



Red Hat AMQ Broker 7.11

Red Hat AMQ Broker 7.11 릴리스 정보

AMQ Broker용 릴리스 정보

Red Hat AMQ Broker 7.11 Red Hat AMQ Broker 7.11 릴리스 정보

AMQ Broker용 릴리스 정보

법적 공지

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.

초록

이 릴리스 노트에는 AMQ Broker 7.11 릴리스에 포함된 새로운 기능, 개선 사항, 수정 사항 및 문제에 대한 최신 정보가 포함되어 있습니다.

차례

보다 포괄적 수용을 위한 오픈 소스 용어 교체	3
1장. AMQ BROKER 7.11에 대한 장기 지원	4
2장. 지원되는 구성	5
3장. 새로운 기능 및 변경된 기능	6
4장. 더 이상 사용되지 않는 기능	18
5장. 삭제된 기능	19
6장. 해결된 문제	20
7장. 수정된 COMMON VULNERABILITIES AND EXPOSURES	21
8장. 확인된 문제	22
9장. 중요한 링크	27

보다 포괄적 수용을 위한 오픈 소스 용어 교체

Red Hat은 코드, 문서, 웹 속성에서 문제가 있는 용어를 교체하기 위해 최선을 다하고 있습니다. 먼저 마스터(master), 슬레이브(slave), 블랙리스트(blacklist), 화이트리스트(whitelist) 등 네 가지 용어를 교체하고 있습니다. 이러한 변경 작업은 작업 범위가 크므로 향후 여러 릴리스에 걸쳐 점차 구현할 예정입니다. 자세한 내용은 [CTO Chris Wright의 메시지](#)를 참조하십시오.

1장. AMQ BROKER 7.11에 대한 장기 지원

AMQ Broker 7.11은 장기 지원 (LTS) 릴리스 버전으로 지정되었습니다. LTS 릴리스의 약관에 대한 자세한 내용은 [AMQ LTS 릴리스의 지원 기간](#)을 참조하십시오.

Red Hat Enterprise Linux 및 OpenShift Container Platform 지원

AMQ Broker 7.11 LTS 버전은 다음을 지원합니다.

- Red Hat Enterprise Linux 7, 8 및 9
- OpenShift Container Platform 4.12, 4.13, 4.14 또는 4.15

Red Hat은 AMQ Broker가 향후 OpenShift Container Platform 버전과의 호환성을 유지하기 위해 노력합니다. 그러나 이 호환성은 보장되지 않습니다. 각 새로운 OpenShift Container Platform 버전에 대해 상호 운용성 테스트가 수행됩니다. 호환성 문제가 없는 경우 [Red Hat AMQ Broker 7 지원 구성](#)에 새 OpenShift Container Platform 버전이 추가됩니다.

2장. 지원되는 구성

지원되는 구성에 대한 자세한 내용은 [Red Hat AMQ Broker 7 지원 구성을 참조하십시오](#).

최소 Java 버전

최소한 AMQ Broker 7.11을 실행하려면 Java 버전 11이 필요합니다.

OpenWire 지원

AMQ 7 Broker는 클라이언트 애플리케이션을 AMQ 7로 마이그레이션하는 방법으로 2017년에 출시된 이후 OpenWire 프로토콜에 대한 지원을 제공했습니다. 2021년 AMQ Broker 7.9.0이 릴리스되면서 OpenWire 프로토콜이 더 이상 사용되지 않으며 고객은 기존 OpenWire 클라이언트 애플리케이션을 AMQ 7(CORE, AMQP, MQTT 또는 STOMP)의 완전히 지원되는 프로토콜 중 하나로 마이그레이션하도록 권장되었습니다. AMQ Broker 8.0 릴리스부터 OpenWire 프로토콜은 AMQ Broker에서 제거됩니다.

3장. 새로운 기능 및 변경된 기능

이 섹션에서는 AMQ Broker 7.11의 개선 사항 및 새로운 기능 집합을 설명합니다.

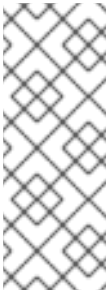
CR의 이미지 및 버전 속성에 대한 검증 변경

7.11.5에서 Operator는 CR의 이미지 및 버전 속성 구성의 구성을 이전 버전과 다르게 검증합니다. 7.11.5 이전에는 Operator에 CR이 포함되어 있지 않은지 확인합니다.

- **spec.deploymentPlan.initImage** 속성이 없는 **spec.deploymentPlan.image** 속성 또는 그 반대의 경우도 마찬가지로입니다.
- **spec.deploymentPlan.image** 및 **spec.deploymentPlan.initImage** 속성이 있거나 둘 다 있는 **spec.version** 속성입니다.

7.11.5에서 Operator는 **spec.deploymentPlan.initImage** 속성 없이 CR에 **spec.deploymentPlan.image** 속성이 없거나 그 반대의 경우도 있는지 계속 확인합니다. 7.11.5에서는 Operator 검증도 다음과 같이 변경되었는지 확인합니다.

- **spec.version** 속성에 지정된 버전 번호(있는 경우)는 배포된 브로커 컨테이너 이미지의 버전과 일치합니다.
버전이 동일하지 않으면 Operator는 CR의 **BrokerVersionAligned** 조건을 정보 목적에 대한 불일치를 강조 표시하도록 **Unknown** 으로 설정하지만 불일치는 배포 중 브로커 실행에는 영향을 미치지 않습니다.
- CR에는 **spec.version** 속성이 없는 **spec.deploymentPlan.image** 및 **spec.deploymentPlan.initImage** 속성이 없습니다.
CR에 **spec.deploymentPlan.image** 및 **spec.deploymentPlan.initImage** 속성이 없는 경우 Operator는 CR의 **Valid** 조건의 상태를 **Unknown** 으로 설정하여 구성이 불완전하다고 경고합니다.



참고

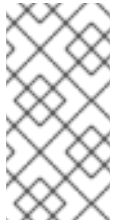
spec.version 속성이 누락되면 Operator가 Operator를 업그레이드할 때마다 배포에서 브로커 Pod를 다시 시작합니다. **spec.version** 속성에 버전 번호를 명시적으로 설정하지 않는 한 Operator는 **StatefulSet**의 레이블을 최신 지원 브로커 버전으로 업데이트하므로 Pod를 다시 시작해야 합니다.

향후 각 Operator 업그레이드가 브로커를 다시 시작하지 않도록 하려면 CR의 **spec.version** 속성에 배포된 브로커의 버전 번호를 설정해야 합니다.

- 7.11.5에서는 브로커를 시작한 후 CR의 **status** 섹션에 배포된 브로커의 버전 번호를 찾을 수 있습니다. 자세한 내용은 [OpenShift에 AMQ Broker 배포에서 브로커 배포에 대한 상태 정보 보기를 참조하십시오.](#)
- 7.11.5 이전에는 **OpenShift Container Platform** 웹 콘솔을 사용하여 브로커 Pod의 로그 파일에서 버전 번호를 찾을 수 있습니다.

- i. 워크로드 → 포드를 클릭합니다.
- ii. 브로커 포드 이름을 클릭합니다.
- iii. 로그 탭을 클릭합니다.

버전 번호는 로그 출력 상단에 있는 **Artemis** 배너 뒤에 표시됩니다.



참고

상태 값이 **Unknown** 인 상태에서는 **Operator**가 브로커 배포를 완료하지 못하도록 하지 않습니다.

Operator의 리더 선택 설정을 사용자 정의할 수 있습니다

7.11.5부터는 **Operator**가 리더 선택을 위해 사용하는 설정을 사용자 지정할 수 있습니다. **Operator Hub**를 사용하여 **Operator**를 설치하는 경우 **OpenShift Container Platform** 웹 콘솔을 사용하여 **Operator** 서브스크립션을 설치한 후 리더 선택 설정을 구성할 수 있습니다. **OpenShift Container Platform** 명령줄 인터페이스를 사용하여 **Operator**를 설치하는 경우 **Operator** 구성 파일 `operator.yaml` 에서 리더 선택 설정을 구성하는 경우 **Operator**를 설치하기 전이나 후에 **Operator** 구성 파일에서 리더 선택을 구성할 수 있습니다.

다음은 `operator.yaml` 파일에 구성된 리더 선택 설정의 예입니다.

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

다음은 **Operator** 서브스크립션에 구성된 리더 선택 설정의 예입니다.

```
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: Operator가 다른 인스턴스와의 리더 선택에서 경쟁하여 한 번에 둘 이상의 인스턴스가 작동하지 않도록 합니다.

Lease-duration: Leader가 아닌 Operator가 이전 리더에 의해 갱신되지 않은 리스를 받기 전에 대기하는 기간(초)입니다. 기본값은 15 입니다.

renew-deadline: 리더 Operator가 리더 역할을 갱신하려고 할 때까지 몇 초 내에 리더 역할을 갱신합니다. 기본값은 10입니다.

retry-period: Operator가 리더 역할을 획득하고 갱신하기 위한 시도 사이에 대기하는 기간(초)입니다. 기본값은 2 입니다.

address-settings 요소의 개별 속성을 업데이트할 수 있습니다.

7.11.5에서는 Jolokia REST 인터페이스를 사용하여 JSON 형식의 **address-settings** 요소에서 개별 속성을 업데이트할 수 있습니다. 이전 버전에서는 **address-settings** 요소에서 속성 또는 속성의 하위 집합을 업데이트하려면 업데이트 작업에 변경되지 않은 **address-settings** 특성을 모두 포함해야 했습니다.

경고 수준 메시지가 **Critical Analyzer**에서 오류 수준으로 변경되었습니다.

7.11.5에서 **Critical Analyzer**는 이전에 **WARN** 수준이 할당된 메시지에 **ERROR** 수준을 할당합니다.

페이지링된 메시지의 메모리 흐름을 보다 효과적으로 제어할 수 있는 새 매개변수

7.11.3 이전에는 **max-read-page-bytes** 및 **max-read-page-messages** 매개변수에 대한 제한을 설정하여 페이지링된 메시지의 흐름을 메모리로 제어합니다. 이러한 제한을 적용할 때 브로커는 메모리에 있는 메시지를 모두 계산하여 소비자와 현재 전달 중인 메시지를 전달할 준비가 된 메시지를 계산합니다. 소비자가 메시지를 확인하는 속도가 느리면 현재 전달되는 메시지에 메모리 또는 메시지 제한을 사용할 수 있으므로 브로커가 새 메시지를 메모리에 읽는 것을 방지할 수 있습니다. 그 결과 브로커에게 메시지가 표시될 수 있습니다.

7.11.3부터는 두 개의 새로운 매개변수에 대한 제한을 설정하여 페이지링된 메시지의 메모리 흐름을 제어할 수 있습니다. 이러한 제한을 적용할 때 브로커는 메시지 전달을 고려하지 않습니다.

- 호출된 메시지를 큐당 메모리로 읽는 데 사용할 수 있는 미리 페이지바이트 메모리(바이트)입니다. 기본값은 **20MB**입니다.
- **prefetch-page-messages** 브로커가 디스크에서 큐당 메모리로 읽을 수 있는 페이지링된 메시지 수입니다. 기본값은 **-1**이며, 이는 제한이 적용되지 않음을 의미합니다.

소비자가 메시지를 승인하는 속도가 느리면 **max-read-page-bytes** 및 **max-read-page-message** 매개변수의 기본 제한을 늘려 메시지를 전달하는 용량을 제공할 수 있습니다. **prefetch-page-bytes** 및 **prefetch-page-messages** 매개변수에 대한 기본 제한으로 인해 브로커가 새 메시지를 메모리에 읽을 수 있습니다.



참고

prefetch-page-bytes 매개변수의 값 앞에 **max-read-page-bytes** 매개변수의 값에 도달하면 브로커는 추가 페이지가 지정된 메시지를 메모리로 읽는 것을 중지합니다.

AMQ Core Protocol JMS 클라이언트는 클러스터의 다른 라이브 브로커로 장애 조치될 수 있습니다.

7.11.2 이전에는 **AMQ Core Protocol JMS** 클라이언트에서 라이브 브로커에 대한 연결이 끊어지면 두 브로커가 **HA**(고가용성)를 위해 라이브/백업 쌍으로 구성된 경우에만 백업 브로커에 장애 조치할 수 있습니다. 7.11.2부터 **AMQ Core Protocol JMS** 클라이언트는 **HA** 쌍의 백업 브로커 또는 클러스터의 다른 라이브 브로커에 장애 조치하도록 구성할 수 있습니다.

클라이언트가 클러스터의 라이브 브로커로 장애 조치(**failover**)를 활성화하려면 클라이언트의 연결 **URI**에 **failoverAttempts** 구성 옵션을 지정합니다. 예를 들면 다음과 같습니다.

```
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 옵션을 2 값으로 설정하면 클라이언트는 클러스터의 다른 두 개의 라이브 브로커에 연결을 시도합니다. 클라이언트가 비**HA** 구성의 라이브 브로커 및 **HA** 구성의 라이브 및 백업 브로커에 대한 모든 연결 시도에 실패하면 장애 조치 시도가 발생합니다.



참고

예제에 표시된 **reconnectAttempts** 구성 옵션은 두 브로커가 **HA** 쌍으로 구성된 경우 라이브 브로커에서만 백업 브로커로 장애 조치하는 데 사용됩니다.

AMQ Broker에서 JDBC 지속성을 사용하는 경우 성능 개선 사항 페이지

7.11.2부터 AMQ Broker가 JDBC 지속성을 사용하도록 구성되면 페이지 성능이 향상됩니다. 페이지 개선 사항의 일부로 새 매개변수인 `jdbc-max-page-size-bytes` 는 JDBC를 사용할 때 기본적으로 페이지 크기를 100KB로 제한합니다. 기본 제한을 사용자 지정할 수 있습니다. 자세한 내용은 [AMQ Broker 구성에서 JDBC 지속성 구성](#) 을 참조하십시오.

페더레이션 메시지 배치

대기열의 백로그가 로컬 소비자의 사용 가능한 용량을 초과하면 우선 순위가 낮은 페더레이션 소비자는 메시지를 수신할 후보가 됩니다. 결국 너무 많은 메시지가 페더레이션 소비자로 이동할 수 있고 다른 클러스터에서 동일한 시나리오가 발생하여 브로커 간에 메시지가 이동할 수 있습니다.

7.11.2부터는 로컬 큐에 초과 용량이 있는 경우에만 메시지 일괄 처리를 가져오도록 페더레이션 소비자를 구성할 수 있습니다. 결과적으로 페더레이션은 통합 소비자가 처리할 수 있는 것보다 더 많은 메시지를 이동하지 않으므로 브로커 간에 메시지가 이동하는 시나리오를 방지할 수 있습니다.

메시지 일괄 처리를 가져오도록 페더레이션 소비자를 구성하려면 페더레이션 소비자의 연결 URI에서 `consumerWindowSize` 값을 0 으로 설정합니다.

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

`consumerWindowSize` 값을 0 으로 설정하면 AMQ Broker는 일치하는 주소에 대한 주소 설정에서 `defaultConsumerWindowSize` 속성 값을 사용하여 브로커 간에 이동되는 메시지의 배치 크기를 결정합니다. `defaultConsumerWindowSize` 속성의 기본값은 1048576 바이트입니다.

활성 브로커 간 양방향 페더레이션에 이 배치 모드를 사용합니다.

페더레이션에 대한 자세한 내용은 [AMQ Broker 구성에서 주소 및 큐 구성](#) 을 참조하십시오.

브로커 시작 상태 점검

7.11.2부터 시작 프로브를 구성하여 OpenShift Container Platform 컨테이너 내의 AMQ Broker 애플리케이션이 성공적으로 시작되었는지 확인할 수 있습니다. 상태 점검을 구성하는 방법을 알아보려면 [Openshift에 AMQ Broker 배포에서 브로커 상태 점검 구성](#) 을 참조하십시오.

페이징에 사용되는 디스크 공간 제한

AMQ Broker를 페이지 메시지에 맞게 구성하는 경우 페이징 작업이 과도한 디스크 공간을 사용하지 못하도록 들어오는 메시지를 페이징하는 데 사용되는 디스크 공간을 제한할 수 있습니다. 자세한 내용은 [AMQ Broker 구성의 특정 주소에 대한 페이징 시 디스크 사용량 제한](#) 을 참조하십시오.

페이징에서 읽은 메시지에 사용되는 메모리 제한

AMQ Broker를 페이지 메시지로 구성하는 경우 클라이언트가 메시지를 사용할 준비가 되면 브로커가 디스크에서 메모리로 다시 전송하는 메시지를 저장하는 데 사용되는 메모리를 제한할 수 있습니다. 자세한 내용은 [AMQ Broker 구성에서 페이지링된 메시지의 흐름을 메모리로 제어를 참조하십시오.](#)

중요

클라이언트 애플리케이션이 승인 보류 중 메시지를 너무 많이 남겨 두면 브로커는 보류 중인 메시지가 승인될 때까지 페이지가 지정된 메시지를 읽지 않으므로 브로커에서 메시지 중단을 유발할 수 있습니다.

예를 들어, 페이지링된 메시지를 메모리로 전송하는 데 대한 제한이 기본적으로 20MB인 경우 브로커는 더 많은 메시지를 읽기 전에 클라이언트에서 승인을 기다립니다. 동시에 클라이언트가 브로커에 승인을 보내기 전에 충분한 메시지를 받기를 기다리는 경우 클라이언트가 사용하는 배치 크기에 따라 결정됩니다.

중단을 방지하기 위해 페이지링된 메시지를 메모리로 전송하거나 메시지 전달 수를 줄이는 브로커 제한을 늘립니다. 메시지를 더 빨리 커밋하거나 브로커에서 더 이상 메시지가 수신되지 않으면 시간 초과를 커밋하고 승인을 커밋하도록 하여 전송 메시지 수를 줄일 수 있습니다.

AMQ Management Console에서 큐의 제공 수 및 Delivering Cryostats 메트릭에서 메시지 전달 의 수와 크기를 확인할 수 있습니다.

Log4j 2 로깅 지원

7.11부터 AMQ Broker는 JBoss Logging 프레임워크 대신 Log4j 2 로깅 유틸리티를 사용하여 메시지 로깅을 제공합니다. OpenShift Container Platform 및 RHEL 플랫폼 모두에서 Log4j 2 로깅 구성을 사용자 지정할 수 있습니다.

Operator 로깅 수준 변경

OpenShift Container Platform의 AMQ Broker 7.11에서는 기본 로깅 수준을 변경하여 Operator 로그에 기록된 세부 정보를 늘리거나 줄일 수 있습니다. 자세한 내용은 [Openshift에 AMQ Broker 를 배포 할 때 Operator의 로깅 수준 변경을 참조하십시오.](#)

JAAS(Java Authentication and Authorization Service) 로그인 모듈 지원

OpenShift Container Platform의 AMQ Broker 7.11에서는 ActiveMQArtemisSecurity CR을 사용하여 AMQ Broker에 대한 사용자 인증 및 권한 부여를 구성하지 않고 시크릿에 CloudEvent 로그인 모듈을 구성할 수 있습니다. 시크릿에 CloudEvent 로그인 모듈을 구성하면 브로커를 재시작하지 않고도 속성 파일에서 사용자 및 역할 정보를 업데이트할 수 있습니다. 또한 CR에서 구성할 수 없는 LDAP와 같은 로그인 모듈을 구성할 수 있습니다. 자세한 내용은 [Openshift에서 AMQ Broker 배포의 시크릿에서 ScanSetting 로그인 모듈 구성을 참조하십시오.](#)

업그레이드 제한

OpenShift Container Platform의 AMQ Broker 7.11에서 Operator는 브로커 컨테이너 이미지를 사용 가능한 최신 버전으로 자동으로 업그레이드합니다. 자동 업그레이드를 방지하거나 특정 브로커 및 init 컨테이너 이미지로만 자동 업그레이드할 수 있도록 배포에 대해 CR(사용자 정의 리소스)을 구성할 수 있습니다.



참고

자동 업그레이드를 제한하려면 CR에서 결합된 `spec.deploymentPlan.image` 및 `spec.deploymentPlan.initImage` 속성 또는 `spec.version` 특성 중 하나를 지정해야 합니다.

자세한 내용은 [OpenShift에 AMQ Broker를 배포할 때 자동 업그레이드 제한](#)에서 참조하십시오.

확장 상태 보고

OpenShift Container Platform의 AMQ Broker 7.11에서 기본 브로커 CR의 Operator에서 보고한 상태 정보는 다음과 같이 확장됩니다.

- CR 콘텐츠의 유효성입니다.
- `brokerProperties` 속성에 구성된 속성의 애플리케이션입니다.
- 브로커 Pod에 시크릿의 JAAS(Java Authentication and Authorization Service) 로그인 모듈 파일을 전파합니다.
- 배포된 브로커의 버전과 해당 버전의 브로커 및 init 컨테이너 이미지의 URL입니다.
- 배포에 메이저, 마이너, 패치 및 보안 업그레이드를 적용하는 기능

동기 미러링 지원

7.11부터 브로커 간 동기 미러링을 구성하여 메시지가 미러의 두 브로커 볼륨에 동시에 기록되도록 할 수 있습니다. 동기 미러링을 사용하여 재해 복구를 위해 미러링된 브로커가 최신 버전인지 확인합니다. 자세한 내용은 [AMQ Broker 구성에서 브로커 연결](#) 구성을 참조하십시오.

Pod 중단 예산

OpenShift Container Platform의 AMQ Broker 7.11에서는 유지 관리 기간과 같이 자발적으로 중단되는 동안 동시에 사용할 수 있어야 하는 클러스터에서 최소 Pod 수를 지정하도록 Pod 중단 예산을 구성할 수 있습니다. 자세한 내용은 [OpenShift에 AMQ Broker를 배포 할 때 Pod 중단 예산 구성을 참조하십시오](#).

brokerProperties 구성은 변경 가능한 시크릿에 저장됩니다.

OpenShift Container Platform의 AMQ Broker 7.11에서 CR에 brokerProperties 특성을 사용하여 생성된 구성은 변경 가능한 시크릿에 저장됩니다. 변경 가능한 시크릿은 브로커를 다시 시작할 필요 없이 업데이트할 수 있습니다. 따라서 브로커가 구성을 정기적으로 다시 로드할 때 특히 브로커를 다시 시작해야 하는 업데이트 외에 구성 업데이트가 적용됩니다.

포함된 웹 서버를 제어하는 작업

7.11부터는 ActiveMQServerControl>-<에서 stopEmbeddedWebServer, startEmbeddedWebServer 및 restartEmbeddedWebServer 작업을 사용하여 AMQ Broker에 포함된 웹 서버를 중지하고 다시 시작할 수 있습니다. 이러한 작업을 사용하면 AMQ Broker의 SSL 인증서를 갱신하는 경우 AMQ Broker를 다시 시작하는 것을 방지할 수 있습니다.

AMQ Management Console에 로그인하는 데 사용되는 인증 정보는 메시지를 보내는 데 사용됩니다.

이전 버전의 AMQ Broker에서는 AMQ Management Console에서 메시지를 보내려면 AMQ Management Console에 사용자 이름과 암호를 지정해야 했습니다. 7.11부터 AMQ 관리 콘솔에 로그인하는 데 사용하는 인증 정보를 사용하여 메시지가 전송됩니다.

기본 동작을 재정의하고 다른 인증 정보를 지정하여 개별 메시지를 보낼 수 있습니다. 자세한 내용은 [Managing AMQ Broker](#)에서 [주소로 메시지 보내기](#)에서 참조하십시오.

OpenShift Container Platform의 AMQ Broker는 Prometheus에 대한 지표 데이터를 수집하도록 사전 구성되어 있습니다.

OpenShift Container Platform의 AMQ Broker 7.11에서 AMQ Broker 컨테이너 Pod는 Prometheus Operator 서비스 모니터가 지표 데이터를 수집할 수 있도록 사전 구성됩니다. 이전 릴리스에서는 지표 데이터를 수집하기 위해 서비스 모니터에 필요한 포트를 노출해야 했습니다.

브로커 컨테이너에 대한 환경 변수 설정

OpenShift Container Platform의 AMQ Broker 7.11에서는 각 브로커 컨테이너에 전달되는 사용자 정의 리소스(CR)에서 환경 변수를 설정할 수 있습니다. 예를 들어 TZ 환경 변수를 추가하여 브로커 컨테이너의 시간대를 설정할 수 있습니다. 자세한 내용은 [OpenShift에 AMQ Broker를 배포 할 때 브로커 컨테이너의 환경 변수 설정을 참조하십시오](#).

OpenShift Container Platform에서 프록시 전달 지원

OpenShift Container Platform의 AMQ Broker 7.11에서 AMQ Management Console을 호스팅하는 임베디드 웹 서버는 X-Forwarded 헤더를 처리하도록 사전 구성됩니다. X-Forwarded 헤더를 처리하여 AMQ Management Console은 프록시가 요청 경로에 있을 때 변경되거나 손실되는 헤더 정보를 수신할 수 있습니다. 예를 들어 AMQ Management Console은 HTTP를 사용하지만 프록시인 OpenShift Container Platform 라우터는 라우터에서 종료되는 HTTPS 경로를 사용하여 AMQ

Management Console을 노출합니다. **X-Forwarded** 헤더에서 **AMQ Management Console**은 브라우저와 라우터 간 연결을 사용하여 **HTTPS**로 전환하고 **HTTPS**로 전환하여 브라우저 요청을 제공할 수 있습니다.

일부 재전송 속성은 **brokerProperties CR** 특성에서만 지원됩니다.

7.8.x 또는 7.9.x 배포에 **spec.deploymentPlan.address>-<.addressSetting CR** 요소에 구성된 **redeliveryDelayMultiplier** 또는 **redeliveryCollisionAvoidanceFactor** 속성이 있는 경우 7.11. x로 업그레이드한 후 **brokerProperties CR** 속성에 이러한 속성을 구성해야 합니다. 이는 두 속성의 데이터 유형이 7.10.0에서 문자열로 변경되었기 때문에 필요합니다. 결과적으로 이러한 속성은 **spec.deploymentPlan.address>-<.addressSetting** 특성에서 더 이상 지원되지 않습니다.

다음 예제에서는 **brokerProperties** 요소에서 두 속성을 구성하는 방법을 보여줍니다.

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

참고

brokerProperties 속성에서 **spec.deploymentPlan.address>-<.addressSetting** 요소에서 사용된 **redeliveryDelayMultiplier** 특성 이름 대신 **redeliveryMultiplier** 특성 이름을 사용합니다.

XML 외부 엔티티(XXE) 처리 비활성화

broker.xml 파일에 포함된 별도의 파일에 브로커 구성을 모듈화할 필요가 없는 경우 **XXE** 처리를 비활성화하여 **XXE** 보안 취약점으로부터 보호할 수 있습니다. 가능한 경우 **XXE** 처리를 비활성화할 것을 권장합니다. 자세한 내용은 **AMQ Broker** 구성에서 **XXE(외부 XML 엔티티) 처리 비활성화** 를 참조하십시오.

JGroups 5.x

7.10.0 이전의 **AMQ Broker** 버전은 **CloudEvent 3.x**를 사용했습니다. **AMQ Broker 7.11**은 **CloudEvent 3.x**와 호환되지 않는 **CloudEvent 5.x**를 사용합니다. 일부 프로토콜 및 프로토콜 속성이 두 가지 **JGroup** 버전 간에 변경되어 **AMQ Broker 7.11**로 업그레이드할 때 **CloudEvent** 스택 구성을 변경해야 할 수 있습니다.

특정 디렉터리 이름으로 추출된 **AMQ Broker** 아카이브의 콘텐츠

Red Hat Enterprise Linux에서 **AMQ Broker** 아카이브를 추출하면 아카이브 내용이 현재 디렉터리의 **apache-artemis-2.28.0.redhat-00022** 디렉터리에 추출됩니다.

Operator 채널

AMQ Broker Operator, Red Hat Integration - AMQ Broker for RHEL 8 (Multiarch) 은 다음 채널에서 사용할 수 있습니다.

- 7.11.x - 이 채널은 버전 7.11에만 업데이트를 제공하며 장기 지원(LTS) 채널입니다.
- 7.10.x - 이 채널은 버전 7.10에 대해서만 업데이트를 제공하며 장기 지원(LTS) 채널입니다.



참고

채널을 전환하여 **Operator**를 업그레이드할 수 없습니다. 기존 **Operator**를 설치 제거하고 적절한 채널에서 새 버전의 **Operator**를 설치해야 합니다.

선택할 **Operator**를 확인하려면 [Red Hat Enterprise Linux Container Compatibility Matrix](#) 를 참조하십시오.

Prometheus 지표 플러그인의 클래스 이름으로 변경

AMQ Broker 7.11에서 AMQ Broker에 포함된 Prometheus 지표 플러그인의 클래스 이름이 `org.apache.activemq.artemis.core.core.metrics.plugins.ArtemisPrometheusMetricsPlugin` 에서 `com.redhat.amq.broker.core.plugins.ArtemisPrometheusMetric slugin` 에서 변경되었습니다. 이전 버전의 AMQ Broker에서 Prometheus 메트릭 플러그인이 활성화된 경우 AMQ Broker 7.11로 업그레이드할 때 `broker.xml` 구성 파일에서 클래스 이름을 업데이트해야 합니다. 자세한 내용은 **AMQ Broker 관리** 의 7.10.x에서 7.11.x로 브로커 인스턴스 업그레이드를 참조하십시오.

listProducers API 메서드 및 listProducersInfoAsJSONay 메서드에서 반환된 데이터 변경

AMQ Broker 7.11에서는 AMQ 관리 콘솔에서 사용하는 `listProducers` 메서드에서 데이터를 반환하는 방법에 대한 향상된 기능입니다.

- 이전 버전에서 반환된 데이터에서 생산자가 세션당 표시되었습니다. 따라서 생성자당 두 세션을 생성하는 Core 프로토콜을 사용하여 생산자를 생성한 경우 두 세션마다 별도의 생산자가 표시됩니다. 또한 생산자에서 메시지를 보내지 않고 생산자를 생성한 경우 프로듀서에 대해 반환된 주소가 비어 있습니다. AMQ Broker 7.11에서 `listProducers` 메서드는 코어 프로토콜에 의해 생성된 두 세션에 대해 단일 생산자를 반환합니다. 또한, 메시지를 보내기 전에 `address` 열에 올바른 주소를 표시합니다.
- 이전에는 익명 생산자를 사용하여 Core 또는 AMQP 프로토콜을 사용하여 여러 주소로 메시지를 보낼 때 각 주소에 대해 생산자가 표시되었습니다. 또한 표시된 주소는 생산자가 메시지를 보낸 첫 번째 주소이며, 이로 인해 단일 큐로 보내는 일반 생산자처럼 표시되었습니다. AMQ Broker

7.11에서 익명 생산자를 사용하여 여러 주소로 메시지를 보내면 각 익명 생산자에 대해 단일 생산자가 표시됩니다. 또한 주소는 특정 주소에 연결되지 않으며 **ANONYMOUS** 의 값을 갖습니다.

이전에는 **listProducersInfoAsJSON** 메서드에서 특정 세션별로 각 큐에 전송된 메시지 수를 제공했습니다. 그러나 이 방법은 메시지를 보낸 각 큐에 대해 생산자를 잘못 반환했습니다. 예를 들어 익명 생산자가 1000개 대기열로 메시지를 보낸 경우 이 방법은 1000명의 생산자를 반환했습니다. **AMQ Broker 7.11**에서 **listProducersInfoAsJSON** 메서드에서 **listProducers** 메서드와 동일한 데이터를 다른 형식으로 반환합니다.

AMQ Broker 7.11에서는 다음과 같은 새 지표 데이터가 반환됩니다.

Consumer

MessageInTransit - 아직 승인되지 않은 전송 메시지 수

messagesInTransitSize - 아직 승인되지 않은 전송의 총 메시지 크기

messagesDelivered - 전송된 메시지 수

messagesDeliveredSize - 전송된 총 메시지 크기

messageAcknowledged - 확인되는 총 메시지 수

messagesAckldgedAwaitingCommit - 확인되었지만 커밋 대기 중인 트랜잭션의 총 메시지 수

lastDeliveredTime - 마지막 메시지의 시간(밀리초)

lastAcknowledgedTime - 마지막 메시지의 밀리초 단위 시간

생산자

msgSent - 생산자가 보낸 메시지 수

msgSizeSent - 생산자가 보낸 메시지의 총 크기

lastProducedMessageID - 전송된 마지막 메시지의 ID

4장. 더 이상 사용되지 않는 기능

이 섹션에서는 지원되지만 **AMQ Broker**에서 더 이상 사용되지 않는 기능에 대해 설명합니다.

사용자 정의 리소스의 업그레이드 속성

7.11부터 업그레이드 속성 및 관련 활성화 및 마이너 속성은 원래 설계된 대로 작동할 수 없기 때문에 더 이상 사용되지 않습니다. 이미지 또는 버전 속성을 사용하여 특정 브로커 컨테이너 이미지를 배포합니다.

대기열 구성 요소

7.10부터 <queues> 구성 요소는 더 이상 사용되지 않습니다. <addresses> 구성 요소를 사용하여 주소 및 관련 대기열을 생성할 수 있습니다. <queues> 구성 요소는 향후 릴리스에서 제거됩니다.

getAddresses>-< 방법

7.10부터 org.apache.activemq.artemis.core.config.Configuration 인터페이스에 포함된 getAddressesSettings 방법이 더 이상 사용되지 않습니다. getAddressSettings 방법을 사용하여 브로커의 주소 및 대기열을 프로그래밍 방식으로 구성합니다.

Open>-<re 프로토콜

7.9부터 시작으로 Open>-<re 프로토콜은 더 이상 사용되지 않는 기능입니다. 새로운 AMQ Broker 기반 시스템을 생성하는 경우 다른 지원되는 프로토콜 중 하나를 사용합니다. 8.0 릴리스에서는 Openwire 프로토콜이 AMQ Broker에서 제거됩니다.

브로커 인스턴스가 실행되지 않을 때 사용자 추가

7.8부터 AMQ Broker 인스턴스가 실행되지 않으면 CLI 인터페이스에서 브로커에 사용자를 추가하는 기능이 제거됩니다.

네트워크 pinger

7.5부터 네트워크 ping은 더 이상 사용되지 않는 기능입니다. 네트워크 ping은 복구 불가능한 메시지 손실을 초래할 수 있는 네트워크 격리 문제로부터 브로커 클러스터를 보호할 수 없습니다. 이 기능은 향후 릴리스에서 제거될 예정입니다. Red Hat은 네트워크 ping을 사용하는 기존 AMQ Broker 배포를 계속 지원합니다. 그러나 Red Hat은 더 이상 새 배포에서 네트워크 ping을 사용할 것을 권장하지 않습니다. 고가용성을 위해 브로커 클러스터를 구성하고 네트워크 격리 문제를 방지하는 방법에 대한 지침은 [AMQ Broker 구성의 고가용성](#)을 참조하십시오.

Hawtio 디스패치 콘솔 플러그인

7.3부터 AMQ Broker는 더 이상 Hawtio 디스패치 콘솔 플러그인 dispatch-hawtio-console.war와 함께 제공되지 않습니다. 이전에는 디스패치 콘솔을 사용하여 AMQ Interconnect를 관리했습니다. 그러나 AMQ Interconnect는 이제 자체 독립 실행형 웹 콘솔을 사용합니다.

5장. 삭제된 기능

이 섹션에서는 **AMQ Broker**에서 제거된 기능 및 기능에 대해 설명합니다.

AMQ Broker 웹 서버의 루트에 액세스

이전 버전의 **AMQ Broker**에서 **AMQ Broker** 웹 서버의 루트 URL을 엽니다(예: <http://localhost:8161/>) 브라우저 창에서 시작 페이지가 표시됩니다. 시작 페이지에는 **AMQ Management Console** 및 **AMQ Broker** 설명서에 대한 링크가 있습니다. 7.11에서는 **AMQ Broker**에서 모든 정적 **HTML** 콘텐츠가 제거됩니다. 따라서 **AMQ Broker** 웹 서버의 루트 URL을 열면 중단 페이지가 표시되지 않습니다. 대신 브라우저 세션이 **AMQ** 관리 콘솔로 자동으로 리디렉션됩니다.

6장. 해결된 문제

릴리스에서 수정된 전체 문제 목록은 [AMQ Broker 7.11.0 해결 문제](#) 및 [AMQ Broker - 7.11.x 해결된 문제](#) 목록은 패치 릴리스에서 수정된 문제 목록을 참조하십시오.

7장. 수정된 COMMON VULNERABILITIES AND EXPOSURES

이 섹션에서는 **AMQ Broker 7.11** 릴리스에서 수정된 **CVE(Common Vulnerabilities and Exposures)**에 대해 자세히 설명합니다.

- [ENTMQBR-6630](#) - CVE-2022-1278 WildFly: 가능한 정보 공개
- [ENTMQBR-7397](#) - CVE-2022-22970 Springframework: 다중 부분 또는 서블릿 부분에 데이터 바인딩을 통한 DoS
- [ENTMQBR-7398](#) - CVE-2022-22971 Springframework: WebSocket을 통해 STOMP 포함
- [ENTMQBR-7005](#) - CVE-2022-2047 jetty-http: 개선된 호스트 이름 입력 처리
- [ENTMQBR-7640](#) - CVE-2022-3782 keycloak: 경로 트래버스: 이중 URL 인코딩

8장. 확인된 문제

이 섹션에서는 **AMQ Broker 7.11**에서 알려진 문제에 대해 설명합니다.

- **ENTMQBR-8106 - AMQ Broker Drainer Pod가 CR에서 MessageMigration을 변경한 후 제대로 작동하지 않음**

실행 중인 브로커 배포에서 `messageMigration` 속성의 값을 변경할 수 없습니다. 이 문제를 해결하려면 새 **ActiveMQ Artemis CR**에서 `messageMigration` 속성에 필요한 값을 설정하고 새 브로커 배포를 생성해야 합니다.
- **ENTMQBR-8166 - UseClientAuth=true가 있는 자체 서명된 인증서로 인해 Jolokia와 Operator의 통신을 방지합니다.**

ActiveMQ Artemis CR의 `console` 섹션에서 `useClientAuth` 속성이 `true` 로 설정된 경우 **Operator**는 브로커에서 특정 기능을 구성할 수 없습니다. **Operator** 로그에 원격 오류: `tls: bad` 인증서로 끝나는 오류 메시지가 표시됩니다.
- **ENTMQBR-7385 - 느린 소비자의 페더링 큐 주변의 메시지 플로핑**

로컬 애플리케이션 사용자가 매우 느리거나 메시지를 사용할 수 없는 경우 페더레이션 연결을 통해 메시지를 백과 사용하여 애플리케이션 소비자가 마지막으로 사용하기 전에 여러 번 보낼 수 있습니다.
- **ENTMQBR-7820 - [Operator] 7.11.0 OPR1 Operator 로그에 나열된 지원 버전이 올바르지 않음**

Operator 로그는 다음 **AMQ Broker** 이미지 버전 : `7.10.00.1 7.10.2 7.11.0 7.8.1 7.8.2 7.8.3 7.9.0 7.9.2 7.9.3 7.9.3 7.9.4`에 대한 지원이 나열됩니다. **Operator**는 실제로 **7.10.0**으로 시작하는 **AMQ Broker** 이미지 버전을 지원합니다.
- **ENTMQBR-7359 - 7.10.0 Operator로 인증 정보 시크릿의 현재 처리로 변경**

Operator는 시크릿의 브로커에 연결하기 위해 관리자 사용자 이름 및 암호를 저장합니다. 기본 시크릿 이름은 `<custom-resource-name>-credentials-secret` 형식으로 되어 있습니다. 시크릿을 수동으로 생성하거나 **Operator**에서 보안을 생성하도록 허용할 수 있습니다.

`adminUser` 및 `adminPassword` 속성이 **7.10.0** 이전의 사용자 정의 리소스에 구성된 경우

Operator는 이러한 속성의 값을 사용하여 수동으로 생성된 시크릿을 업데이트합니다. **7.10.0**부터 **Operator**는 더 이상 수동으로 생성된 시크릿을 업데이트하지 않습니다. 따라서 **CR**에서 **adminUser** 및 **adminPassword** 속성 값을 변경하는 경우 다음 중 하나를 수행해야 합니다.

- 새 사용자 이름 및 암호로 시크릿을 업데이트
- 보안을 삭제하고 **Operator**에서 보안을 생성하도록 허용합니다. **Operator**에서 시크릿을 생성할 때 **CR**에 지정된 경우 **adminUser** 및 **adminPassword** 속성 값을 추가합니다. 이러한 속성이 **CR**에 없는 경우 **Operator**는 보안에 대한 임의의 인증 정보를 생성합니다.
- **ENTMQBR-7111** - 7.10 버전의 **Operator**는 업그레이드하는 동안 **StatefulSet**을 제거하는 경향이 있습니다.

AMQ Broker Operator 7.10.0으로 업그레이드 중인 경우 새 **Operator**는 조정 프로세스 중 각 배포에 대한 기존 **StatefulSet**을 자동으로 삭제합니다. **Operator**가 **StatefulSet**을 삭제하면 기존 브로커 **Pod**가 삭제되어 일시적인 브로커 중단이 발생합니다.

다음 명령을 실행하여 이 문제를 해결하기 위해 **Operator**에서 **StatefulSet: oc delete statefulset < statefulset-name > --cascade=orphan**을 삭제하기 전에 **StatefulSet**을 수동으로 삭제하고 실행 중인 **Pod**를 분리할 수 있습니다.

업그레이드 프로세스 중에 **StatefulSet**을 수동으로 삭제하면 새 **Operator**에서 실행 중인 **Pod**를 삭제하지 않고 **StatefulSet**을 조정할 수 있습니다. 자세한 내용은 *OpenShift에 AMQ Broker를 배포할 때 OperatorHub를 사용하여 Operator 업그레이드를 참조하십시오.*

- **ENTMQBR-6473** - 스키마 URL 변경으로 인한 호환되지 않는 구성

버전 **7.9** 또는 **7.10** 인스턴스가 있는 이전 릴리스의 브로커 인스턴스 구성을 사용하려는 경우 스키마 URL 변경으로 인한 호환되지 않는 구성이 있으면 브로커가 충돌합니다. 이 문제를 해결하려면 **Linux**의 **7.9.0**에서 **7.10.0**으로 요약된 대로 관련 구성 파일의 스키마 URL을 업데이트합니다.

- **ENTMQBR-4813** 대규모 메시지 및 여러 **C++** 구독자가 포함된 **ENTMQBR-48Exception**

AMQP 프로토콜을 사용하는 여러 **C++ publisher** 클라이언트가 구독자 및 브로커와 동일한 호스트에서 실행되고 있고 게시자가 대규모 메시지를 전송하는 경우 구독자 중 하나가 충돌합니다.

- **ENTMQBR-5749 - OperatorHub에 표시되는 지원되지 않는 Operator 제거**

OperatorHub에서 **Operator** 배포에 언급된 **Operator** 및 **Operator** 채널만 지원됩니다. **Operator** 게시와 관련된 기술적 이유로 **OperatorHub**에 다른 **Operator** 및 채널이 표시되고 무시해야 합니다. 다음 목록은 참조를 위해 표시되는 **Operator**는 표시되지만 지원되지 않는음을 보여줍니다.

 - Red Hat Integration - AMQ Broker LTS - 모든 채널
 - Red Hat Integration - AMQ Broker - alpha, current, current-76
- **ENTMQBR-569 - OpenJDK에서 AMQP로 ID를 변환하면 ID를 바이너리로 보냅니다.**

A-MQ 6 OpenJDK 클라이언트에서 **AMQP** 클라이언트로 교차 프로토콜을 통신할 때 추가 정보가 애플리케이션 메시지 속성에 인코딩됩니다. 이는 브로커가 내부적으로 사용하는 일반 정보이며 무시할 수 있습니다.
- **ENTMQBR-655 - [AMQP] fill-validated-user 가 활성화된 경우 메시지를 전송할 수 없음**

AMQP 프로토콜을 사용하여 생성된 메시지에서는 설정 옵션 **populate-validated-user** 가 지원되지 않습니다.
- **ENTMQBR-1875 - [AMQ 7, ha, 복제된 저장소] 백업 브로커는 이동하지 않고 "live" 또는 종료되지 않음 - ActiveMQIllegalStateException errorType=ILLEGAL_STATE message=AMQ119026: Backup Server가 라이브와 동기화되지 않았습니다.**

백업 브로커가 기본 브로커와 동기화하는 동안 기본 브로커의 페이징 디스크를 제거하면 주 서버가 실패합니다. 또한 백업 브로커는 기본 브로커와 계속 동기화되기 때문에 실시간 상태가 될 수 없습니다.
- **ENTMQBR-2068 - 일부 메시지가 수신되었지만 HA 장애 조치, 장애 복구 시나리오 중에 전달되지 않음**

현재 **OpenWire** 클라이언트가 메시지를 보내는 동안 브로커가 백업으로 장애 조치되면 장애 조치가 발생할 때 브로커로 전달되는 메시지가 손실될 수 있습니다. 이 문제를 해결하려면 브로커가 메시지를 승인하기 전에 메시지를 지속해야 합니다.

- ENTMQBR-3331 - Stateful set** 컨트롤러가 **CreateContainerError**에서 복구할 수 없으므로 **Operator**를 차단합니다.

AMQ Broker Operator가 구성 오류가 있는 **CR**(사용자 정의 리소스)에서 상태 저장 세트를 생성하는 경우 오류가 해결되면 상태 저장 세트 컨트롤러에서 업데이트된 상태 저장 세트를 롤아웃할 수 없습니다.

예를 들어 기본 브로커 **CR**의 **image** 특성 값이 잘못 입력되면 상태 저장 세트 컨트롤러에서 생성한 첫 번째 **Pod**의 상태가 보류 중 상태로 유지됩니다. 그런 다음 맞춤 오류를 수정하고 **CR** 변경 사항을 적용하면 **AMQ Broker Operator**에서 상태 저장 세트를 업데이트합니다. 그러나 **Kubernetes** 알려진 문제로 인해 상태 저장 세트 컨트롤러에서 업데이트된 상태 저장 세트를 롤아웃하지 못합니다. 컨트롤러는 보류 중 상태의 **Pod**가 준비 상태가 될 때까지 무기한 대기하므로 새 **Pod**가 배포되지 않습니다.

이 문제를 해결하려면 상태 저장 세트 컨트롤러에서 새 **Pod**를 배포할 수 있도록 **Pending** 상태인 **Pod**를 삭제해야 합니다. 보류 중 상태의 **Pod**를 확인하려면 **oc get pods --field-selector=status.phase=Pending** 명령을 사용합니다. **Pod**를 삭제하려면 **oc delete pod <pod name>** 명령을 사용합니다.
- ENTMQBR-3846 - MQTT** 클라이언트가 브로커 재시작 시 다시 연결하지 않음

브로커를 다시 시작하거나 브로커가 장애 조치되면 활성 브로커가 이전에 연결된 **MQTT** 클라이언트의 연결을 복원하지 않습니다. 이 문제를 해결하려면 **MQTT** 클라이언트를 다시 연결하려면 클라이언트에서 수동으로 **subscribe()** 메서드를 호출해야 합니다.
- ENTMQBR-4127 - AMQ Broker Operator: OpenShift**에서 생성한 경로 이름이 너무 길 수 있습니다.

Operator 기반 배포의 각 브로커 **Pod**에 대해 **Operator**에서 **AMQ Broker** 관리 콘솔에 액세스하기 위해 생성하는 경로의 기본 이름에는 사용자 정의 리소스(**CR**) 인스턴스 이름, **OpenShift** 프로젝트의 이름, **OpenShift** 클러스터 이름이 포함됩니다. 예를 들면 **my-broker-deployment-wconsj-0-svc-rte-my-openshift-project.my-openshift-domain** 입니다. 이러한 이름 중 일부가 긴 경우 기본 경로 이름은 **OpenShift**에서 적용하는 63자 제한을 초과할 수 있습니다. 이 경우 **OpenShift Container Platform** 웹 콘솔에서 경로는 **Rejected**.

이 문제를 해결하려면 **OpenShift Container Platform** 웹 콘솔을 사용하여 경로 이름을 수동으로 편집합니다. 콘솔에서 경로를 클릭합니다. 오른쪽 상단의 작업 드롭다운 메뉴에서 경로 편집을 선택합니다. **YAML** 편집기에서 **spec.host** 속성을 찾아 값을 편집합니다.
- ENTMQBR-4140 - AMQ Broker Operator: storage.size** 가 부적절하게 지정된 경우 설치를

사용할 수 없습니다.

영구 스토리지 배포의 브로커에 필요한 PVC(영구 볼륨 클레임) 크기를 지정하도록 CR(사용자 정의 리소스) 인스턴스의 **storage.size** 속성을 구성하는 경우 이 값을 올바르게 지정하지 않으면 **Operator** 설치를 사용할 수 없게 됩니다. 예를 들어 **storage.size**의 값을 1로 설정한다고 가정합니다(즉, 단위를 지정하지 않고). 이 경우 **Operator**는 CR을 사용하여 브로커 배포를 생성할 수 없습니다. 또한 CR을 제거하고 **storage.size**가 올바르게 지정된 새 버전을 배포하더라도 **Operator**는 여전히 이 CR을 사용하여 예상대로 배포를 생성할 수 없습니다.

이 문제를 해결하려면 먼저 **Operator**를 중지합니다. **OpenShift Container Platform** 웹 콘솔에서 배포를 클릭합니다. **AMQ Broker Operator**에 해당하는 **Pod**의 추가 옵션 메뉴(세 개의 수직 점)를 클릭합니다. **Pod** 개수 편집을 클릭하고 값을 0으로 설정합니다. **Operator Pod**가 중지되면 **storage.size**가 올바르게 지정된 새 버전의 CR을 생성합니다. 그런 다음 **Operator**를 다시 시작하려면 **Pod** 개수 편집을 다시 클릭하고 값을 다시 1로 설정합니다.

- ENTMQBR-4141 - AMQ Broker Operator:** 영구 볼륨 크기를 늘리려면 상태 저장 세트를 다시 생성한 후에도 수동 개입이 필요합니다.

영구 스토리지 배포에서 브로커가 요구하는 PVC(영구 볼륨 클레임) 크기를 늘리려는 경우 추가 수동 단계없이 변경 사항이 적용되지 않습니다. 예를 들어 사용자 정의 리소스(CR) 인스턴스의 **storage.size** 속성을 구성하여 PVC의 초기 크기를 지정한다고 가정합니다. 다른 값을 **storage.size**로 지정하도록 CR을 수정하면 기존 브로커는 원래 PVC 크기를 계속 사용합니다. 배포를 0 브로커로 축소한 다음 원래 번호로 백업하는 경우에도 마찬가지입니다. 그러나 추가 브로커를 추가하기 위해 배포 크기를 확장하면 새 브로커는 새 PVC 크기를 사용합니다.

이 문제를 해결하고 배포의 모든 브로커가 동일한 PVC 크기를 사용하는지 확인하려면 **OpenShift Container Platform** 웹 콘솔을 사용하여 배포에 사용되는 PVC 크기를 확장합니다. 콘솔에서 스토리지 → 영구 볼륨 클레임을 클릭합니다. 배포를 클릭합니다. 오른쪽 상단의 작업 드롭다운 메뉴에서 PVC 확장을 선택하고 새 값을 입력합니다.

9장. 중요한 링크

- [Red Hat AMQ Broker 7.10 릴리스 노트](#)
- [Red Hat AMQ Broker 7.9 릴리스 노트](#)
- [Red Hat AMQ Broker 7.8 릴리스 노트](#)
- [Red Hat AMQ Broker 7.7 릴리스 노트](#)
- [Red Hat AMQ Broker 7.6 릴리스 노트](#)
- [Red Hat AMQ Broker 7.1 - 7.5 릴리스 노트](#)
- [Red Hat AMQ 7 지원 설정](#)
- [Red Hat AMQ 7 구성 요소 세부 정보](#)

2024-10-18에 최종 업데이트된 문서