Red Hat Integration 2021.Q3

Release Notes for Red Hat Integration 2021.Q3

What's new in Red Hat Integration
What’s new in Red Hat Integration
Abstract

Describes the Red Hat Integration platform and provides the latest details on what’s new in this release.
# Table of Contents

## CHAPTER 1. RED HAT INTEGRATION

## CHAPTER 2. NEW FEATURES IN THIS RELEASE

2.1. NEW INTEGRATION FEATURES .......................... 4
2.2. NEW COMPONENT FEATURES .......................... 4

## CHAPTER 3. SERVICE REGISTRY RELEASE NOTES

3.1. SERVICE REGISTRY INSTALLATION OPTIONS ............. 5
3.2. SERVICE REGISTRY NEW FEATURES ..................... 5
   Service Registry security .................................. 5
   Service Registry core ................................... 5
   Service Registry data storage ............................ 5
   Service Registry v2 REST API ............................ 6
   Service Registry Operator ................................ 6
   Service Registry platform component versions .......... 6
   Service Registry user documentation and examples ... 6
3.3. SERVICE REGISTRY DEPRECATED AND REMOVED FEATURES .. 7
   Service Registry deprecated features .................. 7
   Service Registry removed features ..................... 7
3.4. MIGRATING SERVICE REGISTRY DEPLOYMENTS ........... 7
3.5. SERVICE REGISTRY RESOLVED ISSUES .................. 7
   Service Registry core resolved issues ................ 7
   Service Registry Operator resolved issues .......... 8
3.6. SERVICE REGISTRY KNOWN ISSUES .................... 8
   Service Registry core known issues .................. 8
   Service Registry operator known issues .............. 8

## CHAPTER 4. RED HAT INTEGRATION OPERATORS

4.1. WHAT OPERATORS ARE ................................ 9
4.2. RED HAT INTEGRATION OPERATOR ....................... 9
   4.2.1. Support life cycle ................................ 10
4.3. RED HAT INTEGRATION COMPONENT OPERATORS .......... 10
   4.3.1. 3scale Operators ................................ 10
   4.3.2. AMQ Operators ................................... 10
   4.3.3. Camel K Operator ................................ 10
   4.3.4. Fuse Operators ................................... 10
   4.3.5. Service Registry Operator ...................... 10
CHAPTER 1. RED HAT INTEGRATION

Red Hat Integration is a comprehensive set of integration and event processing technologies for creating, extending, and deploying container-based integration services across hybrid and multicloud environments. Red Hat Integration provides an agile, distributed, and API-centric solution that organizations can use to connect and share data between applications and systems required in a digital world.

Red Hat Integration includes the following capabilities:

- Real-time messaging
- Cross-datacenter message streaming
- API connectivity
- Application connectors
- Enterprise integration patterns
- API management
- Data transformation
- Service composition and orchestration

Additional resources

- Understanding enterprise integration
CHAPTER 2. NEW FEATURES IN THIS RELEASE

This section provides a summary of the key new features in Red Hat Integration 2021.Q3 and provides links to more details on new features available in different components.

NOTE

These release notes include details on components updated in Red Hat Integration 2021.Q3 only. For details on the latest versions of other components, such as Debezium, Camel K, and Camel Quarkus, see the Red Hat Integration Release Notes for 2021-Q1 and Red Hat Integration Release Notes for 2021-Q2.

2.1. NEW INTEGRATION FEATURES

Kafka schema registry

- Improved Apache Kafka schema registry and API registry with OpenShift Operator in Service Registry 2.0 General Availability

2.2. NEW COMPONENT FEATURES

For more details on what’s new in Red Hat Integration 2021.Q3 components:

- Red Hat 3scale API Management
  - Red Hat 3scale API Management 2.10 On-Premises Release Notes
  - Red Hat 3scale API Management SaaS Release Notes
- Red Hat AMQ Product Documentation
- Red Hat Fuse 7.8 Release Notes
CHAPTER 3. SERVICE REGISTRY RELEASE NOTES

Red Hat Integration - Service Registry 2.0 is available as a General Availability component in Red Hat Integration 2021.Q3. Service Registry is a datastore for standard event schemas and API designs, which is based on the Apicurio Registry open source community project.

You can use Service 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 Service Registry without needing to redeploy. You can also use Service Registry to create optional rules to govern how registry content evolves over time. For example, this includes rules for content validation or backwards and forwards compatibility of schema or API versions.

3.1. SERVICE REGISTRY INSTALLATION OPTIONS

You can install Service Registry on OpenShift with the following data storage options:

Table 3.1. Service Registry storage options

<table>
<thead>
<tr>
<th>Storage option</th>
<th>Release category</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMQ Streams 1.7</td>
<td>General Availability</td>
</tr>
<tr>
<td>PostgreSQL 12 database</td>
<td>General Availability</td>
</tr>
</tbody>
</table>

3.2. SERVICE REGISTRY NEW FEATURES

Service Registry security

- **Authentication based on Red Hat Single Sign-On** - optionally protect the registry so that its REST API requires users to authenticate (OAuth and HTTP basic are supported)

- **Role-based authorization** - when authentication is enabled, users must have at least one role of sr-admin, sr-developer, or sr-readonly

- **Creator-only authorization** - option to prevent changes to artifacts unless the authenticated user originally created the artifact

Service Registry core

- **Registry artifact groups** - optionally organize schema and API artifacts into custom named logical groupings

- **Refactored Kafka serializer/deserializer (SerDe) classes** - significant updates to the Java SerDe layer to address ease of use, consistency, and functionality

- **Event sourcing** - option to configure the registry to trigger events whenever a change is made, based on the CloudEvents specification

Service Registry data storage

- **SQL-based storage** - new SQL storage implementation with support for PostgreSQL database
- Kafka-based storage - new hybrid storage using AMQ Streams to store artifact data and an embedded SQL database to represent it in memory

**Service Registry v2 REST API**

- Custom versioning - option to provide a custom version number when using the REST API to create or update artifacts
- Improved artifact searching - updates to the REST API to allow improved searching of artifacts
- Import/export API - updates to the REST API with operations to export and import registry data in .zip format
- CNCF Schema Registry API support - implementation of the Cloud Native Computing Foundation Schema Registry REST API

**NOTE**

The Service Registry v2 REST API is compatible with the Confluent Schema Registry REST API, which does not include the new artifact groups. Backwards compatibility is also maintained with the existing Service Registry v1 REST API.

**Service Registry Operator**

- Improved performance and streamlining - Operator uses Deployment on OpenShift (instead of DeploymentConfig), predictable resource naming (no random suffixes), and resources created in parallel.
- Registry data storage - support for the new SQL and Kafka-based storage options.
- Registry security - support for authentication and authorization configuration using Red Hat Single Sign-On.
- ApicurioRegistry CRD v1 - uses standardized conditions field in the status block to better indicate issues or errors in the Operator or the application.
- Multi-namespace deployment - When the Operator is installed in a namespace, it can watch all namespaces (or a selected subset), so applications can be deployed in any or multiple namespaces.

**Service Registry platform component versions**

- OpenShift Container Platform 4.6 or 4.7
- OpenJDK 11
- AMQ Streams 1.7
- PostgreSQL 12
-Debezium 1.4
- Camel Kafka Connector - Technology Preview

**Service Registry user documentation and examples**

- New documentation library updated with new features in version 2.0:
Installing and deploying Service Registry on OpenShift

Migrating Service Registry deployments

Service Registry User Guide

Registry v2 core REST API documentation

- New open source demonstration applications:
  - https://github.com/Apicurio/apicurio-registry-examples

3.3. SERVICE REGISTRY DEPRECATED AND REMOVED FEATURES

Service Registry deprecated features

- Service Registry version 1.x has been deprecated in version 2.0 and will soon go out of full support. For more details, see the Red Hat Middleware Product Update and Support Policy.

Service Registry removed features

- Infinispan cache-based storage option has been removed
- Java Persistence API (JPA) storage option has been replaced by the new PostgreSQL database storage option
- Kafka-based storage option in AMQ Streams has been replaced by the new hybrid storage option in AMQ Streams with in-memory H2 database
- Service Registry Java client no longer supports OpenJDK 8 and supports OpenJDK 11 instead

3.4. MIGRATING SERVICE REGISTRY DEPLOYMENTS

For details migrating from Service Registry version 1.1 to 2.x, see Migrating Service Registry deployments.

For details on migrating registry data between Service Registry version 2.x instances, see Exporting and importing registry content using the Registry REST API.

3.5. SERVICE REGISTRY RESOLVED ISSUES

Service Registry core resolved issues

Registry-1289 - Registry does not work on IPv6

When trying to deploy Service Registry using the Operator on a Kubernetes server with Internet Protocol v6, the registry server fails to start.

Registry-1151 - Error fetching JavaScript libraries when running in a closed network

When running in a closed network, the Redoc JavaScript libraries do not load properly because they reference a CDN rather than get included or bundled in the application.

Registry-1007 - Registry REST API returns 406 error

The Registry REST API returns a 406 error when the Accept: application/json header is included in the request.
**Registry-711 - Service Registry client does not work with Jersey HTTP client**

When the Jersey and RESTEasy JAX-RS providers are both in the classpath, RESTEasy takes precedence and breaks other HTTP client functionality relying on Jersey client support for the `application/octet-stream` transport, which RESTEasy does not seem to support.

**Service Registry Operator resolved issues**

**Operator-41 - Example CRD should not be empty**

The provided example `ApicurioRegistry` custom resource definition should not be empty.

### 3.6. SERVICE REGISTRY KNOWN ISSUES

**Service Registry core known issues**

**Registry-1610 - Service Registry web console does not properly disable user roles for authorization**

You can configure the Service Registry server to disable role-based authorization. When configured, authentication credentials are required, but roles are ignored, and any authenticated user can perform any action. The web console does not support this, and assumes that if authentication is enabled, role-based authorization is also enabled.

**Registry-1619 - Service Registry server cannot be properly configured to require authentication without role-based authorization**

When role-based authorization is disabled in the Service Registry server, authentication is effectively also disabled. Even when OpenID Connect is enabled in Quarkus, users are not required to provide credentials. If a user provides invalid credentials, a request fails. However, if a user provides no credentials, the request succeeds on behalf of an anonymous user. And because roles are disabled, no additional checking is done.

**Service Registry operator known issues**

**Operator-32 - Operator should support SCRAM authorization without TLS, not only SCRAM+TLS**

The Service Registry Operator should support Salted Challenge Response Authentication Mechanism (SCRAM) authorization without Transport Layer Security (TLS), not only SCRAM+TLS.

**Operator-42 - Auto-generation of OpenShift route may use wrong base host value**

The auto-generation of the Service Registry OpenShift route may use a wrong base host value if there are multiple `routerCanonicalHostname` values.
CHAPTER 4. RED HAT INTEGRATION OPERATORS

Red Hat Integration provides Operators to automate the deployment of Red Hat Integration components on OpenShift. You can use the Red Hat Integration Operator to manage multiple component Operators. Alternatively, you can manage each component Operator individually. This section introduces Operators and provides links to detailed information on how to use Operators to deploy Red Hat Integration components.

4.1. WHAT OPERATORS ARE

Operators are a method of packaging, deploying, and managing a Kubernetes application. They take human operational knowledge and encode it into software that is more easily shared with consumers to automate common or complex tasks.

In OpenShift Container Platform 4.x, the Operator Lifecycle Manager (OLM) helps users install, update, and generally manage the life cycle of all Operators and their associated services running across their clusters. It is part of the Operator Framework, an open source toolkit designed to manage Kubernetes native applications (Operators) in an effective, automated, and scalable way.

The OLM runs by default in OpenShift Container Platform 4.x, which aids cluster administrators in installing, upgrading, and granting access to Operators running on their cluster. The OpenShift Container Platform web console provides management screens for cluster administrators to install Operators, as well as grant specific projects access to use the catalog of Operators available on the cluster.

OperatorHub is the graphical interface that OpenShift cluster administrators use to discover, install, and upgrade Operators. With one click, these Operators can be pulled from OperatorHub, installed on the cluster, and managed by the OLM, ready for engineering teams to self-service manage the software in development, test, and production environments.

Additional resources

- For more information about Operators, see the OpenShift documentation.

4.2. RED HAT INTEGRATION OPERATOR

You can use the Red Hat Integration Operator to install and upgrade multiple Red Hat Integration component Operators:

- 3scale
- 3scale APIcast
- AMQ Broker
- AMQ Interconnect
- AMQ Streams
- API Designer
- Camel K
- Fuse Console
4.2.1. Support life cycle

To remain in a supported configuration, you must deploy the latest Red Hat Integration Operator version. Each Red Hat Integration Operator release version is only supported for 3 months.

Additional resources

- For more details on managing multiple Red Hat Integration component Operators, see Installing the Red Hat Integration Operator on OpenShift.

4.3. RED HAT INTEGRATION COMPONENT OPERATORS

You can install and upgrade each Red Hat Integration component Operator individually, for example, using the 3scale Operator, the Camel K Operator, and so on.

4.3.1. 3scale Operators

- 3scale Operator
- 3scale APIcast Operator

4.3.2. AMQ Operators

- AMQ Broker Operator
- AMQ Interconnect Operator
- AMQ Streams Cluster Operator
- AMQ Online Operator

4.3.3. Camel K Operator

- Camel K Operator - Technology Preview

4.3.4. Fuse Operators

- Fuse on OpenShift - Samples Operator
- Fuse on OpenShift - Fuse Console Operator
- Fuse on OpenShift - API Designer Operator
- Fuse Online Operator

4.3.5. Service Registry Operator

- Service Registry Operator
Additional resources

- For details on managing multiple Red Hat Integration component Operators, see Installing the Red Hat Integration Operator on OpenShift.