

9.2

컨테이너의 *IBM MQ*

IBM

참고

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, [151 페이지의 『주의사항』](#)에 있는 정보를 확인하십시오.

이 개정판은 새 개정판에 별도로 명시하지 않는 한, IBM® MQ의 버전 9 릴리스 2 및 모든 후속 릴리스와 수정에 적용됩니다.

IBM은 귀하가 IBM으로 보낸 정보를 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 사용하거나 배포할 수 있습니다.

© Copyright International Business Machines Corporation 2007년, 2024.

목차

컨테이너의 IBM MQ 및 IBM Cloud Pak for Integration.....	5
컨테이너에 IBM MQ 계획.....	5
컨테이너에서 IBM MQ를 사용하는 방법 선택.....	5
IBM MQ Operator에 대한 지원.....	6
IBM MQ Operator에 대한 종속 항목.....	9
IBM MQ Operator에서 필요로 하는 클러스터 범위 권한.....	10
IBM MQ Operator의 스토리지 고려사항.....	10
자체 IBM MQ 큐 관리자 컨테이너 이미지 빌드 지원.....	12
컨테이너에서 IBM MQ에 대한 고가용성.....	15
컨테이너에서 IBM MQ의 재해 복구.....	17
컨테이너에서 IBM MQ에 대한 사용자 인증 및 권한 부여.....	17
컨테이너에서 IBM MQ에 대한 확장성 및 성능 계획.....	18
IBM Cloud Pak for Integration 및 Red Hat OpenShift에서 IBM MQ 사용.....	18
IBM MQ Operator의 릴리스 히스토리.....	19
IBM MQ를 IBM Cloud Pak for Integration으로 마이그레이션.....	35
Red Hat OpenShift에 IBM MQ Operator 설치 및 설치 제거.....	56
IBM MQ Operator 및 큐 관리자 업그레이드.....	68
IBM MQ Operator를 사용하여 큐 관리자 배치 및 구성.....	75
IBM MQ Operator를 사용하여 IBM MQ 운영.....	108
IBM MQ Operator의 문제점 해결.....	118
IBM MQ Operator에 대한 API 참조.....	119
사용자 고유의 IBM MQ 컨테이너 및 배치 코드 빌드.....	139
컨테이너를 사용하여 IBM MQ 큐 관리자 이미지 계획.....	139
샘플 IBM MQ 큐 관리자 컨테이너 이미지 빌드.....	140
별도의 컨테이너에서 로컬 바인딩 애플리케이션 실행.....	142
자체 컨테이너 작성 시 원시 HA 그룹 작성.....	144
주의사항.....	151
프로그래밍 인터페이스 정보.....	152
상표.....	152

컨테이너를 통해 모든 종속 항목과 함께 IBM MQ 큐 관리자 또는 IBM MQ 클라이언트 애플리케이션을 표준화된 소프트웨어 개발 단위로 패키징할 수 있습니다.

Red Hat® OpenShift®에서 IBM MQ Operator 를 사용하여 IBM MQ 를 실행할 수 있습니다. IBM Cloud Pak® for Integration, IBM MQ Advanced 또는 IBM MQ Advanced for Developers를 사용하여 이를 수행할 수 있습니다. 직접 빌드한 컨테이너에서도 IBM MQ를 실행할 수 있습니다.

MQ Adv.

CD

IBM MQ Operator에 대한 자세한 정보는 다음 링크를 참조하십시오.

컨테이너에서 IBM MQ를 계획할 때, IBM MQ에서 다양한 아키텍처 옵션에 제공하는 지원을 고려하십시오(예: 고가용성이 관리되는 방법 및 큐 관리자를 보호할 방법).

이 태스크 정보

컨테이너 아키텍처에서 IBM MQ를 계획하기 전에 기본 IBM MQ 개념([IBM MQ 기술 개요 참조](#)) 및 기본 Kubernetes/Red Hat OpenShift 개념([Red Hat OpenShift Container Platform 아키텍처 참조](#))에 익숙해져야 합니다.

프로시저

- [5 페이지의 『컨테이너에서 IBM MQ를 사용하는 방법 선택』](#).
- [6 페이지의 『IBM MQ Operator에 대한 지원』](#).
- [12 페이지의 『자체 IBM MQ 큐 관리자 컨테이너 이미지 빌드 지원』](#).
- [10 페이지의 『IBM MQ Operator의 스토리지 고려사항』](#).
- [15 페이지의 『컨테이너에서 IBM MQ에 대한 고가용성』](#).
- [17 페이지의 『컨테이너에서 IBM MQ의 재해 복구』](#).
- [17 페이지의 『컨테이너에서 IBM MQ에 대한 사용자 인증 및 권한 부여』](#).

컨테이너에서 IBM MQ를 사용하는 방법 선택

컨테이너에서 IBM MQ를 사용하기 위한 여러 옵션이 있습니다. 패키징되기 이전의 컨테이너 이미지를 사용하는 IBM MQ Operator를 사용하도록 선택하거나 자체 이미지 및 배치 코드를 빌드할 수 있습니다.

IBM MQ Operator 사용

OpenShift

CP4I

Red Hat OpenShift Container Platform에 배치할 계획인 경우 IBM MQ Operator를 사용하려고 할 수도 있습니다.

IBM MQ Operator은(는) 새 QueueManager 사용자 정의 자원을 Red Hat OpenShift Container Platform에 추가합니다. 운영자는 새 큐 관리자 정의를 감시하고 필요한 하위 레벨 자원(예: StatefulSet 및 Service 자원)으로 전환합니다. 기본 HA의 경우, 운영자는 큐 관리자 인스턴스의 복잡한 롤링 업데이트를 수행할 수도 있습니다. [146 페이지의 『고유한 HA 큐 관리자의 롤링 업데이트를 수행하기 위한 고려사항』](#) 참조

일부 IBM MQ 기능은 IBM MQ Operator를 사용할 때 지원되지 않습니다. 다음을 수행하려는 경우 자체 이미지 및 차트를 빌드해야 합니다.

- 관리 또는 메시징에 REST API 사용
- 다음 MQ 컴포넌트를 사용하십시오.

- Managed File Transfer 에이전트 및 해당 자원. 그러나 IBM MQ Operator 를 사용하여 하나 이상의 조정, 명령 또는 에이전트 큐 관리자를 제공할 수 있습니다.
- AMQP
- IBM MQ Bridge to Salesforce
- IBM MQ Bridge to blockchain(컨테이너에서 지원되지 않음)
- IBM MQ Telemetry Transport (MQTT).
- **crtmqm, strmqm** 및 **endmqm**과 함께 사용되는 옵션을 사용자 정의합니다(예: 로그 파일 페이지 구성). 대부분의 옵션은 INI 파일을 사용하여 구성할 수 있습니다.

IBM MQ Operator 및 컨테이너는 빠르게 진화하고 있으므로 Long Term Support 릴리스에서는 지원되지 않습니다.

IBM MQ Operator에는 사전 빌드된 컨테이너 이미지와 Red Hat OpenShift Container Platform에서 실행하기 위한 배치 코드가 모두 포함됩니다. IBM MQ Operator는 제공된 IBM MQ 컨테이너 이미지 또는 그 위에 계층화된 컨테이너 이미지를 배치하는 데 사용할 수 있지만 사용자 정의 빌드된 MQ 컨테이너 이미지를 배치하는 데 사용할 수 없습니다.

자체 이미지 및 배치 코드 빌드

Multi

이 솔루션은 가장 유연한 컨테이너 솔루션이지만, 이 솔루션을 사용하려면 컨테이너를 구성하는 측면에서 뛰어난 기술을 보유하고 있어야 하며 결과 컨테이너를 "소유"하고 있어야 합니다. Red Hat OpenShift Container Platform 사용을 계획 중인 경우, 자체 이미지 및 배치 코드를 빌드해야 합니다.

자체 이미지를 빌드하기 위한 샘플을 사용할 수 있습니다. [139 페이지의 『사용자 고유의 IBM MQ 컨테이너 및 배치 코드 빌드』](#)의 내용을 참조하십시오.

관련 개념

[6 페이지의 『IBM MQ Operator에 대한 지원』](#)

IBM MQ Operator는 Red Hat OpenShift Container Platform에 배치된 경우에만 지원됩니다.

[12 페이지의 『자체 IBM MQ 큐 관리자 컨테이너 이미지 빌드 지원』](#)

IBM MQ는 GitHub에 IBM MQ 큐 관리자 컨테이너를 빌드하기 위한 코드를 제공합니다. 이는 IBM에서 자체 지원 컨테이너를 빌드하는 데 사용하는 프로세스를 기반으로 하며 이 GitHub 저장소를 사용하여 자체 컨테이너 이미지 빌드를 단순화하고 가속화할 수 있습니다.

OpenShift CP4I CD EUS IBM MQ Operator에 대한 지원

IBM MQ Operator는 Red Hat OpenShift Container Platform에 배치된 경우에만 지원됩니다.

EUS(Extended Update Support) 릴리스를 IBM MQ Operator에서 사용할 수 있지만 는 IBM MQ Continuous Delivery(CD) 릴리스를 기반으로 하는 이미지를 사용합니다 IBM Cloud Pak for Integration. CD 릴리스는 최대 1년 또는 두 번의 CD 릴리스(둘 중 더 긴 기간)동안 지원됩니다. IBM MQ의 Long Term Support 릴리스는 IBM MQ Operator을(를) 통해 사용할 수 없습니다. IBM Cloud Pak for Integration 2020.4.1은 -eus(으)로 표시된 IBM MQ 버전을 사용하는 경우 18개월동안 지원되는 EUS (Extended Update Support) 릴리스입니다. 그렇지 않으면, IBM MQ 9.2은(는) IBM MQ Operator과(와) 함께 Continuous Delivery 릴리스로 간주됩니다.

IBM MQ Operator 는 IBM MQ에서 사용하는 키 Linux® 라이브러리 및 유틸리티를 포함하는 Red Hat Universal Base Image (UBI) 에 IBM MQ 설치를 제공하는 컨테이너 이미지를 사용합니다. UBI는 Red Hat OpenShift에서 실행될 때 Red Hat 에서 지원됩니다.

IBM MQ Operator은(는) amd64 및 s390x(z/Linux) 아키텍처에서 지원됩니다.

관련 개념

[12 페이지의 『자체 IBM MQ 큐 관리자 컨테이너 이미지 빌드 지원』](#)

IBM MQ는 GitHub에 IBM MQ 큐 관리자 컨테이너를 빌드하기 위한 코드를 제공합니다. 이는 IBM에서 자체 지원 컨테이너를 빌드하는 데 사용하는 프로세스를 기반으로 하며 이 GitHub 저장소를 사용하여 자체 컨테이너 이미지 빌드를 단순화하고 가속화할 수 있습니다.

IBM MQ, Red Hat OpenShift Container Platform 및 IBM Cloud Pak for Integration의 지원되는 버전 사이의 맵핑입니다.

- 7 페이지의 『[사용 가능한 IBM MQ 버전](#)』
- 7 페이지의 『[호환 가능한 Red Hat OpenShift Container Platform 버전](#)』
- 8 페이지의 『[IBM Cloud Pak for Integration 버전](#)』
- 8 페이지의 『[이전 Operator에서 사용 가능한 IBM MQ 버전](#)』
- 8 페이지의 『[이전 Operator에 호환 가능한 Red Hat OpenShift Container Platform 버전](#)』

사용 가능한 IBM MQ 버전

Operator 채널	Operator 버전	IBM MQ 버전							
		9.1.5	9.2.0 CD	9.2.0 EUS	9.2.1	9.2.2	9.2.3	9.2.4	9.2.5
v1.6	1.6	⚠	⚠	→	⚠	●	●		
v1.7	1.7	⚠	⚠	→	⚠	●	●	●	
v1.8	1.8	⚠	⚠	→	⚠	⚠	●	●	●

키:

- Continuous Delivery 지원 사용 가능
- ☒ Extended Update Support 사용 가능
- Extended Update Support Operand에서 Continuous Delivery Operand로 마이그레이션하는 중에만 사용 가능합니다.
- ⚠ 더 이상 사용되지 않습니다. IBM MQ 릴리스 지원이 중단되면 Operator에서 계속 구성할 수 있지만 더 이상 지원을 받을 수 없고 향후 릴리스에서 제거될 수 있습니다.

각 버전에서의 상세 기능, 변경사항 및 수정사항을 포함하여 각 버전의 전체 세부사항에 대해서는 [19 페이지의 『IBM MQ Operator의 릴리스 히스토리』](#)의 내용을 참조하십시오.

호환 가능한 Red Hat OpenShift Container Platform 버전

Operator 채널	Operator 버전	Red Hat OpenShift Container Platform 버전 ¹				
		4.6	4.7 ²	4.8	4.9	4.10
v1.6	1.6	●	●	●	●	●
v1.7	1.7	●	●	●	●	●
v1.8	1.8	●	●	●	●	●

키:

¹ Red Hat OpenShift Container Platform 버전에는 자체 지원 날짜가 적용됩니다. 자세한 정보는 [Red Hat OpenShift Container Platform 라이프사이클 정책](#)을 참조하십시오.
² IBM MQ Operator은(는) IBM Cloud Pak foundational services에 종속됩니다. Red Hat OpenShift Container Platform 4.7을 사용하려면 먼저 IBM Cloud Pak foundational services 버전을 업그레이드해야 합니다.

- Continuous Delivery 지원 사용 가능
- ☒ Extended Update Support 사용 가능

IBM Cloud Pak for Integration 버전

IBM MQ Operator 1.8.x는 IBM Cloud Pak for Integration 버전 2021.4.1의 일부 또는 독립적으로 사용하도록 지원됩니다.

IBM MQ Operator 1.7.x는 IBM Cloud Pak for Integration 버전 2021.4.1의 일부로 또는 독립적으로 사용하도록 지원됩니다.

IBM MQ Operator 1.6.x는 IBM Cloud Pak for Integration 버전 2021.2.1, 2021.3.1의 일부로 또는 독립적으로 사용하도록 지원됩니다.

IBM MQ Operator 1.5.x 는 더 이상 지원되지 않습니다.

IBM MQ Operator 1.4.x 는 더 이상 지원되지 않습니다.

IBM MQ Operator 1.3.x 는 더 이상 지원되지 않습니다.

IBM MQ Operator 1.2.x 은 더 이상 지원되지 않습니다.

IBM MQ Operators 1.1.x 및 1.0.x 는 더 이상 지원되지 않습니다.

이전 Operator에서 사용 가능한 IBM MQ 버전

다음 표는 "수명 종료"에 도달한 IBM MQ Operator의 버전에 적용됩니다.

Operator 채널	Operator 버전	IBM MQ 버전							
		9.1.5	9.2.0 CD	9.2.0 EUS	9.2.1	9.2.2	9.2.3	9.2.4	9.2.5
v1.0	1.0	⚠							
v1.1	1.1	⚠	⚠						
v1.2	1.2	⚠	⚠						
v1.3-eus	1.3	⚠	⚠	⚠					
v1.4	1.4	⚠	⚠	→	⚠				
v1.5	1.5	⚠	⚠	→	⚠	⚠			

키:

→

Extended Update Support Operand에서 Continuous Delivery Operand로 마이그레이션하는 중에만 사용 가능합니다.

⚠

더 이상 사용되지 않습니다. IBM MQ 릴리스 지원이 중단되면 IBM MQ Operator에서 계속 구성할 수 있지만 더 이상 지원을 받을 수 없습니다.

각 버전에서의 상세 기능, 변경사항 및 수정사항을 포함하여 각 버전의 전체 세부사항에 대해서는 [19 페이지의 『IBM MQ Operator의 릴리스 히스토리』](#)의 내용을 참조하십시오.

이전 Operator에 호환 가능한 Red Hat OpenShift Container Platform 버전

다음 표는 "수명 종료"에 도달한 IBM MQ Operator의 버전에 적용됩니다.

Operator 채널	Operator 버전	Red Hat OpenShift Container Platform 버전 ³						
		4.4 ⁴	4.5 ⁵	4.6	4.7 ⁶	4.8	4.9	4.10
v1.0	1.0	⚠	⚠	⚠	⚠			
v1.1	1.1	⚠	⚠	⚠	⚠	⚠		
v1.2	1.2	⚠	⚠	⚠	⚠	⚠		
v1.3-eus	1.3			⚠	→	→	→	→
v1.4	1.4			⚠	⚠	⚠	⚠	
v1.5	1.5			⚠	⚠	⚠	⚠	⚠

키:

→

Extended Update Support Operand에서 Continuous Delivery Operand로 마이그레이션하는 중에만 사용 가능합니다.

⚠

IBM MQ Operator 버전이 "수명 종료"에 도달했지만 이전에 이 버전의 Red Hat OpenShift Container Platform에서 사용 가능했습니다.

OpenShift CP4I IBM MQ Operator에 대한 종속 항목

IBM MQ Operator는 IBM Cloud Pak foundational services Operator에 의존하고, IBM Operand Deployment Lifecycle Manager(ODLM) Operator도 설치합니다. 이러한 Operator는 IBM MQ Operator 설치 시 자동으로 설치됩니다. 이러한 종속 Operator는 CPU 및 메모리 풋프린트가 작고 일부 상황에서는 추가 자원을 배치하는 데 사용됩니다.

QueueManager를 작성할 때 IBM MQ Operator가 필요한 추가 서비스에 대한 OperandRequest를 작성합니다. OperandRequest는 ODLM Operator에서 이행되며 필요한 경우 필수 서비스를 설치하고 인스턴스화합니다. 필요한 서비스는 큐 관리자를 배치할 때 동의한 라이선스 계약과 요청된 큐 관리자 컴포넌트에 따라 판별됩니다.

- IBM MQ Advanced 또는 IBM MQ Advanced for Developers 라이선스를 선택하는 경우 추가 서비스가 요청되지 않습니다. 예를 들어, 다음 경우에는 IBM Cloud Pak foundational services가 사용되지 않습니다.

```
spec:
  license:
    accept: true
    license: L-APIG-BZDDDY
    use: "Production"
```

- IBM Cloud Pak for Integration 라이선스를 선택하고 웹 서버를 사용으로 설정하도록 선택하는 경우 IBM MQ Operator가 싱글 사인온을 사용으로 설정하기 위해 IBM IAM(Identity and Access Management) Operator도 인스턴스화합니다. IBM Cloud Pak for Integration Operator를 설치한 경우 IAM Operator가 이미 사용 가능합니다. 예를 들면, 다음과 같습니다.

```
spec:
  license:
    accept: true
```

³ Red Hat OpenShift Container Platform 버전에는 자체 지원 날짜가 적용됩니다. 자세한 정보는 [Red Hat OpenShift Container Platform 라이프사이클 정책](#)을 참조하십시오.

⁴ Red Hat OpenShift Container Platform 4.4는 "수명 종료"에 도달했습니다. 자세한 정보는 [Red Hat OpenShift Container Platform 라이프사이클 정책](#)을 참조하십시오.

⁵ Red Hat OpenShift Container Platform 4.5는 "수명 종료"에 도달했습니다. 자세한 정보는 [Red Hat OpenShift Container Platform 라이프사이클 정책](#)을 참조하십시오.

⁶ IBM MQ Operator은(는) IBM Cloud Pak foundational services에 종속됩니다. Red Hat OpenShift Container Platform 4.7을 사용하려면 먼저 IBM Cloud Pak foundational services 버전을 업그레이드해야 합니다.

```
license: L-RJON-BUVMQX
use: "Production"
```

그러나 웹 서버를 사용 안함으로 설정하면 IBM Cloud Pak foundational services가 요청되지 않습니다. 예를 들면, 다음과 같습니다.

```
spec:
  license:
    accept: true
    license: L-RJON-BUVMQX
    use: "Production"
  web:
    enabled: false
```

이전 버전의 IBM MQ Operator는 라이선스 사용을 추적하기 위해 항상 IBM Licensing Operator(및 해당 종속 항목)의 설치를 요청했습니다. IBM MQ Operator 1.5부터 라이선싱 서비스가 요청되지 않으며 별도로 요청해야 합니다.

IBM MQ Operator에는 1개의 CPU 코어와 1GB 메모리가 필요합니다. 종속 Operator에 대한 하드웨어 및 소프트웨어 요구사항의 자세한 분석은 [기본 서비스에 대한 하드웨어 요구사항 및 권장사항](#)을 참조하십시오.

큐 관리자에서 사용되는 CPU 및 메모리의 양을 선택할 수 있습니다. 자세한 정보는 [128 페이지의 『.spec.queueManager.resources』](#)의 내용을 참조하십시오.

관련 참조

[119 페이지의 『mq.ibm.com/v1beta1에 대한 라이선스 부여 참조』](#)

OpenShift CP4I IBM MQ Operator에서 필요로 하는 클러스터 범위 권한

IBM MQ Operator에서는 허가 웹훅 및 샘플을 관리하고, 스토리지 클래스 및 클러스터 버전 정보를 읽기 위해 클러스터 범위 권한을 필요로 합니다.

IBM MQ Operator에서는 다음 클러스터 범위 권한을 필요로 합니다.

- 허가 웹훅을 관리할 권한. 이는 Operator에서 제공하는 컨테이너를 작성하고 관리하는 프로세스에서 사용되는 특정 웹훅을 작성하고, 검색하고, 업데이트할 수 있게 해 줍니다.
 - API 그룹: **admissionregistration.k8s.io**
 - 자원: **validatingwebhookconfigurations**
 - verb: **create, get, update**
- 사용자 정의 자원을 작성할 때 샘플 및 스니펫을 제공하기 위해 Red Hat OpenShift 콘솔에서 사용되는 자원을 작성하고 관리할 권한.
 - API 그룹: **console.openshift.io**
 - 자원: **consoleyamlsamples**
 - verb: **create, get, update, delete**
- 클러스터 버전을 읽을 권한. 이는 Operator가 클러스터 환경에 대한 문제를 피드백할 수 있게 해 줍니다.
 - API 그룹: **config.openshift.io**
 - 자원: **clusterversions**
 - verb: **get, list, watch**
- 클러스터의 스토리지 클래스를 읽을 권한. 이는 Operator가 컨테이너에서 선택된 스토리지 클래스에 대한 문제를 피드백할 수 있게 해 줍니다.
 - API 그룹: **storage.k8s.io**
 - 자원: **storageclasses**
 - verb: **get, list**

OpenShift CP4I Kubernetes IBM MQ Operator의 스토리지 고려사항

IBM MQ Operator는 두 스토리지 모드로 실행됩니다.

- **임시 스토리지**는 컨테이너 재시작 시 컨테이너에 대한 모든 상태 정보를 제거할 수 있는 경우 사용됩니다. 이는 일반적으로 환경이 데모용으로 작성되거나 독립형 큐 관리자와 함께 개발할 때 사용됩니다.
- **지속 스토리지**는 IBM MQ의 공통 구성이며 컨테이너 재시작되면 기존 구성, 로그 및 지속 메시지가 재시작된 컨테이너에서 사용 가능한지 확인합니다.

IBM MQ Operator는 환경 및 원하는 스토리지 모드에 따라 상당히 다를 수 있는 스토리지 특성을 사용자 정의하는 기능을 제공합니다.

임시 스토리지

IBM MQ는 Stateful 애플리케이션이며 재시작 시 복구를 위해 이 상태를 스토리지로 유지합니다. 임시 스토리지를 사용하는 경우에는 큐 관리자에 대한 모든 상태 정보가 재시작 시 손실됩니다. 여기에는 다음이 포함됩니다.

- 모든 메시지
- 모든 큐 관리자 대 큐 관리자 통신 상태(채널 메시지 순서 번호)
- 큐 관리자의 MQ 클러스터 ID
- 모든 트랜잭션 상태
- 모든 큐 관리자 구성
- 모든 로컬 진단 데이터

이러한 이유로 인해 임시 스토리지가 프로덕션, 테스트 또는 개발 시나리오에 적합한 방법인지 고려해야 합니다. 예를 들어, 모든 메시지가 비지속적이고 큐 매니저가 MQ 클러스터의 멤버가 아니라는 것을 알 수 있습니다. 재시작 시 모든 메시징 상태가 삭제될 뿐만 아니라 큐 관리자의 구성도 제거됩니다. 완전히 임시 컨테이너를 사용으로 설정하려면, IBM MQ 구성이 컨테이너 이미지 자체(자세한 정보는 [106 페이지의 『Red Hat OpenShift CLI를 사용하여 사용자 정의 MQSC 및 INI 파일이 포함된 이미지 빌드』의 내용 참조](#))에 추가되어야 합니다. 이를 완료하지 않으면 컨테이너를 다시 시작할 때마다 IBM MQ가 구성되어야 합니다.

 예를 들어, IBM MQ를 임시 스토리지로 구성하려면, QueueManager의 스토리지 유형은 다음을 포함합니다.

```
queueManager:
  storage:
    queueManager:
      type: ephemeral
```

지속 스토리지



IBM MQ는 일반적으로 큐 관리자가 재시작 후 지속 메시지 및 구성을 유지하도록 하기 위해 지속 스토리지로 실행합니다. 그러므로 이것이 기본 동작입니다. 다양한 스토리지 제공자 및 다른 각 지원 기능으로 인해 이는 종종 구성의 사용자 정의가 필요함을 의미합니다. 아래 예제에서는 v1beta1 API에서 MQ 스토리지 구성을 사용자 정의하는 공용 필드에 대해 설명합니다.

- `spec.queueManager.availability`는 가용성 모드를 제어합니다. `ReadWriteOnce` 스토리지만 필요한 `SingleInstance`를 사용 중인 경우, `multiInstance`는 오라클 파일 잠금 특성을 가진 `ReadWriteMany`를 지원하는 스토리지 클래스를 필요로 합니다. IBM MQ는 지원 명령문 및 명령문 테스트를 제공합니다. 또한 가용성 모드는 지속 볼륨 레이아웃에 영향을 줍니다. 추가 정보는 [15 페이지의 『컨테이너에서 IBM MQ에 대한 고가용성』](#)의 내용을 참조하십시오.
- `spec.queueManager.storage`는 개별 스토리지 설정을 제어합니다. 큐 관리자는 1~3개 사이의 지속 볼륨을 사용하도록 구성될 수 있습니다.

다음 예는 단일 인스턴스 큐 관리자를 사용하는 단순 구성의 스니펫을 표시합니다.

```
spec:
  queueManager:
    storage:
      queueManager:
        enabled: true
```

다음 예는 기본이 아닌 스토리지 클래스 및 추가 그룹이 필요한 파일 스토리지를 포함한 다중 인스턴스 큐 관리자 구성의 스니펫을 표시합니다.

```
spec:
  queueManager:
    availability:
      type: MultiInstance
    storage:
      queueManager:
        class: ibmc-file-gold-gid
      persistedData:
        enabled: true
        class: ibmc-file-gold-gid
      recoveryLogs:
        enabled: true
        class: ibmc-file-gold-gid
  securityContext:
    supplementalGroups: [99]
```

참고: 단일 인스턴스 큐 관리자로 추가 그룹을 구성할 수도 있습니다.

참고: 원시 HA를 사용하는 경우 공유 파일 시스템이 필요하지 않습니다 (15 페이지의 『컨테이너에서 IBM MQ에 대한 고가용성』 참조). 특히 NFSv3을 사용해서는 안 됩니다.

Linux 자체 IBM MQ 큐 관리자 컨테이너 이미지 빌드 지원

IBM MQ는 GitHub에 IBM MQ 큐 관리자 컨테이너를 빌드하기 위한 코드를 제공합니다. 이는 IBM에서 자체 지원 컨테이너를 빌드하는 데 사용하는 프로세스를 기반으로 하며 이 GitHub 저장소를 사용하여 자체 컨테이너 이미지 빌드를 단순화하고 가속화할 수 있습니다.

이 코드는 mq-container GitHub 저장소(<https://github.com/ibm-messaging/mq-container>)에 제공됩니다. 이는 커뮤니티에서 제공하는 지원과 함께 Apache 2.0 라이선스에 따라 제공됩니다.

이 저장소는 표준 Linux rpm 패키지를 사용하지 않습니다. 압축된 패키지를 컨테이너 배치에 사용합니다. 이 접근 방식의 이점은 에스컬레이션된 권한 없이도 더 안전한 컨테이너 환경에서 실행할 수 있다는 점입니다. 그러나 IBM MQ는 일반적으로 운영 체제 기반 인증에 에스컬레이션된 권한을 사용하므로 이 접근 방식은 사용 가능한 보안 옵션에 영향을 줍니다. 컨테이너 배치의 경우 운영 체제 기반 인증을 사용하는 것은 일반적으로 좋은 방법이 아닙니다. 대신 상호 TLS 또는 LDAP 인증을 사용할 수 있습니다. IBM MQ Advanced for Developers이(가) 있으면 파일 기반 인증을 사용하여 사용자가 빠르게 시작하도록 할 수도 있습니다.

복제된 데이터 큐 관리자 (RDQM) 는 컨테이너 환경에서 지원되지 않습니다. 87 페이지의 『고유 HA』 을(를) 사용하여 RDQM과 유사한 기능을 얻을 수 있습니다.

관련 개념

6 페이지의 『IBM MQ Operator에 대한 지원』

IBM MQ Operator는 Red Hat OpenShift Container Platform에 배치된 경우에만 지원됩니다.

[IBM MQ 비설치 이미지](#)

Linux IBM MQ 컨테이너 이미지를 직접 빌드하는 경우의 라이선스 어노테이션

라이선스 어노테이션은 기본 시스템이 아니라 컨테이너에 정의된 한계에 따라 사용량을 추적할 수 있게 해 줍니다. 사용자는 IBM License Service에서 사용량을 추적하는 데 사용하는 특정 어노테이션과 함께 컨테이너를 배치하도록 클라이언트를 구성합니다.

직접 빌드한 IBM MQ 컨테이너 이미지를 배치하는 경우의 라이선스 부여에는 두 가지 접근법이 있습니다.

- 컨테이너를 실행하는 전체 시스템에 라이선스를 부여합니다.
- 연관된 한계에 따라 컨테이너에 라이선스를 부여합니다.

두 옵션 모두 클라이언트가 사용할 수 있으며 자세한 내용은 Passport Advantage®의 [IBM 컨테이너 라이선스 페이지](#)에서 찾을 수 있습니다.

컨테이너 한계에 따라 IBM MQ 컨테이너에 라이선스가 부여되는 경우에는 사용량 추적을 위해 IBM License Service를 설치해야 합니다. 지원되는 환경 및 설치 지시사항에 대한 추가적인 정보는 GitHub의 [ibm-licensing-operator](#) 페이지에 있습니다.

IBM License Service 는 IBM MQ 컨테이너가 배치된 Kubernetes 클러스터에 설치되고 팟 어노테이션은 사용량을 추적하는 데 사용됩니다. 따라서 클라이언트는 IBM License Service에서 사용하는 특정 어노테이션과 함께 팟(Pod)을 배치해야 합니다. 사용자의 인타이틀먼트 및 컨테이너에 배치되는 기능에 따라 다음 어노테이션을 하나 이상 사용하십시오.

- [13 페이지의 『IBM MQ Advanced 컨테이너』](#)
- [13 페이지의 『IBM MQ Advanced High Availability Replica 컨테이너』](#)
- [13 페이지의 『IBM MQ Base 컨테이너』](#)
- [13 페이지의 『IBM MQ Base High Availability Replica 컨테이너』](#)
- [13 페이지의 『IBM MQ Advanced for Developers 컨테이너』](#)
- [14 페이지의 『CP4I 인타이틀먼트\(프로덕션\) 포함 IBM MQ Advanced 컨테이너』](#)
- [14 페이지의 『CP4I 인타이틀먼트\(프로덕션\) 포함 IBM MQ Advanced High Availability Replica 컨테이너』](#)
- [14 페이지의 『CP4I 인타이틀먼트\(비프로덕션\) 포함 IBM MQ Advanced 컨테이너』](#)
- [14 페이지의 『CP4I 인타이틀먼트\(비프로덕션\) 포함 IBM MQ Advanced High Availability Replica 컨테이너』](#)
- [14 페이지의 『CP4I 인타이틀먼트\(프로덕션\) 포함 IBM MQ Base 컨테이너』](#)
- [14 페이지의 『CP4I 인타이틀먼트\(프로덕션\) 포함 IBM MQ Base High Availability Replica 컨테이너』](#)
- [15 페이지의 『CP4I 인타이틀먼트\(비프로덕션\) 포함 IBM MQ Base 컨테이너』](#)
- [15 페이지의 『CP4I 인타이틀먼트\(비프로덕션\) 포함 IBM MQ Base High Availability Replica 컨테이너』](#)

IBM MQ Advanced 컨테이너

```
productName: "IBM MQ Advanced"  
productID: "208423bb063c43288328b1d788745b0c"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"
```

IBM MQ Advanced High Availability Replica 컨테이너

```
productName: "IBM MQ Advanced High Availability Replica"  
productID: "546cb719714942c18748137ddd8d5659"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"
```

IBM MQ Base 컨테이너

```
productName: "IBM MQ"  
productID: "c661609261d5471fb4ff8970a36bccea"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"
```

IBM MQ Base High Availability Replica 컨테이너

```
productName: "IBM MQ High Availability Replica"  
productID: "2a2a8e0511c849969d2f286670ea125e"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"
```

IBM MQ Advanced for Developers 컨테이너

```
productName: "IBM MQ Advanced for Developers"
```

```
productID: "2f886a3eefbe4ccb89b2adb97c78b9cb"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productMetric: "FREE"
```

CP4I 인타이틀먼트(프로덕션) 포함 IBM MQ Advanced 컨테이너

```
productName: "IBM MQ Advanced with CP4I License"  
productID: "208423bb063c43288328b1d788745b0c"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productMetric: "VIRTUAL_PROCESSOR_CORE"  
productCloudpakRatio: "2:1"  
cloudpakName: "IBM Cloud Pak for Integration"  
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

CP4I 인타이틀먼트(프로덕션) 포함 IBM MQ Advanced High Availability Replica 컨테이너

```
productName: "IBM MQ Advanced High Availability Replica with CP4I License"  
productID: "546cb719714942c18748137ddd8d5659"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productMetric: "VIRTUAL_PROCESSOR_CORE"  
productCloudpakRatio: "10:1"  
cloudpakName: "IBM Cloud Pak for Integration"  
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

CP4I 인타이틀먼트(비프로덕션) 포함 IBM MQ Advanced 컨테이너

```
productName: "IBM MQ Advanced for Non-Production with CP4I License"  
productID: "21dfe9a0f00f444f888756d835334909"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productMetric: "VIRTUAL_PROCESSOR_CORE"  
productCloudpakRatio: "4:1"  
cloudpakName: "IBM Cloud Pak for Integration"  
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

CP4I 인타이틀먼트(비프로덕션) 포함 IBM MQ Advanced High Availability Replica 컨테이너

```
productName: "IBM MQ Advanced High Availability Replica for Non-Production with CP4I License"  
productID: "b3f8f984007d47fb981221589cc50081"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productMetric: "VIRTUAL_PROCESSOR_CORE"  
productCloudpakRatio: "20:1"  
cloudpakName: "IBM Cloud Pak for Integration"  
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

CP4I 인타이틀먼트(프로덕션) 포함 IBM MQ Base 컨테이너

```
productName: "IBM MQ with CP4I License"  
productID: "c661609261d5471fb4ff8970a36bceca"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"  
productMetric: "VIRTUAL_PROCESSOR_CORE"  
productCloudpakRatio: "4:1"  
cloudpakName: "IBM Cloud Pak for Integration"  
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

CP4I 인타이틀먼트(프로덕션) 포함 IBM MQ Base High Availability Replica 컨테이너

```
productName: "IBM MQ High Availability Replica with CP4I License"  
productID: "2a2a8e0511c849969d2f286670ea125e"  
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

```
productMetric: "VIRTUAL_PROCESSOR_CORE"
productCloudpakRatio: "20:1"
cloudpakName: "IBM Cloud Pak for Integration"
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

CP4I 인타이틀먼트(비프로덕션) 포함 IBM MQ Base 컨테이너

```
productName: "IBM MQ with CP4I License Non-Production"
productID: "151bec68564a4a47a14e6fa99266deff"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productCloudpakRatio: "8:1"
cloudpakName: "IBM Cloud Pak for Integration"
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

CP4I 인타이틀먼트(비프로덕션) 포함 IBM MQ Base High Availability Replica 컨테이너

```
productName: "IBM MQ High Availability Replica with CP4I License Non-Production"
productID: "f5d0e21c013c4d4b8b9b2ce701f31928"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productCloudpakRatio: "40:1"
cloudpakName: "IBM Cloud Pak for Integration"
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

OpenShift

CP4I

Kubernetes

컨테이너에서 IBM MQ에 대한 고가용성

IBM MQ Operator의 고가용성을 위한 세 가지 선택사항이 있습니다. **고유 HA 큐 관리자**(하나의 활성 복제본과 두 개의 대기 복제본이 있음), **다중 인스턴스 큐 관리자**(네트워크로 연결된 공유 파일 시스템을 사용하는 활성-대기 쌍) 또는 **단일 복원성 큐 관리자**(네트워크로 연결된 스토리지를 사용하는 단순한 HA 접근 방식을 제공함)입니다. 후자의 두 가지는 복구 가능한 데이터의 가용성을 보장하기 위해 파일 시스템에 의존하지만 고유 HA는 그렇지 않습니다. 따라서 고유 HA를 사용하지 않는 경우 파일 시스템의 가용성이 큐 관리자 가용성에 중요합니다. 데이터 복구가 중요한 경우 파일 시스템은 복제를 통해 중복성을 보증해야 합니다.

메시지 및 서비스 가용성을 별도로 고려해야 합니다. IBM MQ for Multiplatforms에서 메시지는 정확히 하나의 큐 관리자에 저장됩니다. 따라서 해당 큐 관리자가 사용 불가능해지면 여기서 보유하는 메시지에 대한 액세스 권한을 일시적으로 잃을 수 있습니다. 메시지 고가용성을 얻기 위해 가능한 한 빨리 큐 관리자를 복구할 수 있어야 합니다. 서비스 가용성은 예를 들어, IBM MQ 균일 클러스터를 사용하여 클라이언트 애플리케이션의 다중 인스턴스를 사용할 수 있도록 구성해 얻을 수 있습니다.

큐 관리자는 디스크에 저장된 데이터와 데이터에 액세스할 수 있는 실행 중인 프로세스와 같은 두 개의 파트로 간주할 수 있습니다. 동일한 데이터 (Kubernetes Persistent Volumes에서 제공)를 유지하고 클라이언트 애플리케이션이 네트워크를 통해 여전히 주소 지정 가능하면 모든 큐 관리자를 다른 Kubernetes 노드로 이동할 수 있습니다. Kubernetes에서 서비스는 일관된 네트워크 ID를 제공하기 위해 사용됩니다.

IBM MQ는 지속적 볼륨의 데이터 가용성에 의존합니다. 따라서 지속적 볼륨을 제공하는 스토리지의 가용성은 큐 관리자 가용성에 매우 중요합니다. IBM MQ는 사용하는 스토리지 가용성보다 더 높을 수 없기 때문입니다. 전체 가용성 구역의 가동 중단을 허용하려는 경우 디스크 쓰기를 다른 구역으로 복제하는 볼륨 제공자를 사용해야 합니다.

고유 HA 큐 관리자

CP4I

V 9.2.3

Native HA queue managers are available from IBM Cloud Pak for Integration 2021.2.1, using IBM MQ Operator 1.6 or higher, with IBM MQ 9.2.3 or higher.

고유 HA 큐 관리자에는 하나의 **활성** 및 두 개의 **복제본** Kubernetes 팟(Pod)이 포함되고, 이는 정확히 세 개의 복제본이 있고 각각에는 자체 Kubernetes 지속 볼륨 세트가 있는 Kubernetes StatefulSet의 일부로서 실행됩니다. 고유 HA 큐 관리자를 사용하는 경우에도 공유 파일 시스템에 대한 IBM MQ 요구사항이 적용되지만(리스 기반 잠금 제외), 반드시 공유 파일 시스템을 사용해야 하는 것은 아닙니다. 적합한 파일 시스템 외에도 블록 스토리

지를 사용할 수 있습니다. 예를 들어, *xfs* 또는 *ext4*와 같습니다. 고유 HA 큐 관리자의 복구 시간은 다음 요소로 제어됩니다.

1. 복제본 인스턴스가 활성 인스턴스의 장애 여부를 발견하는 데 소요되는 시간 이는 구성 가능합니다.
2. Kubernetes 팟(Pod) 준비 프로브가 준비 컨테이너가 변경되어 네트워크 트래픽의 경로 재지정이 발생했음을 발견하는 데 걸리는 시간. 이는 구성 가능합니다.
3. IBM MQ 클라이언트가 다시 연결하는 데 소요되는 시간

자세한 정보는 [87 페이지의 『고유 HA』](#)의 내용을 참조하십시오.

다중 인스턴스 큐 관리자



다중 인스턴스 큐 관리자는 **활성 및 대기** Kubernetes 팟(Pod)을 포함합니다. 이는 정확히 2개의 복제본과 Kubernetes 지속적 볼륨 세트를 포함하는 Kubernetes 상태 지원 세트의 일부로 실행됩니다. 큐 관리자 트랜잭션 로그 및 데이터는 공유 파일 시스템을 사용하여 두 개의 지속적 볼륨에 보유됩니다.

다중 인스턴스 큐 관리자에는 지속적 볼륨에 대한 동시 액세스를 지원하기 위해 **활성 및 대기** 팟(Pod)이 모두 필요합니다. To configure this, you use Kubernetes Persistent Volumes with **access mode set to ReadWriteMany**. 또한 이 볼륨은 IBM MQ의 공유 파일 시스템에 대한 요구사항을 충족해야 합니다. IBM MQ는 큐 관리자 장애 복구를 시행하기 위해 파일 잠금의 자동 릴리스에 의존합니다. IBM MQ에서는 [테스트하는 파일 시스템 목록을 생성합니다](#).

멀티 인스턴스 큐 관리자의 복구 시간은 다음 요소로 제어됩니다.

1. 실패 후 공유된 파일 시스템이 활성 인스턴스에서 처음에 설정한 잠금을 릴리스하는 데 걸리는 시간.
2. 대기 인스턴스가 잠금을 확보하고 시작하는 데 걸리는 시간.
3. Kubernetes 팟(Pod) 준비 프로브가 준비 컨테이너가 변경되어 네트워크 트래픽의 경로 재지정이 발생했음을 발견하는 데 걸리는 시간. 이는 구성 가능합니다.
4. IBM MQ 클라이언트를 다시 연결하는 데 걸리는 시간.

단일 복원성 큐 관리자



단일 복원력 큐 관리자는 단일 Kubernetes 팟(Pod)에서 실행되는 큐 관리자의 단일 인스턴스입니다. 여기서, Kubernetes는 큐 관리자를 모니터링하고 필요한 경우 팟(Pod)을 대체합니다.

IBM MQ 공유 파일 시스템의 요구사항은 단일 복원력 큐 관리자(리스 기반 잠금 제외)를 사용하는 경우에도 적용되지만, 공유 파일 시스템을 사용할 필요는 없습니다. 적합한 파일 시스템 외에도 블록 스토리지를 사용할 수 있습니다. 예를 들어, *xfs* 또는 *ext4*와 같습니다.

단일 복원력 큐 관리자의 복구 시간은 다음 요소로 제어됩니다.

1. 라이브 프로브를 실행하는 데 걸리는 시간과 허용하는 장애 수. 이는 구성 가능합니다.
2. Kubernetes 스케줄러가 새 노드로 장애가 발생한 팟(Pod)을 다시 스케줄하는 데 걸리는 시간.
3. 컨테이너 이미지를 새 노드로 다운로드하는 데 걸리는 시간. IfNotPresent의 **imagePullPolicy** 값을 사용하는 경우 이미지는 해당 노드에서 이미 사용 가능할 수 있습니다.
4. 새 큐 관리자 인스턴스를 시작하는 데 걸리는 시간.
5. Kubernetes 팟(Pod) 프로브가 컨테이너가 준비되었는지 감지하는 데 걸리는 시간. 이는 구성 가능합니다.
6. IBM MQ 클라이언트를 다시 연결하는 데 걸리는 시간.

중요사항:

단일 복원력 큐 관리자 패턴에서 몇 가지 혜택을 제공해도, 노드 장애에 관한 제한사항이 적용된 상태에서 가용성 목표를 달성할 수 있는지 여부를 파악해야 합니다.

Kubernetes에서 장애가 발생한 팟(Pod)은 일반적으로 빠르게 복구됩니다. 하지만 전체 노드의 장애는 다르게 처리됩니다. When using a stateful workload like IBM MQ with a Kubernetes StatefulSet, if a Kubernetes Master Node loses contact with a worker node, it cannot determine if the node has failed or if it has simply

lost network connectivity. 따라서, Kubernetes에서 다음 이벤트 중 하나가 발생할 때까지 이 경우에는 조치를 취하지 않습니다.

1. 노드는 Kubernetes 마스터 노드가 이와 통신할 수 있는 상태로 복구됩니다.
2. Kubernetes 마스터 노드에서 명시적으로 팟(Pod)을 삭제하도록 관리 조치가 수행됩니다. 이로 인해 반드시 팟(Pod)이 실행되지 않는 것은 아니며, Kubernetes 저장소에서만 삭제될 뿐입니다. 따라서 이 관리 조치는 매우 신중하게 수행해야 합니다.

관련 태스크

87 페이지의 『IBM MQ Operator를 사용하여 큐 관리자에 대한 고가용성 구성』

관련 참조

[고가용성 구성](#)

OpenShift

CP4I

Kubernetes

컨테이너에서 IBM MQ의 재해 복구

준비 중인 재해의 종류를 고려해야 합니다. 클라우드 환경에서 가용성 구역을 사용하면 어느 정도 재해를 견딜 수 있으므로 사용하기가 훨씬 용이해집니다. (쿼럼에 대해) 홀수의 데이터 센터와 대기 시간이 짧은 하나의 네트워크 링크가 있는 경우, 여러 가용성 구역에서 단일 Red Hat OpenShift Container Platform 또는 Kubernetes 클러스터를 각각의 실제 위치에서 한 번씩 실행할 수 있습니다. 이 토픽에서는 이러한 기준을 충족할 수 없는 재해 복구를 위한 고려사항 즉, 짝수의 데이터 센터 또는 대기 시간이 긴 네트워크 링크에 대해 설명합니다.

재해 복구의 경우 다음을 고려해야 합니다.

- IBM MQ 데이터(하나 이상의 PersistentVolume 자원에 보유됨)를 재해 복구 위치에 복제
- 복제된 데이터를 사용하여 큐 관리자 다시 작성
- IBM MQ 클라이언트 애플리케이션 및 기타 큐 관리자에 표시되는 큐 관리자 네트워크 ID. 예를 들어, 이 ID는 DNS 항목일 수 있습니다.

지속 데이터는 재해 복구 사이트에 동기식 또는 비동기식으로 복제되어야 합니다. 이는 일반적으로 스토리지 제 공자에만 해당되지만 VolumeSnapshot(를) 사용하여 수행할 수도 있습니다. 볼륨 스냅샷에 대한 자세한 정보는 [CSI 볼륨 스냅샷](#)을 참조하십시오.

재해로부터 복구 중인 경우 복제된 데이터를 사용하여 새 Kubernetes 클러스터에서 큐 관리자 인스턴스를 다시 작성해야 합니다. IBM MQ Operator(를) 사용하는 경우, ConfigMap 또는 Secret과(와) 같은 자원을 지원하는 YAML 뿐만 아니라 QueueManager YAML이 필요할 것입니다.

관련 정보

[ha_for_ctr.dita](#)

컨테이너에서 IBM MQ에 대한 사용자 인증 및 권한 부여

IBM MQ는 LDAP 사용자 및 그룹을 사용하도록 구성할 수 있습니다. 또는 컨테이너 이미지 내에서 로컬 운영 체제 사용자 및 그룹을 사용할 수 있습니다. IBM MQ Operator에서는 보안 문제로 인해 운영 체제 사용자 및 그룹의 사용자가 허용되지 않습니다.

다중 테넌트 컨테이너형 환경에서 보안 제한조건은 보통 다음과 같은 잠재적인 보안 문제를 예방하기 위해 적용됩니다.

- 컨테이너 내에서 "루트" 사용자 사용 금지
- 임의적 **UID** 사용 강제 실행. 예를 들어 Red Hat OpenShift Container Platform에서 기본 SecurityContextConstraints(restricted라고 함)에서는 각 컨테이너에 대해 무작위 사용자 ID를 사용합니다.
- 권한 상승 사용 금지. Linux의 IBM MQ는 권한 에스컬레이션을 사용하여 사용자의 비밀번호를 확인합니다. 사용자는 "setuid" 프로그램을 사용하여 이를 수행하는 "root" 사용자가 됩니다.

OpenShift

CP4I

이러한 보안 조치를 준수하도록 IBM MQ Operator에서는 컨테이너 내부의 운영 체제 라이브러리에 정의된 ID를 사용하지 못하도록 합니다. 컨테이너에 mqm 사용자 ID 및 그룹이 정의되지 않습니다. IBM Cloud Pak for Integration 및 Red Hat OpenShift에서 IBM MQ를 사용하는 경우 사용자 인증 및 권한 부여를 위해 LDAP을 사용하도록 큐 관리자를 구성해야 합니다. 이 작업을 수행하도록 IBM MQ를 구성하는 방법에 대한 정보는 [연결 인증: 사용자 저장소 및 LDAP 권한 부여](#)를 참조하십시오.

대부분의 경우 컨테이너에서 IBM MQ 의 스케일링 및 성능은 멀티플랫폼용 IBM MQ 와 동일합니다. 그러나 컨테이너 플랫폼에 의해 부과될 수 있는 몇 가지 추가 한계가 있습니다.

이 태스크 정보

컨테이너에서 IBM MQ 의 확장성 및 성능을 계획할 때 다음 옵션을 고려하십시오.

프로시저

- 스레드 및 프로세스 수를 제한하십시오.

IBM MQ 는 스레드를 사용하여 동시성을 관리합니다. Linux에서는 스레드가 프로세스로 구현되므로 최대 프로세스 수에 따라 컨테이너 플랫폼 또는 운영 체제에서 부과하는 한계가 발생할 수 있습니다. Red Hat OpenShift Container Platform에서 컨테이너당 기본 4096프로세스 한계가 있습니다 (1024프로세스는 OpenShift 4.11까지). 이는 대부분의 시나리오에 적합하지만 큐 관리자의 클라이언트 연결 수에 영향을 줄 수 있는 경우가 있을 수 있습니다.

Kubernetes 의 프로세스 한계는 kubelet 구성 설정 `podPidsLimit`를 사용하여 클러스터 관리자가 구성할 수 있습니다. Kubernetes 문서의 [프로세스 ID 한계 및 예약](#) 을 참조하십시오. Red Hat OpenShift Container Platform에서도 CRI-O 매개변수를 편집하기 위한 `ContainerRuntimeConfig` 사용자 정의 자원 작성을 사용할 수 있습니다.

IBM MQ 구성에서 큐 관리자에 대한 최대 클라이언트 연결 수를 설정할 수도 있습니다. 개별 서버 연결 채널에 제한을 적용하려면 [서버 연결 채널 한계](#) 를, 전체 큐 관리자에 제한을 적용하려면 [MAXCHANNELS INI](#) 속성을 참조하십시오.

- 볼륨 수를 제한하십시오.

클라우드 및 컨테이너 시스템에서는 네트워크 접속 스토리지 볼륨이 일반적으로 사용됩니다. Linux 노드에 첨부할 수 있는 볼륨 수에 제한이 있습니다. 예를 들어, AWS EC2 는 VM당 30볼륨 이하로 제한됩니다. Red Hat OpenShift Container Platform 유사한 한계를 가지고 있다, as do Microsoft Azure and Google Cloud Platform.

기본 HA큐 관리자는 세 개의 인스턴스 각각에 대해 하나의 볼륨을 필요로 하고 노드 사이에 인스턴스를 적용해야 합니다. 그러나 인스턴스당 세 개의 볼륨 (큐 관리자 데이터, 복구 로그 및 지속된 데이터) 을 사용하도록 큐 관리자를 구성할 수 있습니다.

- IBM MQ 스케일링 기술을 사용하십시오.

적은 수의 대형 큐 관리자 대신 IBM MQ 균일 클러스터와 같은 IBM MQ 스케일링 기술을 사용하여 동일한 구성의 다중 큐 관리자를 실행하는 것이 좋습니다. 이는 단일 컨테이너 다시 시작 (예: 컨테이너 플랫폼 유지보수의 일부) 이 미치는 영향이 감소하는 장점이 있습니다.

Red Hat OpenShift에서 IBM MQ 사용

IBM MQ Operator는 IBM MQ를 IBM Cloud Pak for Integration의 일부로 또는 독립형으로 Red Hat OpenShift Container Platform에 배치하고 관리합니다.

프로시저

- [19 페이지의 『IBM MQ Operator의 릴리스 히스토리』](#).
- [35 페이지의 『IBM MQ를 IBM Cloud Pak for Integration으로 마이그레이션』](#).
- [56 페이지의 『Red Hat OpenShift에 IBM MQ Operator 설치 및 설치 제거』](#).
- [68 페이지의 『IBM MQ Operator 및 큐 관리자 업그레이드』](#).
- [75 페이지의 『IBM MQ Operator를 사용하여 큐 관리자 배치 및 구성』](#).
- [108 페이지의 『IBM MQ Operator 를 사용하여 IBM MQ 운영』](#).
- [119 페이지의 『IBM MQ Operator에 대한 API 참조』](#).

IBM MQ Operator

IBM MQ Operator 1.8.2

CD

IBM Cloud Pak for Integration 버전

IBM Cloud Pak for Integration 2021.4.1

Operator 채널

v1.8

.spec.version에 허용되는 값

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.0-r3, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.1.0-r1, 9.2.1.0-r2, 9.2.2.0-r1, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3

Red Hat OpenShift Container Platform 버전

Red Hat OpenShift Container Platform 4.6 이상

IBM Cloud Pak foundational services 버전

IBM Cloud Pak foundational services 3.8이상(v3 채널)

새로운 기능

- IBM MQ Operator 1.8.0. 위에 빌드된 보안 전용 업데이트
- 주소 지정된 취약성은 이 보안 게시판에 자세히 설명되어 있습니다.

IBM MQ Operator 1.8.1

CD

IBM Cloud Pak for Integration 버전

IBM Cloud Pak for Integration 2021.4.1

Operator 채널

v1.8

.spec.version에 허용되는 값

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.0-r3, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.1.0-r1, 9.2.1.0-r2, 9.2.2.0-r1, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2

Red Hat OpenShift Container Platform 버전

Red Hat OpenShift Container Platform 4.6 이상

IBM Cloud Pak foundational services 버전

IBM Cloud Pak foundational services 3.8이상(v3 채널)

새로운 기능

- IBM MQ Operator 1.8.0. 위에 빌드된 보안 전용 업데이트
- 주소 지정된 취약성은 이 보안 게시판에 자세히 설명되어 있습니다.

IBM MQ Operator 1.8.0

CD

IBM Cloud Pak for Integration 버전

IBM Cloud Pak for Integration 2021.4.1

Operator 채널

v1.8

.spec.version에 허용되는 값

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.0-r3, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.1.0-r1, 9.2.1.0-r2, 9.2.2.0-r1, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1

Red Hat OpenShift Container Platform 버전

Red Hat OpenShift Container Platform 4.6 이상

IBM Cloud Pak foundational services 버전

IBM Cloud Pak foundational services 3.8이상(v3 채널)

새로운 기능

- 사용되지 않는 IBM MQ 버전에 대한 상태 조건을 추가합니다.

변경사항

- Docker 허브에서 IBM Container Registry(으)로 이미지가 이동했습니다.
 - 방화벽 규칙이 있는 고객은 IBM Container Registry의 이미지에 액세스하도록 이를 조정해야 합니다.
 - Airgap 고객은 IBM MQ Operator 1.8.0으로 업그레이드할 때 노드 다시 시작을 경험합니다.
- 사용되지 않는 버전: IBM MQ 9.1.5, 9.2.0 CD, 9.2.1, 9.2.2. 이러한 버전은 IBM MQ Operator의 향후 버전에 의해 조정되지 않을 수 있습니다.
- 라이선스 로직 변경: IBM MQ 9.2.5(으)로 업그레이드하는 고객은 IBM MQ 9.2.5(으)로 작업하도록 지정된 라이선스만 사용할 수 있습니다. [119 페이지의 『mq.ibm.com/v1beta1에 대한 라이선스 부여 참조』](#)의 내용을 참조하십시오.
- 주소 지정된 취약성은 이 [보안 게시판](#)에 자세히 설명되어 있습니다.

IBM MQ Operator 1.7.0

CD

IBM Cloud Pak for Integration 버전

IBM Cloud Pak for Integration 2021.4.1

Operator 채널

v1.7

.spec.version에 허용되는 값

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.0-r3, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.1.0-r1, 9.2.1.0-r2, 9.2.2.0-r1, 9.2.3.0-r1, 9.2.4.0-r1

Red Hat OpenShift Container Platform 버전

Red Hat OpenShift Container Platform 4.6 이상

IBM Cloud Pak foundational services 버전

IBM Cloud Pak foundational services 3.8이상(v3 채널)

새로운 기능

- IBM MQ 9.2.4가 Continuous Delivery 릴리스로 추가되었습니다.

IBM MQ Operator 1.6.0

CD

IBM Cloud Pak for Integration 버전

IBM Cloud Pak for Integration 2021.2.1

Operator 채널

v1.6

.spec.version에 허용되는 값

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.0-r3, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.1.0-r1, 9.2.1.0-r2, 9.2.2.0-r1, 9.2.3.0-r1

Red Hat OpenShift Container Platform 버전

Red Hat OpenShift Container Platform 4.6 이상

IBM Cloud Pak foundational services 버전

IBM Cloud Pak foundational services 3.7이상(v3 채널)

새로운 기능

- 연속 전달 릴리스로 IBM MQ 9.2.3을 추가합니다(IBM MQ 라이선스를 사용하는 경우 IBM Cloud Pak for Integration 2021.2.1, amd64 또는 s390x 의 경우에만amd64).
- 큐 관리자에 대한 새로운 유형의 가용성: 고유 HA. IBM Cloud Pak for Integration 2021.2.1의 일부로 프로덕션용으로 사용할 수 있습니다.

변경사항

- IBM MQ Operator 1.6 이상에서는 Docker Hub 대신 IBM Container Registry를 사용합니다. 이는 icr.io에서 CatalogSource을(를) 사용해야 함을 의미합니다. 56 페이지의 『Red Hat OpenShift에 IBM MQ Operator 설치 및 설치 제거』의 내용을 참조하십시오.
- 고유 HA 롤링 업데이트는 다음 복제본으로 이동하기 전에 복제본이 동기화될 때까지 더 이상 대기하지 않습니다.
- OCP 4.7 이상에서 고유 HA 연관관계에 대한 문제가 수정되었습니다.
- CA 서명 인증서를 고유 HA와 함께 사용할 때 발생하는 문제가 수정되었습니다.

IBM MQ Operator 1.5.0

CD

IBM Cloud Pak for Integration 버전

IBM Cloud Pak for Integration 2021.1.1

Operator 채널

v1.5

.spec.version에 허용되는 값

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.0-r3, 9.2.0.1-r1-eus, 9.2.1.0-r1, 9.2.1.0-r2, 9.2.2.0-r1

Red Hat OpenShift Container Platform 버전

Red Hat OpenShift Container Platform 4.6 이상

IBM Cloud Pak foundational services 버전

IBM Cloud Pak foundational services 3.7이상(v3 채널)

새로운 기능

- 연속 전달 릴리스로 IBM MQ 9.2.2를 추가합니다(IBM MQ 라이선스를 사용하는 경우 IBM Cloud Pak for Integration 2021.1.1; amd64 또는 s390x용으로만 amd64)
- 큐 관리자에 대한 새로운 유형의 가용성: 고유 HA. 평가 용도로만 사용 가능하고, IBM Cloud Pak for Integration 2021.1.1의 일부로서 사용 가능합니다.
- ServiceMonitor 자원을 제공하여 Prometheus 메트릭에 대한 Red Hat OpenShift Container Platform 클러스터 모니터링과 통합

변경사항

- IBM Licensing Operator는 큐 관리자를 작성할 때 기본적으로 더 이상 작성되지 않습니다.
- 다중 인스턴스 큐 관리자에 대한 업데이트는 이제 롤링 순서대로 처리됩니다. 이 변경사항의 일부로, 라이브 프로브를 구성할 때 사용되는 값에 영향을 주는 Kubernetes 시동 프로브가 도입되었습니다. 시동 프로브는 즉시 시작된 후 큐 관리자가 성공적으로 시작될 때까지 대기합니다. 이 대기 기간 내에 언제든지 시동 프로브가 통과되면 라이브 및 준비 프로브가 시작됩니다. 이전에 시작하는 데 느린 큐 관리자가 있는 경우, 활동 프로브에서 initialDelaySeconds 설정을 증가시켰을 수 있습니다. 이를 수행한 경우, 이제 initialDelaySeconds을(를) 이전 설정으로 되돌립니다.
- CustomResourceDefinition이(가) apiextensions.k8s.io/v1beta1에서 apiextensions.k8s.io/v1(으)로 업그레이드됩니다.

알려진 문제점 및 제한사항

- IBM Cloud Pak foundational services 3.7이 필요하며, IAM(Identity and Access Management) 컴포넌트에서 호환 가능하지 않은 변경사항이 포함되어 있습니다. IBM Cloud Pak for Integration 라이선스를 사용하는 큐 관리자가 있는 경우, 이 업그레이드 후 웹 콘솔에 액세스하려면 큐 관리자를 재시작해야 하며 웹 콘솔에 기타 오류 로깅이 표시됩니다. 운영자 업그레이드가 완료된 후 선택한 IBM MQ 버전에 대해 `.spec.version`의 최신 값으로 업그레이드하여 이러한 오류를 수정할 수 있습니다.
- MQ 버전을 업그레이드 중인 경우 업데이트 롤링이 자동으로 시작되지 않습니다. 수동으로 팟(Pod)을 삭제해야 합니다.

IBM MQ Operator 1.4.0

CD

IBM Cloud Pak for Integration 버전

IBM Cloud Pak for Integration 2020.4.1(IBM MQ Operator 1.4.0은 CD 릴리스이며 Extended Update Support의 대상이 아님)

Operator 채널

v1.4

.spec.version에 허용되는 값

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.1-r1-eus, 9.2.1.0-r1

Red Hat OpenShift Container Platform 버전

Red Hat OpenShift Container Platform 4.6 이상

새로운 기능

- IBM MQ 9.2.1이 Continuous Delivery 릴리스로 추가되었습니다.
- 이제 `.spec.queueManager.route.enabled`을(를) `false`(으)로 설정하여 기본 큐 관리자 라우트의 작성을 방지할 수 있습니다.

알려진 문제점 및 제한사항

- 가용성 유형이 MultiInstance인 QueueManager을(를) 업데이트하는 경우, 두 팟(Pod) 모두 즉시 삭제됩니다. 이들은 모두 Red Hat OpenShift Container Platform에 의해 신속하게 재시작되어야 합니다.

IBM MQ Operator 1.3.8 (EUS)

EUS

IBM Cloud Pak for Integration 버전

IBM Cloud Pak for Integration 2020.4.1

Operator 채널

v1.3-eus

.spec.version에 허용되는 값

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus

Red Hat OpenShift Container Platform 버전

Red Hat OpenShift Container Platform 4.6 전용

IBM Cloud Pak foundational services 버전

IBM Cloud Pak foundational services 3.6(stable-v1 채널)

새로운 기능

- 새 피연산자 버전 9.2.0.6-r3-eus를 추가합니다.
- 주소 지정된 취약성은 이 [보안 게시판](#)에 자세히 설명되어 있습니다.

IBM MQ Operator 1.3.7 (EUS)

EUS

IBM Cloud Pak for Integration 버전

IBM Cloud Pak for Integration 2020.4.1

Operator 채널

v1.3-eus

.spec.version에 허용되는 값

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, [9.2.0.6-r2-eus](#)

Red Hat OpenShift Container Platform 버전

Red Hat OpenShift Container Platform 4.6 전용

IBM Cloud Pak foundational services 버전

IBM Cloud Pak foundational services 3.6(stable-v1 채널)

새로운 기능

- 새 피연산자 버전 [9.2.0.6-r2-eus](#)를 추가합니다.
- 주소 지정된 취약성은 이 [보안 게시판](#)에 자세히 설명되어 있습니다.

IBM MQ Operator 1.3.6 (EUS)

EUS

IBM Cloud Pak for Integration 버전

IBM Cloud Pak for Integration 2020.4.1

Operator 채널

v1.3-eus

.spec.version에 허용되는 값

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, [9.2.0.6-r1-eus](#)

Red Hat OpenShift Container Platform 버전

Red Hat OpenShift Container Platform 4.6 전용

IBM Cloud Pak foundational services 버전

IBM Cloud Pak foundational services 3.6(stable-v1 채널)

새로운 기능

- 새 피연산자 버전 [9.2.0.6-r1-eus](#)를 추가합니다.
- 주소 지정된 취약성은 이 [보안 게시판](#)에 자세히 설명되어 있습니다.

IBM MQ Operator 1.3.5 (EUS)

EUS

IBM Cloud Pak for Integration 버전

IBM Cloud Pak for Integration 2020.4.1

Operator 채널

v1.3-eus

.spec.version에 허용되는 값

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, [9.2.0.5-r3-eus](#)

Red Hat OpenShift Container Platform 버전

Red Hat OpenShift Container Platform 4.6 전용

IBM Cloud Pak foundational services 버전

IBM Cloud Pak foundational services 3.6(stable-v1 채널)

새로운 기능

- 새 피연산자 버전 [9.2.0.5-r3-eus](#)를 추가합니다.

- 주소 지정된 취약성은 이 [보안 게시판](#)에 자세히 설명되어 있습니다.

IBM MQ Operator 1.3.4 (EUS)

EUS

IBM Cloud Pak for Integration 버전

IBM Cloud Pak for Integration 2020.4.1

Operator 채널

v1.3-eus

.spec.version에 허용되는 값

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus

Red Hat OpenShift Container Platform 버전

Red Hat OpenShift Container Platform 4.6 전용

IBM Cloud Pak foundational services 버전

IBM Cloud Pak foundational services 3.6(stable-v1 채널)

새로운 기능

- 새 피연산자 버전 9.2.0.5-r2-eus 를 추가합니다.
- 주소 지정된 취약성은 이 [보안 게시판](#)에 자세히 설명되어 있습니다.

IBM MQ Operator 1.3.3 (EUS)

EUS

IBM Cloud Pak for Integration 버전

IBM Cloud Pak for Integration 2020.4.1

Operator 채널

v1.3-eus

.spec.version에 허용되는 값

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus

Red Hat OpenShift Container Platform 버전

Red Hat OpenShift Container Platform 4.6 전용

IBM Cloud Pak foundational services 버전

IBM Cloud Pak foundational services 3.6(stable-v1 채널)

새로운 기능

- 새 피연산자 버전 9.2.0.5-r1-eus 를 추가합니다.
- 주소 지정된 취약성은 이 [보안 게시판](#)에 자세히 설명되어 있습니다.

IBM MQ Operator 1.3.2(EUS)

EUS

IBM Cloud Pak for Integration 버전

IBM Cloud Pak for Integration 2020.4.1

Operator 채널

v1.3-eus

.spec.version에 허용되는 값

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.4-r1-eus

Red Hat OpenShift Container Platform 버전

Red Hat OpenShift Container Platform 4.6 전용

IBM Cloud Pak foundational services 버전

IBM Cloud Pak foundational services 3.6(stable-v1 채널)

새로운 기능

- 새 Operand 버전 [9.2.0.4-r1-eus](#)가 추가되었습니다.

IBM MQ Operator 1.3.1(EUS)

EUS

IBM Cloud Pak for Integration 버전

IBM Cloud Pak for Integration 2020.4.1

Operator 채널

v1.3-eus

.spec.version에 허용되는 값

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, 9.2.0.1-r1-eus, [9.2.0.2-r1-eus](#)

Red Hat OpenShift Container Platform 버전

Red Hat OpenShift Container Platform 4.6 전용

IBM Cloud Pak foundational services 버전

IBM Cloud Pak foundational services 3.6(stable-v1 채널)

새로운 기능

- 새 Operand 버전 [9.2.0.2-r1-eus](#)가 추가되었습니다.

IBM MQ Operator 1.3.0(EUS)

EUS

IBM Cloud Pak for Integration 버전

IBM Cloud Pak for Integration 2020.4.1

Operator 채널

v1.3-eus

.spec.version에 허용되는 값

9.1.5.0-r2, 9.2.0.0-r1, 9.2.0.0-r2, [9.2.0.1-r1-eus](#)

Red Hat OpenShift Container Platform 버전

Red Hat OpenShift Container Platform 4.6 전용

IBM Cloud Pak foundational services 버전

IBM Cloud Pak foundational services 3.6(stable-v1 채널)

새로운 기능

- IBM Cloud Pak for Integration 라이선스를 사용하는 경우 확장 업데이트 지원(EUS)이 -eus(으)로 끝나 는 .spec.version 필드에 대해 제공됩니다.
- .spec.labels 및 .spec.annotations을(를) 사용하여 QueueManager 자원에서 레이블 및 주석을 설정하는 새로운 방법을 추가합니다.

변경사항

- 단일 인스턴스에서 다중 인스턴스로 변경할 때의 오류 처리가 개선되었습니다.
- QueueManager 특성이 IBM Cloud Pak for Integration Platform Navigator 및 Red Hat OpenShift Container Platform 웹 콘솔의 "양식 보기"에서 렌더링되는 방법에 대한 개선사항
- IBM Cloud Pak for Integration 라이선스를 사용할 때 기본 라이선스 메트릭을 수정합니다 (VirtualProcessorCore).
- Red Hat OpenShift Container Platform 웹 콘솔에서 QueueManager의 자원 탭을 수정하며, 이제 해당 큐 관리자에 대해 IBM MQ Operator에서 관리하는 자원을 올바르게 표시합니다.

알려진 문제점 및 제한사항

- 가용성 유형이 MultiInstance인 QueueManager을(를) 업데이트하는 경우, 두 팟(Pod) 모두 즉시 삭제 됩니다. 이들은 모두 Red Hat OpenShift Container Platform에 의해 신속하게 재시작되어야 합니다.

IBM MQ Operator 1.2.0

CD

IBM Cloud Pak for Integration 버전

IBM Cloud Pak for Integration 2020.3.1

Operator 채널

v1.2

.spec.version에 허용되는 값

9.1.5.0-r2, 9.2.0.0-r1, [9.2.0.0-r2](#)

Red Hat OpenShift Container Platform 버전

Red Hat OpenShift Container Platform 4.4 이상

새로운 기능

- z/Linux에 대한 지원이 추가되었습니다.
- QueueManager 자원에 자세한 상태 조건을 추가합니다. 자세한 정보는 [137 페이지의 『QueueManager의 상태\(mq.ibm.com/v1beta1\)』](#)의 내용을 참조하십시오.
- 올바르게 않은 스토리지 클래스 사용을 방지하기 위해 런타임 검사가 추가되었습니다. 자세한 정보는 [108 페이지의 『런타임 웹훅 검사 사용 안함』](#)의 내용을 참조하십시오.
- 다중 인스턴스 큐 관리자에 대한 경험을 단순화합니다. 이제 QueueManager 자원에서 하나의 특성 (.spec.queueManager.availability.type)만으로 선택할 수 있습니다.
- QueueManager 자원에서 .spec.queueManager.storage.defaultClass 특성을 도입하여 비기본 스토리지 클래스 선택을 단순화합니다.

변경사항

- QueueManager 특성이 IBM Cloud Pak for Integration Platform Navigator 및 Red Hat OpenShift Container Platform 웹 콘솔의 "양식 보기"에서 렌더링되는 방법에 대한 개선사항
- 업그레이드된 큐 관리자 버전이 사용 가능한 경우 이제 해당 버전이 IBM Cloud Pak for Integration Platform Navigator에서 플래그 지정됩니다.

IBM MQ Operator 1.1.0

CD

IBM Cloud Pak for Integration 버전

IBM Cloud Pak for Integration 2020.2.1

Operator 채널

v1.1

.spec.version에 허용되는 값

9.1.5.0-r2, [9.2.0.0-r1](#)

Red Hat OpenShift Container Platform 버전

Red Hat OpenShift Container Platform 4.4 이상

새로운 기능

- IBM MQ Advanced 9.2.0이 Continuous Delivery 릴리스로 추가되었습니다.
- ConfigMap 또는 시크릿에 INI 및 MQSC 정보를 지정하는 기능이 추가되었습니다.
- Red Hat OpenShift Container Platform 웹 콘솔 사용 시 스키마 네비게이터를 사용할 수 있습니다.

변경사항

- Red Hat OpenShift on IBM Cloud®에 영향을 주는 네트워크 정책 관련 문제가 수정되었습니다.
- QueueManager 자원에서 유효하지 않은 설정 조합을 방지하기 위해 유효성 검증 웹 후크의 개선

IBM MQ Operator 1.0.0

CD

IBM Cloud Pak for Integration 버전

IBM Cloud Pak for Integration 2020.2.1

Operator 채널

v1.0

.spec.version에 허용되는 값

[9.1.5.0-r2](#)

Red Hat OpenShift Container Platform 버전

Red Hat OpenShift Container Platform 4.4 이상

새로운 기능

- [mq.ibm.com/v1beta1 API](#)를 소개하는 운영자의 초기 버전

IBM MQ Operator에 사용할 큐 관리자 컨테이너 이미지

9.2.5.0-r3

CD

필요한 Operator 버전

[1.8.2](#) 이상

지원되는 아키텍처

amd64, s390x

이미지

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.5.0-r3](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.2.5.0-r3](#)
- [icr.io/ibm-messaging/mq:9.2.5.0-r3](#)

새로운 기능

- [IBM MQ 9.2.5의 새로운 기능](#)

변경사항

- [IBM MQ 9.2.5의 변경된 기능](#)
- [Red Hat Universal Base 이미지 8.6-751](#) 기반

9.2.5.0-r2

CD

필요한 Operator 버전

[1.8.1](#) 이상

지원되는 아키텍처

amd64, s390x

이미지

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.5.0-r2](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.2.5.0-r2](#)
- [icr.io/ibm-messaging/mq:9.2.5.0-r2](#)

새로운 기능

- [IBM MQ 9.2.5의 새로운 기능](#)

변경사항

- [IBM MQ 9.2.5의 변경된 기능](#)
- [Red Hat Universal Base Image 8.5-240.1648458092](#) 에 기반함

9.2.5.0-r1



필요한 Operator 버전

[1.8.0](#) 이상

지원되는 아키텍처

amd64, s390x

이미지

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.5.0-r1](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.2.5.0-r1](#)
- [icr.io/ibm-messaging/mq:9.2.5.0-r1](#)

새로운 기능

- [IBM MQ 9.2.5의 새로운 기능](#)

변경사항

- [IBM MQ 9.2.5의 변경된 기능](#)
- 올바르지 않은 리모트 큐 관리자 옵션이 IBM MQ Console에서 제거되었습니다.
- [Red Hat Universal Base Image 8.5-240](#)에 기반하여

9.2.4.0-r1



필요한 Operator 버전

[1.7.0](#) 이상

지원되는 아키텍처

amd64, s390x

이미지

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.4.0-r1](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.2.4.0-r1](#)
- [docker.io/ibmcom/mq:9.2.4.0-r1](#)

새로운 기능

- [IBM MQ 9.2.4의 새로운 기능](#)

변경사항

- [IBM MQ 9.2.4의 변경된 기능](#)
- [Red Hat Universal Base Image 8.5-204](#)를 기반으로 합니다.

9.2.3.0-r1



필요한 Operator 버전

[1.6.0](#) 이상

지원되는 아키텍처

amd64, s390x

이미지

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.3.0-r1 (amd64 만 해당)
- cp.icr.io/cp/ibm-mqadvanced-server:9.2.3.0-r1
- docker.io/ibmcom/mq:9.2.3.0-r1

새로운 기능

- [IBM MQ 9.2.3의 새로운 기능](#)
- IBM Cloud Pak for Integration 라이선스와 함께 사용하는 경우 프로덕션용으로 MQ 원시 HA 를 지원합니다. IBM MQ 9.2.2가 있는 평가 라이선스에서 고유 HA를 사용하는 큐 관리자는 9.2.3으로 업그레이드할 수 없습니다. 평가 기간이 종료되었습니다.

변경사항

- [IBM MQ 9.2.3의 변경된 기능](#)
- [Red Hat Universal Base Image 8.4-205](#)를 기반으로 합니다.

9.2.2.0-r1



필요한 Operator 버전

1.5.0 이상

지원되는 아키텍처

amd64, s390x

이미지

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.2.0-r1 (amd64 만 해당)
- cp.icr.io/cp/ibm-mqadvanced-server:9.2.2.0-r1
- docker.io/ibmcom/mq:9.2.2.0-r1

새로운 기능

- [IBM MQ 9.2.2의 새로운 기능](#)
- IBM Cloud Pak for Integration 라이선스와 함께 사용하는 경우 평가 목적으로 MQ 원시 HA 지원

변경사항

- [IBM MQ 9.2.2의 변경된 기능](#)
- IBM MQ Advanced for Developers 큐 관리자 종료 시 FDC를 유발하는 수정된 문제점
- [Red Hat Universal Base Image 8.3-291](#)을 기반으로 합니다.

9.2.1.0-r2



필요한 Operator 버전

1.5.0 이상

지원되는 아키텍처

amd64, s390x

이미지

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.1.0-r2
- cp.icr.io/cp/ibm-mqadvanced-server:9.2.1.0-r2
- docker.io/ibmcom/mq:9.2.1.0-r2

변경사항

- IBM Cloud Pak foundational services 3.7 이상의 싱글 사인온에 대한 문제가 수정되었습니다.

- [Red Hat Universal Base Image 8.3-291](#)을 기반으로 합니다.

9.2.1.0-r1

CD

필요한 Operator 버전

1.4.0 이상

지원되는 아키텍처

amd64, s390x

이미지

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.1.0-r1](#)
- [cp.icr.io/cp/ibm-mqadvanced-server:9.2.1.0-r1](#)
- [docker.io/ibmcom/mq:9.2.1.0-r1](#)

새로운 기능

- [IBM MQ 9.2.1의 새로운 기능](#)
- MQ 웹 콘솔에서 기본 라우트의 연결 정보를 사용할 수 있습니다.

변경사항

- [IBM MQ 9.2.1의 변경된 기능](#)
- [Red Hat Universal Base Image 8.3-230](#)을 기반으로 합니다.

9.2.0.6-r3-eus

EUS

필요한 Operator 버전

1.3.8 및 향후 수정팩

지원되는 아키텍처

amd64, s390x

이미지

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.6-r3-eus](#)

변경사항

- IBM MQ 9.2.0 Fix Pack 6을(를) 포함합니다. For more information, see [IBM MQ 버전 9.2 LTS의 수정사항 목록](#).
- [Red Hat Universal Base Image 8.6-941](#)을 기반으로 합니다.

9.2.0.6-r2-eus

EUS

필요한 Operator 버전

1.3.7 및 향후 수정팩

지원되는 아키텍처

amd64, s390x

이미지

- [cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.6-r2-eus](#)

변경사항

- IBM MQ 9.2.0 Fix Pack 6을(를) 포함합니다. For more information, see [IBM MQ 버전 9.2 LTS의 수정사항 목록](#).
- [Red Hat Universal Base Image 8.6-902](#)에 기반합니다.

9.2.0.6-r1-eus

EUS

필요한 Operator 버전

[1.3.6](#) 및 이후의 수정팩

지원되는 아키텍처

amd64, s390x

이미지

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.6-r1-eus

변경사항

- IBM MQ 9.2.0 Fix Pack 6을(를) 포함합니다. For more information, see [IBM MQ 버전 9.2 LTS의 수정사항 목록](#).
- [Red Hat Universal Base Image 8.6-854](#)를 기반으로 합니다.

9.2.0.5-r3-eus

EUS

필요한 Operator 버전

[1.3.5](#) 및 이후 수정팩

지원되는 아키텍처

amd64, s390x

이미지

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.5-r3-eus

변경사항

- IBM MQ 9.2.0 Fix Pack 5을(를) 포함합니다. 자세한 정보는 [IBM MQ 9.2.0 Fix Pack 5에서 무엇이 변경되었는지 및 IBM MQ 버전 9.2 LTS의 수정사항 목록의 내용을 참조하십시오](#).
- [Red Hat Universal Base Image 8.6-751.1655117800](#)을 기반으로 합니다.

9.2.0.5-r2-eus

EUS

필요한 Operator 버전

[1.3.4](#) 및 이후 수정팩

지원되는 아키텍처

amd64, s390x

이미지

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.5-r2-eus

변경사항

- IBM MQ 9.2.0 Fix Pack 5을(를) 포함합니다. 자세한 정보는 [IBM MQ 9.2.0 Fix Pack 5에서 변경된 기능 및 Fix list for IBM MQ Version 9.2 LTS를 참조하십시오](#).
- [Red Hat Universal Base 이미지 8.6-751](#) 기반

9.2.0.5-r1-eus

EUS

필요한 Operator 버전

[1.3.3](#) 및 이후 수정팩

지원되는 아키텍처
amd64, s390x

이미지

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.5-r1-eus

변경사항

- IBM MQ 9.2.0 Fix Pack 5을(를) 포함합니다. 자세한 정보는 [IBM MQ 9.2.0 Fix Pack 5에서 변경된 기능 및 Fix list for IBM MQ Version 9.2 LTS](#)를 참조하십시오.
- [Red Hat Universal Base Image 8.5-240.1648458092](#)에 기반함

9.2.0.4-r1-eus

EUS

필요한 Operator 버전
1.3.2 및 이후 수정팩

지원되는 아키텍처
amd64, s390x

이미지

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.4-r1-eus

변경사항

- IBM MQ 9.2.0 Fix Pack 4을(를) 포함합니다. 자세한 정보는 [IBM MQ 9.2.0 Fix Pack 4에서 변경된 기능 및 Fix list for IBM MQ Version 9.2 LTS](#)를 참조하십시오.
- [Red Hat Universal Base Image 8.5-204](#)를 기반으로 합니다.

9.2.0.2-r2-eus

EUS

필요한 Operator 버전
1.6.0 이상

지원되는 아키텍처
amd64, s390x

이미지

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.2-r2-eus

변경사항

- EUS 릴리스에서 CD 릴리스로 마이그레이션하는 경우에만 필요한 IBM Cloud Pak foundational services 3.7 이상의 싱글 사인온에 대한 문제가 수정되었습니다.
- [Red Hat Universal Base Image 8.4-200.1622548483](#)을 기반으로 합니다.

9.2.0.2-r1-eus

EUS

필요한 Operator 버전
1.3.1 및 이후 수정팩, 1.6.0 이상

지원되는 아키텍처
amd64, s390x

이미지

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.2-r1-eus

변경사항

- 조작 대시보드 통합에서는 추적 에이전트 및 콜렉터 버전 1.0.8을 사용합니다.
- IBM MQ 9.2.0 Fix Pack 2을(를) 포함합니다. 자세한 정보는 [IBM MQ 9.2.0 Fix Pack 2에서 변경된 기능 및 Fix list for IBM MQ Version 9.2 LTS](#)를 참조하십시오.
- [Red Hat Universal Base Image 8.4-200.1622548483](#)을 기반으로 합니다.

9.2.0.1-r1-eus

EUS

필요한 Operator 버전

1.3.0 이상

지원되는 아키텍처

amd64, s390x

이미지

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.1-r1-eus`

새로운 기능

- IBM Cloud Pak for Integration 라이선스 사용 시에만 사용할 수 있습니다.
- Extended Update Support (EUS) is available when using IBM MQ Operator 1.3.x and IBM Common Services 3.6, on Red Hat OpenShift Container Platform 4.6

변경사항

- IBM MQ 9.2.0 Fix Pack 1을(를) 포함합니다. 자세한 정보는 [IBM MQ 9.2.0 Fix Pack 1에서 변경된 기능 및 Fix list for IBM MQ Version 9.2 LTS](#)를 참조하십시오.
- [Red Hat Universal Base Image 8.3-201](#)을 기반으로 합니다.
- 특권 에스컬레이션을 허용하는 `SecurityContextConstraints` 하에서 실행될 때 활동 프로브 (`chkmqhealthy`) 및 준비 프로브(`chkmqready`)를 수정합니다.

9.2.0.0-r3

CD

필요한 Operator 버전

1.5.0 이상

지원되는 아키텍처

amd64, s390x

이미지

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.0-r3`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.2.0.0-r3`
- `docker.io/ibmcom/mq:9.2.0.0-r3`

변경사항

- [Red Hat Universal Base Image 8.3-291](#)을 기반으로 합니다.

9.2.0.0-r2

CD

필요한 Operator 버전

1.2.0 이상

지원되는 아키텍처

amd64, s390x

이미지

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.0-r2
- cp.icr.io/cp/ibm-mqadvanced-server:9.2.0.0-r2
- docker.io/ibmcom/mq:9.2.0.0-r2

새로운 기능

- 이제 z/Linux에서 사용할 수 있습니다.

변경사항

- [Red Hat Universal Base Image 8.2-349](#)를 기반으로 합니다.

9.2.0.0-r1

CD

필요한 Operator 버전

[1.1.0](#) 이상

지원되는 아키텍처

amd64

이미지

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.0.0-r1-amd64
- cp.icr.io/cp/ibm-mqadvanced-server:9.2.0.0-r1-amd64
- docker.io/ibmcom/mq:9.2.0.0-r1

새로운 기능

- [IBM MQ 9.2.0의 새로운 기능](#)

변경사항

- [IBM MQ 9.2.0의 변경된 기능](#)
- -ic 인수를 crtmqm에 사용하여 MQSC 파일을 자동으로 적용합니다. runmqsc 명령의 이전 사용을 대체합니다.
- [Red Hat Universal Base Image 8.2-301.1593113563](#)을 기반으로 합니다.

9.1.5.0-r2

CD

필요한 Operator 버전

[1.0.0](#) 이상

지원되는 아키텍처

amd64

이미지

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.1.5.0-r2-amd64
- cp.icr.io/cp/ibm-mqadvanced-server:9.1.5.0-r2-amd64
- docker.io/ibmcom/mq:9.1.5.0-r2

변경사항

- [Red Hat Universal Base Image 8.2-267](#)을 기반으로 합니다.

Integration으로 마이그레이션

이 주제에서는 IBM Cloud Pak for Integration의 IBM MQ Operator 를 사용하여 기존 IBM MQ 큐 관리자를 컨테이너 환경으로 마이그레이션하기 위한 주요 단계에 대해 설명합니다.

이 태스크 정보

Red Hat OpenShift에 IBM MQ를 배치하는 클라이언트는 다음 시나리오로 구분할 수 있습니다.

1. 새 애플리케이션을 위해 Red Hat OpenShift에서 새 IBM MQ 배치 작성
2. Red Hat OpenShift에서 새 애플리케이션을 위해 IBM MQ 네트워크를 Red Hat OpenShift로 확장
3. 기존 애플리케이션을 계속 지원하기 위해 IBM MQ 배치를 Red Hat OpenShift로 이동

IBM MQ 구성을 마이그레이션해야 하는 시나리오 3에만 해당합니다. 기타 시나리오는 새 배치를 고려합니다.

이 주제 세트는 시나리오 3에 초점을 맞추고 IBM MQ Operator를 사용하여 기존 IBM MQ 큐 관리자를 컨테이너 환경으로 마이그레이션하기 위한 주요 단계를 설명합니다. IBM MQ는 유연하고 광범위하게 사용되므로 여러 가지 선택적 단계가 있습니다. 각각의 단계에는 "수행 필요성" 절이 있습니다. 마이그레이션 도중 사용자 요구사항을 확인하여 시간을 절약해야 합니다.

또한 마이그레이션해야 하는 데이터도 고려해야 합니다.

1. 구성은 동일하지만 기존 큐잉 메시지 없이 IBM MQ를 마이그레이션합니다.
2. 동일한 구성으로 기존 메시지와 함께 IBM MQ를 마이그레이션합니다.

일반적인 버전간 마이그레이션에서는 두 방법 중 하나를 사용할 수 있습니다. 마이그레이션 시 일반 IBM MQ 큐 관리자에서는 큐에 저장된 메시지가 적으므로 많은 경우에 옵션 1이 적합하게 됩니다. 컨테이너 플랫폼으로의 마이그레이션인 경우 옵션 1을 사용하는 것이 마이그레이션의 복잡도를 줄이고 블루 그린 배치가 허용되므로 더 일반적입니다. 그러므로 설명은 이 시나리오에 초점을 맞춥니다.

이 시나리오의 목적은 기존 큐 관리자의 정의와 일치하는 컨테이너 환경에서 큐 관리자를 작성하는 것입니다. 그러면 다른 구성 또는 애플리케이션 논리를 변경하지 않고도 새 큐 관리자를 가리키도록 기존의 네트워크 연결 애플리케이션을 간단히 재구성할 수 있습니다.

이 마이그레이션 전체에서 새 큐 관리자에 적용할 여러 구성 파일을 생성합니다. 이러한 파일의 관리를 단순화하기 위해 디렉토리를 작성하고 해당 디렉토리 내부에서 구성 파일을 생성해야 합니다.

프로시저

1. [35 페이지의 『필수 기능이 사용 가능한지 확인』](#)
2. [36 페이지의 『큐 관리자 구성 추출』](#)
3. 옵션: [37 페이지의 『선택사항: 큐 관리자 키 및 인증서 추출 및 확보』](#)
4. 옵션: [38 페이지의 『선택사항: LDAP 구성』](#)
5. 옵션: [45 페이지의 『선택사항: IBM MQ 구성에서 IP 주소 및 호스트 이름 변경』](#)
6. [47 페이지의 『컨테이너 환경을 위한 큐 관리자 구성 업데이트』](#)
7. [49 페이지의 『컨테이너에서 실행 중인 IBM MQ의 대상 HA 아키텍처 선택』](#)
8. [50 페이지의 『큐 관리자의 자원 작성』](#)
9. [51 페이지의 『Red Hat OpenShift에서 새 큐 관리자 작성』](#)
10. [55 페이지의 『새 컨테이너 배치 확인』](#)

IBM MQ Operator에 IBM MQ Advanced에서 사용 가능한 모든 기능이 포함되는 것은 아니므로 이러한 기능이 필요하지 않은지 확인해야 합니다. 기타 기능은 부분적으로 지원되므로 컨테이너에서 사용 가능한 내용과 일치하도록 재구성할 수 있습니다.

시작하기 전에

이는 35 페이지의 『IBM MQ를 IBM Cloud Pak for Integration으로 마이그레이션』에서의 첫 번째 단계입니다.

프로시저

1. 대상 컨테이너 이미지에 필요한 모든 기능이 포함되는지 확인하십시오.
최신 정보는 5 페이지의 『컨테이너에서 IBM MQ를 사용하는 방법 선택』의 내용을 참조하십시오.
2. IBM MQ Operator에는 리스너라는 단일 IBM MQ 트래픽 포트가 있습니다. 여러 개의 리스너가 있는 경우 컨테이너에서 단일 리스너를 사용하도록 이를 단순화하십시오. 이것이 공통 시나리오가 아니므로 이러한 수정 사항에 대해서는 자세히 설명하지 않습니다.
3. IBM MQ 엑시트가 사용되는 경우 IBM MQ 엑시트 바이너리에서 계층화하여 컨테이너로 마이그레이션하십시오. 이는 고급 마이그레이션 시나리오이므로 여기에는 포함하고 있지 않습니다. 단계에 대한 개요는 106 페이지의 『Red Hat OpenShift CLI를 사용하여 사용자 정의 MQSC 및 INI 파일이 포함된 이미지 빌드』의 내용을 참조하십시오.
4. IBM MQ 시스템에 고가용성이 포함되는 경우, 사용 가능한 옵션을 검토하십시오.
15 페이지의 『컨테이너에서 IBM MQ에 대한 고가용성』의 내용을 참조하십시오.

다음에 수행할 작업

이제 큐 관리자 구성을 추출할 준비가 되었습니다.

V 9.2.1 OpenShift CD EUS 큐 관리자 구성 추출

대다수의 구성은 큐 관리자 사이에서 이동 가능합니다. 예를 들어 큐, 토픽 및 채널 정의와 같이 애플리케이션이 상호 작용하는 대상입니다. 기존 IBM MQ 큐 관리자에서 구성을 추출하려면 이 태스크를 사용하십시오.

시작하기 전에

이 태스크에서는 필수 기능이 사용 가능한지 검사한 것으로 간주합니다.

프로시저

1. 기존 IBM MQ 설치가 있는 시스템에 로그인합니다.
2. 구성을 백업합니다.

다음 명령을 실행하십시오.

```
dmpmqcfg -m QMGR_NAME > /tmp/backup.mqsc
```

이 명령에 대한 사용법 참고사항입니다.

- 이 명령은 백업을 tmp 디렉토리에 저장합니다. 백업을 다른 위치에 저장할 수 있지만 이 시나리오에서는 후속 명령의 경우 tmp 디렉토리로 간주합니다.
- 사용자 환경의 큐 관리자 이름으로 QMGR_NAME을 바꾸십시오. 값이 확실하지 않으면 dspmq 명령을 실행하여 시스템에서 사용 가능한 큐 관리자를 확인하십시오. 다음은 큐 관리자 qm1의 샘플 dspmq 명령입니다.

```
QMNAME(qm1) STATUS(Running)
```

dspmq 명령에서는 IBM MQ 큐 관리자를 시작해야 합니다. 그렇지 않으면 다음 오류가 수신됩니다.

```
AMQ8146E: IBM MQ queue manager not available.
```

필요한 경우 다음 명령을 실행하여 큐 관리자를 시작하십시오.

```
strmqm QMGR_NAME
```

다음에 수행할 작업

이제는 큐 관리자 키 및 인증서를 추출하여 확보할 준비가 되었습니다.

V 9.2.1 OpenShift CD EUS 선택사항: 큐 관리자 키 및 인증서 추출 및 확보

IBM MQ는 큐 관리자에 대한 트래픽을 암호화하기 위해 TLS를 사용하여 구성할 수 있습니다. 이 태스크는 큐 관리자가 TLS를 사용하는지 확인하고, 키 및 인증서를 추출하고, 마이그레이션된 큐 관리자에서 TLS를 구성하는데 사용됩니다.

시작하기 전에

이 태스크에서는 사용자가 [큐 관리자 구성을 추출했다](#)고 간주합니다.

이 태스크 정보

수행 필요성

IBM MQ는 큐 관리자에 대한 트래픽을 암호화하도록 구성할 수 있습니다. 이 암호화는 큐 관리자에서 구성된 키 저장소를 사용하여 완료됩니다. 그러면 IBM MQ 채널에서 TLS 통신을 사용할 수 있습니다. 사용자 환경에서 구성되었는지 확실하지 않으면 다음 명령을 실행하여 확인하십시오.

```
grep 'SECCOMM(ALL\|SECCOMM(ANON\|SSLCIPH)' backup.mqsc
```

결과를 찾을 수 없으면 TLS가 사용되고 있지 않은 것입니다. 그러나 찾을 수 없다고 해서 마이그레이션된 큐 관리자에서 TLS를 구성해서는 안된다는 의미는 아닙니다. 이 작동을 변경하려는 여러 이유가 있습니다.

- Red Hat OpenShift 환경에 대한 보안 접근 방식이 이전 환경과 비교하여 개선되어야 합니다.
- Red Hat OpenShift 환경 외부에서 마이그레이션된 큐 관리자에 액세스해야 하는 경우 Red Hat OpenShift Route를 통과하려면 TLS가 필요합니다.

프로시저

1. 기존 저장소에서 신뢰할 수 있는 인증서를 추출합니다.

현재 큐 관리자에서 TLS를 사용 중인 경우, 큐 관리자에서는 신뢰할 수 있는 여러 인증서가 저장되어 있을 수 있습니다. 이들을 추출하여 새 큐 관리자로 복사해야 합니다. 다음 선택적 단계 중 하나를 완료하십시오.

- 인증서 추출이 원활하게 수행되도록 로컬 시스템에서 다음 스크립트를 실행하십시오.

```
#!/bin/bash

keyr=$(grep SSLKEYR $1)
if [ -n "${keyr}" ]; then
    keyrlocation=$(sed -n "s/^\.*\(.*\)'.*$/\1/ p" <<< ${keyr})
    mapfile -t runmqckmResult < <(runmqckm -cert -list -db ${keyrlocation}.kdb -stashed)
    cert=1
    for i in "${runmqckmResult[@]:1}"
    do
        certlabel=$(echo ${i} | xargs)
        echo Extracting certificate $certlabel to $cert.cert
        runmqckm -cert -extract -db ${keyrlocation}.kdb -label "$certlabel" -target $
    {cert}.cert -stashed
    cert=${cert+1}
    done
fi
```

스크립트 실행 시 인수로서 IBM MQ 백업의 위치를 지정하면 인증서가 추출됩니다. 예를 들어 스크립트가 `extractCert.sh`이고 IBM MQ 백업이 `/tmp/backup.mqsc`에 있는 경우 다음 명령을 실행하십시오.

```
extractCert.sh /tmp/backup.mqsc
```

- 또는 표시된 순서 대로 다음 명령을 실행하십시오.

a. TLS 저장소의 위치를 식별하십시오.

```
grep SSLKEYR /tmp/backup.mqsc
```

샘플 출력:

```
SSLKEYR('/run/runmqserver/tls/key') +
```

여기서 키 저장소는 /run/runmqserver/tls/key.kdb에 있습니다.

b. 이 위치 정보를 기반으로 키 저장소를 조회하여 저장되는 인증서를 판별하십시오.

```
runmqckm -cert -list -db /run/runmqserver/tls/key.kdb -stashed
```

샘플 출력:

```
Certificates in database /run/runmqserver/tls/key.kdb:
  default
  CN=cs-ca-certificate,0=cert-manager
```

c. 나열되는 각 인증서를 추출하십시오. 다음 명령을 실행하여 이를 수행하십시오.

```
runmqckm -cert -extract -db KEYSTORE_LOCATION -label "LABEL_NAME" -target OUTPUT_FILE -stashed
```

이전에 표시된 샘플에서 이는 다음과 동일합니다.

```
runmqckm -cert -extract -db /run/runmqserver/tls/key.kdb -label "CN=cs-ca-certificate,0=cert-manager" -target /tmp/cert-manager.crt -stashed
runmqckm -cert -extract -db /run/runmqserver/tls/key.kdb -label "default" -target /tmp/default.crt -stashed
```

2. 큐 관리자의 새 키 및 인증서를 확보합니다.

마이그레이션된 큐 관리자에서 TLS를 구성하려면 새 키와 인증서를 생성하십시오. 이는 배치 도중 사용됩니다. 많은 조직에서 이는 보안 팀에 문의하여 키와 인증서를 요청함을 의미합니다. 일부 조직에서는 이 옵션을 사용할 수 없으므로 자체 서명 인증서가 사용됩니다.

다음 예제에서는 만기가 10년으로 설정되는 자체 서명 인증서를 생성합니다.

```
openssl req \
  -newkey rsa:2048 -nodes -keyout qmgr.key \
  -subj "/CN=mq queuemanager/OU=ibm mq" \
  -x509 -days 3650 -out qmgr.crt
```

다음과 같은 두 개의 새 파일이 작성됩니다.

- qmgr.key는 큐 관리자의 개인 키입니다.
- qmgr.crt는 공용 인증서입니다.

다음에 수행할 작업

이제 LDAP을 구성할 준비가 되었습니다.

V 9.2.1 OpenShift CD EUS 선택사항: LDAP 구성

IBM MQ Operator는 여러 다른 보안 접근법을 사용하도록 구성될 수 있습니다. 일반적으로 엔터프라이즈 배치의 경우 LDAP이 가장 효과적이므로 LDAP이 이 마이그레이션 시나리오에 사용됩니다.

시작하기 전에

이 태스크에서는 사용자가 큐 관리자 키 및 인증서를 추출하고 확보한 것으로 간주합니다.

이 태스크 정보

수행 필요성

이미 인증 및 권한 부여를 위해 LDAP을 사용 중인 경우 변경할 필요가 없습니다.

LDAP이 사용되고 있는지 확실하지 않으면, 다음 명령을 실행하십시오.

```
connauthname="$(grep CONNAUTH backup.mqsc | cut -d "(" -f2 | cut -d ")" -f1)"; grep -A 20 AUTHINFO(\$connauthname\) backup.mqsc
```

샘플 출력:

```
DEFINE AUTHINFO('USE.LDAP') +
  AUTHTYPE(IDPWLDAP) +
  ADOPTCTX(YES) +
  CONNAME('ldap-service.ldap(389)') +
  CHCKCLNT(REQUIRED) +
  CLASSGRP('groupOfUniqueNames') +
  FINDGRP('uniqueMember') +
  BASEDNG('ou=groups,dc=ibm,dc=com') +
  BASEDNU('ou=people,dc=ibm,dc=com') +
  LDAPUSER('cn=admin,dc=ibm,dc=com') +
  * LDAPPWD('*****') +
  SHORTUSR('uid') +
  GRPFIELD('cn') +
  USRFIELD('uid') +
  AUTHORMD(SEARCHGRP) +
  * ALTDAT(2020-11-26) +
  * ALTTIME(15.44.38) +
  REPLACE
```

출력에 있는 두 가지 속성에 특히 관심이 있습니다.

AUTHTYPE

여기에 IDPWLDAP 값이 있으면 인증에 LDAP을 사용하는 것입니다.

값이 공백이거나 다른 값이면 LDAP이 구성되지 않은 것입니다. 이 경우 AUTHORMD 속성을 검사하여 LDAP 사용자가 권한 부여에 사용되고 있는지 확인하십시오.

AUTHORMD

여기에 OS 값이 있으면 권한 부여에 LDAP을 사용하고 있지 않은 것입니다.

LDAP을 사용하도록 권한 부여 및 인증을 수정하려면 다음 태스크를 완료하십시오.

프로시저

1. LDAP 서버용으로 IBM MQ 백업을 업데이트합니다.
2. LDAP 권한 부여 정보용으로 IBM MQ 백업을 업데이트합니다.

V 9.2.1 **OpenShift** **CD** **EUS** **LDAP 파트 1: LDAP 서버의 IBM MQ 백업 업**

데이트

LDAP 설정 방법에 대한 포괄적인 설명은 이 시나리오의 범위를 벗어납니다. 이 토픽에서는 프로세스 요약, 샘플 및 추가 정보에 대한 참조를 제공합니다.

시작하기 전에

이 태스크에서는 사용자가 큐 관리자 키 및 인증서를 추출하고 확보한 것으로 간주합니다.

이 태스크 정보

수행 필요성

이미 인증 및 권한 부여를 위해 LDAP을 사용 중인 경우 변경할 필요가 없습니다. LDAP이 사용 중인지 확실하지 않으면 [38 페이지의 『선택사항: LDAP 구성』](#)의 내용을 참조하십시오.

LDAP 서버 설정에는 두 파트가 있습니다.

1. LDAP 구성 정의.
2. 큐 관리자 정의와 LDAP 구성 연관시키기.

이 구성에 도움이 되는 추가 정보는 다음과 같습니다.

- [사용자 저장소 개요](#)
- [AUTHINFO 명령에 대한 참조 안내서](#)

프로시저

1. LDAP 구성을 정의합니다.

LDAP 시스템에 대한 새 **AUTHINFO** 오브젝트를 정의하도록 backup.mqsc 파일을 편집하십시오. 예를 들면, 다음과 같습니다.

```
DEFINE AUTHINFO(USE.LDAP) +
  AUTHTYPE(IDPWLDAP) +
  CONNAME('ldap-service.ldap(389)') +
  LDAPUSER('cn=admin,dc=ibm,dc=com') +
  LDAPPWD('admin') +
  SECCOMM(NO) +
  USRFIELD('uid') +
  SHORTUSR('uid') +
  BASEDNU('ou=people,dc=ibm,dc=com') +
  AUTHORMD(SEARCHGRP) +
  BASEDNG('ou=groups,dc=ibm,dc=com') +
  GRPFIELD('cn') +
  CLASSGRP('groupOfUniqueNames') +
  FINDGRP('uniqueMember')
REPLACE
```

여기서,

- **CONNAME**는 LDAP 서버에 해당하는 호스트 이름 및 포트입니다. 탄성을 위해 여러 주소가 존재하는 경우 이들은 쉼표로 구분되는 목록을 사용하여 구성할 수 있습니다.
- **LDAPUSER**는 LDAP에 연결하여 사용자 레코드를 조회하는 경우 IBM MQ에서 사용하는 사용자에게 해당하는 식별 이름입니다.
- **LDAPPWD**는 **LDAPUSER** 사용자에게 해당하는 비밀번호입니다.
- **SECCOM**은 LDAP 서버에 대한 통신에서 TLS를 반드시 사용하는지 여부를 지정합니다. 가능한 값:
 - YES: TLS가 사용되고 인증서는 IBM MQ 서버가 제공합니다.
 - ANON: IBM MQ 서버에서 제공 중인 인증서 없이 TLS가 사용됩니다.
 - NO: 연결 도중 TLS가 사용되지 않습니다.
- **USRFIELD**는 제공된 사용자 이름이 일치하는 LDAP 레코드 내의 필드를 지정합니다.
- **SHORTUSR**는 길이가 12자를 초과하지 않는 LDAP 레코드 내의 필드입니다. 이 필드 내의 값은 인증이 성공하면 인증된 ID가 됩니다.
- **BASEDNU**는 LDAP 검색에 반드시 사용해야 하는 기본 DN입니다.
- **BASEDNG**는 LDAP 내에서 그룹의 기본 DN입니다.
- **AUTHORMD**는 사용자의 그룹 멤버십을 해결하기 위해 사용하는 메커니즘을 정의합니다. 네 개의 옵션이 있습니다.
 - OS: 축약형 이름과 연관되는 그룹의 운영 체제를 조회합니다.
 - SEARCHGRP: 인증된 사용자에게 대한 그룹 항목을 LDAP에서 검색합니다.
 - SEARCHUSR: 인증된 사용자 레코드에서 그룹 멤버십 정보를 검색합니다.
 - SRCHGRPSN: LDAP의 그룹 항목에서 인증된 사용자 축약 이름(SHORTUSR 필드에 의해 정의됨)을 검색합니다.

- **GRPFIELD**는 단순 이름에 해당하는 LDAP 그룹 레코드 내에서의 속성입니다. 속성이 지정되는 경우 이는 권한 부여 레코드를 정의하는 데 사용될 수 있습니다.
- **CLASSUSR**는 사용자에게 해당하는 LDAP 오브젝트 클래스입니다.
- **CLASSGRP**는 그룹에 해당하는 LDAP 오브젝트 클래스입니다.
- **FINDGRP**는 그룹 멤버십에 해당하는 LDAP 레코드 내의 속성입니다.

파일 내의 임의 위치에 새 항목을 배치할 수 있지만 새 항목을 파일의 처음에 배치하는 것이 유용하다는 것을 알 수 있습니다.

```

Open [ ]
backup.mqsc
*****
* Script generated on 2020-10-21 at 11.48.32
* Script generated by user ' CallumJackso' on host 'LAPTOP-VL(
* Queue manager name: qm1
* Queue manager platform: Windows
* Queue manager command level: (920/920)
* Command issued: dmpmqcfg -m qm1
*****
DEFINE AUTHINFO(USE.LDAP) +
  AUTHTYPE(IDPWLDAP) +
  CONNAME('ldap-service.ldap(389)') +
  LDAPUSER('cn=admin,dc=ibm,dc=com') +
  LDAPPWD('admin') +
  SECCOMM(NO) +
  USRFIELD('uid') +
  SHORTUSR('uid') +
  BASEDNU('ou=people,dc=ibm,dc=com') +
  AUTHORMD(SEARCHGRP) +
  BASEDNG('ou=groups,dc=ibm,dc=com') +
  GRPFIELD('cn') +
  CLASSGRP('groupOfUniqueNames') +
  FINDGRP('uniqueMember') +
  REPLACE
ALTER QMGR +
* ALTDATA(2020-10-26) +
* ALTTIME(11.43.11) +

```

2. 큐 관리자 정의와 LDAP 구성을 연관시킵니다.

LDAP 구성을 큐 관리자 정의와 연관시켜야 합니다. DEFINE AUTHINFO 항목 바로 아래는 ALTER QMGR 항목입니다. 새로 작성된 AUTHINFO 이름에 해당하도록 CONNAUTH 항목을 수정하십시오. 예를 들어 이전 예제에서는 AUTHINFO(USE.LDAP)가 정의되었고 이는 이름이 USE.LDAP임을 의미합니다. 그러므로 CONNAUTH('SYSTEM.DEFAULT.AUTHINFO.IDPWOS')를 CONNAUTH('USE.LDAP')로 변경하십시오.

```

Open ▾ [icon]
backup.mqsc
*****
* Script generated on 2020-10-21 at 11.48.32
* Script generated by user ' CallumJackso' on host 'l
* Queue manager name: qm1
* Queue manager platform: Windows
* Queue manager command level: (920/920)
* Command issued: dmpmqcfg -m qm1
*****
DEFINE AUTHINFO(USE.LDAP) +
  AUTHTYPE(IDPWLDAP) +
  CONNAME('ldap-service.ldap(389)') +
  LDAPUSER('cn=admin,dc=ibm,dc=com') +
  LDAPPWD('admin') +
  SECCOMM(NO) +
  USRFIELD('uid') +
  SHORTUSR('uid') +
  BASEDNU('ou=people,dc=ibm,dc=com') +
  AUTHORMD(SEARCHGRP) +
  BASEDNG('ou=groups,dc=ibm,dc=com') +
  GRPFIELD('cn') +
  CLASSGRP('groupOfUniqueNames') +
  FINDGRP('uniqueMember') +
  REPLACE
ALTER QMGR +
* ALTDATE(2020-10-26) +
* ALTTIME(11.43.11) +
  CCSID(850) +
  CERTLABL('default') +
  CLWLUSEQ(LOCAL) +
* COMMANDQ(SYSTEM_ADMIN_COMMAND.QUEUE) +
  CONNAUTH('USE.LDAP') +

```

LDAP에 대한 전환이 즉시 발생하도록 하려면 ALTER QMGR 명령 바로 다음에 라인을 추가하여 REFRESH SECURITY 명령을 호출하십시오.

```

*backup.mqsc
*****
* Script generated on 2020-10-21 at 11.48.32
* Script generated by user ' CallumJackso' on host 'LAPTOP-VLQKJ5UH'
* Queue manager name: qm1
* Queue manager platform: Windows
* Queue manager command level: (920/920)
* Command issued: dmpmqcfc -m qm1
*****
DEFINE AUTHINFO(USE.LDAP) +
  AUTHTYPE(IDPWLDAP) +
  CONNAME('ldap-service.ldap(389)') +
  LDAPUSER('cn=admin,dc=ibm,dc=com') +
  LDAPPWD('admin') +
  SECCOMM(NO) +
  USRFIELD('uid') +
  SHORTUSR('uid') +
  BASEDNU('ou=people,dc=ibm,dc=com') +
  AUTHORMD(SEARCHGRP) +
  BASEDNG('ou=groups,dc=ibm,dc=com') +
  GRPFIELD('cn') +
  CLASSGRP('groupOfUniqueNames') +
  FINDGRP('uniqueMember') +
  REPLACE
ALTER QMGR +
* ALTDATA(2020-10-26) +
* ALTTIME(11.43.11) +
  CCSID(850) +
  CERTLABL('default') +
  CLWLUSEQ(LOCAL) +
* COMMANDQ(SYSTEM.ADMIN.COMMAND.QUEUE) +
  CONNAUTH('USE.LDAP') +
* CRDATE(2020-10-26) +
* CRTIME(11.43.11) +
* QMID(qm1_2020-10-26_11.43.11) +
  SSLCRYP(' ') +
  SSLKEYR('/run/runmqserver/tls/key') +
  SUITEB(NONE) +
* VERSION(09020000) +
  FORCE
REFRESH SECURITY

```

다음에 수행할 작업

이제 LDAP 권한 부여 정보를 위해 IBM MQ 백업을 업데이트할 준비가 되었습니다.

IBM MQ에서는 IBM MQ 오브젝트에 대한 액세스를 제어하는 정제된 권한 부여 규칙을 제공합니다. LDAP에 대한 인증 및 권한 부여를 변경한 경우 권한 부여 규칙이 무효로 되므로 업데이트가 필요할 수 있습니다.

시작하기 전에

이 태스크에서는 사용자가 LDAP 서버에 대한 백업을 업데이트했다고 간주합니다.

이 태스크 정보

수행 필요성

이미 인증 및 권한 부여를 위해 LDAP을 사용 중인 경우 변경할 필요가 없습니다. LDAP이 사용 중인지 확실하지 않으면 38 페이지의 『선택사항: LDAP 구성』의 내용을 참조하십시오.

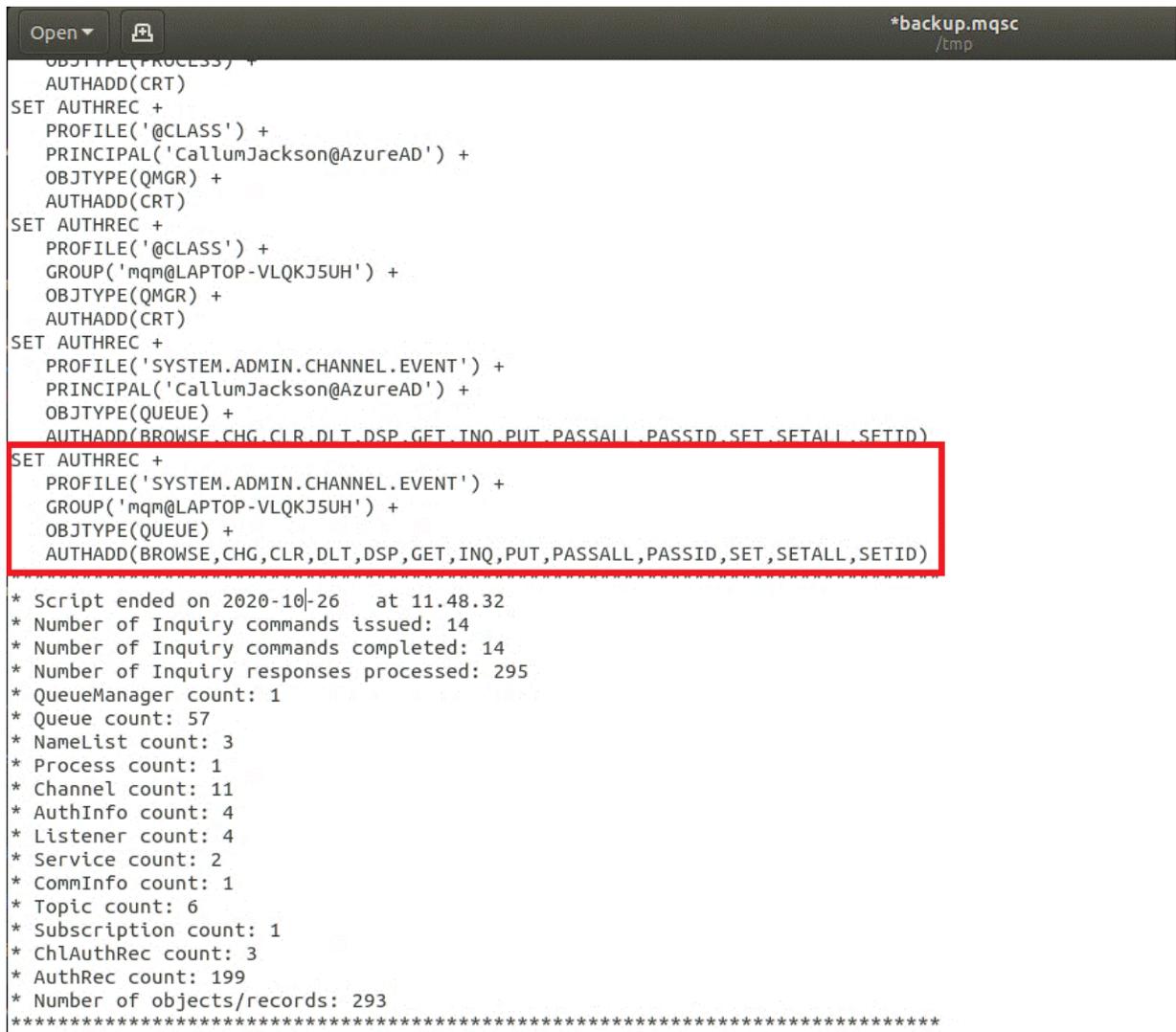
LDAP 권한 부여 정보 업데이트에는 두 파트가 있습니다.

1. 파일에서 기존의 모든 권한 부여 제거.
2. LDAP의 새 권한 부여 정보 정의.

프로시저

1. 파일에서 기존의 모든 권한 부여를 제거합니다.

백업 파일에서는 파일의 거의 맨 끝에 SET AUTHREC로 시작하는 여러 항목이 표시되어야 합니다.



```
Open [icon] *backup.mqsc /tmp
OBJTYPE(PROCESS) +
AUTHADD(CRT)
SET AUTHREC +
PROFILE('@CLASS') +
PRINCIPAL('CallumJackson@AzureAD') +
OBJTYPE(QMGR) +
AUTHADD(CRT)
SET AUTHREC +
PROFILE('@CLASS') +
GROUP('mqm@LAPTOP-VLQKJ5UH') +
OBJTYPE(QMGR) +
AUTHADD(CRT)
SET AUTHREC +
PROFILE('SYSTEM.ADMIN.CHANNEL.EVENT') +
PRINCIPAL('CallumJackson@AzureAD') +
OBJTYPE(Queue) +
AUTHADD(BROWSE,CHG,CLR,DLT,DSP,GET,INQ,PUT,PASSALL,PASSID,SET,SETALL,SETID)
SET AUTHREC +
PROFILE('SYSTEM.ADMIN.CHANNEL.EVENT') +
GROUP('mqm@LAPTOP-VLQKJ5UH') +
OBJTYPE(Queue) +
AUTHADD(BROWSE,CHG,CLR,DLT,DSP,GET,INQ,PUT,PASSALL,PASSID,SET,SETALL,SETID)

* Script ended on 2020-10-26 at 11.48.32
* Number of Inquiry commands issued: 14
* Number of Inquiry commands completed: 14
* Number of Inquiry responses processed: 295
* QueueManager count: 1
* Queue count: 57
* NameList count: 3
* Process count: 1
* Channel count: 11
* AuthInfo count: 4
* Listener count: 4
* Service count: 2
* CommInfo count: 1
* Topic count: 6
* Subscription count: 1
* ChlAuthRec count: 3
* AuthRec count: 199
* Number of objects/records: 293
*****
```

기존 항목을 찾아 삭제하십시오. 가장 직접적인 방법은 기존의 모든 SET AUTHREC 규칙을 제거하고 LDAP 항목을 기반으로 새 항목을 작성하는 것입니다.

2. LDAP의 새 권한 부여 정보를 정의합니다.

사용자의 큐 관리자 구성과 자원 및 그룹의 수에 따라 이는 상당한 시간이 소요되거나 직접적인 활동이 될 수 있습니다. 다음 예제에서는 사용자의 큐 관리자에 Q1이라는 단일 큐가 있고 사용자는 LDAP 그룹 apps에서 액세스할 수 있도록 허용하는 것으로 간주합니다.

```
SET AUTHREC GROUP('apps') OBJTYPE(QMGR) AUTHADD(ALL)
SET AUTHREC PROFILE('Q1') GROUP('apps') OBJTYPE(Queue) AUTHADD(ALL)
```

첫 번째 AUTHREC 명령은 큐 관리자에 액세스하기 위한 권한을 추가하고, 두 번째는 큐에 대한 액세스를 제공합니다. 두 번째 큐에 대한 액세스가 필요한 경우 와일드카드를 사용하여 더 일반적인 액세스를 제공하도록 결정하는 경우가 아니면 세 번째 AUTHREC 명령이 필요합니다.

다음은 또 다른 예제입니다. 관리자 그룹(admins)에 큐 관리자에 대한 전체 액세스가 필요한 경우 다음 명령을 추가하십시오.

```
SET AUTHREC PROFILE('*') OBJTYPE(Queue) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(Topic) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(Channel) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(ClntConn) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(AuthInfo) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(Listener) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(NameList) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(Process) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(Service) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(QMGR) GROUP('admins') AUTHADD(ALL)
```

다음에 수행할 작업

이제는 IBM MQ 구성에서 IP 주소 및 호스트 이름을 변경할 준비가 되었습니다.

V 9.2.1 OpenShift CD EUS 선택사항: IBM MQ 구성에서 IP 주소 및 호스트 이름 변경

IBM MQ 구성에 IP 주소 및 호스트 이름이 지정되어 있을 수 있습니다. 일부 상황에서는 이 지정이 유지될 수 있지만 기타 상황에서는 업데이트해야 합니다.

시작하기 전에

이 태스크에서는 사용자가 LDAP을 구성했다고 간주합니다.

이 태스크 정보

수행 필요성

먼저 이전 절에서 정의한 LDAP 구성과 별도로 IP 주소 또는 호스트 이름이 지정되어 있는지 판별하십시오. 이를 위해서는 다음 명령을 실행하십시오.

```
grep 'CONNAME\|LOCLADDR\|IPADDRV' -B 3 backup.mqsc
```

샘플 출력:

```
*****
DEFINE AUTHINFO(USE.LDAP) +
  AUTHTYPE(IDPWLDAP) +
  CONNAME('ldap-service.ldap(389)') +
--
DEFINE AUTHINFO('SYSTEM.DEFAULT.AUTHINFO.IDPWLDAP') +
  AUTHTYPE(IDPWLDAP) +
  ADOPTCTX(YES) +
  CONNAME(' ') +
--
REPLACE
DEFINE AUTHINFO('SYSTEM.DEFAULT.AUTHINFO.CRLLDAP') +
```

```
AUThTYpE(CRLLDAP) +  
CONNAME(' ') +
```

이 예제에서는 검색에서 세 가지 결과를 리턴합니다. 한 가지 결과는 이전에 정의한 LDAP 구성에 해당됩니다. LDAP 서버의 호스트 이름은 동일하게 유지되므로 이는 무시할 수 있습니다. 다른 두 개의 결과는 빈 연결 항목이므로 이 또한 무시할 수 있습니다. 추가 항목이 없는 경우 이 절의 나머지는 건너뛴 수 있습니다.

프로시저

1. 리턴된 항목을 이해합니다.

IBM MQ에는 구성의 다양한 측면 내에 IP 주소, 호스트 이름 및 포트가 포함될 수 있습니다. 이를 두 가지 범주로 분류할 수 있습니다.

- 이 큐 관리자의 위치:** 이 큐 관리자가 사용하거나 발행하며 IBM MQ 네트워크 내의 다른 큐 관리자 또는 애플리케이션에서 연결을 위해 사용할 수 있는 위치 정보입니다.
- 큐 관리자 종속 항목의 위치:** 이 큐 관리자가 알고 있어야 하는 기타 큐 관리자 또는 시스템의 위치.

이 시나리오에는 이 큐 관리자 구성 변경사항에만 집중하므로 범주의 구성 업데이트만을 핸들링합니다. 그러나 이 큐 관리자 위치를 다른 큐 관리자 또는 애플리케이션에서 참조하는 경우, 해당 구성은 이 큐 관리자의 새 위치에 일치하도록 업데이트해야 합니다.

업데이트해야 하는 정보를 포함할 수 있는 두 가지 키 오브젝트가 있습니다.

- **리스너:** 이는 IBM MQ가 청취하는 네트워크 주소를 나타냅니다.
- **CLUSTER RECEIVER 채널:** 큐 관리자가 IBM MQ 클러스터의 일부인 경우 이 오브젝트가 존재합니다. 이는 기타 큐 관리자가 연결할 수 있는 네트워크 주소를 지정합니다.

2. `grep 'CONNAME\|LOCLADDR\|IPADDRV' -B 3 backup.mqsc` 명령의 원래 출력에서 CLUSTER RECEIVER 채널이 정의되었는지 여부를 식별하십시오. 정의되어 있는 경우 IP 주소를 업데이트하십시오.

CLUSTER RECEIVER 채널이 정의되어 있는지 식별하기 위해서는 최초 출력에서 `CHLTYPE(CLUSRCVR)`이 있는 항목을 찾으십시오.

```
DEFINE CHANNEL(ANY_NAME) +  
CHLTYPE(CLUSRCVR) +
```

항목이 있는 경우 `CONNAME`를 IBM MQ Red Hat OpenShift Route로 업데이트하십시오. 이 값은 Red Hat OpenShift 환경을 기반으로 하며 예측 가능한 구문을 사용합니다.

```
queue_manager_resource_name-ibm-mq-qm-openshift_project_name.openshift_app_route_hostname
```

예를 들어, `cp4i` 네임스페이스 내에서 큐 관리자 배치의 이름이 `qm1`이고 `openshift_app_route_hostname`이 `apps.callumj.icp4i.com`인 경우 라우트 URL은 다음과 같습니다.

```
qm1-ibm-mq-qm-cp4i.apps.callumj.icp4i.com
```

라우트의 포트 번호는 일반적으로 443입니다. Red Hat OpenShift 관리자가 다르게 알려주지 않는 한, 일반적으로 이 값이 올바른 값입니다. 이 정보를 사용하여 `CONNAME` 필드를 업데이트하십시오. 예를 들면, 다음과 같습니다.

```
CONNAME('qm1-ibm-mq-qm-cp4i.apps.callumj.icp4i.com(443)')
```

`grep 'CONNAME\|LOCLADDR\|IPADDRV' -B 3 backup.mqsc` 명령의 원래 출력에서 `LOCLADDR` 또는 `IPADDRV`에 대한 항목이 있는지 확인하십시오. 있으면 삭제하십시오. 이는 컨테이너 환경과 관련이 없습니다.

다음에 수행할 작업

이제는 컨테이너 환경의 큐 관리자 구성을 업데이트할 준비가 되었습니다.

이트

컨테이너에서 실행 중인 경우 컨테이너에 의해 특정 구성 측면이 정의되므로 내보낸 구성과 충돌할 수 있습니다.

시작하기 전에

이 태스크에서는 사용자가 IP 주소 및 호스트 이름의 IBM MQ 구성을 변경한 것으로 간주합니다.

이 태스크 정보

다음 구성 측면은 컨테이너가 정의합니다.

- 리스너 정의(공개된 포트에 해당함)
- 잠재적인 TLS 저장소의 위치.

그러므로 내보낸 구성을 업데이트해야 합니다.

1. 리스너 정의 제거.
2. TLS 키 저장소의 위치 정의.

프로시저

1. 리스너 정의를 제거합니다.

백업 구성에서 DEFINE LISTENER를 검색하십시오. 이는 AUTHINFO와 SERVICE 정의 사이에 있어야 합니다. 영역을 강조표시하고 삭제하십시오.

*backup.mqsc

```
** ALTDATE(2020-11-20) +
* ALTTIME(11.43.28) +
  REPLACE
DEFINE AUTHINFO('SYSTEM.DEFAULT.AUTHINFO.CRLLDAP') +
  AUTHTYPE(CRLLDAP) +
  CONNAME(' ') +
* ALTDATE(2020-10-26) +
* ALTTIME(11.43.28) +
  REPLACE
DEFINE LISTENER('SYSTEM.DEFAULT.LISTENER.LU62') +
  TRPTYPE(LU62) +
  CONTROL(MANUAL) +
* ALTDATE(2020-10-26) +
* ALTTIME(11.43.28) +
  REPLACE
DEFINE LISTENER('SYSTEM.DEFAULT.LISTENER.NETBIOS') +
  TRPTYPE(NETBIOS) +
  CONTROL(MANUAL) +
  LOCLNAME(' ') +
* ALTDATE(2020-10-26) +
* ALTTIME(11.43.28) +
  REPLACE
DEFINE LISTENER('SYSTEM.DEFAULT.LISTENER.SPX') +
  TRPTYPE(SPX) +
  CONTROL(MANUAL) +
* ALTDATE(2020-10-26) +
* ALTTIME(11.43.28) +
  REPLACE
DEFINE LISTENER('SYSTEM.DEFAULT.LISTENER.TCP') +
  TRPTYPE(TCP) +
  CONTROL(MANUAL) +
* ALTDATE(2020-10-26) +
* ALTTIME(11.43.28) +
  REPLACE
DEFINE SERVICE('SYSTEM.AMQP.SERVICE') +
  CONTROL(QMGR) +
  SERVTYPE(SERVER) +
  STARTCMD('+MQ_INSTALL_PATH+\bin\amqp.bat') +
  STARTARG('start -m +QMNAME+ -d "+MQ_Q_MGR_DATA_PATH+\.'
  STOPCMD('+MQ_INSTALL_PATH+\bin64\endmqsd.exe') +
```

2. TLS 키 저장소의 위치를 정의합니다.

큐 관리자 백업에는 최초 환경에 대한 TLS 구성이 포함되어 있습니다. 이는 컨테이너 환경과 다르므로 몇 번의 업데이트가 필요합니다.

- **CERTLABL** 항목을 default로 변경하십시오.
- TLS 키 저장소의 위치(**SSLKEYR**)를 /run/runmqserver/tls/key로 변경하십시오.

파일에서 **SSLKEYR** 속성의 위치를 찾으려면 **SSLKEYR**를 검색하십시오. 일반적으로 하나의 항목만 발견됩니다. 여러 항목이 발견되면 다음 설명에 표시된 대로 사용자가 **QMGR** 오브젝트를 편집 중인지 확인하십시오.

```

*backup.mqsc
*****
* Script generated on 2020-10-21   at 11.48.32
* Script generated by user ' CallumJackso' on host 'LAPTOP-VLQKJ5UH'
* Queue manager name: qm1
* Queue manager platform: Windows
* Queue manager command level: (920/920)
* Command issued: dmpmqcfg -m qm1
*****
DEFINE AUTHINFO(USE.LDAP) +
  AUTHTYPE(IDPWLDAP) +
  CONNAME('ldap-service.ldap(389)') +
  LDAPUSER('cn=admin,dc=ibm,dc=com') +
  LDAPPWD('admin') +
  SECCOMM(NO) +
  USRFIELD('uid') +
  SHORTUSR('uid') +
  BASEDNU('ou=people,dc=ibm,dc=com') +
  AUTHORMD(SEARCHGRP) +
  BASEDNG('ou=groups,dc=ibm,dc=com') +
  GRPFIELD('cn') +
  CLASSGRP('groupOfUniqueNames') +
  FINDGRP('uniqueMember') +
  REPLACE
ALTER QMGR +
* ALTDATE(2020-10-26) +
* ALTTIME(11.43.11) +
  CCSTD(850) +
  CERTLABL('default') +
  CLWLUSEQ(LOCAL) +
* COMMANDQ(SYSTEM.ADMIN.COMMAND.QUEUE) +
  CONNAUTH('USE.LDAP') +
* CRDATE(2020-10-26) +
* CRTIME(11.43.11) +
* QMID(qm1_2020-10-26_11.43.11) +
  SSLCRYP(' ') +
  SSLKEYR('/run/runmqserver/tls/key') +
  SUITEB(NONE) +
* VERSION(09020000) +
  FORCE
REFRESH SECURITY

```

다음에 수행할 작업

이제는 컨테이너에서 실행 중인 IBM MQ의 대상 아키텍처를 선택할 준비가 되었습니다.

V 9.2.1 OpenShift CD EUS 컨테이너에서 실행 중인 IBM MQ의 대상 HA 아키텍처 선택

고가용성 요구사항을 충족하도록 단일 인스턴스(단일 Kubernetes 팟(Pod))와 다중 인스턴스(두 개의 팟(Pod)) 중에서 선택하십시오.

시작하기 전에

이 태스크에서는 사용자가 컨테이너 환경의 큐 관리자 구성을 업데이트했다고 가정합니다.

이 태스크 정보

IBM MQ Operator에서는 두 개의 고가용성 옵션을 제공합니다.

- **단일 인스턴스:** 단일 컨테이너(팟(Pod))가 시작되며 장애 발생 시 재시작하는 것은 Red Hat OpenShift의 책임입니다. Kubernetes 내에서 stateful 세트의 특성으로 인해 이 장애 복구 시간이 길어지거나 관리 조치를 완료해야 하는 여러 상황이 있습니다.
- **다중 인스턴스:** 두 개의 컨테이너(각 개별 팟(Pod)에 있음)가 시작되고 하나는 활성 모드이고 다른 하나는 대기 모드입니다. 이 토폴로지를 사용하면 장애 복구가 더 신속해집니다. 여기에서는 IBM MQ 요구사항을 충족하는 Read Write Many 파일 시스템이 필요합니다.

이 태스크에서는 대상 HA 아키텍처만을 선택합니다. 선택한 아키텍처를 구성하는 단계에 대해서는 이 시나리오의 후속 태스크(51 페이지의 『Red Hat OpenShift에서 새 큐 관리자 작성』)에서 설명합니다.

프로시저

1. 두 개의 옵션을 검토합니다.

이러한 두 옵션에 대한 포괄적인 설명은 15 페이지의 『컨테이너에서 IBM MQ에 대한 고가용성』의 내용을 참조하십시오.

2. 대상 HA 아키텍처를 선택합니다.

선택할 옵션이 확실하지 않으면 **단일 인스턴스** 옵션으로 시작하고 이것이 고가용성 요구사항을 충족하는지 확인하십시오.

다음에 수행할 작업

이제는 큐 관리자 자원을 작성할 준비가 되었습니다.

V 9.2.1 OpenShift CD EUS 큐 관리자의 자원 작성

IBM MQ 구성과 TLS 인증서 및 키를 Red Hat OpenShift 환경으로 가져오십시오.

시작하기 전에

이 태스크에서는 사용자가 컨테이너에서 실행 중인 IBM MQ의 대상 아키텍처를 선택했다고 간주합니다.

이 태스크 정보

이전 섹션에서 두 가지 자원을 추출, 업데이트 및 정의했습니다.

- IBM MQ 구성
- TLS 인증서 및 키

큐 관리자가 배치되기 전에 이러한 자원을 Red Hat OpenShift 환경으로 가져와야 합니다.

프로시저

1. IBM MQ 구성을 Red Hat OpenShift로 가져오십시오.

다음 지시사항에서는 backup.mqsc 파일에서 현재 디렉토리에 IBM MQ 구성이 있다고 가정합니다. 그렇지 않으면 사용자 환경을 기반으로 파일 이름을 사용자 정의해야 합니다.

- a) oc login을 사용하여 클러스터에 로그인합니다.
- b) IBM MQ 구성을 configmap로 로드하십시오.

다음 명령을 실행하십시오.

```
oc create configmap my-mqsc-migrated --from-file=backup.mqsc
```

- c) 파일이 성공적으로 로드되었는지 확인합니다.

다음 명령을 실행하십시오.

```
oc describe configmap my-mqsc-migrated
```

2. IBM MQ TLS 자원을 가져옵니다.

37 페이지의 『[선택사항: 큐 관리자 키 및 인증서 추출 및 확보](#)』에서 설명한 바와 같이 TLS는 큐 관리자 배치에 필요할 수 있습니다. 필요한 경우 .crt 및 .key로 끝나는 여러 파일이 이미 있어야 합니다. 배치 시 참조하려면 이를 큐 관리자의 Kubernetes 시크릿에 추가해야 합니다.

예를 들어 큐 관리자의 키 및 인증서가 있는 경우 이름은 다음과 같습니다.

- qmgr.crt
- qmgr.key

이러한 파일을 가져오려면 다음 명령을 실행하십시오.

```
oc create secret tls my-tls-migration --cert=qmgr.crt --key=qmgr.key
```

Kubernetes에서는 일치하는 공용 및 개인 키를 가져올 때 이 유용한 유틸리티를 제공합니다. 예를 들어 큐 관리자 신뢰 저장소에 추가할 추가 인증서가 있는 경우, 다음 명령을 실행하십시오.

```
oc create secret generic my-extra-tls-migration --from-file=comma_separated_list_of_files
```

예를 들어, 가져올 파일이 trust1.crt, trust2.crt 및 trust3.crt인 경우 명령은 다음과 같습니다.

```
oc create secret generic my-extra-tls-migration --from-file=trust1.crt,trust2.crt,trust3.crt
```

다음에 수행할 작업

이제 [Red Hat OpenShift에서 새 큐 관리자를 작성할 준비가 되었습니다](#).

Red Hat OpenShift에서 새 큐 관리자 작성

Red Hat OpenShift에서 단일 인스턴스 또는 다중 인스턴스 큐 관리자를 배치합니다.

시작하기 전에

이 태스크에서는 큐 관리자 자원을 작성했으며 [IBM MQ Operator를 Red Hat OpenShift에 설치했다고 가정합니다](#).

이 태스크 정보

49 페이지의 『[컨테이너에서 실행 중인 IBM MQ의 대상 HA 아키텍처 선택](#)』에서 대략적으로 설명한 바와 같이 두 가지 가능한 배치 토폴로지가 있습니다. 그러므로 이 토픽에서는 두 개의 서로 다른 템플리트를 제공합니다.

- [단일 인스턴스 큐 관리자 배치](#).
- [다중 인스턴스 큐 관리자 배치](#).

중요사항: 선호하는 토폴로지를 기반으로 두 템플리트 중 하나만을 완료하십시오.

프로시저

- [단일 인스턴스 큐 관리자를 배치합니다](#).

마이그레이션된 큐 관리자는 YAML 파일을 사용하여 Red Hat OpenShift에 배치됩니다. 다음은 샘플이며 이전 토픽에서 사용된 이름을 기반으로 합니다.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: qm1
spec:
  version: 9.2.5.0-r3
  license:
    accept: true
    license: L-RJON-C7QG3S
    use: "Production"
  pki:
    keys:
      - name: default
      secret:
        secretName: my-tls-migration
        items:
          - tls.key
          - tls.crt
  web:
    enabled: true
  queueManager:
    name: QM1
  mqsc:
    - configMap:
        name: my-mqsc-migrated
        items:
          - backup.mqsc
```

수행한 단계에 따라 이전 YAML을 사용자 정의해야 할 수도 있습니다. 이를 위해서 다음은 이 YAML에 대한 설명입니다.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: qm1
```

이는 Kubernetes 오브젝트, 유형 및 이름을 정의합니다. 사용자 정의가 필요한 유일한 필드는 name 필드입니다.

```
spec:
  version: 9.2.5.0-r3
  license:
    accept: true
    license: L-RJON-C7QG3S
    use: "Production"
```

이는 배치를 위한 버전 및 라이선스 정보에 해당합니다. 이를 사용자 정의해야 하는 경우, [119 페이지의 『mq.ibm.com/v1beta1에 대한 라이선스 부여 참조』](#)에서 제공하는 정보를 사용하십시오.

```
pki:
  keys:
    - name: default
      secret:
        secretName: my-tls-migration
        items:
          - tls.key
          - tls.crt
```

TLS를 사용하도록 큐 관리자를 구성하려면 관련 인증서 및 키를 참조해야 합니다. secretName 필드는 [IBM MQ TLS 리소스 가져오기](#) 섹션 내에서 작성된 Kubernetes 시크릿을 참조하고 항목 목록(tls.key 및 tls.crt)은 oc create secret tls 구문을 사용할 때 할당되는 표준 이름 Kubernetes입니다. 신뢰 저장소에 추가할 추가 인증서가 있으면 이는 유사한 방식으로 추가될 수 있지만 항목은 가져오기 도중 사용되는 해당 파일 이름입니다. 예를 들어 다음 코드를 사용하여 신뢰 저장소 인증서를 작성할 수 있습니다.

```
oc create secret generic my-extra-tls-migration --from-file=trust1.crt,trust2.crt,trust3.crt
```

```

pki:
  trust:
    - name: default
  secret:
    secretName: my-extra-tls-migration
  items:
    - trust1.crt
    - trust2.crt
    - trust3.crt

```

중요사항: TLS가 필요하지 않으면 YAML의 TLS 섹션을 삭제하십시오.

```

web:
  enabled: true

```

그러면 배치에 웹 콘솔을 사용할 수 있습니다.

```

queueManager:
  name: QM1

```

이는 큐 관리자의 이름을 QM1로 정의합니다. 큐 관리자는 예를 들어 최초 큐 관리자 이름과 같은 사용자 요구 사항에 따라 사용자 정의됩니다.

```

mqsc:
  - configMap:
      name: my-mqsc-migrated
    items:
      - backup.mqsc

```

이전 코드는 [IBM MQ 구성 가져오기](#) 섹션에서 가져온 큐 관리자 구성에 삽입합니다. 다른 이름을 사용한 경우에는 my-mqsc-migrated 및 backup.mqsc(를) 수정해야 합니다.

샘플 YAML에서는 Red Hat OpenShift 환경의 기본 스토리지 클래스가 RWX 또는 RWO 스토리지 클래스로 정의되었다고 가정합니다. 사용자 환경에서 기본값이 정의되어 있지 않으면 사용할 스토리지 클래스를 지정해야 합니다. 다음과 같이 YAML을 확장하여 이를 수행할 수 있습니다.

```

queueManager:
  name: QM1
  storage:
    defaultClass: my_storage_class
  queueManager:
    type: persistent-claim

```

사용자 환경에 일치하도록 클래스 속성이 사용자 정의된 강조표시된 텍스트를 추가하십시오. 사용자 환경에서 스토리지 클래스 이름을 검색하려면 다음 명령을 실행하십시오.

```
oc get storageclass
```

다음은 이 명령에서 리턴하는 샘플 출력입니다.

NAME	PROVISIONER	RECLAIMPOLICY
aws-efs	openshift.org/aws-efs	Delete
gp2 (default)	kubernetes.io/aws-efs	Delete

다음 코드는 [IBM MQ 구성을 가져오기](#) 섹션에서 가져온 IBM MQ 구성을 참조하는 방법을 보여줍니다. 다른 이름을 사용한 경우에는 my-mqsc-migrated 및 backup.mqsc(를) 수정해야 합니다.

```

mqsc:
  - configMap:
      name: my-mqsc-migrated
    items:
      - backup.mqsc

```

단일 인스턴스 큐 관리자를 배치했습니다. 이는 템플릿을 완성합니다. 이제는 새 컨테이너 배치 확인을 수행할 준비가 되었습니다.

- 다중 인스턴스 큐 관리자를 배치합니다.

마이그레이션된 큐 관리자는 YAML 파일을 사용하여 Red Hat OpenShift에 배치됩니다. 다음 샘플은 이전 섹션에서 사용되는 이름을 기반으로 합니다.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: qm1mi
spec:
  version: 9.2.5.0-r3
  license:
    accept: true
    license: L-RJON-C7QG3S
    use: "Production"
  pki:
    keys:
      - name: default
        secret:
          secretName: my-tls-migration
          items:
            - tls.key
            - tls.crt
  web:
    enabled: true
  queueManager:
    name: QM1
    availability: MultiInstance
  storage:
    defaultClass: aws-efs
    persistedData:
      enabled: true
    queueManager:
      enabled: true
    recoveryLogs:
      enabled: true
  mqsc:
    - configMap:
        name: my-mqsc-migrated
        items:
          - backup.mqsc
```

다음은 이 YAML에 대한 설명입니다. 대다수의 구성은 단일 인스턴스 큐 관리자 배치와 동일한 방식을 수행하므로 여기에서는 큐 관리자 가용성 및 스토리지 측면에 대해서만 설명합니다.

```
queueManager:
  name: QM1
  availability: MultiInstance
```

이는 큐 관리자 이름을 QM1(으)로 지정하고 기본 단일 인스턴스 대신 MultiInstance이(가) 되도록 배치를 설정합니다.

```
storage:
  defaultClass: aws-efs
  persistedData:
    enabled: true
  queueManager:
    enabled: true
  recoveryLogs:
    enabled: true
```

IBM MQ 다중 인스턴스 큐 관리자는 RWX 스토리지에 따라 다릅니다. 기본적으로 큐 관리자는 단일 인스턴스 모드에 배치되므로 다중 인스턴스 모드로 변경하는 경우 추가 스토리지 옵션이 필요합니다. 이전 YAML 샘플에서는 세 개의 스토리지 지속 볼륨과 하나의 지속 볼륨 클래스를 정의했습니다. 이 지속 볼륨 클래스는 RWX 스토리지 클래스가 되어야 합니다. 사용자 환경에서 스토리지 클래스 이름이 확실하지 않으면 다음 명령을 실행하여 이를 검색할 수 있습니다.

```
oc get storageclass
```

다음은 이 명령에서 리턴하는 샘플 출력입니다.

NAME	PROVISIONER	RECLAIMPOLICY
aws-efs	openshift.org/aws-efs	Delete
gp2 (default)	kubernetes.io/aws-efs	Delete

다음 코드는 [IBM MQ 구성을 가져오기](#) 섹션에서 가져온 IBM MQ 구성을 참조하는 방법을 보여줍니다. 다른 이름을 사용한 경우에는 my-mqsc-migrated 및 backup.mqsc을(를) 수정해야 합니다.

```
mqsc:
- configMap:
  name: my-mqsc-migrated
  items:
  - backup.mqsc
```

다중 인스턴스 큐 관리자를 배치했습니다. 이는 템플릿을 완성합니다. 이제는 [새 컨테이너 배치 확인](#)을 수행할 준비가 되었습니다.

V 9.2.1 OpenShift CD EUS 새 컨테이너 배치 확인

이제 IBM MQ는 Red Hat OpenShift에 배치되었으므로 IBM MQ 샘플을 사용하여 환경을 확인할 수 있습니다.

시작하기 전에

이 태스크에서는 사용자가 Red Hat OpenShift에서 새 큐 관리자를 작성했다고 간주합니다.

중요사항: 이 태스크는 TLS가 큐 관리자에서 사용으로 설정되지 않은 것으로 간주합니다.

이 태스크 정보

이 태스크에서는 마이그레이션된 큐 관리자의 컨테이너 내부에서 IBM MQ 샘플을 실행합니다. 그러나 다른 환경에서 실행 중인 자체 애플리케이션을 사용하려 할 수 있습니다.

다음 정보가 필요합니다.

- LDAP 사용자 이름
- LDAP 비밀번호
- IBM MQ 채널 이름
- 큐 이름

이 예제 코드에서는 다음 설정을 사용합니다. 사용자 설정이 다를 것임을 참고하십시오.

- LDAP 사용자 이름: mqapp
- LDAP 비밀번호: mqapp
- IBM MQ 채널 이름: DEV.APP.SVRCONN
- 큐 이름: Q1

프로시저

1. 실행 중인 IBM MQ 컨테이너에서 실행합니다.

다음 명령을 사용하십시오.

```
oc exec -it qm1-ibm-mq-0 /bin/bash
```

여기서 qm1-ibm-mq-0은 51 페이지의 [『Red Hat OpenShift에서 새 큐 관리자 작성』](#)에 배치한 팟(Pod)입니다. 배치를 다른 이름으로 명명한 경우 이 값을 사용자 정의하십시오.

2. 메시지를 송신합니다.

다음 명령을 실행하십시오.

```
cd /opt/mqm/samp/bin
export IBM MQSAMP_USER_ID=mqapp
export IBM MQSERVÉR=DEV.APP.SVRCONN/TCP/'localhost(1414)'
./amqsputc Q1 QM1
```

프롬프트를 표시하여 비밀번호를 입력한 후 메시지를 송신할 수 있습니다.

3. 메시지가 성공적으로 수신되었는지 확인합니다.

GET 샘플을 실행하십시오.

```
./amqsgetc Q1 QM1
```

결과

35 페이지의 『IBM MQ를 IBM Cloud Pak for Integration으로 마이그레이션』을 완료했습니다.

다음에 수행할 작업

다음 정보를 사용하면 좀 더 복잡한 마이그레이션 시나리오에서 도움이 됩니다.

큐에 있는 메시지 마이그레이션

큐에 있는 기존의 메시지를 마이그레이션하려면 새 큐 관리자를 배치한 후 두 시스템 사이에서 `dmpmqmsg` 유틸리티 사용 토폴에 있는 지침에 따라 메시지를 내보내기 및 가져오기하십시오.

Red Hat OpenShift 환경 외부에서 IBM MQ에 연결

배치된 큐 관리자는 Red Hat OpenShift 환경 외부의 IBM MQ 클라이언트 및 큐 관리자에 공개될 수 있습니다. 프로세스는 Red Hat OpenShift 환경에 연결하는 IBM MQ 버전에 따라 다릅니다. 102 페이지의 『Red Hat OpenShift 클러스터 외부에서 큐 관리자에 연결하도록 Route 구성』의 내용을 참조하십시오.

OpenShift

CP4I

Red Hat OpenShift에 IBM MQ Operator 설치 및 설치 제거

Operator Hub를 사용하여 IBM MQ Operator를 Red Hat OpenShift에 설치할 수 있습니다.

프로시저

- 9 페이지의 『IBM MQ Operator에 대한 종속 항목』.
- 10 페이지의 『IBM MQ Operator에서 필요로 하는 클러스터 범위 권한』.
- 56 페이지의 『Red Hat OpenShift 웹 콘솔을 사용하여 IBM MQ Operator 설치』.
- 58 페이지의 『Red Hat OpenShift CLI를 사용하여 IBM MQ Operator 설치』.
- 62 페이지의 『에어 갭 환경에 IBM MQ Operator 설치』.

관련 태스크

58 페이지의 『Red Hat OpenShift 웹 콘솔을 사용하여 IBM MQ Operator 설치 제거』

Red Hat OpenShift 웹 콘솔을 사용하여 Red Hat OpenShift에서 IBM MQ Operator를 설치 제거할 수 있습니다.

61 페이지의 『Red Hat OpenShift CLI를 사용하여 IBM MQ Operator 설치 제거』

Red Hat OpenShift CLI를 사용하여 Red Hat OpenShift에서 IBM MQ Operator를 설치 제거할 수 있습니다. IBM MQ Operator의 설치 위치가 하나의 네임스페이스인지, 또는 클러스터의 모든 네임스페이스에 설치되어 사용 가능함에 따라 설치 제거 프로세스에 차이가 있습니다.

OpenShift

CP4I

Red Hat OpenShift 웹 콘솔을 사용하여 IBM MQ Operator 설치

Operator Hub를 사용하여 IBM MQ Operator를 Red Hat OpenShift에 설치할 수 있습니다.

시작하기 전에

Red Hat OpenShift 클러스터 웹 콘솔에 로그인하십시오.

프로시저

1.

옵션: IBM Common Services Operator를 설치 가능한 Operator의 목록에 추가하십시오.

참고:

이 단계는 IBM MQ Operator 1.5 이하 릴리스에 적용됩니다. 이 단계는 별도의 Common Services 카탈로그를 추가합니다. 이후 Operator 릴리스에서는 Common Services가 IBM 카탈로그에 포함됩니다.

- 화면의 오른쪽 상단에 있는 더하기 아이콘을 클릭하십시오. **YAML 가져오기** 대화 상자를 볼 수 있습니다.
- 대화 상자에서 다음 자원 정의를 붙여넣으십시오.

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: opencloud-operators
  namespace: openshift-marketplace
spec:
  displayName: IBMCS Operators
  publisher: IBM
  sourceType: grpc
  image: icr.io/cpopen/ibm-common-service-catalog:latest
  updateStrategy:
    registryPoll:
      interval: 45m
```

c) **작성**을 클릭하십시오.

2. IBM 연산자를 설치 가능한 연산자 목록에 추가

- 화면의 오른쪽 상단에 있는 더하기 아이콘을 클릭하십시오. **YAML 가져오기** 대화 상자를 볼 수 있습니다.
- 대화 상자에서 다음 자원 정의를 붙여넣으십시오.

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-operator-catalog
  namespace: openshift-marketplace
spec:
  displayName: IBM Operator Catalog
  image: icr.io/cpopen/ibm-operator-catalog:latest
  publisher: IBM
  sourceType: grpc
  updateStrategy:
    registryPoll:
      interval: 45m
```

c) **작성**을 클릭하십시오.

3. IBM MQ Operator에 사용할 네임스페이스 작성

IBM MQ Operator는 하나의 네임스페이스나 모든 네임스페이스를 범위로 지정하여 설치할 수 있습니다. 이 단계는 아직 존재하지 않는 특정 네임스페이스에 설치하려는 경우에만 필요합니다.

- 탐색 패널에서 **홈 > 프로젝트**를 클릭하십시오.
프로젝트 페이지가 표시됩니다.
- 프로젝트 작성**을 클릭하십시오. 프로젝트 작성 범위가 표시됩니다.
- 작성 중인 네임스페이스의 세부사항을 입력하십시오. 예를 들어, "ibm-mq"를 이름으로 지정할 수 있습니다.
- 작성**을 클릭하십시오. IBM MQ Operator의 네임스페이스가 작성됩니다.

4. IBM MQ Operator를 설치하십시오.

- 탐색 패널에서 **Operators > OperatorHub**를 클릭하십시오.
OperatorHub 페이지가 표시됩니다.

- b) 모든 항목 필드에 "IBM MQ"을 입력하십시오.
IBM MQ 카탈로그 항목이 표시됩니다.
- c) **IBM MQ**를 선택하십시오.
IBM MQ 창이 표시됩니다.
- d) 설치를 클릭하십시오.
Operator 구독 작성 페이지를 볼 수 있습니다.
- e) 선택할 Operator 채널을 판별하려면 7 페이지의 『IBM MQ Operator에 대한 버전 지원』의 내용을 검토하십시오.
- f) 설치 모드를 사용자가 작성한 특정 네임스페이스 또는 클러스터 전체 범위로 설정하십시오.
다른 네임스페이스에 다른 버전의 연산자를 설치하면 문제점이 발생할 수 있으므로 클러스터 전체 범위를 선택하는 것이 좋습니다. 연산자는 제어 플레인의 확장이 되도록 설계되었습니다.
- g) 구독을 클릭하십시오.
설치된 연산자 페이지에서 IBM MQ를 볼 수 있습니다.
- h) 설치된 연산자 페이지에서 연산자의 상태를 확인하십시오. 설치가 완료되면 상태가 성공으로 변경됩니다.

다음에 수행할 작업

76 페이지의 『Red Hat OpenShift 웹 콘솔을 사용하여 IBM MQ용 Red Hat OpenShift 프로젝트 준비』

 **Red Hat OpenShift 웹 콘솔을 사용하여 IBM MQ Operator 설치 제거**
Red Hat OpenShift 웹 콘솔을 사용하여 Red Hat OpenShift에서 IBM MQ Operator를 설치 제거할 수 있습니다.

시작하기 전에

Red Hat OpenShift 클러스터의 웹 콘솔에 로그인하십시오.

IBM MQ Operator가 클러스터의 모든 프로젝트/네임스페이스에 설치된 경우에는 큐 관리자를 삭제할 각 프로젝트에 대해 다음 프로시저의 1 - 5단계를 반복하십시오.

프로시저

1. **Operator > 설치된 Operator**를 선택하십시오.
2. **프로젝트** 드롭 다운 목록에서 프로젝트를 선택하십시오.
3. **IBM MQ Operator**를 클릭하십시오.
4. **큐 관리자** 탭을 클릭하여 이 IBM MQ Operator에서 관리하는 큐 관리자를 보십시오.
5. 하나 이상의 큐 관리자를 삭제하십시오.
이러한 큐 관리자는 계속해서 실행되지만 IBM MQ Operator 없이는 예상한 대로 작동하지 않을 수 있다는 점을 참고하십시오.
6. 옵션: 해당되는 경우에는 큐 관리자를 삭제할 각 프로젝트에 대해 1 - 5단계를 반복하십시오.
7. **Operator > 설치된 Operator**로 되돌아가십시오.
8. **IBM MQ Operator** 옆에 있는, 점 세 개로 표시된 메뉴를 클릭하고 **Operator 설치 제거**를 선택하십시오.
9. Red Hat OpenShift Container Platform 4.7을 사용 중인 경우 명령행에서 유효성 검증 웹훅을 수동으로 삭제해야 할 수 있습니다.

```
oc delete validatingwebhookconfiguration namespace.validator.queuemanagers.mq.ibm.com
```

 **Red Hat OpenShift CLI를 사용하여 IBM MQ Operator 설치**

Operator Hub를 사용하여 IBM MQ Operator를 Red Hat OpenShift에 설치할 수 있습니다.

시작하기 전에

oc login를 사용하여 Red Hat OpenShift 명령행 인터페이스 (CLI) 에 로그인하십시오. 이 단계에서는 클러스터 관리자여야 합니다.

프로시저

1.

옵션: IBM Common Services 운영자에 대한 **CatalogSource**을(를) 작성하십시오.

참고:

이 단계는 IBM MQ Operator 1.5 이하 릴리스에 적용됩니다. 이 단계는 별도의 Common Services 카탈로그를 추가합니다. 이후 Operator 릴리스에서는 Common Services가 IBM 카탈로그에 포함됩니다.

a) **CatalogSource** 자원을 정의하는 YAML 파일을 작성하십시오.

다음 콘텐츠를 사용하여 "operator-source-cs.yaml"라는 파일 작성:

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: opencloud-operators
  namespace: openshift-marketplace
spec:
  displayName: IBMCS Operators
  publisher: IBM
  sourceType: grpc
  image: icr.io/cpopen/ibm-common-service-catalog:latest
  updateStrategy:
    registryPoll:
      interval: 45m
```

b) 서버에 **CatalogSource**을(를) 적용하십시오.

```
oc apply -f operator-source-cs.yaml -n openshift-marketplace
```

2. IBM 운영자에 대한 **CatalogSource** 작성

a) **CatalogSource** 자원을 정의하는 YAML 파일 작성

다음 콘텐츠를 사용하여 "operator-source-ibm.yaml"이라는 파일 작성:

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-operator-catalog
  namespace: openshift-marketplace
spec:
  displayName: IBM Operator Catalog
  image: icr.io/cpopen/ibm-operator-catalog:latest
  publisher: IBM
  sourceType: grpc
  updateStrategy:
    registryPoll:
      interval: 45m
```

b) 서버에 **CatalogSource**을(를) 적용하십시오.

```
oc apply -f operator-source-ibm.yaml -n openshift-marketplace
```

3. IBM MQ Operator에 사용할 네임스페이스 작성

IBM MQ Operator는 하나의 네임스페이스나 모든 네임스페이스를 범위로 지정하여 설치할 수 있습니다. 이 단계는 아직 존재하지 않는 특정 네임스페이스에 설치하려는 경우에만 필요합니다.

```
oc new-project ibm-mq
```

4. OperatorHub에서 클러스터에 사용 가능한 Operator의 목록 보기

```
oc get packagemanifests -n openshift-marketplace
```

5. IBM MQ Operator를 검사하여 지원되는 InstallMode 및 사용 가능한 채널 확인

```
oc describe packagemanifests ibm-mq -n openshift-marketplace
```

6. OperatorGroup 오브젝트 YAML 파일 작성

OperatorGroup은(는) **OperatorGroup**과(와) 동일한 네임스페이스에 있는 모든 운영자에 필요한 RBAC 액세스를 생성하는 대상 네임스페이스를 선택하는 OLM 자원입니다.

운영자에 등록하는 네임 스페이스에는 **AllNamespaces** 또는 **SingleNamespace** 모드 중 하나인 운영자의 **InstallMode**과(와) 일치하는 **OperatorGroup**이(가) 있어야 합니다. 설치하려는 운영자가 **AllNamespaces**를 사용하는 경우, **openshift-operators** 네임스페이스에 이미 적절한 **OperatorGroup**이(가) 있습니다.

그러나 운영자가 **SingleNamespace** 모드를 사용하고 적절한 **OperatorGroup**이(가) 없는 경우에는 이를 작성해야 합니다.

a) 다음 콘텐츠를 사용하여 "mq-operator-group.yaml"이라는 파일을 작성합니다.

```
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: <operatorgroup_name>
  namespace: <namespace_name>
spec:
  targetNamespaces:
  - <namespace_name>
```

b) **OperatorGroup** 오브젝트를 작성하십시오.

```
oc apply -f mq-operator-group.yaml
```

7. Subscription 오브젝트 YAML 파일을 작성하여 IBM MQ Operator에 네임스페이스를 등록합니다.

a) 선택할 Operator 채널을 판별하려면 7 페이지의 『IBM MQ Operator에 대한 버전 지원』의 내용을 검토하십시오.

b) 다음 내용을 사용하여 "mq-sub.yaml" 파일을 작성하십시오. 그러나 **channel**을(를) 변경하여 설치하려는 IBM MQ Operator의 버전에 대한 채널과 일치시키십시오.

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-mq
  namespace: openshift-operators
spec:
  channel: <ibm-mq-operator-channel>
  name: ibm-mq
  source: ibm-operator-catalog
  sourceNamespace: openshift-marketplace
```

AllNamespaces InstallMode 사용의 경우, 네임스페이스에 **openshift-operators**을(를) 지정하십시오. 그렇지 않으면, **SingleNamespace InstallMode** 사용에 대한 관련 단일 네임스페이스를 지정하십시오. **sourceNamespace** 필드는 그대로 두고 **namespace** 필드를 변경해야 합니다.

c) **Subscription** 오브젝트를 작성하십시오.

```
oc apply -f mq-sub.yaml
```

8. 운영자의 상태 확인

운영자 설치에 성공하면 팟(Pod) 상태는 실행 중으로 표시됩니다. **AllNamespaces InstallMode** 사용의 경우, **openshift-operators**을(를) 네임스페이스로 지정하십시오. 그렇지 않으면, **SingleNamespace InstallMode** 사용에 대한 관련 단일 네임스페이스를 지정하십시오.

```
oc get pods -n <namespace_name>
```

다음에 수행할 작업

76 페이지의 『Red Hat OpenShift CLI를 사용하여 IBM MQ용 Red Hat OpenShift 프로젝트 준비』

OpenShift CP4I Red Hat OpenShift CLI를 사용하여 IBM MQ Operator 설치 제거

Red Hat OpenShift CLI를 사용하여 Red Hat OpenShift에서 IBM MQ Operator를 설치 제거할 수 있습니다. IBM MQ Operator의 설치 위치가 하나의 네임스페이스인지, 또는 클러스터의 모든 네임스페이스에 설치되어 사용 가능한지에 따라 설치 제거 프로세스에 차이가 있습니다.

시작하기 전에

oc login를 사용하여 Red Hat OpenShift 클러스터에 로그인하십시오.

프로시저

- IBM MQ Operator가 하나의 네임스페이스에 설치된 경우에는 다음 하위 단계를 완료하십시오.

- a) 현재 위치가 올바른 프로젝트인지 확인하십시오.

```
oc project <project_name>
```

- b) 프로젝트에 설치된 큐 관리자를 보십시오.

```
oc get qmgr
```

- c) 하나 이상의 큐 관리자를 삭제하십시오.

```
oc delete qmgr <qmgr_name>
```

이러한 큐 관리자는 계속해서 실행되지만 IBM MQ Operator 없이는 예상한 대로 작동하지 않을 수 있다는 점을 참고하십시오.

- d) **ClusterServiceVersion** 인스턴스를 보십시오.

```
oc get csv
```

- e) IBM MQ **ClusterServiceVersion**을 삭제하십시오.

```
oc delete csv <ibm_mq_csv_name>
```

- f) 구독을 보십시오.

```
oc get subscription
```

- g) 모든 구독을 삭제하십시오.

```
oc delete subscription <ibm_mq_subscription_name>
```

- h) 옵션: Common Services를 사용하고 있는 다른 항목이 없는 경우에는 Common Services Operator를 설치 제거하고 Operator 그룹을 삭제할 수도 있습니다.

- a. IBM Cloud Pak foundational services 제품 문서에서 [공통 서비스 설치 제거](#)의 지시사항에 따라 공통 서비스 운영자를 설치 제거하십시오.

- b. Operator 그룹을 보십시오.

```
oc get operatorgroup
```

- c. Operator 그룹을 삭제하십시오.

```
oc delete OperatorGroup <operator_group_name>
```

- IBM MQ Operator가 클러스터의 모든 네임스페이스에 설치되어 사용 가능한 경우에는 다음 하위 단계를 완료하십시오.

- a) 설치된 모든 큐 관리자를 보십시오.

```
oc get qmgr -A
```

b) 하나 이상의 큐 관리자를 삭제하십시오.

```
oc delete qmgr <qmgr_name> -n <namespace_name>
```

이러한 큐 관리자는 계속해서 실행되지만 IBM MQ Operator 없이는 예상한 대로 작동하지 않을 수 있다는 점을 참고하십시오.

c) **ClusterServiceVersion** 인스턴스를 보십시오.

```
oc get csv -A
```

d) 클러스터에서 IBM MQ **ClusterServiceVersion**을 삭제하십시오.

```
oc delete csv <ibm_mq_csv_name> -n openshift-operators
```

e) 구독을 보십시오.

```
oc get subscription -n openshift-operators
```

f) 구독을 삭제하십시오.

```
oc delete subscription <ibm_mq_subscription_name> -n openshift-operators
```

g) Red Hat OpenShift Container Platform 4.7을 사용 중인 경우 유효성 검증 웹훅을 수동으로 삭제해야 할 수 있습니다.

```
oc delete validatingwebhookconfiguration namespace.validator.queuemanagers.mq.ibm.com
```

h) 옵션: Common Services를 사용하고 있는 다른 항목이 없는 경우에는 Common Services Operator를 설치 제거할 수도 있습니다.

IBM Cloud Pak foundational services 제품 문서에서 [공통 서비스 설치 제거](#)의 지시사항을 따르십시오.

OpenShift CP4I Linux 에어 갭 환경에 IBM MQ Operator 설치

이 학습에서는 인터넷이 연결되지 않은 Red Hat OpenShift 클러스터에 IBM MQ Operator를 설치하는 작업을 안내합니다. 휴대용 스토리지 디바이스를 사용하거나 베스틴 시스템을 사용하여 에어 갭 환경에서 IBM MQ Operator를 설치할 수 있습니다.

휴대용 스토리지 디바이스를 사용하여 에어 갭 환경에서 IBM MQ Operator 설치

설치를 완료하는 단계는 IBM Cloud Pak for Integration 문서의 [휴대용 스토리지 디바이스로 이미지 미러링](#)을 참조하십시오. IBM MQ만 설치하는 경우 다음 환경 변수의 모든 발생을 여기에 제공된 값으로 바꾸십시오.

```
export CASE_NAME=ibm-mq
export CASE_ARCHIVE_VERSION=version_number
export CASE_INVENTORY_SETUP=ibmMQOperator
```

여기서 *version_number*는 에어 갭 설치를 위해 사용할 케이스의 버전입니다. 사용 가능한 케이스 버전의 목록은 <https://github.com/IBM/cloud-pak/tree/master/repo/case/ibm-mq>를 참조하십시오. 선택할 Operator 채널을 판별하려면 7 페이지의 [『IBM MQ Operator에 대한 버전 지원』](#)의 내용을 검토하십시오.

베스틴 시스템을 사용하여 에어 갭 환경에서 IBM MQ Operator 설치

1. 63 페이지의 [『전제조건』](#)
2. 63 페이지의 [『Docker 레지스트리 준비』](#)
3. 64 페이지의 [『베스틴 호스트 준비』](#)
4. 64 페이지의 [『설치 프로그램 및 이미지 인벤토리에 환경 변수 작성』](#)
5. 65 페이지의 [『IBM MQ 설치 프로그램 및 이미지 인벤토리 다운로드』](#)

6. 65 페이지의 『[Red Hat OpenShift Container Platform 클러스터에 클러스터 관리자로 로그인](#)』
7. 65 페이지의 『[IBM MQ Operator에 대한 Kubernetes 네임스페이스 작성](#)』
8. 65 페이지의 『[이미지 미리 및 클러스터 구성](#)』
9. 67 페이지의 『[IBM MQ Operator를 설치하십시오.](#)』
10. 68 페이지의 『[IBM MQ Queue Manager 배치](#)』

전제조건

1. Red Hat OpenShift Container Platform 클러스터를 설치해야 합니다. 지원되는 Red Hat OpenShift Container Platform 버전의 경우 7 페이지의 『[IBM MQ Operator에 대한 버전 지원](#)』의 내용을 참조하십시오.
2. Docker 레지스트리가 사용 가능해야 합니다. 자세한 정보는 63 페이지의 『[Docker 레지스트리 준비](#)』의 내용을 참조하십시오.
3. 기본 서버를 구성해야 합니다. 자세한 정보는 64 페이지의 『[베스천 호스트 준비](#)』의 내용을 참조하십시오.

Docker 레지스트리 준비

로컬 Docker 레지스트리는 로컬 환경에 모든 이미지를 저장하는데 사용됩니다. 이러한 레지스트리를 작성해야 하며 다음 요구사항을 충족하는지 확인해야 합니다.

- [Docker Manifest V2, Schema 2](#)를 지원합니다.
- 멀티 아키텍처 이미지를 지원합니다.
- 베스천 서버와 Red Hat OpenShift Container Platform 클러스터 노드 모두에서 액세스 가능합니다.
- 베스천 호스트에서 대상 레지스트리에 쓸 수 있는 사용자의 사용자 이름과 비밀번호입니다.
- Red Hat OpenShift 클러스터 노드에 있는 대상 레지스트리에서 읽을 수 있는 사용자의 사용자 이름과 비밀번호가 있습니다.
- 이미지 이름에서 경로 구분 기호를 허용합니다.

Docker 레지스트리를 작성한 후에는 다음과 같이 레지스트리를 구성해야 합니다.

1. 레지스트리 네임스페이스 작성

- [ibmcom - dockerhub.io/ibmcom](#) 네임스페이스에서 모든 이미지를 저장하는 네임스페이스입니다.
ibmcom 네임스페이스는 공개적으로 사용할 수 있는 모든 IBM 이미지용이며 pull할 신임 정보를 필요로 하지 않습니다.
- [cp - cp.icr.io/cp](#) 저장소에서 IBM 이미지를 저장하기 위한 네임스페이스입니다.
cp 네임스페이스는 pull할 인타이틀먼트 키 및 신임 정보를 필요로 하는 IBM Entitled Registry의 이미지용입니다. 자격 부여 키를 가져오려면 자격 부여된 소프트웨어와 연관된 IBM ID 및 비밀번호를 사용하여 [MyIBM 컨테이너 소프트웨어 라이브러리](#)에 로그인하십시오. **인타이틀먼트 키** 섹션에서 **키 복사**를 선택하여 인타이틀먼트 키를 클립보드에 복사한 다음 다음 단계에서 사용하도록 저장하십시오.
- [opencloudio - quay.io/opencloudio](#)에서 이미지를 저장할 네임스페이스입니다.
opencloudio 네임스페이스는 [quay.io](#)에서 사용 가능한 IBM 오픈 소스 컴포넌트 이미지를 선택하기 위한 것입니다. IBM Cloud Pak foundational services 이미지는 opencloudio에 호스트됩니다.

2. 각 네임스페이스가 다음 요구사항을 충족하는지 검증하십시오.

- 자동 저장소 작성을 지원합니다.
- 저장소를 작성하고 작성할 수 있는 사용자의 신임 정보가 있습니다. 베스천 호스트는 이 신임 정보를 사용합니다.
- 모든 저장소를 읽을 수 있는 사용자의 신임 정보가 있습니다. Red Hat OpenShift Container Platform 클러스터는 이러한 신임 정보를 사용합니다.

베스천 호스트 준비

Red Hat OpenShift Container Platform 클러스터, 로컬 Docker 레지스트리 및 인터넷에 액세스할 수 있는 베스천 호스트를 준비하십시오. 기본 호스트는 IBM Cloud Pak CLI 및 Red Hat OpenShift Container Platform CLI가 지원하는 운영 체제가 있는 Linux for x86-64 플랫폼에 있어야 합니다.

베스천 노드에서 다음 단계를 완료하십시오.

1. OpenSSL 버전 1.11.1 이상을 설치하십시오.
2. 베스천 노드에 Docker 또는 Podman을 설치하십시오.

- Docker를 설치하려면 이 명령을 실행하십시오.

```
yum check-update
yum install docker
```

- Podman을 설치하려면 [Podman 설치 지시사항](#)을 참조하십시오.

3. 베스천 노드에 skopeo 버전 1.x.x를 설치하십시오. skopeo를 설치하려면 다음 명령을 실행하십시오.

```
yum check-update
yum install skopeo
```

4. IBM Cloud Pak CLI를 설치하십시오. 플랫폼에 대한 2진 파일의 최신 버전을 설치하십시오. 자세한 정보는 [cloud-pak-cli](#)의 내용을 참조하십시오.

- a. 2진 파일을 다운로드하십시오.

```
wget https://github.com/IBM/cloud-pak-cli/releases/download/vversion-number/binary-file-name
```

예를 들면, 다음과 같습니다.

```
wget https://github.com/IBM/cloud-pak-cli/releases/latest/download/cloudctl-linux-amd64.tar.gz
```

- b. 2진 파일을 추출하십시오.

```
tar -xvf binary-file-name
```

- c. 다음 명령을 실행하여 파일을 수정하고 이동하십시오.

```
chmod 755 file-name
mv file-name /usr/local/bin/cloudctl
```

- d. cloudctl가 설치되어 있는지 확인하십시오.

```
cloudctl --help
```

5. oc Red Hat OpenShift Container Platform CLI 도구를 설치하십시오.

자세한 정보는 [Red Hat OpenShift Container Platform CLI 도구](#)를 참조하십시오.

6. 오프라인 상점 역할을 하는 디렉토리를 작성하십시오.

다음은 예제 디렉토리입니다. 이 예는 후속 단계에서 사용됩니다.

```
mkdir $HOME/offline
```

참고: 이 오프라인 상점은 데이터를 두 번 이상 전송하지 않도록 지속적이어야 합니다. 이 지속성은 미러링 프로세스를 여러 번 실행하거나 스케줄에 따라 실행하는 데도 도움이 됩니다.

설치 프로그램 및 이미지 인벤토리에 환경 변수 작성

설치 프로그램 이미지 이름 및 이미지 인벤토리를 사용하여 다음 환경 변수를 작성하십시오.

```
export CASE_ARCHIVE_VERSION=version_number
export CASE_ARCHIVE=ibm-mq-$CASE_ARCHIVE_VERSION.tgz
export CASE_INVENTORY=ibmMQOperator
```

여기서 *version_number*는 에어 갭 설치를 위해 사용할 케이스의 버전입니다. 사용 가능한 케이스 버전의 목록은 <https://github.com/IBM/cloud-pak/tree/master/repo/case/ibm-mq>를 참조하십시오. **IBM MQ Operator에 대한 버전 지원** 을 검토하여 선택할 운영자 채널을 판별하십시오.

IBM MQ 설치 프로그램 및 이미지 인벤토리 다운로드

ibm-mq 설치 프로그램 및 이미지 자원 명세를 bastion 호스트에 다운로드하십시오.

```
cloudctl case save \  
  --case https://github.com/IBM/cloud-pak/raw/master/repo/case/ibm-mq/$CASE_ARCHIVE_VERSION/  
$CASE_ARCHIVE \  
  --outputdir $HOME/offline/
```

Red Hat OpenShift Container Platform 클러스터에 클러스터 관리자로 로그인

다음은 Red Hat OpenShift Container Platform 클러스터에 로그인하기 위한 예 명령입니다.

```
oc login cluster_host:port --username=cluster_admin_user --password=cluster_admin_password
```

IBM MQ Operator에 대한 Kubernetes 네임스페이스 작성

네임스페이스가 있는 환경 변수를 작성하여 IBM MQ Operator를 설치한 후 네임스페이스를 작성하십시오.

```
export NAMESPACE=ibm-mq-test  
oc create namespace }${NAMESPACE}
```

이미지 미리 및 클러스터 구성

다음 단계를 완료하여 이미지를 미리링하고 클러스터를 구성합니다.

참고: 명령에서 큰따옴표 내에 물결 기호를 사용하지 마십시오. 예를 들어, args "--registry *registry* --user *registry_userid* --pass *registry_password* --inputDir ~/offline"을(를) 사용하지 마십시오. 틸드가 확장되지 않으며 명령이 실패할 수 있습니다.

1. 모든 소스 Docker 레지스트리에 대한 인증 신임 정보를 저장합니다.

모든 IBM Cloud Platform Common Services, IBM MQ Operator 이미지 및 IBM MQ Advanced Developer 이미지는 인증이 필요하지 않은 공용 레지스트리에 저장됩니다. 그러나 IBM MQ Advanced Server(비 개발자), 기타 제품 및 써드파티 컴포넌트는 하나 이상의 인증된 레지스트리를 필요로 합니다. 다음 레지스트리에는 인증이 필요합니다.

- cp.icr.io
- registry.redhat.io
- registry.access.redhat.com

이러한 레지스트리에 대한 자세한 정보는 [레지스트리 네임스페이스 작성](#)을 참조하십시오.

인증이 필요한 모든 레지스트리에 대한 신임 정보를 구성하려면 다음 명령을 실행해야 합니다. 각 레지스트리에 대해 개별적으로 명령을 실행하십시오.

```
cloudctl case launch \  
  --case $HOME/offline/}${CASE_ARCHIVE} \  
  --inventory }${CASE_INVENTORY} \  
  --action configure-creds-airgap \  
  --namespace }${NAMESPACE} \  
  --args "--registry registry --user registry_userid --pass registry_password --inputDir $HOME/offline"
```

명령은 `$HOME/.airgap/secrets` 위치의 파일 시스템의 파일에 레지스트리 신임 정보를 저장하고 캐시합니다.

2. 로컬 Docker 레지스트리 연결 정보로 환경 변수를 작성합니다.

```
export LOCAL_DOCKER_REGISTRY=IP_or_FQDN_of_local_docker_registry
export LOCAL_DOCKER_USER=username
export LOCAL_DOCKER_PASSWORD=password
```

참조: Docker 레지스트리는 80 또는 443과 같은 표준 포트를 사용합니다. Docker 레지스트리는 비 표준 포트를 사용하는 경우, 구문 `host:port`를 사용하여 포트를 지정하십시오. 예를 들면, 다음과 같습니다.

```
export LOCAL_DOCKER_REGISTRY=myregistry.local:5000
```

3. 로컬 Docker 레지스트리에 대한 인증 본인확인정보를 구성하십시오.

참고: 이 단계는 한 번만 수행해야 합니다.

```
cloudctl case launch \
--case $HOME/offline/${CASE_ARCHIVE} \
--inventory ${CASE_INVENTORY} \
--action configure-creds-airgap \
--namespace ${NAMESPACE} \
--args "--registry ${LOCAL_DOCKER_REGISTRY} --user ${LOCAL_DOCKER_USER} --pass ${LOCAL_DOCKER_PASSWORD}"
```

명령은 `$HOME/.airgap/secrets` 위치의 파일 시스템의 파일에 레지스트리 신임 정보를 저장하고 캐시합니다.

4. 글로벌 이미지 pull 본인확인정보 및 **ImageContentSourcePolicy**를 구성하십시오.

- a. 노드를 다시 시작해야 하는지 여부를 확인하십시오.

- Red Hat OpenShift Container Platform 버전 4.4 이상 및 에어 갭을 사용하는 IBM MQ Operator 의 새 설치에서 이 단계는 모든 클러스터 노드를 다시 시작합니다. 클러스터 자원은 새 풀 본인확인정보가 적용될 때까지 사용 불가능할 수 있습니다.
- IBM MQ Operator 1.8에서 CASE 는 이미지에 대한 추가 미러링 소스를 포함하도록 업데이트됩니다. 따라서 이전 버전의 IBM MQ Operator 에서 버전 1.8 이상으로 업그레이드하면 노드 다시 시작이 트리거됩니다.
- 이 단계에서 노드를 다시 시작해야 하는지 확인하려면 이 단계의 코드에 `--dry - run` 옵션을 추가하십시오. 그러면 최신 **ImageContentSourcePolicy** 가 생성되어 콘솔 창 (**stdout**) 에 표시됩니다. 이 **ImageContentSourcePolicy** 가 구성된 클러스터와 다른 **ImageContentSourcePolicy** 경우 다시 시작됩니다.

```
cloudctl case launch \
--case $HOME/offline/${CASE_ARCHIVE} \
--inventory ${CASE_INVENTORY} \
--action configure-cluster-airgap \
--namespace ${NAMESPACE} \
--args "--registry ${LOCAL_DOCKER_REGISTRY} --user ${LOCAL_DOCKER_USER} --pass ${LOCAL_DOCKER_PASSWORD} --inputDir $HOME/offline --dryRun"
```

- b. 글로벌 이미지 풀 시크릿 및 **ImageContentSourcePolicy**를 구성하려면 `--dry - run` 옵션 없이 이 단계에 대한 코드를 실행하십시오.

```
cloudctl case launch \
--case $HOME/offline/${CASE_ARCHIVE} \
--inventory ${CASE_INVENTORY} \
--action configure-cluster-airgap \
--namespace ${NAMESPACE} \
--args "--registry ${LOCAL_DOCKER_REGISTRY} --user ${LOCAL_DOCKER_USER} --pass ${LOCAL_DOCKER_PASSWORD} --inputDir $HOME/offline"
```

5. **ImageContentSourcePolicy** 자원이 작성되었는지 확인하십시오.

```
oc get imageContentSourcePolicy
```

- 선택사항: 비보안 레지스트리를 사용하는 경우, 로컬 레지스트리를 클러스터 **insecureRegistries**에 추가해야 합니다.

```
oc patch image.config.openshift.io/cluster --type=merge -p '{"spec":{"registrySources":{"insecureRegistries":["${LOCAL_DOCKER_REGISTRY}"]}}}'
```

- 클러스터 노드 상태를 확인하십시오.

```
oc get nodes
```

imageContentsourcePolicy 및 글로벌 이미지 pull 본인확인정보가 적용된 후, **Ready, Scheduling** 또는 **Disabled**로 노드 상태를 참조할 수 있습니다. 모든 노드가 **Ready** 상태를 표시할 때까지 기다리십시오.

- 이미지를 로컬 레지스트리에 미러링합니다.

```
cloudctl case launch \
--case $HOME/offline/${CASE_ARCHIVE} \
--inventory ${CASE_INVENTORY} \
--action mirror-images \
--namespace ${NAMESPACE} \
--args "--registry ${LOCAL_DOCKER_REGISTRY} --user ${LOCAL_DOCKER_USER} --pass $
${LOCAL_DOCKER_PASSWORD} --inputDir $HOME/offline"
```

IBM MQ Operator를 설치하십시오.

- Red Hat OpenShift 클러스터 웹 콘솔에 로그인하십시오.
- 카탈로그 소스를 작성하십시오. 이전 단계를 실행한 것과 동일한 터미널을 사용하십시오.

```
cloudctl case launch \
--case $HOME/offline/${CASE_ARCHIVE} \
--inventory ${CASE_INVENTORY} \
--action install-catalog \
--namespace ${NAMESPACE} \
--args "--registry ${LOCAL_DOCKER_REGISTRY} --recursive"
```

- Common Services 설치 프로그램 Operator에 대해 **CatalogSource**가 작성되었는지 확인하십시오.

```
oc get pods -n openshift-marketplace
oc get catalogsource -n openshift-marketplace
```

- OLM을 사용하여 IBM MQ Operator를 설치하십시오.

- 탐색 패널에서 **Operators > OperatorHub**를 클릭하십시오.

OperatorHub 페이지가 표시됩니다.

- 모든 항목 필드에 IBM MQ를 입력하십시오.

IBM MQ 카탈로그 입력 항목이 표시됩니다.

- IBM MQ**를 선택하십시오.

"**IBM MQ**" 창이 표시됩니다.

- 설치를 클릭하십시오.

Operator 구독 작성 페이지가 표시됩니다.

- 선택할 Operator 채널을 판별하려면 7 페이지의 『[IBM MQ Operator에 대한 버전 지원](#)』의 내용을 검토하십시오.

- 설치 모드를 작성한 특정 네임스페이스 또는 클러스터 전체 범위로 설정하십시오.

- 구독을 클릭하십시오.

IBM MQ는 **Installed Operators** 페이지에 추가됩니다.

- 설치된 **Operator** 페이지에서 Operator의 상태를 확인하십시오. 이 상태는 설치가 완료되면 **Succeeded**로 변경됩니다.

IBM MQ Queue Manager 배치

설치된 연산자에서 새 큐 관리자를 작성하려면 75 페이지의 『IBM MQ Operator를 사용하여 큐 관리자 배치 및 구성』의 내용을 참조하십시오.

관련 태스크

68 페이지의 『에어갭 환경에서 IBM MQ Operator 또는 큐 관리자 업그레이드 준비』

인터넷 연결이 없는 Red Hat OpenShift 클러스터에는 IBM MQ Operator를 업그레이드하기 전에 수행해야 하는 준비 단계가 있습니다.

OpenShift

CP4I

IBM MQ Operator 및 큐 관리자 업그레이드

IBM MQ Operator를 사용하면 큐 관리자를 업그레이드할 수 있습니다.

프로시저

- 71 페이지의 『Red Hat OpenShift 웹 콘솔을 사용하여 IBM MQ Operator 업그레이드』.
- 72 페이지의 『Red Hat OpenShift CLI를 사용하여 IBM MQ Operator 업그레이드』.
- 74 페이지의 『Red Hat OpenShift 웹 콘솔을 사용하여 IBM MQ 큐 관리자 업그레이드』.
- 74 페이지의 『Red Hat OpenShift CLI를 사용하여 IBM MQ 큐 관리자 업그레이드』.

OpenShift

CP4I

Linux

에어갭 환경에서 IBM MQ Operator 또는 큐 관리자

업그레이드 준비

인터넷 연결이 없는 Red Hat OpenShift 클러스터에는 IBM MQ Operator를 업그레이드하기 전에 수행해야 하는 준비 단계가 있습니다.

시작하기 전에

이 주제에서는 이전에 릴리스된 IBM Cloud Pak for Integration 이미지가 미러링되는 로컬 이미지 레지스트리를 이미 구성했다고 가정합니다.

이 태스크 정보

에어 갭 환경에서 IBM MQ Operator 또는 큐 관리자를 업그레이드하려면 먼저 최신 IBM Cloud Pak for Integration 이미지를 미러링해야 합니다.

이 태스크의 처음 네 단계는 62 페이지의 『에어 갭 환경에 IBM MQ Operator 설치』인 경우 수행하는 단계와 동일합니다.

프로시저

1. 설치 프로그램과 이미지 자원 명세를 위한 환경 변수를 작성하십시오.

설치 프로그램 이미지 이름 및 이미지 인벤토리를 사용하여 다음 환경 변수를 작성하십시오.

```
export CASE_ARCHIVE_VERSION=version_number
export CASE_ARCHIVE=ibm-mq-$CASE_ARCHIVE_VERSION.tgz
export CASE_INVENTORY=ibmMQOperator
```

여기서 *version_number*는 에어 갭 설치를 위해 사용할 케이스의 버전입니다. 사용 가능한 케이스 버전의 목록은 <https://github.com/IBM/cloud-pak/tree/master/repo/case/ibm-mq>를 참조하십시오. IBM MQ Operator에 대한 버전 지원을 검토하여 선택할 운영자 채널을 판별하십시오.

2. IBM MQ 설치 프로그램 및 이미지 인벤토리를 다운로드하십시오.

ibm-mq 설치 프로그램 및 이미지 자원 명세를 bastion 호스트에 다운로드하십시오.

```
cloudctl case save \  
--case https://github.com/IBM/cloud-pak/raw/master/repo/case/ibm-mq/
```

```
$CASE_ARCHIVE_VERSION/$CASE_ARCHIVE \  
--outputdir $HOME/offline/
```

3. Red Hat OpenShift Container Platform 클러스터에 클러스터 관리자로 로그인하십시오.

다음은 Red Hat OpenShift Container Platform 클러스터에 로그인하기 위한 예 명령입니다.

```
oc login cluster_host:port --username=cluster_admin_user --password=cluster_admin_password
```

4. 이미지를 미러링하고 클러스터를 구성하십시오.

다음 단계를 완료하여 이미지를 미러링하고 클러스터를 구성합니다.

참고: 명령에서 큰따옴표 내에 물결 기호를 사용하지 마십시오. 예를 들어, args "--registry registry --user registry_userid --pass registry_password --inputDir ~/offline"을(를) 사용하지 마십시오. 틸드가 확장되지 않으며 명령이 실패할 수 있습니다.

a. 모든 소스 Docker 레지스트리에 대한 인증 신임 정보를 저장합니다.

모든 IBM Cloud Platform Common Services, IBM MQ Operator 이미지 및 IBM MQ Advanced Developer 이미지는 인증이 필요하지 않은 공용 레지스트리에 저장됩니다. 그러나 IBM MQ Advanced Server(비 개발자), 기타 제품 및 써드파티 컴포넌트는 하나 이상의 인증된 레지스트리를 필요로 합니다. 다음 레지스트리에는 인증이 필요합니다.

- cp.icr.io
- registry.redhat.io
- registry.access.redhat.com

이러한 레지스트리에 대한 자세한 정보는 [레지스트리 네임스페이스](#) 작성을 참조하십시오.

인증이 필요한 모든 레지스트리에 대한 신임 정보를 구성하려면 다음 명령을 실행해야 합니다. 각 레지스트리에 대해 개별적으로 명령을 실행하십시오.

```
cloudctl case launch \  
--case $HOME/offline/${CASE_ARCHIVE} \  
--inventory ${CASE_INVENTORY} \  
--action configure-creds-airgap \  
--namespace ${NAMESPACE} \  
--args "--registry registry --user registry_userid --pass registry_password --inputDir $HOME/offline"
```

명령은 \$HOME/.airgap/secrets 위치의 파일 시스템의 파일에 레지스트리 신임 정보를 저장하고 캐시합니다.

b. 로컬 Docker 레지스트리 연결 정보로 환경 변수를 작성합니다.

```
export LOCAL_DOCKER_REGISTRY=IP_or_FQDN_of_local_docker_registry  
export LOCAL_DOCKER_USER=username  
export LOCAL_DOCKER_PASSWORD=password
```

참조: Docker 레지스트리는 80 또는 443과 같은 표준 포트를 사용합니다. Docker 레지스트리는 비 표준 포트를 사용하는 경우, 구문 host:port를 사용하여 포트를 지정하십시오. 예를 들면, 다음과 같습니다.

```
export LOCAL_DOCKER_REGISTRY=myregistry.local:5000
```

c. 로컬 Docker 레지스트리에 대한 인증 본인확인정보를 구성하십시오.

참고: 이 단계는 한 번만 수행해야 합니다.

```
cloudctl case launch \  
--case $HOME/offline/${CASE_ARCHIVE} \  
--inventory ${CASE_INVENTORY} \  
--action configure-creds-airgap \  
--namespace ${NAMESPACE} \  
--args "--registry ${LOCAL_DOCKER_REGISTRY} --user ${LOCAL_DOCKER_USER} --pass ${LOCAL_DOCKER_PASSWORD}"
```

명령은 \$HOME/.airgap/secrets 위치의 파일 시스템의 파일에 레지스트리 신임 정보를 저장하고 캐시합니다.

d. 글로벌 이미지 pull 본인확인정보 및 **ImageContentSourcePolicy**를 구성하십시오.

i) 노드를 다시 시작해야 하는지 여부를 확인하십시오.

- Red Hat OpenShift Container Platform 버전 4.4 이상 및 에어 갭을 사용하는 IBM MQ Operator의 새 설치에서 이 단계는 모든 클러스터 노드를 다시 시작합니다. 클러스터 자원은 새 풀 본인확인정보가 적용될 때까지 사용 불가능할 수 있습니다.
- IBM MQ Operator 1.8에서 CASE는 이미지에 대한 추가 미러링 소스를 포함하도록 업데이트됩니다. 따라서 이전 버전의 IBM MQ Operator에서 버전 1.8 이상으로 업그레이드하면 노드 다시 시작이 트리거됩니다.
- 이 단계에서 노드를 다시 시작해야 하는지 확인하려면 이 단계의 코드에 `--dry - run` 옵션을 추가하십시오. 그러면 최신 **ImageContentSourcePolicy**가 생성되어 콘솔 창 (**stdout**)에 표시됩니다. 이 **ImageContentSourcePolicy**가 구성된 클러스터와 다른 **ImageContentSourcePolicy**경우 다시 시작됩니다.

```
cloudctl case launch \  
--case $HOME/offline/${CASE_ARCHIVE} \  
--inventory ${CASE_INVENTORY} \  
--action configure-cluster-airgap \  
--namespace ${NAMESPACE} \  
--args "--registry ${LOCAL_DOCKER_REGISTRY} --user ${LOCAL_DOCKER_USER} --pass $  
{LOCAL_DOCKER_PASSWORD} --inputDir $HOME/offline --dryRun"
```

ii) 글로벌 이미지 풀 시크릿 및 **ImageContentSourcePolicy**를 구성하려면 `--dry - run` 옵션 없이 이 단계에 대한 코드를 실행하십시오.

```
cloudctl case launch \  
--case $HOME/offline/${CASE_ARCHIVE} \  
--inventory ${CASE_INVENTORY} \  
--action configure-cluster-airgap \  
--namespace ${NAMESPACE} \  
--args "--registry ${LOCAL_DOCKER_REGISTRY} --user ${LOCAL_DOCKER_USER} --pass $  
{LOCAL_DOCKER_PASSWORD} --inputDir $HOME/offline"
```

e. **ImageContentSourcePolicy** 자원이 작성되었는지 확인하십시오.

```
oc get imageContentSourcePolicy
```

f. 선택사항: 비보안 레지스트리를 사용하는 경우, 로컬 레지스트리를 클러스터 **insecureRegistries**에 추가해야 합니다.

```
oc patch image.config.openshift.io/cluster --type=merge -p '{"spec":{"registrySources":  
{"insecureRegistries":["${LOCAL_DOCKER_REGISTRY}"]}}'
```

g. 클러스터 노드 상태를 확인하십시오.

```
oc get nodes
```

imageContentsourcePolicy 및 글로벌 이미지 pull 본인확인정보가 적용된 후, **Ready**, **Scheduling** 또는 **Disabled**로 노드 상태를 참조할 수 있습니다. 모든 노드가 **Ready** 상태를 표시할 때까지 기다리십시오.

h. 이미지를 로컬 레지스트리에 미러링합니다.

```
cloudctl case launch \  
--case $HOME/offline/${CASE_ARCHIVE} \  
--inventory ${CASE_INVENTORY} \  
--action mirror-images \  
--namespace ${NAMESPACE} \  
--args "--registry ${LOCAL_DOCKER_REGISTRY} --user ${LOCAL_DOCKER_USER} --pass $  
{LOCAL_DOCKER_PASSWORD} --inputDir $HOME/offline"
```

5. 카탈로그 소스를 업그레이드하십시오.

이전 단계를 실행한 것과 동일한 터미널을 사용하십시오.

```
cloudctl case launch \
--case $HOME/offline/${CASE_ARCHIVE} \
--inventory ${CASE_INVENTORY} \
--action install-catalog \
--namespace ${NAMESPACE} \
--args "--registry ${LOCAL_DOCKER_REGISTRY} --recursive"
```

다음에 수행할 작업

이제 다음 태스크 중 하나를 완료하여 IBM MQ Operator 및 큐 관리자를 업그레이드할 준비가 되었습니다.

- [71 페이지의 『Red Hat OpenShift 웹 콘솔을 사용하여 IBM MQ Operator 업그레이드』](#)
- [72 페이지의 『Red Hat OpenShift CLI를 사용하여 IBM MQ Operator 업그레이드』](#)
- [74 페이지의 『Red Hat OpenShift 웹 콘솔을 사용하여 IBM MQ 큐 관리자 업그레이드』](#)
- [74 페이지의 『Red Hat OpenShift CLI를 사용하여 IBM MQ 큐 관리자 업그레이드』](#)
- [75 페이지의 『Platform Navigator를 사용하여 Red Hat OpenShift에서 IBM MQ 큐 관리자 업그레이드』](#)

Red Hat OpenShift 웹 콘솔을 사용하여 IBM MQ Operator 업그레이드

IBM MQ Operator는 OperatorHub를 사용하여 업그레이드할 수 있습니다.

시작하기 전에

Red Hat OpenShift 클러스터 웹 콘솔에 로그인하십시오.

에어 갭 환경에서 IBM MQ Operator 를 업그레이드하려면 먼저 최신 IBM Cloud Pak for Integration 이미지를 미리링해야 합니다. [에어 갭 환경에서 IBM MQ Operator 또는 큐 관리자 업그레이드 준비를 참조하십시오.](#)

프로시저

1. [7 페이지의 『IBM MQ Operator에 대한 버전 지원』](#) 를 검토하여 업그레이드하는 Operator를 판별하십시오.
2. 옵션: 1.5보다 오래된 IBM MQ Operator 버전에서 IBM MQ Operator 1.5 이상으로 업그레이드하는 경우, 먼저 IBM Cloud Pak foundational services 버전을 업그레이드해야 합니다.

자세한 정보는 [72 페이지의 『Red Hat OpenShift 웹 콘솔을 사용하여 IBM Cloud Pak foundational services 업그레이드』](#) 의 내용을 참조하십시오.

3. IBM MQ Operator를 업그레이드하십시오. 새 주 또는 부 IBM MQ Operator 버전은 새 등록 채널을 통해 전달됩니다. 운영자를 새 주 버전 또는 부 버전으로 업그레이드하려면 IBM MQ Operator 등록에서 선택한 채널을 업데이트해야 합니다.
 - a) 탐색 패널에서 **연산자 > 설치된 연산자**를 클릭하십시오.
지정된 프로젝트에 설치된 모든 연산자가 표시됩니다.
 - b) **IBM MQ Operator**를 선택하십시오.
 - c) **구독 탭**으로 이동하십시오.
 - d) **채널**을 클릭하십시오.
구독 업데이트 채널 변경 창이 표시됩니다.
 - e) 원하는 채널을 선택하고 **저장**을 클릭합니다.
연산자는 새 채널에서 사용 가능한 최신 버전으로 업그레이드합니다. [7 페이지의 『IBM MQ Operator에 대한 버전 지원』](#) 의 내용을 참조하십시오.

다음에 수행할 작업

IBM Cloud Pak foundational services 3.7로 업그레이드한 경우 IBM Cloud Pak for Integration 라이선스를 사용하는 큐 관리자는 업그레이드하거나 재시작해야 합니다. 이를 수행하는 방법에 대한 자세한 정보는 [74 페이지의 『Red Hat OpenShift 웹 콘솔을 사용하여 IBM MQ 큐 관리자 업그레이드』](#) 의 내용을 참조하십시오.

foundational services 업그레이드

1.5보다 오래된 IBM MQ Operator 버전에서 IBM MQ Operator 1.5 이상으로 업그레이드하는 경우, 먼저 IBM Cloud Pak foundational services 버전을 업그레이드해야 합니다.

시작하기 전에

참고: 1.5보다 오래된 IBM MQ Operator 버전에서 IBM MQ Operator 1.5 이상으로 업그레이드하는 경우에만 이 태스크를 완료해야 합니다.

CP4I

IBM Cloud Pak for Integration 라이선스를 사용하는 큐 관리자가 있는 경우, 이 업그레이드 후 웹 콘솔에 액세스하려면 큐 관리자를 재시작해야 하며 웹 콘솔에 [기타 오류](#) 로깅이 표시됩니다. 운영자 업그레이드가 완료된 후 선택한 IBM MQ 버전에 대해 `.spec.version`의 최신 값으로 업그레이드하여 이러한 오류를 수정할 수 있습니다.

CP4I

기존 큐 관리자가 있고 IBM Cloud Pak for Integration 조작 대시보드를 사용하는 경우 업그레이드하기 전에 105 페이지의 『IBM Cloud Pak for Integration 2021.4에서 조작 대시보드 통합으로 IBM MQ 9.2.2 또는 9.2.3 배치 또는 업그레이드』의 내용을 참조하십시오.

프로시저

1. Red Hat OpenShift 클러스터 웹 콘솔에 로그인하십시오.
2. 탐색 패널에서 **연산자 > 설치된 연산자**를 클릭하십시오.
지정된 프로젝트에 설치된 모든 연산자가 표시됩니다.
3. **IBM Cloud Pak foundational services Operator**를 선택하십시오. 버전 3.7 이전에는 이를 **IBM Common Services Operator**라고 했습니다.
4. 등록 탭으로 이동하십시오.
5. 채널을 클릭하십시오.
구독 업데이트 채널 변경 창이 표시됩니다.
6. **v3** 채널을 선택한 후 **저장**을 클릭하십시오.
IBM Cloud Pak foundational services 운영자는 새 채널에서 사용 가능한 최신 버전으로 업그레이드됩니다.
7 페이지의 『IBM MQ Operator에 대한 버전 지원』을 참조하십시오.

다음에 수행할 작업

이제 IBM MQ Operator을(를) 업그레이드할 준비가 되었습니다.

이드

명령행에서 IBM MQ Operator을(를) 업그레이드할 수 있습니다.

시작하기 전에

cloudctl login(IBM Cloud Pak for Integration의 경우) 또는 **oc login**을 사용하여 클러스터에 로그인하십시오.

에어 갭 환경에서 IBM MQ Operator 를 업그레이드하려면 먼저 최신 IBM Cloud Pak for Integration 이미지를 미러링해야 합니다. [에어 갭 환경에서 IBM MQ Operator 또는 큐 관리자 업그레이드 준비](#)를 참조하십시오.

프로시저

1. 7 페이지의 『IBM MQ Operator에 대한 버전 지원』를 검토하여 업그레이드하는 Operator를 판별하십시오.
2. 옵션: 1.5보다 오래된 IBM MQ Operator 버전에서 IBM MQ Operator 1.5 이상으로 업그레이드하는 경우, 먼저 IBM Cloud Pak foundational services 버전을 업그레이드해야 합니다.

자세한 정보는 73 페이지의 『Red Hat OpenShiftCLI를 사용하여 IBM Cloud Pak foundational services 업그레이드』의 내용을 참조하십시오.

3. IBM MQ Operator를 업그레이드하십시오. 새로운 주/부 IBM MQ Operator 버전이 새 구독 채널을 통해 전달됩니다. Operator를 새로운 주/부 버전으로 업그레이드하려면, IBM MQ Operator 구독에서 선택된 채널을 업데이트해야 합니다.

- a) 필수 IBM MQ Operator 업그레이드 채널이 사용 가능한지 확인하십시오.

```
oc get packagemanifest ibm-mq -o=jsonpath='{.status.channels[*].name}'
```

- b) Subscription을(를) 패치하여 원하는 업데이트 채널로 이동합니다. 여기서 vX.Y는 이전 단계에서 식별된 원하는 업데이트 채널입니다.

```
oc patch subscription ibm-mq --patch '{"spec":{"channel":"vX.Y"}}' --type=merge
```

다음에 수행할 작업

IBM Cloud Pak foundational services 3.7로 업그레이드한 경우 IBM Cloud Pak for Integration 라이선스를 사용하는 큐 관리자는 업그레이드하거나 재시작해야 합니다. 이를 수행하는 방법에 대한 자세한 정보는 74 페이지의 『Red Hat OpenShift CLI를 사용하여 IBM MQ 큐 관리자 업그레이드』의 내용을 참조하십시오.

Red Hat OpenShiftCLI를 사용하여 IBM Cloud Pak foundational services 업그레이드

1.5보다 오래된 IBM MQ Operator 버전에서 IBM MQ Operator 1.5 이상으로 업그레이드하는 경우, 먼저 IBM Cloud Pak foundational services 버전을 업그레이드해야 합니다.

시작하기 전에

참고: 1.5보다 오래된 IBM MQ Operator 버전에서 IBM MQ Operator 1.5 이상으로 업그레이드하는 경우에만 이 작업을 완료해야 합니다.

 IBM Cloud Pak for Integration 라이선스를 사용하는 큐 관리자가 있는 경우, 이 업그레이드 이후 웹 콘솔에 액세스하려면 큐 관리자를 재시작해야 합니다. 웹 콘솔에 대한 기타 오류 로깅도 참조하십시오. 운영자 업그레이드가 완료된 후 선택한 IBM MQ 버전에 대해 .spec.version의 최신 값으로 업그레이드하여 이러한 오류를 수정할 수 있습니다.

 기존 큐 관리자가 있고 IBM Cloud Pak for Integration 조작 대시보드를 사용하는 경우 업그레이드하기 전에 105 페이지의 『IBM Cloud Pak for Integration 2021.4에서 조작 대시보드 통합으로 IBM MQ 9.2.2 또는 9.2.3 배치 또는 업그레이드』의 내용을 참조하십시오.

프로시저

1. **cloudctl login**(IBM Cloud Pak for Integration의 경우) 또는 **oc login**을 사용하여 클러스터에 로그인하십시오.
2. v3 IBM Cloud Pak foundational services 업그레이드 채널이 사용 가능한지 확인하십시오.

```
oc get packagemanifest -n ibm-common-services ibm-common-service-operator -o=jsonpath='{.status.channels[*].name}'
```

3. Subscription을(를) 패치하여 원하는 업데이트 채널로 이동하십시오v3

```
oc patch subscription ibm-common-service-operator --patch '{"spec":{"channel":"v3"}}' --type=merge
```

다음에 수행할 작업

이제 IBM MQ Operator을(를) 업그레이드할 준비가 되었습니다.

레이드

IBM MQ Operator를 사용하여 배치된 IBM MQ 큐 관리자는 Red Hat OpenShift에서 Operator Hub를 사용하여 업그레이드할 수 있습니다.

시작하기 전에

- Red Hat OpenShift 클러스터 웹 콘솔에 로그인하십시오.
- IBM MQ Operator가 원하는 업데이트 채널을 사용하고 있는지 확인하십시오. [68 페이지의 『IBM MQ Operator 및 큐 관리자 업그레이드』](#) 을 참조하십시오.

에어 갭 환경에서 큐 관리자를 업그레이드하려면 먼저 최신 IBM Cloud Pak for Integration 이미지를 미러링해야 합니다. [에어 갭 환경에서 IBM MQ Operator 또는 큐 관리자 업그레이드 준비를 참조하십시오.](#)

프로시저

1. 탐색 패널에서 **연산자 > 설치된 연산자**를 클릭하십시오.
지정된 프로젝트에 설치된 모든 연산자가 표시됩니다.
2. **IBM MQ Operator**를 선택하십시오.
IBM MQ Operator 창이 표시됩니다.
3. **큐 관리자** 탭으로 이동하십시오.
큐 관리자 세부사항 창이 표시됩니다.
4. 업그레이드할 큐 관리자를 선택하십시오.
5. YAML 탭으로 이동하십시오.
6. 원하는 IBM MQ 큐 관리자 버전 업그레이드와 일치하도록 다음 필드를 업데이트하십시오(필요한 경우).
 - spec.version
 - spec.license.licence
 IBM MQ Operator 버전 및 IBM MQ 큐 관리자 버전에 대한 채널 맵핑은 [7 페이지의 『IBM MQ Operator에 대한 버전 지원』](#) 를 참조하십시오.
7. 업데이트된 큐 관리자 YAML을 저장하십시오.

이드

IBM MQ Operator를 사용하여 배치된 IBM MQ 큐 관리자는 Red Hat OpenShift에서 명령행을 사용하여 업그레이드할 수 있습니다.

시작하기 전에

이러한 단계를 완료하려면 클러스터 관리자여야 합니다.

- `oc login`를 사용하여 Red Hat OpenShift 명령행 인터페이스 (CLI) 에 로그인하십시오.
- IBM MQ Operator가 원하는 업데이트 채널을 사용하고 있는지 확인하십시오. [68 페이지의 『IBM MQ Operator 및 큐 관리자 업그레이드』](#) 을 참조하십시오.

에어 갭 환경에서 큐 관리자를 업그레이드하려면 먼저 최신 IBM Cloud Pak for Integration 이미지를 미러링해야 합니다. [에어 갭 환경에서 IBM MQ Operator 또는 큐 관리자 업그레이드 준비를 참조하십시오.](#)

프로시저

QueueManager 자원을 편집하여 원하는 IBM MQ 큐 관리자 버전 업그레이드와 일치하도록 다음 필드를 업데이트하십시오(필요한 경우).

- spec.version

- spec.license.licence

IBM MQ Operator 버전 및 IBM MQ 큐 관리자 버전에 대한 채널 매핑은 7 페이지의 『IBM MQ Operator에 대한 버전 지원』를 참조하십시오.

다음 명령을 사용하십시오.

```
oc edit queuemanager my_qmgr
```

여기서 `my_qmgr`은 업그레이드할 QueueManager 자원의 이름입니다.

CP4I Platform Navigator를 사용하여 Red Hat OpenShift에서 IBM MQ 큐 관리자 업그레이드

IBM MQ Operator를 사용하여 배치된 IBM MQ 큐 관리자는 Red Hat OpenShift에서 IBM Cloud Pak for Integration Platform Navigator를 사용하여 업그레이드할 수 있습니다.

시작하기 전에

- 업그레이드할 큐 관리자를 포함하는 네임스페이스에서 IBM Cloud Pak for Integration Platform Navigator에 로그인하십시오.
- IBM MQ Operator가 원하는 업데이트 채널을 사용하고 있는지 확인하십시오. 68 페이지의 『IBM MQ Operator 및 큐 관리자 업그레이드』을 참조하십시오.

에어 갭 환경에서 큐 관리자를 업그레이드하려면 먼저 최신 IBM Cloud Pak for Integration 이미지를 미러링해야 합니다. [에어 갭 환경에서 IBM MQ Operator 또는 큐 관리자 업그레이드 준비](#)를 참조하십시오.

프로시저

1. IBM Cloud Pak for Integration Platform Navigator 홈 페이지에서 **런타임** 탭을 클릭하십시오.
2. 사용 가능한 업그레이드가 있는 큐 관리자는 **버전** 옆에 파란색 **i**가 있습니다. **i**를 클릭하여 **새 버전 사용 가능**을 표시하십시오.
3. 업그레이드할 큐 관리자의 오른쪽 가장자리에 있는 점 세 개를 클릭한 후 **버전 변경**을 클릭하십시오.
4. **새 채널 또는 버전 선택**에서 필요한 업그레이드 버전을 선택하십시오.
5. **버전 변경**을 클릭하십시오.

결과

큐 관리자가 업그레이드됩니다.

OpenShift CP4I IBM MQ Operator를 사용하여 큐 관리자 배치 및 구성

IBM MQ 9.1.5 이상은 IBM MQ Operator를 사용하여 Red Hat OpenShift에 배치됩니다.

이 태스크 정보

프로시저

- 75 페이지의 『IBM MQ를 위한 Red Hat OpenShift 프로젝트 준비』.
- 77 페이지의 『Red Hat OpenShift Container Platform 클러스터에 큐 관리자 배치』.

OpenShift CP4I IBM MQ를 위한 Red Hat OpenShift 프로젝트 준비

큐 관리자를 배치할 준비를 하기 위해 Red Hat OpenShift Container Platform 클러스터를 준비하십시오.

프로시저

- 76 페이지의 『Red Hat OpenShift 웹 콘솔을 사용하여 IBM MQ용 Red Hat OpenShift 프로젝트 준비』.

- 76 페이지의 『Red Hat OpenShift CLI를 사용하여 IBM MQ용 Red Hat OpenShift 프로젝트 준비』.

관련 태스크

77 페이지의 『Red Hat OpenShift Container Platform 클러스터에 큐 관리자 배치』

QueueManager 사용자 정의 자원을 사용하여 Red Hat OpenShift Container Platform 클러스터에 큐 관리자를 배치하십시오.

Red Hat OpenShift 웹 콘솔을 사용하여 IBM MQ용 Red Hat OpenShift 프로젝트 준비

Prepare your Red Hat OpenShift Container Platform cluster, so that it's ready to deploy a queue manager using the IBM MQ Operator. 이 태스크는 프로젝트 관리자에서 완료해야 합니다.

시작하기 전에

참고: 다른 IBM Cloud Pak for Integration 구성요소가 이미 설치된 프로젝트에서 IBM MQ 를 사용하려는 경우 다음 지시사항을 따를 필요가 없습니다.

Red Hat OpenShift 클러스터 웹 콘솔에 로그인하십시오.

이 태스크 정보

라이선스 부여 검사를 수행하는 컨테이너 레지스트리에서 IBM MQ Operator 이미지를 가져옵니다. 이 검사에는 docker-registry pull 시크릿에 저장되는 인타이틀먼트 키가 필요합니다. 아직 인타이틀먼트 키가 없는 경우 다음 지시사항에 따라 인타이틀먼트 키를 가져오고 풀 시크릿을 작성하십시오.

프로시저

1. 사용자 ID에 지정된 인타이틀먼트 키를 가져오십시오.
 - a) 자격이 있는 소프트웨어와 연관된 IBM ID 및 비밀번호를 사용하여 [MyIBM](#) 에 로그인하십시오.
 - b) 인타이틀먼트 키 섹션에서 키 복사를 선택하여 인타이틀먼트 키를 클립보드에 복사하십시오.
2. 큐 관리자를 배치할 프로젝트에서 인타이틀먼트 키를 포함하는 본인확인정보를 작성하십시오.
 - a) 탐색 패널에서 워크로드 > 본인확인정보에서 클릭하십시오.
본인확인정보 페이지가 표시됩니다.
 - b) 프로젝트 드롭다운에서, IBM MQ를 설치할 프로젝트를 선택하십시오.
 - c) 작성 단추를 클릭하고 이미지 풀 본인확인정보를 선택하십시오.
 - d) 이름 필드에 ibm-entitlement-key를 입력하십시오.
 - e) 레지스트리 서버 주소 필드에 cp.icr.io를 입력하십시오.
 - f) 사용자 이름 필드에 cp를 입력하십시오.
 - g) 비밀번호 필드에 이전 단계에서 복사한 인타이틀먼트 키를 입력하십시오.
 - h) 이메일 필드에 인타이틀먼트 소프트웨어와 연관된 IBM ID를 입력하십시오.

다음에 수행할 작업

78 페이지의 『Red Hat OpenShift 웹 콘솔을 사용하여 큐 관리자 배치』

Red Hat OpenShift CLI를 사용하여 IBM MQ용 Red Hat OpenShift 프로젝트 준비

Prepare your Red Hat OpenShift Container Platform cluster, so that it's ready to deploy a queue manager using the IBM MQ Operator. 이 태스크는 프로젝트 관리자에서 완료해야 합니다.

시작하기 전에

참고: 다른 IBM Cloud Pak for Integration 구성요소가 이미 설치된 프로젝트에서 IBM MQ 를 사용하려는 경우 다음 지시사항을 따를 필요가 없습니다.

cloudctl login(IBM Cloud Pak for Integration의 경우) 또는 **oc login**을 사용하여 클러스터에 로그인하십시오.

이 태스크 정보

라이선스 부여 검사를 수행하는 컨테이너 레지스트리에서 IBM MQ Operator 이미지를 가져옵니다. 이 검사에는 `docker-registry pull` 시크릿에 저장되는 인타이틀먼트 키가 필요합니다. 아직 인타이틀먼트 키가 없는 경우 다음 지시사항에 따라 인타이틀먼트 키를 가져오고 풀 시크릿을 작성하십시오.

프로시저

1. 사용자 ID에 지정된 인타이틀먼트 키를 가져오십시오.
 - a) 자격이 있는 소프트웨어와 연관된 IBM ID 및 비밀번호를 사용하여 [MyIBM](#)에 로그인하십시오.
 - b) **인타이틀먼트 키** 섹션에서 **키 복사**를 선택하여 인타이틀먼트 키를 클립보드에 복사하십시오.
2. 큐 관리자를 배치할 프로젝트에서 인타이틀먼트 키를 포함하는 본인확인정보를 작성하십시오.
다음 명령을 실행하십시오. 여기서 `<entitlement-key>`는 1단계에서 검색된 키이고 `<user-email>`은 권한 부여된 소프트웨어와 연관된 IBM ID입니다.

```
oc create secret docker-registry ibm-entitlement-key \
--docker-server=cp.icr.io \
--docker-username=cp \
--docker-password=<entitlement-key> \
--docker-email=<user-email>
```

다음에 수행할 작업

79 페이지의 [『Red Hat OpenShift CLI를 사용하여 큐 관리자 배치』](#)

Red Hat OpenShift Container Platform 클러스터에 큐 관리자 배치

QueueManager 사용자 정의 자원을 사용하여 Red Hat OpenShift Container Platform 클러스터에 큐 관리자를 배치하십시오.

프로시저

-  77 페이지의 [『IBM Cloud Pak for Integration Platform Navigator를 사용하여 큐 관리자 배치』](#).
-  78 페이지의 [『Red Hat OpenShift 웹 콘솔을 사용하여 큐 관리자 배치』](#).
-  79 페이지의 [『Red Hat OpenShift CLI를 사용하여 큐 관리자 배치』](#).

관련 태스크

81 페이지의 [『큐 관리자 구성의 예』](#)

큐 관리자는 QueueManager 사용자 정의 자원의 콘텐츠를 조정하여 구성할 수 있습니다.

IBM Cloud Pak for Integration Platform Navigator를 사용하여 큐 관리자 배치

QueueManager 사용자 정의 자원을 사용하여 Red Hat OpenShift를 사용하는 IBM Cloud Pak for Integration Platform Navigator Container Platform 클러스터에 큐 관리자를 배치하십시오. 프로젝트 관리자가 이 태스크를 완료해야 함

시작하기 전에

브라우저에서 IBM Cloud Pak for Integration Platform Navigator를 시작하십시오.

큐 관리자를 이 Red Hat OpenShift 프로젝트에 배치하는 것이 처음이면 75 페이지의 [『IBM MQ를 위한 Red Hat OpenShift 프로젝트 준비』](#)의 단계를 따르십시오.

프로시저

1. 큐 관리자를 배치하십시오.

다음 예는 "빠른 시작" 큐 관리자를 배치하며, 임시(비지속) 스토리지를 사용하며 MQ 보안을 끕니다. 메시지가 큐 관리자 재시작 시 지속되지 않습니다. 구성을 조정하여 많은 큐 관리자 설정을 변경할 수 있습니다.

a) IBM Cloud Pak for Integration Platform Navigator에서 **관리**를 클릭한 후 **통합 런타임**을 클릭하십시오. IBM Cloud Pak for Integration Platform Navigator의 이전 버전에서 **런타임 및 인스턴스**를 클릭하십시오.

b) **인스턴스 작성**을 클릭하십시오.

c) **메시징**을 선택하고 **다음**을 클릭하십시오. IBM Cloud Pak for Integration Platform Navigator의 이전 버전에서 **큐 관리자**를 클릭하고 **다음**을 클릭하십시오.

QueueManager의 인스턴스를 작성할 양식이 표시됩니다.

참고: 코드를 클릭하여 QueueManager 구성 YAML을 보거나 변경할 수도 있습니다.

d) **세부사항** 섹션에서 **이름** 필드를 확인하거나 업데이트하고 큐 관리자 인스턴스를 작성할 위치에서 **네임스페이스**를 지정하십시오.

e) IBM Cloud Pak for Integration 라이선스 계약을 승인하는 경우, **라이선스 수락**을 **킵**으로 변경하십시오. 큐 관리자를 배치하려면 라이선스를 승인해야 합니다.

f) **큐 관리자** 섹션에서 기본 큐 관리자의 **이름**을 확인하거나 업데이트하십시오. IBM Cloud Pak for Integration Platform Navigator의 이전 버전에서는 **큐 관리자 구성** 섹션을 사용하십시오.

기본적으로, IBM MQ 클라이언트 애플리케이션에서 사용된 큐 관리자의 이름은 QueueManager의 이름과 동일하지만 올바르게 읽은 문자(하이픈과 같이)가 제거됩니다.

g) **작성**을 클릭하십시오.

이제 현재 프로젝트(네임스페이스)의 큐 관리자 목록이 표시됩니다. 새 QueueManager는 Pending의 상태에 있어야 합니다.

2. 큐 관리자가 실행 중인지 확인

QueueManager 상태가 Running인 경우 작성이 완료됩니다.

관련 태스크

[102 페이지의 『Red Hat OpenShift 클러스터 외부에서 큐 관리자에 연결하도록 Route 구성』](#)

Red Hat OpenShift 클러스터 외부에서 IBM MQ 큐 관리자에 애플리케이션을 연결하려면 Red Hat OpenShift 라우트가 필요합니다. SNI는 TLS 1.2 이상의 프로토콜이 사용될 때 TLS 프로토콜에서만 사용 가능하므로 IBM MQ 큐 관리자 및 클라이언트 애플리케이션에서 TLS를 사용으로 설정해야 합니다. Red Hat OpenShift Container Platform Router에서는 IBM MQ 큐 관리자로 요청을 라우팅하는 데 SNI를 사용합니다.

[108 페이지의 『Red Hat OpenShift 클러스터에 배치된 IBM MQ Console에 연결』](#)

Red Hat OpenShift Container Platform 클러스터에 배치된 큐 관리자의 IBM MQ Console에 연결하는 방법입니다.

Red Hat OpenShift 웹 콘솔을 사용하여 큐 관리자 배치

QueueManager 사용자 정의 자원을 사용하여 Red Hat OpenShift 웹 콘솔을 사용하는 Red Hat OpenShift Container Platform 클러스터에 큐 관리자를 배치하십시오. 프로젝트 관리자가 이 태스크를 완료해야 함

시작하기 전에

Red Hat OpenShift 클러스터 웹 콘솔에 로그인하십시오. 사용할 기존 프로젝트(네임스페이스)를 선택하여 새 프로젝트를 작성해야 합니다.

큐 관리자를 이 Red Hat OpenShift 프로젝트에 배치하는 것이 처음이면 [75 페이지의 『IBM MQ를 위한 Red Hat OpenShift 프로젝트 준비』](#)의 단계를 따르십시오.

프로시저

1. 큐 관리자를 배치하십시오.

다음 예는 "빠른 시작" 큐 관리자를 배치하며, 임시(비지속) 스토리지를 사용하며 MQ 보안을 끕니다. 메시지가 큐 관리자 재시작 시 지속되지 않습니다. 구성을 조정하여 많은 큐 관리자 설정을 변경할 수 있습니다.

- a) Red Hat OpenShift 웹 콘솔의 탐색 분할창에서 **Operator > 설치된 Operator**를 클릭하십시오.
- b) **IBM MQ**를 클릭하십시오.
- c) 큐 관리자 탭을 클릭하십시오.
- d) 큐 관리자 작성 단추를 클릭하십시오.

QueueManager 자원의 예 YAML을 포함한 YAML 편집기가 표시됩니다.

참고: 또한 **양식 편집**을 클릭하여 QueueManager 구성을 보거나 변경할 수 있습니다.

- e) 라이선스 계약을 승인하는 경우, **라이선스 수락**을 **검**으로 변경하십시오.

IBM MQ는 여러 다른 라이선스에서 사용 가능합니다. 유효한 라이선스에 대한 자세한 정보는 [119 페이지의 『mq.ibm.com/v1beta1에 대한 라이선스 부여 참조』](#)의 내용을 참조하십시오. 큐 관리자를 배치하려면 라이선스를 승인해야 합니다.

- f) **작성**을 클릭하십시오.

이제 현재 프로젝트(네임스페이스)의 큐 관리자 목록이 표시됩니다. QueueManager는 Pending 상태에 있어야 합니다.

2. 큐 관리자가 실행 중인지 확인

QueueManager 상태가 Running인 경우 작성이 완료됩니다.

관련 태스크

[102 페이지의 『Red Hat OpenShift 클러스터 외부에서 큐 관리자에 연결하도록 Route 구성』](#)

Red Hat OpenShift 클러스터 외부에서 IBM MQ 큐 관리자에 애플리케이션을 연결하려면 Red Hat OpenShift 라우트가 필요합니다. SNI는 TLS 1.2 이상의 프로토콜이 사용될 때 TLS 프로토콜에서만 사용 가능하므로 IBM MQ 큐 관리자 및 클라이언트 애플리케이션에서 TLS를 사용으로 설정해야 합니다. Red Hat OpenShift Container Platform Router에서는 IBM MQ 큐 관리자로 요청을 라우팅하는 데 SNI를 사용합니다.

[108 페이지의 『Red Hat OpenShift 클러스터에 배치된 IBM MQ Console에 연결』](#)

Red Hat OpenShift Container Platform 클러스터에 배치된 큐 관리자의 IBM MQ Console 에 연결하는 방법입니다.

Red Hat OpenShift CLI를 사용하여 큐 관리자 배치

QueueManager 사용자 정의 자원을 사용하여 명령행 인터페이스(CLI)를 사용한 Red Hat OpenShift Container Platform 클러스터에 큐 관리자를 배치하십시오. 프로젝트 관리자가 이 태스크를 완료해야 함

시작하기 전에

Red Hat OpenShift Container Platform 명령행 인터페이스를 설치해야 합니다.

cloudctl login(IBM Cloud Pak for Integration의 경우) 또는 **oc login**을 사용하여 클러스터에 로그인하십시오.

큐 관리자를 이 Red Hat OpenShift 프로젝트에 배치하는 것이 처음이면 [75 페이지의 『IBM MQ를 위한 Red Hat OpenShift 프로젝트 준비』](#)의 단계를 따르십시오.

프로시저

1. 큐 관리자를 배치하십시오.

다음 예는 "빠른 시작" 큐 관리자를 배치하며, 임시(비지속) 스토리지를 사용하며 MQ 보안을 끕니다. 메시지가 큐 관리자 재시작 시 지속되지 않습니다. YAML의 콘텐츠를 조정하여 많은 큐 매니저 설정을 변경할 수 있습니다.

- a) QueueManager YAML 파일을 작성하십시오.

예를 들어, 기본 큐 관리자를 IBM Cloud Pak for Integration에 설치하려면, 다음 콘텐츠로 파일 "mq-quickstart.yaml"을 작성하십시오.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
```

```

metadata:
  name: quickstart-cp4i
spec:
  version: 9.2.5.0-r3
  license:
    accept: false
    license: L-RJON-C7QG3S
    use: NonProduction
  web:
    enabled: true
  queueManager:
    name: "QUICKSTART"
    storage:
      queueManager:
        type: ephemeral
  template:
    pod:
      containers:
        - name: qmgr
          env:
            - name: MQSNOAUT
              value: "yes"

```

중요사항: IBM Cloud Pak for Integration 라이선스 계약에 동의하는 경우 `accept: false`을(를) `accept: true`으로 변경하십시오. 라이선스에 대한 자세한 내용은 [119 페이지의 『mq.ibm.com/v1beta1에 대한 라이선스 부여 참조』](#)의 내용을 참조하십시오.

이 예는 IBM Cloud Pak Identity and Access Manager를 사용한 싱글 사인온이 사용으로 설정된 웹 콘솔을 포함하는, 큐 관리자를 사용하여 배치된 웹 서버를 포함합니다.

IBM Cloud Pak for Integration와 독립적으로 기본 큐 관리자를 설치하려면, 다음 콘텐츠로 파일 "mq-quickstart.yaml"을 작성하십시오.

```

apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart
spec:
  version: 9.2.5.0-r3
  license:
    accept: false
    license: L-APIG-BZDDDY
  web:
    enabled: true
  queueManager:
    name: "QUICKSTART"
    storage:
      queueManager:
        type: ephemeral
  template:
    pod:
      containers:
        - name: qmgr
          env:
            - name: MQSNOAUT
              value: "yes"

```

중요: MQ 라이선스 계약에 동의하는 경우 `accept: false`을(를) `accept: true`으로 변경하십시오. 라이선스에 대한 자세한 내용은 [119 페이지의 『mq.ibm.com/v1beta1에 대한 라이선스 부여 참조』](#)의 내용을 참조하십시오.

b) QueueManager 오브젝트를 작성하십시오.

```
oc apply -f mq-quickstart.yaml
```

2. 큐 관리자가 실행 중인지 확인

실행 중인 배치의 유효성을 검증할 수 있습니다.

```
oc describe queuemanager <QueueManagerResourceName>
```

을 실행하여 배치의 유효성을 검증한 다음 상태를 확인합니다.

예를 들어,

```
oc describe queuemanager quickstart
```

, 그리고 `status.Phase` 필드가 `Running`을(를) 표시하는지 확인하십시오.

관련 태스크

102 페이지의 『[Red Hat OpenShift 클러스터 외부에서 큐 관리자에 연결하도록 Route 구성](#)』

Red Hat OpenShift 클러스터 외부에서 IBM MQ 큐 관리자에 애플리케이션을 연결하려면 Red Hat OpenShift 라우트가 필요합니다. SNI는 TLS 1.2 이상의 프로토콜이 사용될 때 TLS 프로토콜에서만 사용 가능하므로 IBM MQ 큐 관리자 및 클라이언트 애플리케이션에서 TLS를 사용으로 설정해야 합니다. Red Hat OpenShift Container Platform Router에서는 IBM MQ 큐 관리자로 요청을 라우팅하는 데 SNI를 사용합니다.

108 페이지의 『[Red Hat OpenShift 클러스터에 배치된 IBM MQ Console에 연결](#)』

Red Hat OpenShift Container Platform 클러스터에 배치된 큐 관리자의 IBM MQ Console 에 연결하는 방법입니다.

OpenShift CP4I 큐 관리자 구성의 예

큐 관리자는 QueueManager 사용자 정의 자원의 콘텐츠를 조정하여 구성할 수 있습니다.

이 태스크 정보

QueueManager YAML 파일을 사용하여 큐 관리자를 구성하는 데 도움을 받으려면 다음 예를 사용하십시오.

프로시저

- 81 페이지의 『[예: MQSC 및 INI 파일 제공](#)』
- 82 페이지의 『[예: TLS 구성](#)』

OpenShift CP4I 예: MQSC 및 INI 파일 제공

이 예제에서는 두 개의 MQSC 파일과 하나의 INI 파일을 포함하는 Kubernetes ConfigMap을 작성합니다. 그 후에는 이러한 MQSC 및 INI 파일을 처리하는 큐 관리자가 배치됩니다.

이 태스크 정보

큐 관리자가 배치되면 MQSC 및 INI 파일을 제공할 수 있습니다. MQSC 및 INI 데이터는 하나 이상의 Kubernetes ConfigMaps 및 Secret에서 정의되어야 합니다. 이러한 항목은 큐 관리자를 배치할 네임스페이스(프로젝트)에 작성되어야 합니다.

참고: Kubernetes Secret은 MQSC 또는 INI 파일에 민감한 데이터가 포함되는 경우 사용해야 합니다.

이 방법으로 MQSC 및 INI를 제공하려면 IBM MQ Operator 1.1 이상이 필요합니다.

예

이 예제에서는 두 개의 MQSC 파일과 하나의 INI 파일을 포함하는 하나의 Kubernetes ConfigMap을 작성합니다. 그 후에는 이러한 MQSC 및 INI 파일을 처리하는 큐 관리자가 배치됩니다.

ConfigMap 예 - 클러스터에 다음 YAML을 적용하십시오.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: mqsc-ini-example
data:
  example1.mqsc: |
    DEFINE QLOCAL('DEV.QUEUE.1') REPLACE
    DEFINE QLOCAL('DEV.QUEUE.2') REPLACE
  example2.mqsc: |
    DEFINE QLOCAL('DEV.DEAD.LETTER.QUEUE') REPLACE
  example.ini: |
    Channels:
      MQIBindType=FASTPATH
```

QueueManager 예 - 명령행 또는 IBM Cloud Pak for Integration Platform Navigator를 사용하여 다음 구성으로 큐 관리자를 배치하십시오.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: mqsc-ini-cp4i
spec:
  version: 9.2.5.0-r3
  license:
    accept: false
    license: L-RJON-C7QG3S
    use: NonProduction
  web:
    enabled: true
  queueManager:
    name: "MQSCINI"
  mqsc:
    - configMap:
        name: mqsc-ini-example
        items:
          - example1.mqsc
          - example2.mqsc
  ini:
    - configMap:
        name: mqsc-ini-example
        items:
          - example.ini
  storage:
    queueManager:
      type: ephemeral
```

중요사항: IBM Cloud Pak for Integration 라이선스 계약에 동의하는 경우 `accept: false`(를) `accept: true`(으)로 변경하십시오. 라이선스의 세부사항은 [mq.ibm.com/v1beta1에 대한 라이선스 부여 참조](https://www.ibm.com/support/cti/cpe?id=MQSCINI)를 참조하십시오.

추가 정보:

- 큐 관리자는 단일 Kubernetes ConfigMap 또는 Secret(이 예제에 표시된 대로) 또는 다중 Kubernetes ConfigMap 및 Secret을 사용하도록 구성할 수 있습니다.
- Kubernetes ConfigMap 또는 Secret에서 MQSC 및 INI 데이터 모두를 사용하도록(이 예제에 표시된 대로) 선택하거나 사용 가능한 파일의 서브세트만을 사용하도록 각 큐 관리자를 구성하도록 선택할 수 있습니다.
- MQSC 및 INI 파일은 키를 기반으로 알파벳순으로 처리됩니다. 따라서 큐 관리자 구성에 표시되는 순서에 관계 없이 `example1.mqsc`가 항상 `example2.mqsc`보다 먼저 처리됩니다.
- 여러 MQSC 또는 INI 파일에 동일한 키가 있으면 여러 Kubernetes ConfigMap 또는 Secret에서 이 파일 세트는 큐 관리자 구성에 파일이 정의되는 순서를 기반으로 처리됩니다.

OpenShift CP4I Linux 예: TLS 구성

이 예는 IBM MQ Operator를 사용하여 Red Hat OpenShift Container Platform에 큐 관리자를 배치합니다. 샘플 클라이언트와 큐 관리자 사이에는 단방향 TLS 통신이 구성됩니다. 이 예는 메시지를 넣고 가져오으로써 구성이 완료되었음을 보여줍니다.

시작하기 전에

이 예를 완료하려면 먼저 다음 필수조건을 완료해야 합니다.

- IBM MQ client를 설치하고 `samp/bin` 및 `bin`을 `PATH`에 추가하십시오. 사용자에게는 **runmqakm**, **amqsputc** 및 **amqsgetc** 애플리케이션이 필요하며, 이들은 다음과 같이 IBM MQ client의 일부로서 설치할 수 있습니다.
 - **Windows** > **Linux** Windows 및 Linux의 경우: <https://ibm.biz/mq92redistclients> 에서 운영 체제에 대한 IBM MQ 재배포 가능 클라이언트를 설치하십시오.
 - **mac OS** Mac: IBM MQ MacOS Toolkit를 다운로드하여 설정하십시오. <https://developer.ibm.com/tutorials/mq-macos-dev/>
- 자신의 운영 체제에 맞는 OpenSSL 도구를 설치하십시오.

- 이 예를 위한 Red Hat OpenShift Container Platform(OCP) 프로젝트/네임스페이스를 작성하십시오.
- 명령행에서 OCP 클러스터에 로그인한 후 위 네임스페이스로 전환하십시오.
- 위 네임스페이스에 IBM MQ Operator가 설치되어 사용 가능하도록 하십시오.

이 태스크 정보

이 예에서는 Red Hat OpenShift Container Platform에 배치되는 큐 관리자를 정의하는 사용자 정의 자원 YAML 을 제공합니다. TLS를 사용 가능하도록 하여 큐 관리자를 배치하는 데 필요한 추가 단계 또한 자세히 설명되어 있습니다. 완료 후 메시지를 넣고 가져오면 큐 관리자가 TLS를 사용하여 구성되었는지 유효성 검증할 수 있습니다.

IBM MQ 서버를 위한 TLS 개인 키 및 인증서 작성

다음 코드 예는 큐 관리자에 대한 자체 서명 인증서를 작성하고, 클라이언트의 신뢰 저장소 역할을 하는 키 데이터베이스에 해당 인증서를 추가하는 방법을 보여줍니다. 이미 개인 키와 인증서가 있는 경우에는 이를 사용할 수도 있습니다.

자체 서명 인증서는 개발 용도로만 사용해야 한다는 점을 참고하십시오.

현재 디렉토리에서 자체 서명 인증서 및 공용 인증서 작성

다음 명령을 실행하십시오.

```
openssl req -newkey rsa:2048 -nodes -keyout tls.key -subj "/CN=localhost" -x509 -days 3650 -out tls.crt
```

서버 공개 키를 클라이언트 키 데이터베이스에 추가

키 데이터베이스는 클라이언트 애플리케이션의 신뢰 저장소로 사용됩니다.

클라이언트 키 데이터베이스를 작성하십시오.

```
runmqakm -keydb -create -db clientkey.kdb -pw password -type cms -stash
```

이전에 생성된 공개 키를 클라이언트 키 데이터베이스에 추가하십시오.

```
runmqakm -cert -add -db clientkey.kdb -label mqservercert -file tls.crt -format ascii -stashed
```

큐 관리자 배치를 위한 TLS 인증서 구성

큐 관리자가 위에서 작성된 파일을 참조하여 키 및 인증서를 참조 및 적용하고, Kubernetes TLS 시크릿을 작성할 수 있도록 하십시오. 이 태스크를 시작할 때는 위치가 이전에 작성한 네임스페이스인지 확인하십시오.

```
oc create secret tls example-tls-secret --key="tls.key" --cert="tls.crt"
```

MQSC 명령을 포함하는 ConfigMap 작성

새 큐 및 SVRCONN 채널을 작성하고, *nobody*라는 사용자만 차단하여 채널에 대한 액세스를 허용하는 채널 인증 레코드를 추가하기 위해 MQSC 명령을 포함하는 Kubernetes ConfigMap을 작성하십시오.

이 접근법은 개발 용도로만 사용해야 한다는 점을 참고하십시오.

위치 [이전\(시작하기 전에 참조\)](#)에 작성한 네임스페이스인지 확인한 후 OCP UI에서, 또는 명령행을 사용하여 다음 YAML을 입력하십시오.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: example-tls-configmap
data:
  tls.mqsc: |
    DEFINE QLOCAL('EXAMPLE.QUEUE') REPLACE
    DEFINE CHANNEL(SECUREQMCHL) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCAUTH(OPTIONAL)
    SSLCIPH('ANY_TLS12_OR_HIGHER')
    SET CHLAUTH(SECUREQMCHL) TYPE(BLOCKUSER) USERLIST('nobody') ACTION(ADD)
```

필수 OCP 라우트 작성

위치 [이 태스크를 시작하기 전에 작성한 네임스페이스인지 확인한 후 OCP UI에서](#), 또는 명령행을 사용하여 다음 YAML을 입력하십시오.

```

apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: example-tls-route
spec:
  host: securemqchl.chl.mq.ibm.com
  to:
    kind: Service
    name: securemq-ibm-mq
  port:
    targetPort: 1414
  tls:
    termination: passthrough

```

Red Hat OpenShift Container Platform Router에서는 IBM MQ 큐 관리자로 요청을 라우팅하는 데 SNI를 사용한다는 점을 참고하십시오. 이전에 작성된 ConfigMap의 MQSC에 지정된 채널 이름을 변경하는 경우에는 여기에 있는 host 필드, 그리고 나중에 작성되는 CCDT 파일의 host 필드도 변경해야 합니다. 추가 정보는 [102 페이지의 『Red Hat OpenShift 클러스터 외부에서 큐 관리자에 연결하도록 Route 구성』](#)의 내용을 참조하십시오.

큐 관리자 배치

중요: 이 예에서는 `MQSNOAUT` 변수를 사용하여 큐 관리자에 대한 권한 부여를 사용 안함으로 설정하며, 이는 TLS를 사용하여 클라이언트를 연결하는 단계에 집중할 수 있게 해 줍니다. 이는 개별 애플리케이션의 권한을 낮출 수 있는 메커니즘 없이 연결하는 애플리케이션이 전체 관리 권한을 갖도록 하므로 IBM MQ의 프로덕션 배치에서는 권장되지 않습니다.

다음 사용자 정의 자원 YAML을 사용하여 새 큐 관리자를 작성하십시오. 이는 이전에 작성된 ConfigMap 및 시크릿과 `MQSNOAUT` 변수를 참조한다는 점을 참고하십시오.

위치가 이 태스크를 시작하기 전에 작성한 네임스페이스인지 확인한 후 OCP UI에서, 명령행 또는 IBM Cloud Pak for Integration Platform Navigator를 사용하여 다음 YAML을 입력하십시오. 올바른 라이선스가 지정되었는지 확인하고 `false`를(를) `true`(으)로 변경하여 라이선스를 승인하십시오.

```

apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: securemq
spec:
  license:
    accept: false
    license: L-RJON-C7QG3S
    use: Production
  queueManager:
    name: SECUREQM
    mqsc:
      - configMap:
          name: example-tls-configmap
          items:
            - tls.mqsc
      storage:
        queueManager:
          type: ephemeral
  template:
    pod:
      containers:
        - env:
            - name: MQSNOAUT
              value: 'yes'
          name: qmgr
  version: 9.2.5.0-r3
  web:
    enabled: true
  pki:
    keys:
      - name: example
        secret:
          secretName: example-tls-secret
          items:
            - tls.key
            - tls.crt

```

큐 관리자가 실행 중인지 확인

이제 큐 관리자가 배치되었습니다. 계속하기 전에 Running 상태에 있는지 확인하십시오. 예를 들면, 다음과 같습니다.

```
oc get qmgr secureqm
```

큐 관리자에 대한 연결 테스트

큐 관리자가 단방향 TLS 통신을 위해 구성되었는지 확인하려면 **amqsputc** 및 **amqsgetc** 샘플 애플리케이션을 사용하십시오.

큐 관리자 호스트 이름 찾기

다음 명령을 사용하여 라우트 `secureqm-ibm-mq-qm`에 대한 큐 관리자의 완전한 호스트 이름을 찾으십시오.

```
oc get routes secureqm-ibm-mq-qm
```

큐 관리자 세부사항 지정

큐 관리자 세부사항을 지정하는 CCDT.JSON 파일을 작성하십시오. host 값을 이전 단계의 호스트 이름으로 바꾸십시오.

```
{
  "channel":
  [
    {
      "name": "SECUREQMCHL",
      "clientConnection":
      {
        "connection":
        [
          {
            "host": "<hostname from previous step>",
            "port": 443
          }
        ],
        "queueManager": "SECUREQM"
      },
      "transmissionSecurity":
      {
        "cipherSpecification": "ECDHE_RSA_AES_128_CBC_SHA256"
      },
      "type": "clientConnection"
    }
  ]
}
```

환경 변수 내보내기

자신의 운영 체제에 맞는 방식으로 다음 환경 변수를 내보내십시오. 이러한 변수는 **amqsputc** 및 **amqsgetc**가 읽습니다.

시스템에 있는 파일의 경로를 업데이트하십시오.

```
export MQCCDTURL='<full path to file>/CCDT.JSON'
export MQSSLKEYR='<full path to file>/clientkey'
```

큐에 메시지 넣기

다음 명령을 실행하십시오.

```
amqsputc EXAMPLE.QUEUE SECUREQM
```

큐 관리자에 대한 연결이 성공한 경우에는 다음 응답이 출력됩니다.

```
target queue is EXAMPLE.QUEUE
```

임의의 텍스트를 입력하고 **Enter**를 누르기를 몇 번 반복하여 큐에 몇 가지 메시지를 넣으십시오.

완료하려면 **Enter**를 두 번 누르십시오.

큐에서 메시지 검색

다음 명령을 실행하십시오.

```
amqsgetc EXAMPLE.QUEUE SECUREQM
```

이전 단계에서 추가한 메시지가 이용되어 출력됩니다.

몇 초 후 명령이 종료됩니다.

축하합니다, TLS를 사용으로 설정하여 큐 관리자를 배치하고 클라이언트로부터 큐 관리자로 메시지를 안전하게 넣고 가져올 수 있는 것을 확인했습니다.

OpenShift CP4I 예: 라이선스 서비스 어노테이션 사용자 정의

IBM MQ Operator는 배치된 자원을 IBM License Service에 자동으로 추가합니다. 이는 IBM License Service에서 모니터링되며 필수 인타이틀먼트에 해당하는 보고서가 생성됩니다.

이 태스크 정보

IBM MQ Operator가 추가한 어노테이션은 표준 상황에서 예상되는 어노테이션이며 큐 관리자 배치 중에 선택된 라이선스 값을 기반으로 합니다.

예

License이(가) L-RJON-BZFQU2(IBM Cloud Pak for Integration 2021.2.1)로 설정되고 **Use**이(가) NonProduction으로 설정된 경우 다음 어노테이션이 적용됩니다.

- cloudpakId: c8b82d189e7545f0892db9ef2731b90d
- cloudpakName: IBM Cloud Pak for Integration
- productChargedContainers: qmgr
- productCloudpakRatio: '4:1'
- productID: 21dfe9a0f00f444f888756d835334909
- productName: IBM MQ Advanced for Non-Production
- productMetric: VIRTUAL_PROCESSOR_CORE
- productVersion: 9.2.3.0

IBM Cloud Pak for Integration 내에서 IBM App Connect Enterprise 배치에는 IBM MQ에 대한 제한된 인타이틀먼트가 포함됩니다. 이러한 상황에서는 IBM License Service에서 올바른 사용을 캡처하도록 이러한 어노테이션을 대체해야 합니다. 이를 수행하려면 107 페이지의 [『큐 관리자 자원에 사용자 정의 어노테이션 및 레이블 추가』](#)에 설명된 접근 방식을 사용하십시오.

예를 들어, IBM MQ가 IBM App Connect Enterprise 인타이틀먼트로 배치된 경우 다음 코드 단편에 표시된 접근 방식을 사용하십시오.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: mq4ace
  namespace: cp4i
spec:
  annotations:
    productMetric: FREE
```

라이선스 어노테이션을 수정해야 하는 두 가지 다른 일반적인 이유는 다음과 같습니다.

1. IBM MQ Advanced가 다른 IBM 제품의 인타이틀먼트에 포함되어 있습니다.
 - 이 상황에서는 이전에 IBM App Connect Enterprise에 대해 설명된 접근 방식을 사용하십시오.
2. IBM MQ가 IBM Cloud Pak for Integration 라이선스로 배치됩니다.
 - IBM Cloud Pak for Integration 라이선스가 있는 경우 IBM MQ 또는 IBM MQ Advanced 비율에 따라 큐 관리자를 배치하도록 결정할 수 있습니다. IBM MQ 비율에 따라 배치하는 경우 고유 HA 또는 Advanced Message Security와 같은 고급 기능을 사용하지 않도록 해야 합니다.
 - 이 상황에서 프로덕션용에는 다음 어노테이션을 사용하십시오.

```

apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: mq4ace
  namespace: cp4i
spec:
  annotations:
    productID: c661609261d5471fb4ff8970a36bccea
    productCloudpakRatio: '4:1'
    productName: IBM MQ for Production
    productMetric: VIRTUAL_PROCESSOR_CORE

```

- 비프로덕션용에는 다음 어노테이션을 사용하십시오.

```

apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: mq4ace
  namespace: cp4i
spec:
  annotations:
    productID: 151bec68564a4a47a14e6fa99266deff
    productCloudpakRatio: '8:1'
    productName: IBM MQ for Non-Production
    productMetric: VIRTUAL_PROCESSOR_CORE

```

OpenShift > CP4I IBM MQ Operator를 사용하여 큐 관리자에 대한 고가용성 구성

이 태스크 정보

프로시저

- [V 9.2.3](#)
87 페이지의 『[고유 HA](#)』.
- [V 9.2.3](#)
89 페이지의 『[예: 고유 HA 큐 관리자 구성](#)』.
- 97 페이지의 『[예: 다중 인스턴스 큐 관리자 구성](#)』.

CP4I > CD > V 9.2.3 고유 HA

고유 HA는 클라우드 블록 스토리지에서 사용하기에 적합한 IBM MQ의 고유(내장) 고가용성 솔루션입니다.

고유 HA 구성에서는 여러 스토리지 세트 간에 복구 가능 MQ 데이터(예: 메시지)를 복제하여, 스토리지 장애 시 데이터 유실을 방지하는 고가용성 큐 관리자를 제공합니다. 큐 관리자는 실행 중인 여러 개의 인스턴스로 구성됩니다. 하나는 리더이고 나머지는 장애 발생 시 신속하게 이어 받을 준비가 된 인스턴스입니다. 이를 통해 큐 관리자와 메시지에 대한 액세스를 최대화합니다.

고유 HA 구성은 세 개의 Kubernetes 팟(Pod)으로 구성되며, 각각에는 큐 관리자 인스턴스가 있습니다. 하나의 인스턴스는 활성 큐 관리자이며 메시지를 처리하고 복구 로그에 기록합니다. 복구 로그가 기록될 때마다 활성 큐 관리자는 복제본으로 알려진 다른 두 인스턴스에 데이터를 전송합니다. 각 복제본은 자체 복구 로그에 쓰고, 데이터를 확인한 후 복제된 복구 로그에서 자체 큐 데이터를 업데이트합니다. 활성 큐 관리자를 실행하는 팟(Pod)이 실패하면, 큐 관리자의 복제본 인스턴스 중 하나가 활성 역할을 인계받고, 현재 데이터로 작업합니다.

로그 유형은 '복제된 로그' 로 알려져 있습니다. 복제된 로그는 자동 로그 관리 및 자동 매체 이미지가 사용 가능한 선형 로그입니다. [로그 유형](#)을 참조하십시오. 선형 로그를 관리하는 데 사용하는 복제된 로그를 관리하기 위해 동일한 기술을 사용합니다.

Kubernetes Service는 현재 활성 인스턴스로 TCP/IP 클라이언트 연결을 라우팅할 때 사용되며, 이는 네트워크 트래픽 준비가 된 유일한 팟(Pod)으로서 식별됩니다. 이는 클라이언트 애플리케이션이 다른 인스턴스를 인식하지 않아도 발생합니다.

분리 뇌(split-brain) 상황이 발생할 가능성을 현저히 감소시키기 위해 세 개의 팟(Pod)이 사용됩니다. 두 개의 팟(Pod)이 있는 고가용성 시스템에서는 두 팟(Pod) 사이의 연결이 중단될 때 분할 뇌 상황이 발생할 수 있습니다. 연결되지 않은 두 팟(Pod)은 서로 다른 데이터를 누적하면서 동시에 큐 관리자를 실행할 수 있습니다. 연결이 복

원되면 서로 다른 버전의 두 데이터('분할 뇌')가 있게 되고 보존할 데이터 세트 및 제거할 데이터 세트를 판별하려면 수동 개입이 필요합니다.

고유 HA에서는 퀴럼이 있는 세 개의 팟(Pod) 시스템을 사용하여 분할 뇌 상황을 방지합니다. 다른 팟(Pod) 중 하나 이상과 통신할 수 있는 팟(Pod)은 퀴럼을 구성합니다. 퀴럼이 있는 팟(Pod)에서는 큐 관리자가 활성 인스턴스가 될 수 있습니다. 큐 관리자는 하나 이상의 다른 팟(Pod)에 연결되지 않은 팟(Pod)에서 활성화될 수 없으므로 동시에 두 개의 활성 인스턴스가 있을 수 없습니다.

- 단일 팟(Pod)이 실패하면 다른 두 팟(Pod) 중 하나의 큐 관리자가 인계할 수 있습니다. 두 팟(Pod)이 실패하면, 팟(Pod)에 퀴럼이 없으므로 나머지 팟(Pod)에서 큐 관리자가 활성 인스턴스가 될 수 없습니다(나머지 팟(Pod)은 다른 두 팟(Pod)이 실패했는지 또는 계속 실행 중이고 연결이 끊어졌는지를 알릴 수 없습니다).
- 단일 팟(Pod)에서 연결이 끊어지면 팟(Pod)에는 퀴럼이 없으므로 큐 관리자는 이 팟(Pod)에서 활성화될 수 없습니다. 나머지 두 팟(Pod) 중 퀴럼이 있는 하나의 큐 관리자가 인계할 수 있습니다. 모든 팟(Pod)에서 연결이 끊어지면 퀴럼이 있는 팟(Pod)이 없으므로 큐 관리자는 어느 팟(Pod)에서도 활성화될 수 없습니다.

활성 팟(Pod)이 실패한 후 다음에 복구되면 이는 복제본 역할로 그룹에 다시 결합될 수 있습니다.

다음 그림은 한 큐 관리자의 세 인스턴스가 세 컨테이너에 배치되는 일반 배치를 표시합니다.

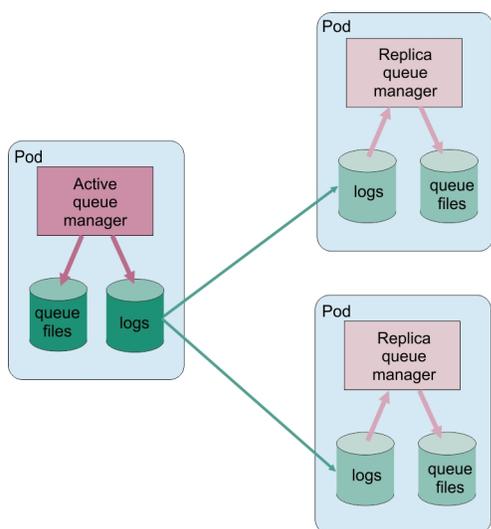


그림 1. 고유 HA 구성의 예제

CP4I CD V9.2.3 IBM MQ Operator를 사용하여 고유 HA 구성

원시 HA는 QueueManager API를 사용하여 구성되며 고급 옵션은 INI 파일을 사용하여 사용할 수 있습니다.

원시 HA는 `.spec.queueManager.availability` API의 QueueManager을(를) 사용하여 구성됩니다. 예를 들면 다음과 같습니다.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: nativeha-example
spec:
  license:
    accept: false
    license: L-RJON-C7QG3S
    use: Production
  queueManager:
    availability:
      type: NativeHA
    version: 9.2.5.0-r3
```

`.spec.queueManager.availability.type` 필드는 NativeHA(으)로 설정되어야 합니다.

고유 HA는 IBM MQ 9.2.3 이상에서 사용할 수 있습니다.

.spec.queueManager.availability에서, 복제할 때 큐 관리자 인스턴스 간에 사용할 TLS 시크릿 및 암호를 구성할 수도 있습니다. 이는 강력하게 추천하며 89 페이지의 『예: 고유 HA 큐 관리자 구성』에서는 단계별 안내서를 사용 가능합니다.

관련 참조

89 페이지의 『예: 고유 HA 큐 관리자 구성』

이 예제는 IBM MQ Operator를 사용하여 원시 고가용성 기능을 사용하여 Red Hat OpenShift Container Platform (OCP) 로 큐 관리자를 배치하는 방법을 보여줍니다.

 예: 고유 HA 큐 관리자 구성

이 예제는 IBM MQ Operator를 사용하여 원시 고가용성 기능을 사용하여 Red Hat OpenShift Container Platform (OCP) 로 큐 관리자를 배치하는 방법을 보여줍니다.

시작하기 전 주의사항

이 예를 완료하려면 먼저 다음 필수조건을 완료해야 합니다.

- IBM MQ client을(를) 설치하고 설치된 samp/bin 및 bin 디렉토리를 사용자의 PATH에 추가하십시오. 클라이언트는 이 예제에 필요한 **runmqakm**, **amqspuic** 및 **amqsgetc** 애플리케이션을 제공합니다. 다음과 같이 IBM MQ client를 설치하십시오.
 -  Windows 및 Linux의 경우: <https://ibm.biz/mq92redistclients> 에서 운영 체제에 대한 IBM MQ 재배포 가능 클라이언트를 설치하십시오.
 -  Mac: IBM MQ MacOS Toolkit를 다운로드하여 설정하십시오. <https://ibm.biz/mqdevmacclient>의 내용을 참조하십시오.
- 자신의 운영 체제에 맞는 OpenSSL 도구를 설치하십시오. 아직 개인 키 및 인증서가 없는 경우 큐 관리자의 자체 서명 인증서를 생성하기 위해 필요합니다.
- 이 예제를 위한 Red Hat OpenShift Container Platform(OCP) 프로젝트/네임스페이스를 작성하고 75 페이지의 『IBM MQ를 위한 Red Hat OpenShift 프로젝트 준비』 태스크의 단계를 따르십시오.
- 명령행에서는 OCP 클러스터에 로그인하고 방금 작성한 네임스페이스로 전환하십시오.
- IBM MQ Operator가 설치되고 네임스페이스에서 사용 가능한지 확인하십시오.
- 사용자의 큐 관리자에서 사용할 기본 스토리지 클래스를 OCP에서 구성하십시오. 기본 스토리지 클래스를 설정하지 않고 이 학습을 완료하려는 경우 [참고 2: 기본이 아닌 스토리지 클래스 사용을 참조하십시오](#).

이 태스크에 대한 정보

고유 HA 큐 관리자에는 하나의 활성 및 두 개의 복제본 Kubernetes 팟(Pod)이 있습니다. 이들은 정확히 세 개의 복제본과 한 세트의 Kubernetes 지속 볼륨이 있는 Kubernetes Stateful Set의 일부로서 실행됩니다. 고유 HA 큐 관리자에 대한 자세한 정보는 15 페이지의 『컨테이너에서 IBM MQ에 대한 고가용성』의 내용을 참조하십시오.

이 예제에서는 지속적 스토리지를 사용하고 TLS로 구성된 고유 HA 큐 관리자를 정의하는 사용자 정의 자원 YAML을 제공합니다. OCP에 큐 관리자를 배치한 후 활성 큐 관리자 팟(Pod)의 실패를 시뮬레이션합니다. 자동 복구 발생을 보고 실패 후 메시지 넣기 및 가져오기를 통해 성공했음을 입증합니다.

예

MQ 서버의 TLS 개인 키 및 인증서 작성

큐 관리자의 자체 서명 인증서를 작성하고, 클라이언트의 신뢰 저장소 역할을 하는 키 데이터베이스로 인증서를 추가할 수 있습니다. 이미 개인 키와 인증서가 있는 경우에는 이를 사용할 수도 있습니다. 개발을 위해서는 자체 서명 인증서만 사용해야 함을 참고하십시오.

현재 디렉토리에서 자체 서명 개인 키 및 공용 인증서를 작성하려면 다음 명령을 실행하십시오.

```
openssl req -newkey rsa:2048 -nodes -keyout tls.key -subj "/CN=localhost" -x509 -days 3650 -out tls.crt
```

고유 HA에서 내부적으로 사용할 TLS 개인 키 및 인증서 작성

고유 HA 큐 관리자에 있는 세 개의 팟(Pod)은 네트워크를 통해 데이터를 복제합니다. 내부적으로 복제할 때 사용할 자체 서명 인증서를 작성할 수 있습니다. 개발을 위해서는 자체 서명 인증서만 사용해야 함을 참고하십시오.

현재 디렉토리에서 자체 서명 개인 키 및 공용 인증서를 작성하려면 다음 명령을 실행하십시오.

```
openssl req -newkey rsa:2048 -nodes -keyout nativeha.key -subj "/CN=localhost" -x509 -days 3650 -out nativeha.crt
```

클라이언트 키 데이터베이스에 큐 관리자 공개 키 추가

클라이언트 키 데이터베이스는 클라이언트 애플리케이션에 대해 신뢰 저장소로서 사용됩니다.

클라이언트 키 데이터베이스를 작성하십시오.

```
runmqakm -keydb -create -db clientkey.kdb -pw password -type cms -stash
```

이전에 생성된 공개 키를 클라이언트 키 데이터베이스에 추가하십시오.

```
runmqakm -cert -add -db clientkey.kdb -label mqservercert -file tls.crt -format ascii -stashed
```

큐 관리자 배치를 위한 TLS 인증서를 포함하는 시크릿 작성

큐 관리자가 키 및 인증서를 참조하고 적용할 수 있도록 위에서 작성된 파일을 참조하는 Kubernetes TLS 시크릿을 작성하십시오. 이 태스크를 시작할 때는 위치가 이전에 작성한 네임스페이스인지 확인하십시오.

```
oc create secret tls example-ha-secret --key="tls.key" --cert="tls.crt"
```

내부 고유 HA TLS 인증서 및 키를 포함하는 시크릿 작성

큐 관리자가 키 및 인증서를 참조하고 적용할 수 있도록 위에서 작성된 파일을 참조하는 Kubernetes TLS 시크릿을 작성하십시오. 이 태스크를 시작할 때는 위치가 이전에 작성한 네임스페이스인지 확인하십시오.

```
oc create secret tls example-ha-secret-internal --key="nativeha.key" --cert="nativeha.crt"
```

MQSC 명령을 포함하는 ConfigMap 작성

MQSC 명령을 포함하는 Kubernetes 구성 맵을 작성하여 새 큐 및 SVRCONN 채널을 작성하고, *nobody* 사용자만을 차단하여 채널에 대한 액세스를 허용하는 채널 인증 레코드를 추가하십시오.

이 접근법은 개발 용도로만 사용해야 한다는 점을 참고하십시오.

이전에 작성한 네임스페이스에 있는지 확인하고(89 페이지의 『시작하기 전 주의사항』 참조), OCP UI에서 또는 명령행을 사용하여 다음 YAML을 입력하십시오.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: example-mi-configmap
data:
  tls.mqsc: |
    DEFINE QLOCAL('EXAMPLE.QUEUE') DEFPSIST(YES) REPLACE
    DEFINE CHANNEL(HAQMCHL) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCAUTH(OPTIONAL)
    SSLCIPH('ANY_TLS12_OR_HIGHER')
    SET CHLAUTH(HAQMCHL) TYPE(BLOCKUSER) USERLIST('nobody') ACTION(ADD)
```

라우팅 구성

IBM MQ 9.2.1 또는 나중에 IBM MQ client 또는 툴킷을 사용하는 경우, 큐 관리자 구성 파일 (INI 파일) 을 사용하여 큐 관리자에 대한 라우팅을 구성할 수 있습니다. 파일 내에서는 채널 이름이 아닌 호스트 이름을 기반으로 라우트할 *OutboundSNI* 변수를 설정하십시오.

명령을 실행 중인 디렉토리에서 정확히 다음 텍스트를 포함하는 *mqclient.ini* 파일을 작성하십시오.

```
SSL:
  OutboundSNI=HOSTNAME
```

이 INI 파일에 있는 값은 변경하지 마십시오. 예를 들어, 문자열 *HOSTNAME*을(를) 변경할 수 없습니다.

자세한 내용은 클라이언트 구성 파일의 SSL 스탠자를 참조하십시오.

IBM MQ 9.2.1이전에 IBM MQ client 또는 툴킷을 사용하는 경우 이전 구성 파일 대신 OCP 라우트를 작성해야 합니다. [참고 1: 라우트 작성의 단계를 수행하십시오.](#)

큐 관리자 배치

중요: 이 예에서는 `MQSNOAUT` 변수를 사용하여 큐 관리자에 대한 권한 부여를 사용 안함으로 설정하며, 이는 TLS를 사용하여 클라이언트를 연결하는 단계에 집중할 수 있게 해 줍니다. 이는 개별 애플리케이션의 권한을 낮출 수 있는 메커니즘 없이 연결하는 애플리케이션이 전체 관리 권한을 갖도록 하므로 IBM MQ의 프로덕션 배치에서는 권장되지 않습니다.

다음 YAML을 복사한 후 업데이트합니다.

- 올바른 라이선스가 지정되었는지 확인하십시오. mq.ibm.com/v1beta1의 라이선싱 참조를 참조하십시오. IBM Cloud Pak for Integration 2021.1.1에서 라이선스는 평가 라이선스 L-RJON-BYRMW(이)어야 합니다.
- `false`을(를) `true`(으)로 변경하여 라이선스를 승인하십시오.

큐 관리자 사용자 정의 자원 YAML:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: nativeha-example
spec:
  license:
    accept: false
    license: L-RJON-C7QG3S
    use: Production
  queueManager:
    name: HAEXAMPLE
    availability:
      type: NativeHA
    tls:
      secretName: example-ha-secret-internal
  mqsc:
    - configMap:
        name: example-mi-configmap
        items:
          - tls.mqsc
  template:
    pod:
      containers:
        - env:
            - name: MQSNOAUT
              value: 'yes'
          name: qmgr
  version: 9.2.5.0-r3
  pki:
    keys:
      - name: example
        secret:
          secretName: example-ha-secret
          items:
            - tls.key
            - tls.crt
```

사용자가 이전에 작성한 네임스페이스에 있는지 확인하며 Red Hat OpenShift Container Platform 웹 콘솔, 명령행 또는 IBM Cloud Pak for Integration Platform Navigator을 사용하여 업데이트된 YAML을 배치하십시오.

시스템에서 고유 HA 큐 관리자를 구성하는 동안 잠시 작업이 지연되며, 그 후에 큐 관리자를 사용할 수 있어야 합니다.

유효성 검증

이 절에서는 큐 관리자가 예상 대로 작동하는지 유효성 검증합니다.

큐 관리자가 실행 중인지 확인

이제 큐 관리자가 배치되었습니다. 계속하기 전에 Running 상태에 있는지 확인하십시오. 예를 들면, 다음과 같습니다.

```
oc get qmgr nativeha-example
```

큐 관리자에 대한 연결 테스트

큐 관리자가 단방향 TLS 통신을 위해 구성되었는지 확인하려면 **amqsputc** 및 **amqsgetc** 샘플 애플리케이션을 사용하십시오.

큐 관리자 호스트 이름 찾기

nativeha-example-ibm-mq-qm 라우트에 대한 큐 관리자 호스트 이름을 찾으려면 다음 명령을 실행하십시오. 호스트 이름은 HOST 필드에 리턴됩니다.

```
oc get routes nativeha-example-ibm-mq-qm
```

큐 관리자 세부사항 지정

큐 관리자 세부사항을 지정하는 CCDT.JSON 파일을 작성하십시오. 이전 단계에서 리턴되는 호스트 이름으로 호스트 값을 바꾸십시오.

```
{
  "channel":
  [
    {
      "name": "HAQMCHL",
      "clientConnection":
      {
        "connection":
        [
          {
            "host": "<host from previous step>",
            "port": 443
          }
        ],
        "queueManager": "HAEXAMPLE"
      },
      "transmissionSecurity":
      {
        "cipherSpecification": "ECDHE_RSA_AES_128_CBC_SHA256"
      },
      "type": "clientConnection"
    }
  ]
}
```

환경 변수 내보내기

자신의 운영 체제에 맞는 방식으로 다음 환경 변수를 내보내십시오. 이러한 변수는 **amqsputc** 및 **amqsgetc**가 읽습니다.

시스템에 있는 파일의 경로를 업데이트하십시오.

```
export MQCCDTURL='<full_path_to_file>/CCDT.JSON'
export MQSSLKEYR='<full_path_to_file>/clientkey'
```

큐에 메시지 넣기

다음 명령을 실행하십시오.

```
amqsputc EXAMPLE.QUEUE HAEXAMPLE
```

큐 관리자에 대한 연결이 성공한 경우에는 다음 응답이 출력됩니다.

```
target queue is EXAMPLE.QUEUE
```

임의의 텍스트를 입력하고 **Enter**를 누르기를 몇 번 반복하여 큐에 몇 가지 메시지를 넣으십시오.

완료하려면 **Enter**를 두 번 누르십시오.

큐에서 메시지 검색

다음 명령을 실행하십시오.

```
amqsgetc EXAMPLE.QUEUE HAEXAMPLE
```

이전 단계에서 추가한 메시지가 이용되어 출력됩니다.

몇 초 후 명령이 종료됩니다.

활성 팟(Pod)이 실패하도록 강제 실행

큐 관리자의 자동 복구를 유효성 검증하기 위해 팟(Pod) 실패를 시뮬레이션합니다.

활성 및 대기 팟(Pod) 보기

다음 명령을 실행하십시오.

```
oc get pods --selector app.kubernetes.io/instance=nativeha-example
```

READY 필드에서 활성 팟(Pod)은 1/1 값을 리턴하지만 복제본 팟(Pod)은 0/1 값을 리턴합니다.

활성 팟(Pod) 삭제

활성 팟(Pod)의 전체 이름을 지정하며 다음 명령을 실행하십시오.

```
oc delete pod nativeha-example-ibm-mq-<value>
```

팟(Pod) 상태 다시 보기

다음 명령을 실행하십시오.

```
oc get pods --selector app.kubernetes.io/instance=nativeha-example
```

큐 관리자 상태 보기

다른 팟(Pod) 중 하나의 전체 이름을 지정하며 다음 명령을 실행하십시오.

```
oc exec -t Pod -- dspmq -o nativeha -x -m HAEXAMPLE
```

예를 들어 활성 인스턴스가 변경되었음을 상태에서 표시해야 합니다.

```
QMNAME(HAEXAMPLE) ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDAT(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDAT(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDAT(2022-01-12) ALTTIME(12.03.44)
```

메시지 넣기 및 다시 가져오기

대기 팟(Pod)이 활성 팟(Pod)이 된 후(즉, **READY** 필드 값이 1/1이(가) 된 후), 이전에 설명한 대로 다음 명령을 다시 사용하여 큐 관리자에 메시지를 넣은 후 큐 관리자에서 메시지를 검색하십시오.

```
amqsputc EXAMPLE.QUEUE HAEXAMPLE
```

```
amqsgetc EXAMPLE.QUEUE HAEXAMPLE
```

축하합니다. 고유 HA 큐 관리자를 성공적으로 배치하고 팟(Pod) 실패에서 자동으로 복구될 수 있음을 보여 주었습니다.

추가 정보

참고 1: 라우트 작성

IBM MQ 9.2.1 이전에 IBM MQ client 또는 툴킷을 사용하는 경우 경로를 작성해야 합니다.

라우트를 작성하려면 사용자가 이전에 작성한 네임스페이스에 있는지 확인하고(89 페이지의 『시작하기 전 주의사항』 참조), Red Hat OpenShift Container Platform 웹 콘솔에서 또는 명령행을 사용하여 다음 YAML 을 입력하십시오.

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: example-mi-route
spec:
```

```
host: hamqchl.ch1.mq.ibm.com
to:
  kind: Service
  name: nativeha-example-ibm-mq
port:
  targetPort: 1414
tls:
  termination: passthrough
```

Red Hat OpenShift Container Platform Router에서는 IBM MQ 큐 관리자로 요청을 라우팅하는 데 SNI를 사용한다는 점을 참고하십시오. MQSC 명령을 포함하는 구성 맵에 지정된 채널 이름을 변경하는 경우, 여기서 그리고 큐 관리자 세부사항을 지정하는 CCDT. JSON 파일에서도 호스트 필드를 변경해야 합니다. 자세한 정보는 102 페이지의 『Red Hat OpenShift 클러스터 외부에서 큐 관리자에 연결하도록 Route 구성』의 내용을 참조하십시오.

참고 2: 기본이 아닌 스토리지 클래스 사용

이 예제는 기본 스토리지 클래스가 Red Hat OpenShift Container Platform에서 구성되었을 것으로 예상하므로, 큐 관리자 사용자 정의 자원 YAML에서는 스토리지 정보가 필요하지 않습니다. 기본값으로 구성된 스토리지 클래스가 없거나 다른 스토리지 클래스를 사용하려는 경우 `spec.queueManager.storage` 아래에 `defaultClass: <storage_class_name>`을(를) 추가하십시오.

스토리지 클래스 이름은 이미 존재하는 스토리지 클래스의 이름과 정확히 일치해야 합니다. 즉, `oc get storageclass` 명령이 리턴한 이름과 일치해야 합니다. 또한 `ReadWriteMany`도 지원해야 합니다. 추가 정보는 10 페이지의 『IBM MQ Operator의 스토리지 고려사항』의 내용을 참조하십시오.

관련 태스크

94 페이지의 『IBM MQ 인증 컨테이너에 대한 원시 HA 큐 관리자의 상태 보기』

IBM MQ 인증 컨테이너의 경우 실행 중인 Pods 중 하나에서 `dspmqr` 명령을 실행하여 원시 HA 인스턴스의 상태를 볼 수 있습니다.

CP4I **CD** **V9.2.2** IBM MQ 인증 컨테이너에 대한 원시 HA 큐 관리자의 상태 보기
IBM MQ 인증 컨테이너의 경우 실행 중인 Pods 중 하나에서 `dspmqr` 명령을 실행하여 원시 HA 인스턴스의 상태를 볼 수 있습니다.

이 태스크 정보

중요사항:

실행 중인 팟(Pod) 중 하나에서 `dspmqr` 명령을 사용하여 큐 관리자 인스턴스의 운영 상태를 볼 수 있습니다. 리턴되는 정보는 인스턴스가 활성화인지 또는 복제본인지에 따라 달라집니다. 활성화 인스턴스에서 제공하는 정보는 결정적이며 복제본 노드의 정보는 이전 정보일 수 있습니다.

다음 조치를 수행할 수 있습니다.

- 현재 노드의 큐 관리자 인스턴스가 활성화인지 또는 복제본인지 여부를 확인합니다.
- 현재 노드에서 인스턴스의 고유 HA 운영 상태를 확인합니다.
- 고유 HA 구성에서 세 개 인스턴스 모두의 운영 상태를 확인합니다.

다음 상태 필드는 고유 HA 구성 상태를 보고하는 데 사용됩니다.

ROLE

인스턴스의 현재 역할을 지정하며 Active, Replica 또는 Unknown 중 하나입니다.

INSTANCE

`crtmqm` 명령의 `-lr` 옵션을 사용하여 작성된 경우 큐 관리자의 이 인스턴스에 대해 제공된 이름입니다.

INSYNC

인스턴스가 필요한 경우 활성화 인스턴스로서 인계받을 수 있는지 여부를 표시합니다.

QUORUM

`number_of_instances_in-sync/number_of_instances_configured` 양식으로 쿼럼 상태를 보고합니다.

REPLADDR

큐 관리자 인스턴스의 복제 이름.

CONNECTV

노드가 활성 인스턴스에 연결되어 있는지 여부를 표시합니다.

BACKLOG

인스턴스가 뒤에 있는 KB 수를 표시합니다.

CONNINST

이름 지정된 인스턴스가 이 인스턴스에 연결되는지 여부를 표시합니다.

ALTDATE

이 정보가 마지막으로 업데이트된 날짜를 표시합니다(업데이트된 적이 없으면 공백).

ALTTIME

이 정보가 마지막으로 업데이트된 시간을 표시합니다(업데이트된 적이 없으면 공백).

프로시저

- 큐 관리자의 일부인 포트를 찾으십시오.

```
oc get pod --selector app.kubernetes.io/instance=nativeha-qm
```

- 포크 중 하나에서 dspmq을(를) 실행하십시오.

```
oc exec -t Pod dspmq
```

```
oc rsh Pod
```

대화식 셸의 경우 dspmq을(를) 직접 실행할 수 있습니다.

- 큐 관리자 인스턴스가 활성 인스턴스로 실행 중인지 또는 복제본으로 실행 중인지를 판별합니다.

```
oc exec -t Pod dspmq -o status -m QMgrName
```

큐 관리자 BOB의 활성 인스턴스는 다음 상태를 보고합니다.

```
QMNAME(BOB)          STATUS(Running)
```

큐 관리자 BOB의 복제본 인스턴스는 다음 상태를 보고합니다.

```
QMNAME(BOB)          STATUS(Replica)
```

비활성 인스턴스는 다음 상태를 보고합니다.

```
QMNAME(BOB)          STATUS(Ended Immediately)
```

- 지정된 팟(Pod)에서 인스턴스의 고유 HA 운영 상태를 판별합니다.

```
oc exec -t Pod dspmq -o nativeha -m QMgrName
```

큐 관리자 BOB의 활성 인스턴스는 다음 상태를 보고합니다.

```
QMNAME(BOB)          ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
```

큐 관리자 BOB의 복제본 인스턴스는 다음 상태를 보고합니다.

```
QMNAME(BOB)          ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
```

큐 관리자 BOB의 비활성 인스턴스는 다음 상태를 보고합니다.

```
QMNAME(BOB)          ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
```

- 고유 HA 구성에서 모든 인스턴스의 고유 HA 운영 상태를 판별합니다.

```
oc exec -t Pod dspmq -o nativeha -x -m QMgrName
```

큐 관리자 BOB의 활성 인스턴스를 실행 중인 노드에서 이 명령을 실행하는 경우 다음 상태를 수신합니다.

```
QMNAME(BOB)          ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

큐 관리자 BOB의 복제본 인스턴스를 실행 중인 노드에서 이 명령을 실행하면, 다음 상태를 수신할 수 있습니다. 이는 복제본 중 하나가 뒤쳐지고 있음을 표시합니다.

```
QMNAME(BOB)          ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(No) BACKLOG(435)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

큐 관리자 BOB의 비활성 인스턴스를 실행 중인 노드에서 이 명령을 실행하면 다음 상태를 수신할 수 있습니다.

```
QMNAME(BOB)          ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
INSTANCE(inst1) ROLE(Unknown) REPLADDR(9.20.123.45) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
INSTANCE(inst2) ROLE(Unknown) REPLADDR(9.20.123.46) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
INSTANCE(inst3) ROLE(Unknown) REPLADDR(9.20.123.47) CONNACTV(No) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
```

인스턴스가 계속 무엇이 활성인지 그리고 무엇이 복제본인지 조정하는 경우 명령을 실행하면, 다음 상태를 수신할 수 있습니다.

```
QMNAME(BOB)          STATUS(Negotiating)
```

관련 참조

[dspmq\(큐 관리자 표시\) 명령](#)

89 페이지의 『예: 고유 HA 큐 관리자 구성』

이 예제는 IBM MQ Operator를 사용하여 원시 고가용성 기능을 사용하여 Red Hat OpenShift Container Platform (OCP) 로 큐 관리자를 배치하는 방법을 보여줍니다.

CP4I **CD** **V9.2.3** **고유 HA를 위한 고급 성능 조정**

타이밍과 간격을 조정하기 위한 고급 설정입니다. 기본값이 시스템의 요구사항과 일치하는 경우 이러한 설정을 사용할 필요가 없습니다.

기본 HA 구성을 위한 기본 옵션은 기본 큐 관리자 INI 파일을 구성하는 데 IBM MQ Operator이(가) 사용하는 QueueManager API를 사용하여 처리됩니다. NativeHALocal인스턴스 스탠자아래에 INI 파일을 사용하여 구성할 수 있는 몇 가지 고급 옵션이 있습니다. INI 파일을 구성하는 방법에 대한 자세한 정보는 [81 페이지의 『예: MQSC 및 INI 파일 제공』](#) 도 참조하십시오.

HeartbeatInterval

하트비트 간격은 고유 HA 큐 관리자의 활성 인스턴스가 네트워크 하트비트를 전송하는 빈도(밀리세컨드)를 정의합니다. 유효한 하트비트 간격 값의 범위는 500(0.5초)에서 60000(1분)까지이며, 값이 이 범위를 벗어나면 큐 관리자를 시작하는 데 실패합니다. 이 속성이 생략되는 경우 기본값 5000(5초)가 사용됩니다. 각 인스턴스는 동일한 하트비트 간격을 사용해야 합니다.

HeartbeatTimeout

하트비트 제한시간은 활성 인스턴스가 반응하지 않음을 결정하기 전에 고유 HA 큐 관리자의 복제본 인스턴스가 대기하는 기간을 정의합니다. 유효한 하트비트 간격 제한시간 값의 범위는 500(0.5초)에서 120000(2분)까지입니다. 하트비트 제한시간 값은 하트비트 간격 이상이어야 합니다.

값이 올바르지 않으면 큐 관리자를 시작하는 데 실패합니다. 이 속성을 생략하면 새 활성 인스턴스를 선택하기 위한 프로세스를 시작하기 전에 2 x HeartbeatInterval의 복제본 대기가 대기합니다. 각 인스턴스는 동일한 하트비트 제한시간을 사용해야 합니다.

RetryInterval

재시도 간격은 고유 HA 큐 관리자가 실패한 복제 링크를 재시도해야 하는 빈도(밀리초)를 정의합니다. 올바른 재시도 간격 범위는 500(0.5초)에서 120000(2분)까지입니다. 이 속성을 생략하면 실패한 복제 링크를 재시도하기 전에 2 x HeartbeatInterval의 복제본 대기가 대기합니다.

CP4I 원시 HA 큐 관리자 종료

endmqm 명령을 사용하여 원시 HA 그룹의 일부인 활성 또는 복제본 큐 관리자를 종료할 수 있습니다.

프로시저

- 큐 관리자의 활성 인스턴스를 종료하려면 이 문서의 구성 섹션에서 [원시 HA 큐 관리자 종료](#) 를 참조하십시오.

CP4I **CD** **V9.2.2**

IBM Cloud Pak for Integration 2021.1.1에서 고유 HA 기능 평가 **IBM Cloud Pak for Integration 2021.1.1** 고유 HA 평가 기간이 종료되었습니다. Please use the updated Native HA feature that is available from **IBM Cloud Pak for Integration 2021.2.1**, using **IBM MQ Operator 1.6** or higher with **IBM MQ 9.2.3** or higher.

관련 태스크

94 페이지의 『[IBM MQ 인증 컨테이너에 대한 원시 HA 큐 관리자의 상태 보기](#)』

IBM MQ 인증 컨테이너의 경우 실행 중인 Pods 중 하나에서 **dspm** 명령을 실행하여 원시 HA 인스턴스의 상태를 볼 수 있습니다.

관련 참조

89 페이지의 『[예: 고유 HA 큐 관리자 구성](#)』

이 예제는 IBM MQ Operator를 사용하여 원시 고가용성 기능을 사용하여 Red Hat OpenShift Container Platform (OCP) 로 큐 관리자를 배치하는 방법을 보여줍니다.

OpenShift

CP4I

예: 다중 인스턴스 큐 관리자 구성

이 예제는 IBM MQ Operator를 사용하여 멀티 인스턴스 큐 관리자를 Red Hat OpenShift Container Platform (OCP) 에 배치하는 방법을 보여줍니다. 이 예제에서는 샘플 클라이언트와 큐 관리자 사이에서 단방향 TLS 통신도 구성합니다. 예제는 시뮬레이션된 팟(Pod) 실패 전후에 메시지를 넣고 가져오므로써 성공적인 구성을 보여줍니다.

시작하기 전 주의사항

이 예를 완료하려면 먼저 다음 필수조건을 완료해야 합니다.

- IBM MQ client(를) 설치하고 설치된 samp/bin 및 bin 디렉토리를 사용자의 *PATH*에 추가하십시오. 클라이언트는 이 예제에 필요한 **runmqakm**, **amqsputc** 및 **amqsgetc** 애플리케이션을 제공합니다. 다음과 같이 IBM MQ client를 설치하십시오.
 - **Windows** **Linux** Windows 및 Linux의 경우: <https://ibm.biz/mq92redistclients> 에서 운영 체제에 대한 IBM MQ 재배포 가능 클라이언트를 설치하십시오.
 - **macOS** Mac: IBM MQ MacOS Toolkit를 다운로드하여 설정하십시오. <https://developer.ibm.com/tutorials/mq-macos-dev/>을 참조하십시오.
- 자신의 운영 체제에 맞는 OpenSSL 도구를 설치하십시오. 아직 개인 키 및 인증서가 없는 경우 큐 관리자의 자체 서명 인증서를 생성하기 위해 필요합니다.
- 이 예를 위한 Red Hat OpenShift Container Platform(OCP) 프로젝트/네임스페이스를 작성하십시오.
- 명령행에서 OCP 클러스터에 로그인한 후 위 네임스페이스로 전환하십시오.
- 위 네임스페이스에 IBM MQ Operator가 설치되어 사용 가능하도록 하십시오.
- 사용자의 큐 관리자에서 사용할 기본 스토리지 클래스를 OCP에서 구성하십시오. 기본 스토리지 클래스를 설정하지 않고 이 학습을 완료하려는 경우 [참고 2: 기본이 아닌 스토리지 클래스 사용](#)을 참조하십시오.

이 태스크에 대한 정보

다중 인스턴스 큐 관리자에는 하나의 활성 팟(Pod)과 하나의 대기 Kubernetes 팟(Pod)이 있습니다. 이는 정확히 두 개의 복제본과 한 세트의 Kubernetes 지속 볼륨이 있는 Kubernetes Stateful Set의 일부로서 실행됩니다. 다중 인스턴스 큐 관리자에 대한 자세한 정보는 15 페이지의 『컨테이너에서 IBM MQ에 대한 고가용성』의 내용을 참조하십시오.

예제에서는 지속 스토리지가 있고 TLS로 구성되는 다중 인스턴스 큐 관리자를 정의하는 사용자 정의 YAML을 제공합니다. OCP에 큐 관리자를 배치한 후 활성 큐 관리자 팟(Pod)의 실패를 시뮬레이션합니다. 자동 복구가 발생하는 것을 보고 실패 후 메시지 넣기 및 가져오기를 통해 성공했음을 입증합니다.

예

MQ 서버의 TLS 개인 키 및 인증서 작성

이 절에서는 큐 관리자의 자체 서명 인증서를 작성하는 방법 및 클라이언트에 대한 신뢰 저장소 역할을 하는 키 데이터베이스에 인증서를 추가하는 방법에 대해 설명합니다. 이미 개인 키와 인증서가 있는 경우에는 이를 사용할 수도 있습니다. 개발을 위해서는 자체 서명 인증서만 사용해야 함을 참고하십시오.

현재 디렉토리에서 자체 서명 개인 키 및 공용 인증서를 작성하려면 다음 명령을 실행하십시오.

```
openssl req -newkey rsa:2048 -nodes -keyout tls.key -subj "/CN=localhost" -x509 -days 3650 -out tls.crt
```

클라이언트 키 데이터베이스에 큐 관리자 공개 키 추가

클라이언트 키 데이터베이스는 클라이언트 애플리케이션에 대해 신뢰 저장소로서 사용됩니다.

클라이언트 키 데이터베이스를 작성하십시오.

```
runmqakm -keydb -create -db clientkey.kdb -pw password -type cms -stash
```

이전에 생성된 공개 키를 클라이언트 키 데이터베이스에 추가하십시오.

```
runmqakm -cert -add -db clientkey.kdb -label mqservercert -file tls.crt -format ascii -stashed
```

큐 관리자 배치를 위한 TLS 인증서를 포함하는 시크릿 작성

큐 관리자가 키 및 인증서를 참조하고 적용할 수 있도록 위에서 작성된 파일을 참조하는 Kubernetes TLS 시크릿을 작성하십시오. 이 태스크를 시작할 때는 위치가 이전에 작성한 네임스페이스인지 확인하십시오.

```
oc create secret tls example-mi-secret --key="tls.key" --cert="tls.crt"
```

MQSC 명령을 포함하는 ConfigMap 작성

MQSC 명령을 포함하는 Kubernetes 구성 맵을 작성하여 새 큐 및 SVRCONN 채널을 작성하고, *nobody* 사용자만을 차단하여 채널에 대한 액세스를 허용하는 채널 인증 레코드를 추가하십시오.

이 접근법은 개발 용도로만 사용해야 한다는 점을 참고하십시오.

이전에 작성한 네임스페이스에 있는지 확인하고(97 페이지의 『시작하기 전 주의사항』 참조), OCP UI에서 또는 명령행을 사용하여 다음 YAML을 입력하십시오.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: example-mi-configmap
data:
  tls.mqsc: |
    DEFINE QLOCAL('EXAMPLE.QUEUE') DEFPSIST(YES) REPLACE
    DEFINE CHANNEL(MIQMCHL) CHLTYPE(SVRCONN) TRPTYPE(TCP) SSLCAUTH(OPTIONAL)
    SSLCIPH('ANY_TLS12_OR_HIGHER')
    SET CHLAUTH(MIQMCHL) TYPE(BLOCKUSER) USERLIST('nobody') ACTION(ADD)
```

라우팅 구성

IBM MQ 9.2.1 또는 나중에 IBM MQ client 또는 툴킷을 사용하는 경우, 큐 관리자 구성 파일 (INI 파일) 을 사용하여 큐 관리자에 대한 라우팅을 구성할 수 있습니다. 파일 내에서는 채널 이름이 아닌 호스트 이름을 기반으로 라우트할 *OutboundSNI* 변수를 설정하십시오.

명령을 실행 중인 디렉토리에서 다음 텍스트를 포함하는 `mqclient.ini` 파일을 작성하십시오.

```

## Module Name: mqclient.ini                                ##
## Type       : IBM MQ MQI client configuration file      ##
## Function   : Define the configuration of a client      ##
##           ##
## Notes      :                                           ##
## 1) This file defines the configuration of a client     ##
##           ##
##           ##
SSL:
  OutboundSNI=HOSTNAME

```

참고: 이 페이지에 있는 값은 변경하지 마십시오. 예를 들어, 문자열 HOSTNAME은(는) 그대로 두어야 합니다.

자세한 내용은 클라이언트 구성 파일의 SSL 스탠자를 참조하십시오.

IBM MQ 9.2.1이전에 IBM MQ client 또는 툴킷을 사용하는 경우 이전 구성 파일 대신 OCP 라우트를 작성해야 합니다. [참고 1: 라우트 작성](#)의 단계를 수행하십시오.

큐 관리자 배치

중요: 이 예에서는 *MQSNOAUT* 변수를 사용하여 큐 관리자에 대한 권한 부여를 사용 안함으로 설정하며, 이는 TLS를 사용하여 클라이언트를 연결하는 단계에 집중할 수 있게 해 줍니다. 이는 개별 애플리케이션의 권한을 낮출 수 있는 메커니즘 없이 연결하는 애플리케이션이 전체 관리 권한을 갖도록 하므로 IBM MQ의 프로덕션 배치에서는 권장되지 않습니다.

다음 YAML을 복사한 후 업데이트합니다.

- 올바른 라이선스가 지정되었는지 확인하십시오. mq.ibm.com/v1beta1의 라이선싱 참조를 참조하십시오.
- `false`을(를) `true`으로 변경하여 라이선스를 승인하십시오.
- IBM Cloud File Storage를 사용 중인 경우 [참고 3: IBM Cloud File Storage 사용](#)을 참조하십시오.

큐 관리자 사용자 정의 자원 YAML:

```

apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: miexample
spec:
  license:
    accept: false
    license: L-RJON-C7QG3S
    use: NonProduction
  queueManager:
    name: MIEXAMPLE
    availability:
      type: MultiInstance
  mqsc:
    - configMap:
        name: example-mi-configmap
        items:
          - tls.mqsc
  template:
    pod:
      containers:
        - env:
            - name: MQSNOAUT
              value: 'yes'
          name: qmgr
  version: 9.2.5.0-r3
  web:
    enabled: true
  pki:
    keys:
      - name: example
        secret:
          secretName: example-mi-secret
          items:
            - tls.key
            - tls.crt

```

사용자가 이전에 작성한 네임스페이스에 있는지 확인한 후 OCP UI에서, 명령행을 사용하여 또는 IBM Cloud Pak for Integration Platform Navigator를 사용하여 업데이트된 YAML을 배치하십시오.

유효성 검증

잠깐 지연된 후에는 다중 인스턴스 큐 관리자를 구성 및 사용 가능해야 합니다. 이 절에서는 큐 관리자가 예상대로 작동하는지 유효성 검증합니다.

큐 관리자가 실행 중인지 확인

이제 큐 관리자가 배치되었습니다. 계속하기 전에 Running 상태에 있는지 확인하십시오. 예를 들면, 다음과 같습니다.

```
oc get qmgr miexample
```

큐 관리자에 대한 연결 테스트

큐 관리자가 단방향 TLS 통신을 위해 구성되었는지 확인하려면 **amqsputc** 및 **amqsgetc** 샘플 애플리케이션을 사용하십시오.

큐 관리자 호스트 이름 찾기

miexample-ibm-mq-qm 라우트에 대한 큐 관리자 호스트 이름을 찾으려면 다음 명령을 실행하십시오. 호스트 이름은 HOST 필드에 리턴됩니다.

```
oc get routes miexample-ibm-mq-qm
```

큐 관리자 세부사항 지정

큐 관리자 세부사항을 지정하는 CCDT.JSON 파일을 작성하십시오. 이전 단계에서 리턴되는 호스트 이름으로 호스트 값을 바꾸십시오.

```
{
  "channel":
  [
    {
      "name": "MIQMCHL",
      "clientConnection":
      {
        "connection":
        [
          {
            "host": "<host from previous step>",
            "port": 443
          }
        ],
        "queueManager": "MIEXAMPLE"
      },
      "transmissionSecurity":
      {
        "cipherSpecification": "ECDHE_RSA_AES_128_CBC_SHA256"
      },
      "type": "clientConnection"
    }
  ]
}
```

환경 변수 내보내기

자신의 운영 체제에 맞는 방식으로 다음 환경 변수를 내보내십시오. 이러한 변수는 **amqsputc** 및 **amqsgetc**가 읽습니다.

시스템에 있는 파일의 경로를 업데이트하십시오.

```
export MQCCDTURL='<full_path_to_file>/CCDT.JSON'
export MQSSLKEYR='<full_path_to_file>/clientkey'
```

큐에 메시지 넣기

다음 명령을 실행하십시오.

```
amqsputc EXAMPLE.QUEUE MIEXAMPLE
```

큐 관리자에 대한 연결이 성공한 경우에는 다음 응답이 출력됩니다.

```
target queue is EXAMPLE.QUEUE
```

임의의 텍스트를 입력하고 **Enter**를 누르기를 몇 번 반복하여 큐에 몇 가지 메시지를 넣으십시오.
완료하려면 **Enter**를 두 번 누르십시오.

큐에서 메시지 검색

다음 명령을 실행하십시오.

```
amqsgetc EXAMPLE.QUEUE MIEXAMPLE
```

이전 단계에서 추가한 메시지가 이용하여 출력됩니다.

몇 초 후 명령이 종료됩니다.

활성 팟(Pod)이 실패하도록 강제 실행

다중 인스턴스 큐 관리자의 자동 복구를 유효성 검증하기 위해 팟(Pod) 실패를 시뮬레이션합니다.

활성 및 대기 팟(Pod) 보기

다음 명령을 실행하십시오.

```
oc get pods
```

READY 필드에서 활성 팟(Pod)은 1/1 값을 리턴하는 반면, 대기 팟(Pod)은 0/1 값을 리턴합니다.

활성 팟(Pod) 삭제

활성 팟(Pod)의 전체 이름을 지정하며 다음 명령을 실행하십시오.

```
oc delete pod miexample-ibm-mq-<value>
```

팟(Pod) 상태 다시 보기

다음 명령을 실행하십시오.

```
oc get pods
```

대기 팟(Pod) 로그 보기

대기 팟(Pod)의 전체 이름을 지정하면서 다음 명령을 실행하십시오.

```
oc logs miexample-ibm-mq-<value>
```

다음 메시지가 표시되어야 합니다.

```
IBM MQ queue manager 'MIEXAMPLE' becoming the active instance.
```

메시지 넣기 및 다시 가져오기

대기 팟(Pod)이 활성 팟(Pod)이 된 후(즉, **READY** 필드 값이 1/1이(가) 된 후), 이전에 설명한 대로 다음 명령을 다시 사용하여 큐 관리자에 메시지를 넣은 후 큐 관리자에서 메시지를 검색하십시오.

```
amqsputc EXAMPLE.QUEUE MIEXAMPLE
```

```
amqsgetc EXAMPLE.QUEUE MIEXAMPLE
```

축하합니다. 다중 인스턴스 큐 관리자를 성공적으로 배치하고 팟(Pod) 실패에서 자동으로 복구될 수 있음을 보여 주었습니다.

추가 정보

참고 1: 라우트 작성

IBM MQ 9.2.1이전에 IBM MQ client 또는 툴킷을 사용하는 경우 OCP 라우트를 작성해야 합니다.

라우트를 작성하려면 사용자가 이전에 작성한 네임스페이스에 있는지 확인하고(97 페이지의 『[시작하기 전 주의사항](#)』 참조), OCP UI에서 또는 명령행을 사용하여 다음 YAML을 입력하십시오.

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
```

```

name: example-mi-route
spec:
  host: miqmchl.ch1.mq.ibm.com
  to:
    kind: Service
    name: miexample-ibm-mq
  port:
    targetPort: 1414
  tls:
    termination: passthrough

```

Red Hat OpenShift Container Platform Router에서는 IBM MQ 큐 관리자로 요청을 라우팅하는 데 SNI를 사용한다는 점을 참고하십시오. MQSC 명령을 포함하는 구성 맵에 지정된 채널 이름을 변경하는 경우, 여기서 그리고 큐 관리자 세부사항을 지정하는 CCDT JSON 파일에서도 호스트 필드를 변경해야 합니다. 자세한 정보는 102 페이지의 『Red Hat OpenShift 클러스터 외부에서 큐 관리자에 연결하도록 Route 구성』의 내용을 참조하십시오.

참고 2: 기본이 아닌 스토리지 클래스 사용

이 예제는 기본 스토리지 클래스가 OCP에서 구성된 것으로 예상하므로, 큐 관리자 사용자 정의 자원 YAML에서는 스토리지 정보가 필요하지 않습니다. 기본값으로 구성된 스토리지 클래스가 없거나 다른 스토리지 클래스를 사용하려는 경우 `spec.queueManager.storage` 아래에 `defaultClass`:

`<storage_class_name>`을(를) 추가하십시오.

스토리지 클래스 이름은 사용자의 OCP 시스템에 존재하는 스토리지 클래스의 이름과 정확하게 일치해야 합니다. 즉, `oc get storageclass` 명령이 리턴한 이름과 일치해야 합니다. 또한 `ReadWriteMany`도 지원해야 합니다. 추가 정보는 10 페이지의 『IBM MQ Operator의 스토리지 고려사항』의 내용을 참조하십시오.

참고 3: IBM Cloud File Storage 사용

예를 들어 IBM Cloud File Storage을(를) 사용할 때 일부 상황에서는 큐 관리자 사용자 정의 자원 YAML의 **securityGroups** 필드도 지정해야 합니다. 예를 들어, 다음 하위 필드를 `spec`: 바로 아래에 추가하여

```

securityContext:
  supplementalGroups: [99]

```

자세한 정보는 10 페이지의 『IBM MQ Operator의 스토리지 고려사항』의 내용을 참조하십시오.

OpenShift CP4I CD Red Hat OpenShift 클러스터 외부에서 큐 관리자에 연결하도록 Route 구성

Red Hat OpenShift 클러스터 외부에서 IBM MQ 큐 관리자에 애플리케이션을 연결하려면 Red Hat OpenShift 라우트가 필요합니다. SNI는 TLS 1.2 이상의 프로토콜이 사용될 때 TLS 프로토콜에서만 사용 가능하므로 IBM MQ 큐 관리자 및 클라이언트 애플리케이션에서 TLS를 사용으로 설정해야 합니다. Red Hat OpenShift Container Platform Router에서는 IBM MQ 큐 관리자로 요청을 라우팅하는 데 SNI를 사용합니다.

이 태스크 정보



주의: 이 문서는 IBM MQ 클라이언트의 버전 9.2.1 Continuous Delivery 이상에 적용됩니다. 클라이언트가 버전 9.2.0 Long Term Support 이하를 사용 중인 경우에는 IBM MQ 9.1 문서 페이지 [Red Hat OpenShift 클러스터에 배치된 큐 관리자에 연결](#)을 참조하십시오.

V 9.2.1 Red Hat OpenShift 라우트 의 필수 구성은 클라이언트 애플리케이션의 SNI (Server Name Indication) 동작에 따라 다릅니다. IBM MQ에서는 구성 및 클라이언트 유형에 따라 서로 다른 두 SNI 헤더 설정을 지원합니다. SNI 헤더는 클라이언트 대상의 호스트 이름으로 설정되거나 IBM MQ 채널 이름으로 설정됩니다. IBM MQ에서 채널 이름을 호스트 이름에 매핑하는 방법에 대한 정보는 IBM MQ에서 다중 인증서 기능을 제공하는 방법을 참조하십시오.

V 9.2.1 SNI 헤더가 IBM MQ 채널 이름 또는 호스트 이름으로 설정되는지 여부는 **OutboundSNI** 속성을 사용하여 제어됩니다. 가능한 값은 `OutboundSNI=CHANNEL` (기본값) 또는 `OutboundSNI=HOSTNAME`입니다. 자세한 정보는 클라이언트 구성 파일의 SSL 스탠자를 참조하십시오. CHANNEL 및 HOSTNAME 는 사용자가 사용하는 정확한 값입니다. 이 값은 실제 채널 이름 또는 호스트 이름으로 대체하는 변수 이름이 아닙니다.

OutboundSNI 설정이 다른 클라이언트 동작

OutboundSNI가 HOSTNAME으로 설정된 경우 다음 클라이언트는 연결 이름에 호스트 이름이 제공되는 한 호스트 이름 SNI를 설정합니다.

- C 클라이언트
- 비관리 모드의 .NET 클라이언트
- Java/JMS 클라이언트

OutboundSNI가 HOSTNAME으로 설정되었으며 연결 이름에 IP 주소가 사용된 경우 다음 클라이언트는 공백 SNI 헤더를 설정합니다.

- C 클라이언트
- 비관리 모드의 .NET 클라이언트
- Java/JMS 클라이언트(호스트 이름에 대한 역방향 DNS 검색을 수행할 수 없는 클라이언트)

OutboundSNI가 CHANNEL로 설정되거나 아예 설정되지 않은 경우에는 IBM MQ 채널 이름이 대신 사용되며, 호스트 이름 또는 IP 주소 연결 이름의 사용 여부에 상관없이 항상 전송됩니다.

다음 클라이언트 유형은 SNI 헤더를 IBM MQ 채널 이름으로 설정하는 것을 지원하지 않으며, 따라서 **OutboundSNI** 설정에 상관없이 항상 SNI 헤더를 호스트 이름으로 설정하려 시도합니다.

- AMQP 클라이언트
- XR 클라이언트
- 관리 모드의 .NET 클라이언트(Long Term Support용 IBM MQ 9.2.0 Fix Pack 4 이전 및 Continuous Delivery용 IBM MQ 9.2.3 이전)

Long Term Support 용 IBM MQ 9.2.0 Fix Pack 4 및 Continuous Delivery 용 IBM MQ 9.2.3부터 **OutboundSNI** 특성이 IBM MQ 관리 .NET 클라이언트가 Red Hat OpenShift 라우트를 사용하여 큐 관리자에 연결할 수 있도록 허용하는 호스트 이름으로 설정된 경우 IBM MQ 관리 .NET 클라이언트가 서버 이름을 각 호스트 이름으로 설정하도록 업데이트되었습니다. IBM MQ 9.2.0 Fix Pack 4에서 **OutboundSNI** 특성은 mqclient.ini 파일에서만 추가되고 지원됩니다. .NET 애플리케이션에서는 특성을 설정할 수 없습니다.

클라이언트 애플리케이션이 IBM MQ Internet Pass-Thru(MQIPT)을(를) 통해 Red Hat OpenShift 클러스터에 배치된 큐 관리자에 연결하는 경우 라우트 정의에서 **SSLClientOutboundSNI** 특성을 사용하여 SNI를 호스트 이름으로 설정하도록 MQIPT을(를) 구성할 수 있습니다.

OutboundSNI, 다중 인증서 및 Red Hat OpenShift 라우트

IBM MQ 는 SNI 헤더를 사용하여 여러 인증서 기능을 제공합니다. 애플리케이션이 CERTLABL 필드를 통해 다른 인증서를 사용하도록 구성된 IBM MQ 채널에 연결 중인 경우 애플리케이션은 CHANNEL의 **OutboundSNI** 설정을 사용하여 연결해야 합니다.

Red Hat OpenShift 라우트 구성에 HOSTNAME SNI가 필요한 경우 IBM MQ 의 다중 인증서 기능을 사용할 수 없으며 IBM MQ 채널 오브젝트에서 CERTLABL 설정을 설정할 수 없습니다.

CHANNEL 이외의 **OutboundSNI** 설정을 갖는 애플리케이션이 인증서 레이블이 구성된 채널에 연결되는 경우, 애플리케이션은 MQRC_SSL_INITIALIZATION_ERROR와 함께 거부되고 AMQ9673 메시지가 큐 관리자 오류 로그에 인쇄됩니다.

IBM MQ 가 다중 인증서 기능을 제공하는 방법에 대한 자세한 정보는 [IBM MQ 가 다중 인증서 기능을 제공하는 방법](#) 을 참조하십시오.

예

SNI를 MQ 채널로 설정하는 클라이언트 애플리케이션의 경우 연결할 각각의 채널에 대해 새 Red Hat OpenShift Route를 작성해야 합니다. 또한 올바른 큐 관리자로 라우팅하기 위해 Red Hat OpenShift Container Platform 클러스터 전체에서 고유한 채널 이름을 사용해야 합니다.

IBM MQ에서 채널 이름을 SNI 헤더에 매핑하는 방식으로 인해 MQ 채널 이름이 소문자로 끝나지 않는 것이 중요합니다.

각각의 새 Red Hat OpenShift Route에 필요한 호스트 이름을 판별하려면 각 채널 이름을 SNI 주소에 매핑해야 합니다. 자세한 정보는 [IBM MQ에서 다중 인증서 기능을 제공하는 방법을 참조하십시오](#).

그런 다음 클러스터에서 다음 yaml 을 적용하여 각 채널에 대해 새 Red Hat OpenShift 라우트를 작성해야 합니다.

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: <provide a unique name for the Route>
  namespace: <the namespace of your MQ deployment>
spec:
  host: <SNI address mapping for the channel>
  to:
    kind: Service
    name: <the name of the Kubernetes Service for your MQ deployment (for example "<Queue Manager Name>-ibm-mq")>
  port:
    targetPort: 1414
  tls:
    termination: passthrough
```

클라이언트 애플리케이션 연결 세부사항 구성

다음 명령을 실행하여 클라이언트 연결에 사용할 호스트 이름을 판별할 수 있습니다.

```
oc get route <Name of hostname based Route (for example "<Queue Manager Name>-ibm-mq-qn")>
-n <namespace of your MQ deployment> -o jsonpath="{.spec.host}"
```

클라이언트 연결을 위한 포트는 Red Hat OpenShift Container Platform 라우터 - 일반적으로 443에서 사용되는 포트에 설정해야 합니다.

관련 태스크

[108 페이지의 『Red Hat OpenShift 클러스터에 배치된 IBM MQ Console에 연결』](#)

Red Hat OpenShift Container Platform 클러스터에 배치된 큐 관리자의 IBM MQ Console 에 연결하는 방법입니다.

CP4I IBM Cloud Pak for Integration 조작 대시보드와 통합

IBM Cloud Pak for Integration을 통해 트랜잭션을 추적하는 기능이 조작 대시보드에서 제공됩니다.

이 태스크 정보

조작 대시보드로 통합하면 큐 관리자에 MQ API 엑시트를 설치합니다. API 엑시트는 큐 관리자를 통해 수행 중인 메시지에 대한 조작 대시보드 데이터 저장소로 추적 데이터를 전송합니다.

MQ 클라이언트 바인딩을 사용하여 송신되는 메시지만 추적됩니다.

1.5 이전 IBM MQ Operator 버전의 경우, 추적이 사용으로 설정되면 큐 관리자와 함께 배치되는 추적 에이전트 및 콜렉터 이미지는 항상 최신 버전을 사용할 수 있고, 최신 버전의 IBM Cloud Pak for Integration를 사용하지 않는 경우 비호환성이 소개될 수 있음을 주의하십시오.

프로시저

1. 추적이 사용 가능한 큐 관리자 배치

기본적으로 추적 기능은 사용 불가능합니다.

IBM Cloud Pak for Integration Platform Navigator를 사용하여 배치하는 경우, 배치하는 동안 추적을 사용 가능하게 하고, **추적 사용**을 **켄**으로 설정하고 조작 대시보드가 설치된 네임스페이스로 **네임스페이스 추적**을 설정하는 동안 추적을 사용할 수 있습니다. 큐 관리자 배치에 대한 자세한 정보는 77 페이지의 『IBM Cloud Pak for Integration Platform Navigator를 사용하여 큐 관리자 배치』의 내용을 참조하십시오.

Red Hat OpenShift CLI 또는 Red Hat OpenShift 웹 콘솔을 사용하여 배치하는 경우 다음 YAML 스니펫을 통해 추적을 사용으로 설정할 수 있습니다.

```
spec:
  tracing:
    enabled: true
    namespace: <Operations_Dashboard_Namespace
```

중요: MQ가 조작 대시보드로 조작되어야 큐 관리자가 시작됩니다(다음 단계 참조).

이 기능이 사용되면 큐 관리자 컨테이너 뿐만 아니라 두 개의 사이드카 컨테이너("Agent" 및 "Collector")가 실행됩니다. 이러한 사이드카 컨테이너에 대한 이미지를 기본 MQ 이미지와 동일한 레지스트리에서 사용 가능하고 동일한 PULL 정책 및 PULL 시크릿을 사용합니다. CPU 및 메모리 제한사항을 구성할 수 있는 추가 설정이 있습니다.

2. 조작 대시보드 통합이 있는 큐 관리자가 이 네임스페이스에 처음으로 배치된 경우 조작 대시보드에 등록해야 합니다.

등록하면 큐 관리자 Pod가 성공적으로 시작해야 하는 시크릿 오브젝트를 작성합니다.

CP4I CD IBM Cloud Pak for Integration 2021.4에서 조작 대시보드 통합으로 IBM MQ 9.2.2 또는 9.2.3 배치 또는 업그레이드

각 IBM MQ 버전은 큐 관리자와 함께 배치되는 운영 대시보드 에이전트 및 콜렉터 구성요소의 특정 버전과 연관됩니다. IBM Cloud Pak for Integration 2021.4.1은 이전 에이전트 및 콜렉터 구성요소가 운영 대시보드에서 작동하지 않도록 하는 변경을 도입합니다. 이를 수정하려면 IBM MQ 9.2.2 또는 9.2.3을 사용할 때 사용하는 조작 대시보드 에이전트 및 콜렉터 이미지의 버전을 대체해야 합니다.

새 IBM MQ 9.2.2 또는 9.2.3 큐 관리자 배치

IBM Cloud Pak for Integration 또는 9.2.3에서 IBM MQ 9.2.2 2021.4.1을 사용하는 경우 운영 대시보드 에이전트 및 콜렉터 이미지를 QueueManager YAML의 2.4버전으로 대체해야 합니다. 예를 들면, 다음과 같습니다.

```
spec:
  tracing:
    agent:
      image: cp.icr.io/cp/icp4i/od/icp4i-od-
      agent@sha256:27a211f0f78eff765d1f9520e0f9841f902600bb556827477b206e209cb44d20
    collector:
      image: cp.icr.io/cp/icp4i/od/icp4i-od-
      collector@sha256:dc70b1341b23dc72642ce68809811f9db0e8a0c46bda2508e8eb3d4035e04f4b
```

이를 수행하지 않으면 QueueManager 팟(Pod)이 Pending 상태로 유지됩니다. IBM MQ 9.2.4로 업그레이드하면 이러한 대체를 제거할 수 있습니다.

IBM Cloud Pak for Integration 2021.4.1로 업그레이드

참고: IBM MQ 9.2.2 또는 9.2.3 큐 관리자를 보존하는 경우에는 3단계를 완료하지 마십시오.

1. 이전에 설명한 대로 에이전트 및 콜렉터 이미지를 대체하도록 QueueManager을(를) 업데이트하십시오.
2. 68 페이지의 『IBM MQ Operator 및 큐 관리자 업그레이드』에 설명된 대로 조작 대시보드 및 IBM MQ Operator를 포함하여 IBM Cloud Pak for Integration Operator를 업그레이드하십시오.
3. (선택사항) IBM MQ 9.2.4 이상으로 업그레이드하려면 QueueManager을(를) 업데이트하여 .spec.version 버전에 대해 IBM MQ을(를) 사용한 후 에이전트 및 콜렉터 이미지의 대체를 제거하십시오.

Red Hat OpenShift CLI를 사용하여 사용자 정의 MQSC 및 INI 파일이 포함된 이미지 빌드

Red Hat OpenShift Container Platform 파이프라인을 사용하여 이 이미지를 사용한 큐 관리자에 적용할 MQSC 와 INI 파일을 포함한 새 IBM MQ 컨테이너 이미지를 작성하십시오. 프로젝트 관리자가 이 작업을 완료해야 함

시작하기 전에

Red Hat OpenShift Container Platform 명령행 인터페이스를 설치해야 합니다.

cloudctl login(IBM Cloud Pak for Integration의 경우) 또는 **oc login**을 사용하여 클러스터에 로그인하십시오.

Red Hat OpenShift 프로젝트에 IBM Entitled Registry용 Red Hat OpenShift 시크릿이 없는 경우 [75 페이지의 『IBM MQ를 위한 Red Hat OpenShift 프로젝트 준비』](#) 단계를 따르십시오.

프로시저

1. 작성ImageStream

이미지 스트림 및 연관된 태그는 Red Hat OpenShift Container Platform 내 컨테이너 이미지를 참조하기 위한 추상화를 제공합니다. 이미지 스트림 및 해당 태그를 사용하면 사용 가능한 이미지를 확인하고 저장소의 이미지가 변경되는 경우에도 필요한 특정 이미지를 사용하고 있는지 확인할 수 있습니다.

```
oc create imagestream mymq
```

2. 새 이미지에 대한 BuildConfig 작성

BuildConfig에서는 IBM 공식 이미지를 기반으로 하는 새 이미지에 대한 빌드를 허용하지만 컨테이너 시작 시 실행할 MQSC 또는 INI 파일을 추가합니다.

a) BuildConfig 자원을 정의하는 YAML 파일 작성

예를 들어, 다음 콘텐츠를 사용하여 "mq-build-config.yaml"이라는 파일을 작성합니다.

```
apiVersion: build.openshift.io/v1
kind: BuildConfig
metadata:
  name: mymq
spec:
  source:
    dockerfile: |-
      FROM cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.5.0-r3
      RUN printf "DEFINE QLOCAL(foo) REPLACE\n" > /etc/mqm/my.mqsc \
        && printf "Channels:\n\tMQIBindType=FASTPATH\n" > /etc/mqm/my.ini
      LABEL summary "My custom MQ image"
  strategy:
    type: Docker
    dockerStrategy:
      from:
        kind: "DockerImage"
        name: "cp.icr.io/cp/ibm-mqadvanced-server-integration:9.2.5.0-r3"
      pullSecret:
        name: ibm-entitlement-key
  output:
    to:
      kind: ImageStreamTag
      name: 'mymq:latest-amd64'
```

버전에 대해 올바른 기본 이미지와 사용할 수정사항을 가리키도록 기본 IBM MQ가 언급된 두 지점을 바꿔야 합니다(세부사항은 [19 페이지의 『IBM MQ Operator의 릴리스 히스토리』](#) 참조). 수정사항이 적용되면 이미지를 다시 빌드하려면 이 단계를 반복해야 합니다.

이 예제는 IBM 공식 이미지를 기반으로 하는 새 이미지를 작성하고 "my.mqsc" 및 "my.ini" 파일을 /etc/mqm 디렉토리에 추가합니다. 이 디렉토리에 있는 MQSC나 INI 파일들은 시작 시 컨테이너에 의해 적용됩니다. INI 파일은 **crtmqm -ii** 옵션을 사용하여 적용되며 기존 INI 파일과 병합됩니다. MQSC 파일은 알파벳 순서로 적용됩니다.

큐 관리자가 시작할 때마다 실행되기 때문에 MQSC 명령이 반복 가능하다는 것이 중요합니다. 이는 일반적으로 DEFINE 명령에 REPLACE 매개변수를 추가하고 START 또는 STOP 명령에 IGNSTATE(YES) 매개변수를 추가하는 것을 의미합니다.

- b) 서버에 BuildConfig을(를) 적용하십시오.

```
oc apply -f mq-build-config.yaml
```

3. 빌드를 실행하여 이미지 작성

- a) 빌드 시작

```
oc start-build mymq
```

다음과 유사한 출력이 표시되어야 합니다.

```
build.build.openshift.io/mymq-1 started
```

- b) 빌드 상태 확인

예를 들어, 이전 단계에서 리턴된 빌드 식별자를 사용하여 다음 명령을 실행할 수 있습니다.

```
oc describe build mymq-1
```

4. 새 이미지를 사용하여 큐 관리자 배치

77 페이지의 『Red Hat OpenShift Container Platform 클러스터에 큐 관리자 배치』에서 설명된 단계에 따라 새 사용자 정의 이미지를 YAML에 추가.

YAML의 다음 스니펫을 일반 QueueManager YAML에 추가할 수 있습니다. 여기서 내 이름 공간은 사용 중인 Red Hat OpenShift project/namespace이고 이미지는 이전에 작성한 이미지의 이름입니다(예: "mymq:latest-amd64").

```
spec:
  queueManager:
    image: image-registry.openshift-image-registry.svc:5000/my-namespace/my-image
```

관련 태스크

77 페이지의 『Red Hat OpenShift Container Platform 클러스터에 큐 관리자 배치』

QueueManager 사용자 정의 자원을 사용하여 Red Hat OpenShift Container Platform 클러스터에 큐 관리자를 배치하십시오.

OpenShift CP4I 큐 관리자 자원에 사용자 정의 어노테이션 및 레이블 추가

QueueManager 메타데이터에 사용자 정의 어노테이션 및 레이블을 추가할 수 있습니다.

이 태스크 정보

사용자 정의 어노테이션 및 레이블은 PVC를 제외한 모든 자원에 추가할 수 있습니다. 사용자 정의 어노테이션 또는 레이블이 기존 키와 일치하는 경우에는 IBM MQ Operator에서 설정한 값이 사용됩니다.

프로시저

- 사용자 정의 어노테이션을 추가하십시오.

큐 관리자 자원에 사용자 정의 어노테이션을 추가하려면 metadata에 어노테이션을 추가하십시오. 예를 들면, 다음과 같습니다.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
  annotations:
    annotationKey: "value"
```

- 사용자 정의 레이블을 추가하십시오.

큐 관리자 자원에 사용자 정의 레이블을 추가하려면 metadata에 레이블을 추가하십시오. 예를 들면, 다음과 같습니다.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
  labels:
    labelKey: "value"
```

OpenShift CP4I 런타임 웹훅 검사 사용 안함

런타임 웹훅 검사는 스토리지 클래스가 큐 관리자에 대해 적절한지 확인합니다. 성능을 개선하려 하거나 자신의 환경에 적절하지 않은 경우에는 이 검사를 사용 안함으로 설정할 수 있습니다.

이 태스크 정보

런타임 웹훅 검사는 큐 관리자 구성에 대해 수행됩니다. 여기서는 선택된 큐 관리자 유형에 대해 스토리지 클래스가 적절한지 확인합니다.

큐 관리자 작성에 소요되는 시간을 줄이려 하거나, 이러한 검사가 자신의 환경에 적절하지 않은 경우에는 이를 사용 안함으로 설정하도록 선택할 수 있습니다.

참고: 런타임 웹훅 검사를 사용 안함으로 설정하고 나면 모든 스토리지 클래스 값이 허용됩니다. 이는 불량 큐 관리자를 발생시킬 수 있습니다.

런타임 검사에 대한 지원은 IBM MQ Operator 1.2에서 도입되었습니다.

프로시저

- 런타임 웹훅 검사를 사용 안함으로 설정하십시오.

metadata에 다음 어노테이션을 추가하십시오. 예를 들면, 다음과 같습니다.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
  annotations:
    "com.ibm.cp4i/disable-webhook-runtime-checks" : "true"
```

OpenShift CP4I IBM MQ Operator 를 사용하여 IBM MQ 운영

이 태스크 정보

프로시저

- [75 페이지의 『IBM MQ를 위한 Red Hat OpenShift 프로젝트 준비』](#).
- [77 페이지의 『Red Hat OpenShift Container Platform 클러스터에 큐 관리자 배치』](#).

OpenShift CP4I Red Hat OpenShift 클러스터에 배치된 IBM MQ Console에 연결

Red Hat OpenShift Container Platform 클러스터에 배치된 큐 관리자의 IBM MQ Console 에 연결하는 방법입니다.

이 태스크 정보

IBM MQ Console URL은 Red Hat OpenShift 웹 콘솔 또는 IBM Cloud Pak for Integration Platform Navigator의 QueueManager 세부사항 페이지에서 찾을 수 있습니다. 또는 Red Hat OpenShift CLI에서 다음 명령을 실행하여 찾을 수 있습니다.

```
oc get queuemanager <QueueManager Name> -n <namespace of your MQ deployment> --output jsonpath='{.status.adminUiUrl}'
```

IBM Cloud Pak for Integration 라이선스를 사용 중인 경우 IBM MQ 웹 콘솔은 IBM Cloud Pak IAM(Identity and Access Manager)을 사용하도록 구성됩니다. IAM 컴포넌트는 이미 클러스터 관리자에 의해 설정되어 있을 수 있습니다. 그러나 IAM이 Red Hat OpenShift 클러스터에서 처음 사용된 경우 초기 관리자 비밀번호를 검색해야 합니다. 자세한 정보는 [초기 관리자 비밀번호 가져오기](#)를 참조하십시오.

IBM MQ 라이선스를 사용 중인 경우, MQ 웹 콘솔이 미리 구성되지 않으므로 자체에서 이를 구성해야 합니다. 자세한 정보는 [사용자 및 역할 구성](#)을 참조하십시오.

관련 태스크

102 페이지의 『Red Hat OpenShift 클러스터 외부에서 큐 관리자에 연결하도록 Route 구성』

Red Hat OpenShift 클러스터 외부에서 IBM MQ 큐 관리자에 애플리케이션을 연결하려면 Red Hat OpenShift 라우트가 필요합니다. SNI는 TLS 1.2 이상의 프로토콜이 사용될 때 TLS 프로토콜에서만 사용 가능하므로 IBM MQ 큐 관리자 및 클라이언트 애플리케이션에서 TLS를 사용으로 설정해야 합니다. Red Hat OpenShift Container Platform Router에서는 IBM MQ 큐 관리자로 요청을 라우팅하는 데 SNI를 사용합니다.

 **IBM Cloud Pak IAM을 사용하여 IBM MQ Console에 대한 권한 부여**
IBM MQ Console에 대한 권한은 IBM Cloud Pak for Integration Platform Navigator가 아니라 IBM Cloud Pak 관리 허브를 통해 관리됩니다. IBM MQ는 IBM Cloud Pak for Integration에서 제공하는 "Automation" 권한을 사용하지 않지만 대신 IAM (IBM Cloud Pak Identity and Access Manager)에서 사용 가능한 기본 권한을 사용합니다.

프로시저

1. IBM Cloud Pak 관리 콘솔을 여십시오.

IBM Cloud Pak for Integration Platform UI에서 도구 모음의 오른쪽 상단에 있는 Cloud Pak 스위치 (9-dot 아이콘)를 클릭한 후 **IBM Cloud Pak Administration** 패널을 클릭하십시오.

2. 왼쪽 상단의 탐색 메뉴에서 **ID 및 액세스**를 선택한 후 **팀 및 서비스 ID**를 선택하십시오.

3. 팀을 작성한 후 사용자를 추가하십시오.

- a) **팀 작성**을 선택하십시오.

- b) 팀 이름을 입력한 후 관리할 사용자의 보안 도메인을 선택하십시오.

- c) 사용자를 검색하십시오.

이 사용자는 이미 ID 제공자에 있어야 합니다.

- d) 각 사용자를 찾으면 역할을 부여하십시오. 이는 IBM MQ Console을 사용하여 IBM MQ를 관리하려면 "Administrator" 또는 "Cluster Administrator"여야 합니다.

4. 각 사용자를 네임스페이스에 추가한다.

- a) 편집할 팀을 선택합니다.

- b) **자원 > 자원 관리**를 선택하십시오.

- c) 이 팀에서 관리할 네임스페이스를 선택하십시오. 큐 관리자가 있는 모든 네임스페이스가 될 수 있습니다.

IBM MQ Operator 사용 시 모니터링

IBM MQ Operator에서 관리되는 큐 관리자는 Prometheus와 호환 가능한 메트릭을 생성할 수 있습니다.

Red Hat OpenShift Container Platform (OCP) 모니터링 스택을 사용하여 이러한 메트릭을 볼 수 있습니다. OCP에서 **메트릭** 탭을 연 다음 **관찰 > 메트릭**을 클릭하십시오. 큐 관리자 메트릭은 기본적으로 사용 가능하지만 **.spec.metrics.enabled**를 **false**로 설정하여 사용 불가능하게 할 수 있습니다.

Prometheus는 메트릭을 위한 시계열 데이터베이스 및 규칙 평가 엔진입니다. IBM MQ 컨테이너는 Prometheus에서 조회할 수 있는 메트릭 엔드포인트를 공개합니다. 메트릭은 MQ 시스템 토평에서 모니터링 및 활동 추적을 위해 생성합니다.

Red Hat OpenShift Container Platform에는 Prometheus 서버를 사용하는 사전 구성되고, 사전 설치되고, 자체 업데이트되는 모니터링 스택이 포함되어 있습니다. Red Hat OpenShift Container Platform 모니터링 스택은 사용자 정의 프로젝트를 모니터링하도록 구성해야 합니다. 자세한 정보는 [사용자 정의 프로젝트에 모니터링 사용을 참조하십시오](#). IBM MQ Operator은(는) Prometheus 운영자가 발견할 수 있는 메트릭을 사용하여 QueueManager을(를) 작성할 때 ServiceMonitor을(를) 작성합니다.

이전 버전의 IBM Cloud Pak for Integration에서는 대신 [IBM Cloud Platform Monitoring](#) 서비스를 사용하여 Prometheus 서버를 제공했습니다.

OpenShift CP4I IBM MQ Operator 사용 시 공개된 메트릭

큐 관리자 컨테이너는 Red Hat OpenShift Monitoring과 호환 가능한 메트릭을 공개할 수 있습니다.

메트릭	유형	설명
ibmmq_qmgr_commit_total	counter	커밋 수
ibmmq_qmgr_cpu_load_fifteen_minute_average_percentage	gauge	CPU 로드 - 15분 평균
ibmmq_qmgr_cpu_load_five_minute_average_percentage	gauge	CPU 로드 - 5분 평균
ibmmq_qmgr_cpu_load_one_minute_average_percentage	gauge	CPU 로드 - 1분 평균
ibmmq_qmgr_destructive_get_bytes_total	counter	중간 가져오기 후 삭제(destructive get) 합계 - 바이트 수
ibmmq_qmgr_destructive_get_total	counter	중간 가져오기 후 삭제(destructive get) 합계 - 수
ibmmq_qmgr_durable_subscription_alter_total	counter	지속 가능 구독 대체 수
ibmmq_qmgr_durable_subscription_create_total	counter	지속 가능 구독 작성 수
ibmmq_qmgr_durable_subscription_delete_total	counter	지속 가능 구독 삭제 수
ibmmq_qmgr_durable_subscription_resume_total	counter	지속 가능 구독 재개 수
ibmmq_qmgr_errors_file_system_free_space_percentage	gauge	MQ 오류 파일 시스템 - 여유 공간
ibmmq_qmgr_errors_file_system_in_use_bytes	gauge	MQ 오류 파일 시스템 - 사용 중인 바이트

메트릭	유형	설명
ibmmq_qmgr_expired_message_total	counter	만기된 메시지 수
ibmmq_qmgr_failed_browse_total	counter	실패한 찾아보기 수
ibmmq_qmgr_failed_mqcb_total	counter	실패한 MQCB 수
ibmmq_qmgr_failed_mqclose_total	counter	실패한 MQCLOSE 수
ibmmq_qmgr_failed_mqconn_mqconnx_total	counter	실패한 MQCONN/MQCONNX 수
ibmmq_qmgr_failed_mqget_total	counter	실패한 MQGET - 수
ibmmq_qmgr_failed_mqinq_total	counter	실패한 MQINQ 수
ibmmq_qmgr_failed_mqopen_total	counter	실패한 MQOPEN 수
ibmmq_qmgr_failed_mqput1_total	counter	실패한 MQPUT1 수
ibmmq_qmgr_failed_mqput_total	counter	실패한 MQPUT 수
ibmmq_qmgr_failed_mqset_total	counter	실패한 MQSET 수
ibmmq_qmgr_failed_mqsubrq_total	counter	실패한 MQSUBRQ 수
ibmmq_qmgr_failed_subscription_create_alter_resume_total	counter	실패한 구독 작성/변경/재개 수
ibmmq_qmgr_failed_subscription_delete_total	counter	구독 삭제 실패 수
ibmmq_qmgr_failed_topic_mqput_mqput1_total	counter	실패한 토픽 MQPUT/MQPUT1 수
ibmmq_qmgr_fdc_files	gauge	MQ FDC 파일 수
ibmmq_qmgr_log_file_system_in_use_bytes	gauge	로그 파일 시스템 - 사용 중인 바이트
ibmmq_qmgr_log_file_system_max_bytes	gauge	로그 파일 시스템 - 최대 바이트
ibmmq_qmgr_log_in_use_bytes	gauge	로그 - 사용 중인 바이트

메트릭	유형	설명
ibmmq_qmgr_log_logical_written_bytes_total	counter	로그 - 기록된 논리 바이트
ibmmq_qmgr_log_max_bytes	gauge	로그 - 최대 바이트
ibmmq_qmgr_log_physical_written_bytes_total	counter	로그 - 기록된 실제 바이트
ibmmq_qmgr_log_primary_space_in_use_percentage	gauge	로그 - 사용 중인 현재 1차 공간
ibmmq_qmgr_log_workload_primary_space_utilization_percentage	gauge	로그 - 워크로드 1차 공간 이용률
ibmmq_qmgr_log_write_latency_seconds	gauge	로그 - 쓰기 대기 시간
ibmmq_qmgr_log_write_size_bytes	gauge	로그 - 쓰기 크기
ibmmq_qmgr_mqcb_total	counter	MQCB 수
ibmmq_qmgr_mqclose_total	counter	MQCLOSE 수
ibmmq_qmgr_mqconn_mqconnx_total	counter	MQCONN/MQCONNX 수
ibmmq_qmgr_mqctl_total	counter	MQCTL 수
ibmmq_qmgr_mqdisc_total	counter	MQDISC 수
ibmmq_qmgr_mqinq_total	counter	MQINQ 수
ibmmq_qmgr_mqopen_total	counter	MQOPEN 수
ibmmq_qmgr_mqput_mqput1_bytes_total	counter	간격 MQPUT/MQPUT1 바이트 수 합계
ibmmq_qmgr_mqput_mqput1_total	counter	간격 MQPUT/MQPUT1 수 합계
ibmmq_qmgr_mqset_total	counter	MQSET 수
ibmmq_qmgr_mqstat_total	counter	MQSTAT 수
ibmmq_qmgr_mqsubrq_total	counter	MQSUBRQ 수

메트릭	유형	설명
ibmmq_qmgr_non_durable_subscription_create_total	counter	지속 불가능 구독 작성 수
ibmmq_qmgr_non_durable_subscription_delete_total	counter	지속 불가능 구독 삭제 수
ibmmq_qmgr_non_persistent_message_browse_bytes_total	counter	비지속 메시지 찾아보기 - 바이트 수
ibmmq_qmgr_non_persistent_message_browse_total	counter	비지속 메시지 찾아보기 - 수
ibmmq_qmgr_non_persistent_message_destructive_get_total	counter	비지속 메시지 가져오기 후 삭제(destructive get) - 수
ibmmq_qmgr_non_persistent_message_get_bytes_total	counter	가져온 비지속 메시지 - 바이트 수
ibmmq_qmgr_non_persistent_message_mqput1_total	counter	비지속 메시지 MQPUT1 수
ibmmq_qmgr_non_persistent_message_mqput_total	counter	비지속 메시지 MQPUT 수
ibmmq_qmgr_non_persistent_message_put_bytes_total	counter	비지속 메시지 넣기 - 바이트 수
ibmmq_qmgr_non_persistent_topic_mqput_mqput1_total	counter	비지속 - 토픽 MQPUT/MQPUT1 수
ibmmq_qmgr_persistent_message_browse_bytes_total	counter	지속 메시지 찾아보기 - 바이트 수
ibmmq_qmgr_persistent_message_browse_total	counter	지속 메시지 찾아보기 - 수
ibmmq_qmgr_persistent_message_destructive_get_total	counter	지속 메시지 가져오기 후 삭제(destructive get) - 수
ibmmq_qmgr_persistent_message_get_bytes_total	counter	가져온 지속 메시지 - 바이트 수
ibmmq_qmgr_persistent_message_mqput1_total	counter	지속 메시지 MQPUT1 수

메트릭	유형	설명
ibmmq_qmgr_persistent_message_mqput_total	counter	지속 메시지 MQPUT 수
ibmmq_qmgr_persistent_message_put_bytes_total	counter	지속 메시지 넣기 - 바이트 수
ibmmq_qmgr_persistent_topic_mqput_mqput1_total	counter	지속 - 토픽 MQPUT/MQPUT1 수
ibmmq_qmgr_published_to_subscribers_bytes_total	counter	구독자에게 발행 - 바이트 수
ibmmq_qmgr_published_to_subscribers_message_total	counter	구독자에게 발행 - 메시지 수
ibmmq_qmgr_purged_queue_total	counter	영구 제거된 큐 수
ibmmq_qmgr_queue_manager_file_system_free_space_percentage	gauge	큐 관리자 파일 시스템 - 여유 공간
ibmmq_qmgr_queue_manager_file_system_in_use_bytes	gauge	큐 관리자 파일 시스템 - 사용 중인 바이트
ibmmq_qmgr_ram_free_percentage	gauge	RAM 여유분 백분율
ibmmq_qmgr_ram_usage_estimate_for_queue_manager_bytes	gauge	RAM 바이트 합계 - 큐 관리자 추정치
ibmmq_qmgr_rollback_total	counter	롤백 수
ibmmq_qmgr_system_cpu_time_estimate_for_queue_manager_percentage	gauge	시스템 CPU 시간 - 큐 관리자의 백분율 추정치
ibmmq_qmgr_system_cpu_time_percentage	gauge	시스템 CPU 시간 백분율
ibmmq_qmgr_topic_mqput_mqput1_total	counter	토픽 MQPUT/MQPUT1 간격 합계
ibmmq_qmgr_topic_put_bytes_total	counter	간격 토픽 바이트 넣기 합계
ibmmq_qmgr_trace_file_system_free_space_percentage	gauge	MQ 추적 파일 시스템 - 여유 공간

메트릭	유형	설명
ibmmq_qmgr_trace_file_system_in_use_bytes	gauge	MQ 추적 파일 시스템 - 사용 중인 바이트
ibmmq_qmgr_user_cpu_time_estimate_for_queue_manager_percentage	gauge	사용자 CPU 시간 - 큐 관리자의 백분율 추정치
ibmmq_qmgr_user_cpu_time_percentage	gauge	사용자 CPU 시간 백분율

CP4I > CD > V9.2.2 IBM MQ 인증 컨테이너에 대한 원시 HA 큐 관리자의 상태 보기

IBM MQ 인증 컨테이너의 경우 실행 중인 Pods 중 하나에서 **dspmqr** 명령을 실행하여 원시 HA 인스턴스의 상태를 볼 수 있습니다.

이 태스크 정보

중요사항:

실행 중인 팟(Pod) 중 하나에서 **dspmqr** 명령을 사용하여 큐 관리자 인스턴스의 운영 상태를 볼 수 있습니다. 리턴되는 정보는 인스턴스가 활성화인지 또는 복제본인지에 따라 달라집니다. 활성화 인스턴스에서 제공하는 정보는 결정적이며 복제본 노드의 정보는 이전 정보일 수 있습니다.

다음 조치를 수행할 수 있습니다.

- 현재 노드의 큐 관리자 인스턴스가 활성화인지 또는 복제본인지 여부를 확인합니다.
- 현재 노드에서 인스턴스의 고유 HA 운영 상태를 확인합니다.
- 고유 HA 구성에서 세 개 인스턴스 모두의 운영 상태를 확인합니다.

다음 상태 필드는 고유 HA 구성 상태를 보고하는 데 사용됩니다.

ROLE

인스턴스의 현재 역할을 지정하며 Active, Replica 또는 Unknown 중 하나입니다.

INSTANCE

crtmqm 명령의 **-lr** 옵션을 사용하여 작성된 경우 큐 관리자의 이 인스턴스에 대해 제공된 이름입니다.

INSYNC

인스턴스가 필요한 경우 활성화 인스턴스로서 인계받을 수 있는지 여부를 표시합니다.

QUORUM

number_of_instances_in-sync/number_of_instances_configured 양식으로 쿼럼 상태를 보고합니다.

REPLADDR

큐 관리자 인스턴스의 복제 이름.

CONNECTV

노드가 활성화 인스턴스에 연결되어 있는지 여부를 표시합니다.

BACKLOG

인스턴스가 뒤에 있는 KB 수를 표시합니다.

CONNINST

이름 지정된 인스턴스가 이 인스턴스에 연결되는지 여부를 표시합니다.

ALTDAT

이 정보가 마지막으로 업데이트된 날짜를 표시합니다(업데이트된 적이 없으면 공백).

ALTTIME

이 정보가 마지막으로 업데이트된 시간을 표시합니다(업데이트된 적이 없으면 공백).

프로시저

- 큐 관리자의 일부인 포트를 찾으십시오.

```
oc get pod --selector app.kubernetes.io/instance=nativeha-qm
```

- 포크 중 하나에서 dspmq을(를) 실행하십시오.

```
oc exec -t Pod dspmq
```

```
oc rsh Pod
```

대화식 셸의 경우 dspmq을(를) 직접 실행할 수 있습니다.

- 큐 관리자 인스턴스가 활성 인스턴스로 실행 중인지 또는 복제본으로 실행 중인지를 판별합니다.

```
oc exec -t Pod dspmq -o status -m QMgrName
```

큐 관리자 BOB의 활성 인스턴스는 다음 상태를 보고합니다.

```
QMNAME(BOB)          STATUS(Running)
```

큐 관리자 BOB의 복제본 인스턴스는 다음 상태를 보고합니다.

```
QMNAME(BOB)          STATUS(Replica)
```

비활성 인스턴스는 다음 상태를 보고합니다.

```
QMNAME(BOB)          STATUS(Ended Immediately)
```

- 지정된 팟(Pod)에서 인스턴스의 고유 HA 운영 상태를 판별합니다.

```
oc exec -t Pod dspmq -o nativeha -m QMgrName
```

큐 관리자 BOB의 활성 인스턴스는 다음 상태를 보고합니다.

```
QMNAME(BOB)          ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
```

큐 관리자 BOB의 복제본 인스턴스는 다음 상태를 보고합니다.

```
QMNAME(BOB)          ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
```

큐 관리자 BOB의 비활성 인스턴스는 다음 상태를 보고합니다.

```
QMNAME(BOB)          ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
```

- 고유 HA 구성에서 모든 인스턴스의 고유 HA 운영 상태를 판별합니다.

```
oc exec -t Pod dspmq -o nativeha -x -m QMgrName
```

큐 관리자 BOB의 활성 인스턴스를 실행 중인 노드에서 이 명령을 실행하는 경우 다음 상태를 수신합니다.

```
QMNAME(BOB)          ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

큐 관리자 BOB의 복제본 인스턴스를 실행 중인 노드에서 이 명령을 실행하면, 다음 상태를 수신할 수 있습니다. 이는 복제본 중 하나가 뒤처지고 있음을 표시합니다.

```
QMNAME(BOB)          ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
```

```
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(No) BACKLOG(435)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

큐 관리자 BOB의 비활성 인스턴스를 실행 중인 노드에서 이 명령을 실행하면 다음 상태를 수신할 수 있습니다.

```
QMNAME(BOB) ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
INSTANCE(inst1) ROLE(Unknown) REPLADDR(9.20.123.45) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
INSTANCE(inst2) ROLE(Unknown) REPLADDR(9.20.123.46) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
INSTANCE(inst3) ROLE(Unknown) REPLADDR(9.20.123.47) CONNACTV(No) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
```

인스턴스가 계속 무엇이 활성화인지 그리고 무엇이 복제본인지 조정하는 경우 명령을 실행하면, 다음 상태를 수신할 수 있습니다.

```
QMNAME(BOB) STATUS(Negotiating)
```

관련 참조

[dspmq\(큐 관리자 표시\) 명령](#)

89 페이지의 『예: 고유 HA 큐 관리자 구성』

이 예제는 IBM MQ Operator를 사용하여 원시 고가용성 기능을 사용하여 Red Hat OpenShift Container Platform (OCP) 로 큐 관리자를 배치하는 방법을 보여줍니다.

Red Hat OpenShift CLI를 사용하여 큐 관리자 구성 백업 및 복원

큐 관리자 구성을 백업하면 큐 관리자 구성이 유실된 경우 해당 정의에서 큐 관리자를 다시 빌드할 수 있습니다. 이 프로시저에서는 큐 관리자 로그 데이터를 백업하지 않습니다. 메시지의 임시 네이처 때문에 히스토리 로그 데이터는 복원 시점에서 무관할 수 있습니다.

시작하기 전에

cloudctl login(IBM Cloud Pak for Integration의 경우) 또는 **oc login**을 사용하여 클러스터에 로그인하십시오.

프로시저

- 큐 관리자 구성을 백업하십시오.

dmpmqcfig 명령을 사용하여 IBM MQ 큐 관리자의 구성을 덤프할 수 있습니다.

- a) 큐 관리자의 팟(pod) 이름을 가져옵니다.

예를 들어, *queue_manager_name*이 QueueManager 자원인 다음 명령을 실행할 수 있습니다.

```
oc get pods --selector app.kubernetes.io/name=ibm-mq,app.kubernetes.io/instance=queue_manager_name
```

- b) 로컬 시스템의 파일로 출력 경로를 지정하여 팟(Pod)에서 **dmpmqcfig** 명령을 실행하십시오.

dmpmqcfig는 큐 관리자의 MQSC 구성을 출력합니다.

```
oc exec -it pod_name -- dmpmqcfig > backup.mqsc
```

- 큐 관리자 구성을 복원하십시오.

이전 단계에서 간략히 설명한 백업 프로시저에 따르는 경우 큐 관리자 구성을 포함하는 *backup.mqsc* 파일을 보유해야 합니다. 이 파일을 새 큐 관리자에 적용하여 구성을 복원할 수 있습니다.

- a) 큐 관리자의 팟(pod) 이름을 가져옵니다.

예를 들어, `queue_manager_name`이 QueueManager 자원인 다음 명령을 실행할 수 있습니다.

```
oc get pods --selector app.kubernetes.io/name=ibm-mq,app.kubernetes.io/instance=queue_manager_name
```

b) `backup.mqsc` 파일의 콘텐츠 경로를 지정하여 팟(Pod)에서 `runmqsc` 명령을 실행하십시오.

```
oc exec -i pod_name -- runmqsc < backup.mqsc
```

OpenShift CP4I IBM MQ Operator 의 문제점 해결

IBM MQ Operator에 문제가 있는 경우에는 설명된 기술을 사용하여 문제를 진단하고 해결하십시오.

프로시저

- [118 페이지의 『문제점 해결: 큐 관리자 데이터에 대한 액세스 권한 수집』](#)

OpenShift CP4I 문제점 해결: 큐 관리자 데이터에 대한 액세스 권한 수집

PVC 검사기 도구를 사용하여 큐 관리자 팟 (Pod) 에 원격 셸을 설정할 수 없는 큐 관리자 PVC의 파일에 대한 액세스 권한을 얻으십시오. 이는 팟 (Pod) 이 **Error** 또는 **CrashLoopBackOff** 상태에 있기 때문일 수 있습니다. 이 도구는 IBM MQ Operator에 의해 배치된 큐 관리자와 함께 사용하도록 설계되었습니다.

시작하기 전에

PVC 검사기 도구를 사용합니다. 큐 관리자 네임스페이스에 대한 액세스 권한이 있어야 합니다.

이 태스크 정보

문제점 해결을 돕기 위해 지정된 큐 관리자와 연관된 PVC (Persistent Volume Claims) 에 저장된 데이터에 액세스할 수 있습니다. 이를 수행하려면 도구를 사용하여 검사기 팟 (Pod) 세트에 PVC를 마운트하십시오. 그런 다음 원격 셸을 검사기 팟 (Pod) 에 가져와서 파일을 읽을 수 있습니다.

배치 유형에 따라 1-3개의 검사기 팟 (Pod) 이 작성됩니다. 고유 HA 또는 다중 인스턴스 큐 관리자의 지정된 팟 (Pod) 에 특정한 볼륨은 연관된 PVC 검사기 팟 (Pod) 에서 사용 가능합니다. 공유 볼륨은 모든 검사기에서 사용 가능합니다. 검사기 팟 (Pod) 의 이름에는 연관된 큐 관리자 팟 (Pod) 의 이름이 포함되어 있습니다.

프로시저

1. MQ PVC 검사기 도구를 다운로드하십시오.

이 도구는 <https://github.com/ibm-messaging/mq-pvc-tool>에서 사용 가능합니다.

2. 클러스터에 로그인했는지 확인하십시오.
3. 큐 관리자의 이름 및 큐 관리자가 실행 중인 네임스페이스를 찾으십시오.
4. 큐 관리자에 대해 검사기 도구를 실행하십시오.
 - a) 큐 관리자 이름 및 해당 네임스페이스 이름을 지정하여 다음 명령을 실행하십시오.

```
./pvc-tool.sh queue_manager_name queue_manager_namespace_name
```

- b) 도구가 완료되면 다음 명령을 실행하여 작성 중인 검사기 팟 (Pod) 을 보십시오.

```
oc get pods
```

5. 검사기 팟 (Pod) 에 마운트된 파일을 보십시오.

- a) 각 PVC 검사기 팟 (Pod) 은 큐 관리자 팟 (Pod) 과 연관되므로 여러 검사기 팟 (Pod) 이 있을 수 있습니다. 다음 명령을 실행하여 이러한 팟 (Pod) 중 하나에 액세스하십시오.

```
oc ish pvc-inspector-pod-name
```

마운트된 PVC 디렉토리를 포함하는 디렉토리에 배치됩니다.

b) 다음 명령을 실행하여 팟 (Pod) 에 대한 원격 셸을 여십시오.

```
ls
```

c) 마운트된 PVC와 동일한 이름의 디렉토리를 볼 수 있습니다. 이 디렉토리를 찾아보고 큐 관리자 PVC의 파일에 액세스하십시오. PVC 목록을 보려면 원격 셸 세션 외부에서 다음 명령을 실행하십시오.

```
oc get pvc
```

d) 다음 명령을 실행하여 도구에서 작성된 팟 (Pod) 을 정리하십시오.

```
'oc delete pods -l tool=mq-pvc-inspector
```

OpenShift CP4I IBM MQ Operator에 대한 API 참조

IBM MQ는 Red Hat OpenShift 컨테이너 플랫폼과의 기본 통합을 제공하는 Kubernetes Operator를 제공합니다.

OpenShift CP4I mq.ibm.com/v1beta1용 API 참조

v1beta1 API를 사용하여 QueueManager 자원을 작성하고 관리할 수 있습니다.

OpenShift CP4I CD EUS mq.ibm.com/v1beta1에 대한 라이선스 부여 참조

현재 라이선스 버전

spec.license.license 필드에는 승인하는 라이선스에 대한 라이선스 ID가 있어야 합니다. 유효값은 다음과 같습니다.

다음의 값 spec.license.license	다음의 값 spec.license.use	라이선스 정보	해당 IBM MQ 버전
L-RJON-C7QG3S	Production 또는 NonProduction	IBM Cloud Pak for Integration 2021.4.1	9.2.4 또는 9.2.5
L-RJON-C7QFZX	Production 또는 NonProduction	IBM Cloud Pak for Integration Limited Edition 2021.4.1	9.2.4 또는 9.2.5
L-RJON-C5CSNH	Production 또는 NonProduction	IBM Cloud Pak for Integration 2021.3.1	9.2.3 또는 9.2.4
L-RJON-C5CSM2	Production 또는 NonProduction	IBM Cloud Pak for Integration Limited Edition 2021.3.1	9.2.3 또는 9.2.4
L-RJON-BZFQU2	Production 또는 NonProduction	IBM Cloud Pak for Integration 2021.2.1	9.2.3
L-RJON-BZFQSB	Production 또는 NonProduction	IBM Cloud Pak for Integration Limited Edition 2021.2.1	9.2.3
L-RJON-BUVMQX	Production 또는 NonProduction	IBM Cloud Pak for Integration 2020.4.1	9.2.0 EUS 또는 9.2.1
L-RJON-BUVMYB	Production 또는 NonProduction	IBM Cloud Pak for Integration Limited Edition 2020.4.1	9.2.0 EUS 또는 9.2.1

다음의 값 <code>spec.license.license</code>	다음의 값 <code>spec.license.use</code>	라이선스 정보	해당 IBM MQ 버전
L-APIG-BZDDDY	Production	IBM MQ Advanced 및 IBM MQ Advanced for Non-Production Environment 9.2 -07/2021	9.2.3, 9.2.4 또는 9.2.5
L-APIG-BYHCL7	Development	IBM MQ Advanced for Developers (Non-Warranted) V9.2 -07/2021	9.2.3, 9.2.4 또는 9.2.5
L-APIG-BVJJB3	Production	IBM MQ Advanced 및 IBM MQ Advanced for Non-Production Environment 9.2 -03/2021	9.2.2
L-APIG-BMJJBM	Production	IBM MQ Advanced V9.2	9.2.0 CD 또는 9.2.1
L-APIG-BMKG5H	Development	IBM MQ Advanced for Developers(Non-Warranted) V9.2	9.2.0 CD, 9.2.1 또는 9.2.2

라이선스 버전이 지정되며, 항상 IBM MQ의 버전과 동일하지는 않습니다.

이전 라이선스 버전

`spec.license.license` 필드에는 승인하는 라이선스에 대한 라이선스 ID가 있어야 합니다. 유효값은 다음과 같습니다.

다음의 값 <code>spec.license.license</code>	다음의 값 <code>spec.license.use</code>	라이선스 정보	해당 IBM MQ 버전
L-RJON-BXUPZ2	Production 또는 NonProduction	IBM Cloud Pak for Integration 2021.1.1	9.2.2
L-RJON-BXUQ34	Production 또는 NonProduction	IBM Cloud Pak for Integration Limited Edition 2021.1.1	9.2.2
L-RJON-BYRMW	NonProduction	IBM Cloud Pak for Integration Eval-데모 2021.1.1. IBM MQ Operator 1.5 에서만 원시 HA 와 함께 사용하기 위한 초기 릴리스입니다.	9.2.2
L-RJON-BQPGWD	Production 또는 NonProduction	IBM Cloud Pak for Integration 2020.3.1	9.2.0 CD
L-RJON-BN7PN3	Production 또는 NonProduction	IBM Cloud Pak for Integration 2020.2.1	9.1.5 또는 9.2.0 CD
L-RJON-BPHL2Y	Production 또는 NonProduction	IBM Cloud Pak for Integration Limited Edition 2020.2.1	9.1.5
L-APIG-BJAKBF	Production	IBM MQ Advanced V9.1 -04/2020	9.1.5
L-APIG-BM7GDH	Development	IBM MQ Advanced for Developers(Non-Warranted) V9.1 - 04/2020	9.1.5

라이선스 버전이 지정되며, 항상 IBM MQ의 버전과 동일하지는 않습니다.

QueueManager

QueueManager는 애플리케이션에 큐잉 및 발행/구독 서비스를 제공하는 IBM MQ 서버입니다.

필드	설명
apiVersion 문자열	APIVersion은 오브젝트의 이 표현의 버전화된 스키마를 정의합니다. 서버는 인식된 스키마를 최신 내부 값으로 변환해야 하며 인식되지 않는 값을 거부할 수 있습니다. 추가 정보: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources .
kind 문자열	Kind는 이 오브젝트가 나타내는 REST 자원을 나타내는 문자열 값입니다. 서버는 클라이언트가 요청을 제출한 엔드포인트에서 이를 추론할 수 있습니다. CamelCase에서 업데이트할 수 없습니다. 추가 정보: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds .
metadata	
spec QueueManagerSpec	원하는 상태의 QueueManager입니다.
status QueueManagerStatus	QueueManager의 관찰된 상태입니다.

.spec

원하는 상태의 QueueManager입니다.

다음과 같이 표시됩니다.

- 121 페이지의 『QueueManager』

필드	설명
affinity	표준 Kubernetes 연관관계 규칙입니다. 자세한 정보는 https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#affinity-v1-core 를 참조하십시오.
annotations Annotations	annotations 필드는 팟(Pod) 어노테이션의 전달자 역할을 수행합니다. 사용자는 임의의 어노테이션을 이 필드에 추가하여 팟(Pod)에 적용되도록 할 수 있습니다. 이 필드의 어노테이션은 제공되는 경우 기본 어노테이션을 덮어씁니다. MQ Operator 1.3.0 이상이 필요합니다.
imagePullSecrets LocalObjectReference 배열	이 QueueManager가 사용하는 이미지를 PULL하기 위해 사용할 동일한 네임스페이스의 본인확인정보에 대한 선택적 참조 목록입니다. 지정된 경우 이러한 본인확인정보는 사용할 개별 풀러 구현으로 전달됩니다. 예를 들어, docker의 경우, DockerConfig 유형 본인확인정보만이 인정됩니다. 자세한 정보는 https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod 를 참조하십시오.
labels Labels	labels 필드는 팟(Pod) 레이블의 전달자 역할을 수행합니다. 사용자는 임의의 레이블을 이 필드에 추가하여 팟(Pod)에 적용되도록 할 수 있습니다. 이 필드의 레이블은 제공되는 경우 기본 레이블을 덮어씁니다. MQ Operator 1.3.0 이상이 필요합니다.
license 라이선스	라이선스의 승인을 제어하는 설정 및 사용할 라이선스 메트릭입니다.
pki PKI	TLS(Transport Layer Security) 또는 AMS(Advanced Message Security)에서 사용할 키와 인증서를 정의하기 위한 공개 키 인프라 설정
queueManager QueueManagerConfig	큐 관리자 컨테이너 및 기본 큐 관리자에 대한 설정입니다.

필드	설명
securityContext SecurityContext	큐 관리자 팟(Pod)의 securityContext에 추가할 보안 설정입니다.
template Template	Kubernetes 자원을 위한 고급 템플릿입니다. 템플릿을 사용하면 사용자가 IBM MQ가 StatefulSet, Pods 및 Services와 같이 기본 Kubernetes 자원을 생성하는 방법을 오버라이드할 수 있습니다. 이는 고급 사용자만을 위한 것이며 잘못 사용된 경우 MQ의 정상 작동을 방해할 가능성이 있기 때문입니다. QueueManager 자원의 다른 위치에 지정된 모든 값은 템플릿의 설정으로 대체됩니다.
terminationGracePeriod Seconds 정수	팟(Pod)가 적절하게 종료되는 선택적 기간(초)입니다. 값은 음이 아닌 정수여야 합니다. 0값은 즉시 삭제를 표시합니다. 큐 관리자 종료가 시도되는 대상 시간이며, 애플리케이션 연결 중단의 단계를 에스컬레이션합니다. 필요한 경우 필수 큐 관리자 유지보수 태스크가 인터럽트됩니다. 기본적으로 30초로 설정됩니다.
tracing TracingConfig	통합 운영 대시보드에 대한 클라우드 팩과의 통합을 위한 설정입니다.
version 문자열	사용할 MQ의 버전을 제어하는 설정입니다(필수). 예를 들어, 9.1.5.0-r2는 컨테이너 이미지의 두 번째 개정판을 사용하여 MQ 버전 9.1.5.0을 지정합니다. 컨테이너 특정 수정사항은 종종 기본 이미지에 대한 수정사항과 같은 개정판에 적용됩니다.
web WebServerConfig	MQ 웹 서버를 위한 설정입니다.

.spec.annotations

annotations 필드는 팟(Pod) 어노테이션의 전달자 역할을 수행합니다. 사용자는 임의의 어노테이션을 이 필드에 추가하여 팟(Pod)에 적용되도록 할 수 있습니다. 이 필드의 어노테이션은 제공되는 경우 기본 어노테이션을 덮어 씁니다. MQ Operator 1.3.0 이상이 필요합니다.

다음과 같이 표시됩니다.

- [121 페이지의 『.spec』](#)

.spec.imagePullSecrets

LocalObjectReference는 동일한 네임스페이스 내에서 참조된 오브젝트를 찾을 수 있는 충분한 정보가 포함됩니다.

다음과 같이 표시됩니다.

- [121 페이지의 『.spec』](#)

필드	설명
name 문자열	참조 이름입니다. 자세한 정보: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names TODO: Add other useful fields. apiVersion, kind, uid?.

.spec.labels

labels 필드는 팟(Pod) 레이블의 전달자 역할을 수행합니다. 사용자는 임의의 레이블을 이 필드에 추가하여 팟(Pod)에 적용되도록 할 수 있습니다. 이 필드의 레이블은 제공되는 경우 기본 레이블을 덮어 씁니다. MQ Operator 1.3.0 이상이 필요합니다.

다음과 같이 표시됩니다.

- [121 페이지의 『.spec』](#)

.spec.license

라이센스의 승인을 제어하는 설정 및 사용할 라이선스 메트릭입니다.

다음과 같이 표시됩니다.

- [121 페이지의 『.spec』](#)

필드	설명
accept boolean	이 소프트웨어와 연관된 라이선스에 동의하는지 여부입니다(필수).
license 문자열	승인 중인 라이선스의 ID입니다. 사용 중인 MQ의 버전으로 올바른 라이선스 ID여야 합니다. 올바른 값은 http://ibm.biz/BdqvCF 를 참조하십시오.
metric 문자열	사용할 라이선스 메트릭을 지정하는 설정입니다. 예를 들어, ProcessorValueUnit, VirtualProcessorCore 또는 ManagedVirtualServer입니다. MQ 라이선스를 사용하는 경우에는 기본적으로 ProcessorValueUnit으로, Cloud Pak for Integration 라이선스를 사용하는 경우에는 기본적으로 VirtualProcessorCore로 설정됩니다.
use 문자열	소프트웨어가 사용될 방법을 제어하는 설정이며, 여기서 라이선스는 다중 사용을 지원합니다. 올바른 값은 http://ibm.biz/BdqvCF 를 참조하십시오.

.spec.pki

TLS(Transport Layer Security) 또는 AMS(Advanced Message Security)에서 사용할 키와 인증서를 정의하기 위한 공개 키 인프라 설정

다음과 같이 표시됩니다.

- [121 페이지의 『.spec』](#)

필드	설명
keys PKISource array	큐 관리자의 키 저장소에 추가할 개인 키입니다.
trust PKISource array	큐 관리자의 키 저장소에 추가할 인증서입니다.

.spec.pki.keys

PKISource는 키 또는 인증서와 같은 공개 키 인프라 정보의 소스를 정의합니다.

다음과 같이 표시됩니다.

- [123 페이지의 『.spec.pki』](#)

필드	설명
name 문자열	이름은 키 또는 인증서의 레이블로 사용됩니다. 소문자 영숫자 문자열이어야 합니다.
secret Secret	Kubernetes 본인확인정보를 사용하여 키를 제공합니다.

.spec.pki.keys.secret

Kubernetes 본인확인정보를 사용하여 키를 제공합니다.

다음과 같이 표시됩니다.

- [123 페이지의 『.spec.pki.keys』](#)

필드	설명
items 배열	큐 관리자 컨테이너에 추가해야 하는 Kubernetes 본인확인정보 내 키입니다.
secretName 문자열	Kubernetes 본인확인정보의 이름입니다.

.spec.pki.trust

PKISource는 키 또는 인증서와 같은 공개 키 인프라 정보의 소스를 정의합니다.

다음과 같이 표시됩니다.

- [123 페이지의 『.spec.pki』](#)

필드	설명
name 문자열	이름은 키 또는 인증서의 레이블로 사용됩니다. 소문자 영숫자 문자열이어야 합니다.
secret Secret	Kubernetes 본인확인정보를 사용하여 키를 제공합니다.

.spec.pki.trust.secret

Kubernetes 본인확인정보를 사용하여 키를 제공합니다.

다음과 같이 표시됩니다.

- [124 페이지의 『.spec.pki.trust』](#)

필드	설명
items 배열	큐 관리자 컨테이너에 추가해야 하는 Kubernetes 본인확인정보 내 키입니다.
secretName 문자열	Kubernetes 본인확인정보의 이름입니다.

.spec.queueManager

큐 관리자 컨테이너 및 기본 큐 관리자에 대한 설정입니다.

다음과 같이 표시됩니다.

- [121 페이지의 『.spec』](#)

필드	설명
availability Availability	활성-대기 쌍을 사용하는지 아니면 고유 고가용성을 사용하는지와 같이 큐 관리자의 가용성 설정입니다.
debug boolean	컨테이너 특정 코드에서 컨테이너 로그로 디버그 메시지를 로그할 지 여부입니다. 기본적으로 false로 설정됩니다.
image 문자열	사용할 컨테이너 이미지입니다.
imagePullPolicy 문자열	kubelet이 지정된 이미지를 풀하려고 시도할 때 제어하는 설정입니다. 기본적으로 IfNotPresent로 설정됩니다.
ini INISource array	큐 관리자에 INI를 제공하기 위한 설정입니다. MQ Operator 1.1.0 이상이 필요합니다.
livenessProbe QueueManagerLivenessProbe	라이브 프로브를 제어하는 설정입니다.
logFormat 문자열	이 컨테이너에 사용할 로그 형식입니다. 컨테이너에서 JSON 형식의 로그에 JSON을 사용합니다. 텍스트 형식으로 된 메시지에 기본을 사용하십시오. 기본적으로 Basic으로 설정됩니다.
metrics QueueManagerMetrics	Prometheus-스타일 메트릭의 설정입니다.
mqsc MQSCSource array	큐 매니저에 MQSC를 제공하기 위한 설정입니다. MQ Operator 1.1.0 이상이 필요합니다.

필드	설명
name 문자열	metadata.name과 다른 경우 기본 MQ Queue Manager의 이름입니다. 이름으로 Kubernetes 규칙을 준수하지 않는 Queue Manager 이름을 원하는 경우 이 필드를 사용하십시오(예를 들어, 대문자를 포함한 이름).
readinessProbe QueueManagerReadinessProbe	준비 프로브를 제어하는 설정입니다.
resources Resources	자원 요구사항을 제어하는 설정입니다.
route Route	큐 관리자 라우트에 대한 설정입니다. MQ Operator 1.4.0 이상이 필요합니다.
startupProbe StartupProbe	시동 프로브를 제어하는 설정입니다. MultiInstance 및 NativeHA 배치에만 적용됩니다. MQ Operator 1.5.0 이상이 필요합니다.
storage QueueManagerStorage	큐 관리자의 지속적 볼륨 및 스토리지 클래스 사용을 제어하는 스토리지 설정입니다.

.spec.queueManager.availability

활성-대기 쌍을 사용하는지 아니면 고유 고가용성을 사용하는지와 같이 큐 관리자의 가용성 설정입니다.

다음과 같이 표시됩니다.

- [124 페이지의 『.spec.queueManager』](#)

필드	설명
tls Tls	NativeHA 복제본 사이에서의 보안 통신을 구성하기 위한 선택적 TLS 설정입니다. MQ Operator 1.5.0 이상이 필요합니다.
type 문자열	사용할 가용성 유형입니다. 단일 팟(Pod)에 SingleInstance를 사용하며, Kubernetes에 의해 자동으로 재시작됩니다(일부 경우에). 한 쌍의 팟(Pod)에 MultiInstance를 사용하며, 이 중 하나는 활성 큐 관리자이며 다른 하나는 대기입니다. 고유 고가용성 복제본의 경우 NativeHA를 사용합니다(MQ Operator 1.5.0 이상이 필요함). 기본적으로 SingleInstance로 설정됩니다. 세부사항은 http://ibm.biz/BdqAQa 를 참조하십시오.
updateStrategy 문자열	MultiInstance 및 NativeHA 큐 관리자에 사용할 업데이트 전략입니다. 큐 관리자 구성이 변경될 때마다 자동 롤링 업데이트를 사용으로 설정하려면 RollingUpdate를 사용하십시오. 자동 롤링 업데이트를 사용 안함으로 설정하려면 OnDelete를 사용하십시오. 그러면 팟(Pod)이 삭제되는 경우에만(외부 요인으로 트리거된 팟(Pod) 삭제 포함) 큐 관리자 변경사항이 적용됩니다. 기본값은 RollingUpdate입니다. MQ Operator 1.6.0 이상이 필요합니다.

.spec.queueManager.availability.tls

NativeHA 복제본 사이에서의 보안 통신을 구성하기 위한 선택적 TLS 설정입니다. MQ Operator 1.5.0 이상이 필요합니다.

다음과 같이 표시됩니다.

- [125 페이지의 『.spec.queueManager.availability』](#)

필드	설명
cipherSpec 문자열	NativeHA TLS를 위한 CipherSpec의 이름입니다.
secretName 문자열	Kubernetes 본인확인정보의 이름입니다.

.spec.queueManager.ini

INI 구성 파일의 소스입니다.

다음과 같이 표시됩니다.

- [124 페이지의 『.spec.queueManager』](#)

필드	설명
configMap ConfigMapINISource	configMap은 INI 정보를 포함하는 Kubernetes ConfigMap을 나타냅니다.
secret SecretINISource	secret은 INI 정보를 포함하는 Kubernetes 시크릿을 나타냅니다.

.spec.queueManager.ini.configMap

configMap은 INI 정보를 포함하는 Kubernetes ConfigMap을 나타냅니다.

다음과 같이 표시됩니다.

- [126 페이지의 『.spec.queueManager.ini』](#)

필드	설명
items 배열	적용되어야 하는 Kubernetes 소스 내 키입니다.
name 문자열	Kubernetes 소스의 이름입니다.

.spec.queueManager.ini.secret

secret은 INI 정보를 포함하는 Kubernetes 시크릿을 나타냅니다.

다음과 같이 표시됩니다.

- [126 페이지의 『.spec.queueManager.ini』](#)

필드	설명
items 배열	적용되어야 하는 Kubernetes 소스 내 키입니다.
name 문자열	Kubernetes 소스의 이름입니다.

.spec.queueManager.livenessProbe

라이브 프로브를 제어하는 설정입니다.

다음과 같이 표시됩니다.

- [124 페이지의 『.spec.queueManager』](#)

필드	설명
failureThreshold 정수	성공 후 프로브가 실패한 것으로 간주되는 최소 연속 실패입니다. 기본적으로 1로 설정됩니다.
initialDelaySeconds 정수	컨테이너가 시작된 후 프로브를 시작하기 전의 시간(초)입니다. SingleInstance의 경우 기본값은 90초입니다. MultiInstance 및 NativeHA 배치의 경우 기본값은 0초입니다. 추가 정보: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
periodSeconds 정수	프로브를 수행하는 빈도(초)입니다. 기본적으로 10초로 설정됩니다.
successThreshold 정수	실패 후 성공한 것으로 간주되는 프로브의 최소 연속 성공입니다. 기본적으로 1로 설정됩니다.

필드	설명
timeoutSeconds 정수	프로브가 제한시간을 초과한 시간(초)입니다. 기본적으로 5초로 설정됩니다. 추가 정보: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

.spec.queueManager.metrics

Prometheus-스타일 메트릭의 설정입니다.

다음과 같이 표시됩니다.

- [124 페이지의 『.spec.queueManager』](#)

필드	설명
enabled boolean	엔드포인트의 Prometheus 호환 메트릭 사용 설정 여부입니다. 기본적으로 true로 설정됩니다.

.spec.queueManager.mqsc

MQSC 구성 파일의 소스입니다.

다음과 같이 표시됩니다.

- [124 페이지의 『.spec.queueManager』](#)

필드	설명
configMap ConfigMapMQSCSource	configMap은 MQSC 정보를 포함하는 Kubernetes ConfigMap을 나타냅니다.
secret SecretMQSCSource	secret은 MQSC 정보를 포함하는 Kubernetes 시크릿을 나타냅니다.

.spec.queueManager.mqsc.configMap

configMap은 MQSC 정보를 포함하는 Kubernetes ConfigMap을 나타냅니다.

다음과 같이 표시됩니다.

- [127 페이지의 『.spec.queueManager.mqsc』](#)

필드	설명
items 배열	적용되어야 하는 Kubernetes 소스 내 키입니다.
name 문자열	Kubernetes 소스의 이름입니다.

.spec.queueManager.mqsc.secret

secret은 MQSC 정보를 포함하는 Kubernetes 시크릿을 나타냅니다.

다음과 같이 표시됩니다.

- [127 페이지의 『.spec.queueManager.mqsc』](#)

필드	설명
items 배열	적용되어야 하는 Kubernetes 소스 내 키입니다.
name 문자열	Kubernetes 소스의 이름입니다.

.spec.queueManager.readinessProbe

준비 프로브를 제어하는 설정입니다.

다음과 같이 표시됩니다.

- [124 페이지의 『.spec.queueManager』](#)

필드	설명
failureThreshold 정수	성공 후 프로브가 실패한 것으로 간주되는 최소 연속 실패입니다. 기본적으로 1로 설정됩니다.
initialDelaySeconds 정수	컨테이너가 시작된 후 프로브를 시작하기 전의 시간(초)입니다. SingleInstance의 경우 기본값은 10초입니다. MultiInstance 및 NativeHA 배치의 경우 기본값은 0입니다. 추가 정보: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
periodSeconds 정수	프로브를 수행하는 빈도(초)입니다. 기본적으로 5초로 설정됩니다.
successThreshold 정수	실패 후 성공한 것으로 간주되는 프로브의 최소 연속 성공입니다. 기본적으로 1로 설정됩니다.
timeoutSeconds 정수	프로브가 제한시간을 초과한 시간(초)입니다. 기본적으로 3초로 설정됩니다. 추가 정보: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

.spec.queueManager.resources

자원 요구사항을 제어하는 설정입니다.

다음과 같이 표시됩니다.

- [124 페이지의 『.spec.queueManager』](#)

필드	설명
limits Limits	CPU & 메모리 설정입니다.
requests 요청	CPU & 메모리 설정입니다.

.spec.queueManager.resources.limits

CPU & 메모리 설정입니다.

다음과 같이 표시됩니다.

- [128 페이지의 『.spec.queueManager.resources』](#)

필드	설명
cpu	
memory	

.spec.queueManager.resources.requests

CPU & 메모리 설정입니다.

다음과 같이 표시됩니다.

- [128 페이지의 『.spec.queueManager.resources』](#)

필드	설명
cpu	

필드	설명
memory	

.spec.queueManager.route

큐 관리자 라우트에 대한 설정입니다. MQ Operator 1.4.0 이상이 필요합니다.

다음과 같이 표시됩니다.

- [124 페이지의 『.spec.queueManager』](#)

필드	설명
enabled boolean	라우트 사용 설정 여부입니다. 기본적으로 true로 설정됩니다.

.spec.queueManager.startupProbe

시동 프로브를 제어하는 설정입니다. MultiInstance 및 NativeHA 배치에만 적용됩니다. MQ Operator 1.5.0 이상이 필요합니다.

다음과 같이 표시됩니다.

- [124 페이지의 『.spec.queueManager』](#)

필드	설명
failureThreshold 정수	프로브를 실패한 것으로 간주하는 최소 연속 실패 수입니다. 기본값은 60입니다.
initialDelaySeconds 정수	컨테이너가 시작된 후 프로브를 시작하기 전의 시간(초)입니다. 기본값은 0초입니다. 추가 정보: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
periodSeconds 정수	프로브를 수행하는 빈도(초)입니다. 기본적으로 5초로 설정됩니다.
successThreshold 정수	프로브를 성공한 것으로 간주하는 최소 연속 성공 수입니다. 기본적으로 1로 설정됩니다.
timeoutSeconds 정수	프로브가 제한시간을 초과한 시간(초)입니다. 기본적으로 5초로 설정됩니다. 추가 정보: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

.spec.queueManager.storage

큐 관리자의 지속적 볼륨 및 스토리지 클래스 사용을 제어하는 스토리지 설정입니다.

다음과 같이 표시됩니다.

- [124 페이지의 『.spec.queueManager』](#)

필드	설명
defaultClass 문자열	기본적으로 이 큐 관리자의 모든 지속적 볼륨에 적용할 스토리지 클래스입니다. 특정 지속적 볼륨은 고유한 스토리지 클래스를 정의할 수 있으며 이는 이 기본 스토리지 클래스 설정을 대체합니다. type of availability가 SingleInstance 또는 NativeHA인 경우 스토리지 클래스는 ReadWriteOnce 또는 ReadWriteMany 유형이 될 수 있습니다. type of availability가 MultiInstance인 경우 스토리지 클래스는 ReadWriteMany 유형이어야 합니다.
defaultDeleteClaim 부울	큐 관리자 삭제 시 모든 볼륨을 삭제해야 하는지의 여부입니다. 특정 지속적 볼륨은 deleteClaim에 대해 고유한 값을 정의할 수 있으며 이는 이 defaultDeleteClaim 설정을 대체합니다. 기본적으로 false로 설정됩니다.

필드	설명
<code>persistedData</code> QueueManagerOptionalVolume	구성, 큐 및 메시지를 포함하는 MQ 지속 데이터에 대한 PersistentVolume 세부 사항입니다. 다중 인스턴스 큐 관리자를 사용할 때 필요합니다.
<code>queueManager</code> QueueManagerVolume	일반적으로 <code>/var/mqm</code> 아래 데이터에 대한 기본 PersistentVolume입니다. 다른 볼륨이 지정되지 않은 경우 모든 지속 데이터 및 복구 로그를 포함합니다.
<code>recoveryLogs</code> QueueManagerOptionalVolume	MQ 복구 로그에 대한 지속적 볼륨 세부사항입니다. 다중 인스턴스 큐 관리자를 사용할 때 필요합니다.

.spec.queueManager.storage.persistedData

구성, 큐 및 메시지를 포함하는 MQ 지속 데이터에 대한 PersistentVolume 세부 사항입니다. 다중 인스턴스 큐 관리자를 사용할 때 필요합니다.

다음과 같이 표시됩니다.

- [129 페이지의 『.spec.queueManager.storage』](#)

필드	설명
<code>class</code> 문자열	이 볼륨에 사용할 스토리지 클래스입니다. 유형이 <code>persistent-claim</code> 인 경우에만 유효합니다. <code>type of availability</code> 가 <code>SingleInstance</code> 또는 <code>NativeHA</code> 인 경우 스토리지 클래스는 <code>ReadWriteOnce</code> 또는 <code>ReadWriteMany</code> 유형이 될 수 있습니다. <code>type of availability</code> 가 <code>MultiInstance</code> 인 경우 스토리지 클래스는 <code>ReadWriteMany</code> 유형이어야 합니다.
<code>deleteClaim</code> 부울	큐 관리자 삭제 시 이 볼륨을 삭제해야 하는지의 여부입니다.
<code>enabled</code> boolean	이 볼륨을 별도의 볼륨으로 사용할지 또는 기본 <code>queueManager</code> 볼륨에 있는지 여부입니다. 기본적으로 <code>false</code> 로 설정됩니다.
<code>size</code> 문자열	SI 단위를 포함하여 Kubernetes로 전달할 PersistentVolume의 크기입니다. 유형이 <code>persistent-claim</code> 인 경우에만 유효합니다. 예를 들어, <code>2Gi</code> 입니다. 기본적으로 <code>2Gi</code> 로 설정됩니다.
<code>sizeLimit</code> 문자열	임시 볼륨을 사용할 때 크기 제한입니다. 파일은 여전히 임시 디렉토리에 기록되므로 이 옵션을 사용하여 크기를 제한할 수 있습니다. <code>type</code> 이 <code>ephemeral</code> 인 경우에만 유효합니다.
<code>type</code> 문자열	사용할 볼륨의 유형입니다. 비지속적 스토리지를 사용하려면 <code>ephemeral</code> 를 선택하거나 지속적 볼륨을 사용하려면 <code>persistent-claim</code> 을 선택하십시오. 기본적으로 <code>persistent-claim</code> 으로 설정됩니다.

.spec.queueManager.storage.queueManager

일반적으로 `/var/mqm` 아래 데이터에 대한 기본 PersistentVolume입니다. 다른 볼륨이 지정되지 않은 경우 모든 지속 데이터 및 복구 로그를 포함합니다.

다음과 같이 표시됩니다.

- [129 페이지의 『.spec.queueManager.storage』](#)

필드	설명
class 문자열	이 볼륨에 사용할 스토리지 클래스입니다. 유형이 persistent-claim인 경우에만 유효합니다. type of availability가 SingleInstance 또는 NativeHA인 경우 스토리지 클래스는 ReadWriteOnce 또는 ReadWriteMany 유형이 될 수 있습니다. type of availability가 MultiInstance인 경우 스토리지 클래스는 ReadWriteMany 유형이어야 합니다.
deleteClaim 부울	큐 관리자 삭제 시 이 볼륨을 삭제해야 하는지의 여부입니다.
size 문자열	SI 단위를 포함하여 Kubernetes로 전달할 PersistentVolume의 크기입니다. 유형이 persistent-claim인 경우에만 유효합니다. 예를 들어, 2Gi입니다. 기본적으로 2Gi로 설정됩니다.
sizeLimit 문자열	임시 볼륨을 사용할 때 크기 제한입니다. 파일은 여전히 임시 디렉토리에 기록되므로 이 옵션을 사용하여 크기를 제한할 수 있습니다. type이 ephemeral인 경우에만 유효합니다.
type 문자열	사용할 볼륨의 유형입니다. 비지속적 스토리지를 사용하려면 ephemeral를 선택하거나 지속적 볼륨을 사용하려면 persistent-claim을 선택하십시오. 기본적으로 persistent-claim으로 설정됩니다.

.spec.queueManager.storage.recoveryLogs

MQ 복구 로그에 대한 지속적 볼륨 세부사항입니다. 다중 인스턴스 큐 관리자를 사용할 때 필요합니다. 다음과 같이 표시됩니다.

- [129 페이지의 『.spec.queueManager.storage』](#)

필드	설명
class 문자열	이 볼륨에 사용할 스토리지 클래스입니다. 유형이 persistent-claim인 경우에만 유효합니다. type of availability가 SingleInstance 또는 NativeHA인 경우 스토리지 클래스는 ReadWriteOnce 또는 ReadWriteMany 유형이 될 수 있습니다. type of availability가 MultiInstance인 경우 스토리지 클래스는 ReadWriteMany 유형이어야 합니다.
deleteClaim 부울	큐 관리자 삭제 시 이 볼륨을 삭제해야 하는지의 여부입니다.
enabled boolean	이 볼륨을 별도의 볼륨으로 사용할지 또는 기본 queueManager 볼륨에 있는지 여부입니다. 기본적으로 false로 설정됩니다.
size 문자열	SI 단위를 포함하여 Kubernetes로 전달할 PersistentVolume의 크기입니다. 유형이 persistent-claim인 경우에만 유효합니다. 예를 들어, 2Gi입니다. 기본적으로 2Gi로 설정됩니다.
sizeLimit 문자열	임시 볼륨을 사용할 때 크기 제한입니다. 파일은 여전히 임시 디렉토리에 기록되므로 이 옵션을 사용하여 크기를 제한할 수 있습니다. type이 ephemeral인 경우에만 유효합니다.
type 문자열	사용할 볼륨의 유형입니다. 비지속적 스토리지를 사용하려면 ephemeral를 선택하거나 지속적 볼륨을 사용하려면 persistent-claim을 선택하십시오. 기본적으로 persistent-claim으로 설정됩니다.

.spec.securityContext

큐 관리자 팟(Pod)의 securityContext에 추가할 보안 설정입니다. 다음과 같이 표시됩니다.

- [121 페이지의 『.spec』](#)

필드	설명
fsGroup 정수	팟(Pod)의 모든 컨테이너에 적용되는 특수 보조 그룹입니다. 일부 볼륨 유형을 사용하면 Kubelet가 팟(Pod)에서 소유할 해당 볼륨의 소유권을 변경할 수 있습니다. 1. 소유_gid는 FSGroup입니다. 2. setgid 비트가 설정됩니다(볼륨에 작성된 새 파일은 FSGroup이 소유함). 3. 권한 비트는 rw-rw----와 OR됩니다. 설정되지 않은 경우 Kubelet에서 모든 볼륨의 소유권과 권한을 수정하지 않습니다.
initVolumeAsRoot boolean	PersistentVolume을 초기화하는 컨테이너에서 사용하는 securityContext에 영향을 줍니다. 새로 프로비저닝된 볼륨에 액세스하기 위해 루트 사용자가 되어야 하는 스토리지 제공자를 사용하는 경우 이 값을 true로 설정하십시오. 이를 true로 설정하면 사용할 수 있는 SCC(Security Context Constraints)에 영향을 주며, 루트 사용자를 허용하는 SCC를 사용할 권한이 없는 경우 Queue Manager를 시작하는 데 실패할 수 있습니다. 기본적으로 false로 설정됩니다. 자세한 정보는 https://docs.openshift.com/container-platform/latest/authentication/managing-security-context-constraints.html 의 내용을 참조하십시오.
supplementalGroups 배열	컨테이너의 기본_gid 외에 각 컨테이너에서 실행되는 첫 번째 프로세스에 적용된 그룹 목록입니다. 지정하지 않으면 어떤 그룹도 컨테이너에 추가되지 않습니다.

.spec.template

Kubernetes 자원을 위한 고급 템플릿입니다. 템플릿을 사용하면 사용자가 IBM MQ가 StatefulSet, Pods 및 Services와 같이 기본 Kubernetes 자원을 생성하는 방법을 오버라이드할 수 있습니다. 이는 고급 사용자만을 위한 것이며 잘못 사용된 경우 MQ의 정상 작동을 방해할 가능성이 있기 때문입니다. QueueManager 자원의 다른 위치에 지정된 모든 값은 템플릿의 설정으로 대체됩니다.

다음과 같이 표시됩니다.

- [121 페이지의 『.spec』](#)

필드	설명
pod	팟(Pod)에 사용된 템플릿을 대체합니다. https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#podspec-v1-core 을 참조하십시오.

.spec.tracing

통합 운영 대시보드에 대한 클라우드 팩과의 통합을 위한 설정입니다.

다음과 같이 표시됩니다.

- [121 페이지의 『.spec』](#)

필드	설명
agent TracingAgent	통합용 클라우드 팩에서는 선택적 추적 에이전트에 대한 설정을 구성할 수 있습니다.
collector TracingCollector	통합용 클라우드 팩에서는 선택적 추적 콜렉터에 대한 설정을 구성할 수 있습니다.
enabled boolean	추적을 통해 통합 운영 대시보드의 클라우드 팩과의 통합을 사용할지 여부입니다. 기본적으로 false로 설정됩니다.
namespace 문자열	통합 운영 대시보드용 클라우드 팩이 설치된 네임스페이스입니다.

.spec.tracing.agent

통합용 클라우드 팩에서는 선택적 추적 에이전트에 대한 설정을 구성할 수 있습니다.

다음과 같이 표시됩니다.

- [132 페이지의 『.spec.tracing』](#)

필드	설명
image 문자열	사용할 컨테이너 이미지입니다.
imagePullPolicy 문자열	kubelet이 지정된 이미지를 풀하려고 시도할 때 제어하는 설정입니다. 기본적으로 IfNotPresent로 설정됩니다.
livenessProbe TracingProbe	라이브 프로브를 제어하는 설정입니다.
readinessProbe TracingProbe	준비 프로브를 제어하는 설정입니다.

.spec.tracing.agent.livenessProbe

라이브 프로브를 제어하는 설정입니다.

다음과 같이 표시됩니다.

- 132 페이지의 『.spec.tracing.agent』

필드	설명
failureThreshold 정수	성공 후 프로브가 실패한 것으로 간주되는 최소 연속 실패입니다. 기본적으로 1로 설정됩니다.
initialDelaySeconds 정수	라이브 프로브가 초기화되기 전에 컨테이너가 시작된 후의 시간(초)입니다. 기본적으로 10초로 설정됩니다. 추가 정보: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
periodSeconds 정수	프로브를 수행하는 빈도(초)입니다. 기본적으로 10초로 설정됩니다.
successThreshold 정수	실패 후 성공한 것으로 간주되는 프로브의 최소 연속 성공입니다. 기본적으로 1로 설정됩니다.
timeoutSeconds 정수	프로브가 제한시간을 초과한 시간(초)입니다. 기본적으로 2초로 설정됩니다. 추가 정보: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

.spec.tracing.agent.readinessProbe

준비 프로브를 제어하는 설정입니다.

다음과 같이 표시됩니다.

- 132 페이지의 『.spec.tracing.agent』

필드	설명
failureThreshold 정수	성공 후 프로브가 실패한 것으로 간주되는 최소 연속 실패입니다. 기본적으로 1로 설정됩니다.
initialDelaySeconds 정수	라이브 프로브가 초기화되기 전에 컨테이너가 시작된 후의 시간(초)입니다. 기본적으로 10초로 설정됩니다. 추가 정보: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
periodSeconds 정수	프로브를 수행하는 빈도(초)입니다. 기본적으로 10초로 설정됩니다.
successThreshold 정수	실패 후 성공한 것으로 간주되는 프로브의 최소 연속 성공입니다. 기본적으로 1로 설정됩니다.
timeoutSeconds 정수	프로브가 제한시간을 초과한 시간(초)입니다. 기본적으로 2초로 설정됩니다. 추가 정보: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

.spec.tracing.collector

통합용 클라우드 팩에서는 선택적 추적 콜렉터에 대한 설정을 구성할 수 있습니다.

다음과 같이 표시됩니다.

- 132 페이지의 『.spec.tracing』

필드	설명
image 문자열	사용할 컨테이너 이미지입니다.
imagePullPolicy 문자열	kubelet이 지정된 이미지를 풀하려고 시도할 때 제어하는 설정입니다. 기본적으로 IfNotPresent로 설정됩니다.
livenessProbe TracingProbe	라이브 프로브를 제어하는 설정입니다.
readinessProbe TracingProbe	준비 프로브를 제어하는 설정입니다.

.spec.tracing.collector.livenessProbe

라이브 프로브를 제어하는 설정입니다.

다음과 같이 표시됩니다.

- 134 페이지의 『.spec.tracing.collector』

필드	설명
failureThreshold 정수	성공 후 프로브가 실패한 것으로 간주되는 최소 연속 실패입니다. 기본적으로 1로 설정됩니다.
initialDelaySeconds 정수	라이브 프로브가 초기화되기 전에 컨테이너가 시작된 후의 시간(초)입니다. 기본적으로 10초로 설정됩니다. 추가 정보: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
periodSeconds 정수	프로브를 수행하는 빈도(초)입니다. 기본적으로 10초로 설정됩니다.
successThreshold 정수	실패 후 성공한 것으로 간주되는 프로브의 최소 연속 성공입니다. 기본적으로 1로 설정됩니다.
timeoutSeconds 정수	프로브가 제한시간을 초과한 시간(초)입니다. 기본적으로 2초로 설정됩니다. 추가 정보: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

.spec.tracing.collector.readinessProbe

준비 프로브를 제어하는 설정입니다.

다음과 같이 표시됩니다.

- 134 페이지의 『.spec.tracing.collector』

필드	설명
failureThreshold 정수	성공 후 프로브가 실패한 것으로 간주되는 최소 연속 실패입니다. 기본적으로 1로 설정됩니다.
initialDelaySeconds 정수	라이브 프로브가 초기화되기 전에 컨테이너가 시작된 후의 시간(초)입니다. 기본적으로 10초로 설정됩니다. 추가 정보: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
periodSeconds 정수	프로브를 수행하는 빈도(초)입니다. 기본적으로 10초로 설정됩니다.

필드	설명
successThreshold 정수	실패 후 성공한 것으로 간주되는 프로브의 최소 연속 성공입니다. 기본적으로 1로 설정됩니다.
timeoutSeconds 정수	프로브가 제한시간을 초과한 시간(초)입니다. 기본적으로 2초로 설정됩니다. 추가 정보: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

.spec.web

MQ 웹 서버를 위한 설정입니다.

다음과 같이 표시됩니다.

- [121 페이지의 『.spec』](#)

필드	설명
enabled boolean	웹 서버 사용 가능 여부입니다. 기본적으로 false로 설정됩니다.

.status

QueueManager의 관찰된 상태입니다.

다음과 같이 표시됩니다.

- [121 페이지의 『QueueManager』](#)

필드	설명
adminUiUrl 문자열	Admin UI의 URL입니다.
availability Availability	큐 관리자의 가용성 상태입니다.
conditions QueueManagerStatusCondition 배열	조건은 큐 관리자 상태의 사용 가능한 최신 관찰을 나타냅니다.
endpoints QueueManagerStatusEndpoint 배열	API 또는 UI 엔드 포인트와 같이 이 Queue Manager가 노출하는 엔드포인트에 대한 정보입니다.
name 문자열	큐 관리자의 이름입니다.
phase 문자열	큐 관리자 상태의 단계입니다.
versions QueueManagerStatusVersion	사용 중인 MQ의 버전 및 IBM Entitled Registry에서 사용 가능한 기타 버전입니다.

.status.availability

큐 관리자의 가용성 상태입니다.

다음과 같이 표시됩니다.

- [135 페이지의 『.status』](#)

필드	설명
initialQuorumEstablished boolean	NativeHA에 대해 초기 쿼럼이 설정되었는지 여부입니다.

.status.conditions

QueueManagerStatusCondition은 Queue Manager의 조건을 정의합니다.

다음과 같이 표시됩니다.

- [135 페이지의 『.status』](#)

필드	설명
lastTransitionTime 문자열	상태가 한 상태에서 다른 상태로 전이된 마지막 시간입니다.
message 문자열	마지막 전이에 대한 세부사항을 표시하는 사용자가 읽을 수 있는 메시지입니다.
reason 문자열	이 상태의 마지막 전이에 대한 이유입니다.
status 문자열	조건의 상태입니다.
type 문자열	조건의 유형입니다.

.status.endpoints

QueueManagerStatusEndpoint는 QueueManager에 대한 엔드포인트를 정의합니다.

다음과 같이 표시됩니다.

- [135 페이지의 『.status』](#)

필드	설명
name 문자열	엔드포인트의 이름입니다.
type 문자열	엔드포인트의 유형입니다(예: UI 엔드포인트의 경우 'UI', API 엔드포인트의 경우 'API', API 문서의 경우 'OpenAPI').
uri 문자열	엔드포인트의 URI입니다.

.status.versions

사용 중인 MQ의 버전 및 IBM Entitled Registry에서 사용 가능한 기타 버전입니다.

다음과 같이 표시됩니다.

- [135 페이지의 『.status』](#)

필드	설명
available QueueManagerStatusVersionAvailable	IBM Entitled Registry에서 사용 가능한 MQ의 다른 버전입니다.
reconciled 문자열	사용 중인 IBM MQ의 특정 버전입니다. 사용자 정의 이미지가 지정되면, 실제로 사용 중인 MQ의 버전과 일치하지 않을 수도 있습니다.

.status.versions.available

IBM Entitled Registry에서 사용 가능한 MQ의 다른 버전입니다.

다음과 같이 표시됩니다.

- [136 페이지의 『.status.versions』](#)

필드	설명
channels 배열	MQ 버전을 자동으로 업데이트하는 데 사용할 수 있는 채널입니다.

필드	설명
versions Versions 배열	사용할 수 있는 MQ의 특정 버전입니다.

.status.versions.available.versions

QueueManagerStatusVersion은 MQ의 버전을 정의합니다.

다음과 같이 표시됩니다.

- [136 페이지](#)의 『.status.versions.available』

필드	설명
name 문자열	QueueManager의 이 버전에 대한 버전 이름을 표시합니다. spec.version 필드에 올바른 값입니다.

QueueManager의 상태(mq.ibm.com/v1beta1)

status.conditions 필드는 QueueManager 자원의 상태를 나타내기 위해 업데이트됩니다. 일반적으로 상태는 비정상 상황을 설명합니다. 정상적인 준비 상태의 큐 관리자에는 **Error** 또는 **Pending** 상태가 없습니다. 권고성 **Warning** 상태는 있을 수 있습니다.

상태에 대한 지원은 IBM MQ Operator 1.2에서 도입되었습니다.

QueueManager 자원에 대해서는 다음 상태가 정의되어 있습니다.

표 1. 큐 관리자 상태			
컴포넌트	상태 유형	이유 코드	메시지 경고
QueueManager ⁷	Pending	Creating	MQ 큐 관리자가 배치되는 중입니다.
	Pending	OidcPending	MQ 큐 관리자가 OIDC 클라이언트 등록을 대기하고 있습니다.
	오류	실패함	MQ 큐 관리자가 배치하는데 실패했습니다.
	경고	UnsupportedVersion	⁸ OCP 버전 <ocp_version>에서 지원되지 않는 Operator가 Operand를 설치했습니다. 이 Operand는 지원되지 않습니다.
	경고	EUSSupport	⁹ EUS Operand <mq_version>이(가) 설치되었지만 연장 지원 기간에 대해 적합하지 않은 Operator가 관리하고 있습니다. 이 Operand는 연장 지원 기간에 대해 적합하지 않습니다.
	경고	EUSSupport	¹⁰ EUS Operand <mq_version>이(가) 설치되었지만 OCP 버전 4<ocp_version>은(는) 연장 지원 기간에 대해 적합하지 않습니다. 이 Operand는 연장 지원 기간에 대해 적합하지 않습니다.
	경고	EUSSupport	¹¹ EUS Operand <mq_version>이(가) 설치되었지만 OCP 버전 <ocp_version>은(는) 연장 지원 기간에 대해 적합하지 않습니다. 이 Operand는 정규 CD 릴리스에 따라 지원됩니다.

⁷ Creating 및 Failed 상태는 큐 관리자 배치의 전체 진행 상태를 모니터링합니다. IBM Cloud Pak for Integration 라이선스를 사용 중이며 MQ 웹 콘솔이 사용으로 설정된 경우 OidcPending 상태는 IAM에 OIDC 클라이언트 등록이 완료되기를 대기하는 동안의 큐 관리자 상태를 로그합니다.

⁸ Operator 1.4.0 이상

⁹ Operator 1.4.0 이상

¹⁰ Operator 1.4.0 이상

¹¹ Operator 1.3.0만 해당

컴포넌트	상태 유형	이유 코드	메시지 경고
팟(Pod) ¹²	Pending	PodPending	MQ 큐 관리자의 팟(Pod)이 배치되는 중입니다.
	오류	PodFailed	MQ 큐 관리자의 팟(Pod)이 배치되는 중입니다.
스토리지 ¹³	Pending	StoragePending	MQ 큐 관리자의 스토리지가 프로비저닝되는 중입니다.
	경고	StorageEphemeral	프로덕션 MQ 큐 관리자에 대해 임시 스토리지를 사용하는 중입니다.
	오류	StorageFailed	MQ 큐 관리자의 스토리지를 프로비저닝하는 데 실패했습니다.

Multi 사용자 고유의 IBM MQ 컨테이너 및 배치 코드 빌드

자체 빌드된 컨테이너를 개발합니다. 이 솔루션은 가장 유연한 컨테이너 솔루션이지만, 이 솔루션을 사용하려면 컨테이너를 구성하는 측면에서 뛰어난 기술을 보유하고 있어야 하며 결과 컨테이너를 "소유"하고 있어야 합니다.

시작하기 전에

자체 컨테이너를 개발하기 전에 IBM에서 제공하는 패키징되기 이전의 컨테이너 중 하나를 대신 사용할 수 있는지 여부를 고려하십시오. [컨테이너의 IBM MQ](#)를 참조하십시오.

이 태스크 정보

컨테이너 이미지로 IBM MQ를 패키징할 때, 애플리케이션에 대한 변경사항이 테스트와 스테이징 시스템에 빠르고 쉽게 배치될 수 있습니다. 이는 엔터프라이즈의 Continuous Delivery에 기본적으로 이익이될 수 있습니다.

프로시저

- [139 페이지의 『컨테이너를 사용하여 IBM MQ 큐 관리자 이미지 계획』](#)
- [140 페이지의 『샘플 IBM MQ 큐 관리자 컨테이너 이미지 빌드』](#)
- [142 페이지의 『별도의 컨테이너에서 로컬 바인딩 애플리케이션 실행』](#)

관련 개념

[컨테이너의 IBM MQ](#)

Multi 컨테이너를 사용하여 IBM MQ 큐 관리자 이미지 계획

컨테이너에서 IBM MQ 큐 관리자를 실행할 때 고려할 여러 요구사항이 있습니다. 샘플 컨테이너 이미지에서는 이러한 요구사항을 처리할 방법을 제공하지만, 자체 이미지를 사용하려는 경우 이러한 요구사항이 처리하는 방법에 대해 고려해야 합니다.

¹² Pod 상태는 큐 관리자 배치 중에 팟(Pod) 상태를 모니터링합니다. PodFailed 상태가 표시되는 경우 전체 큐 관리자 상태도 Failed로 설정됩니다.

¹³ Storage 상태는 지속적 스토리지의 볼륨을 작성하는 요청의 진행 상태(StoragePending 상태)를 모니터링하고 바인딩 오류 및 기타 실패를 보고합니다. 스토리지 프로비저닝 중에 오류가 발생하는 경우에는 상태 목록에 StorageFailed가 추가되며 전체 큐 관리자 상태도 Failed로 설정됩니다.

프로세스 감시

컨테이너를 실행할 때, 단일 프로세스(컨테이너 내의 PID 1)를 실행 중이고, 이를 통해 이후에 하위 프로세스를 올바르게 실행시킬 수 있습니다.

기본 프로세스가 종료되면 컨테이너 런타임이 컨테이너를 중지합니다. IBM MQ 큐 관리자는 여러 프로세스가 백그라운드에서 실행 중이어야 합니다.

이 때문에, 큐 관리자가 실행하고 있는 동안에는 기본 프로세스가 활성 상태로 유지하도록 보장해야 합니다. 큐 관리자가 이 프로세스로부터 활성화 상태인지 확인하는 것이 좋습니다(예: 관리 쿼리 수행).

/var/mqm 채우기

컨테이너는 /var/mqm을 볼륨으로 사용하여 구성되어야 합니다.

이를 수행할 때, 볼륨의 디렉토리는 컨테이너가 처음 시작할 때 비어 있습니다. 이 디렉토리는 주로 설치시 채워지지만 설치 및 런타임은 컨테이너 사용시 독립 환경입니다.

이를 해결하기 위해 컨테이너가 시작될 때 `crtmqdir` 명령을 사용하여 처음 실행될 때 /var/mqm 를 채울 수 있습니다.

컨테이너 보안

런타임 보안 요구사항을 최소화하기 위해 샘플 컨테이너 이미지는 IBM MQ 압축 해제 가능한 설치를 사용하여 설치됩니다. `setuid` 비트가 설정되지 않으며 컨테이너는 권한 상승을 사용하지 않아도 됩니다. 일부 컨테이너 시스템에서는 사용자가 사용할 수 있는 사용자 ID를 정의하고 압축 해제 가능한 설치가 사용 가능한 운영 체제 사용자에게 대해 어떤 가정도 하지 않습니다.

Multi 샘플 IBM MQ 큐 관리자 컨테이너 이미지 빌드

이 정보를 사용하여 컨테이너에서 IBM MQ 큐 관리자를 실행하기 위한 샘플 컨테이너 이미지를 빌드하십시오.

이 태스크 정보

먼저, Red Hat Universal 기본 이미지 파일 시스템 및 IBM MQ의 깔끔한 설치를 포함하는 기본 이미지를 빌드하십시오.

두 번째로, 기본 맨 위에 있는 다른 컨테이너 이미지 계층을 빌드하고, 이는 일부 IBM MQ 구성을 기본 사용자 ID 및 비밀번호 보안을 허용하도록 추가됩니다.

마지막으로, 호스트 파일 시스템의 컨테이너 특정 볼륨에서 제공하는 /var/mqm의 콘텐츠가 있는 파일 시스템으로 이 이미지를 사용하여 컨테이너를 실행하십시오.

프로시저

- 컨테이너에서 IBM MQ 큐 관리자 실행을 위한 샘플 컨테이너 이미지를 빌드하는 방법에 대한 정보는 다음 하위 주제를 참조하십시오.
 - [140 페이지의 『샘플 기본 IBM MQ 큐 관리자 이미지 빌드』](#)
 - [141 페이지의 『샘플 구성된 IBM MQ 큐 관리자 이미지 빌드』](#)

Multi 샘플 기본 IBM MQ 큐 관리자 이미지 빌드

컨테이너 이미지에서 IBM MQ를 사용하려면 깔끔한 IBM MQ 설치로 기본 이미지를 초기에 빌드해야 합니다. 다음 단계에는 GitHub에 호스팅된 샘플 코드를 사용하여 샘플 기본 이미지를 빌드하는 방법에 대해 표시됩니다.

프로시저

- [mq-container GitHub 저장소](#)에서 제공되는 make 파일을 사용하여 프로덕션 컨테이너 이미지를 빌드하십시오.

GitHub에 있는 [Building a container image](#)의 지시사항을 따르십시오. Red Hat OpenShift Container Platform "restricted" Security Context Constraint (SCC) 를 사용하여 보안 액세스를 구성하려는 경우 'No-Install' IBM MQ 패키지를 사용해야 합니다.

결과

IBM MQ가 있는 기본 컨테이너 이미지가 설치되어야 합니다.

이제 [샘플 구성 IBM MQ 큐 관리자 이미지를 빌드할 준비가 되었습니다.](#)

Multi 샘플 구성된 IBM MQ 큐 관리자 이미지 빌드

일반 기본 IBM MQ 컨테이너 이미지를 빌드하고 나면 보안 액세스를 허용하도록 구성을 적용해야 합니다. 이를 수행하려면 일반 이미지를 상위로 사용하여 컨테이너 이미지 계층을 작성하십시오.

시작하기 전에

V 9.2.0 이 태스크에서는 샘플 기본 IBM MQ 큐 관리자 이미지를 빌드할 때 "비설치" IBM MQ 패키지를 사용했다고 가정합니다. 그렇지 않은 경우에는 Red Hat OpenShift Container Platform "restricted" SCC(Security Context Constraint)를 사용하여 보안 액세스를 구성할 수 없습니다. 기본적으로 사용되는 "restricted" SCC는 무작위 사용자 ID를 사용하며 다른 사용자로의 변경을 통한 권한 상승을 금지합니다. IBM MQ의 기존 RPM 기반 설치 프로그램은 mqm 사용자 및 그룹에 의존하며, 실행 가능 프로그램의 `setuid` 비트도 사용합니다. IBM MQ 9.2에서는 "비설치" IBM MQ 패키지를 사용하면 mqm 사용자 또는 mqm 그룹이 생기지 않습니다.

프로시저

1. 새 디렉토리를 작성하고, 다음 콘텐츠를 갖는 `config.mqsc`라는 파일을 추가하십시오.

```
DEFINE QLOCAL(EXAMPLE.QUEUE.1) REPLACE
```

이전 예제는 단순 사용자 ID 및 비밀번호 인증을 사용한다는 사실에 유의하십시오. 그러나 엔터프라이즈가 요구하는 모든 보안 구성을 적용할 수 있습니다.

2. 다음 콘텐츠를 갖는 `Dockerfile`이라는 파일을 작성하십시오.

```
FROM mq
COPY config.mqsc /etc/mqm/
```

3. 다음 명령을 사용하여 사용자 정의 컨테이너 이미지를 빌드하십시오.

```
docker build -t mymq .
```

여기서 "."은 작성한 두 개의 파일이 포함된 디렉토리입니다.

그런 다음, Docker는 해당 이미지를 사용하여 임시 컨테이너를 작성하고 남아 있는 명령을 실행합니다.

참고: Red Hat Enterprise Linux(RHEL)에서 **docker**(RHEL V7) 또는 **podman**(RHEL V7 또는 RHEL V8) 명령을 사용하십시오. Linux에서, 명령의 시작 부분에서 **sudo**로 **docker** 명령을 실행하여 추가 특권을 얻어야 합니다.

4. 새로 사용자 정의된 이미지를 실행하여 방금 작성한 디스크 이미지를 갖는 새 컨테이너를 작성하십시오.

새 이미지 계층은 실행할 특정 명령을 지정하지 않으므로, 상위 이미지에서 상속됩니다. 상위의 시작점(코드 는 GitHub에서 사용 가능):

- 큐 관리자 작성
- 큐 관리자 시작
- 기본 리스너 작성
- 그런 다음 `/etc/mqm/config.mqsc`에서 임의의 MQSC 명령 실행

다음 명령을 실행하여 새로 사용자 정의된 이미지를 실행하십시오.

```
docker run \
  --env LICENSE=accept \
  --env MQ_QMGR_NAME=QM1 \
  --volume /var/example:/var/mqm \
  --publish 1414:1414 \
  --detach \
  mymq
```

여기서,

첫 번째 env 매개변수

환경 변수를 컨테이너로 전달하는데, 이것은 IBM IBM WebSphere® MQ에 대한 라이선스의 허용을 수신 확인합니다. 또한 해당 라이선스를 보기 위해 LICENSE 변수를 view로 설정할 수도 있습니다.

IBM MQ 라이선스에 대한 자세한 정보는 [IBM MQ 라이선스 정보](#) 를 참조하십시오.

두 번째 env 매개변수

사용 중인 큐 관리자 이름을 설정합니다.

Volume 매개변수

MQ가 /var/mqm에 기록하는 모든 내용이 실제로 호스트의 /var/example에 기록되어야 함을 컨테이너에 알립니다.

이 옵션은 나중에 컨테이너를 쉽게 삭제하고 모든 지속 데이터를 계속 보존할 수 있다는 것을 의미합니다. 이 옵션은 또한 로그 파일을 보기가 더 쉽게 합니다.

Publish 매개변수

호스트 시스템의 포트를 컨테이너의 포트에 매핑합니다. 컨테이너는 기본적으로 자체 내부 IP 주소를 갖고 실행하며, 이것은 노출시키려는 임의의 포트를 특정하게 매핑해야 함을 의미합니다.

이 예에서는 호스트의 포트 1414를 컨테이너의 포트 1414에 매핑함을 의미합니다.

Detach 매개변수

컨테이너를 백그라운드에서 실행합니다.

결과

구성된 컨테이너 이미지를 빌드하고 **docker ps** 명령을 사용하여 실행 중인 컨테이너를 볼 수 있습니다.

docker top 명령을 사용하여 컨테이너에서 실행 중인 IBM MQ 프로세스를 볼 수 있습니다.



주의:

docker logs \${CONTAINER_ID} 명령을 사용하여 컨테이너의 로그를 볼 수 있습니다.

다음에 수행할 작업

- 컨테이너가 표시되지 않은 경우 **docker ps** 명령을 사용하면 컨테이너가 실패할 수 있습니다. **docker ps -a** 명령을 사용하여 실패한 컨테이너를 볼 수 있습니다.
- **docker ps -a** 명령을 사용하면 컨테이너 ID가 표시됩니다. 이 ID는 **docker run** 명령을 실행할 때도 출력됩니다.
- **docker logs \${CONTAINER_ID}** 명령을 사용하여 컨테이너의 로그를 볼 수 있습니다.

Multi

별도의 컨테이너에서 로컬 바인딩 애플리케이션 실행

Docker의 컨테이너 간에 프로세스 네임스페이스 공유를 사용하면 IBM MQ 큐 관리자의 별도 컨테이너에서 IBM MQ에 로컬 바인딩이 필요한 애플리케이션을 실행할 수 있습니다.

이 태스크 정보

이 기능은 IBM MQ 9.0.3 이상 큐 관리자에서 지원됩니다.

다음 제한사항을 준수해야 합니다.

- --pid 인수를 사용하여 컨테이너 PID 네임스페이스를 공유해야 합니다.
- --ipc 인수를 사용하여 컨테이너 IPC 네임스페이스를 공유해야 합니다.
- 다음 중 하나를 수행해야 합니다.
 1. --uts 인수를 사용하여 컨테이너 UTS 네임스페이스를 호스트와 공유하십시오.
 2. -h 또는 --hostname 인수를 사용하여 컨테이너가 동일한 호스트 이름을 사용하는지 확인하십시오.
- /var/mqm 디렉토리 아래의 모든 컨테이너에서 사용할 수 있는 볼륨에 IBM MQ 데이터 디렉토리를 마운트해야 합니다.

이미 Docker 가 설치된 Linux 시스템에서 다음 단계를 완료하여 이 기능을 사용할 수 있습니다.

다음 예제는 샘플 IBM MQ 컨테이너 이미지를 사용합니다. [Github](#)에서 이 이미지에 대한 세부사항을 찾을 수 있습니다.

프로시저

1. 다음 명령을 발행하여 볼륨으로 사용할 임시 디렉토리를 작성하십시오.

```
mkdir /tmp/dockerVolume
```

2. 다음 명령을 발행하여 sharedNamespace라는 이름으로 컨테이너에 큐 관리자(QM1)를 작성하십시오.

```
docker run -d -e LICENSE=accept -e MQ_QMGR_NAME=QM1 --volume /tmp/dockerVol:/mnt/mqm --uts host --name sharedNamespace ibmcom/mq
```

3. secondaryContainer라는 두 번째 컨테이너를 시작하고, ibmcom/mq를 기반으로 하지만 다음 명령을 실행하여 큐 관리자를 작성하지 마십시오.

```
docker run --entrypoint /bin/bash --volumes-from sharedNamespace --pid container:sharedNamespace --ipc container:sharedNamespace --uts host --name secondaryContainer -it --detach ibmcom/mq
```

4. 두 번째 컨테이너에서 **dspmqs** 명령을 다음과 같이 실행하여 두 큐 관리자의 상태를 확인하십시오.

```
docker exec secondaryContainer dspmqs
```

5. 다음 명령을 실행하여 다른 컨테이너에서 실행되는 큐 관리자에 대해 MQSC 명령을 처리하십시오.

```
docker exec -it secondaryContainer runmqsc QM1
```

결과

이제 별도의 컨테이너에서 로컬 애플리케이션이 실행 중이며 이제 두 번째 컨테이너에서 **dspmqs**, **amqsput**, **amqsget** 및 **runmqsc**와 같은 명령을 QM1 큐 관리자에 대한 로컬 바인딩으로 성공적으로 실행할 수 있습니다.

예상한 결과를 얻지 못한 경우 자세한 정보는 [143 페이지의 『네임스페이스 애플리케이션 문제점 해결』](#)의 내용을 참조하십시오.

Multi 네임스페이스 애플리케이션 문제점 해결

공유 네임스페이스를 사용할 경우 모든 네임스페이스(IPC, PID, UTS/호스트 이름)와 마운트된 볼륨을 공유해야 합니다. 그렇지 않으면 애플리케이션이 작동하지 않습니다.

준수해야 하는 제한사항 목록은 [142 페이지의 『별도의 컨테이너에서 로컬 바인딩 애플리케이션 실행』](#)의 내용을 참조하십시오.

애플리케이션이 나열된 제한사항을 모두 충족하지 못할 경우, 컨테이너는 시작되지만 예상 기능이 작동하지 않는 문제점이 발견될 수 있습니다.

다음 목록에는 몇 가지 공통 원인 및 제한사항 중 하나가 충족되지 않았는지 확인할 수 있는 동작이 설명되어 있습니다.

- 네임스페이스 (UTS/PID/IPC) 또는 컨테이너의 호스트 이름을 공유하는 것을 잊고 볼륨을 마운트하는 경우, 컨테이너는 큐 관리자를 볼 수 있지만 큐 매니저와는 상호 작용하지 않습니다.
 - **dspmq** 명령의 경우 다음이 표시됩니다.

```
docker exec container dspmq
QMNAME(QM1)                STATUS(Status not available)
```

- **runmqsc** 명령 또는 큐 관리자에 연결하려고 시도하는 다른 명령의 경우 AMQ8146 오류 메시지가 수신될 수 있습니다.

```
docker exec -it container runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2024.
Starting MQSC for queue manager QM1.
AMQ8146: IBM MQ queue manager not available
```

- 필요한 모든 네임스페이스를 공유하지만 공유 볼륨을 /var/mqm 디렉토리에 마운트하지 않은 경우 유효한 IBM MQ 데이터 경로가 있으면 해당 명령이 AMQ8146 오류 메시지도 수신합니다.

그러나 **dspmq**는 큐 관리자를 표시할 수 없어 빈 응답을 대신 리턴합니다.

```
docker exec container dspmq
```

- 필요한 모든 네임스페이스를 공유하지만 공유 볼륨을 /var/mqm 디렉토리에 마운트하지 않았으며 유효한 IBM MQ 데이터 경로가 없는 경우(또는 IBM MQ 데이터 경로가 없음) 데이터 경로는 IBM MQ 설치의 핵심 컴포넌트이므로 다양한 오류가 표시됩니다. 데이터 경로가 없으면 IBM MQ를 작동할 수 없습니다.

다음 명령을 실행하고 이러한 예제에 표시되는 것과 유사한 응답을 참조하는 경우, 디렉토리를 마운트하거나 IBM MQ 데이터 디렉토리를 작성했는지 확인해야 합니다.

```
docker exec container dspmq
'No such file or directory' from /var/mqm/mqs.ini
AMQ6090: IBM MQ was unable to display an error message FFFFFFFF.
AMQffff

docker exec container dspmqver
AMQ7047: An unexpected error was encountered by a command. Reason code is 0.

docker exec container mqrc
<file path>/mqrc.c[1152]
lpi0btainQMDetails --> 545261715

docker exec container crtmqm QM1
AMQ8101: IBM MQ error (893) has occurred.

docker exec container strmqm QM1
AMQ6239: Permission denied attempting to access filesystem location '/var/mqm'.
AMQ7002: An error occurred manipulating a file.

docker exec container endmqm QM1
AMQ8101: IBM MQ error (893) has occurred.

docker exec container dlmqm QM1
AMQ7002: An error occurred manipulating a file.

docker exec container strmqweb
<file path>/mqrc.c[1152]
lpi0btainQMDetails --> 545261715
```

CP4I 자체 컨테이너 작성 시 원시 HA 그룹 작성

기본 HA 그룹을 작성하기 위해 세 개의 큐 관리자를 작성, 구성 및 시작해야 합니다.

이 태스크 정보

원시 HA 솔루션을 작성하는 데 권장되는 방법은 IBM MQ 연산자를 사용하는 것입니다 ([원시 HA](#) 참조). 또는 사용자 고유의 컨테이너를 작성하는 경우 다음 지시사항을 따를 수 있습니다.

원시 HA 그룹을 작성하려면 로그 유형이 `log replication`로 설정된 세 개의 노드에 세 개의 큐 관리자를 작성합니다. 그런 다음 각 큐 관리자의 `qm.ini` 파일을 편집하여 세 노드 각각에 대한 연결 세부사항을 추가하여 로그 데이터를 서로 복제할 수 있도록 합니다.

세 개의 모든 인스턴스가 서로 통신할 수 있는지 확인할 수 있도록 세 개의 큐 관리자를 모두 시작하고 활성 인스턴스와 복제본이 될 인스턴스를 판별해야 합니다.

프로시저

1. 세 개의 노드 각각에서 큐 관리자를 작성하고 로그 복제본의 로그 유형을 지정하며 각 로그 인스턴스에 고유한 이름을 제공하십시오. 각 큐 관리자의 이름은 동일합니다.

```
crtmqm -lr instance_name qmname
```

예를 들면, 다음과 같습니다.

```
node 1> crtmqm -lr qm1_inst1 qm1
node 2> crtmqm -lr qm1_inst2 qm1
node 3> crtmqm -lr qm1_inst3 qm1
```

2. 각 큐 관리자의 작성에 성공하면 `NativeHALocalInstance` 라는 추가 스탠자가 큐 관리자 구성 파일 `qm.ini`에 추가됩니다. 제공된 인스턴스 이름을 지정하는 `Name` 속성이 스탠자에 추가됩니다.

선택적으로 다음 속성을 `qm.ini` 파일의 `NativeHALocalInstance` 스탠자에 추가할 수 있습니다.

KeyRepository

로그 복제 트래픽의 보호에 사용할 디지털 인증서를 보유하는 키 저장소의 위치입니다. 위치는 스탠 형식으로 제공됩니다. 즉, 확장자가 없는 전체 경로 및 파일 이름이 포함됩니다. `KeyRepository` 스탠자 속성을 생략하면 일반 텍스트의 인스턴스 간에 로그 복제 데이터가 교환됩니다.

CertificateLabel

로그 복제 트래픽의 보호에 사용할 디지털 인증서를 식별하는 인증서 레이블입니다. `KeyRepository` 가 제공되지만 `CertificateLabel` 가 생략되면 기본값 `ibmwebspheremqqueue_manager` 가 사용됩니다.

CipherSpec

로그 복제 트래픽을 보호하는 데 사용할 MQ CipherSpec입니다. 이 스탠자 속성이 제공되면 `KeyRepository` 도 제공해야 합니다. `KeyRepository` 가 제공되지만 `CipherSpec` 가 생략되면 기본값 `ANY` 가 사용됩니다.

LocalAddress

로그 복제 트래픽을 허용하는 로컬 네트워크 인터페이스 주소입니다. 이 스탠자 속성이 제공되는 경우 "[addr] [(port)]" 형식을 사용하여 로컬 네트워크 인터페이스 및/또는 포트를 식별합니다. 네트워크 주소는 호스트 이름, IPv4 점분리 십진수 또는 IPv6 16진수형식으로 지정할 수 있습니다. 이 속성을 생략하면 큐 관리자가 모든 네트워크 인터페이스에 바인드하려고 시도하며 로컬 인스턴스 이름과 일치하는 `NativeHAInstances` 스탠자의 `ReplicationAddress` 에 지정된 포트를 사용합니다.

HeartbeatInterval

하트비트 간격은 고유 HA 큐 관리자의 활성 인스턴스가 네트워크 하트비트를 전송하는 빈도(밀리세컨드)를 정의합니다. 유효한 하트비트 간격 값의 범위는 500(0.5초)에서 60000(1분)까지이며, 값이 이 범위를 벗어나면 큐 관리자를 시작하는 데 실패합니다. 이 속성이 생략되는 경우 기본값 5000(5초)가 사용됩니다. 각 인스턴스는 동일한 하트비트 간격을 사용해야 합니다.

HeartbeatTimeout

하트비트 제한시간은 활성 인스턴스가 반응하지 않음을 결정하기 전에 고유 HA 큐 관리자의 복제본 인스턴스가 대기하는 기간을 정의합니다. 유효한 하트비트 간격 제한시간 값의 범위는 500(0.5초)에서 120000(2분)까지입니다. 하트비트 제한시간 값은 하트비트 간격 이상이어야 합니다.

값이 올바르지 않으면 큐 관리자를 시작하는 데 실패합니다. 이 속성을 생략하면 새 활성 인스턴스를 선택하기 위한 프로세스를 시작하기 전에 2 x `HeartbeatInterval`의 복제본 대기가 대기합니다. 각 인스턴스는 동일한 하트비트 제한시간을 사용해야 합니다.

RetryInterval

재시도 간격은 고유 HA 큐 관리자가 실패한 복제 링크를 재시도해야 하는 빈도(밀리초)를 정의합니다. 올바른 재시도 간격 범위는 500(0.5초)에서 120000(2분)까지입니다. 이 속성을 생략하면 실패한 복제 링크를 재시도하기 전에 2 x HeartbeatInterval의 복제본 대기가 대기합니다.

3. 각 큐 관리자의 `qm.ini` 파일을 편집하고 연결 세부사항을 추가하십시오. 기본 HA 그룹 (로컬 인스턴스 포함)의 각 큐 관리자 인스턴스에 대해 하나씩 세 개의 `NativeHAInstance` 스탠자를 추가합니다. 다음 속성을 추가한다.

이름

큐 관리자 인스턴스를 작성할 때 사용한 인스턴스 이름을 지정하십시오.

ReplicationAddress

인스턴스의 호스트 이름, IPv4 점분리 십진수 또는 IPv6 16진수형식 주소를 지정하십시오. 주소를 호스트 이름, IPv4 점분리 십진수 또는 IPv6 16진수형식 주소로 지정할 수 있습니다. 복제 주소는 그룹의 각 인스턴스에서 분석 가능하고 라우팅 가능해야 합니다. 로그 복제에 사용할 포트 번호는 대괄호로 지정해야 합니다. 예를 들면 다음과 같습니다.

```
ReplicationAddress=host1.example.com(4444)
```

참고: `NativeHAInstance` 스탠자는 모든 인스턴스에서 동일하며 자동 구성 (`crtmqm -ii`) 을 사용하여 제공될 수 있습니다.

4. 세 개의 인스턴스 각각을 시작한다.

```
strmqm QMgrName
```

인스턴스가 시작되면 통신하여 세 개의 인스턴스가 모두 실행 중인지 확인한 다음 세 개 중 활성 인스턴스를 결정하는 반면 다른 두 인스턴스는 복제본으로 계속 실행됩니다.

예

다음 예제는 세 개의 인스턴스 중 하나에 필요한 기본 HA 세부사항을 지정하는 `qm.ini` 파일의 섹션을 표시합니다.

```
NativeHALocalInstance:
  LocalName=node-1

NativeHAInstance:
  Name=node-1
  ReplicationAddress=host1.example.com(4444)
NativeHAInstance:
  Name=node-2
  ReplicationAddress=host2.example.com(4444)
NativeHAInstance:
  Name=node-3
  ReplicationAddress=host3.example.com(4444)
```

Kubernetes 고유한 HA 큐 관리자의 롤링 업데이트를 수행하기 위한 고려사항

원시 HA 큐 관리자의 IBM MQ 버전 또는 팟(Pod) 스펙에 대한 업데이트는 큐 관리자 인스턴스의 롤링 업데이트를 수행해야 합니다. IBM MQ Operator이(가) 자동으로 이를 처리하지만, 사용자 고유의 배치 코드를 빌드하는 경우에는 몇 가지 중요한 고려사항이 있습니다.

참고: 샘플 Helm 도표에는 롤링 업데이트를 수행하는 셸 스크립트가 포함되어 있지만 이 주제의 고려사항을 해결하지 않으므로 스크립트는 프로덕션 사용에 적합한 **되지** 않습니다.

Kubernetes에서 `StatefulSet` 자원은 정렬된 시작 및 롤링 업데이트를 관리하는 데 사용됩니다. 시작 프로시저의 일부는 각 팟(Pod)을 개별적으로 시작하고, 준비가 되기를 기다린 후, 다음 팟(Pod)으로 이동하는 것입니다. 이는 모든 팟(Pod)이 리더 선거를 실행할 수 있도록 시작되어야 하므로 고유 HA에는 적용되지 않습니다. 따라서 `.spec.podManagementPolicy`의 `StatefulSet` 필드를 `Parallel`(으)로 설정해야 합니다. 이는 또한 모든 팟(Pod)이 동시에 업데이트될 것이라는 것을 의미하며, 이는 특히 바람직하지 않습니다. 이러한 이유로 `StatefulSet`은(는) `OnDelete` 업데이트 전략도 사용해야 합니다.

StatefulSet 롤링 업데이트 코드를 사용하지 못하면 다음 사항을 고려해야 하는 사용자 정의 롤링 업데이트 코드가 필요합니다.

- 일반적인 롤링 업데이트 프로시저
- 최상의 순서로 팟(Pod)을 업데이트하여 시간 최소화
- 클러스터 상태의 변경 처리
- 오류 핸들링
- 타이밍 문제 처리하기

일반적인 롤링 업데이트 프로시저

롤링 업데이트 코드는 각 인스턴스가 REPLICAs의 dspmq 상태를 표시할 때까지 대기해야 합니다. 이는 인스턴스가 일부 시작 레벨을 수행했음을 의미합니다(예를 들어, 컨테이너가 시작되고 MQ 프로세스가 실행 중임). 그러나 다른 인스턴스와 아직 대화할 필요는 없습니다. 예를 들어, 팟(Pod) A가 다시 시작되고 REPLICAs 상태가 되자마자 팟(Pod) B가 다시 시작됩니다. 팟(Pod) B가 새로운 구성으로 시작되면 팟(Pod) A와 대화할 수 있어야 하고 쿼럼을 형성할 수 있으며 A 또는 B가 새 활성 인스턴스가 됩니다.

이에 따라 각 팟(Pod)이 REPLICAs 상태에 도달한 후 피어에 연결하여 쿼럼을 설정할 수 있도록 하는 지연 시간을 갖는 것이 유용합니다.

최상의 순서로 팟(Pod)을 업데이트하여 시간 최소화

롤링 업데이트 코드는 알려진 오류 상태에 있는 팟(Pod)을 시작으로, 성공적으로 시작되지 않은 팟(Pod)을 시작하여 한 번에 하나씩 팟(Pod)을 삭제해야 합니다. 활성 큐 관리자 팟(Pod)은 일반적으로 마지막으로 업데이트되어야 합니다.

마지막 업데이트가 알려진 오류 상태가 되는 경우 팟(Pod)의 삭제를 일시정지하는 것도 중요합니다. 이렇게 하면 모든 팟(Pod)에서 업데이트된 업데이트의 롤아웃을 방지할 수 있습니다. 예를 들어, 이는 팟(Pod)이 액세스가 가능하지 않은(또는 오타를 포함하는) 새로운 컨테이너 이미지를 사용하도록 업데이트되는 경우에 발생할 수 있다.

클러스터 상태의 변경 처리

롤링 업데이트 코드는 클러스터 상태의 실시간 변경사항에 적절히 반응해야 합니다. 예를 들어, 큐 관리자의 팟(Pod) 중 하나가 노드 재부팅으로 인해 또는 노드 압력으로 인해 축출될 수 있습니다. 클러스터가 사용 중인 경우 축출된 팟(Pod)이 즉시 다시 스케줄되지 않을 수 있습니다. 이 경우, 롤링 업데이트 코드는 다른 팟(Pod)을 다시 시작하기 전에 적절히 대기해야 합니다.

오류 핸들링

롤링 업데이트 코드는 Kubernetes API 및 기타 예상치 못한 클러스터 동작을 호출할 때 실패할 수 있습니다.

또한 롤링 업데이트 코드 자체는 다시 시작할 수 있도록 허용해야 합니다. 롤링 업데이트는 장기 실행될 수 있으며 코드를 다시 시작해야 할 수 있습니다.

타이밍 문제 처리하기

롤링 업데이트 코드는 팟(Pod)의 업데이트 개정판을 확인하여 팟(Pod)이 다시 시작되었는지 확인해야 합니다. 이것은 팟(Pod)이 "시작됨" 이라는 것을 나타낼 수 있지만, 아직 종료되지 않았음을 나타내는 타이밍 문제들을 방지합니다.

관련 개념

5 페이지의 『컨테이너에서 IBM MQ를 사용하는 방법 선택』

컨테이너에서 IBM MQ를 사용하기 위한 여러 옵션이 있습니다. 패키징되기 이전의 컨테이너 이미지를 사용하는 IBM MQ Operator를 사용하도록 선택하거나 자체 이미지 및 배치 코드를 빌드할 수 있습니다.

CP4I 사용자 정의 빌드 컨테이너에 대한 원시 HA 큐 관리자의 상태 보기

사용자 정의 빌드 컨테이너의 경우 dspmq 명령을 사용하여 원시 HA 인스턴스의 상태를 볼 수 있습니다.

이 태스크 정보

dspmqr 명령을 사용하여 노드에서 큐 관리자 인스턴스의 운영 상태를 볼 수 있습니다. 리턴되는 정보는 인스턴스가 활성화인지 또는 복제본인지에 따라 달라집니다. 활성화 인스턴스에서 제공하는 정보는 결정적이며 복제본 노드의 정보는 이전 정보일 수 있습니다.

다음 조치를 수행할 수 있습니다.

- 현재 노드의 큐 관리자 인스턴스가 활성화인지 또는 복제본인지 여부를 확인합니다.
- 현재 노드에서 인스턴스의 고유 HA 운영 상태를 확인합니다.
- 고유 HA 구성에서 세 개 인스턴스 모두의 운영 상태를 확인합니다.

다음 상태 필드는 고유 HA 구성 상태를 보고하는 데 사용됩니다.

ROLE

인스턴스의 현재 역할을 지정하며 Active, Replica 또는 Unknown 중 하나입니다.

INSTANCE

crtmqm 명령의 **-lr** 옵션을 사용하여 작성된 경우 큐 관리자의 이 인스턴스에 대해 제공된 이름입니다.

INSYNC

인스턴스가 필요한 경우 활성화 인스턴스로서 인계받을 수 있는지 여부를 표시합니다.

QUORUM

number_of_instances_in-sync/number_of_instances_configured 양식으로 쿼럼 상태를 보고합니다.

REPLADDR

큐 관리자 인스턴스의 복제 이름.

CONNACTV

노드가 활성화 인스턴스에 연결되어 있는지 여부를 표시합니다.

BACKLOG

인스턴스가 뒤에 있는 KB 수를 표시합니다.

CONNINST

이름 지정된 인스턴스가 이 인스턴스에 연결되는지 여부를 표시합니다.

ALTDAT

이 정보가 마지막으로 업데이트된 날짜를 표시합니다(업데이트된 적이 없으면 공백).

ALTTIME

이 정보가 마지막으로 업데이트된 시간을 표시합니다(업데이트된 적이 없으면 공백).

프로시저

- 큐 관리자 인스턴스가 활성화 인스턴스로 실행 중인지 또는 복제본으로 실행 중인지를 판별합니다.

```
dspmqr -o status -m QMgrName
```

큐 관리자 BOB의 활성화 인스턴스는 다음 상태를 보고합니다.

```
QMNAME(BOB)          STATUS(Running)
```

큐 관리자 BOB의 복제본 인스턴스는 다음 상태를 보고합니다.

```
QMNAME(BOB)          STATUS(Replica)
```

비활성 인스턴스는 다음 상태를 보고합니다.

```
QMNAME(BOB)          STATUS(Ended Immediately)
```

- 현재 노드에서 인스턴스의 원시 HA 작동 상태를 판별하려면 다음을 수행하십시오.

```
dspmqr -o nativeha -m QMgrName
```

큐 관리자 BOB의 활성화 인스턴스는 다음 상태를 보고합니다.

```
QMNAME(BOB)          ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
```

큐 관리자 BOB의 복제본 인스턴스는 다음 상태를 보고합니다.

```
QMNAME(BOB)          ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
```

큐 관리자 BOB의 비활성 인스턴스는 다음 상태를 보고합니다.

```
QMNAME(BOB)          ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
```

- 고유 HA 구성에서 모든 인스턴스의 고유 HA 운영 상태를 판별합니다.

```
dspmqr -o nativeha -x -m QMgrName
```

큐 관리자 BOB의 활성 인스턴스를 실행 중인 노드에서 이 명령을 실행하는 경우 다음 상태를 수신합니다.

```
QMNAME(BOB)          ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

큐 관리자 BOB의 복제본 인스턴스를 실행 중인 노드에서 이 명령을 실행하면, 다음 상태를 수신할 수 있습니다. 이는 복제본 중 하나가 뒤쳐지고 있음을 표시합니다.

```
QMNAME(BOB)          ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(No) BACKLOG(435)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

큐 관리자 BOB의 비활성 인스턴스를 실행 중인 노드에서 이 명령을 실행하면 다음 상태를 수신할 수 있습니다.

```
QMNAME(BOB)          ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
INSTANCE(inst1) ROLE(Unknown) REPLADDR(9.20.123.45) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
INSTANCE(inst2) ROLE(Unknown) REPLADDR(9.20.123.46) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
INSTANCE(inst3) ROLE(Unknown) REPLADDR(9.20.123.47) CONNACTV(No) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
```

인스턴스가 계속 무엇이 활성화인지 그리고 무엇이 복제본인지 조정하는 경우 명령을 실행하면, 다음 상태를 수신할 수 있습니다.

```
QMNAME(BOB)          STATUS(Negotiating)
```

관련 참조

[dspmqr](#)

CP4I 원시 HA 큐 관리자 종료

`endmqm` 명령을 사용하여 원시 HA 그룹의 일부인 활성 또는 복제본 큐 관리자를 종료할 수 있습니다.

프로시저

- 큐 관리자의 활성 인스턴스를 종료하려면 이 문서의 구성 섹션에서 [원시 HA 큐 관리자 종료](#) 를 참조하십시오.

주의사항

이 정보는 미국에서 제공되는 제품 및 서비스용으로 작성된 것입니다.

IBM은 다른 국가에서 이 책에 기술된 제품, 서비스 또는 기능을 제공하지 않을 수도 있습니다. 현재 사용할 수 있는 제품 및 서비스에 대한 정보는 한국 IBM 담당자에게 문의하십시오. 이 책에서 IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산을 침해하지 않는 한, 기능상으로 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수도 있습니다. 그러나 비IBM 제품, 프로그램 또는 서비스의 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대해 특허를 보유하고 있거나 현재 특허 출원 중일 수 있습니다. 이 책을 제공한다고 해서 특허에 대한 라이선스까지 부여하는 것은 아닙니다. 라이선스에 대한 의문사항은 다음으로 문의하십시오.

150-945
서울특별시 영등포구
국제금융로 10, 3IFC
한국 아이.비.엠 주식회사
U.S.A.

2바이트(DBCS) 정보에 관한 라이선스 문의는 한국 IBM에 문의하거나 다음 주소로 서면 문의하시기 바랍니다.

지적 재산권 라이선스 부여
2-31 Roppongi 3-chome, Minato-Ku
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

다음 단락은 현지법과 상충하는 영국이나 기타 국가에서는 적용되지 않습니다. IBM은 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여(단, 이에 한하지 않음) 명시적 또는 묵시적인 일체의 보증 없이 이 책을 "현상태대로" 제공합니다. 일부 국가에서는 특정 거래에서 명시적 또는 묵시적 보증의 면책사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 정보는 주기적으로 변경되며, 변경된 사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및/또는 프로그램을 사전 통지 없이 언제든지 개선 및/또는 변경할 수 있습니다.

이 정보에서 언급되는 비IBM의 웹 사이트는 단지 편의상 제공된 것으로, 어떤 방식으로든 이들 웹 사이트를 옹호하고자 하는 것은 아닙니다. 해당 웹 사이트의 자료는 본 IBM 제품 자료의 일부가 아니므로 해당 웹 사이트 사용으로 인한 위험은 사용자 본인이 감수해야 합니다.

IBM은 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 귀하가 제공한 정보를 사용하거나 배포할 수 있습니다.

(i) 독립적으로 작성된 프로그램과 기타 프로그램(본 프로그램 포함) 간의 정보 교환 및 (ii) 교환된 정보의 상호 이용을 목적으로 본 프로그램에 관한 정보를 얻고자 하는 라이선스 사용자는 다음 주소로 문의하십시오.

서울특별시 영등포구
서울특별시 강남구 도곡동 467-12,
군인공제회관빌딩
한국 아이.비.엠 주식회사
U.S.A.

이러한 정보는 해당 조건(예를 들면, 사용료 지불 등)하에서 사용될 수 있습니다.

이 정보에 기술된 라이선스가 부여된 프로그램 및 프로그램에 대해 사용 가능한 모든 라이선스가 부여된 자료는 IBM이 IBM 기본 계약, IBM 프로그램 라이선스 계약(IPLA) 또는 이와 동등한 계약에 따라 제공한 것입니다.

본 문서에 포함된 모든 성능 데이터는 제한된 환경에서 산출된 것입니다. 따라서 다른 운영 환경에서 얻어진 결과는 상당히 다를 수 있습니다. 일부 성능은 개발 단계의 시스템에서 측정되었을 수 있으므로 이러한 측정치가 일반적으로 사용되고 있는 시스템에서도 동일하게 나타날 것이라고는 보증할 수 없습니다. 또한 일부 성능은 추정

통해 추측되었을 수도 있으므로 실제 결과는 다를 수 있습니다. 이 책의 사용자는 해당 데이터를 본인의 특정 환경에서 검증해야 합니다.

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개 자료 또는 기타 범용 소스로부터 얻은 것입니다. IBM에서는 이러한 제품들을 테스트하지 않았으므로, 비IBM 제품과 관련된 성능의 정확성, 호환성 또는 기타 청구에 대해서는 확신할 수 없습니다. 비IBM 제품의 성능에 대한 의문사항은 해당 제품의 공급업체에 문의하십시오.

IBM이 제시하는 방향 또는 의도에 관한 모든 언급은 특별한 통지 없이 변경될 수 있습니다.

이 정보에는 일상의 비즈니스 운영에서 사용되는 자료 및 보고서에 대한 예제가 들어 있습니다. 이들 예제에는 개념을 가능한 완벽하게 설명하기 위하여 개인, 회사, 상표 및 제품의 이름이 사용될 수 있습니다. 이들 이름은 모두 가공의 것이며 실제 기업의 이름 및 주소와 유사하더라도 이는 전적으로 우연입니다.

저작권 라이선스:

이 정보에는 여러 운영 플랫폼에서의 프로그래밍 기법을 보여주는 원어로 된 샘플 응용프로그램이 들어 있습니다. 귀하는 이러한 샘플 프로그램의 작성 기준이 된 운영 플랫폼의 응용프로그램 프로그래밍 인터페이스(API)에 부합하는 응용프로그램을 개발, 사용, 판매 또는 배포할 목적으로 IBM에 추가 비용을 지불하지 않고 이들 샘플 프로그램을 어떠한 형태로든 복사, 수정 및 배포할 수 있습니다. 이러한 샘플 프로그램은 모든 조건하에서 완전히 테스트된 것은 아닙니다. 따라서 IBM은 이들 샘플 프로그램의 신뢰성, 서비스 가능성 또는 기능을 보증하거나 진술하지 않습니다.

이 정보를 소프트웨어로 확인하는 경우에는 사진과 컬러 삽화가 제대로 나타나지 않을 수도 있습니다.

프로그래밍 인터페이스 정보

프로그래밍 인터페이스 정보는 본 프로그램과 함께 사용하기 위한 응용프로그램 소프트웨어 작성을 돕기 위해 제공됩니다.

이 책에는 고객이 프로그램을 작성하여 WebSphere MQ서비스를 얻을 수 있도록 하는 계획된 프로그래밍 인터페이스에 대한 정보가 포함되어 있습니다.

그러나 본 정보에는 진단, 수정 및 성능 조정 정보도 포함되어 있습니다. 진단, 수정 및 성능 조정 정보는 응용프로그램 소프트웨어의 디버거를 돕기 위해 제공된 것입니다.

중요사항: 이 진단, 수정 및 튜닝 정보는 변경될 수 있으므로 프로그래밍 인터페이스로 사용하지 마십시오.

상표

IBM, IBM 로고, ibm.com®는 전세계 여러 국가에 등록된 IBM Corporation의 상표입니다. 현재 IBM 상표 목록은 웹 "저작권 및 상표 정보"(www.ibm.com/legal/copytrade.shtml)에 있습니다. 기타 제품 및 서비스 이름은 IBM 또는 타사의 상표입니다.

Microsoft 및 Windows는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 상표입니다.

UNIX는 미국 또는 기타 국가에서 사용되는 The Open Group의 등록상표입니다.

Linux는 미국 또는 기타 국가에서 사용되는 Linus Torvalds의 등록상표입니다.

이 제품에는 Eclipse 프로젝트 (<https://www.eclipse.org/>)에서 개발한 소프트웨어가 포함되어 있습니다.

Java 및 모든 Java 기반 상표와 로고는 Oracle 및/또는 그 계열사의 상표 또는 등록상표입니다.



부품 번호:

(1P) P/N: