



## Red Hat Integration 2021.Q2

# Red Hat Integration 2021.Q2 リリースノート

Red Hat Integration の新機能



# Red Hat Integration 2021.Q2 Red Hat Integration 2021.Q2 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\_2021.Q2.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 本リリースの新機能 .....	4
2.1. インテグレーションの新機能 .....	4
第3章 CAMEL QUARKUS リリースノート .....	5
3.1. CAMEL QUARKUS の機能 .....	5
3.1.1. プラットフォームおよびコアコンポーネントのバージョン .....	5
3.1.2. テクノロジープレビューの機能 .....	5
3.1.3. テクノロジープレビューの Camel Quarkus エクステンション .....	6
第4章 SERVICE REGISTRY リリースノート .....	7
4.1. SERVICE REGISTRY のインストールオプション .....	7
4.2. SERVICE REGISTRY の新機能 .....	7
Service Registry のセキュリティー .....	7
Service Registry コア .....	7
Service Registry データストレージ .....	8
Service Registry v2 REST API .....	8
Service Registry Operator .....	8
Service Registry プラットフォームコンポーネントのバージョン .....	8
Service Registry ユーザーのドキュメントおよび例 .....	9
4.3. SERVICE REGISTRY で削除された機能 .....	9
4.4. SERVICE REGISTRY データの移行およびアップグレード .....	9
4.5. SERVICE REGISTRY で解決された問題 .....	9
Service Registry core で解決された問題 .....	9
Service Registry Operator で解決された問題 .....	10
4.6. SERVICE REGISTRY の既知の問題 .....	10



## 第1章 RED HAT INTEGRATION

Red Hat Integration は、ハイブリッド環境およびマルチクラウド環境全体でコンテナベースの統合サービスを作成、拡張、デプロイするための包括的な統合およびイベント処理技術です。Red Hat Integration は、デジタル環境で必要となるアプリケーションとシステム間でデータを接続および共有するために組織が使用できる、アジャイルで API 中心の分散ソリューションを提供します。

Red Hat Integration には、以下の機能が含まれています。

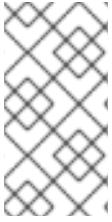
- API の接続
- データの変換
- サービスの設定とオーケストレーション
- リアルタイムのメッセージング
- データセンター間のメッセージストリーミング
- API 管理

### その他のリソース

- [エンタープライズ統合について理解する](#)

## 第2章 本リリースの新機能

ここでは、Red Hat Integration 2021.Q2 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へのリンクを提供します。



### 注記

本リリースノートには、Red Hat Integration 2021.Q2 でのみ更新されたコンポーネントの詳細が含まれています。Debezium や Camel K などの他のコンポーネントの最新バージョンに関する詳細は、[Red Hat Integration リリースノート 2021-Q1](#) を参照してください。

### 2.1. インテグレーションの新機能

#### Camel Quarkus エクステンション

- Camel Quarkus [テクノロジープレビュー](#) の Camel Quarkus エクステンションとして利用可能。

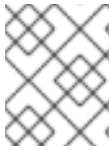
#### Kafka スキーマレジストリー

- [Service Registry 2.0 のテクノロジープレビュー](#) の OpenShift Operator での改善された Apache Kafka スキーマおよび API レジストリー。



## 第3章 CAMEL QUARKUS リリースノート

Camel Quarkus は、Red Hat Integration 2021.Q2 のテクノロジープレビューコンポーネントとして使用できます。



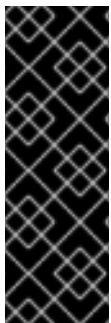
### 注記

このテクノロジープレビューでは、Camel Quarkus は JVM モードでのみサポートされます。

Camel Quarkus は、Apache Camel とその vast コンポーネントライブラリーの統合機能を Quarkus ランタイムに提供します。Camel Quarkus を使用すると、Quarkus が提供するパフォーマンス上の利点、開発者としての楽しさ、コンテナを第一とする精神を活用できます。

Camel Quarkus は、Quarkus ディストリビューションのユニットである Quarkus エクステンションを提供します。エクステンションは、Quarkus アプリケーションで Camel コンポーネントを設定、起動、および統合し、通常プロジェクトの依存関係として設定されます。

Camel Quarkus は、Camel 3 に追加された多くのパフォーマンスパフォーマンスの改善点を活用します。これにより、メモリーフットプリントが削減され、リフレクションへの依存が減り、起動時間が短縮されます。



### 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。

## 3.1. CAMEL QUARKUS の機能

テクノロジープレビューの Camel Quarkus は、以下の主要機能を提供します。

### 3.1.1. プラットフォームおよびコアコンポーネントのバージョン

- OpenShift Container Platform 4.6
- Quarkus 1.11 Java runtime
- Apache Camel 3.7
- Apache Camel Quarkus 1.6
- OpenJDK 11

### 3.1.2. テクノロジープレビューの機能

#### 高速起動および低 RSS メモリー

Quarkus の最適化されたビルド時機能および事前 (AOT) コンパイル機能を使用すると、Camel アプリケーションはビルド時に事前設定できるため、起動時間が短縮されます。

## 優れた設定可能性

Camel Quarkus アプリケーションの重要な要素はすべて、CDI(Contexts and Dependency Injection) または設定プロパティを使用してプログラムで設定することができます。デフォルトでは、CamelContext が設定され、自動的に起動されます。

アプリケーションのブートストラップおよび設定のさまざまな方法については、[Configuring your Quarkus applications](#) ガイドを参照してください。

## 既存の Quarkus エクステンションとの統合

Quarkus は、ネイティブサポートや設定オプションを継承する Camel コンポーネントによって使用されるライブラリーおよびフレームワークのエクステンションを提供します。

### 3.1.3. テクノロジープレビューの Camel Quarkus エクステンション

- camel-quarkus-main
- camel-quarkus-timer
- camel-quarkus-log
- camel-quarkus-file
- camel-quarkus-ftp
- camel-quarkus-direct
- camel-quarkus-bean
- camel-quarkus-mock
- camel-quarkus-bindy
- camel-quarkus-seda
- camel-quarkus-core
- camel-quarkus-microprofile-health

## その他のリソース

- [Camel Quarkus スタートガイド](#)

## 第4章 SERVICE REGISTRY リリースノート

Red Hat Integration - Service Registry 2.0 は、Red Hat Integration 2021.Q2 ではテクノロジープレビューのコンポーネントとして使用できます。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。

Service Registry を使用して、Web コンソール、REST API、Maven プラグイン、または Java クライアント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スキーマ更新を Service Registry に動的にプッシュまたはプルできます。また、Service Registry を使用して任意のルールを作成し、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こともできます。たとえば、これには、コンテンツ検証のルールやスキーマまたは API バージョンの後方互換性と前方互換性に関するルールが含まれます。



### 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。

### 4.1. SERVICE REGISTRY のインストールオプション

以下のデータストレージオプションを使用して、Service Registry を OpenShift にインストールできます。

表4.1 Service Registry ストレージのオプション

ストレージオプション	リリースカテゴリ
AMQ Streams 1.7	テクノロジープレビュー
PostgreSQL 12 database	テクノロジープレビュー

### 4.2. SERVICE REGISTRY の新機能

Service Registry 2.0 には、以下の更新が含まれます。

#### Service Registry のセキュリティ

- **Red Hat Single Sign-On に基づく認証** オプションでレジストリーを保護するため、REST API でユーザーを認証する必要があります (Basic および OAuth はサポートされます)。
- **ロールベースの承認**: 認証が有効になっている場合、ユーザーには少なくとも **sr-admin**、**sr-developer**、または **sr-readonly** ロールの1つが必要です。
- **作成者のみの承認** - 認証されたユーザーがアーティファクトを元々作成しない限り、アーティファクトへの変更が阻止されるオプション。

#### Service Registry コア

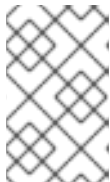
- **レジストリーアーティファクトグループ** - オプションでスキーマと API アーティファクトをカスタムの名前付きの論理グループへ整理
- **リファクタリングされた Kafka シリアライザー/デシリアライザー (SerDe) クラス** - 使いやすく、一貫性、機能に対処するための Java SerDe レイヤーへの大幅な更新
- **イベントソーシング** - CloudEvents 仕様に基づいて変更が行われるたびにイベントを実行するようにレジストリーを設定するオプション。

## Service Registry データストレージ

- **SQL ベースのストレージ** - PostgreSQL データベースをサポートする新しい SQL ストレージ実装
- **Kafka ベースのストレージ** - AMQ Streams を使用してアーティファクトデータと組み込み SQL データベースを保存し、メモリーに表す組み込み SQL データベースを使用した新しいハイブリッドストレージ

## Service Registry v2 REST API

- **カスタムバージョン管理** - REST API を使用してアーティファクトを作成または更新する際に、カスタムバージョン番号を提供するオプション
- **アーティファクト検索の改善** - REST API への更新により、アーティファクトの検索が改善
- **API のインポート/エクスポート** - レジストリーデータを **.zip** 形式でエクスポートおよびインポートする操作が含まれる REST API への更新
- **CNCF Schema Registry API のサポート** - Cloud Native Computing Foundation Schema Registry REST API の実装



### 注記

Service Registry v2 REST API は、新しいアーティファクトグループを含まない Confluent スキーマレジストリー REST API と互換性があります。後方互換性は、既存の Service Registry v1 REST API で維持されます。

## Service Registry Operator

- **パフォーマンスおよび合理化の向上** - Operator は OpenShift での **Deployment** ( **DeploymentConfig** ではなく)、予測可能なリソース命名 (ランダムな接尾辞なし)、および並行して作成されるリソースを使用します。
- **レジストリーデータストレージ** - 新しい SQL および Kafka ベースのストレージオプションのサポート。
- **レジストリーセキュリティー** - Red Hat Single Sign-On を使用した認証および承認設定のサポート。
- **ApicurioRegistry CRD v1- status** ブロックの標準化された **conditions** フィールドを使用して、Operator またはアプリケーションの問題またはエラーをより適切に示します。

## Service Registry プラットフォームコンポーネントのバージョン

- OpenShift Container Platform 4.6 または 4.7
- OpenJDK 11

- AMQ Streams 1.7
- PostgreSQL 12
- Debezium 1.4
- Camel Kafka Connector Technology Preview

### Service Registry ユーザーのドキュメントおよび例

- バージョン 2.0 の新機能で更新された新しいドキュメントライブラリー:
  - [Installing and deploying Service Registry on OpenShift](#)
  - [Service Registry User Guide](#)
  - [Registry v2 core REST API ドキュメント](#)
- 新しいオープンソースデモアプリケーション:
  - <https://github.com/Apicurio/apicurio-registry-examples>

## 4.3. SERVICE REGISTRY で削除された機能

- キャッシュベースの Infinispan ストレージオプションが削除されました。
- Java Persistence API (JPA) ストレージオプションが、新しい PostgreSQL データベースストレージオプションに置き換えられました。
- AMQ Streams の Kafka ベースのストレージオプションは、インメモリー H2 データベースを使用した AMQ Streams の新しいハイブリッドストレージオプションに置き換えられました。
- Service Registry Java クライアントは OpenJDK 8 に対応しなくなり、代わりに OpenJDK 11 をサポートするようになりました。

## 4.4. SERVICE REGISTRY データの移行およびアップグレード

Service Registry バージョン 2.x インスタンス間でのレジストリーデータの移行に関する詳細は、[Exporting and importing registry content using the Registry REST API](#) を参照してください。

Service Registry バージョン 1.x から 2.x に移行およびアップグレードするためのオープンソースツールの詳細については、[Apicurio Registry export utility for 1.x versions](#) を参照してください。

## 4.5. SERVICE REGISTRY で解決された問題

Service Registry 2.0 では、次の問題が解決されました。

### Service Registry core で解決された問題

#### [Registry-1289](#) - Registry does not work on IPv6

Internet Protocol v6 を使用する Kubernetes サーバーに Service Registry をデプロイしようとする、レジストリーサーバーが起動に失敗します。

#### [Registry-1151](#) - Error fetching JavaScript libraries when running in a closed network

閉じられたネットワークで実行する場合、Redoc JavaScript ライブラリーは、アプリケーションに組み込まれたりバンドルされたりするのではなく、CDN を参照するため、正しく読み込まれません。

### Registry-1007 - レジストリー REST API が 406 エラーを返す

リクエストに **Accept: application/json** ヘッダーが含まれる場合、Registry REST API は **406** エラーを返します。

### Registry-711 - Service Registry client does not work with Jersey HTTP client

Jersey プロバイダーと RESTEasy JAX-RS プロバイダーの両方がクラスパスにある場合、RESTEasy が優先され、RESTEasy がサポートしないように見受けられる **application/octet-stream** トランスポートの Jersey クライアントのサポートに依存する他の HTTP クライアント機能が中断されます。

### Service Registry Operator で解決された問題

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

提供された例の **ApicurioRegistry** カスタムリソース定義を空にしない必要があります。

## 4.6. SERVICE REGISTRY の既知の問題

Service Registry 2.0 の既知の問題は以下のとおりです。

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

Service Registry Operator は、SCRAM+TLS だけでなく、Transport Layer Security (TLS) を使用せずに Salted Challenge Response Authentication Mechanism (SCRAM) をサポートする必要があります。

### Operator-42 - OpenShift ルートの自動生成で間違ったベースホスト値が使用される場合がある

複数の **routerCanonicalHostname** 値がある場合、Service Registry OpenShift ルートの自動生成で間違ったベースホスト値が使用される可能性があります。