



# Red Hat Integration 2020.q1

## Red Hat Integration 2020.Q1 リリースノート

Red Hat Integration の新機能



# Red Hat Integration 2020.Q1 Red Hat Integration 2020.Q1 リリースノート

---

Red Hat Integration の新機能

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

## 法律上の通知

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Release\_Notes\_for\_Red\_Hat\_Integration\_2020.Q1.ent file |.

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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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.

## 概要

Red Hat Integration プラットフォームについて説明し、本リリースの新機能に関する最新情報を提供します。

---

## 目次

第1章 RED HAT INTEGRATION .....	3
第2章 新機能 .....	4
2.1. 主な特長 .....	4
2.2. コンポーネントの機能 .....	4
第3章 DEBEZIUM .....	5
3.1. データベースコネクタ .....	5
3.2. インストールオプション .....	5
第4章 DATA VIRTUALIZATION .....	7
4.1. 機能拡張 .....	7
4.2. 既知の問題 .....	7
4.3. 移行に関する考慮事項 .....	8
第5章 SERVICE REGISTRY .....	10
5.1. 機能拡張 .....	10
第6章 RED HAT INTEGRATION の OPERATOR .....	12
6.1. FUSE OPERATOR .....	12
6.2. 3SCALE OPERATORS .....	12
6.3. AMQ OPERATOR .....	12
6.4. データ OPERATOR .....	12
関連情報 .....	12



## 第1章 RED HAT INTEGRATION

Red Hat Integration は、エンドツーエンドの API ライフサイクルサポートによるクラウドネイティブ統合およびアプリケーション開発の統合プラットフォームです。このプラットフォームでは、以下を含む企業や柔軟な統合およびメッセージングテクノロジーのセットを提供します。

- API の接続
- データの変換
- サービスの構成とオーケストレーション
- リアルタイムのメッセージング
- データセンター間のメッセージストリーミング
- API 管理

Red Hat Integration は、アプリケーション、データ、およびデバイスをハイブリッドクラウドアーキテクチャー全体で接続し、API 中心のビジネスサービスを提供します。

## 第2章 新機能

本セクションでは、Red Hat Integration 2020.Q1 の主な新機能の概要を説明し、異なるコンポーネントで利用可能な新機能の詳細情報へのリンクを提供します。

### 2.1. 主な特長

#### Apache Kafka

- [AMQ Streams の MirrorMaker 2.0 のある Kafka クラスタ全体](#)の データレプリケーションを向上
- [Service Registry テクノロジープレビュー](#)の Kafka スキーマレジストリーの強化

#### データ統合

- [Debezium の一般公開](#)を使用した変更データキャプチャーおよびリアルタイムイベント
- [Data Virtualization テクノロジープレビュー](#)を使用した Container-native および API ベースのデータベースアクセス

#### OpenAPI 3.0

- Red Hat Integration コンポーネントは OpenAPI 3.0 および OpenAPI 2.0 をサポートしません。

### 2.2. コンポーネントの機能

Red Hat Integration 2020.Q1 コンポーネントの新機能の詳細は、以下を参照してください。

- [Red Hat Fuse 7.6 リリースノート](#)
- [オンプレミス型 Red Hat 3scale API Management 2.8 - オンプレミス型 3scale API Management リリースノート](#)
- [SaaS 版 Red Hat 3scale API Management リリースノート](#)
- [Red Hat AMQ 7.6 の製品ドキュメント](#)

## 第3章 DEBEZIUM

Red Hat Integration 2020.Q1 には、[Debezium](#) オープンソースプロジェクトをベースとした Debezium on OpenShift の一般公開リリースが含まれています。Debezium は、データベースを監視し、変更イベントストリームを作成する分散プラットフォームです。Debezium は Apache Karaf に構築され、AMQ Streams とデプロイおよび統合されます。

Debezium によって、データベーステーブルの行レベルの変更がキャプチャーされ、対応する変更イベントが AMQ Streams に渡されます。アプリケーションはこれらの **変更イベントストリーム** を読み取りでき、変更イベントが発生した順にアクセスできます。

### 3.1. データベースコネクタ

Debezium は、以下の共通データベースの Kafka Connect をベースとしたコネクタを提供します。

- MySQL
- PostgreSQL
- SQL Server
- MongoDB

データベースコネクタを試す場合は、本リリースに以下のデータベースバージョンが推奨されます。

表3.1 Debezium の推奨されるデータベースバージョン

データベース	バージョン
MySQL	5.7、8.0
PostgreSQL	10, 11, 12
MongoDB	3.6, 4.0, 4.2
SQL Server	2017



#### 注記

PostgreSQL デプロイメントの場合は、PostgreSQL バージョン 10 以降のデフォルトである **pgoutput** 論理デコード出力プラグインを使用できます。

#### 関連情報

- [Getting Started with Debezium](#)
- [Debezium ユーザーガイド](#)

### 3.2. インストールオプション

AMQ Streams で Debezium を OpenShift または RHEL にインストールできます。

- [『Installing Debezium on OpenShift』](#) (GA、一般提供)
- [『Installing Debezium on RHEL』](#) (テクノロジープレビュー)



## 重要

テクノロジープレビューの機能は、Red Hat の本番環境のサービスレベルアグリーメント (SLA) ではサポートされず、機能的に完全ではないことがあります。Red Hat は、本番環境でのテクノロジープレビュー機能の実装は推奨しません。

テクノロジープレビューの機能は、最新の技術をいち早く提供して、開発段階で機能のテストやフィードバックの収集を可能にするために提供されます。サポート範囲の詳細は、[テクノロジープレビュー機能のサポート範囲](#) を参照してください。

## 第4章 DATA VIRTUALIZATION

Data Virtualization は、Red Hat Integration 2020.Q1 ではテクノロジープレビューとして利用できません。Data Virtualization は [Teiid](#) オープンソースプロジェクトをベースとする Container-native サービスで、1つの統一された API 経由でリレーショナルデータベース、MongoDB、Salesforce など、さまざまなデータソースへの統合アクセスを提供します。

アプリケーションおよびユーザーは、標準のインターフェース（OData REST または JDBC/ODBC）で仮想データベースに接続し、SQL を使用して単一のリレーショナルデータベースと対話するのと同じ方法で統合データと対話できます。



### 重要

テクノロジープレビュー機能は、Red Hat の実稼働サービスレベルアグリーメント (SLA) でサポートされておらず、機能的に完全でない可能性があります。Red Hat は、本番環境でのテクノロジープレビュー機能の実装は推奨しません。

テクノロジープレビューの機能は、最新の技術をいち早く提供して、開発段階で機能のテストやフィードバックの収集を可能にするために提供されます。サポート範囲の詳細は、[テクノロジープレビュー機能のサポート範囲](#) を参照してください。

### 4.1. 機能拡張

Red Hat Integration 2020.Q1 は、Data Virtualization に以下の拡張機能を提供します。

- 改善された Data Virtualization Operator は、Maven アーティファクト（JAR、DDL）または組み込み DDL から仮想データベースをデプロイします。
- JDBC/ODBC エンドポイントの TLS/SSL X.509 証明書暗号化は、カスタム証明書または OpenShift サービス証明書を使用して、データベースクライアントと仮想データベース間のトラフィックを暗号化します。
- 新しいモニタリング機能は、データ仮想化メトリクスを自動的に収集し、Prometheus に公開します。
- データソースの互換性を追加。
- 新しいユーザードキュメント：
  - [データ仮想化の使用](#)
  - [Data Virtualization のクライアントの開発](#)
  - [Data Virtualization のリファレンス](#)

### 4.2. 既知の問題

#### [ENTESB-13462](#) - FORMAT functions for different versions of the Data Virtualization Operator return different results

**FORMATBIGDECIMAL** や **FORMATDOUBLE** などのフォーマット機能を使用して仮想データベースのクエリーを行う場合、関数が負の値の戻り値を期待どおりにフォーマットしない場合があります。結果の形式は、Data Virtualization Operator によって指定される基礎となる Java バージョンによって異なります。新しいバージョンの Operator は Java 8 から Java 11 に切り替えられたため、クエリー結果のフォーマットに変更が加えられる可能性があります。

Java 11 よりも前のバージョンの Java では、負の形式パターンの接頭辞または接尾辞としてマイナス記号(-)を指定するかどうかに関わらず、返される負の結果文字列の前にマイナス記号が追加されます。ただし、Java 11 では、負の形式パターンにマイナス記号を追加すると、返される結果文字列に minus 記号が追加されます。

解析関数またはフォーマット関数が返す負の値で予期しない結果を取得する場合は、minus 記号がパターンの残りの部分より優先されるように、関数のパターン文字列を訂正します。

例:

以下のクエリーは、負の形式パターン（**#**、**0**、**#0-**）を指定します。

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

Operator のバージョンによっては、以下のいずれかの結果が返されます。

Java 8 を使用する DV Operator

```
-9,999,901
```

Java 11 を使用する DV Operator

```
9,999,901-
```

Java 8 と同じ結果を実現するには、クエリーのフォーマットパターンを **#0;#** に変更します。

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

Red Hat Single Sign-On で Data Virtualization サービスに設定したリダイレクトが正しく機能しません。

OData クライアントが仮想データベースのエンドポイントへのセキュアな接続を確立できるようにするには、Data Virtualization サービスが Single Sign-On から受信するセキュリティトークンを認証する必要があります。現時点で、Data Virtualization サービスはシングルサインオンから TLS/SSL 証明書を読み取ることができないため、セキュアなチャンネルでトークンの検証を試行することはできません。

一時的な回避策として、以下のプロパティを仮想データベースのカスタムリソース(CR)に追加して、Data Virtualization サービスとシングルサインオンのセキュアな接続を無効にし、仮想データベースを再デプロイします。

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

その後、Postman などのツールを使用して OData HTTP リクエストをサービスエンドポイントに対して発行します。ブラウザークライアントはエンドポイントに接続できません。



### 重要

この設定により、Data Virtualization がセキュアな HTTPS で SSO 証明書を検証しないようにします。ただし、証明書自体はセキュアな接続で交換されます。セキュアでない HTTP 接続は検証のみを目的としており、内部 OpenShift サーバーチャンネルに適用されます。シングルサインオンは、セキュアなチャンネルを介してトークンを Data Virtualization サービスに送信します。この設定は開発目的にのみご利用いただく機能で、実稼働環境での使用には使用しないでください。

## 4.3. 移行に関する考慮事項

JBoss Data Virtualization または Teiid デプロイメント向けに開発した仮想データベースを移行できます。従来の仮想データベースを Data Virtualization に移行する方法は、「[Using Data Virtualization](#)」を参照してください。

## 第5章 SERVICE REGISTRY

Service Registry は、Red Hat Integration 2020.Q1 ではテクノロジープレビューとして利用できます。Service Registry は、[Apicurio Registry](#) オープンソースコミュニティプロジェクトをベースとする標準イベントスキーマおよび API デザインのデータストアです。開発者は Service Registry を使用して REST インターフェースを使用してデータの構造を管理および共有できます。たとえば、クライアントアプリケーションは、再デプロイせずに最新の更新を Service Registry に動的にプッシュまたはプルできます。

Service Registry を使用して、レジストリーコンテンツが時間の経過と共にどのように進化するかを制御するルールを作成することもできます。たとえば、これには、コンテンツの検証およびバージョンの互換性のルールが含まれます。



### 重要

テクノロジープレビュー機能は、Red Hat の実稼働サービスレベルアグリーメント (SLA) でサポートされておらず、機能的に完全でない可能性があります。Red Hat は、本番環境でのテクノロジープレビュー機能の実装は推奨しません。

テクノロジープレビューの機能は、最新の技術をいち早く提供して、開発段階で機能のテストやフィードバックの収集を可能にするために提供されます。サポート範囲の詳細は、[テクノロジープレビュー機能のサポート範囲](#) を参照してください。

### 5.1. 機能拡張

Service Registry は、このテクノロジープレビューで以下の拡張機能を提供します。

#### Maven プラグイン

Service Registry 機能を標準の Maven ビルドと統合し、登録およびダウンロードのアクションを提供します。

#### Kafka Streams ストレージ

Kafka Streams ベースのイメージを使用して Red Hat AMQ Streams 1.4 または 1.3 でストレージをサポートします。

#### Kafka Connect

Apache Avro および Kafka Connect JSON スキーマの Kafka Connect コンバーターを提供します。

#### 追加のシリアライザー/デシリアライザー(SerDes)

- **JSON Schema SerDes:** シリアル化/デシリアライズ中のオプションの検証をサポートします。
- **Protobuf SerDes -** Google Protobuf の File Descriptor and DynamicMessage サポートを使用し、シリアル化/デシリアライズを実装します。

#### 追加のアーティファクトタイプ

- **Protobuf File Descriptor -** バイナリー Protobuf File Descriptor 形式を格納する追加のアーティファクトタイプ。
- **GraphQL -** GraphQL スキーマアーティファクトは Service Registry に追加できます。構文の検証がサポートされています。
- **Kafka Connect スキーマ:** Kafka Connect には、Avro と同様の JSON データ用の独自のスキーマ形式があります。構文の検証がサポートされています。

### アーティファクトのライフサイクルの状態

アーティファクトの状態は ENABLED、DISABLED、および DEPRECATED です。Registry REST API には、アーティファクトの状態を操作する新しいエンドポイントがあります。

### 正規化

コンテンツでアーティファクトバージョンを検索する場合、Service Registry はコンテンツを正規化し、バイト識別は必要ありません。コンテンツは機能的にのみ機能的に指定する必要があります。

### 新しいユーザードキュメント

- [『Getting Started with Service Registry』](#)
- [Service Registry および AMQ Streams でのスキーマの管理](#)

## 第6章 RED HAT INTEGRATION の OPERATOR

Red Hat Integration は Operator を提供し、OpenShift で Red Hat Integration コンポーネントのデプロイメントを自動化できます。このセクションでは、さまざまなコンポーネントに Operator を使用方法について説明します。

### 6.1. FUSE OPERATOR

- [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 OPERATOR

- [AMQ Broker Operator](#)
- [AMQ Interconnect Operator](#)
- [AMQ Streams Cluster Operator](#)
- [AMQ Online Operator](#)

### 6.4. データ OPERATOR

- [Data Virtualization Operator \(テクノロジープレビュー\)](#)

#### 関連情報

OpenShift の Operator の詳細は、以下を参照してください。

- [OpenShift ドキュメントの「Operator」](#)
- [OpenShift tech topic on operator](#)