Red Hat AMQ 2020.Q4

AMQ Streams 1.6 on RHEL リリースノート

AMQ Streams on Red Hat Enterprise Linux の使用

Last Updated: 2022-07-14

Red Hat AMQ 2020.Q4 AMQ Streams 1.6 on RHEL リリースノート

AMQ Streams on Red Hat Enterprise Linux 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

概要

本リリースノートには、AMQ Streams 1.6 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 特長

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。

注記

本リリースで解決された改良機能とバグをすべて確認するには、AMQ Streams の Jira プロジェクト を参照してください。

1.1. AMQ STREAMS 1.6.X の KAFKA サポート（長期サポート）

ここでは、AMQ Streams 1.6 および後続のパッチリリースでサポートされる Kafka および ZooKeeper のバージョンについて説明します。

AMQ Streams 1.6.x は、RHEL 7 および 8 で使用する長期サポートリリースです。

AMQ LTS バージョンのサポート日付の詳細は、Red Hat ナレッジベースソリューションHow long are AMQ LTS releases supported?を参照してください。

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。以前のバージョンの Kafka は、アップグレードの目的のみで AMQ Streams 1.6.x でサポートされます。

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

1.1.1. AMQ Streams 1.6.6 および 1.6.7 での Kafka のサポート

AMQ Streams 1.6.6 および 1.6.7 リリースは Apache Kafka バージョン 2.6.3 をサポートします。

アップグレードの手順は、「AMQ Streams and Kafka upgrades 」を参照してください。

詳細は、Kafka 2.6.3 リリースノートを参照してください。

Kafka 2.6.3 には ZooKeeper バージョン 3.5.9 が必要です。そのため、AMQ Streams 1.6.4 / 1.6.5 からのアップグレード時に ZooKeeper をアップグレードする必要は **ありません**。

1.1.2. AMQ Streams 1.6.4 および 1.6.5 での Kafka のサポート

AMQ Streams 1.6.4 および 1.6.5 リリースではサポートされ、Apache Kafka バージョン 2.6.2 および ZooKeeper バージョン 3.5.9 を使用します。

アップグレードの手順は、「AMQ Streams and Kafka upgrades 」を参照してください。

詳細は、Kafka 2.6.2 Release Notes を参照してください。

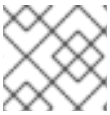
Kafka 2.6.2 には ZooKeeper バージョン 3.5.9 が必要です。そのため、AMQ Streams 1.6.0 からアップグレード時に ZooKeeper をアップグレードする必要があります。

1.1.3. AMQ Streams 1.6.0 での Kafka のサポート

AMQ Streams 1.6.0 は Apache Kafka バージョン 2.6.0 をサポートし、使用します。

アップグレードの手順は、「AMQ Streams and Kafka upgrades 」を参照してください。

詳細は、Kafka 2.5.0 および Kafka 2.6.0 のリリースノートを参照してください。



注記

Kafka 2.5.x は、アップグレードの目的のみで AMQ Streams 1.6.0 でサポートされます。

Kafka 2.6.0 には、Kafka 2.5.x と同じ ZooKeeper バージョン (ZooKeeper バージョン 3.5.7 / 3.5.8) が必要です。そのため、AMQ Streams 1.5 からのアップグレード時に ZooKeeper をアップグレードする必要は **ありません**。

1.2. OAUTH 2.0 承認

OAuth 2.0 承認のサポートは、テクノロジープレビューから AMQ Streams の一般利用可能なコンポーネントに移行されます。

トークンベースの認証に OAuth 2.0 を使用している場合、OAuth 2.0 承認ルールを使用して、クライアントの Kafka ブローカーへのアクセスを制限できるようになりました。

AMQ Streams は、Red Hat Single Sign-On の 承認サービス による OAuth 2.0 トークンベースの承認をサポートします。これにより、セキュリティーポリシーとパーミッションの一元的な管理が可能になります。

Red Hat Single Sign-On で定義されたセキュリティーポリシーおよびパーミッションは、Kafka ブローカーのリソースへのアクセスを付与するために使用されます。ユーザーとクライアントは、Kafka ブローカーで特定のアクションを実行するためのアクセスを許可するポリシーに対して照合されます。

「OAuth 2.0 トークンベース承認の使用」を参照してください。

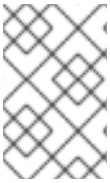
1.3. OPEN POLICY AGENT (OPA) の統合

Open Policy Agent (OPA) は、オープンソースのポリシーエンジンです。OPA と AMQ Streams を統合して、Kafka ブローカーでのクライアント操作を許可するポリシーベースの承認メカニズムとして機能します。

クライアントからリクエストが実行されると、OPA は Kafka アクセスに定義されたポリシーに対してリクエストを評価し、リクエストを許可または拒否します。

Kafka クラスター、コンシューマーグループ、およびトピックのアクセス制御を定義できます。たとえば、プロデューサークライアントから特定のブローカートピックへの書き込みアクセスを許可する承認ポリシーを定義できます。

KafkaAuthorizationOpa スキーマ参照



注記

- Red Hat は OPA サーバーをサポートしません。
- opa 統合は Open JDK 11 でのみサポートされます。

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 の改良された機能

以下で紹介されている機能拡張の概要を説明します。

- Kafka 2.6.2 は Kafka 2.6.2 Release Notes を参照してください（AMQ Streams 1.6.4 のみに適用）。
- Kafka 2.6.1 は Kafka 2.6.1 リリースノートを参照してください (AMQ Streams 1.6.4 のみに適用)。
- Kafka 2.6.0。Kafka 2.6.0 リリースノート を参照してください。

2.2. KAFKA BRIDGE の改良された機能

本リリースには、AMQ Streams の Kafka Bridge コンポーネントに対して改良された以下の機能が含まれています。

パーティションおよびメタデータの取得

Kafka Bridge は以下の操作をサポートするようになりました。

- 特定のトピックのパーティションリストを取得します。

```
GET /topics/{topicname}/partitions
```

- パーティション ID、リーダーブローカー、レプリカ数などの指定のパーティションのメタデータを取得します。

```
GET /topics/{topicname}/partitions/{partitionid}
```

『Kafka Bridge API reference』を参照してください。

Kafka メッセージヘッダーのサポート

Kafka Bridge を使用して送信されたメッセージに Kafka メッセージヘッダーが含まれるようになりました。

/topics エンドポイントへの POST リクエストでは、リクエストボディーに含まれるメッセージペイロードに任意でヘッダーを指定できます。メッセージヘッダーの値はバイナリー形式である必要があり、Base64 としてエンコードされる必要があります。

Kafka メッセージヘッダーのあるリクエストの例

```
curl -X POST \
  http://localhost:8080/topics/my-topic \
  -H 'content-type: application/vnd.kafka.json.v2+json' \
  -d '{
    "records": [
        {
            "key": "my-key",
            "value": "sales-lead-0001"
```

```
            "partition": 2
            "headers": [
              {
                "key": "key1",
                "value": "QXBhY2hlIEthZmthIGlzIHRoZSBib21iIQ=="
              }
            ]
          },
      ]
}'
```

「Kafka Bridge へのリクエスト」を参照してください。

2.3. MIRRORMAKER 2.0 トピック名変更の更新

MirrorMaker 2.0 アーキテクチャーは、リモートトピックの名前を自動変更してソースクラスターを表すことで、双方向レプリケーションをサポートします。元のクラスターの名前の先頭には、トピックの名前が追加されます。

必要に応じて、**IdentityReplicationPolicy** をソースコネクター設定に追加することで、名前の自動変更をオーバーライドできるようになりました。この設定が適用されると、トピックには元の名前が保持されます。

```
replication.policy.class= io.strimzi.kafka.connect.mirror.IdentityReplicationPolicy (1)
```

(1) リモートトピック名の自動変更をオーバーライドするポリシーを追加します。その名前の前にソースクラスターの名前を追加する代わりに、トピックが元の名前を保持します。

オーバーライドは、**アクティブ/パッシブ**クラスター設定でバックアップを作成したり、データを別のクラスターに移行する場合などに便利です。いずれの場合でも、リモートトピックの名前を自動的に変更したくないことがあります。

「AMQ Streams の MirrorMaker 2.0 との使用」を参照してください。

2.4. OAUTH 2.0 認証および承認

本リリースには、OAuth 2.0 トークンベースの認証および承認に対して改良された以下の機能が含まれています。

セッションの再認証
AMQ Streams の OAuth 2.0 認証は Kafka ブローカーの **セッションの再認証** をサポートするようになりました。これは、Kafka クライアントと Kafka ブローカー間の認証された OAuth 2.0 セッションの最大期間を定義します。セッションの再認証は、高速のローカル JWT とイントロスペクションエンドポイントの両方のタイプでサポートされます。

server.properties ファイルの Kafka ブローカーの OAuth 2.0 設定でセッションの再認証を設定します。

- すべてのリスナーに適用するには、connection. **max.reauth.ms** プロパティーをミリ秒単位で設定します。
- 特定のリスナーに適用するには、**listener.name.LISTENER-NAME.oauthbearer.connections.max.reauth.ms** プロパティーをミリ秒単位で設定します。**LISTENER-NAME** は、リスナーの大文字と小文字を区別しない名前です。

認証されたセッションは、設定された最大認証時間を超えた場合や、アクセストークンの有効期限に達した場合に閉じられます。その後、クライアントは再度承認サーバーにログインし、新しいアクセストークンを取得して、Kafka ブローカーへ再認証する必要があります。これにより、既存の接続上で新しい認証セッションが確立されます。

次に再認証が必要な場合、クライアントによって試行された操作 (再認証以外) によって、ブローカーは接続を終了します。

6 分後にセッションの再認証のリスナー設定の例

```
sasl.enabled.mechanisms=OAUTHBEARER
listeners=CLIENT://0.0.0.0:9092
# ...
listener.name.client.oauthbearer.sasl.jaas.config=org.apache.kafka.common.security.oauthbearer.OAuthBearerLoginModule required \
  oauth.valid.issuer.uri="https://AUTH-SERVER-ADDRESS" \
  oauth.jwks.endpoint.uri="https://AUTH-SERVER-ADDRESS/jwks" \
  oauth.username.claim="preferred_username"  \
  oauth.client.id="kafka-broker" \
  oauth.client.secret="kafka-secret" \
  oauth.token.endpoint.uri="https://AUTH-SERVER-ADDRESS/token" ;
listener.name.client.oauthbearer.sasl.login.callback.handler.class=io.strimzi.kafka.oauth.client.JaasClientOauthLoginCallbackHandler
listener.name.client.oauthbearer.connections.max.reauth.ms=3600000
```

参照:「Kafka ブローカーのセッション再認証」および「Kafka ブローカーの OAuth 2.0 サポートの設定」

JWKS キーの更新間隔

高速のローカル JWT トークン検証を使用するように Kafka ブローカーを設定する場合に、外部リスナー設定 (**server.properties**) に **oauth.jwks.refresh.min.pause.seconds** オプションを設定できるようになりました。これは、承認サーバーによって発行される JSON Web Key Set (JWKS) 公開鍵の更新をブローカーが試行する最小間隔を定義します。

今回のリリースにより、Kafka ブローカーが不明な署名キーを検出すると、JWKS キーの更新がすぐに試行され、通常の更新スケジュールを無視します。

JWKS キーの更新を実行する間隔が 2 分間である設定の例

```
listener.name.client.oauthbearer.sasl.jaas.config=org.apache.kafka.common.security.oauthbearer.OAuthBearerLoginModule required \
  oauth.valid.issuer.uri="https://AUTH-SERVER-ADDRESS" \
  oauth.jwks.endpoint.uri="https://AUTH-SERVER-ADDRESS/jwks" \
  oauth.jwks.refresh.seconds="300" \
  oauth.jwks.refresh.min.pause.seconds="120" \
  # ...
  oauth.ssl.truststore.type="PKCS12" ;
```

JWKS キーの更新スケジュールは、**oauth.jwks.refresh.seconds** オプションに設定されます。不明な署名キーが検出されると、JWKS キーの更新は更新スケジュールとは別にスケジュールされます。最後の更新からの経過時間が **oauth.jwks.refresh.min.pause.seconds** で指定される間隔に達するまで、更新は開始されません。デフォルト値は **1** です。

「Kafka ブローカーの OAuth 2.0 サポートの設定」を参照してください。

Red Hat Single Sign-On からの付与 (Grants) の更新

Red Hat Single Sign-On により、OAuth 2.0 のトークンベースの承認に新たな設定オプションが追加されました。Kafka ブローカーの設定時に、Red Hat SSO Authorization Service からの付与 (Grants) の更新に関連する以下のオプションを定義できます。

- **strimzi.authorization.grants.refresh.period.seconds**:連続する付与 (Grants) 更新実行の間隔。デフォルト値は **60** です。**0** 以下に設定されている場合、付与 (Grants) の更新は無効になります。
- **strimzi.authorization.grants.refresh.pool.size**:アクティブなセッションの付与 (Grants) を並行して取得できるスレッドの数。デフォルト値は **5** です。

「OAuth 2.0 トークンベース承認の使用」および「OAuth 2.0 承認サポートの設定」を参照

Red Hat Single Sign-On でのパーミッション変更の検出

本リリースでは、**KeycloakRBACAuthorizer** (Red Hat SSO) 承認によってアクティブなセッションのパーミッションの変更が定期的にチェックされるようになりました。中心ユーザーとパーミッション管理の変更がリアルタイムで検出されるようになりました。

2.5. KAFKA 管理ツールで非推奨になった ZOOKEEPER オプション

--zookeeper オプションは、以下の Kafka 管理ツールで非推奨となりました。

- **bin/kafka-configs.sh**
- **bin/kafka-leader-election.sh**
- **bin/kafka-topics.sh**

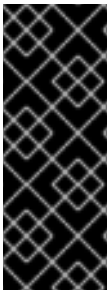
これらのツールを使用する場合は、**--bootstrap-server** オプションを使用して、接続する Kafka ブローカーを指定する必要があります。以下は例になります。

```
/bin/kafka-topics.sh --bootstrap-server localhost:9092 --list
```

--zookeeper オプションは引き続き動作しますが、今後の Kafka リリースのすべての管理ツールから削除されます。これは、Kafka の ZooKeeper への依存関係を削除するために Apache Kafka プロジェクトが進めている作業の一部です。

『AMQ Streams on RHEL の使用』ガイドが、いくつかの手順で **--bootstrap-server** オプションを使用するように更新されました。

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. CRUISE CONTROL によるクラスターのリバランス

Cruise Control をインストールし、Kafka クラスターのリバランスに使用できるようになりました。Cruise Control を使用すると、分散された Kafka クラスターを効率的に実行するための時間および労力を削減できます。

カスタマーポータルからダウンロードするために、Cruise Control の zip 形式のディストリビューションを利用できます。Cruise Control をインストールするには、提供される Metrics Reporter を使用するように各 Kafka ブローカーを設定します。その後、最適化ゴールを含む Cruise Control プロパティーを設定し、提供されたスクリプトを使用して Cruise Control を開始します。

Cruise Control サーバーは、Kafka クラスター全体の単一のマシンでホストされます。

Cruise Control が実行されている場合、REST API を使用して以下を行うことができます。

- 複数の最適化ゴールからドライラン最適化プロポーザルの生成
- 最適化プロポーザルを開始し、Kafka クラスターのリバランスを行います。

異常検出、通知、独自ゴールの作成、トピックレプリケーション係数の変更などの、その他の Cruise Control の機能は現在サポートされていません。

Cruise Control によるクラスターのリバランス を参照してください。

第4章 非推奨の機能

AMQ Streams 1.6 では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

以下のセクションには、AMQ Streams 1.6.x で修正された問題が記載されています。AMQ Streams 1.6.x を RHEL 7 および 8 で使用している場合は、Red Hat は最新のパッチリリースへアップグレードすることを推奨します。

修正された問題の詳細は、以下を参照してください。

- Kafka 2.6.3。Kafka 2.6.3 リリースノート を参照してください。
- Kafka 2.6.2。Kafka 2.6.2 リリースノート を参照してください。
- Kafka 2.6.1。Kafka 2.6.1 リリースノート を参照してください。
- Kafka 2.6.0。Kafka 2.6.0 リリースノート を参照してください。

5.1. AMQ STREAMS 1.6.7 で修正された問題

AMQ Streams 1.6.7 パッチリリース(Long Term Support)が利用可能になりました。

AMQ Streams 1.6.7 は、RHEL 7 および 8 で使用する最新の長期サポートリリースです。

AMQ Streams 1.6.7 で解決された問題の詳細は、「AMQ Streams 1.6..x Resolved Issues 」を参照してください。

Log4j の脆弱性

AMQ Streams には log4j 1.2.17 が含まれています。本リリースでは、log4j の脆弱性が多数修正されています。

本リリースで対応する脆弱性の詳細は、以下の CVE の説明を参照してください。

- CVE-2022-23307
- CVE-2022-23305
- CVE-2022-23302
- CVE-2021-4104
- CVE-2020-9488
- CVE-2019-17571
- CVE-2017-5645

5.2. AMQ STREAMS 1.6.6 で修正された問題

AMQ Streams 1.6.6 で解決された問題の詳細は、「AMQ Streams 1.6..x Resolved Issues 」を参照してください。

Log4j2 の脆弱性

AMQ Streams には log4j2 2.17.1 が含まれています。本リリースでは、log4j2 の脆弱性が多数修正されています。

本リリースで対応する脆弱性の詳細は、以下の CVE の説明を参照してください。

- CVE-2021-45046
- CVE-2021-45105
- CVE-2021-44832
- CVE-2021-44228

5.3. AMQ STREAMS 1.6.5 で修正された問題

AMQ Streams 1.6.5 で解決された問題の詳細は、「AMQ Streams 1.6..x Resolved Issues」を参照してください。

Log4j2 の脆弱性

1.6.5 リリースでは、log4j2 を使用する AMQ Streams コンポーネントのリモートコード実行脆弱性が修正されました。この脆弱性により、システムが承認されていないソースから文字列の値をログに記録した場合に、サーバーでのリモートコード実行が可能になる可能性がありました。これは 2.0 から 2.14.1 までの log4j バージョンに影響を与えます。

詳細は、CVE-2021-44228 を参照してください。

5.4. AMQ STREAMS 1.6.4 で修正された問題

AMQ Streams 1.6.4 で解決された問題の詳細は、「AMQ Streams 1.6..x Resolved Issues」を参照してください。

5.5. AMQ STREAMS 1.6.0 で修正された問題

課題番号	説明
ENTMQST-2049	Kafka Bridge:Kafka コンシューマーは group-consumerid キーで追跡する必要があります。
ENTMQST-2084	ドキュメントの ZooKeeper バージョンが AMQ Streams 1.5 のバージョンと一致しない。

第6章 既知の問題

ここでは、AMQ Streams 1.6 の既知の問題について説明します。

課題番号

ENTMQST-2030 - kafka-ack が **javax.management.InstanceAlreadyExistsException: kafka.admin.client:type=app-info,id=<client_id>with client.id を報告する**

説明

bin/kafka-acls.sh ユーティリティーを **--bootstrap-server** パラメーターと組み合わせて使用して ACL を追加または削除すると、操作に成功しても警告が生成されます。警告の理由は、2 つ目の **AdminClient** インスタンスが作成されることです。これは、Kafka の今後のリリースで修正されます。

第7章 サポートされる統合製品

AMQ Streams 1.6 は、以下の Red Hat 製品との統合をサポートします。

- OAuth 2.0 認証および OAuth 2.0 承認用の **Red Hat Single Sign-On 7.4 以上**。
- データベースを監視し、イベントストリームを作成する **Red Hat Debezium 1.0 以上**。

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 1.6 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： 2022-07-14 11:36:27 +1000