



Red Hat AMQ Broker 7.11

Red Hat AMQ Broker 7.11 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

これらのリリースノートには、AMQ Broker 7.11 リリースに含まれる新機能、機能拡張、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	3
第1章 AMQ BROKER 7.11 の長期サポート	4
第2章 サポートされる構成	5
第3章 新機能と変更点	6
第4章 非推奨の機能	15
第5章 削除された機能	16
第6章 修正された問題	17
第7章 修正された COMMON VULNERABILITIES AND EXPOSURES (CVE)	18
第8章 既知の問題	19
第9章 重要なリンク	23

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、用語の置き換えは、今後の複数のリリースにわたって段階的に実施されます。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 7.11 の長期サポート

AMQ Broker 7.11 は、長期サポート (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 7.11 LTS バージョンは以下をサポートします。

- Red Hat Enterprise Linux 7、8、および 9
- 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 でのサポート対象設定](#) を参照してください。

Java の最小バージョン

AMQ Broker 7.11 を実行するには、少なくとも Java バージョン 11 が必要です。

OpenWire サポート

AMQ Broker 7 は、2017 年のリリース以来、クライアントアプリケーションを AMQ 7 に移行する手段として OpenWire プロトコルのサポートを提供してきました。2021 年の AMQ Broker 7.9.0 のリリースにより、OpenWire プロトコルは非推奨となり、顧客は既存の OpenWire クライアントアプリケーションを AMQ 7 の完全にサポートされているプロトコル (CORE、AMQP、MQTT、または STOMP) の 1 つに移行することが推奨されました。AMQ Broker 8.0 リリース以降、OpenWire プロトコルは AMQ Broker から削除されます。

第3章 新機能と変更点

このセクションでは、AMQ Broker 7.11 の一連の拡張機能と新機能について説明します。

CR のイメージおよびバージョン属性の検証に関する変更

7.11.5 では、Operator は、以前のバージョンとは異なる方法で CR のイメージおよびバージョン属性の設定を検証します。

7.11.5 より前のバージョンでは、Operator は CR に以下が含まれていないことを検証します。

- **spec.deploymentPlan.initImage** 属性のない **spec.deploymentPlan.image** 属性、またはその逆も同様です。
- **spec.deploymentPlan.image** 属性と **spec.deploymentPlan.initImage** 属性のいずれか、またはその両方を持つ **spec.version** 属性。

7.11.5 では、Operator は **spec.deploymentPlan.initImage** 属性のない

spec.deploymentPlan.image 属性、またはその逆が CR に含まれていないことの検証を続けます。

7.11.5 では、Operator 検証が変更され、次の点も検証されます。

- **spec.version** 属性で指定されたバージョン番号 (存在する場合) が、デプロイされたブローカーコンテナイメージのバージョンと一致すること。
バージョンが異なる場合、Operator は CR の **BrokerVersionAligned** 条件のステータスを **Unknown** に設定し、情報提供のために不一致であることを強調して示します。ただし、不一致であることはデプロイメント内のブローカーの実行には影響しません。
- CR に、**spec.version** 属性のない **spec.deploymentPlan.image** 属性と **spec.deploymentPlan.initImage** 属性が含まれていないこと。
CR に **spec.version** 属性のない **spec.deploymentPlan.image** 属性と **spec.deploymentPlan.initImage** 属性が含まれる場合、Operator は CR の **Valid** 条件のステータスを **Unknown** に設定して、設定が不完全であることを警告します。



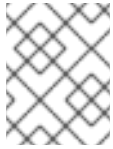
注記

spec.version 属性が欠落していると、Operator がアップグレードされるたびに、Operator はデプロイメント内のブローカー Pod を再起動します。**spec.version** 属性にバージョン番号が明示的に設定されていない限り、Operator はサポートされているブローカーの最新バージョンで StatefulSet のラベルを更新するため、Pod の再起動が必要です。

今後 Operator をアップグレードするたびにブローカーが再起動しないようにするには、デプロイしたブローカーのバージョン番号を CR の **spec.version** 属性に設定する必要があります。

- 7.11.5 では、ブローカーを起動した後に、デプロイしたブローカーのバージョン番号を CR の **status** セクションで確認できます。詳細は、[OpenShift での AMQ Broker のデプロイのブローカーデプロイメントのステータス情報の表示](#) を参照してください。
- 7.11.5 より前のバージョンでは、OpenShift Container Platform Web コンソールを使用して、ブローカー Pod のログファイルでバージョン番号を見つけることができます。
 - i. **Workloads** → **Pods** をクリックします。
 - ii. ブローカーの Pod 名をクリックします。
 - iii. **Logs** タブをクリックします。

バージョン番号は、ログ出力の上部にある **Artemis** バナーの後に表示されます。



注記

条件のステータス値が **Unknown** であることで、Operator によるブローカーデプロイメントの完了が妨げられることはありません。

Operator のリーダー選出設定がカスタマイズ可能に

7.11.5 以降、Operator がリーダー選出に使用する設定をカスタマイズできるようになりました。Operator Hub を使用して Operator をインストールする場合、Operator のインストール後に、OpenShift Container Platform Web コンソールを使用して Operator サブスクリプションのリーダー選出設定を指定できます。OpenShift Container Platform コマンドラインインターフェイスを使用して Operator をインストールする場合は、Operator のインストール前または後に、Operator 設定ファイル **operator.yaml** でリーダー選出設定を指定できます。

以下は、**operator.yaml** ファイルで設定されたリーダー選出設定の例です。

```
apiVersion: apps/v1
kind: Deployment
...
template
...
spec:
containers:
- args:
- --leader-elect
- --lease-duration=60
- --renew-deadline=40
- --retry-period=5
...
```

以下は、Operator サブスクリプションで設定されたリーダー選出設定の例です。

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
...
spec:
...
config:
env:
- name: ARGS
value: "--leader-elect --lease-duration=60 --renew-deadline=60 --retry-period=5"
```

leader-elect: Operator が他のインスタンスとリーダー選出で競合できるようにして、一度に複数のインスタンスが動作しないようにします。

lease-duration: 非リーダーの Operator が、前のリーダーが更新しなかったリースの取得を試行するまで待機する期間 (秒単位)。デフォルトは **15** です。

renew-deadline: リーダーの Operator がリーダーの役割の更新を試行する間隔 (秒単位)。この期間が経過すると、当該 Operator はリーダーでなくなります。デフォルトは **10** です。

retry-period: Operator がリーダーの役割の取得と更新を試行する間隔 (秒単位)。デフォルトは **2** です。

address-settings 要素の個々の属性が更新可能に

7.11.5 では、JMX への Jolokia REST インターフェイスを使用して、**address-settings** 要素の個々の属性を JSON 形式で更新できます。以前は、**address-settings** 要素の属性または属性のサブセットを更新する場合は、変更されていないすべての **address-settings** 属性も更新操作に含める必要がありました。

Critical Analyzer の警告レベルのメッセージをエラーレベルに変更

7.11.5 では、Critical Analyzer は、以前 **WARN** レベルを割り当てていたメッセージに **ERROR** レベルを割り当てます。

ページングされたメッセージのメモリーへのフローをより詳細に制御できる新しいパラメーター

7.11.3 より前のバージョンでは、**max-read-page-bytes** パラメーターと **max-read-page-messages** パラメーターの制限を設定することで、ページングされたメッセージのメモリーへのフローを制御します。これらの制限を適用すると、ブローカーは、コンシューマーに配信する準備ができていないメモリー内のメッセージと、現在配信中のメッセージの両方をカウントします。コンシューマーによるメッセージの確認が遅い場合、現在配信中のメッセージによってメモリーまたはメッセージの制限が使用され、ブローカーが新しいメッセージをメモリーに読み込むことができなくなる可能性があります。その結果、ブローカーでメッセージが不足する可能性があります。

7.11.3 以降では、2つの新しいパラメーターに制限を設定して、ページングされたメッセージのメモリーへのフローを制御できるようになりました。これらの制限を適用した場合、ブローカーは配信中のメッセージを考慮しません。

- **prefetch-page-bytes** ページングされたメッセージをキューごとにメモリーに読み込むために使用できるメモリー (バイト単位)。デフォルト値は 20 MB です。
- **prefetch-page-messages** ブローカーがキューごとにディスクからメモリーに読み込むことができるページングされたメッセージの数。デフォルト値は -1 で、制限が適用されないことを意味します。

コンシューマーによるメッセージの確認が遅い場合は、**max-read-page-bytes** パラメーターと **max-read-page-message** パラメーターのデフォルトの制限を引き上げて、配信中のメッセージ用に容量を提供できます。そして、**prefetch-page-bytes** パラメーターと **prefetch-page-messages** パラメーターのデフォルトの制限により、ブローカーは新しいメッセージをメモリーに読み込むことができます。



注記

prefetch-page-bytes パラメーターの値よりも前に **max-read-page-bytes** パラメーターの値に達すると、ブローカーはページングされたメッセージをそれ以上メモリーに読み込まなくなります。

AMQ Core Protocol JMS クライアントが、クラスター内の他のライブブローカーにフェイルオーバーできるように

7.11.2 より前のバージョンでは、高可用性 (HA) 用のライブ/バックアップペアとしてライブブローカーとバックアップブローカーの両方が設定されている場合、AMQ Core Protocol JMS クライアントは、ライブブローカーへの接続を失ったときに、バックアップブローカーにしかフェイルオーバーできません。7.11.2 以降では、AMQ Core Protocol JMS クライアントは、HA ペア内のバックアップブローカーまたはクラスター内の他のライブブローカーにフェイルオーバーするように設定できます。

クライアントがクラスター内のライブブローカーにフェイルオーバーできるようにするには、クライアントの接続 URI で **failoverAttempts** 設定オプションを指定します。以下に例を示します。

```
ConnectionFactory connectionFactory = new
ActiveMQConnectionFactory("(tcp://host1:port,tcp://host2:port,tcp://host3:port,tcp://host4:port,tcp://host5
:port)?ha=true&failoverAttempts=2&reconnectAttempts=2");
```

failoverAttempts オプションの値を 2 に設定すると、クライアントはクラスター内の他の 2 つのライブブローカーへの接続を試行します。フェイルオーバーの試行は、クライアントが非 HA 設定のライブブローカーへの接続試行と、HA 設定のライブブローカーおよびバックアップブローカーへの接続試行にすべて失敗した場合に行われます。



注記

上記の例に示す **reconnectAttempts** 設定オプションは、両方のブローカーが HA ペアで設定されている場合、ライブブローカーからバックアップブローカーへのフェイルオーバーにのみ使用されます。

AMQ Broker が JDBC 永続性を使用する場合のページングパフォーマンスの向上

7.11.2 以降、AMQ Broker が JDBC 永続性を使用するように設定されている場合、ページングのパフォーマンスが向上します。ページングの改善の一環として、新しいパラメーター **jdbc-max-page-size-bytes** により、JDBC 使用時のページサイズがデフォルトで 100 KB に制限されます。デフォルトの制限はカスタマイズできます。詳細は、[AMQ Broker の設定の JDBC 永続性の設定](#) を参照してください。

フェデレーションされたメッセージのバッチ処理

キューのバックログがローカルコンシューマーの利用可能な容量を超えた場合、優先度の低いフェデレーションコンシューマーがメッセージを受信する候補になります。最終的に、あまりに多くのメッセージがフェデレーテッドコンシューマーに移動して、他のクラスターでも同じ状況を引き起こす可能性があります。その結果、メッセージがブローカー間を行き来することになります。

7.11.2 以降では、ローカルキューに余分な容量がある場合にのみメッセージのバッチをプルするように、フェデレーテッドコンシューマーを設定できます。その結果、フェデレーションは、フェデレーテッドコンシューマーが処理できる以上のメッセージを移動しなくなるため、ブローカー間でメッセージが行き来する状況を回避できます。

メッセージのバッチをプルするようにフェデレーテッドコンシューマーを設定するには、フェデレーテッドコンシューマーの接続 URI で **consumerWindowSize** 値を **0** に設定します。

```
tcp://<host>:<port>?consumerWindowSize=0
```

consumerWindowSize 値が **0** に設定されている場合、AMQ Broker は、一致するアドレスのアドレス設定の **defaultConsumerWindowSize** 属性値を使用して、ブローカー間を移動するメッセージのバッチサイズを決定します。**defaultConsumerWindowSize** 属性のデフォルト値は **1048576** バイトです。

このバッチ処理の操作モードは、アクティブなブローカー間の双方向フェデレーションに使用します。

フェデレーションの詳細は、[AMQ Broker の設定の アドレスおよびキューのフェデレーション](#) を参照してください。

ブローカー起動ヘルスチェック

7.11.2 以降、OpenShift Container Platform のコンテナ内の AMQ Broker アプリケーションが正常に起動したことを確認する startup プローブを設定できるようになりました。ヘルスチェックの設定方法について、詳しくは [OpenShift での AMQ Broker のデプロイの ブローカーヘルスチェックの設定](#) を参照してください。

ページングに使用されるディスク容量を制限する

メッセージをページングするように AMQ Broker を設定する場合、受信メッセージのページングに

使用されるディスク領域を制限して、ページング操作で過剰なディスク領域が使用されるのを防ぐことができます。詳細については、[AMQ Broker の設定の 特定のアドレスのページング中のディスク使用量の制限](#) を参照してください。

ページングから読み取られるメッセージに使用されるメモリーを制限する

メッセージをページングするように AMQ Broker を設定すると、クライアントがメッセージを消費する準備ができたときにブローカーがディスクからメモリーに転送するメッセージの保存に使用されるメモリーを制限できます。詳細は、[AMQ Broker の設定の ページングされたメッセージのメモリーへのフローの制御](#) を参照してください。

重要

クライアントアプリケーションが、完了通知待ちのメッセージをあまりにも多く残す場合、ブローカーは保留中のメッセージが承認されるまでページングされたメッセージを読み取らないため、ブローカー上でメッセージ枯渇が発生する可能性があります。

たとえば、ページングされたメッセージのメモリーへの転送の制限 (デフォルトでは 20MB) に達すると、ブローカーはそれ以上のメッセージを読み取る前にクライアントからの確認応答を待ちます。同時に、クライアントがブローカーに確認応答を送信する前に十分なメッセージの受信を待機している場合 (クライアントが使用するバッチサイズによって決まります)、ブローカーはメッセージが不足します。

枯渇を回避するには、メモリーへのページングメッセージの転送を制御するブローカーの制限を増やすか、配信メッセージの数を減らします。クライアントがメッセージ確認応答をより早くコミットするか、タイムアウトを使用してブローカーからメッセージを受信しなくなったときに確認応答をコミットするようにすることで、配信メッセージの数を減らすことができます。

配信メッセージの数とサイズは、AMQ 管理コンソールのキューの **Delivering Count** メトリクスと **Delivering Bytes** メトリクスで確認できます。

Log4j 2 ログサポート

7.11 以降、AMQ Broker は、JBoss Logging フレームワークの代わりに Log4j 2 ログユーティリティを使用してメッセージログを提供します。OpenShift Container Platform と RHEL プラットフォームの両方で Log4j 2 ログ設定をカスタマイズできます。

Operator ログレベルの変更

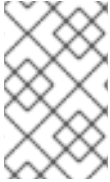
OpenShift Container Platform 上の AMQ Broker 7.11 では、デフォルトのログレベルを変更して、Operator ログに書き込まれる詳細を増減できます。詳細については、[Openshift への AMQ Broker のデプロイ](#) の [Operator ログレベルの変更](#) を参照してください。

Java Authentication and Authorization Service (JAAS) ログインモジュールのサポート

OpenShift Container Platform 上の AMQ Broker 7.11 では、ActiveMQArtemisSecurity CR を使用して AMQ Broker のユーザー認証と承認を設定する代わりに、シークレットで JAAS ログインモジュールを設定できます。JAAS ログインモジュールをシークレットで設定すると、ブローカーを再起動しなくても、プロパティファイル内のユーザーとロールの情報を更新できます。さらに、CR では設定できない LDAP などのログインモジュールを設定できます。詳細については、[OpenShift への AMQ Broker のデプロイ](#) の [シークレットでの JAAS ログインモジュールの設定](#) を参照してください。

アップグレードの制限

OpenShift Container Platform 上の AMQ Broker 7.11 では、Operator がブローカーコンテナイメージを利用可能な最新バージョンに自動的にアップグレードします。デプロイメントのカスタムリソース (CR) を設定して、自動アップグレードを禁止したり、特定のバージョンまたは特定のブローカーおよび初期コンテナイメージへの自動アップグレードのみを許可したりすることができます。



注記

自動アップグレードを制限する場合は、CR で **spec.deploymentPlan.image** 属性と **spec.deploymentPlan.initImage** 属性の組み合わせ、または **spec.version** 属性のいずれかを指定する必要があります (両方を指定することはできません)。

詳細については、OpenShift への AMQ Broker のデプロイの [自動アップグレードの制限](#) を参照してください。

拡張ステータスレポート

OpenShift Container Platform 上の AMQ Broker 7.11 では、メインブローカー CR で Operator によって報告されるステータス情報が拡張されました。

- CR の内容の有効性。
- **BrokerProperties** 属性で設定されたプロパティのアプリケーション。
- シークレット内の Java Authentication and Authorization Service (JAAS) ログインモジュールファイルのブローカー Pod への伝播。
- デプロイされているブローカーのバージョンと、そのバージョンのブローカーおよび初期コンテナイメージの URL。
- メジャー、マイナー、パッチ、およびセキュリティーのアップグレードをデプロイメントに適用する機能。

同期ミラーリングのサポート

7.11 以降では、ブローカー間の同期ミラーリングを設定して、ミラー内の両方のブローカーのボリュームにメッセージが同時に書き込まれるようにすることができます。同期ミラーリングを使用すると、ミラーリングされたブローカーが障害復旧のために最新の状態に保たれます。詳細については、AMQ Broker の設定の [ブローカー接続の設定](#) を参照してください。

Pod の中断バジェット

OpenShift Container Platform 上の AMQ Broker 7.11 では、Pod 中断バジェットを設定して、メンテナンス期間などの自主的な中断中に同時に使用可能にする必要があるクラスター内の Pod の最小数を指定できます。詳細については、OpenShift への AMQ Broker のデプロイで [Pod 中断バジェットの設定](#) を参照してください。

brokerProperties 設定は変更可能なシークレットに保存される

OpenShift Container Platform 上の AMQ Broker 7.11 では、CR の **brokerProperties** 属性を使用して作成された設定は、変更可能なシークレットに保存されます。変更可能なシークレットは、ブローカーを再起動しなくても更新できます。したがって、設定の更新は、特にブローカーの再起動を必要とする更新を除き、ブローカーが設定を定期的に取りロードするときに適用されます。

組み込み Web サーバーを制御するための操作

7.11 以降、ActiveMQServerControl JMX MBean の **stopEmbeddedWebServer**、**startEmbeddedWebServer**、および **restartEmbeddedWebServer** オペレーションを使用して、AMQ Broker の組み込み Web サーバーを停止および再起動できます。これらの操作を使用すると、AMQ Broker の SSL 証明書を更新する場合などに、AMQ Broker の再起動を回避できます。

AMQ 管理コンソールへのログインに使用される認証情報はメッセージの送信に使用される

AMQ Broker の以前のバージョンでは、AMQ 管理コンソールでメッセージを送信するには、AMQ 管理コンソールの **Preferences** ページでユーザー名とパスワードを指定する必要がありました。7.11 以降、メッセージは AMQ 管理コンソールへのログインに使用する認証情報を使用して送信されます。

デフォルトの動作をオーバーライドし、別の認証情報を指定して個別のメッセージを送信できます。詳細については、[AMQ Broker の管理のアドレスへのメッセージの送信](#) を参照してください。

OpenShift Container Platform 上の AMQ Broker が Prometheus のメトリックデータを収集するように事前設定される

OpenShift Container Platform 上の AMQ Broker 7.11 では、Prometheus Operator Service Monitor がメトリックデータを収集できるように AMQ Broker コンテナ Pod が事前設定されています。以前のリリースでは、Service Monitor がメトリックデータを収集するために必要なポートを公開する必要がありました。

ブローカーコンテナの環境変数を設定する

OpenShift Container Platform 上の AMQ Broker 7.11 では、各ブローカーコンテナに渡されるカスタムリソース (CR) に環境変数を設定できます。たとえば、**TZ** 環境変数を追加して、ブローカーコンテナのタイムゾーンを設定できます。詳細については、[OpenShift への AMQ Broker のデプロイのブローカーコンテナの環境変数の設定](#) を参照してください。

OpenShift Container Platform でのプロキシ転送がサポートされる

OpenShift Container Platform 上の AMQ Broker 7.11 では、AMQ 管理コンソールをホスティングする組み込み Web サーバーが、X-Forwarded ヘッダーを処理するように事前設定されています。X-Forwarded ヘッダーを処理することにより、AMQ 管理コンソールは、プロキシがリクエストのパスに関与している場合に変更または失われるヘッダー情報を受け取ることができます。たとえば、AMQ 管理コンソールは HTTP を使用しますが、プロキシである OpenShift Container Platform ルーターは、ルーターで終了する HTTPS ルートを使用して AMQ 管理コンソールを公開します。AMQ 管理コンソールは、X-Forwarded ヘッダーから、ブラウザとルーターの間の接続が HTTPS を使用していることを識別し、ブラウザのリクエストを処理するために HTTPS に切り替えることができます。

一部の再配信属性が brokerProperties CR 属性でのみサポートされる

7.8.x または 7.9.x デプロイメントの **spec.deploymentPlan.addressSettings.addressSetting** CR 要素で **redeliveryDelayMultiplier** または **redeliveryCollisionAvoidanceFactor** 属性が設定されている場合、7.11.x にアップグレードした後、**brokerProperties** CR 属性でこれらの属性を設定する必要があります。7.10.0 では両方の属性のデータ型が float から string に変更されたため、これが必要です。その結果、これらの属性は **spec.deploymentPlan.addressSettings.addressSetting** 属性ではサポートされなくなりました。

次の例は、**brokerProperties** 要素で両方の属性を設定する方法を示しています。

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



注記

brokerProperties 属性では、**spec.deploymentPlan.addressSettings.addressSetting** 要素で使用されていた **redeliveryDelayMultiplier** 属性名の代わりに、**redeliveryMultiplier** 属性名を使用します。

XML 外部エンティティー (XXE) 処理の無効化

broker.xml ファイルに含まれる個別のファイルでブローカー設定をモジュール化する必要がない場合は、XXE セキュリティーの脆弱性から保護するために XXE 処理を無効にできます。可能な場合は、XXE 処理を無効にすることが推奨されます。詳細については、[AMQ Broker の設定の外部](#)

[XML エンティティ \(XXE\) 処理の無効化](#) を参照してください。

JGroups 5.x

7.10.0 より前のバージョンの AMQ Broker では JGroups 3.x が使用されていました。AMQ Broker 7.11 は JGroups 5.x を使用しますが、これには JGroups 3.x との下位互換性がありません。一部のプロトコルとプロトコルプロパティは 2 つの JGroup バージョン間で変更されているため、AMQ Broker 7.11 にアップグレードするときに JGroups スタック設定の変更が必要になる場合があります。

特定のディレクトリー名に展開された AMQ Broker アーカイブの内容

Red Hat Enterprise Linux で AMQ Broker アーカイブを展開すると、アーカイブの内容が現在のディレクトリーの **apache-artemis-2.28.0.redhat-00019** というディレクトリーに展開されます。

Operator チャンネル

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

- **7.11.x** - このチャンネルはバージョン 7.11 のみの更新を提供する長期サポート (LTS) チャンネルです。
- **7.10.x** - このチャンネルはバージョン 7.10 のみの更新を提供する長期サポート (LTS) チャンネルです。



注記

チャン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、適切なチャンネルから 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。

Prometheus メトリックプラグインのクラス名を変更します。

AMQ Broker 7.11 では、AMQ Broker に含まれる Prometheus メトリックプラグインのクラス名が **org.apache.activemq.artemis.core.server.metrics.plugins.ArtemisPrometheusMetricsPlugin** から **com.redhat.amq.broker.core.server.metrics.plugins.ArtemisPrometheusMetricsPlugin** に変更されました。Prometheus メトリックプラグインが AMQ Broker の以前のバージョンで有効になっている場合は、AMQ Broker 7.11 にアップグレードするときに、**broker.xml** 設定ファイル内のクラス名を更新する必要があります。詳細については、[AMQ Broker の管理の ブローカーインスタンスの 7.10.x から 7.11.x へのアップグレード](#) を参照してください。

listProducers API メソッドおよび listProducersInfoAsJSON JMX メソッドによって返されるデータへの変更

AMQ Broker 7.11 では、AMQ 管理コンソールで使用される **listProducers** メソッドによってデータが返される方法が次のように強化されています。

- 以前のバージョンで返されたデータでは、プロデューサーはセッションごとに表示されていました。したがって、プロデューサーごとに 2 つのセッションを作成する Core プロトコルを使用してプロデューサーを作成した場合、2 つのセッションごとに別のプロデューサーが表示されます。また、プロデューサーからメッセージを送信せずにプロデューサーを作成した場合、プロデューサーに返されるアドレスは空でした。AMQ Broker 7.11 では、**listProducers** メソッドは、コアプロトコルによって作成された 2 つのセッションに対して 1 つのプロデューサーを返します。さらに、メッセージを送信する前であっても、アドレス列には正しいアドレスが表示されます。
- 以前は、匿名プロデューサーを使用して Core または AMQP プロトコルを使用して複数のアドレ

スにメッセージを送信すると、アドレスごとにプロデューサーが表示されていました。さらに、表示されたアドレスはプロデューサーがメッセージを送信した最初のアドレスのものであり、単一のキューに送信する通常のプロデューサーのように見えていました。AMQ Broker 7.11 では、匿名プロデューサーを使用して複数のアドレスにメッセージを送信すると、匿名プロデューサーごとに1つのプロデューサーが表示されます。さらに、アドレスは特定のアドレスに関連付けられておらず、**ANONYMOUS** という値を持っています。

以前は、**listProducersInfoAsJSON** メソッドは、特定のセッションによって各キューに送信されたメッセージの数を提供していました。ただし、このメソッドは、メッセージの送信先となる各キューのプロデューサーを誤って返していました。たとえば、匿名プロデューサーが1000個のキューにメッセージを送信した場合、このメソッドは1000個のプロデューサーを返していました。AMQ Broker 7.11 では、**listProducersInfoAsJSON** メソッドは **listProducers** メソッドとまったく同じデータを返しますが、形式は異なります。

AMQ Broker 7.11 では、次の新しいメトリックデータが返されます。

Consumers

messagesInTransit - まだ確認されていない配信メッセージの数

messagesInTransitSize - まだ確認されていない配信メッセージの合計サイズ

messagesDelivered - 配信されたメッセージの数

messagesDeliveredSize - 配信されたメッセージの合計サイズ

messagesAcknowledged - 確認されたメッセージの総数

messagesAcknowledgedAwaitingCommit - 確認されていても、コミットを待っているトランザクション内のメッセージの総数

lastDeliveredTime - 最後にメッセージが配信された時間 (ミリ秒単位)

lastAcknowledgedTime - 最後にメッセージが確認されてからの時間 (ミリ秒単位)

Producers

msgSent - プロデューサーによって送信されたメッセージの数

msgSizeSent - プロデューサーによって送信されたメッセージの合計サイズ

lastProducedMessageID - 最後に送信されたメッセージの ID

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。

カスタムリソースの `upgrade` 属性

7.11 以降、`upgrade` 属性、関連する `enabled` および `minor` 属性は、当初の設計どおりに動作しないため、非推奨になりました。`image` または `version` 属性を使用して、特定のブローカーコンテナイメージをデプロイします。

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 `getAddressesSettings` メソッドは非推奨になりました。`getAddressSettings` 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。8.0 リリースでは、OpenWire プロトコルは AMQ Broker から削除されます。

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。高可用性を実現し、ネットワーク分離の問題を回避するためのブローカークラスターの設定に関するガイダンスについては、[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 削除された機能

このセクションでは、AMQ Broker から削除された機能について説明します。

AMQ Broker Web サーバーのルートへのアクセス

AMQ Broker の以前のバージョンでは、AMQ Broker Web サーバーのルート URL (例: <http://localhost:8161/>) をブラウザウィンドウで開くと、ランディングページが表示されます。ランディングページには、AMQ 管理コンソールと AMQ Broker のドキュメントへのリンクがあります。7.11 では、すべての静的 HTML コンテンツが AMQ Broker から削除されます。したがって、AMQ Broker Web サーバーのルート URL を開いた場合、ランディングページは表示されません。代わりに、ブラウザセッションは自動的に AMQ 管理コンソールにリダイレクトされます。

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は [AMQ Broker 7.11.0 Fixed Issues](#)、パッチリリースで修正された問題のリストは [AMQ Broker - 7.11.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)

このセクションでは、AMQ Broker 7.11 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-6630](#) - CVE-2022-1278 WildFly: 情報漏洩の可能性
- [ENTMQBR-7397](#) - CVE-2022-22970 springframework: multipartFile またはサーブレットパーツへのデータバインディングを介した DoS
- [ENTMQBR-7398](#) - CVE-2022-22971 springframework: WebSocket 上の STOMP による DoS
- [ENTMQBR-7005](#) - CVE-2022-2047 jetty-http: ホスト名入力処理の改善
- [ENTMQBR-7640](#) - CVE-2022-3782 keycloak: 二重 URL エンコードによるパストラバーサル

第8章 既知の問題

このセクションでは、AMQ Broker 7.11 の既知の問題について説明します。

- **ENTMQBR-8106 - AMQ Broker Drainer pod doesn't function properly after changing MessageMigration in CR**
実行中のブローカーデプロイメントでは **messageMigration** 属性の値を変更できません。この問題を回避するには、新しい **ActiveMQ Artemis** CR の **messageMigration** 属性に必要な値を設定し、新しいブローカーデプロイメントを作成する必要があります。
- **ENTMQBR-8166 - UseClientAuth=true の自己署名証明書により、Operator と Jolokia の通信が妨げられる**
ActiveMQ Artemis CR の **console** セクションで **useClientAuth** 属性が **true** に設定されている場合、Operator はブローカー上で特定の機能 (アドレスの作成など) を設定できません。Operator ログに、**remote error: tls: bad certificate** で終わるエラーメッセージが表示されます。
- **ENTMQBR-7385 - 遅いコンシューマーのフェデレーションキューでメッセージがフロップする**
ローカルアプリケーションコンシューマーが非常に遅い場合、またはメッセージを消費できない場合、メッセージはアプリケーションコンシューマーによって最終的に消費されるまでに、フェデレーションされた接続を介して何度も送受信される可能性があります。
- **ENTMQBR-7820 - [Operator] 7.11.0 OPR1 Operator ログにリストされているサポート対象バージョンが間違っている**
Operator ログには、AMQ Broker イメージバージョン 7.10.0、7.10.1、7.10.2、7.11.0、7.8.1、7.8.2、7.8.3、7.9.0、7.9.1、7.9.2、7.9.3、および 7.9.4 のサポートがリストされています。Operator は、実際には、7.10.0 以降の AMQ Broker イメージバージョンをサポートしています。
- **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 がない場合、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-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。詳細については、[OpenShift への AMQ Broker のデプロイ](#) の [OperatorHub](#) を使用した [Operator のアップグレード](#) を参照してください。

- **ENTMQBR-6473 - スキーマ URL の変更により設定に互換性がない**
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、[Linux での 7.9.0 から 7.10.0 へのアップグレード](#) の説明に従って、関連する設定ファイル内のスキーマ URL を更新します。
- **ENTMQBR-4813 大きなメッセージと複数の C++ サブスクリバの発生により 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リバ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、サブスクリバの1つがクラッシュします。
- **ENTMQBR-5749 - OperatorHub に表示されるがサポートされていない Operator を削除する**
[OperatorHub からの Operator のデプロイ](#) で説明されている Operator 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 OperatorHub 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-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-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-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見える**
バックアップブローカーがプライマリブローカーと同期しようとしているときにプライマリブローカーのページングディスクを削除すると、プライマリが失敗します。さらに、バックアップブローカーはプライマリブローカーとの同期を試行し続けるため、ライブになることができません。
- **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 backup 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-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に必要な Persistent Volume Claim (PVC) のサイズを大きくしようとする、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 7.10 リリースノート](#)
- [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: 2024-10-18