Red Hat Integration 2020.Q1

Release Notes for Red Hat Integration 2020.Q1

What's new in Red Hat Integration

Last Updated: 2021-02-19

Red Hat Integration 2020.Q1 Release Notes for Red Hat Integration 2020.Q1

What's new in Red Hat Integration

Legal Notice

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

Red Hat Integration is a unified platform for cloud-native integration and application development with end-to-end API lifecycle support. This platform provides a set of agile and flexible integration and messaging technologies that include:

- API connectivity
- Data transformation
- Service composition and orchestration
- Real-time messaging
- Cross-datacenter message streaming
- API management

Red Hat Integration connects applications, data, and devices across hybrid cloud architectures and delivers API-centric business services.

CHAPTER 2. NEW FEATURES

This section provides a summary of the key new features in Red Hat Integration 2020.Q1 and provides links to more detailed information on new features available in different components.

2.1. KEY FEATURES

Apache Kafka

- Improved data replication across Kafka clusters with MirrorMaker 2.0 in AMQ Streams
- Kafka schema registry enhancements in Service Registry Technology Preview

Data integration

- Change data capture and real-time events with Debezium General Availability
- Container-native and API-based database access with Data Virtualization Technology Preview

OpenAPI 3.0

- Red Hat Integration components support OpenAPI 3.0 and OpenAPI 2.0

2.2. COMPONENT FEATURES

For more details on what's new in Red Hat Integration 2020.Q1 components:

- Red Hat Fuse 7.6 Release Notes
- Red Hat 3scale API Management 2.8 On -Premises Release Notes
- Red Hat 3scale API Management SaaS Release Notes
- Red Hat AMQ 7.6 Product Documentation

CHAPTER 3. DEBEZIUM

Red Hat Integration 2020.Q1 includes a General Availability release of Debezium on OpenShift based on the Debezium open source project. Debezium is a distributed platform that monitors databases and creates change event streams. Debezium is built on Apache Kafka and is deployed and integrated with AMQ Streams.

Debezium captures row-level changes to a database table and passes corresponding change events to AMQ Streams. Applications can read these *change event streams* and access the change events in the order in which they occurred.

3.1. DATABASE CONNECTORS

Debezium provides connectors based on Kafka Connect for the following common databases:

- MySQL
- PostgreSQL
- SQL Server
- MongoDB

When trying out the database connectors, the following database versions are recommended for this release:

Table 3.1. Recommended database versions for Debezium

Database	Version(s)
MySQL	5.7, 8.0
PostgreSQL	10, 11, 12
MongoDB	3.6, 4.0, 4.2
SQL Server	2017

NOTE

For PostgreSQL deployments, you can use the **pgoutput** logical decoding output plugin, which is the default for PostgreSQL versions 10 and later.

Additional resources

- Getting Started with Debezium
- Debezium User Guide

3.2. INSTALLATION OPTIONS

You can install Debezium with AMQ Streams on OpenShift or RHEL:

- Installing Debezium on OpenShift - General Availability
- Installing Debezium on RHEL - Technology Preview

IMPORTANT

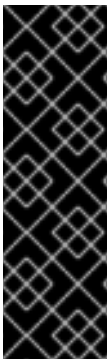
Technology Preview features are not supported with Red Hat production service-level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend implementing any Technology Preview features in production environments.

This Technology Preview feature provides early access to upcoming product innovations, enabling you to test functionality and provide feedback during the development process. For more information about support scope, see Technology Preview Features Support Scope.

CHAPTER 4. DATA VIRTUALIZATION

Data Virtualization is available as a Technology Preview feature in Red Hat Integration 2020.Q1. Data Virtualization is a container-native service, based on the Teiid open source project, that provides integrated access to a range of data sources, including relational databases, MongoDB, and Salesforce, through a single uniform API.

Applications and users connect to a virtual database over standard interfaces (OData REST, or JDBC/ODBC) and can use SQL to interact with the integrated data in the same way that they would interact with a single relational database.



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 implementing any Technology Preview features in production environments.

This Technology Preview feature provides early access to upcoming product innovations, enabling you to test functionality and provide feedback during the development process. For more information about support scope, see Technology Preview Features Support Scope.

4.1. ENHANCEMENTS

Red Hat Integration 2020.Q1 provides the following enhancements for Data Virtualization:

- Improved Data Virtualization Operator deploys virtual databases from Maven artifacts (fat JAR, DDL) or embedded DDL.
- TLS/SSL X.509 certificate encryption of JDBC/ODBC endpoints uses either custom certificates or OpenShift service certificates to encrypt traffic between database clients and a virtual database.
- New monitoring capabilities automatically collect data virtualization metrics and expose them to Prometheus.
- Added data source compatibility.
- New user documentation:
  - Using Data Virtualization
  - Developing Clients for Data Virtualization
  - Data Virtualization Reference

4.2. KNOWN ISSUES

ENTESB-13462 - FORMAT functions for different versions of the Data Virtualization Operator return different results

When you use format functions such as **FORMATBIGDECIMAL** or **FORMATDOUBLE** to query a virtual database, the results that the functions return for negative numbers might not be formatted as expected. The format of the result depends on the underlying Java version specified by the Data

Virtualization Operator. Because recent versions of the Operator switched from using Java 8 to Java 11, you might experience changes in the formatting of your query results.

In Java versions earlier than Java 11, regardless of whether you specify a minus sign character (**-**) as a prefix or suffix of a negative format pattern, the returned negative result string is preceded by a minus sign. However, in Java 11, if you append a minus sign to a negative format pattern, the result string that is returned has the minus sign appended to it.

If you obtain unexpected results for negative numbers that a parsing or formatting function returns, revise the pattern string for the function so that the minus sign precedes the rest of the pattern.

Example:

The following query specifies the negative format pattern, **#,0;#0-**.

```
SELECT FORMATBIGDECIMAL((99 - 10000000), '#,##0;###0-')
```

Depending on the version of the Operator, one of the following results is returned:

DV Operators using Java 8

-9,999,901

DV Operators using Java 11

9,999,901-

To achieve the same result as with Java 8, modify the formatting pattern for the query to **,#0;-#**.

ENTESB-13144 - Redirect URLs set for virtual databases in Red Hat Single Sign-On do not work

Redirects that you configure for the data virtualization service in Red Hat Single Sign-On are not working correctly.

For OData clients to establish secure connections to endpoints of a virtual database, the data virtualization service must be able to authenticate security tokens that it receives from Single Sign-On. Currently, because the data virtualization service is unable to read TLS/SSL certificates from Single Sign-On, attempts to validate the tokens on a secure channel are unsuccessful.

As a temporary workaround, disable secure connections between the data virtualization service and Single Sign-On by adding the following property to the custom resource (CR) for the virtual database, and then redeploy the virtual database:

```
- name: KEYCLOAK_DISABLE_TRUST_MANAGER
  value:  "true"
```

Afterwards, use a tool such as Postman to issue the OData HTTP requests to the service endpoint. Browser clients cannot connect to the endpoint.

IMPORTANT

This setting prevents Data Virtualization from validating SSO certificates over secure HTTPS. However, the certificate itself is exchanged over a secure connection. The unsecured HTTP connection is for validation only, and applies to an internal OpenShift server channel. Single Sign-On continues to transmit the token to the Data Virtualization service over a secure channel. This setting should only be used for development and is not for use in a production environment.

4.3. MIGRATION CONSIDERATIONS

You can migrate virtual database that you developed for JBoss Data Virtualization or Teiid deployments. For information about migrating legacy virtual databases to Data Virtualization, see Using Data Virtualization.

CHAPTER 5. SERVICE REGISTRY

Service Registry is available as a Technology Preview feature in Red Hat Integration 2020.Q1. Service Registry is a datastore for standard event schemas and API designs that is based on the Apicurio Registry open source community project. Developers can use Service Registry to manage and share the structure of their data using a REST interface. For example, client applications can dynamically push or pull the latest updates to or from Service Registry without needing to redeploy.

You can also use Service Registry to create rules that govern how registry content evolves over time. For example, this includes rules for content validation and version compatibility.

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 implementing any Technology Preview features in production environments.

This Technology Preview feature provides early access to upcoming product innovations, enabling you to test functionality and provide feedback during the development process. For more information about support scope, see Technology Preview Features Support Scope.

5.1. ENHANCEMENTS

Service Registry provides the following enhancements in this Technology Preview:

Maven plug-in

Integrates Service Registry functionality with a standard Maven build and provides register and download actions.

Kafka Streams storage

Supports storage in Red Hat AMQ Streams 1.4 or 1.3 using a Kafka Streams-based image.

Kafka Connect

Provides Kafka Connect converters for Apache Avro and Kafka Connect JSON schemas.

Additional serializer/deserializers (SerDes)

- **JSON Schema SerDes** - Supports optional validation during serialization/deserialization.
- **Protobuf SerDes** - Uses the File Descriptor and DynamicMessage support in Google Protobuf to implement serialization/deserialization.

Additional artifact types

- **Protobuf File Descriptor** - An additional artifact type that stores the binary Protobuf File Descriptor format.
- **GraphQL** - GraphQL schema artifacts can be added to Service Registry. Syntax validation is supported.
- **Kafka Connect schema** - Kafka Connect has its own schema format for JSON data, which is similar to Avro. Syntax validation is supported.

Artifact lifecycle states

Artifacts have the following states: ENABLED, DISABLED, and DEPRECATED. The Registry REST API has a new endpoint to manipulate the state of an artifact.

Canonicalization

When searching for an artifact version by content, Service Registry canonicalizes the content so that it does not need to be byte-identical. The content only needs to be functionally equivalent.

New user documentation

- Getting Started with Service Registry
- Managing Schemas with Service Registry and AMQ Streams

CHAPTER 6. RED HAT INTEGRATION OPERATORS

Red Hat Integration provides Operators to enable you to automate the deployment of Red Hat Integration components on OpenShift. This section provides links to detailed information on how to use Operators for different components.

6.1. FUSE OPERATORS

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

6.2. 3SCALE OPERATORS

- 3scale Operator

6.3. AMQ OPERATORS

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

6.4. DATA OPERATORS

- Data Virtualization Operator (Technology Preview)

Additional resources

For more details on Operators in OpenShift:

- Understanding operators in the OpenShift documentation
- OpenShift tech topic on operators