Red Hat AMQ Broker 7.10

Red Hat AMQ Broker 7.10 のリリースノート

AMQ Broker のリリースノート

Last Updated: 2024-06-11

Red Hat AMQ Broker 7.10 Red Hat AMQ Broker 7.10 のリリースノート

AMQ Broker のリリースノート

法律上の通知

Copyright © 2024 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

このリリースノートには、AMQ Broker 7.10 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

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティーにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、Red Hat CTO である Chris Wright のメッセージ をご覧ください。

第1章 AMQ BROKER 7.10 の長期サポート

AMQ Broker 7.10 は、Long Term Support (LTS) リリースバージョンとして指定されています。LTS リリースの条件の詳細については、How long are AMQ LTS releases supported? を参照してください。

Red Hat Enterprise Linux および OpenShift Container Platform のサポート

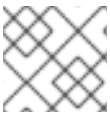
AMQ Broker 7.10 LTS バージョンは以下をサポートします。

- Red Hat Enterprise Linux 7 および 8
- OpenShift Container Platform 4.12、4.13、4.14、または 4.15。

Red Hat は、AMQ Broker が OpenShift Container Platform の将来のバージョンとの互換性を維持できるように努めています。ただし、この互換性は保証できません。相互運用性テストは、新しい OpenShift Container Platform バージョンごとに実行されます。互換性の問題が見つからない場合、OpenShift Container Platform の新しいバージョンが Red Hat AMQ Broker 7 のサポートされる設定 に追加されます。

第2章 サポートされる構成

サポートされている設定については、Red Hat AMQ Broker 7 Supported Configurations を参照してください。



注記

少なくとも、AMQ Broker 7.10 を実行するには Java バージョン 11 が必要です。

第3章 新機能と変更点

ここでは、AMQ Broker 7.10 で主要な機能拡張および新機能強調について説明します。

高可用性レプリケーションの改善

以前のバージョンの AMQ Broker では、レプリケーション高可用性 (HA) ポリシーを使用するには、少なくとも 3 ペアのライブバックアップブローカーのペアが必要でした。ペアのいずれかのブローカーで過半数を超えるクォーラム評価を取得し、両方のブローカーが同時に存続するシナリオを回避するためには、この 3 ペアが必要です。7.10 以降では、Apache Zookeeper コーディネーションサービスを使用して各ライブバックアップブローカーペアを調整するようにブローカーを設定できます。これにより、3 ペア以上のライブバックアップを持つ必要がなくなります。詳細は、**Configuring AMQ Broker**の Configuring a broker cluster for replication high availability using the ZooKeeper coordination service を参照してください。

クライアント接続のパーティション設定

以前のリリースでは、サーバー側でクライアント接続のパーティション設定を行う方法がありませんでした。7.10 以降では、クライアント接続のパーティション設定が可能です。これにより、クライアントが接続を開始するたびに、個別クライアントの接続が同じブローカーにルーティングされます。クライアント接続のパーティション設定には、以下の 2 つのユースケースがあります。

- 永続サブスクリプションのクライアントでパーティション設定を行い、サブスクライバーが永続サブスクライバーキューが置かれているブローカーに常に接続するようにします。
- 元のデータにクライアントを引き付けることにより (データグラビティとも言う)、ブローカー間でデータを移動する必要性を最小限に抑えます。
  クライアント接続のパーティション設定について、詳しくは **Configuring AMQ Broker**の Partitioning client connections を参照してください。

AMQ 管理コンソールの認証

AMQ 管理コンソールでユーザーを認証するには、証明書ベースの認証を設定できます。証明書ベースの認証を設定する方法は、**Configuring AMQ Broker**の Configuring the broker console to use certificate-based authentication を参照してください。

ノード上での Pod 配置の制御

Operator ベースのブローカーデプロイメントでは、ノードセレクター、ノードのアフィニティールール、テイントおよび容認を使用して、OpenShift Container Platform ノード上の AMQ Broker Pod の配置を制御するようにカスタムリソース (CR) を設定できます。詳細は、**Deploying AMQ Broker on OpenShift** の Controlling placement of broker pods on OpenShift Container Platform nodes を参照してください。

ブローカーのヘルスチェック

オペレーターベースのブローカーデプロイメントでは、活性および準備プローブを使用して、実行中のブローカーコンテナーで定期的な可用性チェックを設定できます。活性プローブは、ブローカーの HTTP ポートに ping を実行して、ブローカーが実行されているかどうかを確認します。準備プローブは、ブローカー用に設定された各アクセプターポートへの接続を開くことにより、ブローカーがネットワークトラフィックを受け入れることができるかどうかを確認します。ヘルスチェックの設定方法について、詳しくは **Deploying AMQ Broker on OpenShift**の Configuring broker health checks を参照してください。

デフォルトのメモリー制限のオーバーライド

Operator ベースのブローカーデプロイメントでは、ブローカーに設定されたデフォルトのメモリー制限をオーバーライドできます。デフォルトでは、ブローカーには、ブローカーの Java 仮想マシンで使用可能な最大メモリーの半分が割り当てられます。デフォルトのメモリー制限をオーバーライドする方法については、**Deploying AMQ Broker on OpenShift**の Overriding the default memory limit for a broker を参照してください。

永続ボリュームクレーム (PVC) でのストレージクラスの要求

デフォルトで、OpenShift Container Platform の AMQ Broker による 永続ボリューム要求 (PVC) は、クラスターに設定されたデフォルトのストレージクラスを使用します。今回のリリースにより、AMQ Broker のストレージクラスを指定するように CR を設定できるようになりました。PVC でストレージクラスを指定する方法については、**Deploying AMQ Broker on OpenShift**の Configuring broker storage size and storage class を参照してください。

Pod のセキュリティーコンテキストの設定

Operator ベースのブローカーデプロイメントでは、Pod のセキュリティーコンテキストを設定できます。セキュリティーコンテキストには、Pod の特権とアクセス制御の設定を定義し、任意アクセス制御の属性、Security Enhanced Linux (SELinux)、Secure Computing Mode (seccomp)、sysctl インターフェイス、および Windows で実行されるコンテナーの Window 固有属性が含まれます。詳細は、**Deploying AMQ Broker on OpenShift**の Custom Resource configuration reference を参照してください。

OpenShift Container Platform 4.15 のサポート

OpenShift Container Platform 4.6、4.7、4.8、4.9 および 4.10 の AMQ Broker のサポートに加え、OpenShift Container Platform 4.15 をサポートするようになりました。

ブローカー Pod のデフォルトのサービスアカウント名の変更

ブローカー Pod のデフォルトのサービスアカウント名を変更するには、**serviceAccountName** の name 属性を使用します。詳細は、**Deploying AMQ Broker on OpenShift**の Custom Resource configuration reference を参照してください。

ブローカー Pod のラベル付け

labels 属性を使用して、ラベルをブローカー Pod に割り当てることができます。詳細は、**Deploying AMQ Broker on OpenShift**の Custom Resource configuration reference を参照してください。

***StoreType と *StoreProvider を使用したアクセプターおよびコネクター設定の更新**

アクセプターおよびコネクターの CR 設定で、ブローカーが使用するキーストアおよびトラストストアの詳細を指定できます。

Operator チャンネル

AMQ Broker Operator である **Red Hat Integration - AMQ Broker for RHEL 8 (Multiarch)** は、次のチャネルで入手できます。

- **7.10.x** - このチャネルは、バージョン 7.10 の更新のみを提供し、現在の長期サポート (LTS) チャネルです。
- **7.x** - 現在、このチャネルはバージョン 7.9 の更新のみを提供します。
- **7.8.x** - このチャネルはバージョン 7.8 の更新のみを提供し、以前の長期サポート (LTS) チャネルでした。

注記

チャネルの切り替えにより Operator をアップグレードすることはできません。既存の Operator をアンインストールし、適切なチャネルから Operator の新規バージョンをインストールする必要があります。

選択する Operator を判別するには、Red Hat Enterprise Linux コンテナー互換性マトリクス を参照してください。

ワイルドカード値を使用して、管理 API を使用してすべてのドメインへのアクセスを許可します。

7.10.1 以降では、**management.xml** ファイルで、**entry domain** フィールドにワイルドカード値を指定できます。管理 API にアクセスすると、**entry domain** フィールドのワイルドカード値によって、すべてのドメインへのアクセスが許可されます。

```
<authorisation>
  <allowlist>
    <entry domain="*"/>
  </allowlist>
```

JGroups 5.x

以前のバージョンの AMQ Broker は JGroups 3.x を使用していました。AMQ Broker 7.10 は、JGroups 3.x と下位互換性のない JGroups 5.x を使用します。一部のプロトコルとプロトコルプロパティーが 2 つの JGroup バージョン間で変更されたため、AMQ Broker 7.10 にアップグレードするときに JGroups スタック設定を変更する必要がある場合があります。

第4章 非推奨の機能

このセクションでは、サポートされていても、AMQ Broker では非推奨になっている機能について説明します。

queues 設定要素

7.10 以降では、<queues> 設定要素が非推奨になりました。<addresses> 設定要素を使用して、アドレスと関連付けられたキューを作成できます。<queues> 設定要素は今後のリリースで削除されます。

getAddressesSettings メソッド

7.10 以降、org.apache.activemq.artemis.core.config.Configuration インターフェイスに含まれている get**Addresses**Settings メソッドは非推奨になりました。get**Address**Settings メソッドを使用して、ブローカーのアドレスとキューをプログラムで設定します。

OpenWire プロトコル

7.9 以降、OpenWire プロトコルは非推奨の機能です。新しい AMQ Broker ベースのシステムを作成する場合は、サポートされている他のプロトコルのいずれかを使用してください。この機能は今後のリリースで削除されます。

ブローカーインスタンスが実行されていないときにユーザーを追加する

7.8 以降、AMQ Broker インスタンスが実行されていない場合、CLI インターフェイスからブローカーにユーザーを追加する機能が削除されます。

ネットワーク pinger

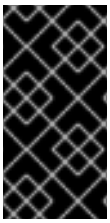
7.5 以降では、ネットワークの ping は非推奨にされています。ネットワークの ping は、ネットワークの分離の問題からブローカークラスターを保護することができません。これにより、修復不能なメッセージが失われることがあります。この機能は今後のリリースで削除されます。Red Hat では、ネットワークの ping を使用する既存の AMQ Broker デプロイメントは引き続きサポートされます。ただし、Red Hat は、新しいデプロイメントでネットワーク ping を使用することは推奨しません。高可用性を確保するためにブローカークラスターを設定し、ネットワーク分離の問題を回避するには、**AMQ Broker の設定** の 高可用性の実装 を参照してください。

Hawtio のディスパッチコンソールプラグイン

7.3 以降、AMQ Broker には Hawtio ディスパッチコンソールプラグインである **dispatch-hawtio-console.war** が同梱されなくなりました。以前のバージョンでは、AMQ Interconnect の管理にディスパッチコンソールを使用していました。ただし、AMQ Interconnect は独自のスタンドアロン Web コンソールを使用するようになりました。

第5章 テクノロジープレビュー

ここでは、AMQ Broker 7.10 のテクノロジープレビュー機能について説明します。



重要

テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は、実稼働環境での使用は推奨していません。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。

ページングのアドレス制限の設定に使用する新しい属性

以下の新しい属性を設定すると、メッセージの数に基づいて個別のアドレス制限およびグローバルアドレス制限を設定できます。

max-size-messages は、ブローカーが address-full-policy に指定されたポリシーを実行する前に、アドレスに許可されるメッセージの最大数です。デフォルト値は -1 で、メッセージ制限がないことを意味します。

global-max-messages は、ブローカーがすべてのアドレスに使用できるメッセージの合計数です。この制限に達すると、受信メッセージに関連付けられたアドレスに対して、ブローカーは address-full-policy の値として指定されたポリシーを実行します。デフォルト値は -1 で、メッセージ制限がないことを意味します。

注記

max-size-message 属性または **global-max-messages** 属性に設定された制限の前に **max-size-bytes** 属性または **global-max-size** 属性に設定された制限に達すると、ブローカーは address-full ポリシーを実行します。

第6章 修正された問題

このリリースで修正された問題の完全なリストについては、次のリンクを参照してください: AMQ Broker 7.10.0 Fixed Issues およびパッチリリースで修正された問題のリストについては、 AMQ Broker - 7.10.x Resolved Issues を参照してください。

第7章 修正された COMMON VULNERABILITIES AND EXPOSURES (CVE)

ここでは、AMQ Broker 7.10 リリースで修正された Common Vulnerabilities and Exposures (CVE) について詳しく説明します。

- ENTMQBR-5140 - CVE-2019-10744 nodejs-lodash: defaultsDeep 関数でのプロトタイプテイントによりプロパティーが変更されます
- ENTMQBR-5893 - CVE-2021-4040 broker: AMQ Broker: 不正な形式のメッセージにより、部分的な DoS (OOM) が発生する可能性があります
- ENTMQBR-5933 - CVE-2021-43797 netty: ヘッダー名の制御文字が原因で HTTP 要求スマグリングが発生する可能性があります
- ENTMQBR-6401 - CVE-2022-23913 artemis-commons: Apache ActiveMQ Artemis DoS
- ENTMQBR-6477: CVE-2020-36518 jackson-databind: ネストされたオブジェクトが深いためサービスが拒否されます

第8章 既知の問題

ここでは、AMQ Broker 7.10 の既知の問題について説明します。

- ENTMQBR-7359 - **7.10.0 Operator による認証情報シークレットの現在の処理方法を変更**
  Operator は、ブローカーに接続するための管理者のユーザー名とパスワードをシークレットに保存します。デフォルトのシークレット名は **<custom-resource-name>-credentials-secret** の形式です。シークレットは手動で作成するか、Operator による作成を許可できます。

  7.10.0 より前のカスタムリソースで **adminUser** および **adminPassword** 属性が設定されている場合、Operator は手動で作成されたシークレットをこれらの属性の値で更新します。7.10.0 以降、Operator は手動で作成されたシークレットを更新しなくなりました。したがって、CR の **adminUser** および **adminPassword** 属性の値を変更する場合は、次のいずれかを行う必要があります。

  - 新しいユーザー名とパスワードでシークレットを更新します。
  - シークレットを削除し、Operator がシークレットを作成できるようにします。Operator がシークレットを作成する場合、**adminUser** および **adminPassword** 属性が CR で指定されていればその値が追加されます。これらの属性が CR にない場合、Operator はシークレットの認証情報をランダムに生成します。

- ENTMQBR-7363 - **7.9 CR で AddressSettingsType の redeliveryDelayMultiplier を調整できない**
  redeliveryDelayMultiplier および **redeliveryCollisionAvoidanceFactor** 属性が 7.8.x または 7.9.x デプロイメントのメインブローカー CR で設定されている場合、7.10.x にアップグレードした後、新しい Operator は CR を調整できません。両方の属性のデータ型が 7.10.x で float から string に変更されたため、調整は失敗します。

  この問題を回避するには、**spec.deploymentPlan.addressSettings.addressSetting** 要素から **redeliveryDelayMultiplier** および **redeliveryCollisionAvoidanceFactor** 属性を削除します。次に、**brokerProperties** 要素で属性を設定します。以下に例を示します。

```
spec:
  ...
  brokerProperties:
  - "addressSettings.#.redeliveryMultiplier=2.1"
  - "addressSettings.#.redeliveryCollisionAvoidanceFactor=1.2"
```

注記

brokerProperties 要素で、削除した **redeliveryDelayMultiplier** 属性名の代わりに **redeliveryMultiplier** 属性名を使用します。

- ENTMQBR-7396 - **[Operator, upgrade] 7.10.1 へのアップグレードが新しい Acceptor/Connector *v1.ServiceAdmission の作成に失敗する**
  AMQ Broker 7.10.0 から 7.10.1 にアップグレードした後、サービスに追加された誤ったラベルと 7.10.0 の Pod セレクターがアップグレード中に削除されなかった場合は、メッセージングが機能しなくなります。アップグレード後にメッセージングが機能しない場合は、次の手順を実行してこの問題を解決してください。

  1. クラスター管理者として OpenShift Container Platform Web コンソールにログインします。

2. ページ上部の Project ドロップダウンメニューから、Operator がインストールされているプロジェクトを選択します。
3. 左側のナビゲーションメニューで、**Networking → Services** をクリックします。
4. **Labels** 列と **Pod Selectors** 列で、いずれかのサービスに **ActiveMQArtemis** と **application** 以外のラベルが設定されているかどうかを確認します。
5. **ActiveMQArtemis** および設定された **application** 以外のラベルを持つサービスごとに、次の手順を実行してラベルを削除します。
   a. サービスをクリックして、**Details** タブを開きます。
   b. **Labels** フィールドで **Edit** をクリックし、**ActiveMQArtemis** および **application** ラベル以外のすべてのラベルを削除します。
   c. **Save** をクリックします。
   d. **YAML** タブをクリックします。
   e. **selector** 要素で、**ActiveMQArtemis**、**application**、および **statefulset.kubernetes.io/podname** ラベルを除くすべてのラベルを削除します。
   f. **Save** をクリックします。

- ENTMQBR-7111 - **Operator の 7.10 バージョンは、アップグレード中に StatefulSet を削除する傾向がある**
  AMQ Broker Operator 7.10.0 にアップグレードする場合、または AMQ Broker Operator 7.10.0 からアップグレードする場合、新しい Operator は調整プロセス中にデプロイメントごとに既存の StatefulSet を自動的に削除します。Operator が StatefulSet を削除すると、既存のブローカー Pod が削除され、一時的なブローカーの停止が発生します。

  Operator が StatefulSet を削除する前に、次のコマンドを実行して StatefulSet を手動で削除し、実行中の Pod を孤立させることで、この問題を回避できます: oc delete statefulset **<statefulset-name>** --cascade=orphan

  アップグレードプロセス中に StatefulSet を手動で削除すると、新しい Operator は実行中の Pod を削除せずに StatefulSet を調整できます。詳細は、**Deploying AMQ Broker on OpenShift** の Upgrading the Operator using OperatorHub を参照してください。

- ENTMQBR-6991 - **7.10-opr-3 が 7.10-opr-2 ユーザーの PV ownerRef を修正しない**
  7.10.0-opr-2 をデプロイまたはアップグレードしてデプロイをスケールアップすると、新しい PV が **ownerReference** 属性で作成され、後でデプロイ CR を削除するとデータが失われる可能性があります。たとえば、7.10.0-opr-1 をデプロイし、7.10.0-opr-2 にアップグレードしてから、3 から 4 のブローカーインスタンスにスケーリングすると、**ActiveMQArtemis** CR を削除するとデータが失われる可能性があります。

  この問題を回避するには、次のことができます。

  - 可能であれば、7.10.0-opr-2 アップグレードをスキップします。
  - 7.10.0-opr-2 リリースがクラスターでアクティブになっている間は、デプロイメントをスケールアップしないでください。7.10.0-opr-3 をデプロイした後、スケールアップできます。
  - 今後のリリースでこの問題が解決されるまで、デプロイメント CR を削除しないでください。

- 影響を受ける PV の **ownerReference** 値を手動で削除します。

- ENTMQBR-6712 - **テイントおよび容認 - tolerationSeconds によりデプロイメントが破損します**
  CR の **tolerations** セクションに **tolerationSeconds** 属性を追加する場合、Operator の調整プロセスは機能せず、ブローカー Pod は適切にスケジュールされません。この問題を回避するには、CR の **tolerations** セクションに **tolerationSeconds** 属性を追加しないでください。

- ENTMQBR-6473 - **スキーマ URL の変更により設定にご完成がありません**
  バージョン 7.9 または 7.10 インスタンスで以前のリリースからのブローカーインスタンス設定の使用を試みた場合、スキーマ URL を変更したことで互換性のない設定があると、ブローカーがクラッシュします。この問題を回避するには、Upgrading from 7.9.0 to 7.10.0 on Linux で説明されているように、関連する設定ファイルのスキーマ URL を更新します。

- ENTMQBR-4813 **大きなメッセージと複数の C++ サブスクライバーの使用により AsynchronousCloseException が発生する**
  AMQP プロトコルを使用する複数の C++ パブリッシャークライアントがサブスクライバーおよびブローカーと同じホストで実行され、パブリッシャーが大きなメッセージを送信すると、サブスクライバーの 1 つがクラッシュします。

- ENTMQBR-6655 - **コマンド artemis チェックキューが Could not start Jolokia agent で失敗します**
  実行前に、**artemis check queue** コマンドによりエラーメッセージ **Could not start Jolokia agent: java.lang.IllegalStateException: Cannot open keystore for https communication: java.net.BindException: Address already in use** が表示されていました。

- ENTMQBR-6654 - **requireLogin:true は、適用された新しいブローカー CR に対してのみ機能し、既存のブローカー CR に対しては機能しません。**
  CR で **requireLogin** プロパティーが **true** に設定されている場合、AMQ_REQUIRE_LOGIN 環境変数は既存のブローカーインスタンスのステートフルセットで更新されず、コンソール認証情報は検証されません。この問題を回避するには、既存インスタンスのステートフルセットで環境変数の値を手動で更新します。

- ENTMQBR-5936 - **URL がクラスター化されていないポートをターゲットにした場合に、クライアントはバックアップサーバーにフェイルオーバーしません。**
  クライアントが HA クラスターへの接続に使用する接続 URL に、ブローカーの **static-connectors** で設定されていないポートがある場合、フェイルオーバーの発生後、クライアントは以前のライブブローカーへの接続を再試行し、新しいライブブローカーへの接続を試行しません。

- ENTMQBR-6728 - **アップグレードパスが破損しています**
  この問題により、**7.x** チャネルにサブスクライブしている AMQ Broker 7.9 ユーザーは、AMQ Broker 7.10 に自動アップグレードできなくなります。この問題を回避するには、**7.10.x** チャネルにサブスクライブします。

- ENTMQBR-5749 - **OperatorHub に表示されていても、サポートされていない Operator を削除する**
  Deploying the Operator from OperatorHub で説明されている Operators チャネルと Operator チャネルのみがサポートされています。Operator の公開に関連する技術的な理由により、他の Operator とチャネルが Operator Hub に表示されますが、無視するようにしてください。参考までに、表示されるがサポートされない Operator を次のリストに示しています。

  - Red Hat Integration-AMQ Broker LTS - すべてのチャネル

  - Red Hat Integration-AMQ Broker - alpha、current、および current-76

- ENTMQBR-17 - **AMQ222117: クラスター接続を開始できない**
  IPv6 をサポートする環境では、ブローカークラスターが適切に初期化に失敗することがあります。この失敗は、ログメッセージ **Can't assign requested address** で示される **SocketException** が原因となります。この問題を回避するには、**java.net.preferIPv4Stack** システムプロパティーを **true** に設定します。

- ENTMQBR-520: **別のアドレスにバインドされたキューと同じ名前のアドレスからの受信は許可されるべきではない**
  アドレスと同じ名前のキューは、アドレスにのみ割り当てる必要があります。既存のアドレスと同じ名前で、異なる名前のアドレスにバインドされるキューを作成することは、無効な設定です。これを実行すると、誤ったメッセージがキューにルーティングされる可能性があります。

- ENTMQBR-569 - **ID を OpenWire から AMQP へ変換すると、ID をバイナリーとして送信する**
  A-MQ 6 OpenWire クライアントから AMQP クライアントに相互プロトコルを通信する場合、追加の情報はアプリケーションメッセージプロパティーにエンコードされます。これは、ブローカーによって内部で使用される無害な情報であり、無視することができます。

- ENTMQBR-636 - **perf load (mpt) の下で、ジャーナルが破損し、JavaNullPointerException が発生する**
  ブローカーが高負荷を管理しているときに IO 関連の問題が発生しないようにするには、JVM に十分なメモリーとヒープ領域が割り当てられていることを確認してください。ActiveMQ Artemis ドキュメントの Performance Tuning 章の Tuning the VM 項を参照してください。

- ENTMQBR-648 - **JMS OpenWire クライアントは、定義された purgeOnNoConsumer またはキュー filter を持つキューにメッセージを送信できない**
  A-MQ 6 JMS クライアントを使用して、**purgeOnNoConsumer** を持つキューが **true** に設定されたアドレスにメッセージを送信します。キューにコンシューマーがない場合は失敗します。A-MQ 6 JMS クライアントを使用する場合は、**purgeOnNoConsumer** オプションを設定しないことが推奨されます。

- ENTMQBR-652 - **既知の amq-jon-plugin のバグのリスト**
  amq-jon-plugin のこのバージョンでは、ブローカーおよびキューの MBean の既知の問題があります。

  ブローカーの MBean の問題:

  - 接続を閉じると **java.net.SocketTimeoutException** 例外が発生する
  - **listSessions()** が **java.lang.ClassCastException** を出力する
  - アドレス設定を追加すると **java.lang.IllegalArgumentException** が発生する
  - **getConnectorServices()** 操作が見つからない
  - **listConsumersAsJSON()** 操作が見つからない
  - **getDivertNames()** 操作が見つからない
  - ネットワークトポロジーのリスト表示で **IllegalArgumentException** が発生する
  - アドレス設定の削除でパラメーター名が誤っている

  キュー MBean の問題:

  - **expireMessage()** で引数型の不一致例外が発生する

- **listDeliveringMessages()** が **IllegalArgumentException** を出力する
- **listMessages()** が **java.lang.Exception** を出力する
- エラーメッセージの引数型不一致で **moveMessages()** が **IllegalArgumentException** を出力する
- エラーメッセージの引数型不一致で **removeMessage()** が **IllegalArgumentException** を出力する
- **removeMessages()** が、Can't find operation removeMessage with 2 arguments の例外を出力する
- **retryMessage()** が引数型の不一致 **IllegalArgumentException** を出力する

- ENTMQBR-655 - **[AMQP]populate-validated-user が有効になっているとメッセージを送信できない**
  設定オプション **populate-validated-user** は、AMQP プロトコルを使用して生成されたメッセージではサポートされません。

- ENTMQBR-897 - **宛先名の特殊文字による OpenWire クライアント/プロトコルの問題**
  現在、AMQ OpenWire JMS クライアントは、その名前にコンマ (','')、ハッシュ ('#')、および空白を含むキューおよびアドレスにアクセスできません。

- ENTMQBR-944 - **[A-MQ7, Hawtio, RBAC] ユーザーは RBAC によって拒否された場合にフィードバックを取得しない**
  コンソールで、許可されていないユーザーが試行した操作が成功しなかったのに成功したことを示すことがあります。

- ENTMQBR-1875 - **[AMQ 7, ha, replicated store] バックアップブローカーがライブにならない、または ActiveMQIllegalStateException errorType=ILLEGAL_STATE message=AMQ119026: Backup Server was not yet in sync with live の後にシャットダウンしているようにみえる**
  バックアップブローカーがマスターブローカーと同期しようとしている間に、マスターブローカーのページングディスクを削除すると、マスターが失敗します。さらに、バックアップブローカーはマスターとの同期を試みるため、ライブになりません。

- ENTMQBR-2068 - **HA フェイルオーバー、フェイルバックのシナリオで、一部のメッセージは受信されるが配信されない**
  現在、OpenWire クライアントがメッセージを送信している間にブローカーがスレーブにフェイルオーバーすると、フェイルオーバー時にブローカーへ配信されるメッセージが失われる可能性があります。この問題を回避するには、承認する前にブローカーがメッセージを永続化していることを確認します。

- ENTMQBR-3331 - **ステートフルセットコントローラーが CreateContainerError から回復できず、Operator がブロックされる**
  AMQ Broker Operator が設定エラーのあるカスタムリソース (CR) からステートフルセットを作成すると、ステートフルセットコントローラーは、エラーが解決されたときに更新されたステートフルセットをロールアウトできません。

  たとえば、メインブローカ CR の **image** 属性の値にスペルミスがあると、ステートフルセットコントローラーによって作成された最初の Pod のステータスが **Pending** のままになります。その後、スペルミスを修正して CR の変更を適用すると、AMQ Broker Operator はステートフルセットを更新します。ただし、Kubernetes の既知の問題により、ステートフルセットコントローラーは更新されたステートフルセットをロールアウトできません。コントローラーは **Pending** ステータスの Pod が **Ready** になるまで無期限に待機するため、新しい Pod はデプロイされません。

この問題を回避するには、**Pending** ステータスの Pod を削除して、ステートフルセットコントローラーが新しい Pod をデプロイできるようにする必要があります。どの Pod のステータスが **Pending** であるかを確認するには、次のコマンドを使用します: **oc get pods --field-selector=status.phase=Pending**。Pod を削除するには、**oc delete pod <pod name>** コマンドを使用します。

- ENTMQBR-3846 - **MQTT クライアントがブローカーの再起動時に再接続されない**
ブローカーを再起動するか、ブローカーがフェイルオーバーすると、アクティブなブローカーは、以前に接続された MQTT クライアントの接続を復元しません。この問題を回避するには、MQTT クライアントを再接続するのに、クライアントで **subscribe()** メソッドを手動で呼び出す必要があります。

- ENTMQBR-4023 - **AMQ Broker Operator: Pod Status の Pod 名が、実際のものとは異なる**
特定の OpenShift プロジェクトでの Operator ベースのブローカーデプロイメントの場合、**oc get pod** コマンドを使用してブローカー Pod をリスト表示すると、Pd の順序値は **0** から始まります (例: **amq-operator-test-broker-ss-0**)。ただし、**oc describe** コマンドを使用して、**activemqartmises** カスタムリソース (**oc describe activemqartemises**) から作成されたブローカー Pod のステータスを取得した場合、Pod の順序値は誤って **1** から開始します (例: **amq-operator-test-broker-ss-1**)。この問題を回避する方法はありません。

- ENTMQBR-4127 - **AMQ Broker Operator: Operator によって生成されるルート (Route) 名が OpenShift で長すぎる可能性がある**
Operator ベースのデプロイメントのブローカー Pod ごとに、Operator が AMQ Broker 管理コンソールにアクセスするために作成するルートのデフォルト名には、カスタムリソース (CR) インスタンスの名前、OpenShift プロジェクトの名前、および OpenShift クラスターの名前が含まれます。たとえば、**my-broker-deployment-wconsj-0-svc-rte-my-openshift-project.my-openshift-domain** になります。これらの名前の一部が長い場合、デフォルトのルート名は OpenShift が実施する 63 文字の制限を超えている可能性があります。この場合、OpenShift Container Platform Web コンソールでは、ルートに表示されるステータスが **Rejected** になります。

  この問題を回避するには、OpenShift Container Platform Web コンソールを使用してルートの名前を手動で編集します。コンソールでルートをクリックします。右上の **Actions** ドロップダウンメニューで、**Edit Route** を選択します。YAML エディターで **spec.host** プロパティーを見つけ、値を編集します。

- ENTMQBR-4140 - **AMQ Broker Operator:storage.size が正しくないとインストールが使用できなくなる**
カスタムリソース (CR) インスタンスの **storage.size** プロパティーを設定し、永続ストレージのデプロイメントでブローカーに必要な Persistent Volume Claim (PVC) のサイズを指定すると、Operator のインストールがこの値を適切に指定しない場合に使用できなくなります。たとえば、**storage.size** の値を **1** (つまり、単位を指定しない) に設定したとします。この場合、Operator は CR を使用してブローカーデプロイメントを作成できません。さらに、CR を削除し、**storage.size** が正しく指定された新規バージョンをデプロイする場合でも、Operator はこの CR を使用して予想通りにデプロイメントを作成することはできません。

  この問題を回避するには、まず Operator を停止します。OpenShift Container Platform Web コンソールで **Deployments** をクリックします。AMQ Broker Operator に対応する Pod の **More options** (3 つの垂直ドット) をクリックします。**Edit Pod Count** をクリックし、値を **0** に設定します。Operator Pod が停止すると、**storage.size** を正しく指定した CR の新規バージョンを作成します。次に、Operator を再起動するには、**Edit Pod Count** を再度クリックし、値を **1** に戻します。

- ENTMQBR-4141 - **AMQ Broker Operator: ステートフルセットを再作成した後も手動での関与が必要になる**
デプロイメントのブローカーで必要な Persistent Volume Claim (PVC) のサイズを大きくしよう

とすると、手動で操作をしなければ変更が反映されません。たとえば、カスタムリソース (CR) インスタンスの **storage.size** プロパティーに、PVC の初期サイズを指定するとします。CR を変更して **storage.size** の **別の** 値を指定する場合、既存のブローカーは元の PVC サイズを引き続き使用します。これは、デプロイメントをゼロブローカーに縮小してから元の数に戻した場合でも当てはまります。ただし、デプロイメントのサイズを拡大してブローカーを追加すると、新しいブローカーは新しい PVC サイズを使用します。

この問題を回避し、デプロイメント内のすべてのブローカーが同じ PVC サイズを使用するようにするには、OpenShift Container Platform Web コンソールを使用してデプロイメントで使用される PVC サイズを拡張します。コンソールで、**Storage → Persistent Volume Claims**をクリックします。デプロイメントをクリックします。右上の **Actions** ドロップダウンメニューで **Expand PVC** を選択し、新規の値を入力します。

第9章 重要なリンク

- Red Hat AMQ Broker 7.9 リリースノート
- Red Hat AMQ Broker 7.8 リリースノート
- Red Hat AMQ Broker 7.7 リリースノート
- Red Hat AMQ Broker 7.6 リリースノート
- Red Hat AMQ Broker 7.1 から 7.5 のリリースノート (まとめ)
- Red Hat AMQ 7 でサポートされる設定
- Red Hat AMQ 7 コンポーネントの詳細

改訂日時: 2024-06-11