



Red Hat Integration 2021.Q4

Red Hat Integration 2021.Q4 リリースノート

Red Hat Integration の新機能

Red Hat Integration の新機能

法律上の通知

Copyright © 2023 Red Hat, Inc.

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[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

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

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] 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[®] 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ついて最新情報を提供します。

目次

第1章 RED HAT INTEGRATION	3
第2章 QUARKUS リリースノートの CAMEL エクステンション	4
2.1. QUARKUS 機能の CAMEL エクステンション	4
2.2. サポートされるプラットフォーム、設定、データベース、およびエクステンション	4
2.3. テクノロジープレビューのエクステンション	4
2.4. 既知の問題	4
2.5. 重要事項	5
2.6. その他のリソース	6
第3章 CAMEL K リリースノート	7
3.1. CAMEL K の新機能	7
3.2. サポートされる構成	7
3.3. 重要事項	8
3.4. サポートされる CAMEL QUARKUS エクステンション	8
3.5. サポート対象の KAMELETS	11
3.6. CAMEL K の既知の問題	14
3.7. CAMEL K の修正された問題	15
第4章 RED HAT INTEGRATION の OPERATOR	18
4.1. OPERATOR とは	18
4.2. RED HAT INTEGRATION OPERATOR	18
4.3. RED HAT INTEGRATION コンポーネント OPERATOR	20

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。

- 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 の接続
- アプリケーションコネクタ
- エンタープライズ統合パターン
- API 管理
- データの変換
- サービスの構成とオーケストレーション

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 QUARKUS リリースノートの CAMEL エクステンション

2.1. QUARKUS 機能の CAMEL エクステンション

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。

アプリケーションジェネレーター

[Quarkus アプリケーションジェネレーター](#) を使用して、アプリケーションをブートストラップし、エクステンションエコシステムを検出します。

優れた設定可能性

Quarkus アプリケーションの Camel エクステンションの重要な側面はすべて、CDI (Contexts and Dependency Injection) または設定プロパティーを使用してプログラマ的に設定できます。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 の Camel エクステンションは、ネイティブサポートおよび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。

2.2. サポートされるプラットフォーム、設定、データベース、およびエクステンション

- Quarkus バージョン 2.2 の Camel エクステンションにおけるサポート対象のプラットフォーム、設定、およびデータベースの詳細は、カスタマーポータル [の Supported Configuration](#) ページを参照してください (ログインが必要です)。
- Quarkus エクステンションの Red Hat Camel エクステンションおよび各エクステンションの Red Hat サポートレベルの一覧は、[Camel Extensions for Quarkus Reference](#) の [Extensions Overview](#) の章を参照してください (ログインが必要です)。

2.3. テクノロジープレビューのエクステンション

Red Hat は、Quarkus の Camel エクステンションの今回のリリースで提供されるテクノロジープレビューのコンポーネントに対するサポートを提供していません。[Camel Extensions for Quarkus Reference](#) の [Extensions Overview](#) の章でテクノロジープレビューに指定された項目については、「テクノロジープレビュー機能のサポート範囲」で定義されているように、サポート範囲が限定されています。

2.4. 既知の問題

CAMEL-17158 AWS2 SQS 遅延のあるキューにメッセージを送信するときに、遅延は考慮されません。

遅延のあるキューを作成する場合、`camel-aws2-sqs` コンポーネントをプロデューサーとして使用して送信されるメッセージは、キューに設定されている遅延を考慮しません。

この動作の理由は、キュー設定をオーバーライドするメッセージを送信するときに、Camel がデフォルトの遅延として '0秒' を設定するためです。

回避策として、Camel プロデューサーを使用する場合は同じ遅延を設定する必要があります。たとえば、5 秒の遅延でキューを作成する場合は、**camel-aws2-sqs** プロデューサーを使用する際に 5 秒の遅延を設定する必要もあります。

ENTESB-17763 Missing productised transitive deps of camel-quarkus-jira extensions

camel-quarkus-jira エクステンションを使用するアプリケーションは、追加の Maven リポジトリー <https://packages.atlassian.com/maven-external/> を Maven **settings.xml** ファイルやアプリケーションプロジェクトの **pom.xml** ファイルのいずれかに設定する必要があります。

2.5. 重要事項

CVEListenerExternal log4j-core: Log4j 2.x のリモートコード実行

Log4j 2.x のセキュリティー問題 [CVE-2021-44228](#) (通称 Log4Shell) が修正された Quarkus の Camel Extensions のパッチが当てられたバージョン (バージョン 2.2.0-1) は、[Quarkus Platform](#) を通じてまもなく利用可能になります。それまでの間、Camel Quarkus のユーザーは、以下の 2 つの条件のいずれかが当てはまらない限り、Log4Shell の影響を受けません。

1. **camel-quarkus-corda** または **camel-quarkus-nsq** をアプリケーションで使用している場合。これらのエクステンションは **log4j-core** に推移的に依存しますが、Quarkus で適切に機能する必要はありません。これは、JBoss log-manager が Quarkus のログインバックエンドとして使用されるためです。そのため、これらの依存関係から **log4j-core** を除外することが最善の軽減策になります。

```
<dependency>
  <groupId>org.apache.camel.quarkus</groupId>
  <artifactId>camel-quarkus-nsq</artifactId>
  <exclusions>
    <exclusion>
      <groupId>org.apache.logging.log4j</groupId>
      <artifactId>log4j-core</artifactId>
    </exclusion>
  </exclusions>
</dependency>

<dependency>
  <groupId>org.apache.camel.quarkus</groupId>
  <artifactId>camel-quarkus-corda</artifactId>
  <exclusions>
    <exclusion>
      <groupId>org.apache.logging.log4j</groupId>
      <artifactId>log4j-core</artifactId>
    </exclusion>
  </exclusions>
</dependency>
```

camel-quarkus-corda または **camel-quarkus-nsq** は、Red Hat によって **サポートされない** ため、これらのエクステンションの使用は自己責任となることに注意してください。

2. アプリケーションが **org.apache.logging.log4j:log4j-core** に直接依存している場合。この場合、Log4Shell の脆弱性が修正されている最新の log4j バージョン (執筆時点では 2.16.0) にアップグレードしてください。

ドキュメントの可用性

Red Hat ビルドの Quarkus バージョン 2.2 に設定したドキュメントには、Red Hat build of

Quarkus のリリースで提供されるガイドの完全なセットが含まれていません。そのため、Quarkus の Camel エクステンションセットのこのドキュメントには、代わりに **Red Hat ビルドの Quarkus 1.11** ドキュメントへのリンクが含まれています。

2.6. その他のリソース

- [Supported Configurations](#)
- [Camel Extensions for Quarkus](#)
- [Camel Extensions for Quarkus の使用](#)
- [Developing Applications with Camel Extensions for Quarkus](#)

第3章 CAMEL K リリースノート

Camel K は、OpenShift のクラウドでネイティブで実行される Apache Camel K からビルドされる軽量 of インテグレーションフレームワークです。Camel K は、サーバーレスおよびマイクロサービスアーキテクチャー向けに特別に設計されています。Camel K を使用すると、Camel Domain Specific Language (DSL) で書かれたインテグレーションコードを直接 OpenShift で即座に実行することができます。

Camel K を OpenShift Serverless および Knative とともに使用すると、コンテナは必要な場合のみ自動的に作成され、負荷時の自動スケーリングやゼロへのスケーリングが行われます。これにより、サーバーのプロビジョニングとメンテナンスのオーバーヘッドがなくなり、アプリケーションの開発に集中することができます。

Camel K を OpenShift Serverless および Knative Eventing とともに使用すると、システムのコンポーネントがサーバーレスアプリケーションのイベント駆動型アーキテクチャーで通信する方法を管理できます。これにより、イベントプロデューサーとコンシューマー間の関係が切り離されたパブリッシュ/サブスクライブモデルまたはイベントストリーミングモデルを使用すると、柔軟性を提供し、効率化を実現できます。

3.1. CAMEL K の新機能

Camel K は、以下の主要機能でクラウドネイティブインテグレーションを提供します。

- 自動スケーリングおよびゼロへのスケーリングを行うための Knative Serving
- イベント駆動型アーキテクチャーのための Knative Eventing
- デフォルトで Quarkus ランタイムを使用するパフォーマンスの最適化
- Java または YAML DSL で書かれた Camel インテグレーション
- OpenShift で Prometheus を使用したインテグレーションのモニタリング
- クイックスタートチュートリアル
- AWS、Jira、Salesforce などの外部システムへのコネクタ用 Kamelet Catalog
- Timer および Log Kamelets のサポート

3.2. サポートされる構成

Camel K のサポートされる構成、標準仕様、およびコンポーネントに関する詳細は、以下のカスタマーポータルの記事を参照してください。

- [Camel K Supported Configurations](#)
- [Camel K Component Details](#)

3.2.1. Camel K Operator のメタデータ

Camel K には、OpenShift OperatorHub から Camel K をインストールするために使用される、更新された Operator メタデータが含まれています。この Operator メタデータには、OpenShift Container Platform 4.6 以降との使用を目的に設計されたリリースパッケージの Operator バンドル形式が含まれます。

その他のリソース

- [Operator バンドルフォーマットに関する OpenShift ドキュメント](#)

3.3. 重要事項

Red Hat Integration - Camel K リリースにおける重要事項

CVEListenerExternal log4j-core: Log4j 2.x のリモートコード実行

Log4j 2.x のセキュリティ問題である [CVE-2021-44228](#) (通称 Log4Shell) に対処するために、パッチが適用されたバージョンの Camel K (version 1.6.0-1) がリリースされました。このパッチが適用されたバージョンを選択するために Camel K デプロイメントおよびアプリケーションプロジェクトを更新するには、[第 4 章のアップグレード手順に従います](#)。『[Getting Started with Camel K](#)』の「[Upgrading Camel K](#)」を参照してください。パッチが適用されたバージョンの Camel K は **1.6.x** Operator チャネルを介して配信されます。

Camel K でサポートされるエンタープライズ統合パターン (EIP)

以下を除くすべての Camel 3 [EIP パターン](#) は、Camel K で完全にサポートされています。

- サークットブレーカー
- Saga
- Change Data Capture の変更

YAML DSL の制限

YAML DSL インテグレーションは Camel K 1.6 でサポートされていますが、誤った YAML DSL コードのエラーメッセージは引き続き開発中です。

JAVA DSL の制限

Camel K 1.6 の Java DSL は単一のクラス/設定メソッドに限定され、すべてのユーティリティはサードパーティー JARS で提供する必要があります。Camel K の依存関係の自動サポートのためには、エンドポイント URI はエンドポイント文字列に直接定義する必要があります。そうでないと、依存関係をモードラインに指定する必要があります。

XML DSL はサポートされません。

XML DSL は Camel K 1.6 ではサポートされません。

Camel K 1.6 ランタイムは、HTTPS をサポートする Maven リポジトリにのみアクセスできます。

HTTPS によってセキュア化された Maven リポジトリのみを使用できます。安全ではない HTTP プロトコルに対応しなくなりました。

3.4. サポートされる CAMEL QUARKUS エクステンション

ここでは、Camel K の本リリースでサポートされる Camel Quarkus エクステンションの一覧が記載されています (Camel K アプリケーション内で使用される場合のみ)。



注記

これらの Camel Quarkus エクステンションは、Camel K アプリケーション内で使用される場合にのみサポートされます。これらの Camel Quarkus エクステンションは、スタンドアロンモード (Camel K なしで) で使用する場合にはサポートされません。

3.4.1. サポートされる Camel Quarkus コネクタエクステンション

以下の表は、本リリースの Camel K でサポートされる Camel Quarkus コネクタエクステンションを示しています (Camel K アプリケーション内で使用される場合のみ)。

名前	パッケージ
AWS 2 Kinesis	camel-quarkus-aws2-kinesis
AWS 2 Lambda	camel-quarkus-aws2-lambda
AWS 2 S3 Storage Service	camel-quarkus-aws2-s3
AWS 2 Simple Notification System (SNS)	camel-quarkus-aws2-sns
AWS 2 Simple Queue Service (SQS)	camel-quarkus-aws2-sqs
File	camel-quarkus-file
FTP	camel-quarkus-ftp
FTPS	camel-quarkus-ftp
SFTP	camel-quarkus-ftp
HTTP	camel-quarkus-http
JMS	camel-quarkus-jms
Kafka	camel-quarkus-kafka
Kamelets	camel-quarkus-kamelet
メトリクス	camel-quarkus-microprofile-metrics
MongoDB	camel-quarkus-mongodb
Salesforce	camel-quarkus-salesforce
SQL	camel-quarkus-sql
Timer	camel-quarkus-timer

3.4.2. サポートされる Camel Quarkus データフォーマットエクステンション

以下の表は、本リリースの Camel K でサポートされる Camel Quarkus データフォーマットエクステンションを示しています (Camel K アプリケーション内で使用される場合のみ)。

名前	パッケージ
Avro	camel-quarkus-avro

名前	パッケージ
Bindy (CSV 用)	camel-qaurkus-bindy
JSON Jackson	camel-quarkus-jackson
Jackson Avro	camel-quarkus-jackson-avro

3.4.3. サポートされる Camel Quarkus 言語エクステンション

本リリースでは、Camel K は以下の Camel Quarkus 言語エクステンションをサポートします (Camel 式および述語での使用)。

- Constant
- ExchangeProperty
- File
- Header
- Ref
- Simple
- Tokenize
- JsonPath

3.4.4. サポートされる Camel K トレイト

本リリースでは、Camel K は以下の Camel K トレイトをサポートします。

- Builder トレイト
- Camel トレイト
- Container トレイト
- Dependencies トレイト
- Deployer トレイト
- Deployment トレイト
- Environment トレイト
- Jvm トレイト
- Kamelets トレイト
- Owner トレイト
- Platform トレイト

- Pull Secret トレイト
- Prometheus トレイト
- Quarkus トレイト
- Route トレイト
- Service トレイト
- Error Handler トレイト

3.5. サポート対象の KAMELETS

以下の表は、Camel K Operator のインストール時に OpenShift リソースとして提供される kamelets を示しています。

これらの kamelets の詳細は、<https://github.com/openshift-integration/kamelet-catalog/tree/kamelet-catalog-1.6> を参照してください。

kamelets を使用してアプリケーションやサービスを接続する方法の詳細は https://access.redhat.com/documentation/ja-jp/red_hat_integration/2021.q4/html/single/integrating_applications_with_kamelets を参照してください。



重要

アスタリスク (*) のマークが付いた Kamelets は、テクノロジープレビュー機能としてのみ提供されています。テクノロジープレビュー機能は、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るため、Red Hat は本番環境での使用は推奨しません。Red Hat は実稼働環境でこれらを使用することを推奨していません。

このような機能は、近々発表予定の製品機能をリリースに先駆けてご提供します。これにより、お客様は機能性をテストし、開発プロセス中にフィードバックをお寄せいただくことができます。Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、「[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。

表3.1 Camel K Operator で提供される Kamelets

Kamelet	ファイル名	タイプ (シンク、ソース、アクション)
Avro Deserialize action	avro-deserialize-action.kamelet.yaml	アクション (データ変換)
Avro Serialize action	avro-serialize-action.kamelet.yaml	アクション (データ変換)
AWS 2 S3 sink	aws-s3-sink.kamelet.yaml	シンク
AWS 2 S3 source	aws-s3-source.kamelet.yaml	ソース

Kamelet	ファイル名	タイプ (シンク、ソース、アクション)
AWS 2 S3 ストリーミングアップロードシンク	aws-s3-streaming-upload-sink.kamelet.yaml	シンク
AWS 2 Kinesis sink	aws-kinesis-sink.kamelet.yaml	シンク
AWS 2 Kinesis source	aws-kinesis-source.kamelet.yaml	ソース
AWS 2 Lambda sink	aws-lambda-sink.kamelet.yaml	シンク
AWS 2 Simple Notification System sink	aws-sns-sink.kamelet.yaml	シンク
AWS 2 Simple Queue Service sink	aws-sqs-sink.kamelet.yaml	シンク
AWS 2 Simple Queue Service source	aws-sqs-source.kamelet.yaml	ソース
AWS SQS FIFO シンク	aws-sqs-fifo-sink.kamelet.yaml	シンク
Cassandra sink*	cassandra-sink.kamelet.yaml	シンク
Cassandra source*	cassandra-source.kamelet.yaml	ソース
Elasticsearch Index sink*	elasticsearch-index-sink.kamelet.yaml	シンク
フィールドアクションの抽出	extract-field-action.kamelet.yaml	アクション
FTP sink	ftp-sink.kamelet.yaml	シンク
FTP source	ftp-source.kamelet.yaml	ソース
ヘッダーフィルターアクションあり	has-header-filter-action.kamelet.yaml	アクション (データ変換)
Hoist フィールドアクション	hoist-field-action.kamelet.yaml	アクション
HTTP sink	http-sink.kamelet.yaml	シンク
Insert Field action	insert-field-action.kamelet.yaml	アクション (データ変換)
Insert Header action	insert-header-action.kamelet.yaml	アクション (データ変換)

Kamelet	ファイル名	タイプ (シンク、ソース、アクション)
Tombstone Filter アクション	is-tombstone-filter-action.kamelet.yaml	アクション (データ変換)
Jira source*	jira-source.kamelet.yaml	ソース
JMS sink	jms-amqp-10-sink.kamelet.yaml	シンク
JMS source	jms-amqp-10-source.kamelet.yaml	ソース
JSON Deserialize action	json-deserialize-action.kamelet.yaml	アクション (データ変換)
JSON Serialize action	json-serialize-action.kamelet.yaml	アクション (データ変換)
Kafka sink	kafka-sink.kamelet.yaml	シンク
Kafka ソース	kafka-source.kamelet.yaml	ソース
Kafka トピック名フィルターアクション	topic-name-matches-filter-action.kamelet.yaml	アクション (データ変換)
Log sink	log-sink.kamelet.yaml	シンク (開発 およびテスト 目的)
mask フィールドアクション	mask-field-action.kamelet.yaml	アクション (データ変換)
メッセージタイムスタンプルー ターアクション	message-timestamp-router- action.kamelet.yaml	アクション (ルーター)
MongoDB sink	mongodb-sink.kamelet.yaml	シンク
MongoDB source	mongodb-source.kamelet.yaml	ソース
MySQL シンクコネクタ	mysql-sink.kamelet.yaml	シンク
PostgreSQL シンクコネクタ	postgresql-sink.kamelet.yaml	シンク
Predicate filter action	predicate-filter-action.kamelet.yaml	アクション (ルーター/ フィルター)

Kamelet	ファイル名	タイプ (シンク、ソース、アクション)
Protobuf Deserialize action	protobuf-deserialize-action.kamelet.yaml	アクション (データ変換)
Protobuf Serialize action	protobuf-serialize-action.kamelet.yaml	アクション (データ変換)
Regex Router action	regex-router-action.kamelet.yaml	アクション (ルーター)
フィールドアクションの置き換え	replace-field-action.kamelet.yaml	アクション
Salesforce source	salesforce-source.kamelet.yaml	ソース
SFTP sink	sftp-sink.kamelet.yaml	シンク
SFTP source	sftp-source.kamelet.yaml	ソース
Slack ソース	slack-source.kamelet.yaml	ソース
SQL サーバーデータベースシンク	sqlserver-sink.kamelet.yaml	シンク
Telegram source*	telegram-source.kamelet.yaml	ソース
Timer source	timer-source.kamelet.yaml	ソース (開発およびテスト目的)
ルーターアクションのタイムスタンプ	timestamp-router-action.kamelet.yaml	アクション (ルーター)
Value to Key action	value-to-key-action.kamelet.yaml	アクション (データ変換)

3.6. CAMEL K の既知の問題

以下の既知の問題は Camel K 1.6 に適用されます。

[ENTESB-15306 - CRD conflicts between Camel K and Fuse Online](#)

古いバージョンの Camel K が同じ OpenShift クラスターにインストールされたことがある場合、カスタムリソース定義との競合により、OperatorHub から Camel K のインストールに失敗します。たとえば、これには Fuse Online で以前使用できた古いバージョンの Camel K が含まれます。

この問題を回避するには、Camel K を別の OpenShift クラスターにインストールするか、Camel K をインストールする前に以下のコマンドを入力します。

```
$ oc get crds -l app=camel-k -o json | oc delete -f -
```

ENTESB-15858 - Added ability to package and run Camel integrations locally or as container images

ローカルでの Camel インテグレーションのパッケージ化および実行、またはコンテナイメージとしての Camel インテグレーションのパッケージ化および実行は、現在 Camel K には含まれておらず、コミュニティのみによるサポートの対象となります。

詳細は「[Apache Camel K community](#)」を参照してください。

ENTESB-16477 - Unable to download jira client dependency with productized build

Camel K Operator を使用する場合、インテグレーションは jira クライアントの依存関係を見つけることができません。回避策は、手動で atlassian リポジトリを追加することです。

```
apiVersion: camel.apache.org/v1
kind: IntegrationPlatform
metadata:
  labels:
    app: camel-k
    name: camel-k
spec:
  configuration:
    - type: repository
      value: <atlassian repo here>
```

ENTESB-17033 - Camel-K ElasticsearchComponent options ignored

Elasticsearch コンポーネントを設定する場合、Camel K ElasticsearchComponent オプションは無視されます。回避策として、Elasticsearch コンポーネントの使用時に `getContext () .setAutowiredEnabled(false)` を追加することができます。

ENTESB-17061 - Can't run mongo-db-source kamelet route with non-admin user - Failed to start route mongodb-source-1 because of null

管理者でないユーザー認証情報で `mongo-db-source kamelet` ルートを実行することはできません。コンポーネントの一部には管理者の認証情報が必要なため、非管理者ユーザーとしてルートを実行することはできません。

3.7. CAMEL K の修正された問題

以下のセクションには、Camel K 1.6.0 で修正された問題が記載されています。

- [「Camel K 1.6.0 で改良された機能」](#)
- [「Camel K 1.6.0 で解決されたバグ」](#)

3.7.1. Camel K 1.6.0 で改良された機能

以下の表に Camel K 1.6.0 で改良された機能を示します。

表3.2 Camel K 1.6.0 で改良された機能

問題	説明
ENTESB-17174	サポートタイマーおよびログの Kamelets (開発用)
ENTESB-17243	Camel K での非推奨の extensions/v1beta1 Ingress の使用の削除

3.7.2. Camel K 1.6.0 で解決されたバグ

以下の表に Camel K 1.6.0 で解決されたバグを示します。

表3.3 Camel K 1.6.0 で解決されたバグ

問題	説明
ENTESB-17974	CVEListenerExternal log4j-core: ログに攻撃者が制御する文字列値が含まれる場合に Log4j 2.x のリモートコード実行 [rhint-camel-k-1]
ENTESB-17626	イベントストリーミングの例の認識されないフィールド "firstTruthyTime"
ENTESB-17412	Backport CAMEL-17039 - Camel-AWS2-S3: includeBody is false, the message Body should be set (CAMEL-17039 のバックポート - Camel-AWS2-S3: includeBody が false の場合、メッセージ本文は設定すべきではない)
ENTESB-17177	Operator 1.4 が AMQ-Streams Operator 1.8 と適切に動作しません。
ENTESB-16733	OCP 開発者コンソールの EventSource カタログには、Sink Kamelets が含まれています。
ENTESB-17123	Quickstart Camel K: Event Streaming Example が UserReportSystem インテグレーションの Secret について警告する
ENTESB-17129	グローバルフラグを使用した誤った kamelet 解決策
ENTESB-17334	AWS Cloud Watch Kamelet: Header mapping is wrong
ENTESB-17174	Camel K API がテクノロジープレビューとマークされています。
ENTESB-17260	Kbind は、namespace "channel" で "channel/messages" を Kamelet "messages" として解決します。
ENTESB-17254	camel-k-example-knative - エンドポイントは非推奨になりました
ENTESB-17314	Telegram-source は適切なクラウドイベントを発行していないようです。
ENTESB-17257	Kbindは、InMemoryChannelでKameletBindingを作成するために、プロパティ "apiVersion" を必要とします。

問題	説明
ENTESB-17004	namespaceのKameletsは、Operatorからの変更で更新されません。
ENTESB-17835	OLM でインストールされた Operator はインテグレーションをビルドできません。

第4章 RED HAT INTEGRATION の OPERATOR

Red Hat Integration は、OpenShift で Red Hat Integration コンポーネントのデプロイメントを自動化するために、operator を提供します。Red Hat Integration Operator を使用して、複数のコンポーネント operator を管理できます。そのため、各コンポーネント operator を個別に管理できます。ここでは operator を紹介し、operator を使用して Red Hat Integration コンポーネントをデプロイする方法の詳細へのリンクを記載します。

4.1. OPERATOR とは

Operator は、Kubernetes アプリケーションのパッケージ化、デプロイメント、および管理を行う方法です。operator は運用上の人間の知識を仕様し、これをコンシューマーと簡単に共有できるソフトウェアにエンコードして、一般的なタスクや複雑なタスクを自動化します。

OpenShift Container Platform 4.x では、**Operator Lifecycle Manager (OLM)** を使用すると、ユーザーはすべての Operator とクラスター全体で実行される関連サービスをインストールおよび更新することができ、それらの Operator と関連サービスのライフサイクルを管理できます。これは、Kubernetes のネイティブアプリケーション (Operator) を効果的かつ自動化されたスケーラブルな方法で管理するために設計されたオープンソースツールキットである Operator Framework の一部です。

OLM は OpenShift Container Platform 4.x でデフォルトで実行されます。これは、クラスター管理者がクラスターで実行している Operator をインストールおよびアップグレードし、アクセスを付与するのに役立ちます。OpenShift Container Platform Web コンソールは、クラスター管理者が Operator をインストールし、クラスターで利用可能な Operator のカタログを使用するために特定のプロジェクトにアクセスを付与するための管理画面を提供します。

OperatorHub は、OpenShift クラスター管理者が Operator を検出、インストール、およびアップグレードするために使用するグラフィカルインターフェースです。1回クリックするだけで、これらの Operator を OpenHub からプルし、クラスターにインストールすることができ、OLM で Operator を管理することで、開発、テスト、および実稼働環境のソフトウェアをエンジニアリングチームが独自に管理することが可能です。

その他のリソース

- Operator に関する詳細は、[OpenShift のドキュメント](#)を参照してください。

4.2. RED HAT INTEGRATION OPERATOR

Red Hat Integration Operator 1.3 を使用して、複数の Red Hat Integration コンポーネント Operator をインストールおよびアップグレードできます。

- 3scale
- 3scale APIcast
- AMQ Broker
- AMQ Interconnect
- AMQ Streams
- API Designer
- Camel K

- [Fuse Console](#)
- [Fuse Online](#)
- [Service Registry](#)

4.2.1. サポートされるコンポーネント

Red Hat Integration Operator 1.3 を使用して Operator をインストールする前に、コンポーネントのリリースノートで更新を確認します。サポートされるバージョンのリリースノートには、追加のアップグレード要件があります。

- [オンプレミス型 Red Hat 3scale API Management 2.10 向けリリースノート](#)
- [Release Notes for Red Hat AMQ Broker 7.8](#)
- [Red Hat AMQ Interconnect 1.10 のリリースノート](#)
- [Red Hat AMQ Streams 2.0 on OpenShift リリースノート](#)
- [Red Hat Fuse 7.10 リリースノート \(Fuse および API Designer\)](#)
- [Release Notes for Red Hat Integration 2021.Q3 \(Red Hat Integration - Service Registry 2.0 release notes\)](#)
- [Red Hat Integration 2021.Q4 リリースノート \(Camel K リリースノート\)](#)

AMQ Streams の新たな API バージョン

Red Hat Integration Operator 1.3 は、AMQ Streams 2.0 の Operator をインストールします。

AMQ Streams バージョン 1.8 以降にアップグレードする前に、API バージョン **v1beta2** を使用するようカスタムリソースをアップグレードする必要があります。

AMQ Streams 1.7 では、AMQ Streams カスタムリソースのスキーマを更新する **v1beta2** API バージョンが導入されました。古い API バージョンは非推奨になりました。AMQ Streams 1.7 にアップグレードした後、AMQ Streams 2.0 にアップグレードする前に、API バージョン **v1beta2** を使用するようカスタムリソースをアップグレードする必要があります。

バージョン 1.7 より前の AMQ Streams バージョンからアップグレードする場合は、以下を行います。

1. AMQ Streams 1.7 へのアップグレード
2. カスタムリソースを v1beta2 に変換します。
3. AMQ Streams 2.0 へのアップグレード

詳細は、以下のドキュメントを参照してください。

- [アップグレードの要件](#)
- [v1beta2 API バージョンの導入](#)



警告

カスタムリソースおよび CRD がバージョン **v1beta2** に変換されていない場合、AMQ Streams Operator をバージョン 2.0 にアップグレードすると、クラスターに失敗します。アップグレードは **Pending** で停止します。この場合は、以下を実行します。

1. Red Hat ソリューション ([Forever pending cluster operator upgrade](#)) で説明されている手順を実行します。
2. Integration Operator をゼロにスケールしてから1に戻し、AMQ Streams 2.0 Operator のインストールをトリガーします。

Service Registry 2.0 の移行

Red Hat Integration Operator は Red Hat Integration Service Registry 2.0 をインストールします。

Service Registry 2.0 は、Service Registry 1.x のインストールを置き換えず、手動でアンインストールする必要があります。

Service Registry バージョン 1.x から 2.0 への移行に関する情報は、『[Service Registry 2.0 release notes](#)』を参照してください。

4.2.2. サポートライフサイクル

サポート対象の設定を維持するには、最新の Red Hat Integration Operator バージョンをデプロイする必要があります。Red Hat Integration Operator の各リリースバージョンは 3 カ月間のみサポートされます。

4.2.3. 修正された問題

Red Hat Integration Operator 1.3 で修正された問題はありません。

その他のリソース

- 複数の Red Hat Integration コンポーネント Operator の管理に関する詳細は、「[Installing the Red Hat Integration Operator on OpenShift](#)」を参照してください。

4.3. RED HAT INTEGRATION コンポーネント OPERATOR

たとえば、3scale Operator や Camel K Operator などを使用して、各 Red Hat Integration コンポーネント Operator を個別にインストールおよびアップグレードできます。

4.3.1. 3scale Operators

- [3scale Operator](#)
- [3scale APIcast Operator](#)

4.3.2. AMQ Operator

- [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 Operator

- [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](#)

その他のリソース

- 複数の Red Hat Integration コンポーネント Operator の管理に関する詳細は、「[Installing the Red Hat Integration Operator on OpenShift](#)」を参照してください。