurio Registry Maven plug-in improvements

- Add the option to skip the **register** goal in the Maven plug-in. For more details, see Registry-3817.

- Automatic detection of references in the Maven plug-in by using the **autoRef** option in the **pom.xml** file. For more details, see Registry-3439. This is a Technology Preview feature.

IMPORTANT

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 .

Apicurio Registry Operator new features

Improved support for SQL data source configuration

- Apicurio Registry Operator supports configuring an SQL data source by using environment variables as an alternative to **spec.configuration.sql.dataSource** fields. You can now provide SQL credentials using Kubernetes secrets instead of plaintext in the **ApicurioRegistry** custom resource. For more details, see https://access.redhat.com/solutions/7059053.

- Apicurio Registry Operator has been improved in this version to better support this use case. You can now use both the **spec.configuration.sql.dataSource** and **spec.configuration.env** fields to define parts of the configuration. For example, the following configuration is now valid:

  ```
  apiVersion: registry.apicur.io/v1
  kind: ApicurioRegistry
  metadata:
    name: myregistry
  spec:
    configuration:
      persistence: sql
    sql:
      dataSource:
        url: "jdbc:postgresql://..."
        userName: "postgres-user"
      env:
        - name: REGISTRY_DATASOURCE_PASSWORD
          valueFrom:
            secretKeyRef:
              name: postgres-secret
              key: password
  ```

  The Operator also detects this type of configuration and applies it immediately without additional user intervention.

Apicurio Registry user documentation and examples

The documentation library has been updated with the new features available in version 2.5:

- 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 core deprecated features

- *Confluent Schema Registry API version 6 (compatibility API)* : Apicurio Registry currently supports two versions of the Confluent Schema Registry API on separate endpoints: version 6 and version 7. The v6 API endpoint is deprecated, and will be removed in a future release. Ensure that you replace all references to the v6 API endpoint with references to the v7 API endpoint.
- *Apicurio Registry Core API version 1* : Apicurio Registry support for the original version 1 of the Apicurio Registry Core API is now deprecated. This v1 legacy API will be removed in the next major release.
- *Dynamic log level configuration*: The **/admin/loggers** and **/admin/loggers/{logger}** API endpoints are now deprecated in the v2 Apicurio Registry Core API. These endpoints will be removed in a future release.
- *Registry V1 export utility* : Apicurio Registry support for the command-line export utility is now deprecated. The export tool, which is used to export data from Apicurio Registry 1.x into a format that can be imported into 2.x, will no longer be released or maintained. All customers should have already upgraded from 1.x to 2.x.

Apicurio Registry Operator deprecated features

- *JAVA_OPTIONS environment variable*: The **JAVA_OPTIONS** environment variable is no longer the preferred way to configure Java options for Apicurio Registry. You can use the **JAVA_OPTS_APPEND** environment variable instead. The **JAVA_OPTS** environment variable is also available, which replaces the default content of Java options. However, it is best to avoid using **JAVA_OPTS** because it might interfere with some Apicurio Registry Operator functionality.
- *Setting environment variables by editing the Deployment resource* : In previous versions, you could set environment variables for Apicurio Registry by directly editing its **Deployment** resource, which was supported by the Apicurio Registry Operator. Now that you can manage environment variables by using the **spec.configuration.env** field in the **ApicurioRegistry** CRD file, the previous procedure is deprecated and the Operator support for it will be removed. Ensure that you use the **spec.configuration.env** field to set all environment variables that are not set by the Operator.
- *Retention of environment variables for features that are not enabled* : The Apicurio Registry Operator sets environment variables to enable and configure various features, such as Salted Challenge Response Authentication Mechanism (SCRAM) security when using Kafka storage. When such features are disabled, the Operator currently retains the associated environment variables, which can cause problems. Retention of such environment variables is deprecated, and the Operator support for it will be removed. Ensure that your deployment does not rely on the retention of such environment variables.
- *Environment variable precedence* : The Apicurio Registry Operator might attempt to set an environment variable that is already explicitly specified in the **spec.configuration.env** field. If an

environment variable has a conflicting value, the value set by the Apicurio Registry Operator takes precedence by default. This behavior will change in the future, to enable users to overwrite most environment variables set by the Operator. Ensure that your deployment does not rely on the original precedence behavior.

1.5. UPGRADING AND MIGRATING APICURIO REGISTRY DEPLOYMENTS

You can upgrade the Apicurio Registry server automatically from Apicurio Registry 2.x to Apicurio Registry 2.5 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. Updating 2.x client dependencies

It is not mandatory to update client dependencies for this release. Existing Apicurio Registry 2.x client applications continue to work with Apicurio Registry 2.5.

However, before the next release of Apicurio Registry, you must update all of your client dependencies to use the latest version of Apicurio Registry. Client dependencies include dependencies for the Apicurio Registry Kafka serializers/deserializers (SerDes), Maven plug-in, and Java client applications.

For example, to update the Maven dependencies for a Java client application, specify the version in your **pom.xml** file as follows:

```
<dependency>
   <groupId>io.apicurio</groupId>
   <artifactId>apicurio-registry-client</artifactId>
   <version>2.5.10.Final-redhat-00001</version>
</dependency>
```

For more details, see Legacy REST API date formats enabled by default .

1.5.2. Upgrading from Apicurio Registry 2.x on OpenShift

You can upgrade from Apicurio Registry 2.x on OpenShift 4.11 to Apicurio Registry 2.5 on OpenShift 4.12 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.x installed on OpenShift 4.11 or later.
- You have backed up your existing Apicurio Registry storage data in your Kafka topic or PostgreSQL database. For more details, see Installing and deploying Red Hat build of Apicurio Registry on OpenShift.

> **IMPORTANT**
>
> In production environments on OpenShift, to help ensure that storage is backed up before upgrading, it is best to set the Operator update approval strategy for Apicurio Registry to manual instead of automatic.

Procedure

1. In the OpenShift Container Platform web console, click **Administration** and then **Cluster Settings**.
2. Click the pencil icon next to the **Channel** field, and select the next minor **candidate** version (for example, change from **stable-4.11** to **candidate-4.12**).
3. Click **Save** and then **Update**, and wait until the upgrade is complete.
4. If the OpenShift version is less than 4.13, repeat steps 2 and 3, and select **candidate-4.13** or later.
5. Click **Operators** > **Installed Operators** > **Red Hat Integration - Service Registry**.
6. Ensure that the **Update channel** is set to **2.x**.
7. If the **Update approval** is set to **Automatic**, the upgrade should be approved and installed immediately after the **2.x** channel is set.
8. If the **Update approval** is set to **Manual**, click **Install**.
9. Wait until the Operator is deployed and the Apicurio Registry pod is deployed.
10. 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.

1.5.3. Migrating from Apicurio Registry 1.1 on OpenShift

For details on migrating from Apicurio Registry 1.1 to Apicurio Registry 2.x, see Migrating Red Hat build of Apicurio Registry deployments.

1.6. APICURIO REGISTRY RESOLVED ISSUES

Table 1.1. Resolved issues in Apicurio Registry 2.5.10

Issue	Description
IPT-1091	Apicurio Registry Operator should support both the JAVA_OPTS_APPEND and JAVA_OPTIONS (deprecated) environment variables.
IPT-1092	Apicurio Registry upgrade breaks Confluent v6 compatibility API.

Table 1.2. Resolved issues in Apicurio Registry 2.5.9

Issue	Description
IPT-1071	Possible data loss when upgrading Apicurio Registry with KafkaSQL storage and Protobuf artifacts with references.

Issue	Description
IPT-1035	Apicurio Registry Pods in CrashLoop after Operator upgrade to 2.2.3.
Registry-4417	Orphaned content not properly deleted from Apicurio Registry.
Registry-4283	Apicurio Registry server should fail when two references with the same name are created for a single artifact.
Registry-4226	Delete all rules REST API operation doesn’t delete the **INTEGRITY** rule (KafkaSQL storage only).
Registry-4215	Avro with different field order is considered equal in canonical form.
Registry-4107	Validity, compatibility, and integrity rule values are not displayed in the Apicurio Registry web console in read-only mode.

Table 1.3. Resolved issues in Apicurio Registry 2.5.5

Issue	Description
Registry-4104	When AMQ Streams storage and OAuth are configured, Apicurio Registry fails to start due to missing kafka-oauth-client class.

Table 1.4. Resolved issues in Apicurio Registry 2.5.4

Issue	Description
Registry-4019	Some health checks are always UP even when a counter hits the limit.
Registry-3956	Schema registry is called even when the schema already exists in the local cache (SerDes).
Registry-3725	Resource owner password grant - basic auth - java.lang.IllegalStateException: Client is closed.
Registry-3647	Protobuf content canonicalHash outdated value detected.

1.7. APICURIO REGISTRY RESOLVED CVES

The following Common Vulnerabilities and Exposures (CVEs) are resolved in Apicurio Registry 2.5:

Table 1.5. CVEs resolved in Apicurio Registry 2.5.9

CVE	Description
CVE-2024-20952 CVE-2024-20921 CVE-2024-20919 CVE-2024-20918	Difficult to exploit vulnerability allows unauthenticated attacker with network access via multiple protocols to compromise Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition.
CVE-2024-20945	Difficult to exploit vulnerability allows low privileged attacker with logon to the infrastructure where Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition executes to compromise Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition.
CVE-2024-20932	Easily exploitable vulnerability allows unauthenticated attacker with network access via multiple protocols to compromise Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition.
CVE-2023-39615	A flaw was found in Libxml2, where it contains a global buffer overflow via the xmlSAX2StartElement() function at /libxml2/SAX2.c.
CVE-2023-38473	A vulnerability was found in Avahi. A reachable assertion exists in the avahi_alternative_host_name() function.
CVE-2023-38472	A vulnerability was found in Avahi. A reachable assertion exists in the avahi_rdata_parse() function.
CVE-2023-38471	A vulnerability was found in Avahi. A reachable assertion exists in the dbus_set_host_name function.
CVE-2023-38470	A vulnerability was found in Avahi. A reachable assertion exists in the avahi_escape_label() function.
CVE-2023-38469	A vulnerability was found in Avahi, where a reachable assertion exists in avahi_dns_packet_append_record.
CVE-2023-27043	The email module of Python through 3.11.3 incorrectly parses e-mail addresses that contain a special character.
CVE-2023-7104	A vulnerability was found in SQLite3. This issue affects the sessionReadRecord function of the ext/session/sqlite3session.c function in the make alltest Handler component. Manipulation may cause a heap-based buffer overflow to occur.
CVE-2023-5981	A vulnerability was found that the response times to malformed ciphertexts in RSA-PSK ClientKeyExchange differ from response times of ciphertexts with correct PKCS#1 v1.5 padding.
CVE-2023-5678	A flaw was found in OpenSSL, which caused the generation or checking of long X9.42 DH keys or parameters to be much slower than expected. This issue could lead to a denial of service.

CVE	Description
CVE-2023-5388	It was discovered that the numerical library used in NSS for RSA cryptography leaks information whether high order bits of the RSA decryption result are zero.
CVE-2023-3817 CVE-2023-3446	A vulnerability was found in OpenSSL. This security issue occurs because the applications that use the DH_check(), DH_check_ex(), or EVP_PKEY_param_check() functions to check a DH key or DH parameters may experience long delays.
CVE-2022-48564	A vulnerability was found in the Python core plistlib library within the read_ints() function in the plistlib.py file.
CVE-2022-48560	A use-after-free vulnerability was found in Python via the heappushpop function in the heapq module.
CVE-2021-3468	A flaw was found in avahi. The event used to signal the termination of the client connection on the avahi Unix socket is not correctly handled in the client_work function, allowing a local attacker to trigger an infinite loop.

Table 1.6. Resolved CVE issues in Apicurio Registry 2.5.4

Issue	Description
IPT-1034	CVE-2023-5072 JSON-java: parser confusion leads to OOM error.
IPT-1030	CVE-2023-31582 jose4j: Insecure iteration count setting.
IPT-1021	CVE-2023-44487 undertow: HTTP/2: Multiple HTTP/2 enabled web servers are vulnerable to a DDoS attack (Rapid Reset Attack).
IPT-1013	CVE-2023-39410 avro: apache-avro: Apache Avro Java SDK: Memory when deserializing untrusted data in Avro Java SDK.
IPT-995	CVE-2023-4853 quarkus-vertx-http: quarkus: HTTP security policy bypass.
IPT-993	CVE-2023-39321 CVE-2023-39322 integration-service-registry-operator-container: various flaws.
IPT-953	CVE-2023-29409 integration-service-registry-operator-container: golang: crypto/tls: slow verification of certificate chains containing large RSA keys.
IPT-948	CVE-2023-29406 integration-service-registry-operator-container: golang: net/http: insufficient sanitization of Host header.
IPT-940	CVE-2023-34462 netty: SniHandler 16MB allocation leads to OutOfMemoryError.
IPT-936	CVE-2023-34455 snappy-java: Unchecked chunk length leads to DoS.

Issue	Description
IPT-935	CVE-2023-35116 jackson-databind: denial of service via cyclic dependencies.
IPT-874	CVE-2023-1584 quarkus-oidc: ID and access tokens leak via the authorization code flow.

Table 1.7. Additional CVEs resolved in Apicurio Registry 2.5.4

CVE	Description
CVE-2023-44483	All versions of Apache Santuario - XML Security for Java prior to 2.2.6, 2.3.4, and 3.0.3, when using the JSR 105 API, are vulnerable to an issue where a private key may be disclosed in log files when generating an XML Signature and logging with debug level is enabled.
CVE-2023-43642	A flaw was found in SnappyInputStream in snappy-java, a data compression library in Java. This issue occurs when decompressing data with a too-large chunk size due to a missing upper bound check on chunk length.
CVE-2023-42503	Apache Commons Compress: Denial of service via CPU consumption for malformed TAR file.
CVE-2023-40217	Python 3 ssl.SSLSocket is vulnerable to a bypass of the TLS handshake in certain instances for HTTPS servers and other server-side protocols that use TLS client authentication such as mTLS.
CVE-2021-39194	Denial of service while parsing polymorphic input with tagged polymorphism style in kaml
CVE-2023-34454 CVE-2023-34453	A flaw was found in Snappy-java's shuffle function, which does not check input sizes before beginning operations.
CVE-2023-29491	A vulnerability was found in ncurses and occurs when used by a setuid application.
CVE-2023-28118	kaml has potential denial of service while parsing input with anchors and aliases.
CVE-2022-24823	When using multipart decoders in netty, local information disclosure can occur via the local system temporary directory if temporary storing of uploads on the disk is enabled.
CVE-2023-4911	A buffer overflow was discovered in the GNU C Library's dynamic loader ld.so while processing the GLIBC_TUNABLES environment variable.
CVE-2023-4813	A flaw was found in glibc. In an uncommon situation, the gaih_inet function may use memory that has been freed, resulting in an application crash.
CVE-2023-4806	A flaw was found in glibc. In an extremely rare situation, the getaddrinfo function may access memory that has been freed, resulting in an application crash.

CVE	Description
CVE-2023-4527	A flaw was found in glibc. When the getaddrinfo function is called with the AF_UNSPEC address family and the system is configured with no-aaaa mode via /etc/resolv.conf, a DNS response via TCP larger than 2048 bytes can potentially disclose stack contents through the function returned address data, and may cause a crash.

1.8. APICURIO REGISTRY KNOWN ISSUES

The following known issues apply in Apicurio Registry 2.5:

Apicurio Registry core known issues

Registry-3413 - Legacy REST API date formats enabled by default

For maximum compatibility and for easier upgrades from older versions of Apicurio Registry, the date format used in the Apicurio Registry REST API is not compliant with OpenAPI standards. This is because of a bug in older versions.

Before the next release of Apicurio Registry, you must upgrade all of your client applications to use the latest Apicurio Registry client version. The next release will fix the date format bug, which will result in older clients no longer being compatible with the REST API.

To update your REST API to be OpenAPI compliant, you can fix the date format bug in *this* version of Apicurio Registry as follows:

1. Update all of your client applications to version **2.5.10.Final-redhat-00001**, as described in Updating 2.x client dependencies.
2. Set the following environment variable to the value shown:

```
REGISTRY_APIS_V2_DATE_FORMAT=yyyy-MM-dd'T'HH:mm:ss'Z'
```

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.

APPENDIX A. USING YOUR SUBSCRIPTION

Apicurio Registry is provided through a software subscription. To manage your subscriptions, access your account at the Red Hat Customer Portal.

Accessing your account

1. Go to access.redhat.com.
2. If you do not already have an account, create one.
3. Log in to your account.

Activating a subscription

1. Go to access.redhat.com.
2. Navigate to **My Subscriptions**.
3. Navigate to **Activate a subscription** and enter your 16-digit activation number.

Downloading ZIP and TAR files

To access ZIP or TAR files, use the customer portal to find the relevant files for download. If you are using RPM packages, this step is not required.

1. Open a browser and log in to the Red Hat Customer Portal **Product Downloads** page at access.redhat.com/downloads.
2. Locate the **Red Hat Integration** entries in the **Integration and Automation** category.
3. Select the desired Apicurio Registry product. The **Software Downloads** page opens.
4. Click the **Download** link for your component.

Revised on 2024-03-22 13:11:17 UTC