



Red Hat AMQ 7.5

AMQ Streams 1.3 on OpenShift リリースノート

AMQ Streams on OpenShift Container Platform の使用

Red Hat AMQ 7.5 AMQ Streams 1.3 on OpenShift リリースノート

AMQ Streams on OpenShift Container Platform の使用

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_AMQ_Streams_1.3_on_OpenShift.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[®] 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

本リリースノートには、AMQ Streams 1.3 リリースに含まれる新機能、改良された機能、修正、および問題に関する最新情報が含まれています。

目次

| | |
|---|-----------|
| 第1章 機能 | 3 |
| 1.1. KAFKA 2.3.0 のサポート | 3 |
| 1.2. AMQ STREAMS KAFKA BRIDGE | 3 |
| 1.2.1. 概要 | 3 |
| 1.2.2. テクノロジープレビューリリース以降の API の変更点 | 5 |
| 1.2.3. 3scale と Kafka Bridge のインテグレーション | 5 |
| 1.2.4. 3scale サービス検出アノテーションおよびラベル | 5 |
| 1.2.5. Kafka Bridge クイックスタート | 5 |
| 1.3. カスタムリソースのステータスプロパティ | 6 |
| 1.4. KAFKA EXPORTER のサポート | 6 |
| 第2章 機能拡張 | 7 |
| 2.1. KAFKA 2.3.0 で改良された機能 | 7 |
| 2.2. メトリクス | 7 |
| 2.3. MIRROR MAKER のヘルスチェック | 7 |
| 2.4. カスタム環境変数 | 7 |
| 第3章 テクノロジープレビュー | 9 |
| 3.1. JAEGER を使用した分散トレーシング | 9 |
| 3.2. OAUTH 2.0 認証 | 10 |
| 3.3. CHANGE DATA CAPTURE の変更 | 11 |
| 第4章 非推奨の機能 | 12 |
| 第5章 修正された問題 | 13 |
| 第6章 既知の問題 | 14 |
| 第7章 サポートされる統合製品 | 15 |
| 第8章 重要なリンク | 16 |

第1章 機能

本リリースで追加され、これまでの AMQ Streams リリースにはなかった機能は次のとおりです。

1.1. KAFKA 2.3.0 のサポート

AMQ Streams は Apache Kafka バージョン 2.3.0 に対応するようになりました。

AMQ Streams は Kafka 2.3.0 をベースにしています。Red Hat によってビルドされた Kafka ディストリビューションのみがサポートされます。

ブローカーおよびクライアントアプリケーションを Kafka 2.3.0 にアップグレードする前に、Cluster Operator を AMQ Streams バージョン 1.3 にアップグレードする必要があります。手順は、「[AMQ Streams および Kafka のアップグレード](#)」を参照してください。

詳細は、[Kafka 2.2.1](#) および [Kafka 2.3.0](#) のリリースノートを参照してください。



注記

Kafka 2.2.x は、アップグレードの目的のみで AMQ Streams でサポートされます。

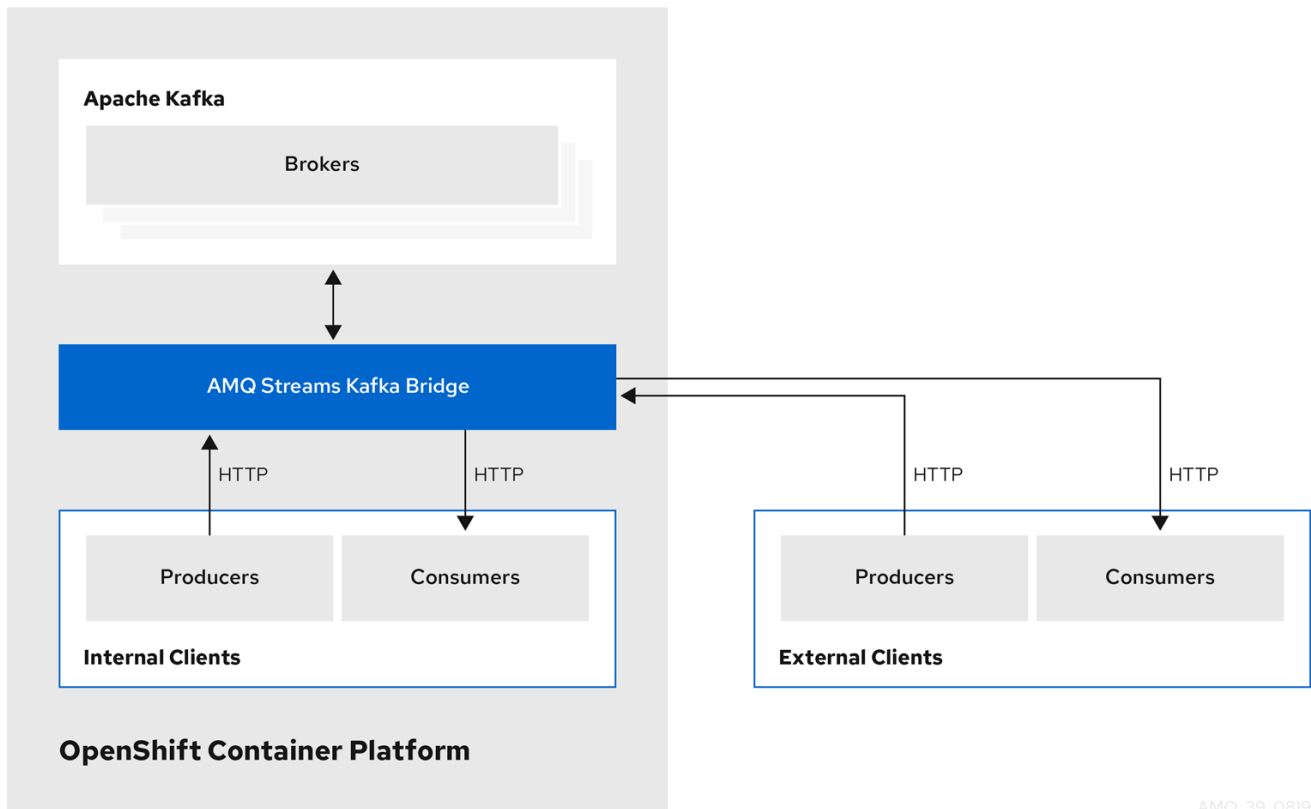
サポートされるバージョンの詳細は、カスタマーポータル「[Red Hat AMQ 7 コンポーネントの詳細](#)」を参照してください。

1.2. AMQ STREAMS KAFKA BRIDGE

AMQ Streams Kafka Bridge はテクノロジープレビューから AMQ Streams の一般提供のコンポーネントに移行します。

1.2.1. 概要

Kafka Bridge は AMQ Streams の RESTful インターフェースを提供するので、クライアントアプリケーションによる Kafka プロトコルの解釈を必要とせずに、AMQ Streams を簡単に使用および接続できるという Web API の利点を活用できます。



API には主に **consumers** と **topics** の2つのリソースがあり、Kafka クラスタでコンシューマーおよびプロデューサーと対話するためにエンドポイント経由で公開され、アクセスが可能になります。リソースと関係があるのは Kafka ブリッジのみで、Kafka に直接接続されたコンシューマーやプロデューサーとは関係はありません。

Kafka Bridge では、以下に対する HTTP 要求がサポートされます。

- トピックにメッセージを送信する。
- トピックからメッセージを取得する。
- コンシューマーを作成および削除する。
- コンシューマーをトピックにサブスクライブし、このようなトピックからメッセージを受信できるようにする。
- コンシューマーがサブスクライブしているトピックの一覧を取得する。
- トピックからコンシューマーのサブスクライブを解除する。
- パーティションをコンシューマーに割り当てる。
- コンシューマーオフセットをコミットする。
- パーティションでシーク操作を実行する。

上記の方法で、JSON 応答と HTTP 応答コードのエラー処理を行います。メッセージは JSON またはバイナリー形式で送信できます。

「[Kafka Bridge の概要](#)」および「[Kafka Bridge の設定](#)」を参照してください。API ドキュメントは、「[Kafka Bridge API reference](#)」を参照してください。

ローカルマシンから API を試すには、「[Kafka Bridge クイックスタート](#)」を参照してください。

1.2.2. テクノロジープレビューリリース以降の API の変更点

テクノロジープレビューリリース以降、Kafka Bridge API に以下の変更が加えられました。

- **openapi** エンドポイントが追加されました。これにより、JSON 形式で Kafka Bridge の OpenAPI 2.0 仕様が取得されます。
- **subscription** エンドポイントは、コンシューマーがサブスクライブしているすべてのトピックの一覧を取得するための **GET** 要求に対応するようになりました。

「Kafka Bridge API リファレンス」の「[openapi](#) および [subscription](#) エンドポイント」を参照してください。

また、以下のコンシューマー設定では OpenAPI Specification (OAS) で宣言されたデータタイプに **重大な変更** があります。

| コンシューマー設定オプション | 以前の宣言 | 現在の宣言 |
|------------------------------------|-------|---------|
| consumer.request.timeout.ms | 文字列 | Integer |
| enable.auto.commit | 文字列 | Boolean |
| fetch.min.bytes | 文字列 | Integer |

この設定のデータ型が OAS によって検証されるようになりました。その結果、無効なデータ型が **/consumers** エンドポイントに送信されると、OAS は **400 Bad Request** コード（無効な設定の詳細を含む）を返します。以前のバージョンでは、一般的な **500 内部サーバーのエラー** コードが返されていました。

1.2.3. 3scale と Kafka Bridge のインテグレーション

Red Hat 3scale API Management を Kafka Bridge と統合できるようになりました。

3scale を使用すると、TLS によって Kafka Bridge のセキュリティーが保護され、認証および承認も提供されます。3scale とのインテグレーションにより、メトリクス、流量制御、請求などの追加機能も利用できるようになりました。

「[Using the Kafka Bridge with 3scale](#)」を参照してください。

1.2.4. 3scale サービス検出アノテーションおよびラベル

Kafka Bridge をデプロイすると、Kafka Bridge の REST インターフェイスを公開するサービスには、3scale による検出に必要なアノテーションとラベルが含まれます。

「[Kafka Bridge サービス検出](#)」を参照してください。

1.2.5. Kafka Bridge クイックスタート

Kafka Bridge の新しいクイックスタートガイドは、ローカルマシンから API を試すのに役立ちます。クイックスタートには、API で最も一般的なメソッドの curl 要求の例が含まれています。

「[Kafka Bridge クイックスタート](#)」を参照してください。

1.3. カスタムリソースのステータスプロパティ

AMQ Streams カスタムリソースのステータスプロパティはテクノロジープレビューから一般提供機能に移行します。

status プロパティをクエリーすると、カスタムリソースの現在のステータスを確認できます。リソースのステータスは、リソースに関する情報をユーザーおよび情報を必要とするツールに公開します。現在の状態と最後の **observedGeneration** はすべてのリソースで利用できます。リソースによっては、リソース固有の情報も公開します。

ステータス情報は以下のリソースで利用できます。

- **Kafka**
- **KafkaTopic**
- **KafkaConnect**
- **KafkaConnectS2I**
- **KafkaMirrorMaker**
- **KafkaUser**
- **KafkaBridge**

リソースのステータスを確認するには、**oc get** コマンドを使用して、JSONPath 式を適用します。以下に例を示します。

```
oc get kafka KafkaTopic -o jsonpath='{.status}'
```

```
oc get kafka KafkaBridge -o jsonpath='{.status.observedGeneration}'
```

「[AMQ Streams カスタムリソースのステータス](#)」および「[カスタムリソースのステータスの確認](#)」を参照してください。

1.4. KAFKA EXPORTER のサポート

[Kafka Exporter](#) は、Apache Kafka ブローカーおよびクライアントの監視を強化するオープンソースプロジェクトです。Kafka Exporter は、Kafka クラスターとのデプロイメントを実現するために AMQ Streams で提供され、オフセット、コンシューマーグループ、コンシューマーラグ、およびトピックに関連する Kafka ブローカーから追加のメトリクスデータを抽出します。

一例として、メトリクスデータを使用すると、低速なコンシューマーの識別に役立ちます。

ラグデータは Prometheus メトリクスとして公開され、解析のために Grafana で使用できます。

ビルトイン Kafka メトリクスの監視のために Prometheus および Grafana をすでに使用している場合、Kafka Exporter Prometheus エンドポイントをスクレープするように Prometheus を設定することもできます。

「[Kafka Exporter](#)」を参照してください。

第2章 機能拡張

このリリースで改良された機能は次のとおりです。

2.1. KAFKA 2.3.0 で改良された機能

Kafka 2.3.0 に導入された改良機能の概要は、[Kafka 2.3.0 Release Notes](#) を参照してください。

2.2. メトリクス

AMQ Streams で提供されるメトリクスの設定例が更新されました。メトリクス設定ファイルのサンプルは、本リリースで利用可能なインストール ZIP ファイルにあります。このファイルは、[AMQ Streams のダウンロードサイト](#) から取得できます。

Prometheus を使用して監視用に設定し、Grafana ダッシュボードのモニタリングデータを取得できます。設定例を以下に示します。

- メトリクス設定の Kafka クラスターへの追加
- Prometheus および Prometheus Alertmanager プラグインのデプロイ
- Grafana のデプロイメント

「[メトリクスの導入](#)」を参照してください。

2.3. MIRROR MAKER のヘルスチェック

Mirror Maker リソースの liveness および readiness プローブを設定できるようになりました。これらの Healthcheck では、AMQ Streams デプロイメントの Mirror Maker レプリカのステータスを確認できます。

「[Kafka Mirror Maker の設定](#) および [Healthcheck の設定](#)」を参照してください。

2.4. カスタム環境変数

リソースにテンプレートプロパティを設定して、Kafka ブローカー、Mirror Maker、および Kafka Connect コンテナのカスタム環境変数を設定できるようになりました。たとえば、**Kafka.spec.kafka** の **kafkaContainer** テンプレートプロパティを使用できます。

```
# ...
spec:
  kafka:
    template:
      kafkaContainer:
        env:
          - name: TEST_ENV_1
            value: test.env.one
          - name: TEST_ENV_2
            value: test.env.two
# ...
```

本リリースでは、テクノロジープレビュー機能である分散トレースの設定時にカスタム環境変数を設定します。

「[環境変数を使用したコンテナのカスタマイズ](#)」を参照してください。

第3章 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 の実稼働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 はテクノロジープレビュー機能を実稼働環境に実装することは推奨しません。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. JAEGER を使用した分散トレーシング



注記

これはテクノロジープレビューの機能です。

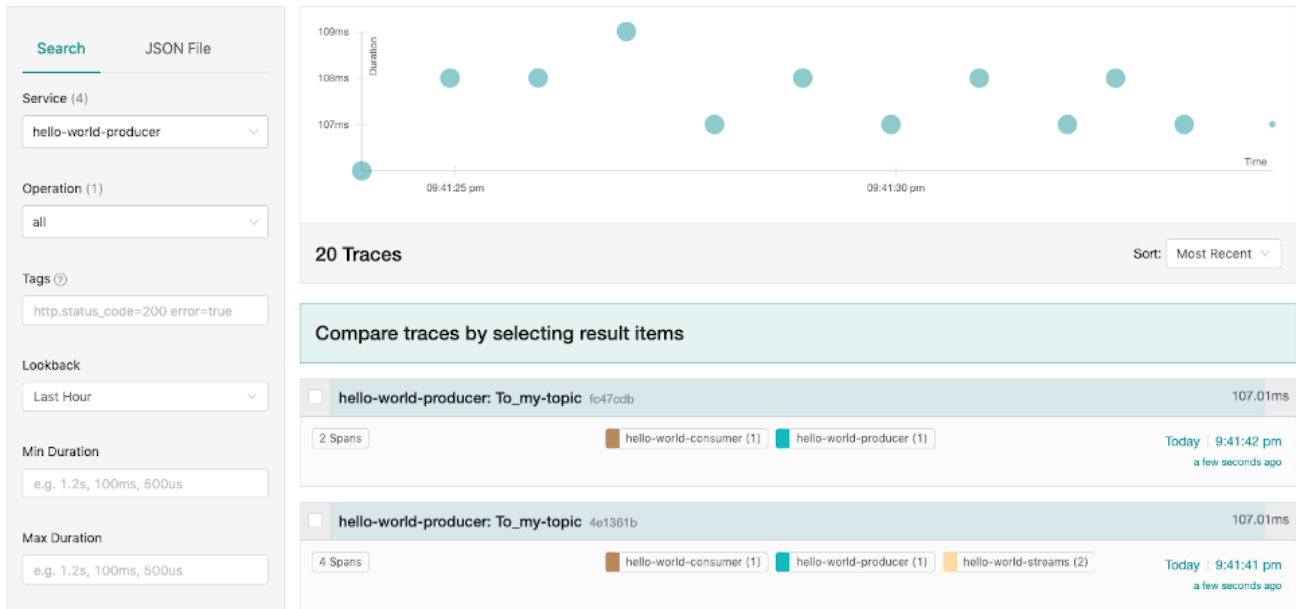
本リリースでは、通常の Kafka アーキテクチャーにおけるトランザクションの分散トレーシングのサポートが追加されました。同梱の OpenTracing Java ライブラリーを使用して、クライアントアプリケーションを活用し、メッセージの生成や消費などのトランザクションの追跡を生成できます。

分散トレースは、以下のコンポーネントでサポートされます。

- Kafka クラスター
- プロデューサーおよびコンシューマー
- Kafka Streams アプリケーション
- Mirror Maker
- Kafka Connect

トレースデータは Jaeger を使用してユーザーインターフェースで確認できます。この情報を使用して、エンドツーエンドの Kafka クラスターの操作を監視し、ターゲットシステムおよびアプリケーションのパフォーマンス問題をデバッグできます。

Jaeger ユーザーインターフェースのクエリー例



「分散トレース」を参照してください。

3.2. OAUTH 2.0 認証



注記

これはテクノロジープレビューの機能です。

AMQ Streams は、**SASL OAUTHBEARER** メカニズムを使用して OAuth 2.0 認証の使用をサポートします。

OAuth 2.0 のトークンベースの認証を使用すると、アプリケーションクライアントはアカウントのクレデンシャルを公開せずにアプリケーションサーバー（「リソースサーバー」と呼ばれる）のリソースにアクセスできます。クライアントは、認証手段としてアクセストークンを提示します。アプリケーションサーバーはこのアクセストークンを使用して、付与されたアクセス権限のレベルに関する情報を検索できます。承認サーバーは、アクセスの付与とアクセスに関する問い合わせを処理します。

AMQ Streams のコンテキストでは以下が行われます。

- Kafka ブローカーはリソースサーバーとして動作する。
- Kafka クライアントはリソースクライアントとして動作する。

ブローカーおよびクライアントは、必要に応じて OAuth 2.0 承認サーバーと通信し、アクセストークンを取得または検証します。

AMQ Streams のデプロイメントでは、OAuth 2.0 インテグレーションは以下を提供します。

- Kafka ブローカーのサーバー側の OAuth 2.0 サポート。
- Kafka MirrorMaker、Kafka Connect、および Kafka Bridge のクライアント側 OAuth 2.0 サポート。

Red Hat Single Sign-On の統合

Red Hat Single Sign-On を承認サーバーとしてデプロイし、AMQ Streams と統合するために設定することができます。

Red Hat Red Hat Single Sign-On を使用して、以下を実行できます。

- Kafka ブローカーの認証の設定
- クライアントの設定および承認
- ユーザーおよびロールの設定
- アクセスの取得およびトークンの更新

「[OAuth 2.0 トークンベース認証の使用](#)」を参照してください。

3.3. CHANGE DATA CAPTURE の変更



注記

これはテクノロジープレビューの機能です。

Red Hat Change Data Capture 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。Change Data Capture は Apache Karaf に構築され、AMQ Streams とデプロイおよび統合できます。

AMQ Streams のデプロイ後に、Kafka Connect で Change Data Capture をコネクタ設定としてデプロイします。

Change Data Capture 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 on OpenShift に渡されます。アプリケーションは **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。

Change Data Capture には以下を含む複数の用途があります。

- データレプリケーション。
- キャッシュの更新およびインデックスの検索。
- モノリシックアプリケーションの簡素化。
- データ統合。
- ストリーミングクエリーの有効化。

Change Data Capture は、以下の共通データベースのコネクタ (Kafka Connect をベースとする) を提供します。

- MySQL
- PostgreSQL
- SQL Server
- MongoDB

Debezium は、Change Data Capture のオープンソースプロジェクトです。

AMQ Streams で Change Data Capture をデプロイするための詳細は、「[Deploying the Debezium 1.0 Technology Preview with AMQ Streams](#)」を参照してください。

第4章 非推奨の機能

AMQ Streams 1.3 で非推奨になった機能はありません。

第5章 修正された問題

AMQ Streams 1.3 で修正された問題は、以下の表のとおりです。

| 課題番号 | 説明 |
|--------------|--|
| ENTMQST-1287 | コンシューマー作成の操作では、応答に HTTP コード 500 が含まれていてはいけない |
| ENTMQST-1236 | Kafka Connect S2I のダッシュボードがない |
| ENTMQST-1216 | [labels] - ラベルの変更 |
| ENTMQST-1194 | 無効な JSON デコード時の未処理の例外 |
| ENTMQST-1158 | サポート対象外の Kafka バージョンで Kafka クラスターをデプロイする場合の NPE |
| ENTMQST-1138 | EU/UO ロールバインディング管理での null チェックの修正 |
| ENTMQST-1137 | 新しいイメージと連携するように非推奨の Topic Operator を修正 |
| ENTMQST-1100 | delete.topic.enable=false の場合に、Topic Operator がカスタムリソースの削除を処理する必要があります。 |

第6章 既知の問題

AMQ Streams 1.3 の既知の問題を以下の表に示します。

| 課題番号 | 説明 |
|------------------------------|---|
| ENTMQST-1402 | <p>AMQ Streams Kafka Bridge リソースのステータス情報にアクセスする場合に、外部クライアントが Kafka Bridge サービスにアクセスするための URL が間違っています。URL の -bridge 部分が重複しています。以下に例を示します。</p> <pre>#... status: conditions: - lastTransitionTime: "2019-10-15T14:21:40.520Z" status: "True" type: Ready observedGeneration: 1 url: http://my-bridge-bridge-bridge-service.myproject.svc:8080</pre> <p>上記の例の正しい URL は http://my-bridge-bridge-service.myproject.svc:8080 です。</p> |

第7章 サポートされる統合製品

AMQ Streams 1.3 は、以下の Red Hat 製品とのインテグレーションをサポートします。

- OAuth 2.0 サポート用の **Red Hat Single Sign-On 7.3.0** (テクノロジープレビュー機能)
- Kafka ブリッジをセキュアにし、追加の API 管理機能を提供する **Red Hat 3scale API Management 2.6**。

これらの製品によって AMQ Streams デプロイメントに導入可能な機能の詳細は、AMQ Streams 1.3 のドキュメントを参照してください。

第8章 重要なリンク

- [Red Hat AMQ 7 でサポートされる構成](#)
- [Red Hat AMQ 7 コンポーネントの詳細](#)

改訂日時 : 2022-07-02 18:13:59 +1000