Red Hat AMQ Broker 7.12

Red Hat AMQ Broker 7.12 のリリースノート

AMQ Broker のリリースノート

Last Updated: 2024-10-20

Red Hat AMQ Broker 7.12 Red Hat AMQ Broker 7.12 のリリースノート

AMQ Broker のリリースノート

法律上の通知

概要

これらのリリースノートには、AMQ Broker 7.12 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 7.12 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 9

第5章 修正された問題 11

第6章 修正された COMMON VULNERABILITIES AND EXPOSURES (CVE) 12

第7章 既知の問題 14

第8章 重要なリンク 16

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、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 7.12 の長期サポート

AMQ Broker 7.12 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 7.12 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、4.15、および 4.16
- Microsoft Windows Server 2016、2019、2022

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 7.12 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 7.12 の一連の拡張機能と変更された機能を説明します。機能強化の完全なリストは、AMQ Broker 7.12.0 の機能強化 を参照してください。

証明書管理のための Openshift の cert-manager Operator と AMQ Broker の統合

OpenShift 上の AMQ Broker 7.12 では、Openshift 用の cert-manager Operator を使用して、AMQ Broker で TLS を設定するために必要な証明書を作成および管理できます。詳細は、**Openshift への AMQ Broker のデプロイ** の Openshift での cert-manager Operator の使用 を参照してください。

AMQ Broker Openshift サービスが TLS 証明書を提供する

同じ Openshift クラスター上のブローカーとクライアント間の内部接続を保護する場合は、アクセプターサービスにアノテーションを追加して、Openshift がサービス提供 TLS 証明書を生成するように要求できます。詳細は、**Openshift への AMQ Broker のデプロイ** の Openshift サービス提供証明書の使用 を参照してください。

プライバシー強化メール (PEM) 証明書のサポート

AMQ Broker 7.12 では、PEM 形式の TLS 証明書のサポートが追加されました。

制限されたポリシーを持つ Openshift namespace でのデプロイメントのサポート

デフォルトでは、制限された OpenShift セキュリティーコンテキスト制約を持つ namespace の OpenShift に AMQ Broker 7.12 をデプロイできます。ブローカーを別の OpenShift セキュリティーコンテキストで実行する場合は、Pod セキュリティーオプションに加えて、CR でコンテナーセキュリティーオプションをカスタマイズできます。詳細は、**Openshift への AMQ Broker のデプロイ** の カスタムリソース設定リファレンス を参照してください。

brokerProperties 設定の分離

OpenShift 上の AMQ Broker 7.12 のカスタムリソース (CR) に **brokerProperties** セクションが含まれており、CR が最大サイズ制限の 1 MB に達している場合は、**brokerProperties** 設定を 1 つ以上の Java プロパティーファイルに分離し、CR で参照することができます。メンテナンスを容易にするために、**brokerProperties** 設定を別のファイルに分離して、**brokerProperties** 項目を論理的にグループ化することもできます。詳細は、**Openshift への AMQ Broker のデプロイ** の **brokerProperties** 設定の分離 を参照してください。

Openshift のルートに加えて Ingress もサポート

OpenShift 上の AMQ Broker 7.12 では、ルートに加えて Ingress を使用して、Openshift クラスターの外部にあるクライアントにアクセプター、コネクター、および管理コンソールを公開できます。詳細は、**Openshift への AMQ Broker のデプロイ** の アクセプターの設定 を参照してください。

サードパーティーの JAR ファイルの共有ボリュームのマウントのサポート

OpenShift 上の AMQ Broker 7.12 では、クラスター内の各ブローカー Pod に共有ボリュームをマウントするように Operator を設定できます。各 Pod に共有ボリュームをマウントするユースケースとしては、ブローカーに必要なサードパーティーの JAR ファイル (JDBC データベースの JAR ファイルなど) を保存することが挙げられます。RHEL と Openshift の両方のプラットフォームで、Java クラスパスを拡張して、実行時にブローカーが追加の JAR ファイルを利用できるようにすることができます。詳細は、**Openshift への AMQ Broker のデプロイ** の サードパーティーの JAR ファイルの追加 を参照してください。

Operator により作成された Openshift リソースのカスタマイズ

OpenShift 上の AMQ Broker 7.12 では、Operator によって作成および管理されるデプロイメント、Pod、サービスなどの Openshift リソースをカスタマイズできます。これらのリソースをカスタマイズすると、次のような特定のタスクを実行する場合に役立ちます。

- 他のサービスによるリソースの処理方法を制御するカスタムアノテーションを追加します。
- ブローカー CR で公開されていない属性を変更します。
  詳細は、**Openshift への AMQ Broker のデプロイ** の Operator によって作成された Openshift リソースのカスタマイズ を参照してください。

Openshift 上の AMQ Broker へのプラグインの追加のサポート

OpenShift 上の AMQ Broker 7.12 では、CR にプラグインを登録することで AMQ Broker の機能を拡張できます。詳細は、**Openshift への AMQ Broker のデプロイ** の AMQ Broker へのプラグインの登録 を参照してください。

クラスター接続を保護のサポート

Openshift 上の AMQ Broker 7.12 では、内部アクセプターとコネクターに対して SSL を有効にすることで、クラスター接続を保護できます。詳細は、**Openshift への AMQ Broker のデプロイ** の クラスター接続のセキュリティー保護 を参照してください。

SSL アーティファクトの自動リロード

OpenShift および RHEL 上の AMQ Broker 7.12 では、ブローカーを再起動せずに、更新された TLS 証明書や、キーストアまたはトラストストア設定へのその他の変更を再ロードするように AMQ Broker を設定できます。自動リロードを設定するには、アクセプターの `sslAutoReload` 属性を設定します。Openshift で SSL アーティファクトの自動リロードを設定する方法の例は、**Openshift への AMQ Broker のデプロイ** の Openshift での cert-manager Operator の使用 を参照してください。

クラスター化されたブローカーのヘルスチェック

AMQ Broker 7.12 では、**artemis check cluster** コマンドラインユーティリティーを使用して、クラスター内のブローカーノードのトポロジーを検証できます。詳細は、**AMQ Broker の管理** の ブローカー、キュー、クラスターの正常性の確認 を参照してください。

AMQP ブローカー接続を使用したフェデレーションサポート

AMQ Broker 7.12 では、アウトバウンド AMQP ブローカー接続を介してアドレスとキューのフェデレーションを設定できます。フェデレーションに AMQP プロトコルを使用すると、Core プロトコルを使用する場合と比べて次の利点があります。

- クライアントがメッセージングに AMQP プロトコルを使用する場合は、フェデレーションに AMQP プロトコルを使用して、AMQP と Core 間のメッセージの変換を排除します。
- AMQP フェデレーションは、単一の送信接続を介した双方向フェデレーションをサポートします。双方向サポートにより、リモートブローカーがローカルブローカーに接続する必要がなくなります。これは、フェデレーションに Core プロトコルを使用する場合の要件であり、ネットワークポリシーによって禁止される可能性があります。
- AMQP フェデレーションは、ブローカー間のメッセージの移動をより適切に制御し、ブローカー間でメッセージが行き来するのを防ぎます。

詳細は、**AMQ ブローカーの設定** の AMQP プロトコルを使用したフェデレーションの設定 を参照してください。

コマンドラインインターフェイスからカスタムシェルを使用する

AMQ Broker 7.12 では、AMQ Broker コマンドラインインターフェイスからカスタム **artemis** シェルを使用してブローカーと対話できます。カスタムシェルには、コマンドとコマンドパラメーターの自動補完機能が組み込まれています。詳細は、**AMQ Broker の管理** の **artemis** シェルでの CLI の使用 を参照してください。

ワイルドカードを含むアドレスのリテラルマッチング

AMQ Broker 7.12 では、ワイルドカードを含むアドレスの一致に対して、ワイルドカード文字をリテラル文字として扱うようにリテラル一致を設定できます。詳細は、**AMQ Broker の設定** の リテラル一致の設定 を参照してください。

JMX 管理操作のためのロールベースのアクセス制御

AMQ Broker 7.12 では、ブローカーを再起動せずに、**view** と **edit** の 2 つの新しい権限を使用して、JMX 管理操作のロールベースのアクセス制御を設定できます。RHEL でのロールベースのアクセス制御の設定は、**AMQ Broker の設定** の **broker.xml** ファイルでのロールベースのアクセス制御の設

定 を参照してください。Openshift でのロールベースのアクセス制御の設定は、**Openshift への AMQ Broker のデプロイ** の 管理操作用のロールベースのアクセス制御の設定 を参照してください。

キュー統計コマンドの出力形式を変更する

queue stat コマンドの出力形式は、AMQ Broker の 7.11 以前のバージョンから変更されており、デプロイメントで実行する自動プロセスに影響する可能性があります。

MQTT アクセプターで設定可能な新しいパラメーターにより、MQTT サブスクリプションキューを自動的に削除できるようになりました。

AMQ Broker 7.12 では、MQTT アクセプターで **defaultMqttSessionExpiryInterval** パラメーターを設定して、対応するクライアントセッションの有効期限が切れたときに削除されない MQTT サブスクリプションキューを自動的に削除できます。新しいパラメーターは、クライアントが切断されてからブローカーがセッション状態とサブスクリプションキューを削除するまでに経過する必要がある **秒** 数を表します。7.12 より前では、クライアントセッションの有効期限が切れたときに削除されなかったキューを削除するには、**address-setting** で **auto-delete-*** パラメーターを設定する必要がありました。

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。

- **7.12.x** - このチャネルはバージョン 7.12 のみの更新を提供する長期サポート (LTS) チャネルです。
- **7.11.x** - このチャネルはバージョン 7.11 のみの更新を提供する長期サポート (LTS) チャネルです。
- **7.10.x** - このチャネルはバージョン 7.10 のみの更新を提供する長期サポート (LTS) チャネルです。

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。

プログラムの例

AMQ Broker 7.12 では、サンプルプログラムはブローカーとともに配布およびインストールされなくなりました。代わりに、次のリポジトリーにあるサンプルプログラムにアクセスできます: https://github.com/apache/activemq-artemis-examples。

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を説明します。

ActiveMQArtemisAddress CRD

7.12 以降では、**ActiveMQArtemisAddress** CRD は非推奨になります。**ActiveMQArtemis** CR の **spec.brokerProperties** 属性を使用して、デプロイメントのアドレスとキューを作成します。

ActiveMQArtemisSecurity CRD

7.12 以降では、**ActiveMQArtemisSecurity** CRD は非推奨になります。**ActiveMQArtemis** CR の **spec.brokerProperties** 属性を使用して、デプロイメントのセキュリティーを設定します。

ActiveMQArtemisScaledown CRD

7.12 以降では、**ActiveMQArtemisScaledown** CRD は非推奨になります。**ActiveMQArtemisScaledown** CRD はブローカーによって内部的に使用されるため、この変更は AMQ Broker 管理者には透過的です。

LDAP クエリーの接続プール

7.12 以降では、LDAP クエリーの接続プールを有効にする **connectionPool** パラメーターは非推奨になりました。組み込みの承認および認証キャッシュは、LDAP クエリーのパフォーマンスを最適化する別の方法を提供します。組み込みキャッシュのカスタマイズは、認証および承認キャッシュの設定 を参照してください。

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ー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 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。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。

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 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 7.12.0 Fixed Issues 、パッチリリースで修正された問題のリストは AMQ Broker - 7.12.x Resolved Issues を参照してください。

第6章 修正された COMMON VULNERABILITIES AND EXPOSURES (CVE)

このセクションでは、AMQ Broker 7.12 リリースで修正された Common Vulnerabilities and Exposures (CVE) を詳しく説明します。

- ENTMQBR-8644 - TRIAGE CVE-2023-6717 keycloak: SAML POST バインディングフローのアサーションコンシューマーサービス URL 経由の XSS [amq-7]
- ENTMQBR-8976 - TRIAGE CVE-2024-29025 netty-codec-http: 制限やスロットルなしでのリソースの割り当て [amq-7]
- ENTMQBR-8927 - CVE-2024-22259 springframework: ホスト検証による URL 解析 [amq-7]
- ENTMQBR-8740 - CVE-2024-1132 keycloak: リダイレクト検証におけるパス横断 [amq-7]
- ENTMQBR-8758 - CVE-2024-1249 keycloak: org.keycloak.protocol.oidc: checkLoginIframe 内の検証されていないクロスオリジンメッセージが DDoS を引き起こす [amq-7]
- ENTMQBR-8626 - CVE-2023-6378 logback: logback レシーバーのシリアル化脆弱性 [amq-7]
- ENTMQBR-8627 - CVE-2023-6481 logback: logback レシーバーのシリアル化脆弱性 [amq-7]
- ENTMQBR-8953 - CVE-2024-29131 CVE-2024-29133 commons-configuration2: さまざまな不具合 [amq-7]
- ENTMQBR-8702 - CVE-2023-44981 zookeeper: Apache ZooKeeper での認証バイパス [amq-7]
- ENTMQBR-8611 - CVE-2022-41678 activemq: Apache ActiveMQ: 認証されたユーザーが RCE を実行できる Jolokia のデシリアライゼーション脆弱性 [amq-7]
- ENTMQBR-8225 - CVE-2023-24540 amq-broker-rhel8-operator-container: golang: html/template: JavaScript の空白の不適切な処理 [amq-7]
- ENTMQBR-8227 - CVE-2022-21698 amq-broker-rhel8-operator-container: prometheus/client_golang: InstrumentHandlerCounter を使用したサービス拒否 [amq-7]
- ENTMQBR-8238 - CVE-2022-21698 CVE-2023-24534 amq-broker-rhel8-operator-container: golang: net/http, net/textproto: 過剰なメモリー割り当てによるサービス拒否 [amq-7]
- ENTMQBR-8239 - CVE-2023-29400 amq-broker-rhel8-operator-container: golang: html/template: 空の HTML 属性の不適切な処理 [amq-7]
- ENTMQBR-8240 - CVE-2023-24539 amq-broker-rhel8-operator-container: golang: html/template: CSS 値の不適切なサニタイズ [amq-7]
- ENTMQBR-8228 - CVE-2021-43565 amq-broker-rhel8-operator-container: golang.org/x/crypto: 空のプレーンテキストパケットによりパニックが発生する [amq-7]
- ENTMQBR-8230 - CVE-2022-41723 amq-broker-rhel8-operator-container: net/http, golang.org/x/net/http2: HPACK デコードにおける二次複雑性の回避 [amq-7]

- ENTMQBR-8236 - CVE-2023-24536 amq-broker-rhel8-operator-container: golang: net/http, net/textproto, mime/multipart: 過剰なリソース消費によるサービス拒否 [amq-7]
- ENTMQBR-8237 - CVE-2023-24537 amq-broker-rhel8-operator-container: golang: go/parser: 解析中の無限ループ [amq-7]
- ENTMQBR-8231 - CVE-2022-2879 amq-broker-rhel8-operator-container: golang: archive/tar: ヘッダーの読み取り時に無制限のメモリー消費が発生する [amq-7]
- ENTMQBR-8229 - CVE-2022-27664 amq-broker-rhel8-operator-container: golang: net/http: GOAWAY 送信後のサーバーエラーを処理する [amq-7]
- ENTMQBR-8226 - CVE-2022-32189 amq-broker-rhel8-operator-container: golang: math/big: エンコードされたメッセージが短すぎる場合、big.Float および big.Rat 型のデコード時にパニックが発生し、サービス拒否攻撃が発生する可能性がある [amq-7]
- ENTMQBR-8232 - CVE-2022-41715 amq-broker-rhel8-operator-container: golang: regexp/syntax: 正規表現の解析で使用されるメモリーを制限する [amq-7]
- ENTMQBR-8241 - CVE-2023-24538 amq-broker-rhel8-operator-container: golang: html/template: バックティックが文字列区切り文字として扱われない [amq-7]
- ENTMQBR-8233 - CVE-2022-2880 amq-broker-rhel8-operator-container: golang: net/http/httputil: ReverseProxy は解析できないクエリーパラメーターを転送してはならない [amq-7]
- ENTMQBR-8234 - CVE-2022-41724 amq-broker-rhel8-operator-container: golang: crypto/tls: 大きなハンドシェイクレコードによりパニックが発生する可能性がある [amq-7]
- ENTMQBR-8608 - CVE-2022-41678 activemq-broker-operator: Apache ActiveMQ: 認証されたユーザーが RCE を実行できる Jolokia のデシリアライゼーション脆弱性 [amq-7]
- ENTMQBR-8235 - CVE-2022-41725 amq-broker-rhel8-operator-container: golang: net/http, mime/multipart: 過剰なリソース消費によるサービス拒否 [amq-7]
- ENTMQBR-8671 - CVE-2023-51074 json-path: Criteria.parse メソッドのスタックベースのバッファーオーバーフロー [amq-7]

第7章 既知の問題

このセクションでは、AMQ Broker 7.12 の既知の問題を説明します。

- ENTMQBR-9103 - **AMQP を消費する複数のスレッドを閉じるときに NullPointerException が発生する**
  AMQP メッセージのマルチスレッドコンシューマーを実行すると、ブローカーは次のような WARN レベルのログメッセージを生成することがあります。

  2024-05-13 18:11:46,048 WARN [io.netty.util.concurrent.AbstractEventExecutor] タスクで例外が発生する。タスク:
  org.apache.activemq.artemis.protocol.amqp.proton.AMQPLargeMessageWriter$$Lambda$643/0
  java.lang.NullPointerException: null

  メッセージは、クライアントがメッセージの消費を完了したときに生成され、スタックトレースが付随することもあります。

  メッセージは失われず、無視できます。

- 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-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-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、**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-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-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8章 重要なリンク

- Red Hat AMQ Broker 7.11 リリースノート
- 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: 2024-09-05