9.4

컨테이너의 *IBM MQ*

참고

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 163 페이지의 『주의사항』에 있는 정보를 확인하십시오.

이 개정판은 새 개정판에 별도로 명시하지 않는 한, IBM® MQ 의 버전 9릴리스 4및 모든 후속 릴리스와 수정에 적용됩니다.

목차

컨테이너 및 IBM Cloud Pak for Integration 의 IBM MQ

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.

IBM MQ Operator에 대한 자세한 정보는 다음 링크를 참조하십시오.

컨테이너의 IBM MQ 정보

컨테이너에서 IBM MQ 를 시작하는 데 도움이 되는 소개 정보입니다.

컨테이너는 동일한 인프라의 다른 소프트웨어에서 격리되는 방식으로 실행될 수 있는 런타임 환경을 사용하여 코드의 패키징 및 격리를 허용하는 기술입니다. 이를 통해 환경 (예: 개발, 테스트 및 프로덕션) 간에 큐 관리자 또는 애플리케이션을 쉽게 이동할 수 있습니다. 최신 컨테이너 오케스트레이터 (예: Red Hat OpenShift Container Platform 및 Kubernetes) 는 동일한 시스템에서 여러 유형의 컨테이너를 실행할 수 있습니다. 각 컨테이너는 자원, 보안 및 장애 측면에서 서로 격리되어 있습니다.

컨테이너에서 IBM MQ 큐 관리자 또는 IBM MQ 애플리케이션을 실행할 수 있습니다.

관련 정보
컨테이너의 개념

IBM MQ Operator의 릴리스 히스토리

참고:

- 이전 IBM MQ 연산자에 대한 정보는 IBM MQ 9.3 문서에서 IBM MQ Operator의 릴리스 히스토리 를 참조하십시오.
- 향후 IBM MQ 업데이트에 대한 정보는 전체 IBM MQ 권장 수정사항 및 계획된 유지보수 릴리스 날짜 페이지를 참조하십시오.

IBM MQ Operator 3.2.1

IBM Cloud Pak for Integration 버전
IBM Cloud Pak for Integration 16.1.0

Operator 채널
`v3.2-sc2`

`.spec.version`에 허용되는 값
9.4.0.0-r1

마이그레이션 중에 `.spec.version` 에 허용되는 값
9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3,9.3.3.2-r3 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1, 9.3.0.16-r2, 9.3.0.17-r1, 9.3.0.17-r2, 9.3.0.17-r3, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3, 9.3.3.3-r1, 9.3.3.3-r2, 9.3.4.0-r1, 9.3.4.1-r1, 9.3.5.0-r1, 9.3.5.0-r2, 9.3.5.1-r1, 9.3.5.1-r2

Red Hat OpenShift Container Platform 버전
OpenShift Container Platform 4.12 이상.

IBM Cloud Pak foundational services 버전
IBM Cloud Pak foundational services 버전 4.6 전용.

변경사항

- 다음 문제를 해결합니다.OpenShift Container Platform4.12 업그레이드하는 곳v3.2-sc2 채널로 인해 예상치 못한 동작이 발생할 수 있습니다.IBM Cloud Pak for Integration 사용자. 자세한 내용은 다음을 참조하세요. 다음에서 업그레이드2023.4 에서IBM Cloud Pak for Integration 선적 서류 비치.

IBM MQ Operator 3.2.0

IBM Cloud Pak for Integration 버전
IBM Cloud Pak for Integration 16.1.0

Operator 채널
`v3.2-sc2`

`.spec.version`에 허용되는 값
9.4.0.0-r1

마이그레이션 중에 `.spec.version` 에 허용되는 값
9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3,9.3.3.2-r3 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1, 9.3.0.16-r2, 9.3.0.17-r1, 9.3.0.17-r2, 9.3.0.17-r3, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3, 9.3.3.3-r1, 9.3.3.3-r2, 9.3.4.0-r1, 9.3.4.1-r1, 9.3.5.0-r1, 9.3.5.0-r2, 9.3.5.1-r1, 9.3.5.1-r2

Red Hat OpenShift Container Platform 버전
OpenShift Container Platform 4.12 이상.

IBM Cloud Pak foundational services 버전
IBM Cloud Pak foundational services 버전 4.6 전용.

새로운 기능

- 91 페이지의 『지속적 볼륨 확장』 이 현재 지원됩니다.
- 이제 `mq.ibm.com/stop` 어노테이션을 추가하고 이를 `true`로 설정하여 큐 관리자를 중지할 수 있습니다. 94 페이지의 『큐 관리자 중지 (mq.ibm.com/stop)』 참조

  참고:

  - 중지된 큐 관리자의 `StatefulSet` 에 *.replica* 필드가 0으로 설정되어 있습니다.
  - 이제 IBM MQ Operator 가 `StatefulSet`의 *.replica* 필드를 능동적으로 관리하므로, 이 필드를 수정하면 운영자가 즉시 되돌립니다.
  - *.replica* 필드를 수정하지만 여전히 수정된 값을 유지하는 경우 IBM MQ 의 이전 버전은 '실패' 상태가 됩니다. 기존 운영 프로시저가 이 동작에 의존하는 경우 IBM MQ 9.4 에서 `mq.ibm.com/stop` 어노테이션을 사용해야 합니다.

변경사항

- Red Hat OpenShift Container Platform 의 홀수 번호 릴리스가 이제 지원됩니다.
- IBM MQ 카탈로그 이미지가 SQLite 데이터베이스 형식에서 파일 기반 카탈로그 형식으로 이동되었습니다.
- Red Hat Universal Base Image 9.4-949.1716471857기반. **참고:** UBI 9에는 보류 중인 FIPS 140-2인증이 있습니다. UBI 9는 Power® 8아키텍처에서 지원되지 않습니다.
- 주소 지정된 취약성은 이 보안 게시판에 자세히 설명되어 있습니다.

OpenShift CP4I IBM MQ Operator 에서 사용할 큐 관리자 컨테이너 이미지의 릴리스 히스토리

참고: 이전 큐 관리자 컨테이너 이미지에 대한 정보는 IBM MQ 9.3 문서의 IBM MQ Operator용 릴리스 히스토리를 참조하십시오.

9.4.0.0-r1

CD CP4I-SC2

필요한 Operator 버전
3.2.0 이상

지원되는 아키텍처
amd64, s390x, ppc64le

이미지

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.4.0.0-r1`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.4.0.0-r1`
- `icr.io/ibm-messaging/mq:9.4.0.0-r1`

새로운 기능

- IBM MQ 9.4.0 for Multiplatforms의 새로운 기능-기본 및 고급 인타이틀먼트

변경사항

- IBM MQ 9.4.0의 변경된 기능
- Deprecated IBM MQ Advanced for Developers를 사용하는 경우 환경 변수를 통해 admin 및 app 사용자의 비밀번호를 설정하는 것은 더 이상 사용되지 않습니다. 대신 비밀 정보를 사용하십시오.
- 환경 변수 *MQ_LOGGING_CONSOLE_SOURCE*에 대해 새 선택적 값 mqsc 가 추가되었습니다. 이 옵션을 사용하여 컨테이너 로그에서 autocfgmqsc.LOG 의 컨텐츠를 반영할 수 있습니다.
- Red Hat Universal Base Image 9.4-949.1716471857기반. **참고:** UBI 9에는 보류 중인 FIPS 140-2인증이 있습니다. UBI 9는 Power 8아키텍처에서 지원되지 않습니다.

컨테이너에 IBM MQ 계획

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(OpenShift Container Platform 아키텍처 참조)에 익숙해져야 합니다.

프로시저

- 8 페이지의 『컨테이너에서 IBM MQ를 사용하는 방법 선택』.
- 8 페이지의 『컨테이너에서 IBM MQ 에 대한 지원』.
- 15 페이지의 『IBM MQ Operator 의 스토리지 계획』.
- 16 페이지의 『컨테이너의 IBM MQ 에 대한 고가용성 계획』.
- 21 페이지의 『컨테이너에서 IBM MQ의 재해 복구』.
- 21 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

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

IBM MQ Operator 는 Red Hat OpenShift Container Platform API를 확장하여 새 `QueueManager` 사용자 정의 자원을 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. :NONE. 20 페이지의 『고유한 HA 큐 관리자의 롤링 업데이트를 수행하기 위한 고려사항』

일부 IBM MQ 기능은 IBM MQ Operator를 사용할 때 지원되지 않습니다. IBM MQ Operator사용 시 지원되는 사항에 대한 세부사항은 8 페이지의 『컨테이너에서 IBM MQ 에 대한 지원』 의 내용을 참조하십시오.

자체 이미지 및 배치 코드 빌드

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. 49 페이지의 『사용자 고유의 컨테이너 이미지를 빌드하여 IBM MQ 준비』의 내용을 참조하십시오.

사용자 고유의 이미지 및 배치 코드를 빌드할 때 지원되는 사항에 대한 세부사항은 8 페이지의 『컨테이너에서 IBM MQ 에 대한 지원』 의 내용을 참조하십시오.

관련 참조

8 페이지의 『컨테이너에서 IBM MQ 에 대한 지원』
모든 IBM MQ 기능이 컨테이너에서 동일한 방식으로 사용 가능하고 지원되는 것은 아닙니다.

OpenShift CP4I CD CP4I-SC2 컨테이너에서 IBM MQ 에 대한 지원

모든 IBM MQ 기능이 컨테이너에서 동일한 방식으로 사용 가능하고 지원되는 것은 아닙니다.

다음 표는 IBM MQ 기능이 IBM MQ Operator에서 지원되는 방법 또는 사용자 고유의 컨테이너 및 배치 코드를 빌드하는 경우를 자세히 표시합니다.

참고:

- IBM Container Registry (icr.io 및 cp.icr.io) 에서 사전 빌드된 IBM MQ 컨테이너 이미지는 IBM MQ Operator와 함께 사용되는 경우에만 지원되며 수정사항에 적합합니다.
- IBM MQ Operator 채널 v3.2에서 Long Term Support (LTS) 의 이름이 Support Cycle 2 (SC2) 로 바뀝니다. 이는 컨테이너의 IBM MQ 에 대해 사용 가능한 유일한 LTS 경로가 IBM Cloud Pak for Integration 인타이틀먼트 하에서 2년동안 지원되고 IBM Cloud Pak for Integration 에서 SC2라는 용어를 채택했기 때문입니다. 다음은 인타이틀먼트의 전체 그림입니다.
  - IBM MQ 인타이틀먼트를 사용하면 IBM MQ Operator 는 IBM MQ Continuous Delivery (CD) 이미지만 배치할 수 있습니다.
  - IBM Cloud Pak for Integration 인타이틀먼트를 사용하여 IBM MQ Operator 는 CD 또는 SC2 (formerly LTS) 이미지를 배치할 수 있습니다.

사전 빌드된 IBM MQ Advanced for Developers 이미지의 라이센스를 다른 라이센스로 "업그레이드" 할 수 없습니다. IBM MQ Operator 는 선택한 라이센스에 따라 다른 이미지를 배치합니다.

이 테이블에서 다음 용어가 적용됩니다.

"컨테이너 인에이블먼트 코드"
실행 파일 **runmqserver**, **runmqintegrationserver**, **chkmqhealthy**, **chkmqready** 및 **chkmqstarted**. 이 코드는 샘플로 제공되며 IBM MQ Operator와 함께 사용될 때 사전 빌드된 컨테이너의 일부로만 지원됩니다.

	IBM MQ Operator 및 IBM Cloud Pak for Integration 라이센스 사용	**IBM MQ Operator 및 IBM MQ Advanced 라이센스 사용**	**IBM MQ Operator 및 IBM MQ Advanced for Developers 라이센스 사용**	**사전 빌드된 IBM MQ Advanced for Developers 이미지**	**Build-your-own 컨테이너**
지원되는 플랫폼	Red Hat OpenShift Container Platform 에서만 지원됩니다. Red Hat OpenShift Container Platform의 릴리스는 Red Hat 가 지원을 중지하면 IBM MQ 에서 더 이상 지원되지 않습니다. 자세한 내용은 13 페이지의 『IBM MQ Operator에 대한 버전 지원』 의 내용을 참조하십시오.		Red Hat OpenShift Container Platform 에서만 사용 가능하지만 지원되지는 않습니다.	Docker, containerd 또는 cri-o 플랫폼에서 작동하지만 지원되지 않습니다. 자세한 내용은 IBM MQ 의 시스템 요구사항의 내용을 참조하십시오.	Docker, containerd 또는 cri-o 플랫폼. 자세한 내용은 IBM MQ 의 시스템 요구사항의 내용을 참조하십시오. 기본 HA는 Kubernetes 또는 Red Hat OpenShift Container Platform에서만 지원됩니다. 샘플 컨테이너 이미지는 IBM MQ 에서 사용되는 Linux® 라이브러리 및 유틸리티를 포함하는 Red Hat Universal Base Image (UBI) 를 사용합니다. UBI는 Red Hat OpenShift 에서 실행될 때 Red Hat 에서 지원됩니다. 컨테이너 인에이블먼트 코드 는 지원되지 않습니다.
CPU 아키텍처	amd64 및 s390x z/Linux에서 지원됩니다. ppc64le Power Systems 버전 9이상 시스템에서도 지원됩니다. Red Hat OpenShift Container Platform 클러스터의 모든 노드는 동일한 CPU 아키텍처를 사용해야 합니다.		amd64 및 s390x z/Linux에서 사용 가능하지만 지원되지 않습니다. ppc64le Power Systems 버전 9 이상 시스템에서도 사용 가능하지만 지원되지 않습니다. Red Hat OpenShift Container Platform 클러스터의 모든 노드는 동일한 CPU 아키텍처를 사용해야 합니다.		IBM MQ 소프트웨어에 따라.

	IBM MQ Operator 및 IBM Cloud Pak for Integration 라이센스 사용	**IBM MQ Operator 및 IBM MQ Advanced 라이센스 사용**	**IBM MQ Operator 및 IBM MQ Advanced for Developers 라이센스 사용**	**사전 빌드된 IBM MQ Advanced for Developers 이미지**	**Build-your-own 컨테이너**
지원 지속 기간	IBM Cloud Pak for Integration - Support Cycle 2 (formerly Long Term Support) 또는 Continuous Delivery.[1] CD 운영자 및 큐 관리자는 다음 IBM Cloud Pak for Integration CD 또는 CP4I-SC2 릴리스까지 지원됩니다. CP4I-SC2 운영자 및 큐 관리자는 다음 IBM Cloud Pak for Integration CP4I-SC2 릴리스까지 지원되며 업그레이드를 허용하기 위한 유예 기간이 추가됩니다.	IBM MQ Operator및 큐 관리자 둘 다에 대해 Continuous Delivery 스트림 전용입니다. 각 IBM MQ Operator 및 큐 관리자 버전은 다음 CD 릴리스까지만 지원됩니다.	지원되지 않음		IBM MQ 소프트웨어에 따라. 장기 지원 및 Continuous Delivery 릴리스에 대한 IBM MQ FAQ을 참조하십시오. 컨테이너 인에이블먼트 코드 는 지원되지 않습니다.
보안 수정사항 가용성	IBM Container Registry 에서 컨테이너 이미지로 사용 가능한 주기적 수정사항				IBM MQ 소프트웨어에 대한 수정사항은 Fix Central에서 소프트웨어로 사용 가능합니다. 컨테이너 인에이블먼트 코드 는 지원되지 않습니다.

[1] IBM MQ Operator 는 IBM MQ CD 릴리스 또는 IBM Cloud Pak for Integration - Support Cycle 2 (formerly Long Term Support) 릴리스로 지원됩니다.

- IBM MQ Operator 3.2.x와 함께 배치된 IBM MQ 9.4.0.x 컨테이너 이미지는 IBM Cloud Pak for Integration 16.1.0의 일부로 사용될 때 CP4I-LTS 지원에 적합합니다. IBM MQ Operator 의 최신 Support Cycle 2 (SC2) 릴리스는 3.2.1이고 최신 SC2 컨테이너 이미지는 9.4.0.0-r1입니다.
- IBM MQ Operator 3.2.x와 함께 배치된 IBM MQ 9.4.0.x 컨테이너 이미지는 IBM Cloud Pak for Integration 16.1.0의 일부로 사용될 때 CD 지원에 적합합니다. IBM MQ Operator 의 최신 Continuous Delivery (CD) 릴리스는 3.2.1이고 최신 CD 컨테이너 이미지는 9.4.0.0-r1입니다.

	IBM MQ Operator 및 IBM Cloud Pak for Integration 라이센스 사용	**IBM MQ Operator 및 IBM MQ Advanced 라이센스 사용**	**IBM MQ Operator 및 IBM MQ Advanced for Developers 라이센스 사용**	**사전 빌드된 IBM MQ Advanced for Developers 이미지**	**Build-your-own 컨테이너**
임시 수정사항 가용성	소프트웨어로 사용 가능한 큐 관리자 수정사항 및 사용자 정의 이미지 빌드가 필요합니다. IBM MQ Operator 수정사항은 임시 수정사항으로 사용할 수 없습니다.		사용 가능한 임시 수정사항이 없습니다.		IBM MQ 소프트웨어에 대한 수정사항은 Fix Central 에서 또는 IBM 지원 센터를 통해 소프트웨어로 사용 가능합니다. 컨테이너 인에이블먼트 코드 는 지원되지 않습니다.
기능: Advanced Message Security	지원됨. IBM MQ Operator 에서는 Advanced Message Security에 사용자 고유의 키 저장소를 직접 지정할 수 없으므로 서버 측 암호화를 사용하는 것이 쉽지 않습니다.		사용 가능하지만 지원되지 않습니다.		IBM MQ 소프트웨어에 따라 지원되지만 사용 가능한 샘플이 없습니다.
기능: Managed File Transfer	사용할 수 없으며 지원되지 않습니다. 그러나 IBM MQ Operator 를 사용하여 하나 이상의 조정, 명령 또는 에이전트 큐 관리자를 제공할 수 있습니다.			사용할 수 없으며 지원되지 않습니다.	IBM MQ 소프트웨어에 따라 에이전트에 대한 샘플 과 함께 지원됩니다.
기능: MQTT	사용할 수 없으며 지원되지 않습니다.				IBM MQ 소프트웨어에 따라 지원되지만 사용 가능한 샘플이 없습니다.
기능: AMQP	사용할 수 없으며 지원되지 않습니다.				IBM MQ 소프트웨어에 따라 지원되지만 사용 가능한 샘플이 없습니다.
기능: REST API	사용 가능하며 지원됩니다.				IBM MQ 소프트웨어에 따라 사용 가능하고 지원됩니다.
기능: 복제된 데이터 큐 관리자	사용할 수 없으며 지원되지 않습니다. 복제된 데이터 큐 관리자 (RDQM) 는 Linux 커널과 단단히 결합되어 있으며 컨테이너에서 지원되지 않습니다.				

	IBM MQ Operator 및 IBM Cloud Pak for Integration 라이센스 사용	**IBM MQ Operator 및 IBM MQ Advanced 라이센스 사용**	**IBM MQ Operator 및 IBM MQ Advanced for Developers 라이센스 사용**	**사전 빌드된 IBM MQ Advanced for Developers 이미지**	**Build-your-own 컨테이너**
기능: 고유 HA	사용 가능하며 지원됩니다.		사용 가능하지만 지원되지 않습니다.		Kubernetes 및 Red Hat OpenShift Container Platform에서만 사용 가능합니다. IBM MQ 소프트웨어에 따라 지원됩니다.
기능: 다중 인스턴스 큐 관리자	사용 가능하며 지원됩니다.		사용 가능하지만 지원되지 않습니다.		IBM MQ 소프트웨어에 따라 사용 가능하고 지원됩니다.
기능: 복구 로그 유형	순환 로깅 또는 복제된 로그만. 선형 로깅은 지원되지 않습니다.				IBM MQ 소프트웨어에 따라 사용 가능하고 지원됩니다. **crtmqm** 옵션을 구성해야 합니다.
기능: crtmqdir, crtmqm, strmqm 및 endmqm에 대한 사용자 정의 명령행 옵션 지정	사용할 수 없으며 지원되지 않습니다. 대부분의 옵션은 INI 파일을 사용하여 구성할 수 있지만 선형 로깅 사용과 같은 일부 옵션은 구성할 수 없습니다.				컨테이너 인에이블먼트 코드를 구현하는 방법에 따라 선택사항입니다.
기능: 운영 체제 (OS) 사용자	사용할 수 없으며 지원되지 않습니다.				RPM을 사용하여 IBM MQ 를 설치하지만 사용 가능한 샘플이 없는 경우 IBM MQ 소프트웨어에 따라 가능하고 지원됩니다. 보안 위험으로 인해 권장되지 않습니다.

참고: " IBM MQ 소프트웨어에 따라 지원됨" 구문은 IBM 기술 지원이 컨테이너 내부에서 실행 중인 코어 IBM MQ 소프트웨어로 제한됨을 의미합니다.

관련 개념

Long Term Support 및 Continuous Delivery 릴리스에 대한 IBM MQ FAQ

관련 참조

IBM Cloud Pak for Integration 소프트웨어 지원 라이프사이클 부록

IBM MQ Operator에 대한 버전 지원

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

- 13 페이지의 『사용 가능한 IBM MQ 버전』
- 13 페이지의 『호환 가능한 Red Hat OpenShift Container Platform 버전』
- 13 페이지의 『IBM Cloud Pak for Integration 버전』
- 14 페이지의 『이전 Operator에서 사용 가능한 IBM MQ 버전』
- 14 페이지의 『이전 Operator에 호환 가능한 OpenShift Container Platform 버전』

사용 가능한 IBM MQ 버전

Operator 채널	Operator 버전	IBM MQ 버전						
		9.4.0	9.3.5	9.3.4	9.3.3	9.3.2	9.3.1	9.3.0
v32-sc2	3.2	CD 및 SC2	dep	dep	dep	dep	dep	MIG

키:

- CD: Continuous Delivery 지원이 사용 가능합니다.
- SC2: IBM Cloud Pak for Integration - Support Cycle 2 (formerly Long Term Support) 를 사용할 수 있습니다.
- MIG: IBM Cloud Pak for Integration - Support Cycle 2 (formerly Long Term Support) 피연산자에서 Continuous Delivery 피연산자로 마이그레이션하는 동안에만 사용 가능합니다.
- DEP: Deprecated 더 이상 사용되지 않습니다. IBM MQ 릴리스가 지원되지 않으므로 운영자에서 계속 구성할 수 있지만 더 이상 지원을 받을 수 없으며 향후 릴리스에서 제거될 수 있습니다.

각 버전에서의 상세 기능, 변경사항 및 수정사항을 포함하여 각 버전의 전체 세부사항에 대해서는 5 페이지의 『IBM MQ Operator의 릴리스 히스토리』의 내용을 참조하십시오.

호환 가능한 Red Hat OpenShift Container Platform 버전

Operator 채널	Operator 버전	OpenShift Container Platform 버전[2]		
		4.15	4.14	4.12
v3.2-sc2	3.2.0 이상	SC2	SC2	SC2

키:

- CD: Continuous Delivery 지원이 사용 가능합니다.
- SC2: IBM Cloud Pak for Integration - Support Cycle 2 (formerly Long Term Support) 를 사용할 수 있습니다.
- EOS: 더 이상 지원되지 않습니다. 이후 OpenShift Container Platform 버전으로 마이그레이션하십시오.

IBM Cloud Pak for Integration 버전

IBM Cloud Pak for Integration 버전 16.1.0의 일부로 또는 독립적으로 사용하도록 지원됩니다.

- IBM MQ Operator 3.2.x

[2] OpenShift Container Platform 버전에는 자체 지원 날짜가 적용됩니다. 자세한 정보는 OpenShift Container Platform 라이프사이클 정책을 참조하십시오.

이전 Operator에서 사용 가능한 IBM MQ 버전

IBM MQ 9.3 문서에서 사용 가능한 IBM MQ 버전 을 참조하십시오.

이전 Operator에 호환 가능한 OpenShift Container Platform 버전

IBM MQ 9.3 문서에서 호환 가능 OpenShift Container Platform 버전 을 참조하십시오.

IBM MQ Operator 에서 작성한 자원 편집

IBM MQ Operator 는 원시 Kubernetes 자원을 작성하고 관리하여 QueueManager 사용자 정의 자원을 조정합니다. 이러한 관리 자원은 직접 편집할 수 **없습니다** .

일반적으로 `ownerReferences`를 보고 다른 상위 레벨 자원이 자원을 소유하는지 여부를 판별할 수 있습니다. 예를 들어, `StatefulSet` 에서 가져온 다음 메타데이터는 `QueueManager` 자원 "qm1" 이 소유함을 표시합니다.

```
metadata:
  ownerReferences:
    - apiVersion: mq.ibm.com/v1beta1
      kind: QueueManager
      name: qm1
      uid: 60fda34c-9f7c-42d2-a293-78fec4315c62
      controller: true
      blockOwnerDeletion: true
```

모든 자원에 이 메타데이터가 있는 것은 아닙니다.

`StatefulSet`, `Service` 및 `Route`와 같은 기본 자원을 관리하는 것은 IBM MQ Operator 의 책임입니다. 이러한 기본 자원 중 하나를 변경하면 IBM MQ Operator 에서 해당 자원을 다시 변경하며 해당 변경에 롤링 업데이트가 필요한 경우 작동 중지 시간이 발생할 수 있습니다.

큐 관리자에 대한 대부분의 중요한 설정은 `QueueManager` 자원에서 사용 가능합니다. 그러나 기본 자원을 완전히 제어해야 하는 경우에는 몇 가지 옵션이 있습니다.

- IBM MQ Operator에서 작성한 팟 (Pod) 의 설정을 대체해야 하는 경우 `QueueManager` YAML 의 `.spec.template` 섹션에서 팟 (Pod) 대체 템플리트를 추가할 수 있습니다.
- IBM MQ Operator에 의해 작성된 큐 관리자 `Route` 의 설정을 대체해야 하는 경우 라우트를 완전히 사용 안함으로 설정하고 `.spec.route.enabled` 설정을 "false" 로 설정한 후 사용자 고유의 라우트를 작성해야 합니다.
- 레이블 및 어노테이션과 같은 설정과 Pod 설정 (예: `security Context`) 은 모두 `QueueManager` 자원에서 설정할 수 있습니다.
- 다른 경우에는 전체 제어가 필요한 경우 IBM MQ Operator 가 유스 케이스에 적합하지 않을 수 있습니다.

컨테이너에서 IBM MQ 라이센싱 계획

컨테이너 라이센싱을 사용하면 컨테이너가 실행 중인 전체 서버에 라이센스를 부여하지 않고 개별 IBM MQ 컨테이너의 사용 가능한 용량에만 라이센스를 부여할 수 있습니다. 컨테이너 라이센싱을 이용하려면 IBM License Service 를 사용하여 라이센스 사용을 추적하고 필요한 인타이틀먼트를 판별해야 합니다.

관련 참조

150 페이지의 『IBM MQ 컨테이너 이미지를 직접 빌드하는 경우의 라이센스 어노테이션』
라이센스 어노테이션은 기반 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.

관련 정보

IBM 컨테이너 라이센스
컨테이너 라이센싱 FAQ
License Service 설치
라이센스 사용 보기 및 추적

OpenShift CP4I Kubernetes IBM MQ Operator 의 스토리지 계획

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 85 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.

OpenShift CP4I 예를 들어, IBM MQ를 임시 스토리지로 구성하려면, `QueueManager`의 스토리지 유형은 다음을 포함합니다.

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

지속 스토리지

OpenShift CP4I

IBM MQ 는 일반적으로 큐 관리자가 재시작 후에 해당 지속 메시지 및 구성을 보유하도록 보장하기 위해 지속적 스토리지와 함께 실행됩니다. 이는 기본 작동입니다. 각각 서로 다른 기능을 지원하는 다양한 스토리지 제공자가 있기 때문에 이는 종종 구성의 사용자 정의가 필요함을 의미합니다. 아래 예제에서는 v1beta1 API에서 IBM MQ 스토리지 구성을 사용자 정의하는 공통 필드의 개요를 설명합니다.

- **`spec.queueManager.availability`** 는 가용성 모드를 제어합니다. `SingleInstance` 또는 `NativeHA`를 사용 중인 경우 `ReadWriteOnce` 스토리지만 필요합니다. `multiInstance` 의 경우 올바른 파일 잠금 특성이 있는 `ReadWriteMany` 를 지원하는 스토리지 클래스가 필요합니다. IBM MQ는 지원 명령문 및 명령문 테스트를 제공합니다. 또한 가용성 모드는 지속 볼륨 레이아웃에 영향을 줍니다. 추가 정보는 16 페이지의 『컨테이너의 IBM MQ 에 대한 고가용성 계획』의 내용을 참조하십시오.
- **`spec.queueManager.storage`** 는 개별 스토리지 설정을 제어합니다. 1-4개의 지속적 볼륨을 사용하도록 큐 관리자를 구성할 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: [65534] # Change to 99 for clusters with RHEL7 or earlier worker nodes
```

고유 HA큐 관리자의 스토리지 고려사항에 대한 정보는 18 페이지의 『고유 HA』의 내용을 참조하십시오.

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

스토리지 용량

OpenShift CP4I

IBM MQ Operator 를 사용하는 경우 진행 중인 요구사항에 맞게 충분히 큰 볼륨을 요청하는지 확인해야 합니다. 그러나 하나 이상의 볼륨의 스토리지 용량을 늘려야 하는 경우 스토리지 클래스가 볼륨 확장을 지원하면 이러한 볼륨을 확장할 수 있습니다. 온라인 또는 오프라인 프로시저로 볼륨을 확장할 수 있습니다. 오프라인 프로시저에서는 QueueManager 팟 (Pod) 을 다시 시작해야 하지만, 온라인 프로시저에서는 그렇지 않습니다. 스토리지 클래스가 볼륨 확장을 지원하는지 여부 및 볼륨 확장이 따르는 프로시저를 판별하려면 스토리지 제공자 문서를 참조하십시오. 스토리지 클래스를 선택할 때 이 정보를 고려해야 합니다. 볼륨 확장에 대한 안내서는 91 페이지의 『지속적 볼륨 확장』의 내용을 참조하십시오.

암호화

OpenShift CP4I

IBM MQ 는 저장된 데이터를 능동적으로 암호화하지 않습니다. 따라서 수동으로 암호화된 스토리지나 IBM MQ Advanced Message Security또는 둘 다 사용하여 메시지를 암호화해야 합니다. IBM Cloud® 에서는 비활성 상태의 수동 암호화를 사용하여 블록 및 파일 스토리지를 둘 다 사용할 수 있습니다.

OpenShift Kubernetes 컨테이너의 IBM MQ 에 대한 고가용성 계획

IBM MQ Operator의 고가용성을 위한 세 가지 선택사항이 있습니다. **원시 HA큐 관리자** (활성 복제본 및 두 개의 대기 복제본이 있음), **멀티 인스턴스 큐 관리자** (공유된 네트워크 파일 시스템을 사용하는 활성-대기 쌍) 또는 **단일 복원력 큐 관리자** (네트워크 스토리지를 사용하는 HA에 대한 단순한 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. 큐 관리자가 동일한 데이터 (Kubernetes 지속적 볼륨에서 제공됨) 를 유지하고 클라이언트 애플리케이션이 네트워크에서 여전히 주소를 지정할 수 있는 한, 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 제공하느 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

기본 HA큐 관리자에는 **활성** 및 두 개의 **복제본** Kubernetes 팟 (Pod) 이 포함되며, 이는 각각 고유한 Kubernetes 지속적 볼륨 세트가 있는 정확히 세 개의 복제본이 있는 Kubernetes StatefulSet 의 일부로 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 스토리지를 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 18 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. 이를 구성하려면 **`access mode`** 가 `ReadWriteMany`로 설정된 Kubernetes 지속적 볼륨을 사용하십시오. 또한 이 볼륨은 IBM MQ 의 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. `IfNotPresent`의 **`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. Kubernetes StatefulSet와 함께 IBM MQ 와 같은 Stateful 워크로드를 사용할 때 Kubernetes 마스터 노드가 작업자 노드와의 연결이 끊어지면 노드가 실패했는지 또는 단순히 네트워크 연결이 끊어졌는지 판별할 수 없습니다. 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.

참고: 복제본 수를 포함하여 IBM MQ 큐 관리자의 StatefulSet 에 대한 세부사항 변경은 큐 관리자가 IBM MQ Operator를 통해 작성될 때 지원되지 않습니다.

관련 개념

고가용성 구성

관련 태스크

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

CP4I MQ Adv. 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 유형을 '복제된 로그' 라고 합니다. 복제된 로그는 기본적으로 자동 로그 관리 및 자동 매체 이미지가 사용 가능한 선형 로그입니다. 로깅 유형을 참조하십시오. 선형 로그를 관리하는 데 사용하는 것과 동일한 기술을 복제된 로그를 관리하는 데 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 복원되면 서로 다른 버전의 두 데이터('분할 뇌')가 있게 되고 보존할 데이터 세트 및 제거할 데이터 세트를 판별하려면 수동 개입이 필요합니다.

고유 HA에서는 쿼럼이 있는 세 개의 팟(Pod) 시스템을 사용하여 분할 뇌 상황을 방지합니다. 다른 팟(Pod) 중 하나 이상과 통신할 수 있는 팟(Pod)은 쿼럼을 구성합니다. 쿼럼이 있는 팟(Pod)에서는 큐 관리자만이 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 쿼럼이 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 쿼럼이 없으므로 큐 관리자는 이 팟(Pod)에서 활성화될 수 없습니다. 나머지 두 팟(Pod) 중 쿼럼이 있는 하나의 큐 관리자가 인계할 수 있습니다. 모든 팟(Pod)에서 연결이 끊어지면 쿼럼이 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.

성능 및 신뢰성을 위해 원시 HA 구성과 함께 사용하도록 RWO (ReadWriteOnce) 지속적 스토리지를 사용하는 것이 좋습니다. 다음 조건을 충족하는 경우 스토리지 제공자의 RWO 볼륨이 지원됩니다.

- 블록 스토리지 제공자에서 얻습니다.
- ext4 또는 XFS (POSIX 준수 확인) 로 형식화됩니다.
- 동적 볼륨 프로비저닝 및 "volumeBinding모드: WaitForFirstConsumer" 를 지원합니다.

다음 제공자는 명시적으로 금지됩니다.

- NFS
- GlusterFS
- 기타 비블록 제공자.

다음 그림은 한 큐 관리자의 세 인스턴스가 세 컨테이너에 배치되는 일반 배치를 표시합니다.

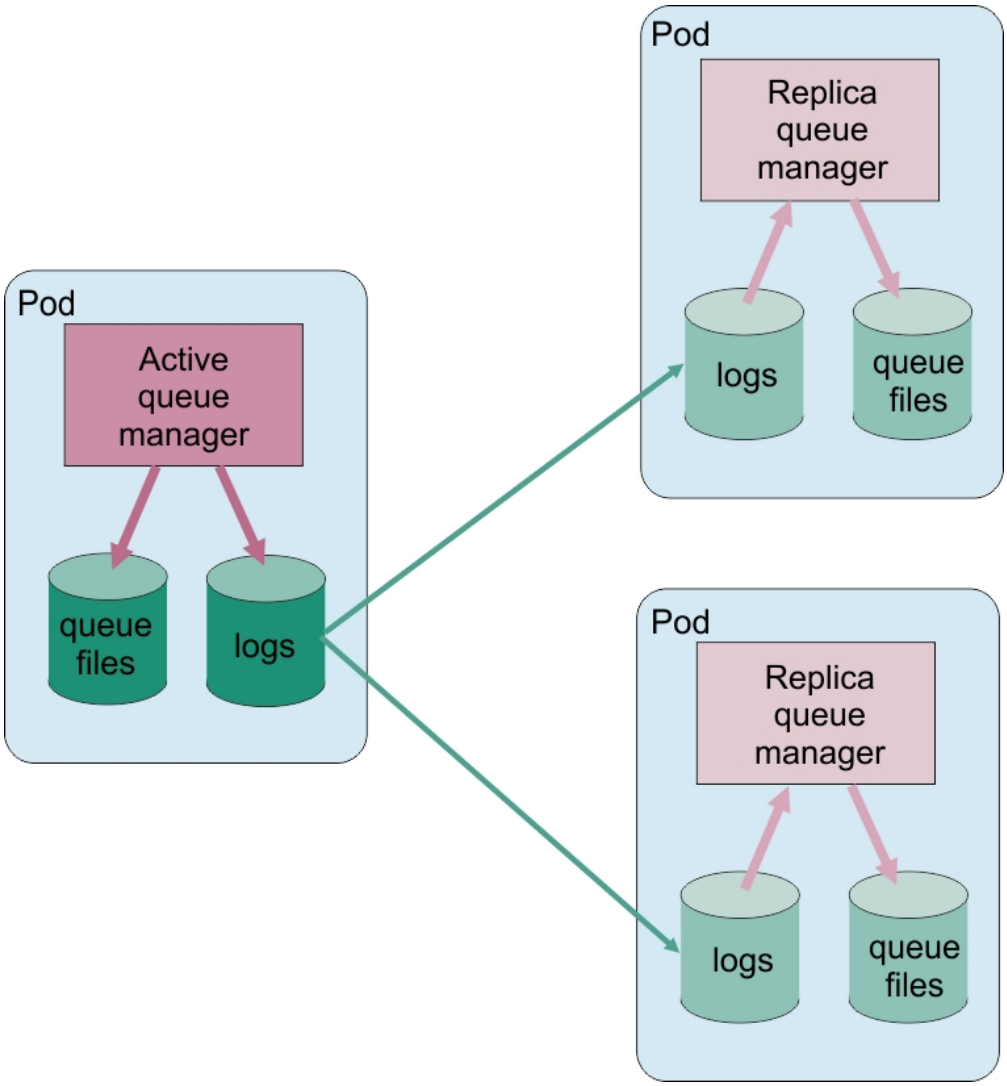


그림 *1.* 고유 *HA* 구성의 예제

MQ Adv. 고유한 HA 큐 관리자의 롤링 업데이트를 수행하기 위한 고려사항

원시 HA큐 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 쉘 스크립트가 포함되어 있지만 이 주제의 고려사항을 해결하지 않으므로 스크립트는 프로덕션 사용에 적합한 **되지 않음**입니다.

Kubernetes 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 `REPLICA`의 `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 `REPLICA` 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 `REPLICA` 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 액세스가능하지 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

8 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.

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 제공자에만 해당되지만 `VolumeSnapshot`을(를) 사용하여 수행할 수도 있습니다. 볼륨 스냅샷에 대한 자세한 정보는 CSI 볼륨 스냅샷을 참조하십시오.

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

관련 정보

ha_for_ctr.dita

OpenShift CP4I 컨테이너에서 IBM MQ 에 대한 보안 계획

컨테이너 구성에서 IBM MQ 를 계획할 때 보안 고려사항입니다.

프로시저

- 21 페이지의 『컨테이너에서 IBM MQ에 대한 사용자 인증 및 권한 부여』
  - 22 페이지의 『컨테이너에서 운영 체제 사용자 사용에 대한 보안 제한조건』
- 22 페이지의 『컨테이너에서 네트워크 트래픽을 IBM MQ 로 제한하기 위한 고려사항』

컨테이너에서 IBM MQ에 대한 사용자 인증 및 권한 부여

컨테이너의 IBM MQ 는 LDAP, 상호 TLS 또는 사용자 정의 MQ 플러그인을 통해 사용자를 인증하도록 구성할 수 있습니다.

IBM MQ 운영자는 컨테이너 이미지 내에서 운영 체제 사용자 및 그룹의 사용을 허용하지 않습니다. 자세한 정보는 22 페이지의 『컨테이너에서 운영 체제 사용자 사용에 대한 보안 제한조건』의 내용을 참조하십시오.

LDAP

LDAP 사용자 저장소를 사용하도록 IBM MQ 를 구성하는 방법에 대한 정보는 연결 인증: 사용자 저장소 및 LDAP 권한 부여를 참조하십시오.

상호 TLS

TLS 인증서 (상호 TLS) 가 필요하도록 큐 관리자에 대한 수신 연결을 구성하는 경우 인증서의 식별 이름을 사용자 이름에 맵핑할 수 있습니다. 다음 두 가지를 수행해야 합니다.

- SSLPEER를 사용하여 사용자 이름에 대한 맵핑을 작성하도록 채널 인증 레코드를 구성하십시오. 자세한 정보는 SSL 또는 TLS 식별 이름을 MCAUSER 사용자 ID에 맵핑을 참조하십시오.
- 시스템에 알려지지 않은 사용자 이름에 대한 권한 레코드를 정의할 수 있도록 큐 관리자를 구성하십시오. 자세한 정보는 qm.ini 파일의 서비스 스탠자를 참조하십시오.

JSON웹 토큰

JWT (JSON Web Tokens) 를 사용하도록 IBM MQ 를 구성하는 방법에 대한 정보는 인증 토큰에 대한 작업을 참조하십시오.

사용자 정의 MQ 플러그인

이는 고급 기술이며 더 많은 작업이 필요합니다. 자세한 정보는 사용자 정의 권한 서비스 사용을 참조하십시오.

관련 태스크

63 페이지의 『예: 상호 TLS 인증을 사용하여 큐 관리자 구성』
이 예는 IBM MQ Operator를 사용하여 OpenShift Container Platform에 큐 관리자를 배치합니다. 상호 TLS는 TLS 인증서에서 큐 관리자의 ID로 맵핑하기 위해 인증에 사용됩니다.

컨테이너에서 운영 체제 사용자 사용에 대한 보안 제한조건

컨테이너에서 운영 체제 사용자를 사용하는 것은 권장되지 않으며 IBM MQ 운영자와 함께 사용할 수 없습니다.

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(restricted`라고 함)에서는 각 컨테이너에 대해 무작위 사용자 ID를 사용합니다.
- **권한 상승 사용 금지**. IBM MQ on Linux 은 권한 상승을 사용하여 사용자의 비밀번호를 확인합니다. 이를 수행하기 위해 "root" 사용자가 되도록 "setuid" 프로그램을 사용합니다.

OpenShift CP4I 이러한 보안 조치를 준수하기 위해 IBM MQ Operator 는 컨테이너 내의 운영 체제 라이브러리에 정의된 ID의 사용을 허용하지 않습니다. 컨테이너에 `mqm` 사용자 ID 및 그룹이 정의되지 않습니다.

컨테이너에서 네트워크 트래픽을 IBM MQ 로 제한하기 위한 고려사항

OpenShift Container Platform 및 Kubernetes에서 클러스터의 팟 (Pod) 에 대한 트래픽을 제한하도록 네트워크 정책을 정의할 수 있습니다. 이 주제에서는 네트워크 정책을 IBM MQ에 적용하는 방법에 대한 몇 가지 고려사항을 설명합니다.

큐 관리자에 대한 네트워크 수집의 경우 고려할 여러 포트가 있습니다.

- 큐 관리자 트래픽에 대한 포트 1414
- 천연 HA에 대한 포트 9414
- 지표용 포트 9157
- 웹 콘솔 및 REST API용 포트 9443

네트워크 출구는 더 복잡하다. 고려할 수 있는 네트워크 유출의 예는 다음과 같습니다.

- DNS-DNS 이름을 사용하는 채널 또는 기타 구성이 있는 경우
- 다른 큐 관리자
- 인증서 제공자가 결정한 온라인 인증서 상태 프로토콜 (OCSP) 과 인증서 해지 목록 (CRL).
- 인증 공급자:

  - LDAP
  - IBM MQ 웹 서버에 대해 ID 연결 또는 기타 구성된 로그인 제공자를 여십시오. 여기에는 IBM Cloud Pak Keycloak가 포함됩니다.
- 추적 제공자:
  - IBM Instana

  참고: 이전 IBM MQ 버전의 경우 IBM Cloud Pak for Integration Operations Dashboard도 추적 제공자로 사용할 수 있었습니다. 그러나 운영 대시보드는 IBM MQ 9.3.3 CD 및 IBM MQ 9.4.0 LTS에서 제거되었습니다.

예제 수집 NetworkPolicy

다음은 Red Hat OpenShift Container Platform에서 사용하기 위해 큐 관리자 "myqm" 에 대한 진입을 제어하는 예제 네트워크 정책입니다.

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: myqm
spec:
  podSelector:
    matchLabels:
      app.kubernetes.io/instance: myqm
      app.kubernetes.io/name: ibm-mq
  ingress:
    # Allow access to queue manager listener from anywhere
    - ports:
        - protocol: TCP
          port: 1414
    # Allow access to Native HA port from other instances of the same queue manager
    - from:
        - podSelector:
            matchLabels:
              app.kubernetes.io/instance: myqm
              app.kubernetes.io/name: ibm-mq
      ports:
        - protocol: TCP
          port: 9414
    # Allow access to metrics from monitoring project
    - from:
        - namespaceSelector:
            matchLabels:
              network.openshift.io/policy-group: monitoring
      ports:
        - protocol: TCP
          port: 9157
    # Allow access to web server via Route
    - from:
        - namespaceSelector:
            matchLabels:
              network.openshift.io/policy-group: ingress
      ports:
        - protocol: TCP
          port: 9443
```

컨테이너의 IBM MQ 에 대한 FIPS 준수

시작 시 컨테이너의 IBM MQ 는 컨테이너가 시작되는 운영 체제가 FIPS를 준수하는지 여부를 발견하고 (해당 경우) FIPS 지원을 자동으로 구성합니다. 요구사항 및 제한사항이 여기에 설명되어 있습니다.

FIPS(Federal Information Processing Standard)

미국 정부에서는 데이터 암호화 같은 IT 시스템과 보안에 대한 기술적 자문을 만들고 있습니다. NIST (National Institute for Standards and Technology) 는 IT 시스템 및 보안과 관련된 정부 기관입니다. NIST는 FIPS(Federal Information Processing Standard)를 포함하는 표준 및 권장사항을 생성합니다.

중요한 FIPS 표준은 강력한 암호화 알고리즘을 사용해야 하는 FIPS 140-2입니다. FIPS 140-2는 전송 중에 수정되지 못하도록 패킷을 보호하는 데 사용될 해싱 알고리즘에 대한 요구사항도 지정합니다.

IBM MQ 는 FIPS 140-2지원을 제공합니다 (FIPS 140-2지원을 제공하도록 구성된 경우).

참고: AIX®, Linux, and Windows에서 IBM MQ 는 IBM Crypto for C (ICC) 암호화 모듈을 통해 FIPS 140-2준수를 제공합니다. 이 모듈의 인증서가 히스토리 상태로 이동되었습니다. 고객은 IBM Crypto for C (ICC) 인증서 를 보고 NIST에서 제공하는 조언을 알고 있어야 합니다. 대체 FIPS 140-2모듈이 현재 진행 중이며 해당 상태는 프로세스 목록의 NIST CMVP 모듈에서 검색하여 볼 수 있습니다.

IBM MQ Operator 3.2.0 및 큐 관리자 컨테이너 이미지 9.4.0.0 이상은 UBI 9를 기반으로 합니다. FIPS 140-2 준수가 현재 보류 중이며 해당 상태는 프로세스 목록의 NIST CMVP 모듈에서 "Red Hat Enterprise Linux 9-OpenSSL FIPS 제공자" 를 검색하여 볼 수 있습니다.

요구사항

클러스터 설정 및 기타 고려사항과 관련된 요구사항은 FIPS Wall: FIPS 준수에 대한 현재 IBM 접근 방식을 참조하십시오.

컨테이너의 IBM MQ 는 FIPS 140-2준수 모드에서 실행할 수 있습니다. 시작 중에 컨테이너의 IBM MQ 는 컨테이너가 시작되는 호스트 운영 체제가 FIPS를 준수하는지 여부를 발견합니다. 호스트 운영 체제가 FIPS를 준수하고 개인 키 및 인증서가 제공된 경우, IBM MQ 컨테이너는 큐 관리자, IBM MQ 웹 서버 및 원시 고가용성 배치의 노드 간 데이터 전송을 FIPS 준수 모드에서 실행하도록 구성합니다.

IBM MQ Operator 를 사용하여 큐 관리자를 배치할 때 운영자는 종료 유형이 **Passthrough**인 라우트를 작성합니다. 이는 트래픽이 TLS 종료를 제공하는 라우터 없이 목적지로 직접 전송됨을 의미합니다. 이 경우 IBM MQ 큐 관리자 및 IBM MQ 웹 서버가 대상이며 이미 FIPS 준수 보안 통신을 제공합니다.

주요 요구사항:

1. 외부 클라이언트가 큐 관리자 및 웹 서버에 안전하게 연결할 수 있도록 하는, 큐 관리자 및 웹 서버에 시크릿으로 제공되는 개인 키 및 인증서.
2. 원시 고가용성 구성에서 서로 다른 노드 간에 데이터를 전송하기 위한 개인 키 및 인증서.

제한사항

컨테이너에서 IBM MQ 의 FIPS 준수 배치의 경우 다음을 고려하십시오.

- 컨테이너의 IBM MQ 는 메트릭 콜렉션에 대한 엔드포인트를 제공합니다. 현재 이 엔드포인트는 HTTP 전용입니다. 나머지 IBM MQ FIPS를 준수하도록 메트릭 엔드포인트를 끌 수 있습니다.
- 컨테이너의 IBM MQ 는 사용자 정의 이미지 대체를 허용합니다. 즉, IBM MQ 컨테이너 이미지를 기본 이미지로 사용하여 사용자 정의 이미지를 빌드할 수 있습니다. 이러한 사용자 정의 이미지에는 FIPS 준수가 적용되지 않을 수 있습니다.
- IBM Instana를 사용하는 메시지 추적의 경우, IBM MQ 와 IBM Instana 간의 통신은 FIPS 준수가 없는 HTTP 또는 HTTPS입니다.
- IBM ID및 액세스 관리 (IAM) /참선 서비스에 대한 IBM MQ Operator 액세스는 FIPS를 준수하지 않습니다.

FIPS 준수가 발견되고 *FIPS* 지원이 자동으로 구성되는 방법

컨테이너가 시작되는 운영 체제가 FIPS를 준수하는 경우 FIPS 지원이 자동으로 구성됩니다.

참고: AIX, Linux, and Windows에서 IBM MQ 는 IBM Crypto for C (ICC) 암호화 모듈을 통해 FIPS 140-2준수를 제공합니다. 이 모듈의 인증서가 히스토리 상태로 이동되었습니다. 고객은 IBM Crypto for C (ICC) 인증서 를 보고 NIST에서 제공하는 조언을 알고 있어야 합니다. 대체 FIPS 140-2모듈이 현재 진행 중이며 해당 상태는 프로세스 목록의 NIST CMVP 모듈에서 검색하여 볼 수 있습니다.

IBM MQ Operator 3.2.0 및 큐 관리자 컨테이너 이미지 9.4.0.0 이상은 UBI 9를 기반으로 합니다. FIPS 140-2 준수가 현재 보류 중이며 해당 상태는 프로세스 목록의 NIST CMVP 모듈에서 "Red Hat Enterprise Linux 9-OpenSSL FIPS 제공자" 를 검색하여 볼 수 있습니다.

시작 중에 컨테이너의 IBM MQ 는 컨테이너가 시작되는 운영 체제가 FIPS를 준수하는지 여부를 발견합니다. 이 경우 다음 조치가 자동으로 수행됩니다.

큐 관리자

호스트 운영 체제가 FIPS를 준수하고 개인 키 및 인증서가 제공되는 경우 큐 관리자 속성 **SSLFIPS** 가 YES로 설정됩니다. 그렇지 않으면 **SSLFIPS** 속성이 NO로 설정됩니다.

IBM MQ 웹 서버
IBM MQ 웹 서버는 IBM MQ관리를 위한 HTTP/HTTPS 인터페이스를 제공합니다. 호스트 운영 체제가 FIPS를 준수하는 경우 웹 서버가 FIPS 준수 암호화를 사용하도록 JVM 옵션이 업데이트됩니다. FIPS를 사용할 수 있으려면 컨테이너 시작 중에 개인 키 및 인증서를 제공해야 합니다.

고유 HA
노드 간에 복제되는 데이터의 보안은 `qm.ini` 파일의 **NativeHALocalInstance** 스탠자에 의해 제어됩니다. 예를 들면, 다음과 같습니다.

```
NativeHALocalInstance:
    KeyRepository=/run/runmqserver/ha/tls/key.kdb
    CertificateLabel=NHAQM
    CipherSpec=ECDHE_RSA_AES_256_GCM_SHA384
```

FIPS가 사용 가능한 경우 **SSLFipsRequired** 속성이 스탠자에 추가되고 값은 `Yes`로 설정됩니다.

```
NativeHALocalInstance:
    KeyRepository=/run/runmqserver/ha/tls/key.kdb
    CertificateLabel=NHAQM
    CipherSpec=ECDHE_RSA_AES_256_GCM_SHA384
    SSLFipsRequired=Yes
```

컨테이너가 FIPS 지원 없이 OpenShift 클러스터에서 실행 중인 경우 큐 관리자, IBM MQ 웹 서버 및 고유 HA 컴포넌트에서 FIPS 지원이 자동으로 사용으로 설정되지 않습니다. x86-64 아키텍처만 현재 FIPS의 OpenShift 플랫폼에서 지원됩니다. Power 및 Linux for IBM Z® 아키텍처의 경우 OpenShift 는 FIPS 지원을 제공하지 않습니다. 이러한 아키텍처에 대해 IBM MQ 컴포넌트에서 FIPS 지원을 명시적으로 사용하려면 큐 관리자 YAML에서 *MQ_ENABLE_FIPS* 환경 변수를 `true` 로 설정하십시오. 다음 YAML 스니펫은 *MQ_ENABLE_FIPS* 환경 변수의 사용법을 설명합니다.

```
template:
    pod:
      containers:
        - env:
            - name: MQ_ENABLE_FIPS
              value: "true"
          name: qmgr
```

컨테이너에서 *IBM MQ* 에 대한 자동 *FIPS* 모드 대체

환경 변수 *MQ_ENABLE_FIPS* 를 사용하여 컨테이너의 IBM MQ 구성요소에 대해 FIPS 모드를 명시적으로 사용 또는 사용 안함으로 설정하십시오.

시작하기 전에

참고: AIX, Linux, and Windows에서 IBM MQ 는 IBM Crypto for C (ICC) 암호화 모듈을 통해 FIPS 140-2준수를 제공합니다. 이 모듈의 인증서가 히스토리 상태로 이동되었습니다. 고객은 IBM Crypto for C (ICC) 인증서 를 보고 NIST에서 제공하는 조언을 알고 있어야 합니다. 대체 FIPS 140-2모듈이 현재 진행 중이며 해당 상태는 프로세스 목록의 NIST CMVP 모듈에서 검색하여 볼 수 있습니다.

IBM MQ Operator 3.2.0 및 큐 관리자 컨테이너 이미지 9.4.0.0 이상은 UBI 9를 기반으로 합니다. FIPS 140-2 준수가 현재 보류 중이며 해당 상태는 프로세스 목록의 NIST CMVP 모듈에서 "Red Hat Enterprise Linux 9-OpenSSL FIPS 제공자" 를 검색하여 볼 수 있습니다.

이 태스크 정보

MQ_ENABLE_FIPS 는 다음 세 개의 값을 지원합니다.

auto
이는 기본값입니다.

호스트 운영 체제에서 FIPS를 사용하는 경우 모든 구성요소 (큐 관리자, IBM MQ 웹 서버 및 고유 HA) 가 FIPS 모드에서 실행됩니다.

호스트 운영 체제에서 FIPS를 사용하지 않는 경우 모든 구성요소가 FIPS 모드에서 실행되지 않습니다.

true
이 값은 컨테이너에서 선택된 컴포넌트에 대해 FIPS를 설정합니다.

FIPS를 준수하지 않는 호스트 운영 체제에서 컨테이너의 IBM MQ 가 실행 중인 경우에도 큐 관리자 속성 **SSLFIPS** 가 YES 로 설정됩니다. 즉, IBM MQ 큐 관리자, 웹 서버 및 고유 HA가 FIPS를 준수하지만 컨테이너의 운영 체제는 FIPS를 준수하지 않는 경우입니다.

false
이 값은 FIPS 준수를 해제합니다.

컨테이너의 IBM MQ 가 FIPS 준수 호스트 시스템에서 실행 중인 경우에도 큐 관리자 속성 **SSLFIPS** 가 NO로 설정됩니다. 그러나 IBM MQ 는 개인 키 및 인증서가 제공되는 경우 여전히 연결을 보호합니다.

JVM 옵션은 IBM MQ 웹 서버에 대해 업데이트되지 않습니다. 그러나 개인 키 및 인증서가 제공되는 경우 IBM MQ 웹 서버는 여전히 HTTPS 엔드포인트를 실행합니다.

고유 HA의 데이터 복제는 FIPS 암호화를 사용하지 않습니다.

예

다음은 큐 관리자 컴포넌트에 대해 TLS및 FIPS 사용을 설명하는 샘플 큐 관리자 YAML입니다.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  namespace: ibm-mq-fips
  name: ibm-mq-qm-ppcle
spec:
  license:
    accept: true
    license: L-EHXT-MQCRN9
    use: Production
  queueManager:
    name: PPCLEQM
    storage:
      queueManager:
        type: ephemeral
  template:
    pod:
      containers:
        - env:
            - name: MQ_ENABLE_FIPS
              value: "true"
          name: qmgr
  version: 9.4.0.0-r1
  web:
    enabled: false
  pki:
    keys:
      - name: ibm-mq-tls-certs
        secret:
          secretName: ibm-mq-tls-secret
          items:
          - tls.key
          - tls.crt
```

컨테이너의 IBM MQ 에 대한 확장성 및 성능 계획

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 에 대한 확장성 및 성능을 계획할 때 다음 옵션을 고려하십시오.

프로시저

- **스레드 및 프로세스 수 제한.**

  IBM MQ 는 스레드를 사용하여 동시성을 관리합니다. Linux에서 스레드는 프로세스로 구현되므로 최대 프로세스 수에 대해 컨테이너 플랫폼 또는 운영 체제에서 부과하는 한계에 직면할 수 있습니다. Red Hat

OpenShift Container Platform 4.11부터 컨테이너당 4096프로세스의 기본 한계가 있습니다. 이는 대부분의 시나리오에 적합하지만 큐 관리자에 대한 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 Platform에서 **ContainerRuntimeConfig** 사용자 정의 자원을 작성하여 CRI-O 매개변수를 편집할 수도 있습니다.

IBM MQ 구성에서 큐 관리자에 대한 최대 클라이언트 연결 수를 설정할 수도 있습니다. 개별 서버 연결 채널에 한계를 적용하려면 서버 연결 채널 한계 를 참조하고 전체 큐 관리자에 한계를 적용하려면 MAXCHANNELS INI 속성 을 참조하십시오.

- **볼륨 수를 제한합니다.**

  클라우드 및 컨테이너 시스템에서는 네트워크 연결 스토리지 볼륨이 일반적으로 사용됩니다. Linux 노드에 연결할 수 있는 볼륨 수에는 제한이 있습니다. 예를 들어, AWS EC2 는 VM당 30개이하의 볼륨으로 제한됩니다. Microsoft Azure 및 Google Cloud Platform과 마찬가지로 Red Hat OpenShift Container Platform 유사한 한계가 있음.

  고유 HA큐 관리자에는 세 개의 인스턴스 각각에 대해 하나의 볼륨이 필요하며 인스턴스가 노드에 분산되도록 강제 실행합니다. 그러나 인스턴스당 세 개의 볼륨 (큐 관리자 데이터, 복구 로그 및 지속 데이터) 을 사용하도록 큐 관리자를 구성할 수 있습니다.

- **IBM MQ 스케일링 기술을 사용하십시오.**

  소수의 대형 큐 관리자 대신, IBM MQ 스케일링 기술 (예: IBM MQ 균등 클러스터) 을 사용하여 동일한 구성의 다중 큐 관리자를 실행하는 것이 유용할 수 있습니다. 이는 단일 컨테이너 다시 시작의 영향 (예를 들어, 컨테이너 플랫폼 유지보수의 일부로) 이 감소되는 추가 이점을 갖습니다.

컨테이너에서 IBM MQ 에 대한 환경 준비, 설치 및 업그레이드

IBM MQ 에 대한 환경을 준비하기 위해 다양한 태스크를 수행합니다.

이 태스크 정보

IBM MQ Operator를 사용 중인 경우 운영자를 설치하여 Red Hat OpenShift Container Platform, 클러스터를 준비합니다. :NONE. 27 페이지의 『IBM MQ Operator 설치 및 업그레이드』

그렇지 않으면 자체 컨테이너 이미지를 빌드하여 컨테이너 환경을 준비합니다. 49 페이지의 『사용자 고유의 컨테이너 이미지를 빌드하여 IBM MQ 준비』의 내용을 참조하십시오.

IBM MQ Operator 설치 및 업그레이드

다양한 태스크를 수행하여 IBM MQ Operator를 설치, 설치 제거 및 업그레이드합니다.

이 태스크 정보

Red Hat OpenShift Container Platform에서 IBM MQ Operator 설치 및 업그레이드를 시작하려면 다음 주제를 참조하십시오.

프로시저

- 28 페이지의 『IBM MQ Operator에 대한 종속 항목』
- 28 페이지의 『IBM MQ Operator에서 필요로 하는 클러스터 범위 권한』
- 28 페이지의 『이미지 서명 확인』
- 29 페이지의 『IBM MQ Operator 설치』
- 38 페이지의 『IBM MQ Operator 및 큐 관리자 업그레이드』
- 48 페이지의 『IBM MQ Operator 설치 제거』

OpenShift CP4I IBM MQ Operator에 대한 종속 항목

IBM MQ Operator를 설치할 때 다른 운영자는 자동으로 설치되지 않습니다.

라이센스 사용을 추적하려면 IBM Licensing Operator를 별도로 설치해야 합니다. IBM Cloud Pak for Integration 문서에서 License Service 를 참조하십시오.

IBM Cloud Pak for Integration 라이센스를 사용하여 `QueueManager` 를 작성할 때 Keycloak의 IBM Cloud Pak for Integration 인스턴스에 싱글 사인온을 사용할지 여부를 선택할 수 있습니다. Keycloak 사용은 기본적으로 IBM Cloud Pak for Integration 라이센스에서 사용으로 설정되지만, 설치되지 않은 경우 `QueueManager` 는 올바른 종속 항목이 설치될 때까지 "차단됨" 상태가 됩니다. 종속성에 대한 자세한 정보는 29 페이지의 『IBM MQ Operator 설치』 의 내용을 참조하십시오.

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**
  - verbs: **get, delete**
- 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**
  - verbs: **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**
  - verbs: **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**
  - verbs: **get, list**

참고: IBM MQ Operator 에는 네임스페이스 범위의 권한도 필요합니다. IBM MQ Operator 가 클러스터 범위에 설치된 경우 네임스페이스 범위의 권한이 모든 네임스페이스에 있습니다.

OpenShift CP4I 이미지 서명 확인

IBM MQ Operator 및 IBM MQ 큐 관리자 컨테이너 이미지는 디지털로 서명됩니다.

이 태스크 정보

디지털 서명은 컨텐츠의 이용자가 다운로드하는 내용이 모두 신뢰할 수 있고 (예상 소스에서 시작됨) 무결성이 있는지 (예상되는 내용임) 확인하는 방법을 제공합니다.

프로시저

- IBM MQ Operator 및 IBM MQ 큐 관리자 컨테이너 이미지의 서명을 확인하십시오.
  - IBM Cloud Pak for Integration (CP4I) 16.1.0 문서에서 이미지 서명 확인 을 참조하십시오.

IBM MQ Operator 설치

IBM MQ Operator 는 OpenShift 콘솔 또는 명령행 인터페이스 (CLI) 를 사용하여 Red Hat OpenShift 에 설치할 수 있습니다.

시작하기 전에

중요사항:

- 이 주제는 독립형 사용 **전용**으로 IBM MQ Operator 를 설치하기 위한 것입니다. 하나 이상의 큐 관리자에 대해 IBM Cloud Pak for Integration또는 Keycloak SSO를 사용하려는 경우 35 페이지의 『CP4I 에서 사용할 IBM MQ Operator 설치』 의 내용을 참조하십시오.
- IBM MQ Operator를 설치하기 전에 배치 구조화 에 대한 지침을 검토하십시오.

설치가 가능한 한 원활하게 진행되도록 하려면 설치를 시작하기 전에 모든 전제조건 및 요구사항을 이해해야 합니다. 7 페이지의 『컨테이너에 IBM MQ 계획』 의 내용을 참조하십시오.

이 태스크 정보

다음 단계는 IBM MQ Operator를 설치하기 위한 일반 태스크 플로우를 나타냅니다.

1. Red Hat OpenShift Container Platform를 설치하십시오.
2. 스토리지를 구성하십시오.
3. 미러 이미지 (에어 갭 전용).
4. IBM MQ Operator 카탈로그를 추가하십시오.
5. IBM MQ Operator를 설치하십시오.
6. 인타이틀먼트 키 시크릿을 작성하십시오 (온라인 설치만 해당).
7. License Service를 배치하십시오.
8. 큐 관리자 배치.

프로시저

1. Red Hat OpenShift Container Platform 설치를 수행하십시오.

   OpenShift를 설치하는 자세한 단계는 Red Hat 소프트웨어 4.6 이상 설치를 참조하십시오.

   중요사항: 지원되는 OpenShift Container Platform버전을 설치했는지 확인하십시오. 예를 들어, IBM MQ Operator 3.2 이상을 사용하려면 OpenShift Container Platform 4.12 이상을 설치해야 합니다. 자세한 정보는 IBM Cloud Pak및 Red Hat OpenShift Container Platform 호환성을 참조하십시오.

   Red Hat OpenShift Container Platform CLI를 사용하는 모든 단계에서는 `oc login`를 사용하여 OpenShift 클러스터에 로그인해야 합니다. CLI를 설치하려면 OpenShift CLI 시작하기를 참조하십시오.

   OpenShift를 설치한 후 인타이틀먼트 키 시크릿 작성에서 작성하는 IBM 인타이틀먼트 키를 사용하여 컨테이너 소프트웨어에 대한 액세스를 확인하고 액세스할 수 있습니다.
2. 스토리지를 구성합니다.

   Red Hat OpenShift Container Platform 에서 스토리지 클래스를 정의하고 크기 조정 요구사항을 충족하도록 스토리지 구성을 설정해야 합니다.

   중요사항: IBM MQ 단일 인스턴스 및 원시 HA큐 관리자는 RWO 액세스 모드를 사용할 수 있는 반면 멀티 인스턴스 큐 관리자에는 15 페이지의 『IBM MQ Operator 의 스토리지 계획』 에 설명된 대로 RWX가 필요합니다. IBM MQ 멀티 인스턴스 큐 관리자에는 IBM MQ 에 대한 공유 파일 시스템 테스트에 대한 지시사항을 사용하여 확인할 수 있는 특정 파일 시스템 특성이 필요합니다.

알려진 준수 및 비준수 파일 시스템 목록 및 기타 제한 또는 제한사항에 대한 참고사항은 IBM MQ 파일 시스템에 대한 테스트 명령문에서 찾을 수 있습니다.

권장 스토리지 제공자는 CP4I 스토리지 고려사항 페이지에서 찾을 수 있습니다.

3. 미러 이미지 (에어 갭 전용).

   클러스터가 제한된 (에어 갭) 네트워크 환경에 있는 경우 다음 값을 사용하여 IBM MQ 이미지를 미러링해야 합니다.

   ```
   export OPERATOR_PACKAGE_NAME=ibm-mq
   export OPERATOR_VERSION=3.2.1
   ```

   미러 이미지를 작성하려면 에어 갭 클러스터에 대한 이미지 미러링을 참조하십시오.

4. IBM MQ Operator 카탈로그 소스를 추가하십시오.

   클러스터에서 연산자를 사용할 수 있도록 하는 카탈로그 소스를 추가하십시오. 31 페이지의 『IBM MQ Operator 카탈로그 소스 추가』의 내용을 참조하십시오.

5. IBM MQ Operator를 설치하십시오.

   다음 두 옵션 중 하나를 선택하십시오 (콘솔을 사용하거나 CLI 사용).

   - 옵션 1: OpenShift 콘솔을 사용하여 IBM MQ Operator 설치.
   - 옵션 2: OpenShift CLI를 사용하여 IBM MQ Operator 설치.

6. 인타이틀먼트 키 시크릿을 작성하십시오 (온라인 설치만 해당).

   IBM MQ Operator 는 라이센스 인타이틀먼트 검사를 수행하는 컨테이너 레지스트리에서 가져온 큐 관리자 이미지를 배치합니다. 이 검사에는 `docker-registry` pull 시크릿에 저장되는 인타이틀먼트 키가 필요합니다. 큐 관리자를 설치할 네임스페이스에 자격 부여 키가 아직 없는 경우 다음 지시사항에 따라 인타이틀먼트 키를 가져오고 풀 비밀 정보를 작성하십시오.

   참고: IBM MQ Advanced for Developers (Non-Warranted) 큐 관리자만 배치될 경우 인타이틀먼트 키는 필요하지 않습니다.

   OpenShift 콘솔 또는 CLI를 사용하여 인타이틀먼트 키 시크릿을 작성할 수 있습니다. 다음 예제는 CLI를 사용합니다.

   a. IBM ID에 지정된 인타이틀먼트 키를 가져옵니다. 자격이 있는 소프트웨어와 연관된 IBM ID및 비밀번호를 사용하여 MyIBM 에 로그인하십시오.

   b. **인타이틀먼트 키** 섹션에서 **키 복사**를 선택하여 인타이틀먼트 키를 클립보드에 복사하십시오.

   c. OpenShift CLI에서 다음 명령을 실행하여 `ibm-entitlement-key`라는 이미지 풀 시크릿을 작성하십시오.

   ```
   oc create secret docker-registry ibm-entitlement-key \
   --docker-server=cp.icr.io \
   --docker-username=cp \
   --docker-password=entitlement_key \
   --docker-email=user_email
   \--namespace=namespace
   ```

   여기서 *entitlement_key* 는 b단계에서 복사한 인타이틀먼트 키이고, *user_email* 은 권한이 부여된 소프트웨어와 연관된 IBM ID이며, *namespace* 는 IBM MQ Operator 를 설치한 네임스페이스입니다.

7. License Service를 배치하십시오.

   이는 큐 관리자의 라이센스 사용을 모니터링하는 데 필요합니다. License Service의 지시사항을 따르십시오.

8. 큐 관리자를 배치하십시오.

   예제 "빠른 시작" 큐 관리자 배치에 대한 지시사항은 57 페이지의 『IBM MQ Operator 를 사용하여 단순 큐 관리자 배치』의 내용을 참조하십시오.

관련 태스크

48 페이지의 『IBM MQ Operator 설치 제거』
Red Hat OpenShift 콘솔 또는 CLI를 사용하여 Red Hat OpenShift에서 IBM MQ Operator 를 설치 제거할 수 있습니다.

OpenShift *IBM MQ Operator* 카탈로그 소스 추가

IBM MQ Operator 카탈로그 소스를 OpenShift 클러스터에 추가하여 IBM MQ Operator 를 설치에 사용할 수 있도록 하십시오. 이 태스크는 업그레이드를 완료하기 전에 카탈로그 소스 수정팩을 적용하는 경우에도 필요합니다.

이 태스크 정보

운영자 카탈로그는 IBM 소프트웨어 제품을 사용하기 위해 Red Hat OpenShift Container Platform 클러스터의 API를 확장하는 데 사용할 수 있는 운영자의 색인입니다.

다음 카탈로그 소스를 사용할 수 있습니다.

옵션 1: IBM MQ Operator에 대한 특정 카탈로그 소스.

특정 IBM MQ Operator 카탈로그 소스를 사용하여 클러스터에서 소프트웨어 버전화 및 업그레이드 발생 시기를 완전히 제어할 수 있습니다. 새 IBM MQ Operator 버전은 카탈로그 소스를 업데이트한 **후에만** OpenShift 클러스터에서 사용 가능하게 됩니다. 이 프로세스는 업그레이드의 수동 제어를 효과적으로 제공하므로 운영자의 **Update approval** 설정에 수동 옵션을 사용할 필요가 없습니다. **수동** 옵션은 모든 가능한 업그레이드가 동시에 수행되도록 강제 실행하고 업그레이드를 차단할 수 있으므로 **자동** 옵션만 사용하십시오. 자세한 정보는 Red Hat OpenShift 콘솔을 사용하여 운영자 설치의 "승인 전략으로 자동 업데이트 제한" 절을 참조하십시오.

업그레이드를 완료하고 새 버전의 IBM MQ Operator 카탈로그 소스를 추가해야 하는 경우 이 옵션을 선택하십시오.

이 옵션을 사용하려면 옵션 1: IBM MQ Operator에 대한 특정 카탈로그 소스 추가로 건너뛰십시오.

옵션 2: IBM 운영자 카탈로그.

이 옵션을 사용하면 새 운영자 버전이 사용 가능하게 되고 사용자의 개입 없이 **적용됩니다** . 따라서 IBM MQ Operator의 **자동** 업그레이드를 원하고 결정적 설치가 필요하지 않은 온라인 설치의 경우에 **만** 이 옵션을 사용하십시오.

참고: 이 옵션은 개념 증명 환경에 유용할 수 있지만 **프로덕션 환경에는 적합하지 않습니다**.

이 옵션을 사용하려면 옵션 2: IBM 운영자 카탈로그 추가로 건너뛰십시오.

프로시저

- **옵션 1: IBM MQ Operator에 대한 특정 카탈로그 소스를 추가하십시오.**

  이 태스크에서는 29 페이지의 『IBM MQ Operator 설치』 의 처음 세 단계를 완료했다고 가정합니다.

  이 태스크는 클러스터 관리자가 수행해야 하며 CLI를 사용하여 수행해야 합니다.

  a) 업그레이드 전용: 업그레이드 전에 카탈로그 소스 수정팩을 적용하는 경우 다음 단계를 완료하십시오.

  - 운영자가 제대로 실행 중인지 확인하십시오.
  - 수동 승인이 필요한 보류 중인 IBM MQ Operator 업데이트가 있는 경우 이 프로시저를 시작하기 전에 해당 업데이트를 승인하십시오. 자세한 정보는 Red Hat OpenShift 콘솔을 사용하여 운영자 설치의 "승인 전략으로 자동 업데이트 제한" 을 참조하십시오.

  b) 아직 설치하지 않았거나 업데이트해야 하는 경우 GitHub에서 IBM Catalog Management 플러그인 (버전 1.6.0 이상) 을 다운로드하십시오.

  이 플러그인을 사용하면 클러스터에 대해 **oc ibm-pak** 명령을 실행할 수 있습니다.

  c) **oc login** 명령 및 사용자 인증 정보를 사용하여 클러스터에 로그인하십시오.

```
oc login openshift_url -u username -p password -n namespace
```

  d) IBM MQ Operator에 대해 다음 환경 변수를 내보내십시오.

```
export OPERATOR_PACKAGE_NAME=ibm-mq
export OPERATOR_VERSION=3.2.1
export ARCH=ARCHITECTURE
```

여기서 *ARCHITECTURE* 은 IBM MQ Operator를 배치하는 시스템의 아키텍처를 나타내며 값은 amd64, ppc64le또는 s390x입니다.

중요사항: IBM 운영자 카탈로그에서 IBM MQ Operator의 특정 카탈로그 소스로 이동하는 경우 OPERATOR_VERSION 를 IBM MQ Operator의 배치 버전으로 설정하십시오.

e) IBM MQ 운영자에 대한 파일을 다운로드하십시오.

참고: 에어 갭 설치를 완료하는 경우 " IBM MQ Operator설치" 의 "미러 이미지" 단계를 완료한 후 필요한 파일이 이미 있어야 합니다. 이 경우 32 페이지의 『8』 " IBM MQ Operator 카탈로그 소스를 클러스터에 적용" 단계로 건너뛸 수 있습니다.

```
oc ibm-pak get ${OPERATOR_PACKAGE_NAME} --version ${OPERATOR_VERSION}
```

f) IBM MQ Operator에 필요한 카탈로그 소스를 생성하십시오.

```
oc ibm-pak generate mirror-manifests ${OPERATOR_PACKAGE_NAME} icr.io --version $
{OPERATOR_VERSION}
```

g) 옵션: 카탈로그 소스를 생성하고 다른 디렉토리에 저장하십시오.

a. 카탈로그 소스를 가져오십시오.

```
cat ~/.ibm-pak/data/mirror/${OPERATOR_PACKAGE_NAME}/${OPERATOR_VERSION}/catalog-
sources.yaml
```

b. (선택사항) 파일 브라우저의 디렉토리를 탐색하여 재사용 또는 파이프라인을 위해 보존할 수 있는 파일로 이러한 아티팩트를 복사하십시오.

h) IBM MQ Operator 카탈로그 소스를 클러스터에 적용하십시오.

```
oc apply -f ~/.ibm-pak/data/mirror/${OPERATOR_PACKAGE_NAME}/${OPERATOR_VERSION}/catalog-
sources.yaml
```

i) IBM MQ Operator 카탈로그 소스가 openshift-marketplace 네임스페이스에서 작성되었는지 확인하십시오.

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

예제 출력:

```
oc get catalogsource -n openshift-marketplace

NAME                            DISPLAY                TYPE   PUBLISHER   AGE
ibmmq-operator-catalogsource    ibm-mq-3.1.3           grpc   IBM         23h
```

이제 IBM MQ Operator설치의 5단계를 완료할 준비가 되었습니다.

- **옵션 2: IBM 운영자 카탈로그를 추가하십시오.**

  중요사항: IBM MQ Operator의 **자동** 업그레이드를 원하고 결정적 설치가 필요하지 않은 온라인 설치의 경우에 **만** IBM 운영자 카탈로그를 사용하십시오. 이 옵션은 개념 증명 환경에 유용할 수 있지만 **프로덕션 환경에는 적합하지 않습니다**.

  IBM 운영자 카탈로그는 IBM 소프트웨어 제품을 사용하기 위해 Red Hat OpenShift Container Platform 클러스터의 API를 확장하는 데 사용할 수 있는 운영자의 색인입니다. OpenShift 클러스터에 카탈로그 소스를 추가하면 설치할 수 있는 연산자 목록에 IBM 연산자가 추가됩니다.

  이 태스크에서는 29 페이지의 『IBM MQ Operator 설치』의 처음 세 단계를 완료했다고 가정합니다.

  이 태스크는 CLI를 사용하거나 OpenShift 웹 콘솔을 사용하여 수행할 수 있습니다.

  CLI 사용

  1. IBM 운영자에 대한 다음 자원 정의를 컴퓨터의 로컬 파일에 복사하십시오.

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-operator-catalog
```

```
  namespace: openshift-marketplace
spec:
  displayName: IBM Operator Catalog
  publisher: IBM
  sourceType: grpc
  image: icr.io/cpopen/ibm-operator-catalog:latest
  updateStrategy:
    registryPoll:
      interval: 45m
```

2. 다음 명령을 실행하십시오. *filename*.yaml을 이전 단계에서 작성한 파일의 이름으로 바꾸십시오.

```
oc apply -f filename.yaml
```

OpenShift 웹 콘솔 사용

1. OpenShift 클러스터 관리자 신임 정보를 사용하여 OpenShift 웹 콘솔에 로그인하십시오.
2. 배너에서 더하기 ("+") 아이콘을 클릭하여 **YAML 가져오기** 대화 상자를 여십시오.

   참고: 프로젝트의 값을 선택할 필요가 없습니다. 다음 단계의 YAML 코드에는 카탈로그 소스가 올바른 프로젝트 (네임스페이스) 에 설치되었는지 확인하는 `metadata:namespace`의 올바른 값이 이미 포함되어 있습니다.
3. 다음 자원 정의를 대화 상자에 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
```

4. **작성**을 클릭하십시오.

이제 IBM MQ Operator설치의 5단계를 완료할 준비가 되었습니다.

OpenShift CP4I *OpenShift 콘솔을 사용하여 IBM MQ Operator 설치*

IBM MQ Operator 는 OperatorHub를 사용하여 Red Hat OpenShift 에 설치할 수 있습니다.

시작하기 전에

이 태스크에서는 29 페이지의 『IBM MQ Operator 설치』 의 1-4단계를 완료했다고 가정합니다.

프로시저

1. Red Hat OpenShift 클러스터 콘솔에 로그인하십시오.
2. 탐색 패널에서 **Operators** > **OperatorHub**를 클릭하십시오.
   OperatorHub 페이지가 표시됩니다.
3. **모든 항목** 필드에 "IBM MQ"을 입력하십시오.
   IBM MQ 카탈로그 항목이 표시됩니다.
4. **IBM MQ**를 선택하십시오.
   IBM MQ 창이 표시됩니다.
5. **설치**를 클릭하십시오.
   운영자 설치 페이지가 표시됩니다.
6. 다음 값을 입력하십시오.
   a) **채널** 을 선택한 버전으로 설정하십시오.

선택할 Operator 채널을 판별하려면 13 페이지의 『IBM MQ Operator에 대한 버전 지원』의 내용을 검토하십시오.

b) **설치 모드** 를 "클러스터의 특정 네임스페이스" (다음 단계에서 작성할 수 있음) 또는 클러스터 전체 범위로 설정하십시오.

연산자의 다른 버전을 다른 네임스페이스에 설치하면 문제가 발생할 수 있으므로 클러스터 전체 범위를 선택하는 것이 좋습니다. 연산자는 제어 평면의 확장이 되도록 설계되었습니다.

c) 옵션: "클러스터의 특정 네임스페이스" 를 선택한 경우 **네임스페이스** 를 연산자를 설치할 프로젝트 (네임스페이스) 값으로 설정하십시오.

참고: 콘솔을 사용하여 운영자를 설치할 때 기존 네임스페이스, 운영자가 제공하는 기본 네임스페이스를 사용하거나 새 네임스페이스를 작성할 수 있습니다. 새 이름 공간을 작성하려면 다음과 같이 이 양식에서 작성할 수 있습니다. 탐색 분할창에서 **홈** > **프로젝트**를 클릭하고 **프로젝트 작성**을 선택한 후 작성할 프로젝트 (이름 공간) 의 **이름** 을 지정하고 **작성**을 클릭하십시오.

d) **승인 전략** 을 자동으로 설정하십시오.

7. **설치** 를 클릭하고 운영자가 설치할 때까지 기다리십시오.

설치가 완료되면 확인이 제공됩니다.

설치를 확인하려면 **운영자** > **설치된 운영자**로 이동하여 **프로젝트** 드롭 다운 목록에서 프로젝트를 선택하십시오. 설치가 완료되면 운영자의 상태가 성공으로 변경됩니다.

다음에 수행할 작업

이제 인타이틀먼트 키 시크릿 작성 (29 페이지의 『IBM MQ Operator 설치』의 6단계) 을 수행할 준비가 되었습니다.

OpenShift CP4I *Red Hat OpenShift CLI*를 사용하여 *IBM MQ Operator* 설치

IBM MQ Operator 는 명령행 인터페이스 (CLI) 를 사용하여 Red Hat OpenShift 에 설치할 수 있습니다.

시작하기 전에

이 태스크에서는 29 페이지의 『IBM MQ Operator 설치』의 1-4단계를 완료했다고 가정합니다.

프로시저

1. **oc login**를 사용하여 Red Hat OpenShift 명령행 인터페이스 (CLI) 에 로그인하십시오.
2. 옵션: IBM MQ Operator에 사용할 네임스페이스를 작성하십시오.

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

CLI에서 새 네임스페이스를 작성하려면 다음 명령을 실행하십시오.

```
oc create namespace namespace_name
```

여기서 *namespace_name* 은 작성할 네임스페이스의 이름입니다.

3. OperatorHub에서 클러스터에 사용 가능한 연산자 목록을 보십시오.

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

4. IBM MQ Operator 를 검사하여 지원되는 **InstallModes** 및 사용 가능한 **Channels**를 확인하십시오.

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

5. 옵션: **OperatorGroup**을 작성하십시오.

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.

설치할 운영자가 AllNamespaces 모드를 사용하는 경우 openshift-operators 네임스페이스에 이미 적절한 **OperatorGroup** 가 있으므로 이 단계를 건너뛸 수 있습니다.

연산자가 SingleNamespace 모드를 사용하고 아직 적절한 **OperatorGroup** 가 없는 경우 다음 명령을 실행하여 작성하십시오.

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: operatorgroup_name
  namespace: namespace_name
spec:
  targetNamespaces:
  - namespace_name
EOF
```

6. 선택할 Operator 채널을 판별하려면 13 페이지의 『IBM MQ Operator에 대한 버전 지원』의 내용을 검토하십시오.
7. 운영자를 설치하십시오.

   다음 명령을 사용하여 *ibm-mq-operator-channel* 을 변경하여 설치하려는 IBM MQ Operator 버전의 채널과 일치시키고, "AllNamespaces" 모드를 사용하는 경우 *namespace_name* 을 **openshift-operators** 로 변경하거나, "SingleNamespace" 모드를 사용하는 경우 IBM MQ Operator를 배치하려는 네임스페이스로 변경하십시오.

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-mq
  namespace: namespace_name
spec:
  channel: ibm-mq-operator-channel
  installPlanApproval: Automatic
  name: ibm-mq
  source: ibm-operator-catalog
  sourceNamespace: openshift-marketplace
EOF
```

8. 몇 분 후에 운영자가 설치됩니다. 다음 명령을 실행하여 모든 구성요소가 성공 상태인지 확인하십시오.

```
oc get csv -n namespace_name | grep ibm-mq
```

   여기서 *namespace_name* 은 "AllNamespaces" 모드를 사용하는 경우 **openshift-operators** 이고, "SingleNamespace" 모드를 사용하는 경우 프로젝트 (네임스페이스) 이름입니다.

다음에 수행할 작업

이제 인타이틀먼트 키 시크릿 작성 (29 페이지의 『IBM MQ Operator 설치』의 6단계) 을 수행할 준비가 되었습니다.

OpenShift CP4I CP4I 에서 사용할 IBM MQ Operator 설치

IBM Cloud Pak for Integration (CP4I) 와 함께 사용하기 위해 OpenShift 콘솔 또는 명령행 인터페이스 (CLI) 를 통해 IBM MQ Operator 를 Red Hat OpenShift 에 설치할 수 있습니다.

시작하기 전에

중요사항:

- 이 주제는 CP4I에서 사용할 IBM MQ Operator 를 설치하거나 **만** CP4I 라이센스를 사용하여 하나 이상의 큐 관리자를 배치하려는 경우에 해당합니다. 독립형 사용을 위한 IBM MQ Operator 설치에 대한 지시사항은 29 페이지의 『IBM MQ Operator 설치』의 내용을 참조하십시오.
- IBM MQ Operator를 설치하기 전에 배치 구조화 에 대한 지침을 검토하십시오.

설치가 가능한 매끄럽게 진행되도록 하려면 설치를 시작하기 전에 모든 전제조건 및 요구사항을 이해해야 합니다. 7 페이지의 『컨테이너에 IBM MQ 계획』 의 내용을 참조하십시오.

이 태스크 정보

다음 단계는 IBM MQ Operator를 설치하기 위한 일반적인 태스크 플로우를 나타냅니다.

1. Red Hat OpenShift Container Platform를 설치하십시오.
2. 스토리지를 구성하십시오.
3. 미러 이미지 (에어 갭 전용).
4. IBM MQ Operator 카탈로그를 추가 하고 클러스터를 준비하십시오.
5. IBM MQ Operator 를 설치하십시오..
6. 인타이틀먼트 키 시크릿을 작성하십시오 (온라인 설치만 해당).
7. 선택사항: IBM Cloud Pak for Integration (CP4I) 및 해당 종속 항목을 설치하십시오.
8. License Service를 배치하십시오.
9. 큐 관리자를 배치하십시오.

프로시저

1. Red Hat OpenShift Container Platform 설치를 수행하십시오.

   OpenShift설치에 대한 자세한 단계는 Red Hat 소프트웨어 4.6 이상 설치를 참조하십시오.

   중요사항: 지원되는 OpenShift Container Platform버전을 설치했는지 확인하십시오. 예를 들어, IBM MQ Operator 3.2 이상을 사용하려면 OpenShift Container Platform 4.12 이상을 설치해야 합니다. 자세한 정보는 IBM Cloud Pak및 Red Hat OpenShift Container Platform 호환성을 참조하십시오.

   Red Hat OpenShift Container Platform CLI를 사용하는 단계의 경우 `oc login`를 사용하여 OpenShift 클러스터에 로그인해야 합니다. CLI를 설치하려면 OpenShift CLI 시작하기를 참조하십시오.

   OpenShift를 설치한 후 인타이틀먼트 키 시크릿 작성에서 작성한 IBM 인타이틀먼트 키를 사용하여 컨테이너 소프트웨어를 확인하고 액세스할 수 있습니다.
2. 스토리지를 구성합니다.

   Red Hat OpenShift Container Platform 에서 스토리지 클래스를 정의하고 크기 조정 요구사항을 충족하도록 스토리지 구성을 설정해야 합니다.

   중요사항: IBM MQ 단일 인스턴스 및 고유 HA큐 관리자는 RWO 액세스 모드를 사용할 수 있는 반면, 다중 인스턴스 큐 관리자는 15 페이지의 『IBM MQ Operator 의 스토리지 계획』 에 설명된 대로 RWX를 필요로 합니다. IBM MQ 다중 인스턴스 큐 관리자에는 IBM MQ에 대한 공유 파일 시스템 테스트의 지시사항을 사용하여 확인할 수 있는 특정 파일 시스템 특성이 필요합니다.

   알려진 준수 및 비준수 파일 시스템 목록과 기타 한계 또는 제한사항에 대한 참고사항은 IBM MQ 파일 시스템에 대한 테스트 명령문에서 찾을 수 있습니다.

   권장되는 스토리지 제공자는 CP4I 스토리지 고려사항 페이지에서 찾을 수 있습니다.
3. 미러 이미지 (에어 갭 전용).

   클러스터가 제한된 (에어 갭) 네트워크 환경에 있는 경우 IBM MQ 이미지를 미러링해야 합니다. 구성에 따라 일부 추가 구성요소를 미러링해야 할 수도 있습니다. 다음 정보를 읽은 후 필요에 따라 이미지를 미러링하십시오.

   - IBM MQ 이미지를 미러링해야 합니다. 다음 값을 사용하십시오.

     ```
     export OPERATOR_PACKAGE_NAME=ibm-mq
     export OPERATOR_VERSION=3.2.1
     ```

   - 또한 다음 **모든** 명령문이 true인 하나 이상의 큐 관리자를 배치하려는 경우 일부 추가 필수 컴포넌트를 미러링해야 합니다.

     – CP4I 라이센스를 사용 중입니다.

- IBM MQ Console 가 사용 가능합니다.
- IBM MQ Console 싱글 사인온 (SSO) 인증 및 권한 부여 (기본값) 를 위해 IBM Cloud Pak for Integration Keycloak 서비스를 사용 중입니다.

이전의 모든 명령문이 true인 경우 SSO는 Keycloak에 의해 제공됩니다. 따라서 IBM MQ Operator 카탈로그 소스의 경우와 마찬가지로 이러한 추가 필수 구성요소 각각에 대해 단계를 반복해야 합니다.

- IBM Cloud Pak foundational services
- IBM Cloud Pak for Integration
- Keycloak (Red Hat OpenShift 연산자)

미러 이미지를 작성하려면 에어 갭 클러스터의 이미지 미러링을 참조하십시오.

4. IBM MQ Operator 카탈로그 소스를 추가하십시오.

   다음 값을 사용하여 클러스터에서 IBM MQ Operator 를 사용할 수 있도록 하는 카탈로그 소스를 추가하십시오.

   ```
   export OPERATOR_PACKAGE_NAME=ibm-mq
   export OPERATOR_VERSION=3.2.1
   export ARCH=ARCHITECTURE
   ```

   여기서 *ARCHITECTURE* 는 시스템 아키텍처를 나타내며 값은 `amd64`, `ppc64le`또는 `s390x`입니다.

   다음 명령문이 **모두** true인 하나 이상의 큐 관리자를 배치할 때 몇 가지 추가 필수 컴포넌트가 있습니다.

   - CP4I 라이센스를 사용 중입니다.
   - IBM MQ Console 가 사용 가능합니다.
   - IBM MQ Console 싱글 사인온 (SSO) 인증 및 권한 부여 (기본값) 를 위해 IBM Cloud Pak for Integration Keycloak 서비스를 사용 중입니다.

   이전의 모든 명령문이 true인 경우 SSO는 Keycloak에 의해 제공됩니다. 따라서 IBM MQ Operator 카탈로그 소스의 경우와 마찬가지로 이러한 추가 필수 구성요소 각각에 대해 단계를 반복해야 합니다.

   - IBM Cloud Pak foundational services
   - IBM Cloud Pak for Integration
   - Keycloak (Red Hat OpenShift 연산자)

   클러스터에 카탈로그 소스 추가의 필수 카탈로그 소스에 대한 단계를 따르십시오.

5. IBM MQ Operator를 설치하십시오.

   다음 두 옵션 중 하나를 선택하십시오 (콘솔 사용 또는 CLI 사용).

   - 옵션 1: OpenShift 콘솔을 사용하여 IBM MQ Operator 를 설치하십시오.
   - 옵션 2: OpenShift CLI를 사용하여 IBM MQ Operator 를 설치하십시오.

6. 인타이틀먼트 키 시크릿을 작성하십시오 (온라인 설치만 해당).

   IBM MQ Operator 는 라이센스 인타이틀먼트 검사를 수행하는 컨테이너 레지스트리에서 가져온 큐 관리자 이미지를 배치합니다. 이 검사에는 `docker-registry` pull 시크릿에 저장되는 인타이틀먼트 키가 필요합니다. 큐 관리자를 설치할 네임스페이스에 아직 인타이틀먼트 키가 없는 경우 다음 지시사항에 따라 인타이틀먼트 키를 가져오고 풀 시크릿을 작성하십시오.

   참고: IBM MQ Advanced for Developers (무보증) 큐 관리자만 배치되는 경우에는 인타이틀먼트 키가 필요하지 않습니다.

   OpenShift 콘솔 또는 CLI를 사용하여 인타이틀먼트 키 시크릿을 작성할 수 있습니다. 다음 예제는 CLI를 사용합니다.

   a. IBM ID에 지정된 인타이틀먼트 키를 가져오십시오. 권한이 있는 소프트웨어와 연관된 IBM ID및 비밀번호를 사용하여 MyIBM Container Software Library 에 로그인하십시오.

   b. **인타이틀먼트 키** 섹션에서 **키 복사**를 선택하여 인타이틀먼트 키를 클립보드에 복사하십시오.

c. OpenShift CLI에서 다음 명령을 실행하여 `ibm-entitlement-key`라는 이미지 풀 시크릿을 작성하십시오.

```
oc create secret docker-registry ibm-entitlement-key \
--docker-server=cp.icr.io \
--docker-username=cp \
--docker-password=entitlement_key \
--docker-email=user_email
\--namespace=namespace
```

여기서 *entitlement_key* 는 b단계에서 복사한 인타이틀먼트 키이고, *user_email*> 은 권한이 부여된 소프트웨어와 연관된 IBM ID이며, *namespace* 는 IBM MQ Operator 를 설치한 네임스페이스입니다.

7. 옵션: CP4I 및 해당 종속 항목을 설치하십시오.

   다음 명령문이 **모두** true인 하나 이상의 큐 관리자를 배치할 때 몇 가지 추가 필수 컴포넌트가 있습니다.

   - CP4I 라이센스를 사용 중입니다.
   - IBM MQ Console 가 사용 가능합니다.
   - IBM MQ Console 싱글 사인온 (SSO) 인증 및 권한 부여 (기본값) 를 위해 CP4I Keycloak 서비스를 사용 중입니다.

   이전의 모든 명령문이 true인 경우 SSO는 Keycloak 에 의해 제공되며 다음 추가 단계를 완료해야 합니다.

   - CP4I 운영자와 동일한 설치 모드에서 IBM Cloud Pak foundational services 운영자를 설치하십시오. 지원되는 버전은 이 릴리스의 운영자 채널 버전을 참조하십시오.
   - CP4I 운영자를 설치하십시오.
   - 선택사항: 플랫폼 UI를 배치하십시오.

     a. `ibm-common-services` 네임스페이스를 작성하십시오. CLI를 통해 OpenShift 클러스터에 로그인한 경우 다음 명령을 실행하십시오.

     ```
     oc new-project ibm-common-services
     ```

     b. 플랫폼 UI를 배치하십시오.

8. License Service를 배치하십시오.

   이는 큐 관리자의 라이센스 사용을 모니터링하는 데 필요합니다. License Service의 지시사항을 따르십시오.

9. 큐 관리자를 배치하십시오.

   예제 "빠른 시작" 큐 관리자 배치에 대한 지시사항은 57 페이지의 『IBM MQ Operator 를 사용하여 단순 큐 관리자 배치』의 내용을 참조하십시오.

관련 태스크

48 페이지의 『IBM MQ Operator 설치 제거』
Red Hat OpenShift 콘솔 또는 CLI를 사용하여 Red Hat OpenShift에서 IBM MQ Operator 를 설치 제거할 수 있습니다.

Operator2.0.0 OpenShift CP4I

IBM MQ Operator 및 큐 관리자 업그레이드

IBM MQ 라이센스를 사용하는지 또는 IBM Cloud Pak for Integration (CP4I) 라이센스를 사용하는지에 따라 IBM MQ Operator의 사용자에 대한 여러 업그레이드 프로세스가 있습니다. 배치 유형에 대한 업그레이드 단계를 완료하십시오.

이 태스크 정보

IBM MQ Operator 및 큐 관리자를 업그레이드하려면 다음 단계 중 하나를 완료하십시오.

프로시저

- 옵션 1: **현재 운영자 채널에서 최신 버전으로 배치를 업그레이드하십시오.**

IBM MQ Operator 의 배치를 현재 운영자 채널의 최신 버전으로 업그레이드하려면 39 페이지의 『IBM MQ Operator 채널 최신 보안 릴리스로 업그레이드』의 내용을 참조하십시오.

- 옵션 2: **IBM MQ Operator for IBM MQ 라이센스를 업그레이드하십시오.**

  만 IBM MQ 라이센스가 사용되는 IBM MQ Operator 의 배치를 업그레이드하려면 39 페이지의 『IBM MQ Operator 업그레이드』의 내용을 참조하십시오.
- 옵션 3: **CP4I 사용자의 IBM MQ Operator 를 업그레이드하십시오.**

  IBM Cloud Pak for Integration의 사용자에 대한 IBM MQ Operator 의 배치를 업그레이드하십시오. 여기에는 CP4I 라이센스 하에 하나 이상의 큐 관리자를 배치한 경우가 포함됩니다. 44 페이지의 『CP4I 사용자를 위한 IBM MQ Operator 업그레이드』의 내용을 참조하십시오.

OpenShift *IBM MQ Operator* 업그레이드

만 IBM MQ 라이센스가 사용되는 IBM MQ Operator 의 배치를 업그레이드하십시오.

시작하기 전에

중요사항: 이 태스크는 IBM MQ Operator 및 **만** IBM MQ 라이센스의 사용자를 위한 것입니다. IBM Cloud Pak for Integration (CP4I) 사용자이거나 CP4I 라이센스를 사용하여 하나 이상의 큐 관리자를 배치한 경우 44 페이지의 『CP4I 사용자를 위한 IBM MQ Operator 업그레이드』의 내용을 참조하십시오.

이 태스크 정보

다음 단계 중 필요한 업그레이드와 일치하는 단계를 완료하십시오.

참고: IBM MQ Operator 의 버전 3.2.x 는 CD 및 SC2 릴리스 둘 다로 릴리스되었습니다.

프로시저

- 옵션 1:39 페이지의 『IBM MQ Operator 채널 최신 보안 릴리스로 업그레이드』
- 옵션 2: 41 페이지의 『2.0.x LTS IBM MQ Operator 를 3.2.x SC2/CD 채널로 업그레이드』
- 옵션 3: 41 페이지의 『CD IBM MQ Operator 를 3.2.x SC2/CD 채널로 업그레이드』

OpenShift CP4I *IBM MQ Operator* 채널 최신 보안 릴리스로 업그레이드

IBM MQ Operator 를 업그레이드하면 큐 관리자를 업그레이드할 수 있습니다.

시작하기 전에

중요사항: 이 주제는 배치의 채널에서 IBM MQ Operator 의 배치를 최신 보안 릴리스로 업그레이드하기 위한 것입니다. 배치에 적용되지 않는 경우 38 페이지의 『IBM MQ Operator 및 큐 관리자 업그레이드』에 설명된 대체 업그레이드 경로를 참조하십시오.

이 태스크 정보

먼저 카탈로그 소스를 업그레이드한 후 큐 관리자를 업그레이드합니다. 업그레이드 중인 IBM MQ Operator 를 배치하는 데 사용되는 카탈로그 소스에 따라 두 가지 옵션이 있습니다.

옵션 1: IBM MQ Operator 에 대한 특정 카탈로그 소스

카탈로그 소스를 업데이트한 **후에만** OpenShift 클러스터에서 새 IBM MQ Operator 버전을 사용할 수 있습니다. 이 프로세스는 업그레이드의 수동 제어를 효과적으로 제공하므로 운영자의 **Update approval** 설정에 **수동** 옵션을 사용할 필요가 없습니다. **수동** 옵션은 모든 가능한 업그레이드가 동시에 수행되도록 강제 실행하고 업그레이드를 차단할 수 있으므로 **자동** 옵션만 사용하십시오. 자세한 정보는 Red Hat OpenShift 콘솔을 사용하여 운영자 설치의 "승인 전략으로 자동 업데이트 제한" 절을 참조하십시오.

이 옵션을 사용하려면 IBM MQ Operator의 특정 카탈로그 소스로 업그레이드로 건너뛰십시오.

옵션 2: IBM 운영자 카탈로그

이 옵션을 사용하면 새 운영자 버전이 사용 가능하게 되고 사용자의 개입 없이 **적용됩니다** . 따라서 IBM MQ Operator의 **자동** 업그레이드를 원하고 결정적 설치가 필요하지 않은 온라인 설치에 **만** 이 옵션을 사용하십시오. 이 옵션은 개념 증명 환경에 유용할 수 있지만 **프로덕션 환경에는 적합하지 않습니다**.

이 옵션을 사용하려면 IBM 운영자 카탈로그로 업그레이드로 건너뛰십시오.

IBM 운영자 카탈로그 사용에서 IBM MQ Operator에 대한 특정 카탈로그 소스 사용으로 이동하여 업그레이드에 대한 제어를 강화하려면 42 페이지의 『IBM MQ Operator 의 특정 카탈로그 소스로 이동』의 내용을 참조하십시오.

프로시저

- **IBM MQ Operator의 특정 카탈로그 소스로 업그레이드**

  a) 최신 카탈로그 소스를 적용하십시오.

  IBM MQ Operator 카탈로그 소스 추가의 " IBM MQ Operator에 대한 특정 카탈로그 소스 추가"에 있는 지시사항을 따르십시오.

  b) IBM MQ Operator 에 대한 **업데이트 승인** 상태가 **자동**으로 설정된 경우 운영자가 업그레이드합니다. **업데이트 승인** 을 **수동**으로 설정한 경우 다음 단계에 따라 IBM MQ Operator를 업그레이드하십시오.

  a. 탐색 패널에서 **연산자 > 설치된 연산자**를 클릭하십시오.

  지정된 프로젝트에 설치된 모든 연산자가 표시됩니다.

  b. **IBM MQ Operator**를 선택하십시오.

  c. **구독** 탭으로 이동하십시오.

  d. **업그레이드 사용 가능** 을 클릭하십시오.

  e. **InstallPlan미리보기** 를 클릭하십시오.

  f. **승인** 을 클릭하여 업그레이드를 완료하십시오.

  운영자가 새 버전으로 업그레이드합니다.

  c) IBM MQ 큐 관리자를 업그레이드하십시오.

  IBM MQ 큐 관리자 업그레이드의 지시사항을 진행하십시오.

- **IBM 운영자 카탈로그로 업그레이드**

  a) IBM MQ Operator 를 새 버전으로 업그레이드하십시오.

  자동 업그레이드가 설정되어 있는 경우 새 보안 릴리스의 릴리스에서 IBM MQ Operator 가 업그레이드를 완료합니다. 자동 업그레이드가 설정되지 않은 경우 IBM MQ Operator 업그레이드를 수동으로 승인하십시오.

  - 사용 가능한 업그레이드가 있는 경우 **Upgrade Status** 은 "사용 가능한 업그레이드" 일 수 있습니다.
  - 이 경우 IBM MQ Operator를 업그레이드하는 **InstallPlan** 을 승인하는 데 사용할 수 있는 사용 가능한 제어가 있을 수 있습니다.

  b) IBM MQ 큐 관리자 업그레이드

  IBM MQ 큐 관리자 업그레이드의 지시사항을 진행하십시오.

- **IBM MQ 큐 관리자를 업그레이드하십시오.**

  IBM MQ Operator를 업그레이드한 후 IBM MQ 큐 관리자를 새 버전으로 업그레이드해야 합니다.

  다음 표에서는 각 활성 운영자 채널에 대한 IBM MQ 큐 관리자의 최신 버전을 설명합니다. 관련 버전을 사용하여 46 페이지의 『Red Hat OpenShift 를 사용하여 IBM MQ 큐 관리자 업그레이드』의 프로시저를 따르십시오.

Operator 채널	최신 IBM MQ 큐 관리자
v3.2 (SC2/CD)	9.4.0.0-r1

Operator2.0.0 OpenShift CP4I V 9.4.0 *2.0.x LTS IBM MQ Operator 를 3.2.x SC2/CD 채널로 업그레이드*

IBM MQ Operator 를 업그레이드하면 큐 관리자를 업그레이드할 수 있습니다.

시작하기 전에

중요사항:

- 이 태스크는 IBM MQ Operator 및 **만** IBM MQ 라이센스의 사용자를 위한 것입니다. IBM Cloud Pak for Integration (CP4I) 사용자이거나 CP4I 라이센스를 사용하여 하나 이상의 큐 관리자를 배치한 경우 44 페이지의 『CP4I 사용자를 위한 IBM MQ Operator 업그레이드』의 내용을 참조하십시오.
- 이 주제는 2.0.x Long Term Support (LTS) IBM MQ Operator 의 배치를 IBM MQ Operator 3.2.x 의 Support Cycle 2 (SC2) 채널로 **만**업그레이드하기 위한 것입니다. 배치에 적용되지 않는 경우 38 페이지의 『IBM MQ Operator 및 큐 관리자 업그레이드』에 설명된 대체 업그레이드 경로를 참조하십시오.

IBM MQ Operator 3.2.1 로 업그레이드하려면 Red Hat OpenShift Container Platform 4.12 이상을 실행 중이어야 합니다. 각 IBM MQ Operator 채널의 호환 가능한 버전을 확인하려면 13 페이지의 『호환 가능한 Red Hat OpenShift Container Platform 버전』의 내용을 참조하십시오. 플랫폼을 업그레이드하려면 Red Hat OpenShift 업그레이드를 참조하십시오.

프로시저

1. **미러 이미지 (에어 갭 전용).**

   IBM MQ 이미지를 미러링해야 합니다. 다음 값만 사용하여 다음 링크의 단계를 완료하십시오.

   ```
   export OPERATOR_PACKAGE_NAME=ibm-mq
   export OPERATOR_VERSION=3.2.1
   ```

   이전 설치 또는 업그레이드 중에 이미지 레지스트리에 대한 연결이 설정되어 있어야 하므로 3.5 "클러스터 구성" 섹션을 생략해야 합니다.

   링크: 에어 갭 클러스터의 이미지 미러링.
2. **IBM MQ Operator 를 3.2.1로 업그레이드하십시오.**

   44 페이지의 『Red Hat OpenShift 를 사용하여 IBM MQ Operator 업그레이드』의 내용을 참조하십시오.
3. **인스턴스를 업그레이드하십시오.**

   최신 기능 및 보안 수정사항을 수신하려면 IBM MQ 피연산자 (큐 관리자 컨테이너 이미지) 를 최신 CD 버전 (9.4.0.0-r1) 으로 업그레이드하십시오. 46 페이지의 『Red Hat OpenShift 를 사용하여 IBM MQ 큐 관리자 업그레이드』을 참조하십시오.

OpenShift CP4I V 9.4.0 *CD IBM MQ Operator 를 3.2.x SC2/CD 채널로 업그레이드*

IBM MQ Operator 를 업그레이드하면 큐 관리자를 업그레이드할 수 있습니다.

시작하기 전에

중요사항:

- 이 태스크는 IBM MQ Operator 및 **만** IBM MQ 라이센스의 사용자를 위한 것입니다. IBM Cloud Pak for Integration (CP4I) 사용자이거나 CP4I 라이센스를 사용하여 하나 이상의 큐 관리자를 배치한 경우 44 페이지의 『CP4I 사용자를 위한 IBM MQ Operator 업그레이드』의 내용을 참조하십시오.
- 이 주제는 3.2.0버전 이전의 IBM MQ Operator 의 Continuous Delivery (CD) 배치를 3.2.1 **버전으로만**업그레이드하기 위한 것입니다. 배치에 적용되지 않는 경우 38 페이지의 『IBM MQ Operator 및 큐 관리자 업그레이드』에 설명된 대체 업그레이드 경로를 참조하십시오.

IBM MQ Operator 3.2.1 로 업그레이드하려면 Red Hat OpenShift Container Platform 4.12 이상을 실행 중이어야 합니다. 각 IBM MQ Operator 채널의 호환 가능한 버전을 확인하려면 13 페이지의 『호환 가능한 Red Hat OpenShift Container Platform 버전』의 내용을 참조하십시오. 플랫폼을 업그레이드하려면 Red Hat OpenShift 업그레이드를 참조하십시오.

프로시저

1. 옵션: **현재 3.0.0이전의 CD 버전에 있는 IBM MQ Operator 를 업그레이드하십시오.**

   IBM MQ Operator 가 현재 3.0.0이전의 CD 버전인 경우, IBM MQ Operator (IBM MQ 9.3 문서) 의 현재 CD 채널로 마이그레이션의 관련 단계를 수행한 후 여기로 돌아와서 최신 CD 버전으로 업그레이드하십시오. 이 단계는 3.2.1버전으로 업그레이드하기 전에 수행해야 하는 필수 전제조건 단계입니다.

2. **미러 이미지 (에어 갭 전용).**

   IBM MQ 이미지를 미러링해야 합니다. 다음 값만 사용하여 다음 링크의 단계를 완료하십시오.

   ```
   export OPERATOR_PACKAGE_NAME=ibm-mq
   export OPERATOR_VERSION=3.2.1
   ```

   이전 설치 또는 업그레이드 중에 이미지 레지스트리에 대한 연결이 설정되어 있어야 하므로 3.5 "클러스터 구성" 섹션을 생략해야 합니다.

   링크: 에어 갭 클러스터의 이미지 미러링.

3. **IBM MQ Operator 를 3.2.1로 업그레이드하십시오.**

   44 페이지의 『Red Hat OpenShift 를 사용하여 IBM MQ Operator 업그레이드』의 내용을 참조하십시오.

4. **인스턴스를 업그레이드하십시오.**

   최신 기능 및 보안 수정사항을 수신하려면 IBM MQ 피연산자 (큐 관리자 컨테이너 이미지) 를 최신 CD 버전 (9.4.0.0-r1) 으로 업그레이드하십시오. 46 페이지의 『Red Hat OpenShift 를 사용하여 IBM MQ 큐 관리자 업그레이드』을 참조하십시오.

IBM MQ Operator 의 특정 카탈로그 소스로 이동

이전 릴리스의 IBM MQ Operator 가 설치되어 있고 IBM 운영자 카탈로그를 사용 중인 경우 특정 카탈로그 소스를 적용하는 것이 클러스터에서 소프트웨어 버전화를 완전히 제어하는 가장 효과적인 방법입니다.

시작하기 전에

중요사항: 이 태스크는 클러스터 관리자가 수행해야 합니다. OpenShift 역할 및 권한을 참조하십시오.

CLI를 사용하여 다음 단계를 완료합니다.

이 태스크 정보

IBM 운영자 카탈로그는 IBM 소프트웨어 제품을 사용하기 위해 Red Hat OpenShift Container Platform 클러스터의 API를 확장하는 데 사용할 수 있는 운영자의 색인입니다.

이 프로시저는 IBM MQ Operator의 특정 카탈로그 소스를 사용할 수 있도록 IBM 운영자 카탈로그에서 IBM MQ Operator 의 설치를 이동합니다.

프로시저

1. IBM MQ Operator 카탈로그를 추가하십시오.

   IBM MQ Operator 카탈로그 소스 추가의 " IBM MQ Operator에 대한 특정 카탈로그 소스 추가"에 있는 지시사항을 따르십시오.

2. IBM MQ Operator 카탈로그 소스가 `openshift-marketplace` 네임스페이스에서 작성되었는지 확인하십시오.

   다음 명령을 실행하십시오.

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

   예제 출력:

   ```
   oc get catalogsource -n openshift-marketplace
   NAME                              DISPLAY                TYPE   PUBLISHER   AGE
   ```

```
ibm-operator-catalog            IBM Operator Catalog   grpc   IBM          23h
ibmmq-operator-catalogsource    ibm-mq-3.1.3           grpc   IBM          23h
```

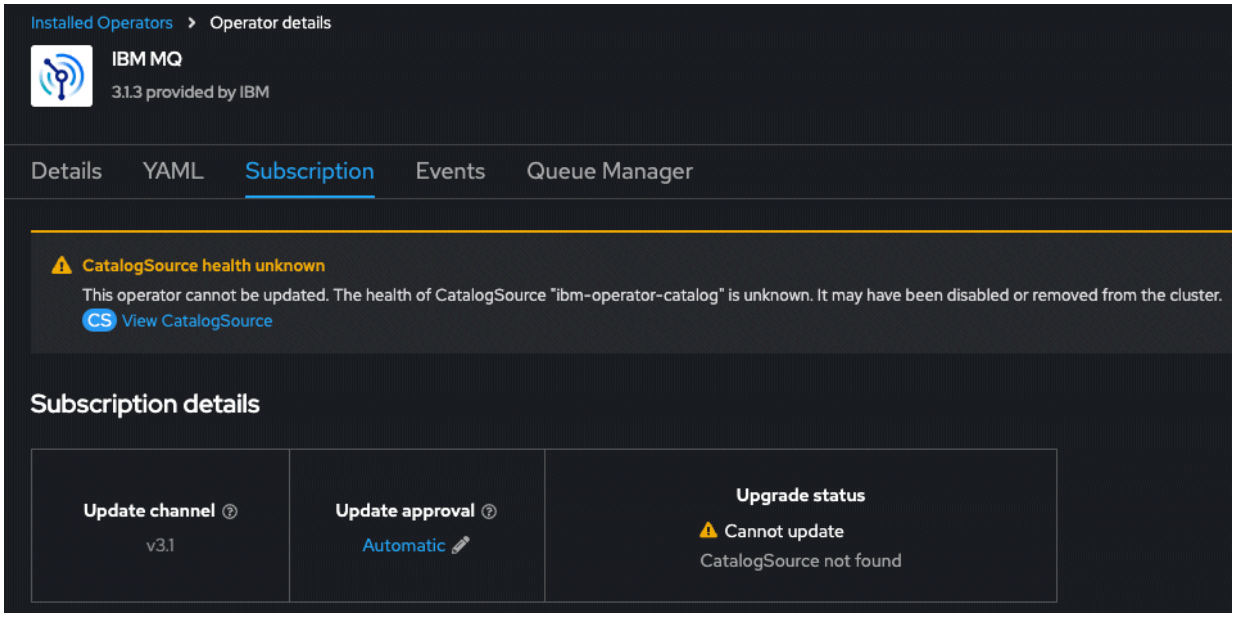
3. 옵션: IBM 운영자 카탈로그 소스를 삭제하십시오.

   경고: IBM 운영자 카탈로그를 사용하는 다른 운영자가 없다고 확신하는 경우에만 이 단계를 완료해야 합니다.

   다음 명령을 실행하십시오.

   ```
   oc delete catalogsource ibm-operator-catalog -n openshift-marketplace
   ```

   IBM MQ Operator 상태가 `CatalogSource not found`로 변경됩니다. 이는 예측된 결과입니다.



4. 새 특정 IBM MQ Operator 카탈로그 소스를 가리키도록 IBM MQ Operator 의 등록을 변경하십시오.

   a) 등록을 편집하십시오.

   다음 명령을 실행하여 *OPERATOR-NAMESPACE* 를 IBM MQ Operator의 클러스터 전체 설치를 위한 `openshift-operators` 또는 IBM MQ Operator 가 배치된 특정 네임스페이스로 대체하십시오.

   ```
   oc edit subscription ibm-mq -n OPERATOR-NAMESPACE
   ```

   b) `spec.source` 값을 `ibm-operator-catalog` 에서 42 페이지의 『1』 단계에서 작성된 카탈로그 소스의 이름으로 변경하십시오.

   예를 들면, 다음과 같습니다.

   ```
   spec:
     channel: v3.1
     installPlanApproval: Automatic
     name: ibm-mq
     source: ibm-operator-catalog # CHANGE --> ibmmq-operator-catalogsource
     sourceNamespace: openshift-marketplace
   ```

   c) 변경사항을 저장하십시오.

   IBM MQ Operator 설치는 이제 IBM MQ Operator 카탈로그 소스를 가리킵니다. IBM 운영자 카탈로그를 삭제한 경우 상태가 "CatalogSource not found" 에서 "Succeeded" 로 되돌아갑니다.

결과

이제 IBM MQ Operator 의 설치가 IBM MQ Operator의 특정 카탈로그 소스를 가리킵니다. 이를 통해 운영자에 대한 업그레이드를 완전히 제어할 수 있습니다.

OpenShift CP4I *CP4I 사용자를 위한 IBM MQ Operator 업그레이드*

IBM Cloud Pak for Integration (CP4I) 라이센스가 사용되는 IBM MQ Operator 의 배치를 업그레이드하십시오.

시작하기 전에

중요사항: 이 태스크는 CP4I 사용자를 위한 것입니다. 여기에는 CP4I 라이센스 하에 하나 이상의 큐 관리자를 배치한 경우가 포함됩니다. 적용되지 않는 경우 39 페이지의 『IBM MQ Operator 업그레이드』의 내용을 참조하십시오.

이 태스크 정보

다음 옵션 중 하나를 완료하십시오.

프로시저

- **옵션 1:** 2.0.x 의 배치 업그레이드 Long Term Support (LTS) IBM MQ Operator

  업그레이드 플랜을 생성하여 2022.2 에서 업그레이드의 단계를 따르십시오.
- **옵션 2:** IBM MQ Operator 의 3.0.x 또는 3.1.x 배치 업그레이드

  업그레이드 플랜을 생성하여 2023.4 에서 업그레이드의 단계를 따르십시오.
- **옵션 3:** IBM MQ Operator 의 다른 배치 업그레이드

  IBM MQ Operator 의 현재 CD 채널로 이주 (IBM MQ 9.3 문서)의 관련 단계를 수행한 후 여기로 돌아와서 **옵션 2**를 진행하십시오. 이는 필수 전제조건 단계입니다.

OpenShift CP4I *Red Hat OpenShift 를 사용하여 IBM MQ Operator 업그레이드*

Red Hat OpenShift 웹 콘솔 또는 CLI를 사용하여 IBM MQ Operator 를 업그레이드할 수 있습니다.

프로시저

Red Hat OpenShift를 사용하여 IBM MQ Operator 를 업그레이드하려면 다음 태스크 중 하나를 완료하십시오.

- 44 페이지의 『Red Hat OpenShift 콘솔을 사용하여 IBM MQ Operator 업그레이드』
- 45 페이지의 『Red Hat OpenShift CLI를 사용하여 IBM MQ Operator 업그레이드』

OpenShift CP4I *Red Hat OpenShift 콘솔을 사용하여 IBM MQ Operator 업그레이드*

IBM MQ Operator는 OperatorHub를 사용하여 업그레이드할 수 있습니다.

시작하기 전에

참고: IBM MQ Operator 의 최신 CD 버전은 3.2.1이며 SC2 및 CD 버전입니다. 최신 IBM MQ Operator 릴리스 정보는 IBM MQ Operator 의 릴리스 히스토리의 내용을 참조하십시오.

Red Hat OpenShift 클러스터 콘솔에 로그인하십시오.

프로시저

1. 13 페이지의 『IBM MQ Operator에 대한 버전 지원』를 검토하여 업그레이드하는 Operator를 판별하십시오.
2. 최신 카탈로그 소스를 적용하십시오.

   `ibm-operator-catalog`대신 IBM MQ Operator에 대한 특정 카탈로그 소스를 사용하는 경우 새 IBM MQ 버전에 대한 카탈로그 소스를 적용해야 합니다.

   IBM 운영자 카탈로그 사용에서 IBM MQ Operator에 대한 특정 카탈로그 소스 사용으로 이동하고 업그레이드에 대한 제어를 강화하려면 45 페이지의 『3』 단계를 완료하기 위해 돌아가기 전에 42 페이지의 『IBM MQ Operator 의 특정 카탈로그 소스로 이동』 의 단계를 참조하십시오.

   IBM 운영자 카탈로그를 사용하는 경우 (일부 온라인 설치만 해당) 45 페이지의 『3』 단계로 진행하십시오.

31 페이지의 『IBM MQ Operator 카탈로그 소스 추가』의 지시사항을 수행하십시오.

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) 탐색 패널에서 **연산자** > **설치된 연산자**를 클릭하십시오.
      지정된 프로젝트에 설치된 모든 연산자가 표시됩니다.
   b) **IBM MQ Operator**를 선택하십시오.
   c) **구독** 탭으로 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. 13 페이지의 『IBM MQ Operator에 대한 버전 지원』의 내용을 참조하십시오.

OpenShift CP4I *Red Hat OpenShift CLI*를 사용하여 *IBM MQ Operator* 업그레이드

명령행에서 IBM MQ Operator을(를) 업그레이드할 수 있습니다.

시작하기 전에

참고: IBM MQ Operator 의 최신 CD 버전은 3.2.1이며 SC2 및 CD 버전입니다. 최신 IBM MQ Operator 릴리스 정보는 IBM MQ Operator 의 릴리스 히스토리의 내용을 참조하십시오.

oc login을 사용하여 클러스터에 로그인합니다.

프로시저

1. 13 페이지의 『IBM MQ Operator에 대한 버전 지원』를 검토하여 업그레이드하는 Operator를 판별하십시오.
2. 최신 카탈로그 소스를 적용하십시오.

   `ibm-operator-catalog`대신 IBM MQ Operator에 대한 특정 카탈로그 소스를 사용하는 경우 새 IBM MQ 버전에 대한 카탈로그 소스를 적용해야 합니다.

   IBM 운영자 카탈로그 사용에서 IBM MQ Operator에 대한 특정 카탈로그 소스 사용으로 이동하고 업그레이드에 대한 제어를 강화하려면 45 페이지의 『3』 단계를 완료하기 위해 돌아가기 전에 42 페이지의 『IBM MQ Operator 의 특정 카탈로그 소스로 이동』의 단계를 참조하십시오.

   IBM 운영자 카탈로그를 사용하는 경우 (일부 온라인 설치만 해당) 45 페이지의 『3』 단계로 진행하십시오.

   31 페이지의 『IBM MQ Operator 카탈로그 소스 추가』의 지시사항을 수행하십시오.
3. IBM MQ Operator를 업그레이드하십시오. 새로운 주/부 IBM MQ Operator 버전이 새 구독 채널을 통해 전달됩니다. 운영자를 새 주 버전 또는 부 버전으로 업그레이드하려면 IBM MQ Operator 등록에서 선택한 채널을 업데이트해야 합니다.
   a) 필수 IBM MQ Operator 업그레이드 채널이 사용 가능한지 확인하십시오.

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

   b) `Subscription`을(를) 패치하여 원하는 업데이트 채널로 이동합니다. 여기서 *vX.Y*는 이전 단계에서 식별된 원하는 업데이트 채널입니다.

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

OpenShift CP4I *Red Hat OpenShift 를 사용하여 IBM MQ 큐 관리자 업그레이드*

시작하기 전에

IBM MQ 큐 관리자를 업그레이드하는 프로세스의 일부로 IBM Cloud Pak for Integration 문서에서 이 토픽으로 송신되었을 수 있습니다.

프로시저

Red Hat OpenShift를 사용하여 IBM MQ 큐 관리자를 업그레이드하려면 다음 태스크 중 하나를 완료하십시오.

- 46 페이지의 『Red Hat OpenShift 콘솔을 사용하여 IBM MQ 큐 관리자 업그레이드』
- 47 페이지의 『Red Hat OpenShift CLI를 사용하여 IBM MQ 큐 관리자 업그레이드』
- 47 페이지의 『플랫폼 UI를 사용하여 Red Hat OpenShift 에서 IBM MQ 큐 관리자 업그레이드』

다음에 수행할 작업

IBM Cloud Pak for Integration 업그레이드를 완료하려면 IBM Cloud Pak for Integration 문서로 돌아가야 합니다.

OpenShift CP4I *Red Hat OpenShift 콘솔을 사용하여 IBM MQ 큐 관리자 업그레이드*

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

시작하기 전에

참고: IBM MQ 큐 관리자의 최신 버전은 9.4.0.0-r1이며 SC2 및 CD 버전입니다. 최신 IBM MQ 큐 관리자 릴리스 정보는 IBM MQ Operator 에서 사용할 큐 관리자 컨테이너 이미지의 릴리스 히스토리의 내용을 참조하십시오.

- Red Hat OpenShift 클러스터 웹 콘솔에 로그인하십시오.
- IBM MQ Operator가 원하는 업데이트 채널을 사용하고 있는지 확인하십시오. 44 페이지의 『Red Hat OpenShift 를 사용하여 IBM MQ Operator 업그레이드』을 참조하십시오.

에어 갭 환경에서 큐 관리자를 업그레이드하려면 먼저 CD IBM MQ Operator 를 3.2.x SC2/CD 채널로 업그레이드의 에어 갭 특정 단계를 통해 최신 IBM Cloud Pak for Integration 이미지를 미러링해야 합니다.

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 컨테이너 이미지의 맵핑은 7 페이지의 『IBM MQ Operator 에서 사용할 큐 관리자 컨테이너 이미지의 릴리스 히스토리』 의 내용을 참조하십시오.
7. 업데이트된 큐 관리자 YAML을 저장하십시오.

OpenShift CP4I *Red Hat OpenShift CLI*를 사용하여 *IBM MQ* 큐 관리자 업그레이드
IBM MQ Operator를 사용하여 배치된 IBM MQ 큐 관리자는 Red Hat OpenShift에서 명령행을 사용하여 업그레이드할 수 있습니다.

시작하기 전에

참고: IBM MQ 큐 관리자의 최신 버전은 9.4.0.0-r1이며 SC2 및 CD 버전입니다. 최신 IBM MQ 큐 관리자 릴리스 정보는 IBM MQ Operator 에서 사용할 큐 관리자 컨테이너 이미지의 릴리스 히스토리의 내용을 참조하십시오.

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. 38 페이지의 『IBM MQ Operator 및 큐 관리자 업그레이드』 을 참조하십시오.

에어 갭 환경에서 큐 관리자를 업그레이드하려면 먼저 CD IBM MQ Operator 를 3.2.x SC2/CD 채널로 업그레이드의 에어 갭 특정 단계를 통해 최신 IBM Cloud Pak for Integration 이미지를 미러링해야 합니다.

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 맵핑은 13 페이지의 『IBM MQ Operator에 대한 버전 지원』 의 내용을 참조하십시오.

다음 명령을 사용하십시오.

```
oc edit queuemanager my_qmgr
```

여기서 *my_qmgr*은 업그레이드할 QueueManager 자원의 이름입니다.

CP4I 플랫폼 *UI*를 사용하여 *Red Hat OpenShift* 에서 *IBM MQ* 큐 관리자 업그레이드
IBM MQ Operator를 사용하여 배치된 IBM MQ 큐 관리자는 Red Hat OpenShift에서 IBM Cloud Pak for Integration Platform UI를 사용하여 업그레이드할 수 있습니다.

시작하기 전에

참고: IBM MQ 큐 관리자의 최신 버전은 9.4.0.0-r1이며 SC2 및 CD 버전입니다. 최신 IBM MQ 큐 관리자 릴리스 정보는 IBM MQ Operator 에서 사용할 큐 관리자 컨테이너 이미지의 릴리스 히스토리의 내용을 참조하십시오.

- 업그레이드할 큐 관리자를 포함하는 네임스페이스에서 IBM Cloud Pak for Integration Platform UI에 로그인하십시오.
- IBM MQ Operator가 원하는 업데이트 채널을 사용하고 있는지 확인하십시오. 38 페이지의 『IBM MQ Operator 및 큐 관리자 업그레이드』 의 내용을 참조하십시오.

에어 갭 환경에서 큐 관리자를 업그레이드하려면 먼저 CD IBM MQ Operator 를 3.2.x SC2/CD 채널로 업그레이드의 에어 갭 특정 단계를 통해 최신 IBM Cloud Pak for Integration 이미지를 미러링해야 합니다.

프로시저

1. IBM Cloud Pak for Integration Platform UI 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.

IBM MQ Operator 설치 제거

Red Hat OpenShift 콘솔 또는 CLI를 사용하여 Red Hat OpenShift에서 IBM MQ Operator 를 설치 제거할 수 있습니다.

프로시저

- 옵션 1: OpenShift 콘솔을 사용하여 IBM MQ Operator 를 설치 제거하십시오.

  참고: IBM MQ Operator 가 클러스터의 모든 프로젝트/네임스페이스에 설치된 경우 큐 관리자를 삭제하려는 각 프로젝트에 대해 다음 프로시저의 2-6단계를 반복하십시오.

  a) Red Hat OpenShift Container Platform 클러스터 관리 신임 정보를 사용하여 Red Hat OpenShift Container Platform 웹 콘솔에 로그인하십시오.

  b) **프로젝트** 를 IBM MQ Operator를 설치 제거할 네임스페이스로 변경하십시오. **프로젝트** 드롭 다운 목록에서 네임스페이스를 선택하십시오.

  c) 탐색 분할창에서 **운영자** > **설치된 운영자**를 클릭하십시오.

  d) **IBM MQ** Operator를 클릭하십시오.

  e) **큐 관리자** 탭을 클릭하여 이 IBM MQ Operator에서 관리하는 큐 관리자를 보십시오.

  f) 하나 이상의 큐 관리자를 삭제하십시오.

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

  g) 옵션: 해당되는 경우 큐 관리자를 삭제할 각 프로젝트에 대해 2-6단계를 반복하십시오.

  h) **Operator** > **설치된 Operator**로 되돌아가십시오.

  i) **IBM MQ** Operator 옆에 있는, 점 세 개로 표시된 메뉴를 클릭하고 **Operator 설치 제거**를 선택하십시오.

- 옵션 2: OpenShift CLI를 사용하여 IBM MQ Operator 설치 제거

  a) `oc login`를 사용하여 Red Hat OpenShift 클러스터에 로그인하십시오.

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

  a. 설치 제거할 IBM MQ Operator 를 포함하는 프로젝트에 있는지 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. Common Services를 사용하고 있는 다른 항목이 없는 경우에는 Common Services Operator를 설치 제거하고 Operator 그룹을 삭제할 수도 있습니다.

i) IBM Cloud Pak foundational services 제품 문서의 기본 서비스 설치 제거 에 있는 지시사항에 따라 공통 서비스 운영자를 설치 제거하십시오.

ii) Operator 그룹을 보십시오.

```
oc get operatorgroup
```

iii) Operator 그룹을 삭제하십시오.

```
oc delete OperatorGroup operator_group_name
```

c) 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. 선택사항: 다른 어떤 것도 공통 서비스를 사용하지 않는 경우 공통 서비스 운영자를 설치 제거할 수 있습니다. 이를 수행하려면 IBM Cloud Pak foundational services 제품 문서의 기본 서비스 설치 제거 에 있는 지시사항을 따르십시오.

사용자 고유의 컨테이너 이미지를 빌드하여 IBM MQ 준비

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 IBM MQ Operator를 대신 사용할 수 있는지 여부를 고려하십시오. :NONE. 8 페이지의 『컨테이너에서 IBM MQ를 사용하는 방법 선택』

이 태스크 정보

프로시저

- 50 페이지의 『사용자 고유의 큐 관리자 이미지 빌드 시 일반 고려사항』
- 50 페이지의 『샘플 IBM MQ 큐 관리자 컨테이너 이미지 빌드』
- 53 페이지의 『별도의 컨테이너에서 로컬 바인딩 애플리케이션 실행』
- IBM MQ 샘플 Helm 차트를 검토하십시오.

사용자 고유의 큐 관리자 이미지 빌드 시 일반 고려사항

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.

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/mqm`을 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 해결하려면, 컨테이너가 시작될 때 **crtmqdir** 명령을 사용하여 처음 실행되는 `/var/mqm`을 채우십시오.

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 사용자에 대해 어떤 가정도 하지 않습니다.

샘플 IBM MQ 큐 관리자 컨테이너 이미지 빌드

이 정보를 사용하여 컨테이너에서 IBM MQ 큐 관리자를 실행하기 위한 샘플 컨테이터 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 제공되는 `/var/mqm`의 컨텐츠와 함께 이 이미지를 해당 파일 시스템으로 사용하여 컨테이너를 실행합니다.

프로시저

- 컨테이너에서 IBM MQ 큐 관리자 실행을 위한 샘플 컨테이너 이미지를 빌드하는 방법에 대한 정보는 다음 하위 주제를 참조하십시오.
  - 51 페이지의 『샘플 기본 IBM MQ 큐 관리자 이미지 빌드』
  - 51 페이지의 『샘플 구성된 IBM MQ 큐 관리자 이미지 빌드』

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 저장소에서 제공되는 make 파일을 사용하여 프로덕션 컨테이너 이미지를 빌드하십시오.

  GitHub에 있는 Building a container image의 지시사항을 따르십시오.
- 옵션: Red Hat OpenShift Container Platform "제한된" 보안 컨텍스트 제한조건 (SCC) 을 사용하여 보안 액세스를 구성하려는 경우 IBM MQ 비설치 이미지 중 하나를 사용하십시오.

  이러한 이미지를 다운로드하기 위한 링크는 IBM MQ 다운로드의 컨테이너 섹션에서 사용 가능합니다.

결과

IBM MQ가 있는 기본 컨테이너 이미지가 설치되어야 합니다.

이제 샘플 구성 IBM MQ 큐 관리자 이미지를 빌드할 준비가 되었습니다.

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

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 MQ의 현재 버전에서는 "No-Install" IBM MQ 패키지를 사용할 때 더 이상 `mqm` 사용자 또는 `mqm` 그룹이 없습니다.

프로시저

1. 새 디렉토리를 작성하고, 다음 컨텐츠를 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 컨텐츠를 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(코드는 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 라이센스의 허용을 수신 확인합니다. 또한 해당 라이센스를 보기 위해 LICENSE 변수를 view로 설정할 수도 있습니다.

IBM MQ 라이센스에 대한 자세한 정보는 IBM MQ 라이센스 정보 를 참조하십시오.

두 번째 env 매개변수
사용 중인 큐 관리자 이름을 설정합니다.

Volume 매개변수
MQ가 `/var/mqm`에 기록하는 모든 내용이 실제로 호스트의 `/var/example`에 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 맵핑합니다. 컨테이너는 기본적으로 자체 내부 IP 주소를 갖고 실행하며, 이것은 노출시키려는 임의의 포트를 특정하게 맵핑해야 함을 의미합니다.

이 예에서는 호스트의 포트 1414를 컨테이너의 포트 1414에 맵핑함을 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.

별도의 컨테이너에서 로컬 바인딩 애플리케이션 실행

컨테이너 간 프로세스 네임스페이스 공유를 사용하여 IBM MQ 큐 관리자와 별도의 컨테이너에서 IBM MQ 에 대한 로컬 바인딩 연결이 필요한 애플리케이션을 실행할 수 있습니다.

이 태스크 정보

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.

다음 예제는 샘플 IBM MQ 컨테이너 이미지를 사용합니다. Github에서 이 이미지에 대한 세부사항을 찾을 수 있습니다.

프로시저

1. 다음 명령을 발행하여 볼륨으로 사용할 임시 디렉토리를 작성하십시오.

```
mkdir /tmp/dockerVolume
```

2. 다음 명령을 발행하여 `sharedNamespace`라는 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. `secondaryContainer`라는 두 번째 컨테이너를 시작하고, `ibmcom/mq`를 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 **dspmq** 명령을 다음과 같이 실행하여 두 큐 관리자의 상태를 확인하십시오.

```
docker exec secondaryContainer dspmq
```

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 **dspmq**, **amqsput**, **amqsget** 및 **runmqsc**와 같은 명령을 `QM1` 큐 관리자에 대한 로컬 바인딩으로 성공적으로 실행할 수 있습니다.

예상한 결과를 얻지 못한 경우 자세한 정보는 54 페이지의 『네임스페이스 애플리케이션 문제점 해결』의 내용을 참조하십시오.

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 53 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]
lpiObtainQMDetails --> 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 dltmqm QM1
AMQ7002: An error occurred manipulating a file.
```

```
docker exec container strmqweb
<file path>/mqrc.c[1152]
lpiObtainQMDetails --> 545261715
```

MQ Adv. 고유 컨테이너를 작성하는 경우 고유 HA 그룹 작성

원시 HA 그룹을 작성하려면 세 개의 큐 관리자를 작성, 구성 및 시작해야 합니다.

이 태스크 정보

원시 HA 솔루션을 작성하기 위해 권장되는 방법은 IBM MQ 연산자를 사용하는 것입니다 (원시 HA참조). 또는 사용자 고유의 컨테이너를 작성하는 경우 다음 지시사항을 따를 수 있습니다.

고유 HA 그룹을 작성하려면 해당 로그 유형이 `log replication`로 설정된 세 개의 노드에서 세 개의 큐 관리자를 작성합니다. 그런 다음 각 큐 관리자의 `qm.ini` 파일을 편집하여 세 개의 노드 각각에 대한 연결 세부사항을 추가하여 로그 데이터를 서로 복제할 수 있습니다.

그런 다음 세 개의 모든 인스턴스가 서로 통신할 수 있는지 확인하고 활성 인스턴스가 될 인스턴스와 복제본이 될 인스턴스를 판별할 수 있도록 세 개의 큐 관리자를 모두 시작해야 합니다.

참고: Kubernetes 또는 Red Hat OpenShift를 실행 중인 경우에만 이러한 방식으로 고유 컨테이너에서 고유 HA 그룹을 작성할 수 있습니다.

프로시저

1. 세 개의 각 노드에서 로그 복제본의 로그 유형을 지정하고 각 로그 인스턴스에 고유한 이름을 제공하여 큐 관리자를 작성하십시오. 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 관리자가 성공적으로 작성되면 `NativeHALocalInstance` 라는 추가 스탠자가 큐 관리자 구성 파일 `qm.ini`에 추가됩니다. `Name` 속성이 제공된 인스턴스 이름을 지정하는 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 스템 형식으로 제공됩니다. 즉, 전체 경로 및 확장자가 없는 파일 이름을 포함합니다. `KeyRepository` 스탠자 속성이 생략되면 로그 복제 데이터가 일반 텍스트로 인스턴스 간에 교환됩니다.

   CertificateLabel
   로그 복제 트래픽의 보호에 사용할 디지털 인증서를 식별하는 인증서 레이블입니다. `KeyRepository` 가 제공되었지만 `CertificateLabel` 가 생략된 경우 기본값 `ibmwebspheremq`*`queue_manager`* 가 사용됩니다.

   CipherSpec
   로그 복제 트래픽을 보호하는 데 사용할 MQ CipherSpec입니다. 이 스탠자 속성이 제공되면 `KeyRepository` 도 제공해야 합니다. `KeyRepository` 가 제공되었지만 `CipherSpec` 가 생략된 경우 기본값 `ANY` 가 사용됩니다.

   LocalAddress
   로그 복제 트래픽을 허용하는 로컬 네트워크 인터페이스 주소입니다. 이 스탠자 속성이 제공되면 "[addr][(port)]" 형식을 사용하여 로컬 네트워크 인터페이스 및/또는 포트를 식별합니다. 네트워크 주소는 호스트 이름, IPv4 점분리 10진수또는 IPv6 16진형식으로 지정할 수 있습니다. 이 속성이 생략되면 큐 관리자가 모든 네트워크 인터페이스에 바인드하려고 시도하고 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(밀리세컨드)를 정의합니다. 올바른 재시도 간격 범위는 500(0.5초)에서 120000(2분)까지입니다. 이 속성을 생략하면 실패한 복제 링크를 재시도하기 전에 2 x `HeartbeatInterval`의 복제본 대기가 대기합니다.

3. 각 큐 관리자의 `qm.ini` 파일을 편집하고 연결 세부사항을 추가하십시오. 원시 HA 그룹 (로컬 인스턴스 포함)의 각 큐 관리자 인스턴스에 대해 하나씩 세 개의 `NativeHAInstance` 스탠자를 추가합니다. 다음 속성을 추가하십시오.

   이름
   큐 관리자 인스턴스를 작성할 때 사용한 인스턴스 이름을 지정하십시오.

   ReplicationAddress
   인스턴스의 호스트 이름, IPv4 점분리 10진수또는 IPv6 16진형식 주소를 지정하십시오. 주소를 호스트 이름, IPv4 점분리 10진수또는 IPv6 16진형식 주소로 지정할 수 있습니다. 복제 주소는 그룹의 각 인스턴스에서 분석 가능하고 라우트 가능해야 합니다. 로그 복제에 사용할 포트 번호는 대괄호 안에 지정해야 합니다. 예를 들어, 다음과 같습니다.

   ```
   ReplicationAddress=host1.example.com(4444)
   ```

   참고: `NativeHAInstance` 스탠자는 모든 인스턴스에서 동일하며 자동 구성을 사용하여 제공될 수 있습니다 (**`crtmqm -ii`**).

4. 세 개의 각 인스턴스를 시작하십시오.

   ```
   strmqm QMgrName
   ```

   인스턴스가 시작되면 세 개의 모든 인스턴스가 실행 중인지 확인하기 위해 통신한 후 세 개의 인스턴스 중 활성 인스턴스를 결정하고 다른 두 개의 인스턴스는 계속 복제본으로 실행됩니다.

예

다음 예제는 세 인스턴스 중 하나에 대해 필수 고유 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)
```

컨테이너에서 큐 관리자 배치 및 구성

다양한 태스크를 수행하여 IBM MQ 큐 관리자를 배치하고 구성합니다.

이 태스크 정보

큐 관리자 배치 및 구성을 시작하려면 다음 주제를 참조하십시오.

프로시저

- 57 페이지의 『IBM MQ Operator를 사용하여 큐 관리자 배치 및 구성』
- 95 페이지의 『Helm 을 사용하여 큐 관리자 배치 및 구성』

OpenShift IBM MQ Operator를 사용하여 큐 관리자 배치 및 구성

구성 예: HA 구성, OpenShift 클러스터 외부에서 연결, CP4i 대시보드와 통합, Instana 추적과 통합, 사용자 정의 MQSC및 INI 파일로 이미지 빌드, 사용자 정의 어노테이션 및 레이블 추가.

이 태스크 정보

프로시저

- 60 페이지의 『큐 관리자 구성의 예』.
- 68 페이지의 『IBM MQ Operator를 사용하여 큐 관리자에 대한 고가용성 구성』.
- 76 페이지의 『Red Hat OpenShift 클러스터 외부에서 큐 관리자에 연결하도록 Route 구성』.
- 78 페이지의 『IBM Instana 추적과 IBM MQ 통합』.
- 85 페이지의 『Red Hat OpenShift CLI를 사용하여 사용자 정의 MQSC 및 INI 파일이 포함된 이미지 빌드』.
- 86 페이지의 『큐 관리자 자원에 사용자 정의 어노테이션 및 레이블 추가』.
- 87 페이지의 『런타임 웹훅 검사 사용 안함』.
- 87 페이지의 『큐 관리자 스펙에 대한 기본 값 업데이트 사용 안함』.

OpenShift CP4I IBM MQ Operator 를 사용하여 단순 큐 관리자 배치

이 예는 임시 (비지속적) 스토리지를 사용하는 "빠른 시작" 큐 관리자를 배치하고 IBM MQ 보안을 끕니다. 큐 관리자를 다시 시작해도 메시지가 지속되지 않습니다. 구성을 조정하여 많은 큐 관리자 설정을 변경할 수 있습니다.

이 태스크 정보

이 태스크는 OpenShift에 큐 관리자를 배치하기 위한 세 가지 옵션을 제공합니다.

1. OpenShift 콘솔을 사용하여 큐 관리자를 배치하십시오.
2. OpenShift CLI를 사용하여 큐 관리자를 배치하십시오.
3. IBM Cloud Pak for Integration Platform UI 를 사용하여 큐 관리자 배치.

프로시저

- **옵션 1: OpenShift 콘솔을 사용하여 큐 관리자를 배치하십시오.**

  a) 큐 관리자를 배치하십시오.

  a. Red Hat OpenShift Container Platform 클러스터 관리자 신임 정보를 사용하여 OpenShift 콘솔에 로그인하십시오.

  b. **프로젝트** 를 IBM MQ Operator를 설치한 네임스페이스로 변경하십시오. **프로젝트** 드롭 다운 목록에서 네임스페이스를 선택하십시오.

  c. 탐색 분할창에서 **운영자** > **설치된 운영자**를 클릭하십시오.

  d. 설치된 운영자 패널의 목록에서 **IBM MQ**를 찾아 클릭하십시오.

  e. **큐 관리자** 탭을 클릭하십시오.

f. **큐 관리자 작성** 단추를 클릭하십시오. 인스턴스 작성 패널이 표시되고 자원을 구성하기 위한 두 가지 메소드 (**양식 보기** 및 **YAML 보기**) 를 제공합니다. **양식 보기** 는 기본적으로 선택되어 있습니다.

b) 큐 관리자를 구성하십시오.

2단계옵션 1: **양식 보기**에서 구성하십시오.

양식 보기 는 자원 구성을 보거나 수정하는 데 사용할 수 있는 양식을 엽니다.

a. **라이센스**옆에 있는 화살표를 클릭하여 라이센스 승인 섹션을 펼치십시오.

b. 라이센스 계약에 동의하는 경우 **라이센스 동의** 를 **true** 로 설정하십시오.

c. 화살표를 클릭하여 드롭 다운 목록을 열고 라이센스를 선택하십시오. IBM MQ는 여러 다른 라이센스에서 사용 가능합니다. 유효한 라이센스에 대한 자세한 정보는 127 페이지의 『mq.ibm.com/v1beta1에 대한 라이센스 부여 참조』의 내용을 참조하십시오. 큐 관리자를 배치하려면 라이센스를 승인해야 합니다.

d. **작성**을 클릭하십시오. 이제 현재 프로젝트(네임스페이스)의 큐 관리자 목록이 표시됩니다. `QueueManager`는 `Pending` 상태에 있어야 합니다.

2단계옵션 2: **YAML 보기**에서 구성하십시오.

YAML 보기 는 `QueueManager`에 대한 예제 YAML 파일을 포함하는 편집기를 엽니다. 아래 단계에 따라 파일의 값을 업데이트하십시오.

a. `metadata.namespace` 을 프로젝트 (네임스페이스) 이름으로 변경하십시오.

b. `spec.license.license` 의 값을 사용자의 요구사항과 일치하는 라이센스 문자열로 변경하십시오. 라이센스 세부사항은 127 페이지의 『mq.ibm.com/v1beta1에 대한 라이센스 부여 참조』 의 내용을 참조하십시오.

c. 라이센스 계약에 동의하는 경우 `spec.license.accept` 를 `true` 로 변경하십시오.

d. **작성**을 클릭하십시오. 이제 현재 프로젝트(네임스페이스)의 큐 관리자 목록이 표시됩니다. `QueueManager`는 `Pending` 상태에 있어야 합니다.

c) 큐 관리자 작성을 확인하십시오.

다음 단계를 완료하여 큐 관리자를 작성했는지 확인할 수 있습니다.

a. IBM MQ Operator 를 작성한 네임스페이스에 있는지 확인하십시오.

b. **홈** 화면에서 **운영자** > **설치된 운영자**를 클릭한 후 큐 관리자를 작성한 설치된 IBM MQ Operator 를 선택하십시오.

c. **큐 관리자** 탭을 클릭하십시오. `QueueManager` 상태가 `Running`인 경우 작성이 완료됩니다.

- **옵션 2: OpenShift CLI를 사용하여 큐 관리자를 배치하십시오.**

a) `QueueManager` YAML 파일을 작성하십시오.

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

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
spec:
  version: 9.4.0.0-r1
  license:
    accept: false
    license: L-BMSF-5YDSLR
    use: NonProduction
  web:
    enabled: true
  queueManager:
    name: "QUICKSTART"
    storage:
      queueManager:
        type: ephemeral
```

중요사항: 라이센스 계약에 동의하는 경우 `accept: false` 를 `accept: true`로 변경하십시오. 라이센스에 대한 자세한 내용은 127 페이지의 『mq.ibm.com/v1beta1에 대한 라이센스 부여 참조』의 내용을 참조하십시오.

이 예에는 큐 관리자를 사용하여 배치된 웹 서버도 포함되며, IBM Cloud Pak for Integration내에서 싱글 사인온을 사용하여 웹 콘솔을 사용할 수 있습니다. 싱글 사인온이 작동하려면 먼저 다른 IBM Cloud Pak for Integration 구성요소를 설치해야 합니다. 35 페이지의 『CP4I 에서 사용할 IBM MQ Operator 설치』의 내용을 참조하십시오.

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

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart
spec:
  version: 9.4.0.0-r1
  license:
    accept: false
    license: L-EHXT-MQCRN9
  web:
    enabled: true
  queueManager:
    name: "QUICKSTART"
    storage:
      queueManager:
        type: ephemeral
```

중요: MQ 라이센스 계약에 동의하는 경우 `accept: false` 를 `accept: true`로 변경하십시오. 라이센스에 대한 자세한 내용은 127 페이지의 『mq.ibm.com/v1beta1에 대한 라이센스 부여 참조』의 내용을 참조하십시오.

b) `QueueManager` 오브젝트를 작성하십시오.

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

c) 큐 관리자 작성을 확인하십시오.
다음 단계를 완료하여 큐 관리자를 작성했는지 확인하십시오.

a. 배치의 유효성을 검증하십시오.

```
oc describe queuemanager Queue_Manager_Resource_Name
```

b. 상태를 확인하십시오.

```
oc describe queuemanager quickstart
```

- **옵션 3: IBM Cloud Pak for Integration Platform UI를 사용하여 큐 관리자를 배치하십시오.**

  플랫폼 UI를 사용하여 인스턴스 배치의 지시사항을 따르십시오.

관련 태스크

76 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.

118 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.

60 페이지의 『큐 관리자 구성의 예』
큐 관리자는 QueueManager 사용자 정의 자원의 컨텐츠를 조정하여 구성할 수 있습니다.

OpenShift CP4I 큐 관리자 구성의 예

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

이 태스크 정보

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

프로시저

- 60 페이지의 『예: MQSC 및 INI 파일 제공』
- 63 페이지의 『예: 상호 TLS 인증을 사용하여 큐 관리자 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.

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
```

예제 QueueManager -명령행 또는 Red Hat OpenShift Container Platform 웹 콘솔을 사용하여 다음 구성으로 큐 관리자를 배치합니다.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: mqsc-ini-qm
spec:
  version: 9.4.0.0-r1
  license:
    accept: false
    license: L-EHXT-MQCRN9
    use: Production
  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 MQ Advanced 라이센스 계약에 동의하는 경우 `accept: false`을(를) `accept: true`(으)로 변경하십시오. 라이센스의 세부사항은 mq.ibm.com/v1beta1에 대한 라이센스 부여 참조를 참조하십시오.

추가 정보:

- 단일 Kubernetes ConfigMap 또는 시크릿 (이 예에 표시된 대로) 또는 다중 ConfigMaps 및 시크릿을 사용하도록 큐 관리자를 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 관계없이 `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.
- 큐 관리자 팟 (Pod) 이 실행 중인 경우, IBM MQ Operator 가 변경사항을 인식하지 못하기 때문에 Kubernetes ConfigMap 에 대한 모든 변경사항이 선택되지 않습니다. ConfigMap을 변경하는 경우 (예를 들어, MQSC 명령 또는 INI 파일을 변경하는 경우), 해당 변경사항을 적용하려면 큐 관리자를 수동으로 다시 시작해야 합니다. 단일 인스턴스 큐 관리자의 경우 팟 (Pod) 을 삭제하여 필요한 다시 시작을 트리거하십시오. 원시 HA 배치의 경우, 먼저 삭제하여 대기 팟 (Pod) 을 다시 시작하십시오. 다시 실행 중 상태가 되면 활성 팟 (Pod) 을 삭제하여 다시 시작하십시오. 이 재시작 순서는 큐 관리자의 최소 중단 시간을 보장합니다.

OpenShift CP4I *OpenSSL* 를 사용하여 자체 서명된 *PKI* 작성

IBM MQ 에서는 인증을 위해 상호 TLS를 사용할 수 있습니다. 여기서 연결의 양쪽 끝은 인증서를 제공하고 인증서의 세부사항은 큐 관리자와 ID를 설정하는 데 사용됩니다. 이 주제에서는 OpenSSL 명령행 도구를 사용하여 다른 예제에서 사용할 수 있는 두 개의 인증서를 작성하여 예제 PKI (Public Key Infrastructure) 를 작성하는 방법을 제공합니다.

시작하기 전에

OpenSSL 명령행 도구가 설치되어 있는지 확인하십시오.

IBM MQ client를 설치하고 `samp/bin` 및 `bin`을 *PATH*에 추가하십시오. 다음과 같이 IBM MQ client 의 일부로 설치할 수 있는 **runmqicred**명령이 필요합니다.

- Windows Linux Windows 및 Linux의 경우: https://ibm.biz/mq94redistclients 에서 운영 체제에 맞는 IBM MQ 재배포 가능 클라이언트를 설치하십시오.
- macOS Mac의 경우: IBM MQ MacOS Toolkit를 다운로드하고 설정하십시오. https://developer.ibm.com/tutorials/mq-macos-dev/

이 태스크 정보

중요사항: 여기에 설명된 예제는 프로덕션 환경에 적합하지 않으며, 단지 신속하게 진행하기 위한 예제로 의도된 것입니다. 인증서 관리는 고급 사용자를 위한 복잡한 주제입니다. 프로덕션의 경우 회전, 취소, 키 길이, 재해 복구 등을 고려해야 합니다.

이 단계는 OpenSSL 3.1.4를 사용하여 테스트되었습니다.

프로시저

1. 내부 인증 기관에 사용할 개인 키 작성

```
openssl genpkey -algorithm rsa -pkeyopt rsa_keygen_bits:4096 -out ca.key
```

내부 인증 기관의 개인 키는 *ca.key*라는 파일에 작성됩니다. 이 파일은 안전하고 비밀로 유지되어야 합니다. 내부 인증 기관의 인증서에 서명하는 데 사용됩니다.

2. 내부 인증 기관에 대해 자체 서명된 인증서 발행

```
openssl req -x509 -new -nodes -key ca.key -sha512 -days 30 -subj "/CN=example-selfsigned-ca"
-out ca.crt
```

`-days` 는 루트 CA 인증서가 유효한 일 수를 지정합니다.

인증서는 *ca.crt*라는 파일에 작성됩니다. 이 인증서에는 내부 인증 기관에 대한 공용 정보가 포함되어 있으며 자유롭게 공유할 수 있습니다.

3. 큐 관리자에 대한 개인 키 및 인증서 작성

a) 큐 관리자에 대한 개인 키 및 인증서 서명 요청 작성

```
openssl req -new -nodes -out example-qm.csr -newkey rsa:4096 -keyout example-qm.key -subj
'/CN=example-qm'
```

개인 키는 *example-qm.key*파일에 작성되고 인증서 서명 요청은 *example-qm.csr* 파일에 작성됩니다.

b) 내부 인증 기관으로 큐 관리자 키 서명

```
openssl x509 -req -in example-qm.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out
example-qm.crt -days 7 -sha512
```

`-days` 는 인증서가 유효한 일 수를 지정합니다.

서명된 인증서는 *example-qm.crt* 라는 파일에 작성됩니다.

c) 큐 관리자 키 및 인증서를 사용하여 Kubernetes 시크릿 작성

```
oc create secret generic example-qm-tls --type="kubernetes.io/tls" --from-
file=tls.key=example-qm.key --from-file=tls.crt=example-qm.crt --from-file=ca.crt
```

example-qm-tls 라는 Kubernetes 시크릿이 작성됩니다. 이 시크릿에는 큐 관리자, 공용 인증서 및 CA 인증서에 대한 개인 키가 포함되어 있습니다.

4. 애플리케이션에 대한 개인 키 및 인증서 작성

a) 애플리케이션에 대한 개인 키 및 인증서 서명 요청 작성

```
openssl req -new -nodes -out example-app1.csr -newkey rsa:4096 -keyout example-app1.key
-subj '/CN=example-app1'
```

개인 키는 *example-app1.key*파일에 작성되고 인증서 서명 요청은 *example-app1.csr* 파일에 작성됩니다.

b) 내부 인증 기관으로 큐 관리자 키 서명

```
openssl x509 -req -in example-app1.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out
example-app1.crt -days 7 -sha512
```

`-days` 는 인증서가 유효한 일 수를 지정합니다.

서명된 인증서는 *example-app1.crt* 라는 파일에 작성됩니다.

c) 애플리케이션의 키 및 인증서를 사용하여 PKCS#12 키 저장소를 작성하십시오.

IBM MQ 는 개별 키 파일이 아닌 키 데이터베이스를 사용합니다. 컨테이너화된 큐 관리자는 시크릿에서 큐 관리자에 대한 키 데이터베이스를 작성하지만 클라이언트 애플리케이션의 경우 키 데이터베이스를 수동으로 작성해야 합니다.

```
openssl pkcs12 -export -in "example-app1.crt" -name "example-app1" -certfile "ca.crt"
-inkey "example-app1.key" -out "example-app1.p12" -passout pass:PASSWORD
```

여기서 *PASSWORD* 는 사용자가 선택한 비밀번호입니다.

키 저장소는 *example-app1.p12*라는 파일에 작성됩니다. 애플리케이션의 키 및 인증서는 CA 인증서뿐만 아니라 "example-app1" 의 "label" 또는 "friendly name" 을 사용하여 내부에 저장됩니다.

d) arm64 Apple Mac을 사용하는 경우 애플리케이션 및 CA 인증서를 결합하여 추가 파일을 구성해야 합니다.
예를 들면, 다음과 같습니다.

```
cat example-app1.crt ca.crt > example-app1-chain.crt
```

관련 태스크

63 페이지의 『예: 상호 TLS 인증을 사용하여 큐 관리자 구성』
이 예는 IBM MQ Operator를 사용하여 OpenShift Container Platform에 큐 관리자를 배치합니다. 상호 TLS는 TLS 인증서에서 큐 관리자의 ID로 맵핑하기 위해 인증에 사용됩니다.

69 페이지의 『예제: IBM MQ Operator 를 사용하여 고유 HA 구성』
이 예에서는 IBM MQ Operator를 사용하여 원시 고가용성 기능을 사용하는 큐 관리자를 OpenShift Container Platform 에 배치합니다. 상호 TLS는 TLS 인증서에서 큐 관리자의 ID로 맵핑하기 위해 인증에 사용됩니다.

74 페이지의 『IBM MQ Operator 를 사용하여 다중 인스턴스 큐 관리자 구성』
이 예는 IBM MQ Operator를 사용하여 OpenShift Container Platform 에 다중 인스턴스 큐 관리자를 배치합니다. 상호 TLS는 TLS 인증서에서 큐 관리자의 ID로 맵핑하기 위해 인증에 사용됩니다.

OpenShift CP4I Linux 예: 상호 *TLS* 인증을 사용하여 큐 관리자 구성

이 예는 IBM MQ Operator를 사용하여 OpenShift Container Platform에 큐 관리자를 배치합니다. 상호 TLS는 TLS 인증서에서 큐 관리자의 ID로 맵핑하기 위해 인증에 사용됩니다.

시작하기 전에

이 예를 완료하려면 먼저 다음 필수조건을 완료해야 합니다.

- 이 예를 위한 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 OpenShift Container Platform에 배치되는 큐 관리자를 정의하는 사용자 정의 자원 YAML을 제공합니다. TLS를 사용 가능하도록 하여 큐 관리자를 배치하는 데 필요한 추가 단계 또한 자세히 설명되어 있습니다.

프로시저

1. 61 페이지의 『OpenSSL 를 사용하여 자체 서명된 PKI 작성』에 설명된 대로 인증서 쌍을 작성하십시오.
2. MQSC 명령 및 INI 파일을 포함하는 구성 맵 작성

   MQSC 명령을 포함하는 Kubernetes ConfigMap 을 작성하여 새 큐 및 SVRCONN 채널을 작성하고 채널에 대한 액세스를 허용하는 채널 인증 레코드를 추가하십시오.

   이전에 작성한 네임스페이스 (시작하기 전에참조) 에 있는지 확인한 후 OCP웹 콘솔에서 또는 명령행을 사용하여 다음 YAML을 입력하십시오.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: example-tls-configmap
data:
  example-tls.mqsc: |
    DEFINE CHANNEL('MTLS.SVRCONN') CHLTYPE(SVRCONN) SSLCAUTH(REQUIRED)
SSLCIPH('ANY_TLS13_OR_HIGHER') REPLACE
    SET CHLAUTH('MTLS.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=*') USERSRC(NOACCESS)
ACTION(REPLACE)
    SET CHLAUTH('MTLS.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=example-app1') USERSRC(MAP)
MCAUSER('app1') ACTION(REPLACE)
    SET AUTHREC PRINCIPAL('app1') OBJTYPE(QMGR) AUTHADD(CONNECT,INQ)
    DEFINE QLOCAL('EXAMPLE.QUEUE') REPLACE
    SET AUTHREC PROFILE('EXAMPLE.QUEUE') PRINCIPAL('app1') OBJTYPE(QUEUE)
AUTHADD(BROWSE,PUT,GET,INQ)
  example-tls.ini: |
    Service:
        Name=AuthorizationService
```

```
        EntryPoints=14
        SecurityPolicy=UserExternal
```

MQSC는 *MTLS.SVRCONN* 이라는 채널 및 *EXAMPLE.QUEUE*를 큐에 지정하십시오. 채널은 *example-app1*의 "공통 이름" 으로 인증서를 제공하는 클라이언트에만 액세스할 수 있도록 구성됩니다. 이는 63 페이지의 『1』 단계에서 작성된 인증서 중 하나에서 사용되는 공통 이름입니다. 이 공용 이름을 사용하는 이 채널의 연결은 큐 관리자에 연결하고 예제 큐에 액세스할 수 있는 권한이 부여된 *app1*의 사용자 ID에 맵핑됩니다. INI 파일은 *app1* 사용자 ID가 외부 사용자 레지스트리에 존재할 필요가 없음을 의미하는 보안 정책을 사용 가능하게 합니다. 이 구성에서는 이름으로만 존재합니다.

3. 큐 관리자 배치

   다음 사용자 정의 자원 YAML을 사용하여 새 큐 관리자를 작성하십시오. 이 태스크를 시작하기 전에 작성한 네임스페이스에 있는지 확인한 후 OCP웹 콘솔에서 또는 명령행을 사용하여 다음 YAML을 입력하십시오. 올바른 라이센스가 지정되었는지 확인하고, `false`를 `true`로 변경하여 라이센스에 동의하십시오.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: exampleqm
spec:
  license:
    accept: false
    license: L-EHXT-MQCRN9
    use: Production
  queueManager:
    name: EXAMPLEQM
    mqsc:
    - configMap:
        name: example-tls-configmap
        items:
        - example-tls.mqsc
    ini:
    - configMap:
        name: example-tls-configmap
        items:
        - example-tls.ini
    storage:
      queueManager:
        type: ephemeral
  version: 9.4.0.0-r1
  pki:
    keys:
      - name: default
        secret:
          secretName: example-qm-tls
          items:
            - tls.key
            - tls.crt
            - ca.crt
```

   시크릿 *example-qm-tls* 는 63 페이지의 『1』 단계에서 작성되었으며 ConfigMap *example-tls-configmap* 은 63 페이지의 『2』 단계에서 작성되었습니다.

4. 큐 관리자가 실행 중인지 확인

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

```
oc get qmgr exampleqm
```

5. 큐 관리자에 대한 연결 테스트

   큐 관리자가 상호 TLS 통신을 위해 구성되었는지 확인하려면 65 페이지의 『랩탑에서 큐 관리자에 대한 상호 TLS 연결 테스트』의 단계를 따르십시오.

결과

축하합니다. TLS가 사용으로 설정된 큐 관리자를 성공적으로 배치했으며, 이는 큐 관리자를 인증하고 ID를 제공하기 위해 TLS 인증서에 제공된 세부사항을 사용합니다.

OpenShift CP4I Linux 랩탑에서 큐 관리자에 대한 상호 *TLS* 연결 테스트

IBM MQ Operator를 사용하여 큐 관리자를 작성한 후 큐 관리자에 연결하고 메시지를 넣고 가져와서 큐 관리자가 작동하는지 테스트할 수 있습니다. 이 태스크는 Kubernetes 클러스터 외부의 시스템 (예: 랩탑) 에서 실행하여 IBM MQ 샘플 프로그램을 사용하여 연결하는 방법을 안내합니다.

시작하기 전에

이 예를 완료하려면 먼저 다음 필수조건을 완료해야 합니다.

- IBM MQ client를 설치하십시오. 다음과 같이 IBM MQ client 의 일부로 설치할 수 있는 **amqsputc** 및 **amqsgetc** 명령이 필요합니다.
  - Windows Linux Windows 및 Linux의 경우: https://ibm.biz/mq94redistclients 에서 운영 체제에 맞는 IBM MQ 재배포 가능 클라이언트를 설치하십시오.
  - macOS Mac의 경우: IBM MQ MacOS Toolkit를 다운로드하고 설정하십시오. https://developer.ibm.com/tutorials/mq-macos-dev/
- 필요한 키 및 인증서 파일을 시스템의 디렉토리에 다운로드하고 키 저장소 비밀번호를 알고 있는지 확인하십시오. 예를 들어, 다음 파일은 61 페이지의 『OpenSSL 를 사용하여 자체 서명된 PKI 작성』 에 작성됩니다.
  - `example-app1.p12`
  - `example-app1-chain.crt` (arm64 Apple Mac를 사용하는 경우에만)
- 예를 들어, 63 페이지의 『예: 상호 TLS 인증을 사용하여 큐 관리자 구성』 의 단계를 수행하여 TLS로 구성된 큐 관리자를 OCP 클러스터에 배치하십시오.

이 태스크 정보

이 예에서는 랩탑과 같은 Kubernetes 클러스터 외부의 시스템에서 실행 중인 IBM MQ 샘플 프로그램을 사용하여 TLS로 구성된 QueueManager 에 연결하고 메시지를 넣고 가져옵니다.

프로시저

1. 큐 관리자가 실행 중인지 확인

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

   ```
   oc get qmgr exampleqm
   ```

2. 큐 관리자 호스트 이름 찾기

   자동으로 작성되는 라우트를 사용하여 OCP 클러스터 외부에서 큐 관리자의 큐 관리자 완전한 호스트 이름을 찾으려면 `exampleqm-ibm-mq-qm`명령을 사용하십시오.

   ```
   oc get route exampleqm-ibm-mq-qm --template="{{.spec.host}}"
   ```

3. IBM MQ 클라이언트 채널 정의 테이블 (CCDT) 작성

   다음 컨텐츠를 사용하여 `ccdt.json` 파일을 작성하십시오.

   ```
   {
       "channel":
       [
           {
               "name": "MTLS.SVRCONN",
               "clientConnection":
               {
                   "connection":
                   [
                       {
                           "host": "hostname from previous step",
                           "port": 443
                       }
                   ],
                   "queueManager": "EXAMPLEQM"
   ```

```
            },
            "transmissionSecurity":
            {
              "cipherSpecification": "ANY_TLS13",
              "certificateLabel": "example-app1"
            },
            "type": "clientConnection"
        }
   ]
}
```

연결은 포트 443을 사용합니다. 이는 Red Hat OpenShift Container Platform 라우터가 청취하는 포트이기 때문입니다. 트래픽은 포트 1414의 큐 관리자로 전달됩니다.

다른 채널 이름을 사용한 경우에는 이를 조정해야 합니다. 상호 TLS 예제는 *MTLS.SVRCONN* 이라는 채널을 사용합니다.

자세한 정보는 JSON 형식 CCDT 구성 을 참조하십시오.

4. 클라이언트 INI 파일을 작성하여 연결 세부사항 구성

   현재 디렉토리에 `mqclient.ini` 라는 파일을 작성하십시오. 이 파일은 **amqsputc** 및 **amqsgetc**에서 읽습니다.

   ```
   Channels:
     ChannelDefinitionDirectory=.
     ChannelDefinitionFile=ccdt.json
   SSL:
     OutboundSNI=HOSTNAME
     SSLKeyRepository=example-app1.p12
     SSLKeyRepositoryPassword=password you used when creating the p12 file
   ```

   *SSLKeyRepository*비밀번호 를 PKCS#12 파일을 작성할 때 선택한 비밀번호로 업데이트해야 합니다. 암호화된 비밀번호 사용을 포함하여 키 저장소 비밀번호를 설정하는 다른 방법이 있습니다. 자세한 정보는 IBM MQ MQI client on AIX, Linux, and Windows에 대한 키 저장소 비밀번호 제공 을 참조하십시오.

   Red Hat OpenShift Container Platform Router에서는 IBM MQ 큐 관리자로 요청을 라우팅하는 데 SNI를 사용한다는 점을 참고하십시오. *OutboundSNI= HOSTNAME* 속성은 라우터가 IBM MQ Operator에 의해 구성된 기본 라우트에 대해 작업하는 데 필요한 정보가 IBM MQ 클라이언트에 포함되도록 합니다. 자세한 정보는 76 페이지의 『Red Hat OpenShift 클러스터 외부에서 큐 관리자에 연결하도록 Route 구성』의 내용을 참조하십시오.

5. arm64 Apple Mac를 사용하는 경우 추가 환경 변수를 구성해야 합니다.

   ```
   export MQSSLTRUSTSTORE=example-app1-chain.crt
   ```

   이 파일에는 애플리케이션 및 CA 인증서를 포함하여 전체 인증서 체인이 포함되어 있습니다.

6. 큐에 메시지 넣기
   다음 명령을 실행하십시오.

   ```
   /opt/mqm/samp/bin/amqsputc EXAMPLE.QUEUE EXAMPLEQM
   ```

   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.

7. 큐에서 메시지 검색
   다음 명령을 실행하십시오.

   ```
   /opt/mqm/samp/bin/amqsgetc EXAMPLE.QUEUE EXAMPLEQM
   ```

   이전 단계에서 추가한 메시지가 이용되어 출력됩니다. 몇 초 후 명령이 종료됩니다.

결과

축하합니다. TLS가 사용으로 설정된 큐 관리자의 연결을 성공적으로 테스트했으며 클라이언트에서 큐 관리자에 안전하게 메시지를 넣고 가져올 수 있음을 표시했습니다.

OpenShift CP4I 예: 라이센스 서비스 어노테이션 사용자 정의

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

이 태스크 정보

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

예

License이(가) `L-RJON-BZFQU2(IBM Cloud Pak for Integration 2021.2.1)`로 설정되고 **Use**이(가) `NonProduction`으로 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 86 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
```

라이센스 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 라이센스로 배치됩니다.
   - IBM Cloud Pak for Integration 라이센스가 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 MQ Adv. IBM MQ Operator를 사용하여 큐 관리자에 대한 고가용성 구성

이 태스크 정보

프로시저

- 18 페이지의 『고유 HA』.
- 69 페이지의 『예제: IBM MQ Operator 를 사용하여 고유 HA 구성』.
- 74 페이지의 『IBM MQ Operator 를 사용하여 다중 인스턴스 큐 관리자 구성』.

OpenShift MQ Adv. *IBM MQ Operator*를 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-EHXT-MQCRN9
    use: Production
  queueManager:
    availability:
      type: NativeHA
  version: 9.4.0.0-r1
```

.spec.queueManager.availability.type 필드는 NativeHA(으)로 설정되어야 합니다.

.spec.queueManager.availability에서, 복제할 때 큐 관리자 인스턴스 간에 사용할 TLS 시크릿 및 암호를 구성할 수도 있습니다. 이는 강력하게 추천하며 69 페이지의 『예제: IBM MQ Operator 를 사용하여 고유 HA 구성』 에서는 단계별 안내서를 사용 가능합니다.

관련 태스크

69 페이지의 『예제: IBM MQ Operator 를 사용하여 고유 HA 구성』
이 예에서는 IBM MQ Operator를 사용하여 원시 고가용성 기능을 사용하는 큐 관리자를 OpenShift Container Platform 에 배치합니다. 상호 TLS는 TLS 인증서에서 큐 관리자의 ID로 맵핑하기 위해 인증에 사용됩니다.

OpenShift MQ Adv. Kubernetes 예제: *IBM MQ Operator* 를 사용하여 고유 *HA* 구성

이 예에서는 IBM MQ Operator를 사용하여 원시 고가용성 기능을 사용하는 큐 관리자를 OpenShift Container Platform 에 배치합니다. 상호 TLS는 TLS 인증서에서 큐 관리자의 ID로 맵핑하기 위해 인증에 사용됩니다.

시작하기 전에

이 예를 완료하려면 먼저 다음 필수조건을 완료해야 합니다.

- 이 예를 위한 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 OpenShift Container Platform에 배치되는 큐 관리자를 정의하는 사용자 정의 자원 YAML을 제공합니다. TLS를 사용 가능하도록 하여 큐 관리자를 배치하는 데 필요한 추가 단계 또한 자세히 설명되어 있습니다.

프로시저

1. 61 페이지의 『OpenSSL 를 사용하여 자체 서명된 PKI 작성』 에 설명된 대로 인증서 쌍을 작성하십시오.
2. MQSC 명령 및 INI 파일을 포함하는 구성 맵 작성

   MQSC 명령을 포함하는 Kubernetes ConfigMap 을 작성하여 새 큐 및 SVRCONN 채널을 작성하고 채널에 대한 액세스를 허용하는 채널 인증 레코드를 추가하십시오.

   이전에 작성한 네임스페이스 (시작하기 전에참조) 에 있는지 확인한 후 OCP웹 콘솔에서 또는 명령행을 사용하여 다음 YAML을 입력하십시오.

   ```
   apiVersion: v1
   kind: ConfigMap
   metadata:
     name: example-nativeha-configmap
   data:
     example-tls.mqsc: |
       DEFINE CHANNEL('MTLS.SVRCONN') CHLTYPE(SVRCONN) SSLCAUTH(REQUIRED)
   SSLCIPH('ANY_TLS13_OR_HIGHER') REPLACE
       SET CHLAUTH('MTLS.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=*') USERSRC(NOACCESS)
   ACTION(REPLACE)
       SET CHLAUTH('MTLS.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=example-app1') USERSRC(MAP)
   MCAUSER('app1') ACTION(REPLACE)
       SET AUTHREC PRINCIPAL('app1') OBJTYPE(QMGR) AUTHADD(CONNECT,INQ)
       DEFINE QLOCAL('EXAMPLE.QUEUE') REPLACE
       SET AUTHREC PROFILE('EXAMPLE.QUEUE') PRINCIPAL('app1') OBJTYPE(QUEUE)
   AUTHADD(BROWSE,PUT,GET,INQ)
     example-tls.ini: |
       Service:
           Name=AuthorizationService
           EntryPoints=14
           SecurityPolicy=UserExternal
   ```

   MQSC는 *MTLS.SVRCONN* 이라는 채널 및 *EXAMPLE.QUEUE*를 큐에 지정하십시오. 채널은 *example-app1*의 "공통 이름" 으로 인증서를 제공하는 클라이언트에만 액세스할 수 있도록 구성됩니다. 이는 69 페이지의 『1』 단계에서 작성된 인증서 중 하나에서 사용되는 공통 이름입니다. 이 공용 이름을 사용하는 이 채널의 연결은 큐 관리자에 연결하고 예제 큐에 액세스할 수 있는 권한이 부여된 *app1*의 사용자 ID에 맵핑됩니다. INI 파일은 *app1* 사용자 ID가 외부 사용자 레지스트리에 존재할 필요가 없음을 의미하는 보안 정책을 사용 가능하게 합니다. 이 구성에서는 이름으로만 존재합니다.
3. 큐 관리자 배치

   다음 사용자 정의 자원 YAML을 사용하여 새 큐 관리자를 작성하십시오. 이 태스크를 시작하기 전에 작성한 네임스페이스에 있는지 확인한 후 OCP웹 콘솔에서 또는 명령행을 사용하여 다음 YAML을 입력하십시오. 올바른 라이센스가 지정되었는지 확인하고, `false`를 `true`로 변경하여 라이센스에 동의하십시오.

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

```
license:
  accept: false
  license: L-EHXT-MQCRN9
  use: Production
queueManager:
  name: EXAMPLEQM
  availability:
    type: NativeHA
    tls:
      secretName: example-qm-tls
  mqsc:
  - configMap:
      name: example-nativeha-configmap
      items:
      - example-tls.mqsc
  ini:
  - configMap:
      name: example-nativeha-configmap
      items:
      - example-tls.ini
  storage:
    queueManager:
      type: persistent-claim
version: 9.4.0.0-r1
pki:
  keys:
    - name: default
      secret:
        secretName: example-qm-tls
        items:
          - tls.key
          - tls.crt
          - ca.crt
```

시크릿 *example-qm-tls* 는 69 페이지의 『1』 단계에서 작성되었으며 ConfigMap *example-nativeha-configmap* 은 69 페이지의 『2』 단계에서 작성되었습니다.

가용성 유형이 *NativeHA*로 설정되고 지속적 스토리지가 선택됩니다. Kubernetes 클러스터에 구성된 기본 스토리지 클래스가 사용됩니다. 기본값으로 구성된 스토리지 클래스가 없거나 다른 스토리지 클래스를 사용하려는 경우 `spec.queueManager.storage` 아래에 `defaultClass: storage_class_name`을(를) 추가하십시오.

고유 HA 큐 관리자에 있는 세 개의 팟(Pod)은 네트워크를 통해 데이터를 복제합니다. 이 링크는 기본적으로 암호화되지 않지만 이 예제에서는 트래픽을 암호화하기 위해 큐 관리자의 인증서를 사용합니다. 추가 보안을 위해 다른 인증서를 지정할 수 있습니다. 고유 HA TLS 시크릿은 특정 구조를 갖는 Kubernetes TLS 시크릿이어야 합니다 (예를 들어, 개인 키는 *tls.key*라고 함).

4. 큐 관리자가 실행 중인지 확인

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

   ```
   oc get qmgr exampleqm
   ```

5. 큐 관리자에 대한 연결 테스트

   큐 관리자가 구성되어 사용 가능한지 확인하려면 65 페이지의 『랩탑에서 큐 관리자에 대한 상호 TLS 연결 테스트』의 단계를 따르십시오.

6. 활성 팟(Pod)이 실패하도록 강제 실행

   큐 관리자의 자동 복구를 유효성 검증하기 위해 팟(Pod) 실패를 시뮬레이션합니다.

   a) 활성 및 대기 팟(Pod) 보기

   다음 명령을 실행하십시오.

   ```
   oc get pods --selector app.kubernetes.io/instance=exampleqm
   ```

   READY 필드에서 활성 팟(Pod)은 `1/1` 값을 리턴하지만 복제본 팟(Pod)은 `0/1` 값을 리턴합니다.

   b) 활성 팟(Pod) 삭제

   활성 팟(Pod)의 전체 이름을 지정하며 다음 명령을 실행하십시오.

   ```
   oc delete pod exampleqm-ibm-mq-value
   ```

c) 팟(Pod) 상태 다시 보기
다음 명령을 실행하십시오.

```
oc get pods --selector app.kubernetes.io/instance=exampleqm
```

d) 큐 관리자 상태 보기
다른 팟(Pod) 중 하나의 전체 이름을 지정하며 다음 명령을 실행하십시오.

```
oc exec -t Pod -- dspmq -o nativeha -x -m EXAMPLEQM
```

예를 들어 활성 인스턴스가 변경되었음을 상태에서 표시해야 합니다.

```
QMNAME(EXAMPLEQM) 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) ALTDATE(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
```

e) 큐 관리자에 대한 연결을 다시 테스트하십시오.
큐 관리자가 복구되었는지 확인하려면 65 페이지의 『랩탑에서 큐 관리자에 대한 상호 TLS 연결 테스트』의 단계를 따르십시오.

결과

축하합니다. 기본 고가용성 및 상호 TLS 인증을 사용하여 큐 관리자를 성공적으로 배치하고 활성 팟 (Pod) 이 실패할 때 자동으로 복구되는지 확인했습니다.

OpenShift MQ Adv. *IBM MQ* 컨테이너에 대한 원시 *HA*큐 관리자의 상태 보기

IBM MQ 컨테이너의 경우 실행 중인 팟 (Pod) 중 하나에서 **dspmq** 명령을 실행하여 고유 HA 인스턴스의 상태를 볼 수 있습니다.

이 태스크 정보

실행 중인 팟(Pod) 중 하나에서 **dspmq** 명령을 사용하여 큐 관리자 인스턴스의 운영 상태를 볼 수 있습니다. 리턴되는 정보는 인스턴스가 활성인지 또는 복제본인지에 따라 달라집니다. 활성 인스턴스에서 제공하는 정보는 결정적이며 복제본 노드의 정보는 이전 정보일 수 있습니다.

다음 조치를 수행할 수 있습니다.

- 현재 노드의 큐 관리자 인스턴스가 활성인지 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 활성 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 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 팟 (Pod) 을 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 쉘의 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 활상 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) ALTDATE(2022-01-12) ALTTIME(12.03.44)
 INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
 INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(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) ALTDATE(2022-01-12) ALTTIME(12.03.44)
 INSTANCE(inst1) ROLE(Active)  REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
 INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(No)  BACKLOG(435)
CONNINST(Yes) ALTDATE(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) ALTDATE() ALTTIME()
 INSTANCE(inst2) ROLE(Unknown) REPLADDR(9.20.123.46) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATE() ALTTIME()
 INSTANCE(inst3) ROLE(Unknown) REPLADDR(9.20.123.47) CONNACTV(No) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATE() ALTTIME()
```

인스턴스가 계속 무엇이 활성인지 그리고 무엇이 복제본인지 조정하는 경우 명령을 실행하면, 다음 상태를 수신할 수 있습니다.

```
QMNAME(BOB)              STATUS(Negotiating)
```

관련 태스크

69 페이지의 『예제: IBM MQ Operator 를 사용하여 고유 HA 구성』
이 예에서는 IBM MQ Operator를 사용하여 원시 고가용성 기능을 사용하는 큐 관리자를 OpenShift Container Platform 에 배치합니다. 상호 TLS는 TLS 인증서에서 큐 관리자의 ID로 맵핑하기 위해 인증에 사용됩니다.

관련 참조

dspmq(큐 관리자 표시) 명령

OpenShift MQ Adv. 고유 *HA*를 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 사용해서만 구성할 수 있는 몇 가지 고급 옵션이 있습니다. INI 파일 구성 방법에 대한 자세한 정보는 60 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 `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(밀리세컨드)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.

OpenShift MQ Adv. 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 절에서 원시 HA큐 관리자 종료 를 참조하십시오.

OpenShift CP4I MQ Adv. Kubernetes *IBM MQ Operator* 를 사용하여 다중 인스턴스 큐 관리자 구성

이 예는 IBM MQ Operator를 사용하여 OpenShift Container Platform 에 다중 인스턴스 큐 관리자를 배치합니다. 상호 TLS는 TLS 인증서에서 큐 관리자의 ID로 맵핑하기 위해 인증에 사용됩니다.

시작하기 전에

이 예를 완료하려면 먼저 다음 필수조건을 완료해야 합니다.

- 이 예를 위한 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 OpenShift Container Platform에 배치되는 큐 관리자를 정의하는 사용자 정의 자원 YAML을 제공합니다. TLS를 사용 가능하도록 하여 큐 관리자를 배치하는 데 필요한 추가 단계 또한 자세히 설명되어 있습니다.

프로시저

1. 적합한 스토리지 클래스 판별

   Kubernetes 클러스터의 스토리지는 다중 지속적 볼륨 액세스 모드를 사용하여 액세스할 수 있습니다. 다중 인스턴스 큐 관리자는 여러 개의 지속적 볼륨을 작성합니다 (각 큐 관리자에 대해 하나씩). 그리고 하나 이상의 공유 볼륨을 작성합니다. 다중 인스턴스 큐 관리자의 공유 볼륨은 ReadWriteMany 스토리지 클래스를 사용해야 합니다. Kubernetes 클러스터의 기본 스토리지 클래스는 일반적으로 ReadWriteOnce 스토리지 클래스 (블록 스토리지) 용입니다. 예를 들어, Red Hat OpenShift Data Foundation를 사용하는 경우 스토리지 클래스 *ocs-storagecluster-cephfs* 는 적합한 공유 파일 시스템을 제공합니다. 모든 공유 파일 시스템이 동일한 방식으로 파일 잠금을 처리하는 것은 아니므로 파일 시스템을 선택하는 것은 매우 중요합니다. 멀티플랫폼에서 파일 시스템 지원 계획 및 IBM MQ 멀티 인스턴스 큐 관리자 파일 시스템에 대한 명령문 테스트를 참조하십시오.
2. 61 페이지의 『OpenSSL 를 사용하여 자체 서명된 PKI 작성』 에 설명된 대로 인증서 쌍을 작성하십시오.
3. MQSC 명령 및 INI 파일을 포함하는 구성 맵 작성

   MQSC 명령을 포함하는 Kubernetes ConfigMap 을 작성하여 새 큐 및 SVRCONN 채널을 작성하고 채널에 대한 액세스를 허용하는 채널 인증 레코드를 추가하십시오.

   이전에 작성한 네임스페이스 (시작하기 전에참조) 에 있는지 확인한 후 OCP웹 콘솔에서 또는 명령행을 사용하여 다음 YAML을 입력하십시오.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: example-miqm-configmap
data:
  example-tls.mqsc: |
    DEFINE CHANNEL('MTLS.SVRCONN') CHLTYPE(SVRCONN) SSLCAUTH(REQUIRED)
SSLCIPH('ANY_TLS13_OR_HIGHER') REPLACE
    SET CHLAUTH('MTLS.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=*') USERSRC(NOACCESS)
```

```
ACTION(REPLACE)
    SET CHLAUTH('MTLS.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=example-app1') USERSRC(MAP)
MCAUSER('app1') ACTION(REPLACE)
    SET AUTHREC PRINCIPAL('app1') OBJTYPE(QMGR) AUTHADD(CONNECT,INQ)
    DEFINE QLOCAL('EXAMPLE.QUEUE') REPLACE
    SET AUTHREC PROFILE('EXAMPLE.QUEUE') PRINCIPAL('app1') OBJTYPE(QUEUE)
AUTHADD(BROWSE,PUT,GET,INQ)
  example-tls.ini: |
    Service:
        Name=AuthorizationService
        EntryPoints=14
        SecurityPolicy=UserExternal
```

MQSC는 *MTLS.SVRCONN* 이라는 채널 및 *EXAMPLE.QUEUE*를 큐에 지정하십시오. 채널은 *example-app1*의 "공통 이름" 으로 인증서를 제공하는 클라이언트에만 액세스할 수 있도록 구성됩니다. 이는 74 페이지의 『2』 단계에서 작성된 인증서 중 하나에서 사용되는 공통 이름입니다. 이 공용 이름을 사용하는 이 채널의 연결은 큐 관리자에 연결하고 예제 큐에 액세스할 수 있는 권한이 부여된 *app1*의 사용자 ID에 맵핑됩니다. INI 파일은 *app1* 사용자 ID가 외부 사용자 레지스트리에 존재할 필요가 없음을 의미하는 보안 정책을 사용 가능하게 합니다. 이 구성에서는 이름으로만 존재합니다.

4. 큐 관리자 배치

   다음 사용자 정의 자원 YAML을 사용하여 새 큐 관리자를 작성하십시오. 이 태스크를 시작하기 전에 작성한 네임스페이스에 있는지 확인한 후 OCP웹 콘솔에서 또는 명령행을 사용하여 다음 YAML을 입력하십시오. 올바른 라이센스가 지정되었는지 확인하고, `false`를 `true`로 변경하여 라이센스에 동의하십시오.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: exampleqm
spec:
  license:
    accept: false
    license: L-EHXT-MQCRN9
    use: Production
  queueManager:
    name: EXAMPLEQM
    availability:
      type: MultiInstance
    mqsc:
    - configMap:
        name: example-miqm-configmap
        items:
        - example-tls.mqsc
    ini:
    - configMap:
        name: example-miqm-configmap
        items:
        - example-tls.ini
    storage:
      defaultClass: STORAGE_CLASS
  version: 9.4.0.0-r1
  pki:
    keys:
      - name: default
        secret:
          secretName: example-qm-tls
          items:
            - tls.key
            - tls.crt
            - ca.crt
```

STORAGE_CLASS 를 74 페이지의 『1』 단계에서 식별한 스토리지 클래스로 변경하십시오.

시크릿 *example-qm-tls* 는 74 페이지의 『2』 단계에서 작성되었으며 ConfigMap *example-miqm-configmap* 은 74 페이지의 『3』 단계에서 작성되었습니다.

가용성 유형이 *MultiInstance*로 설정되어 지속적 스토리지가 자동으로 선택됩니다.

5. 큐 관리자가 실행 중인지 확인

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

```
oc get qmgr exampleqm
```

6. 큐 관리자에 대한 연결 테스트
   큐 관리자가 구성되어 사용 가능한지 확인하려면 65 페이지의 『랩탑에서 큐 관리자에 대한 상호 TLS 연결 테스트』의 단계를 따르십시오.
7. 활성 팟(Pod)이 실패하도록 강제 실행
   큐 관리자의 자동 복구를 유효성 검증하기 위해 팟(Pod) 실패를 시뮬레이션합니다.
   a) 활성 및 대기 팟(Pod) 보기
      다음 명령을 실행하십시오.

      ```
      oc get pods --selector app.kubernetes.io/instance=exampleqm
      ```

      READY 필드에서 활성 팟(Pod)은 `1/1` 값을 리턴하는 반면, 대기 팟(Pod)은 `0/1` 값을 리턴합니다.
   b) 활성 팟(Pod) 삭제
      활성 팟(Pod)의 전체 이름을 지정하며 다음 명령을 실행하십시오.

      ```
      oc delete pod exampleqm-ibm-mq-value
      ```

   c) 팟(Pod) 상태 다시 보기
      다음 명령을 실행하십시오.

      ```
      oc get pods --selector app.kubernetes.io/instance=exampleqm
      ```

   d) 큐 관리자 상태 보기
      다른 팟 (Pod) 의 전체 이름을 지정하여 다음 명령을 실행하십시오.

      ```
      oc exec -t Pod -- dspmq -x
      ```

      예를 들어 활성 인스턴스가 변경되었음을 상태에서 표시해야 합니다.

      ```
      QMNAME(EXAMPLEQM)                                         STATUS(Running as standby)
          INSTANCE(exampleqm-ibm-mq-1) MODE(Active)
          INSTANCE(exampleqm-ibm-mq-0) MODE(Standby)
      ```

   e) 큐 관리자에 대한 연결을 다시 테스트하십시오.
      큐 관리자가 복구되었는지 확인하려면 65 페이지의 『랩탑에서 큐 관리자에 대한 상호 TLS 연결 테스트』의 단계를 따르십시오.

결과

축하합니다. 상호 TLS 인증을 사용하여 다중 인스턴스 큐 관리자를 배치하고 활성 팟 (Pod) 이 실패하면 자동으로 복구되는지 확인했습니다.

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

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 맵핑하는 방법에 대한 정보는 IBM MQ에서 다중 인증서 기능을 제공하는 방법을 참조하십시오.

SNI 헤더가 IBM MQ 채널 이름 또는 호스트 이름으로 설정되는지 여부는 **OutboundSNI** 속성을 사용하여 제어됩니다. 가능한 값은 OutboundSNI=CHANNEL (기본값) 또는 OutboundSNI=HOSTNAME입니다. 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

IBM MQ 관리 .NET 클라이언트는 **OutboundSNI** 특성이 HOSTNAME으로 설정된 경우 SERVERNAME 을 각각의 호스트 이름으로 설정합니다. 그러면 IBM MQ 관리 .NET 클라이언트가 Red Hat OpenShift 라우트를 사용하여 큐 관리자에 연결할 수 있습니다.

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 MQ 가 채널 이름을 SNI 헤더에 맵핑하는 방식 때문에 MQ 채널 이름이 소문자로 끝나지 않는 것이 중요합니다.

각각의 새 Red Hat OpenShift Route에 필요한 호스트 이름을 판별하려면 각 채널 이름을 SNI 주소에 맵핑해야 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: unique_name_for_the_route
  namespace: namespace_of_your_MQ_deployment
spec:
  host: SNI_address_mapping_for_the_channel
  to:
    kind: Service
    name: name_of_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-qm")>
-n namespace of your MQ deployment -o jsonpath="{.spec.host}"
```

클라이언트 연결을 위한 포트는 Red Hat OpenShift Container Platform 라우터 - 일반적으로 443에서 사용되는 포트로 설정해야 합니다.

관련 태스크

118 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 IBM Instana 추적과 IBM MQ 통합

IBM Instana 를 사용하여 IBM Cloud Pak for Integration내에서 트랜잭션을 추적할 수 있습니다.

시작하기 전에

이 문서에서는 시스템을 통해 메시지를 추적하는 프로세스인 IBM Instana 추적을 다룹니다. IBM MQ 큐 관리자의 상태에 대한 세부사항이 검색되는 IBM Instana 모니터링은 다루지 않습니다. IBM Instana 에 의한 IBM MQ 모니터링에 대한 정보는 IBM MQ모니터링을 참조하십시오. 인증된 모니터링에 대한 자세한 지시사항은 80 페이지의 『TLS를 사용하여 인증된 IBM Instana 모니터링 구성』의 내용을 참조하십시오.

참고:

- 이 기능은 IBM MQ 버전 9.3.1.0-r2 이상의 피연산자에서만 지원됩니다.
- 이전 IBM MQ 운영자 및 큐 관리자 버전에서 IBM Instana 추적을 실행할 수 있지만 기본적으로는 실행할 수 없습니다. IBM Instana 문서에서 IBM MQ 추적 구성 을 참조하십시오.

IBM MQ 운영자를 사용하여 IBM Instana 추적을 수행하려면 먼저 IBM Instana 백엔드 및 IBM Instana 에이전트를 모두 배치해야 합니다. 기본적으로 IBM MQ 큐 관리자는 큐 관리자 팟 (Pod) 과 동일한 노드에 배치된 IBM Instana 에이전트와 통신합니다.

이 태스크 정보

IBM Instana 와의 통합을 사용으로 설정하면 IBM MQ API 엑시트가 큐 관리자에 설치됩니다. API 엑시트는 큐 관리자를 통해 플로우되는 메시지에 대한 추적 데이터를 IBM Instana 에이전트에 송신합니다.

API 엑시트는 각 메시지에 RFH2 헤더를 추가합니다. 이 헤더에는 추적 정보가 포함되어 있습니다.

IBM Instana 에이전트는 추적 데이터를 IBM Instana 백엔드로 전송할 책임이 있습니다.

IBM Instana 백엔드 및 IBM Instana 에이전트 배치에 대한 정보는 IBM Instana 문서에서 플랫폼 UI에서 Instana 모니터링 링크 사용 을 참조하십시오.

프로시저

표준 배치

- IBM Instana 추적을 사용하여 큐 관리자를 배치하십시오.

  기본적으로 IBM Instana 추적은 사용 안함으로 설정되어 있습니다.

  IBM Cloud Pak for Integration Platform UI 또는 OpenShift 웹 콘솔을 사용하는 경우:

  1. **텔레메트리** > **추적** > **Instana**를 클릭하십시오.
  2. **인스턴스 추적 사용** 토글을 `true`로 설정하십시오.

  YAML을 통해 배치하는 경우 다음 스니펫을 사용하십시오.

```
spec:
  telemetry:
    tracing:
      instana:
        enabled: true
```

고급 배치

- https를 통해 IBM Instana 에이전트와 통신합니다.

  기본적으로 IBM MQ 의 IBM Instana 엑시트는 http를 통해 IBM Instana 에이전트와 통신합니다. 에이전트의 호스트 주소는 큐 관리자가 실행 중인 노드의 IP 주소로 설정됩니다. 이는 IBM Instana 문서의 IBM Instana 모니터링 사용 에 설명된 구성과 일치합니다. 여기서 IBM Instana 에이전트는 IBM Instana 에이전트 운영자에 의해 디먼 세트로 배치됩니다.

  현재 IBM MQ 용 IBM Instana 엑시트와 IBM Instana 에이전트 간의 통신은 http 또는 https 프로토콜을 지원합니다. https를 사용하려면 먼저 TLS 암호화를 사용하도록 IBM Instana 에이전트를 구성해야 합니다. IBM Instana 문서의 에이전트 엔드포인트에 대한 TLS 암호화 설정 을 참조하십시오. 그런 다음 다음과 같이 프로토콜을 https로 설정할 수 있습니다.

  OpenShift 웹 콘솔을 사용하는 경우 다음을 수행하십시오.

  1. **텔레메트리** > **Instana**를 클릭하십시오.
  2. **고급 구성** 드롭 다운 목록을 펼치십시오.
  3. **Instana 에이전트 통신 프로토콜** 을 `https`로 설정하십시오.

  YAML을 통해 배치하는 경우 다음 스니펫을 사용하십시오.

```
spec:
  telemetry:
    instana:
      enabled: true
      protocol: https
```

- Set the **`agentHost`**

  IBM Instana 에이전트가 큐 관리자가 실행 중인 Openshift 클러스터에서 디먼 세트로 배치되지 않은 경우, **`agentHost`** 값을 IBM Instana 에이전트가 실행 중인 호스트 이름 또는 IP 주소로 설정해야 합니다. **`agentHost`** 값은 프로토콜 또는 포트를 포함하지 않아야 합니다.

  OpenShift 웹 콘솔을 사용하는 경우 다음을 수행하십시오.

  1. **텔레메트리** > **Instana**를 클릭하십시오.
  2. **고급 구성** 드롭 다운 목록을 펼치십시오.
  3. **에이전트 호스트 인스턴스화** 텍스트 상자에 호스트 이름을 입력하십시오.

  YAML을 통해 배치하는 경우 다음 스니펫을 사용하십시오.

```
spec:
  telemetry:
```

```
    instana:
      enabled: true
      agentHost: 9.9.9.9
```

다음에 수행할 작업

57 페이지의 『IBM MQ Operator 를 사용하여 단순 큐 관리자 배치』도 참조하십시오.

OpenShift Operator2.2.0 CP4I TLS를 사용하여 인증된 IBM Instana 모니터링 구성

IBM Instana 에이전트를 통해 큐 관리자를 모니터하려면 에이전트와 큐 관리자를 모두 구성해야 합니다.

시작하기 전에

IBM Instana 문서의 " IBM MQ모니터링" 의 "구성" 섹션 는 IBM Instana 모니터링 구성에 관한 일반 정보를 제공합니다. 그러나 큐 관리자 구성에 대한 세부사항은 포함되어 있지 않습니다.

IBM MQ 운영자를 사용하여 IBM Instana 추적을 수행하기 전에 IBM Instana 백엔드 및 IBM Instana 에이전트를 모두 배치해야 합니다. 이를 수행하려면 IBM Instana 문서에서 CP4I Platform UI에서 IBM Instana 모니터링 사용 을 참조하십시오.

프로시저

1. 인증서를 생성하십시오.
2. IBM Instana 에이전트를 구성하십시오.
3. 큐 관리자를 구성하십시오.
4. 확인 및 디버그.

관련 태스크

78 페이지의 『IBM Instana 추적과 IBM MQ 통합』
IBM Instana 를 사용하여 IBM Cloud Pak for Integration내에서 트랜잭션을 추적할 수 있습니다.

OpenShift Operator2.2.0 CP4I *IBM Instana* 에이전트 및 큐 관리자에 대한 인증서 및 키 생성

IBM Instana 에이전트와 큐 관리자 간의 TLS 통신의 경우 둘 다 인증서와 해당 개인 키가 있어야 합니다.

시작하기 전에

이는 TLS를 사용하여 인증된 IBM Instana 모니터링을 구성하기 위한 네 가지 태스크 중 첫 번째입니다.

참고: 이러한 인증서의 생성에 사용되는 값은 데모 용도로 사용됩니다. 프로덕션 환경에서 배치할 때 인증서의 주제 및 만기가 적절한지 확인하십시오.

프로시저

IBM MQ 큐 관리자

TLS를 통해 IBM Instana 에이전트와 통신하려면 큐 관리자에 인증서 및 해당 개인 키가 있어야 합니다. 이 섹션이 이미 있는 경우에는 이 섹션을 건너뛰십시오.

1. 큐 관리자에 대한 인증서 및 개인 키를 생성하십시오.

   다음 명령을 실행하십시오.

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

IBM Instana 에이전트

에이전트가 IBM MQ 큐 관리자와 TLS 통신을 수행하려면 에이전트에 인증서 및 해당 개인 키가 있어야 합니다. 사용하려는 JKS키 저장소에 개인 키 및 인증서가 이미 있는 경우에는 이 섹션을 건너뛰십시오.

2. IBM Instana 에이전트에 대한 인증서 및 개인 키를 생성하십시오.

   다음 명령을 실행하십시오.

   ```
   openssl req \
          -newkey rsa:2048 -nodes -keyout application.key \
          -subj "/CN=instana-agent/OU=app team1" \
          -x509 -days 3650 -out application.crt
   ```

3. PKCS12 키 저장소에 인증서 및 개인 키를 저장하십시오.

   다음 명령을 실행하여 *your_password* 를 키 저장소를 보안하는 데 사용할 비밀번호로 바꾸십시오. 모든 후속 단계에서 이 대체를 수행하십시오.

   ```
   openssl pkcs12 -export -out application.p12 -inkey application.key -in application.crt
   -passout pass:your_password
   ```

4. PKCS12 키 저장소를 JKS키 저장소로 변환하십시오.

   다음 명령을 실행하십시오.

   ```
   keytool -importkeystore \
           -srckeystore application.p12 \
           -srcstoretype pkcs12 \
           -destkeystore application.jks \
           -deststoretype JKS \
           -srcstorepass your_password \
           -deststorepass your_password \
           -noprompt
   ```

5. 인증서의 레이블을 지정합니다.

   다음 명령을 실행하십시오.

   ```
   keytool -changealias -alias "1" -destalias "instana" -keypass your_password -keystore
   application.jks -storepass your_password -noprompt
   ```

6. 큐 관리자 인증서를 키 저장소로 가져오십시오.

   다음 명령을 실행하십시오.

   ```
   keytool -importcert -file server.crt -keystore application.jks -storepass your_password
   -alias myca -noprompt
   ```

다음에 수행할 작업

이제 IBM Instana 모니터링을 위해 에이전트를 구성할 준비가 되었습니다.

OpenShift Operator2.2.0 CP4I *Instana* 모니터링*:* 에이전트 구성

IBM Instana 에이전트에 키 저장소를 마운트한 후 특정 큐 관리자에 대한 모니터링을 구성하십시오.

시작하기 전에

이 태스크에서는 가 IBM Instana 에이전트 및 큐 관리자에 대한 인증서 및 키를 생성했다고 가정합니다.

프로시저

IBM Instana 에이전트에 키 저장소 마운트

1. IBM Instana 에이전트 네임스페이스의 JKS키 저장소에서 시크릿을 작성하십시오.

   다음 명령을 실행하여 *keystore_secret_name* 을 사용할 이름으로 바꾸십시오. 모든 후속 단계에서 이 대체를 수행하십시오.

   ```
   oc create secret generic keystore_secret_name --from-file=./application.jks -n instana-agent
   ```

2. instana-agent 네임스페이스에서 `oc edit daemonset instana-agent` 명령을 사용하여 다음 추가 volumeMount 및 볼륨을 포함하도록 instana-agent 디먼 세트를 편집하십시오.

```
volumeMounts:
- name: mq-key-jks-name
   subPath: application.jks
   mountPath: /opt/instana/agent/etc/application.jks
volumes:
- name: mq-key-jks-name
   secret:
      secretName: keystore_secret_name
```

특정 큐 관리자에 대한 모니터링 구성

3. instana-agent 네임스페이스에서 `oc edit configmap instana-agent` 명령을 사용하여 instana-agent configmap을 편집하십시오.
4. `configuration.yaml: |`아래에 다음 섹션을 추가하십시오. 이 섹션을 이미 정의한 경우에는 목록에 새 큐 관리자를 추가하십시오.

```
    com.instana.plugin.ibmmq:
      enabled: true
      poll_rate: 60
      queueManagers:
         QUEUE_MANAGER_NAME:
            channel: 'INSTANA.A.SVRCONN'
            keystorePassword: 'your_password'
            keystore: '/opt/instana/agent/etc/application.jks'
            cipherSuite: 'TLS_RSA_WITH_AES_256_CBC_SHA256'
```

여기서,

- *your_password* 는 JKS키 저장소에 대한 비밀번호입니다.
- *QUEUE_MANAGER_NAME* 은 큐 관리자 피연산자의 이름이 아니라 배치할 기본 IBM MQ 큐 관리자의 이름입니다.

참고: *QUEUE_MANAGER_NAME* 이 기본 큐 관리자 이름으로 설정되지 않고 대신 Operand로 설정되면 모니터링이 작동하지 않습니다. 기본 이름은 큐 관리자 피연산자에 대해 `spec.queuemanager.name` 에 정의되어 있습니다.

5. instana-agent 네임스페이스에서 instana-agent팟 (Pod) 을 삭제하십시오. 그러면 사용자가 다시 시작되고 새 설정으로 모니터링을 시작합니다.

다음에 수행할 작업

이제 IBM Instana 모니터링을 위해 큐 관리자를 구성할 준비가 되었습니다.

OpenShift Operator2.2.0 CP4I

Instana 모니터링*:* 큐 관리자 구성

TLS를 사용하여 IBM Instana 에이전트와 통신하는 큐 관리자를 설정하십시오.이 연결에 대한 인증은 SSLPEERMAP를 사용하여 수행됩니다.

시작하기 전에

이 태스크에서는 IBM Instana 모니터링을 위해 에이전트를 구성했다고 가정합니다.

프로시저

1. MQSC및 INI를 모두 사용하여 큐 관리자를 구성하십시오.

   MQSC는 새 TLS 사용 채널을 설정하는 데 사용되며, 필수 필드가 있는 인증서가 있는 경우 연결 IBM Instana 에이전트를 인증하도록 해당 채널을 구성합니다. 이 경우, `CN=instana-agent,OU=app team1` 필드를 포함하는 인증서가 있는 연결 클라이언트를 사용자 app1에 맵핑합니다. 그런 다음 MQSC는 app1 사용자에게 IBM Instana 모니터링에 필요한 조작을 수행할 수 있는 권한을 부여합니다.

   INI 파일은 외부 사용자 app1에게 권한을 부여하는 데 사용됩니다.

다음 configmap에는 필수 MQSC및 INI 설정이 포함되어 있습니다. 이를 큐 관리자 네임스페이스에 배치하십시오.

```
apiVersion: v1
data:
  channel.mqsc: |-
    DEFINE CHANNEL('INSTANA.A.SVRCONN') CHLTYPE(SVRCONN) SSLCAUTH(REQUIRED)
SSLCIPH('ANY_TLS12_OR_HIGHER')
    ALTER QMGR CONNAUTH(' ')
    REFRESH SECURITY
    SET CHLAUTH('INSTANA.A.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=*') USERSRC(NOACCESS)
ACTION(REPLACE)
    SET CHLAUTH('*') TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS) ACTION(REPLACE)
    SET CHLAUTH('INSTANA.A.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=instana-agent,OU=app
team1') USERSRC(MAP) MCAUSER('app1')
    SET AUTHREC PRINCIPAL('app1') OBJTYPE(QMGR) AUTHADD(ALL)
    SET AUTHREC PROFILE('SYSTEM.ADMIN.COMMAND.QUEUE') PRINCIPAL('app1') OBJTYPE(QUEUE)
AUTHADD(PUT,INQ,DSP,CHG)
    SET AUTHREC PROFILE('SYSTEM.**') PRINCIPAL('app1') OBJTYPE(TOPIC) AUTHADD(DSP)
    SET AUTHREC PROFILE('*') PRINCIPAL('app1') OBJTYPE(TOPIC) AUTHADD(DSP)
    SET AUTHREC PROFILE('SYSTEM.**') PRINCIPAL('app1') OBJTYPE(QUEUE) AUTHADD(DSP, CHG, GET)
    SET AUTHREC PROFILE('SYSTEM.**') PRINCIPAL('app1') OBJTYPE(LISTENER) AUTHADD(DSP)
    SET AUTHREC PROFILE('AMQ.*') PRINCIPAL('app1') OBJTYPE(QUEUE) AUTHADD(DSP, CHG)
    REFRESH SECURITY TYPE(CONNAUTH)
  auth.ini: |-
    Service:
      Name=AuthorizationService
      EntryPoints=14
      SecurityPolicy=UserExternal
kind: ConfigMap
metadata:
  namespace: your-queue-manager-namespace
  name: qmgr-monitoring-config
```

여기서 *your-queue-manager-namespace* 는 큐 관리자가 배치될 네임스페이스입니다.

참고: 사용자 정의 큐를 모니터링하는 경우, 해당 큐에 DSP, CHG및 GET 권한을 부여하여 configmap MQSC에 추가 행을 추가해야 합니다. 예를 들면, 다음과 같습니다.

```
SET AUTHREC PROFILE('MYQUEUE') PRINCIPAL('app1') OBJTYPE(QUEUE) AUTHADD(DSP, CHG, GET).
```

이 예제에서는 MQSC및 INI 데이터에 대해 configmap을 사용하지만 사용자가 추가한 사항이 기밀인 경우 시크릿을 사용할 수 있습니다. MQSC및 INI를 사용한 배치에 대한 일반 정보는 60 페이지의 『예: MQSC 및 INI 파일 제공』의 내용을 참조하십시오.

2. TLS 연결을 작성하려면 큐 관리자가 IBM Instana 에이전트의 인증서를 신뢰해야 합니다. 이를 수행하려면 IBM Instana 에이전트의 인증서만 포함하는 시크릿을 작성하십시오.

```
oc create secret generic instana-certificate-secret --from-file=./application.crt -n your-
queue-manager-namespace
```

3. 큐 관리자는 TLS 핸드쉐이크에 대한 자체 인증서를 제공해야 하며 연관된 개인 키에 대한 액세스가 필요합니다. 이전에 작성했거나 이미 소유한 키 및 인증서가 포함된 시크릿을 배치하십시오.

```
oc create secret tls qm-tls-secret --cert server.crt --key server.key -n your-queue-manager-
namespace
```

작성된 configmap및 시크릿을 사용하여 큐 관리자 자체를 작성할 준비가 되었습니다.

4. 큐 관리자 YAML이 큐 관리자 컨테이너에서 환경 변수 **MQSNOAUT** 를 설정하지 않는지 확인하십시오.

그렇지 않으면 사용으로 설정된 후 인증 메커니즘이 작동하지 않습니다. 배치 후 변수를 제거해도 메커니즘이 다시 사용 가능하게 되지 않으며 큐 관리자를 다시 작성해야 합니다.

5. 다음 섹션을 큐 관리자 정의에 추가하십시오. 여기서 *MYQM* 은 큐 관리자의 이름입니다.

```
spec:
  queueManager:
    name: MYQM   #(a)
    ini:   #(b)
      - configMap:
          items:
            - auth.ini
          name: qmgr-monitoring-config
```

```
    mqsc:   #(c)
      - configMap:
          items:
            - channel.mqsc
          name: qmgr-monitoring-config
  pki:
    keys:   #(d)
        - name: default
          secret:
            items:
            - tls.key
            - tls.crt
            secretName: qm-tls-secret
    trust:   #(e)
      - name: app
        secret:
          items:
            - application.crt
          secretName: instana-certificate-secret
```

스펙의 플래그 지정된 섹션은 다음과 같이 설명됩니다.

a. 기본 큐 관리자에 고유한 이름을 지정했는지 확인하십시오. 기본 큐 관리자에 고유한 이름이 없는 경우 모니터링이 의도한 대로 작동하지 않을 수 있습니다. 이 이름은 이전에 편집된 IBM Instana 에이전트 configmap의 이름과 일치해야 합니다.

b. configmap에 기록된 INI 정보가 큐 관리자에 추가됩니다.

c. configmap에 기록된 MQSC 정보가 큐 관리자에 추가됩니다.

d. 큐 관리자 인증서 및 개인 키가 큐 관리자 키 저장소에 추가됩니다.

e. IBM Instana 에이전트 인증서가 큐 관리자 신뢰 저장소에 추가됩니다.

6. 옵션: 모니터된 큐 관리자에서 IBM Instana 추적을 사용으로 설정하십시오.

   이를 수행하려면 78 페이지의 『IBM Instana 추적과 IBM MQ 통합』의 내용을 참조하십시오.

7. 큐 관리자를 배치하십시오.

다음에 수행할 작업

이제 IBM Instana 모니터링을 확인하고 디버그할 준비가 되었습니다.

OpenShift Operator2.2.0 CP4I *Instana* 모니터링*:* 확인 및 디버깅

IBM Instana 에이전트를 통해 큐 관리자를 모니터하려면 에이전트와 큐 관리자를 모두 구성해야 합니다.

시작하기 전에

이 태스크에서는 IBM Instana 모니터링을 위해 큐 관리자를 구성했다고 가정합니다.

프로시저

확인 중

1. 배치에 성공했는지 확인하려면 IBM Instana 대시보드에서 큐 관리자를 보십시오.

   큐 관리자는 애플리케이션 페이지의 서비스 섹션 및 인프라 보기에서도 볼 수 있어야 합니다.

디버그

참고: 이 디버깅 단계에서는 디먼 세트로 실행 중인 IBM Instana 에이전트의 Openshift 배치를 가정합니다.

IBM Instana 대시보드에서 큐 관리자를 볼 수 없는 경우 큐 관리자를 잘못 구성했을 수 있습니다. 다음 단계를 사용하여 조사하십시오.

2. 활성 큐 관리자 팟 (Pod) 이 실행 중인 노드를 식별하십시오.

   큐 관리자 네임스페이스에서 다음 명령을 실행하십시오.

```
oc get pods -o wide -n your-queue-manager-namespace
```

3. 큐 관리자와 동일한 노드에서 실행 중인 IBM Instana 에이전트 팟 (Pod) 을 판별하려면 instana-agent 네임스페이스에서 동일한 명령을 실행하십시오.

```
oc get pods -o wide -n instana-agent-namespace
```

4. IBM Instana 에이전트 측의 문제를 이해하려면 IBM Instana 에이전트 팟 (Pod) 의 로그를 가져오고 'mq' 또는 큐 관리자의 이름과 관련된 항목을 찾으십시오.

   다음 명령을 실행하십시오.

```
oc logs instana-agent-pod -c instana-agent -n instana-agent
```

5. 큐 관리자 로그를 확인하십시오.

   에이전트가 큐 관리자에 연결하려고 시도한 경우 큐 관리자 로그는 연결에 실패한 이유를 표시해야 합니다. 다음 명령을 실행하십시오.

```
oc logs your-queue-manager-name -n your-queue-manager-namespace
```

결과

TLS를 사용하여 인증된 IBM Instana 모니터링을 구성하기 위한 네 가지 태스크를 모두 완료했습니다.

OpenShift CP4I 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 MQSC와 INI 파일을 포함한 새 IBM MQ 컨테이너 이미지를 작성하십시오. 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.

Red Hat OpenShift 프로젝트에 IBM Entitled Registry에 대한 Red Hat OpenShift 시크릿이 없는 경우 인타이틀먼트 키 시크릿 작성의 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 컨텐츠를 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:9.4.0.0-r1
      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:9.4.0.0-r1"
      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 5 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 `BuildConfig`을(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

57 페이지의 『IBM MQ Operator 를 사용하여 단순 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

57 페이지의 『IBM MQ Operator 를 사용하여 단순 큐 관리자 배치』
이 예는 임시 (비지속적) 스토리지를 사용하는 "빠른 시작" 큐 관리자를 배치하고 IBM MQ 보안을 끕니다. 큐 관리자를 다시 시작해도 메시지가 지속되지 않습니다. 구성을 조정하여 많은 큐 관리자 설정을 변경할 수 있습니다.

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 `metadata`에 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 `metadata`에 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.

프로시저

- 런타임 웹훅 검사를 사용 안함으로 설정하십시오.

  `metadata`에 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 Operator2.1.0 CP4I 큐 관리자 스펙에 대한 기본 값 업데이트 사용 안함

IBM MQ Operator 는 큐 관리자 스펙에서 지정되지 않은 값을 기본값으로 업데이트합니다. 큐 관리자 스펙을 수정하지 않으려는 경우 이 작동을 사용 안함으로 설정할 수 있습니다. 큐 관리자 상태 필드는 여전히 업데이트됩니다.

프로시저

- 큐 관리자 기본 값 업데이트를 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.mq/write-defaults-spec" : "false"
```

  참고: 빠른 시작 예제에는 기본적으로 이 어노테이션이 적용되어 있습니다.

읽기 전용 루트 파일 시스템으로 IBM MQ 컨테이너 실행

읽기 전용 루트 파일 시스템을 사용하여 실행하도록 IBM MQ 컨테이너를 구성할 수 있습니다. 이는 공격자가 컨테이너에서 악성 코드를 복사하고 실행하는 것을 방지합니다.

이 태스크 정보

읽기 전용 루트 파일 시스템을 사용하면 컨테이너 파일을 변경할 수 없습니다. 즉, 컨테이너 파일 시스템에서 파일을 볼 수 있지만 수정할 수는 없으며 새 파일을 작성할 수 없습니다. 마운트된 파일 시스템에서만 파일을 수정하거나 작성할 수 있습니다.

읽기 전용 루트 파일 시스템이 사용으로 설정되면 두 개의 임시 볼륨 스크래치 및 Tmp 가 작성되고 컨테이너의 /run 및 /tmp 디렉토리에 각각 마운트됩니다.

- 스크래치 볼륨에는 큐 관리자를 구성하는 데 사용되는 파일, 키 저장소 및 기타 파일이 포함되어 있습니다.
- Tmp 볼륨에는 진단 파일 (예: 큐 관리자 RAS 파일) 이 포함되어 있습니다.

이러한 볼륨은 일시적이기 때문에 이러한 볼륨의 파일은 팟 (Pod) 다시 시작 시 유실됩니다.

큐 관리자 데이터에 대해 작성된 볼륨의 유형은 스토리지 유형에 따라 다릅니다. 기본적으로 지속적 볼륨이 마운트됩니다. 또는 스토리지 유형이 ephemeral인 경우 임시 볼륨이 마운트됩니다. 볼륨에 있는 데이터의 크기가 **sizeLimit** 특성에 지정된 값을 초과하는 경우 Kubernetes 는 컨테이너를 꺼내고 새 컨테이너를 작성할 수 있습니다.

읽기 전용 루트 파일 시스템은 기본적으로 사용으로 설정되어 있지 않습니다. 이를 사용하려면 다음 단계를 완료하십시오.

프로시저

1. spec.securityContext API를 사용하여 읽기 전용 루트 파일 시스템을 사용으로 설정하십시오.

   큐 관리자의 경우 139 페이지의 『.spec.securityContext』 의 **readOnlyRootFilesystem** 특성을 true 로 설정하십시오.

   IBM MQ Operator 는 두 개의 임시 볼륨, 스크래치 및 Tmp를 작성합니다.
2. 옵션: 큐 관리자 데이터 스토리지 유형을 설정하거나 변경하십시오.

   기본적으로 지속적 볼륨 청구는 /mnt/mqm에 마운트됩니다. 또는 **type** 특성이 138 페이지의 『.spec.queueManager.storage.queueManager』에서 ephemeral 로 설정된 경우 임시 볼륨이 작성되고 마운트됩니다.
3. 각 임시 볼륨에 대해 데이터가 증가할 수 있는 양을 신중하게 고려하십시오. SI 단위를 포함하여 **sizeLimit** 특성의 값을 적절하게 설정하십시오.
   - 스크래치 임시 볼륨의 경우 139 페이지의 『.spec.queueManager.storage.scratch』에서 **sizeLimit** 특성을 설정하십시오. 기본값은 "100M" 입니다.
   - Tmp 임시 볼륨의 경우 139 페이지의 『.spec.queueManager.storage.tmp』에서 **sizeLimit** 특성을 설정하십시오. 기본값은 "2Gi" 입니다.

- 큐 관리자 볼륨의 **type** 가 ephemeral로 설정된 경우 138 페이지의 『.spec.queueManager.storage.queueManager』에서 **sizeLimit** 특성을 설정하십시오. 기본값은 "2Gi" 입니다.

OpenShift V 9.4.0 IBM MQ Operator 를 사용하여 기본 레지스트리로 IBM MQ Console 구성

IBM MQ Console에 로그인하기 위해 큐 관리자에 사용자 고유의 구성을 제공할 수 있습니다.

시작하기 전에

IBM MQ Advanced for Developers 라이센스를 사용하여 큐 관리자를 배치하는 경우 기본 제공되는 단순 구성이 있습니다. 157 페이지의 『admin 및 app 사용자의 비밀번호를 지정하는 방법을 설명하는 예제 큐 관리자 YAML』을 참조하십시오. IBM Cloud Pak for Integration 라이센스 큐 관리자를 배치하는 경우 싱글 사인온을 사용하여 IBM MQ Console 에 로그인하도록 IBM Cloud Pak for Integration Keycloak 와의 통합을 사용으로 설정할 수 있습니다. 118 페이지의 『Red Hat OpenShift 클러스터에 배치된 IBM MQ Console에 연결』을(를) 참조하십시오.

프로시저

1. **비밀번호를 작성하고 securityUtility를 사용하여 암호화하십시오.**

   ConfigMap 는 큐 관리자에 액세스하는 데 사용하는 신임 정보를 저장하는 데 사용됩니다. 보안을 개선하기 위해 securityUtility 명령을 사용하여 이러한 신임 정보를 인코딩합니다.

   또는 Kubernetes 계층에서 신임 정보를 보호하는 시크릿을 사용할 수 있습니다. 그러나 모니터링 또는 문제점 해결 도구를 사용하면 기본 파일이 안전하지 않게 노출될 수 있습니다.
2. 옵션: **Red Hat OpenShift 명령행 인터페이스 (CLI) 에 로그인하십시오.**

   OpenShift CLI를 사용하는 경우 `oc login`를 사용하여 로그인하십시오.

   또는 OpenShift 콘솔을 사용할 수 있습니다.
3. **구성을 사용하여 ConfigMap 를 작성하십시오.**

   XML 구성 작성에 대한 도움말은 IBM MQ Console 및 REST API 보안을 참조하십시오.

   다음 예제는 MQWebAdminGroup그룹 내에 사용자를 작성합니다. MQWebAdminGroup 의 구성원에게는 MQWebAdmin 역할이 지정됩니다. 이 예제에서

   - **반드시** *USERNAME* 및 *PASSWORD* 를 사용자 고유의 값으로 대체해야 합니다. *USERNAME* 은 예제에서 두 번 사용됩니다.

     반드시 *NAMESPACE* 를 IBM MQ Operator 가 배치되고 큐 관리자가 배치될 위치 또는 이미 배치된 위치로 지정해야 합니다.

   a) OpenShift 콘솔 또는 명령행을 사용하여 다음 ConfigMap를 작성하십시오.

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: mqwebuserconfigmap
  namespace: NAMESPACE
data:
  mqwebuser.xml: |
    <?xml version="1.0" encoding="UTF-8"?>
    <server>
      <featureManager>
        <feature>appSecurity-2.0</feature>
        <feature>basicAuthenticationMQ-1.0</feature>
      </featureManager>
      <enterpriseApplication id="com.ibm.mq.console">
        <application-bnd>
          <security-role name="MQWebAdmin">
            <group name="MQWebAdminGroup" realm="defaultRealm"/>
          </security-role>
        </application-bnd>
      </enterpriseApplication>
```

```
      <basicRegistry id="basic" realm="defaultRealm">
        <user name="USERNAME" password="PASSWORD"/>
        <group name="MQWebAdminGroup">
          <member name="USERNAME"/>
        </group>
      </basicRegistry>
      <sslDefault sslRef="mqDefaultSSLConfig"/>
    </server>
```

b) 옵션: 명령행을 사용하는 경우 ConfigMap를 적용하십시오.

```
oc apply -f mqwebuserconfigmap.yaml
```

나머지 단계에서는 다음 옵션 중 하나를 선택하십시오.

- IBM MQ Console에 액세스하기 위한 구성으로 새 큐 관리자를 배치하십시오.
- 기존 큐 관리자에 대한 IBM MQ Console 액세스를 제공하는 구성을 적용하십시오.

4. 옵션: **IBM MQ Console에 액세스하기 위한 구성으로 새 큐 관리자를 배치하십시오.**

a) 큐 관리자를 작성하십시오.

인증 및 권한 제공자를 수동 으로 설정하고 다음 옵션 중 하나를 통해 새로 작성된 ConfigMap mqwebuserconfigmap 를 제공하십시오.

- 옵션 1: 큐 관리자 YAML 사용

  큐 관리자 YAML의 web 섹션 아래에 다음 코드를 추가하십시오.

```
...
web:
    enabled: true
    console:
      authentication:
        provider: manual
      authorization:
        provider: manual
    manualConfig:
      configMap:
        name: mqwebuserconfigmap
```

- 옵션 2: OpenShift 콘솔 양식 보기를 통해 다음을 수행하십시오.

  i) OpenShift 콘솔에서 **운영자** > **설치된 운영자**를 선택하십시오.

  ii) IBM MQ Operator의 배치를 선택하십시오.

  iii) **큐 관리자** 를 선택하고 **QueueManager작성**을 클릭하십시오.

  iv) 큐 관리자에 대한 관련 옵션을 선택하십시오.

  v) **웹** 을 선택하고 **웹 서버 사용** 을 true로 설정하십시오.

  vi) **고급 구성** 목록 상자를 여십시오.

  vii) **콘솔** 목록 상자에서 **인증** 및 **권한 부여** 모두에 대한 **제공자** 를 수동으로 설정하십시오.

  viii) **구성** 목록 상자를 여십시오.

  ix) **ConfigMap** 목록 상자를 열고 89 페이지의 『3』 단계에서 작성된 ConfigMap mqwebuserconfigmap 을 선택하십시오.

  x) **작성**을 클릭하십시오.

이제 89 페이지의 『3』 단계에서 작성된 ConfigMap 에 지정된 신임 정보를 통해 새 큐 관리자의 IBM MQ Console 에 액세스할 수 있습니다.

5. 옵션: **기존 큐 관리자에 대해 IBM MQ Console 를 사용으로 설정하는 구성을 적용하십시오.**

IBM MQ Console를 사용하는 큐 관리자의 YAML을 편집하십시오.

a. OpenShift 콘솔에서 **운영자** > **설치된 운영자**를 선택하십시오.

b. IBM MQ Operator의 배치를 선택하십시오.

c. **큐 관리자** 를 선택하고 큐 관리자의 이름을 선택하십시오.

d. **YAML**을 선택하십시오.

e. 큐 관리자 YAML의 기존 web 섹션을 다음 코드로 대체하십시오.

```
...
web:
    enabled: true
    console:
      authentication:
        provider: manual
      authorization:
        provider: manual
    manualConfig:
      configMap:
        name: mqwebuserconfigmap
```

f. **저장** 을 클릭하십시오.

이제 89 페이지의 『3』 단계에서 작성된 ConfigMap 에 지정된 신임 정보를 통해 기존 큐 관리자의 IBM MQ Console 에 액세스할 수 있습니다.

OpenShift V 9.4.0 V 9.4.0 지속적 볼륨 확장

스토리지 제공자가 볼륨 확장을 지원하는 경우 이 태스크를 사용하여 지속적 볼륨을 확장하십시오. 스토리지 제공자에 따라 온라인 또는 오프라인으로 확장이 발생할 수 있습니다.

시작하기 전에

성공적인 볼륨 확장은 스토리지 제공자에 의존하여 확장 요청을 이행합니다. 온라인 크기 조정이 지원되는지 여부를 판별하고 오프라인 크기 조정 프로시저에 대한 정보를 보려면 스토리지 제공자 문서를 참조하십시오.

스토리지 제공자가 확장 요청을 이행할 수 없는 경우 지속적 볼륨 청구가 경고 또는 오류와 함께 상태가 될 수 있습니다. 확장에 실패하면 OpenShift 관리자가 지속적 볼륨 청구 상태를 수동으로 복구하고 확장을 취소할 수 있습니다. Red Hat OpenShift 문서에서 볼륨 확장 시 장애 복구 를 참조하십시오.

이 태스크 정보

지속적 스토리지 관리를 지원하기 위해 Kubernetes 는 두 개의 API 자원을 정의합니다.

- PersistentVolume (PV) - 관리자가 프로비저닝하거나 스토리지 클래스를 사용하여 동적으로 프로비저닝한 클러스터의 스토리지입니다. 정적으로 또는 동적으로 프로비저닝될 수 있습니다.
- 사용자가 스토리지를 요청하는 PersistentVolume청구 (PVC). 또한 자원에 대한 청구 검사 역할도 합니다.

자세한 정보는 Kubernetes 문서의 지속적 볼륨 을 참조하십시오.

경고:

- 큐 관리자 PVC를 작성하는 데 사용된 스토리지 클래스가 온라인 크기 조정을 지원하지 않으면 오프라인 크기 조정이 발생합니다. 오프라인 크기 조정 중에 볼륨 확장을 완료하려면 사용자 개입이 필요하므로 큐 관리자가 작동 중지 시간을 경험합니다.
- 다중 인스턴스 큐 관리자에 대한 공유 볼륨의 오프라인 크기 조정의 경우 사용자 개입을 수행할 때 활성 및 대기 팟 (Pod) 을 동시에 작동 중지해야 합니다.
- OpenShift 는 PVC의 크기 축소를 지원하지 않습니다. 지속적 볼륨의 크기를 줄이려고 하면 큐 관리자가 '실패' 상태가 됩니다.
- 이 프로시저는 임시 볼륨에 적용되지 않습니다.

IBM MQ 컨테이너에서 사용되는 PV를 확장하려면 다음 단계를 완료하십시오.

프로시저

1. **볼륨 확장 준비**

a) 확장할 볼륨을 결정하십시오.

b) 볼륨에서 사용 중인 하나 이상의 스토리지 클래스를 판별하십시오.

예를 들면, 다음과 같습니다.

```
spec:
  queueManager:
    storage:
      persistedData:
        enabled: true
        type: persistent-claim
        class: ocs-storagecluster-cephfs (1)
      queueManager:
        type: persistent-claim
      recoveryLogs:
        enabled: true
        type: persistent-claim
      defaultClass: ocs-storagecluster-ceph-rbd (2)
```

참고:

- (1) 볼륨이 특정 스토리지 클래스를 정의하는 경우 이 유형의 PVC에서 사용됩니다.
- (2) **defaultClass** 가 설정된 경우 이 스토리지 클래스는 특정 스토리지 클래스가 없는 모든 볼륨에 사용됩니다. **defaultClass** 가 설정되지 않고 볼륨 유형이 클래스를 지정하지 않은 경우 클러스터의 기본 스토리지 클래스가 사용됩니다.

기본 PVC를 설명하여 사용 중인 스토리지 클래스를 확인할 수도 있습니다. 예를 들면, 다음과 같습니다.

```
oc describe pvc pvc-name
```

c) 스토리지 클래스가 볼륨 확장을 지원하는지 확인하십시오.

스토리지 클래스에는 **.allowVolumeExpansion** 특성이 정의되어 있을 수 있습니다.

- 이 특성이 `true`로 설정되면 볼륨 확장이 지원됩니다.
- 이 특성이 `false`로 설정되거나 이 특성이 정의되지 않은 경우 스토리지 클래스는 볼륨 확장을 허용하지 않습니다. 이 경우 스토리지 제공자 문서를 참조하여 이 기능을 사용할 수 있는지 확인하십시오.

또한 스토리지 클래스를 설명하여 볼륨 확장을 지원하는지 여부를 판별할 수 있습니다. 예를 들면, 다음과 같습니다.

```
oc describe sc storage-class-name
```

d) 온라인 또는 오프라인 프로시저가 볼륨 확장에 사용되는지 확인하려면 스토리지 제공자 문서를 참조하십시오.

오프라인 프로시저에서는 큐 관리자 팟 (Pod) 을 수동으로 다시 시작해야 하지만 온라인 프로시저에서는 그렇지 않습니다. 오프라인 크기 조정 프로시저에 대해서는 스토리지 제공자 문서를 참조하십시오.

e) 큐 관리자에 이유가 'StorageMismatch' 인 상태 조건이 있는지 확인하십시오.

큐 관리자가 이 상태 조건을 갖는 경우, 볼륨 확장을 사용하면 조건에 나열된 볼륨이 확장됩니다. 이를 수행하지 않으려면 큐 관리자 정의의 각 볼륨 유형과 연관된 크기 필드를 프로비저닝된 PVC와 일치하도록 변경하십시오. 일치하지 않는 모든 볼륨에 대해 이를 수행하면 상태 조건이 제거됩니다.

2. **볼륨 확장**

경고:

- 이전에 큐 관리자 정의에서 볼륨 크기 필드를 수정한 경우, 큐 관리자 정의에서 **.allowVolumeExpansion** 가 `true` 로 설정되면 볼륨이 확장되기 시작합니다.
- 파일 시스템 제한사항 또는 로컬 하드웨어의 가용성으로 인해 스토리지 제공자에 최대 볼륨 크기에 대한 제한사항이 있을 수 있습니다. 실패를 방지하려면 볼륨을 확장하기 전에 스토리지 제공자 문서에서 이러한 제한사항의 유효성을 검증하십시오.
- PVC 크기의 감소는 OpenShift에서 지원되지 않습니다. 볼륨의 크기를 확장하면 볼륨을 줄일 수 없습니다. 이를 수행하려는 시도가 실패하면 IBM MQ Operator 는 PVC를 원래 상태로 되돌릴 수 없습니다.

볼륨 확장을 설명하는 큐 관리자 정의 예:

```
spec:
  queueManager:
    storage:
      allowVolumeExpansion: true (A)
      persistedData:
        enabled: true
        type: persistent-claim
        size: 3Gi (B)
      queueManager:
        type: persistent-claim
        size: 4Gi (B)
      recoveryLogs:
        enabled: true
        type: persistent-claim
        size: 3Gi (B)
```

a) 큐 관리자에 대한 볼륨 확장을 허용하려면 큐 관리자 의 **.spec.queueManager.storage.allowVolumeExpansion** (A) 필드를 true로 설정하십시오.

b) 이제 사용 가능한 볼륨 유형에 대한 크기 필드 (B) 를 늘릴 수 있습니다. 이러한 변경사항을 적용하면 볼륨 확장이 시작됩니다.

3. **PVC의 크기가 조정되었는지 확인하십시오.**

참고:

- 볼륨 확장에는 시간이 걸릴 수 있습니다. 유효성 검증에 실패하면 처음에 몇 분 동안 대기하고 다시 유효성 검증하는 것을 고려하십시오.
- 온라인 크기 조정이 수행될 때 사용자 조치 없이 볼륨 확장만 완료됩니다.
- 일부 스토리지 제공자는 사용자가 요청한 스토리지 크기를 반올림합니다. 확장된 볼륨의 크기는 사용자의 요청과 같거나 커야 합니다.

a) 큐 관리자에서 상태 조건을 확인하십시오. 조건, 설명 및 제안 조치에 대해서는 다음 표를 참조하십시오.

표 *1*. 보관 상태 조건

CONDITION	**메시지**	**설명**
StorageMismatch	Storage sizes defined in the QueueManager resource do not match the capacity of one or more provisioned PVCs [pvc-list]. AllowVolumeExpansion is set to false in the QueueManager resource so the MQ Operator will not attempt to reconcile these differences.	큐 관리자 정의에서 **.allowVolumeExpansion** 가 true 로 설정되지 않았으므로 볼륨 확장이 발생하지 않습니다.
StorageExpansionPending	Volume expansion is pending for the following PVCs [pvc-list]	볼륨 확장이 여전히 발생하고 있습니다. 이 상태 조건이 연장된 기간 동안 지속되면 오프라인 크기 조정 또는 크기 조정 실패가 발생할 수 있으므로 아래 단계를 수행하여 자세한 정보를 수집하십시오.

표 1. 보관 상태 조건 (계속)

CONDITION	메시지	설명
Failed	'Failed' 상태 조건을 작성할 수 있는 여러 가능한 스토리지 관련 메시지가 있습니다. 예: 'MQ Queue Manager failed to deploy: persistentvolumeclaims "<pvc>" is forbidden: only dynamically provisioned pvc can be resized and the storageclass the provisions the pvc must support resize.'	큐 관리자에 스토리지를 참조하는 텍스트가 있는 'Failed' 상태 조건이 있는 경우 상태 조건 내의 메시지를 참조하십시오. 여기에 제공된 예제 메시지는 확장을 지원하지 않는 스토리지 클래스를 사용하여 발생합니다.

b) 확장한 각 PVC에 대해 용량이 큐 관리자 정의에 지정된 값과 일치하거나 이보다 큰지 확인하십시오.

HA큐 관리자에는 각 유형의 여러 PVC가 있을 수 있습니다. PVC의 용량을 가져오려면 다음 명령을 실행하십시오.

```
oc get pvc pvc-name -o template --template '{{.status.capacity.storage}}'
```

c) PVC에 실패한 크기 조정을 제안하는 상태 조건 또는 이벤트가 없는지 확인하십시오.

```
oc describe pvc pvc-name
```

- PVC의 상태 조건은 FileSystemResizePending '사용자가 노드에서 볼륨의 파일 시스템 크기 조정을 완료하기 위해 팟 (Pod) 을 시작 (다시 시작) 하기를 기다리는 중' 입니다. 이 상태 조건은 온라인 및 오프라인 크기 조정 모두에 대해 발생합니다. 온라인 크기 조정의 경우 이 상태 조건은 온라인 크기 조정이 완료된 후 사용자 조치 없이 사라집니다.
- PVC에 실패한 크기 조정을 표시하는 이벤트 또는 상태 조건이 있는 경우 Red Hat OpenShift 문서에서 볼륨 확장 시 실패 복구 를 참조하십시오.

d) 큐 관리자 팟 (Pod) 에 실패한 크기 조정을 제안하는 상태 조건 또는 이벤트가 없는지 확인하십시오. HA 배치의 경우 각 복제본을 확인하십시오.

```
oc describe pod queue-manager-pod-name
```

- 팟 (Pod) 에 실패한 크기 조정을 표시하는 이벤트 또는 상태 조건이 있는 경우 Red Hat OpenShift 문서에서 볼륨 확장 시 실패 복구 를 참조하십시오. 오류 텍스트는 문제점을 해결하는 데 도움이 되거나 복구 후 다시 크기 조정을 시도하는 경우 동일한 문제점이 발생하지 않도록 방지할 수 있습니다.

4. **오프라인 크기 조정 시 팟 (Pod) 다시 시작**

스토리지 제공자가 볼륨을 확장할 때 오프라인 크기 조정 프로시저를 사용하는 경우 볼륨 확장을 완료하려면 크기 조정 중인 볼륨을 마운트하는 큐 관리자 팟 (Pod) 을 다시 시작해야 합니다.

다중 인스턴스 큐 관리자의 경우 복구 로그 및 지속된 데이터 볼륨은 활성 및 대기 팟 (Pod) 둘 다에서 공유됩니다. 이러한 볼륨의 크기를 조정하려면 두 팟 (Pod) 을 동시에 작동 중지하십시오.

오프라인 크기 조정 프로시저에 대해서는 스토리지 제공자 문서를 참조하십시오.

OpenShift V 9.4.0 V 9.4.0 큐 관리자 중지 (mq.ibm.com/stop)

큐 관리자 정의에 어노테이션을 추가하여 큐 관리자를 중지하십시오.

이 태스크 정보

IBM MQ 운영자가 작성한 큐 관리자에는 연관된 StatefulSet가 있습니다. 이 StatefulSet 는 '.replicas' 필드를 통해 지정된 큐 관리자 가용성 유형에 대해 배치될 Pods 의 수를 선언합니다. 이 값은 1 (단일 인스턴스), 2 (다중 인스턴스) 또는 3 (NativeHA) 입니다.

참고: '.replicas' 필드의 값을 수동으로 변경하면 큐 관리자가 올바르게 작동하지 않습니다.

일부 경우에는 StatefulSet 의 복제본 수가 0이고 Pods 가 배치되지 않도록 큐 관리자를 중지할 수 있습니다. 이를 수행하려는 경우의 예로는 유지보수 중 또는 백업 프로시저가 있습니다.

참고: 큐 관리자가 중지되었을 때 배치된 큐 관리자 Pods 가 없기 때문에 큐 관리자가 다시 시작될 때까지 사용자 및 애플리케이션이 큐 관리자에 액세스할 수 없습니다.

프로시저

- 큐 관리자를 중지하려면 '.metadata.annotations' 섹션 아래의 큐 관리자 정의에 다음 어노테이션을 추가하십시오.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: my-qm
  annotations:
    "mq.ibm.com/stop" : "true"
```

- 큐 관리자를 재시작하고 올바른 수의 복제본으로 되돌리려면 큐 관리자에서 어노테이션을 제거하거나 해당 값을 'false'' 로 설정하십시오.

Helm 을 사용하여 큐 관리자 배치 및 구성

샘플 Helm 차트를 사용하여 Kubernetes 에서 큐 관리자를 배치하고 구성할 수 있습니다.

이 태스크 정보

Red Hat OpenShift Container Platform를 사용하지 않는 경우 IBM MQ Operator 가 지원되지 않습니다. 샘플 Helm 차트를 사용하여 다른 유형의 Kubernetes 클러스터에 배치할 수 있습니다.

프로시저

- Helm 을 사용하여 자체 IBM MQ 컨테이너 이미지를 배치하는 방법에 대한 정보는 샘플 IBM MQ Helm 차트를 참조하십시오.

관련 참조

8 페이지의 『컨테이너에서 IBM MQ 에 대한 지원』
모든 IBM MQ 기능이 컨테이너에서 동일한 방식으로 사용 가능하고 지원되는 것은 아닙니다.

OpenShift CD CP4I-SC2 IBM MQ Operator 로 마이그레이션

이 주제 세트에서는 Red Hat OpenShift Container Platform의 IBM MQ Operator 를 사용하여 기존 IBM MQ 큐 관리자를 컨테이너 환경으로 마이그레이션하는 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 맞추며 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. 96 페이지의 『필수 기능이 사용 가능한지 확인』
2. 97 페이지의 『큐 관리자 구성 추출』
3. 옵션: 97 페이지의 『선택사항: 큐 관리자 키 및 인증서 추출 및 확보』
4. 옵션: 99 페이지의 『선택사항: LDAP 구성』
5. 옵션: 106 페이지의 『선택사항: IBM MQ 구성에서 IP 주소 및 호스트 이름 변경』
6. 108 페이지의 『컨테이너 환경을 위한 큐 관리자 구성 업데이트』
7. 110 페이지의 『컨테이너에서 실행 중인 IBM MQ의 대상 HA 아키텍처 선택』
8. 111 페이지의 『큐 관리자의 자원 작성』
9. 112 페이지의 『Red Hat OpenShift에서 새 큐 관리자 작성』
10. 116 페이지의 『새 컨테이너 배치 확인』

OpenShift CD CP4I-SC2 필수 기능이 사용 가능한지 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 95 페이지의 『IBM MQ Operator 로 마이그레이션』에서의 첫 번째 단계입니다.

프로시저

1. 대상 컨테이너 이미지에 필요한 모든 기능이 포함되는지 확인하십시오.

   최신 정보는 8 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 수정사항에 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 85 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.

16 페이지의 『컨테이너의 IBM MQ 에 대한 고가용성 계획』 의 내용을 참조하십시오.

다음에 수행할 작업
이제 큐 관리자 구성을 추출할 준비가 되었습니다.

OpenShift CD CP4I-SC2 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_NAME*을 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.

OpenShift CD CP4I-SC2 선택사항: 큐 관리자 키 및 인증서 추출 및 확보

TLS를 사용하여 큐 관리자에 대한 네트워크 트래픽을 암호화하도록 IBM MQ 를 구성할 수 있습니다. 이 태스크를 사용하여 큐 관리자가 TLS를 사용 중인지 확인하고 키 및 인증서를 추출하고 마이그레이션된 큐 관리자에서 TLS를 구성하십시오.

시작하기 전에
이 태스크에서는 사용자가 큐 관리자 구성을 추출했다고 간주합니다.

이 태스크 정보

수행 필요성

IBM MQ는 큐 관리자에 대한 트래픽을 암호화하도록 구성할 수 있습니다. 이 암호화는 큐 관리자에 구성된 키 저장소를 사용하여 완료됩니다. 그러면 IBM MQ 채널에서 TLS 통신을 사용할 수 있습니다. 사용자 환경에 TLS 통신이 구성되어 있는지 확실하지 않은 경우 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.

참고: 발행자 (CA) 인증서와 동일한 주제 식별 이름 (DN) 을 갖는 큐 관리자 인증서는 지원되지 않습니다. 인증서에는 고유한 주제 식별 이름이 있어야 합니다. 제품은 DN이 동일하지 않은지 확인합니다.

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 완활하게 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 runmqakmResult < <(runmqakm -cert -list -db ${keyrlocation}.kdb -stashed)
        cert=1
        for i in "${runmqakmResult[@]:2}"
        do
           certlabel=$(echo ${i:2} | xargs)
           echo Extracting certificate $certlabel to $cert.cert
           runmqakm -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. 큐 관리자의 TLS키 저장소 위치를 식별하십시오.

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

     샘플 출력:

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

     여기서 키 저장소는 `/run/runmqserver/tls/key.kdb`에 있습니다.

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

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

     샘플 출력:

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

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

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

이전에 표시된 샘플에서 이는 다음 명령과 같습니다.

```
runmqakm -cert -extract -db /run/runmqserver/tls/key.kdb -label "CN=cs-ca-
certificate,O=cert-manager" -target /tmp/cert-manager.crt -stashed
runmqakm -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.key`는 큐 관리자의 개인 키입니다.
- `qmgr.crt`는 공용 인증서입니다.

다음에 수행할 작업

이제 LDAP을 구성할 준비가 되었습니다.

OpenShift CD CP4I-SC2 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) +
*   ALTDATE(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.

OpenShift CD CP4I-SC2 LDAP 파트 1: LDAP 서버의 IBM MQ 백업 업데이트

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 99 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.

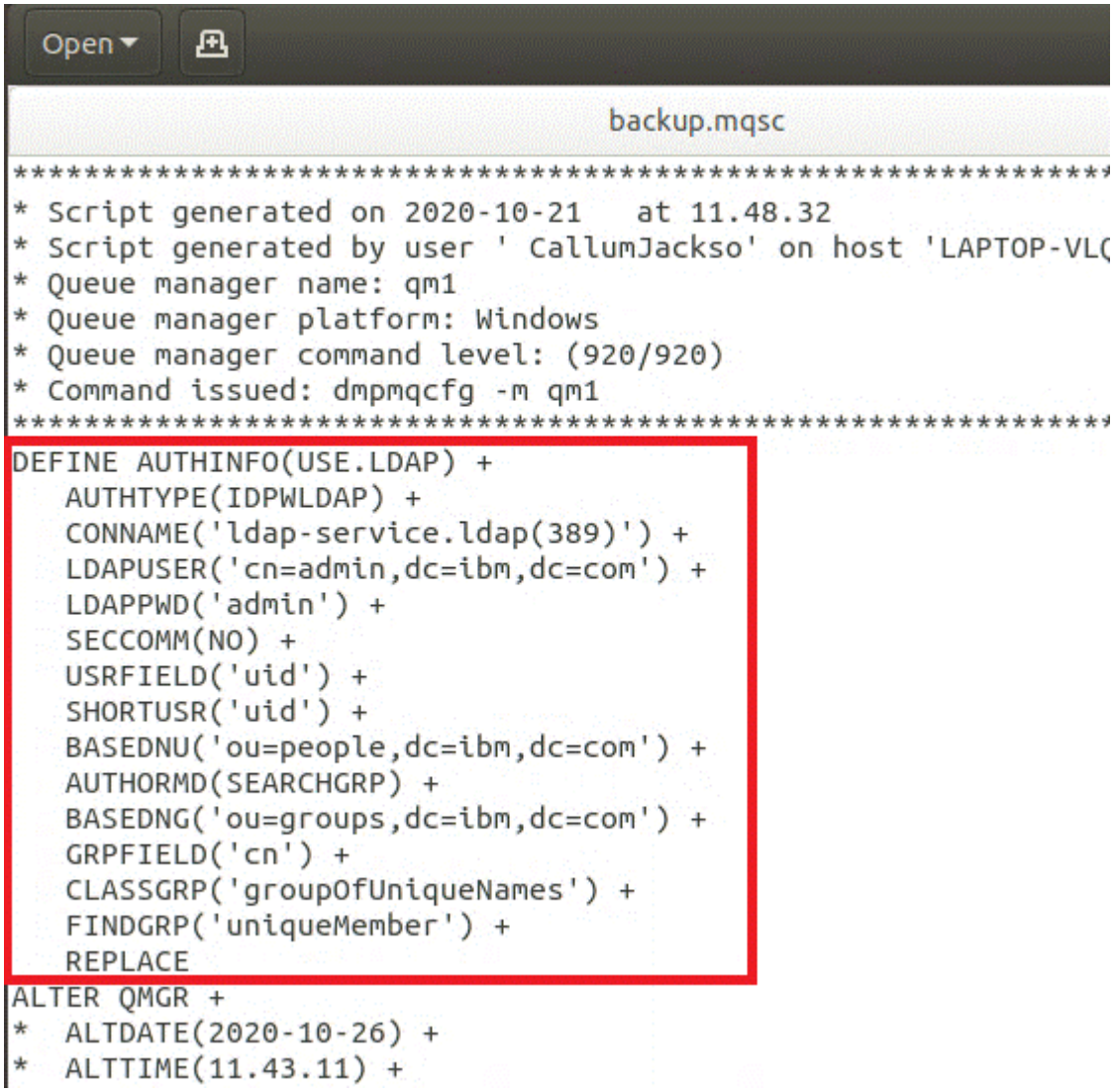
LDAP 시스템에 대한 새 **AUTHINFO** 오브젝트를 정의하도록 backup.mqsc 파일을 편집하십시오. 예를 들면, 다음과 같습니다.

```
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**은 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 사용자에 해당하는 식별 이름입니다.
- **LDAPPWD**는 **LDAPUSER** 사용자에 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 사용자에 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 사용자에 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.



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.

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: 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') +
*  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.

OpenShift CD CP4I-SC2 LDAP 파트 2: LDAP 권한 부여 정보의 IBM MQ 백업 업데이트

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 99 페이지의 『선택사항: LDAP 구성』의 내용을 참조하십시오.

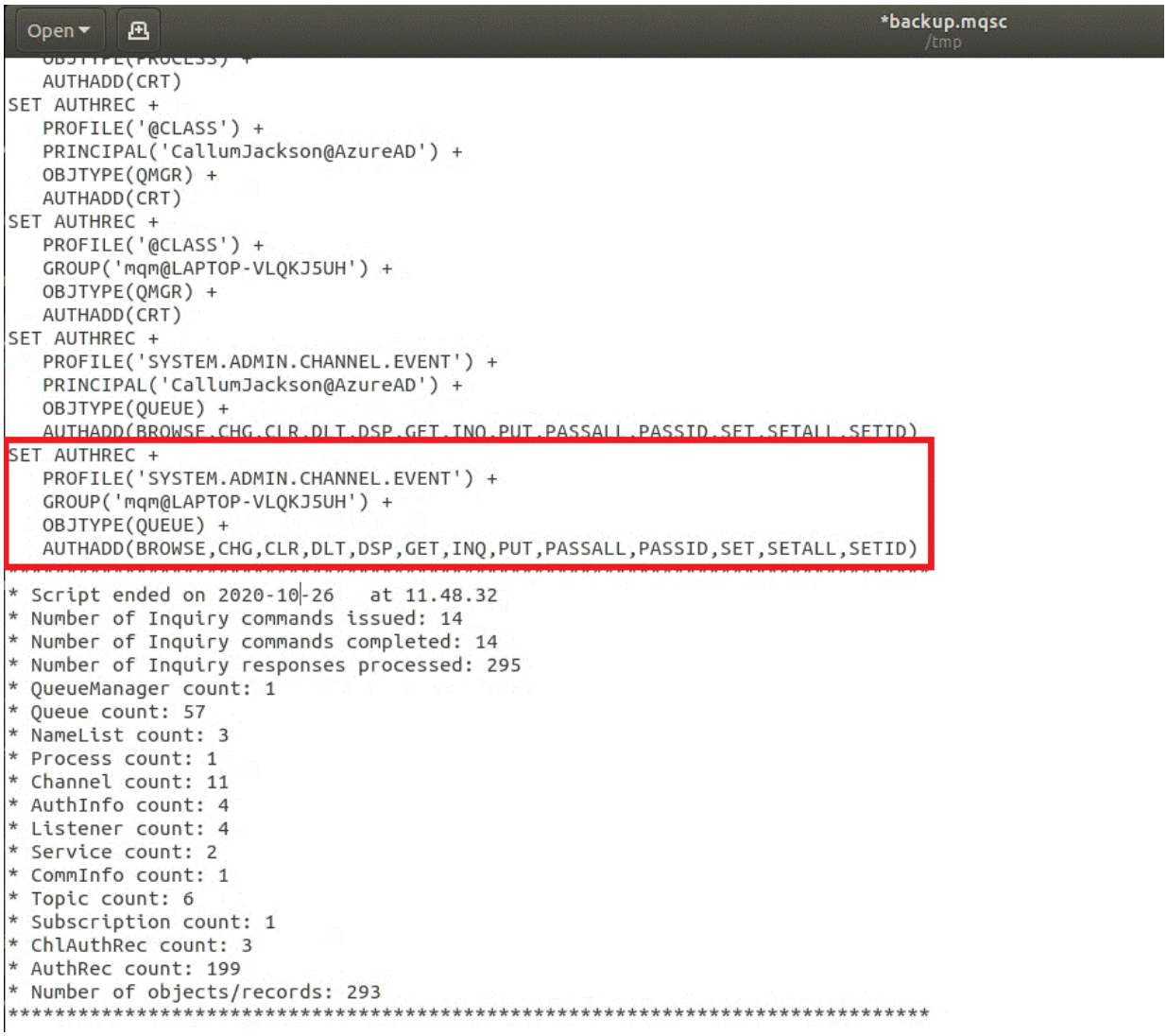
LDAP 권한 부여 정보 업데이트에는 두 파트가 있습니다.

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.



   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 필용한 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.

OpenShift CD CP4I-SC2 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 건너뛸 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.

   a. **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.

   b. **큐 관리자 종속 항목의 위치**: 이 큐 관리자가 알고 있어야 하는 기타 큐 관리자 또는 시스템의 위치.

   이 시나리오는 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 정의되어 있는지 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 CONNAME을 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 IPADDRV에 대한 항목이 있는지 확인하십시오. 있으면 삭제하십시오. 이는 컨테이너 환경과 관련이 없습니다.

다음에 수행할 작업

이제는 컨테이너 환경의 큐 관리자 구성을 업데이트할 준비가 되었습니다.

OpenShift CD CP4I-SC2 컨테이너 환경을 위한 큐 관리자 구성 업데이트

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.

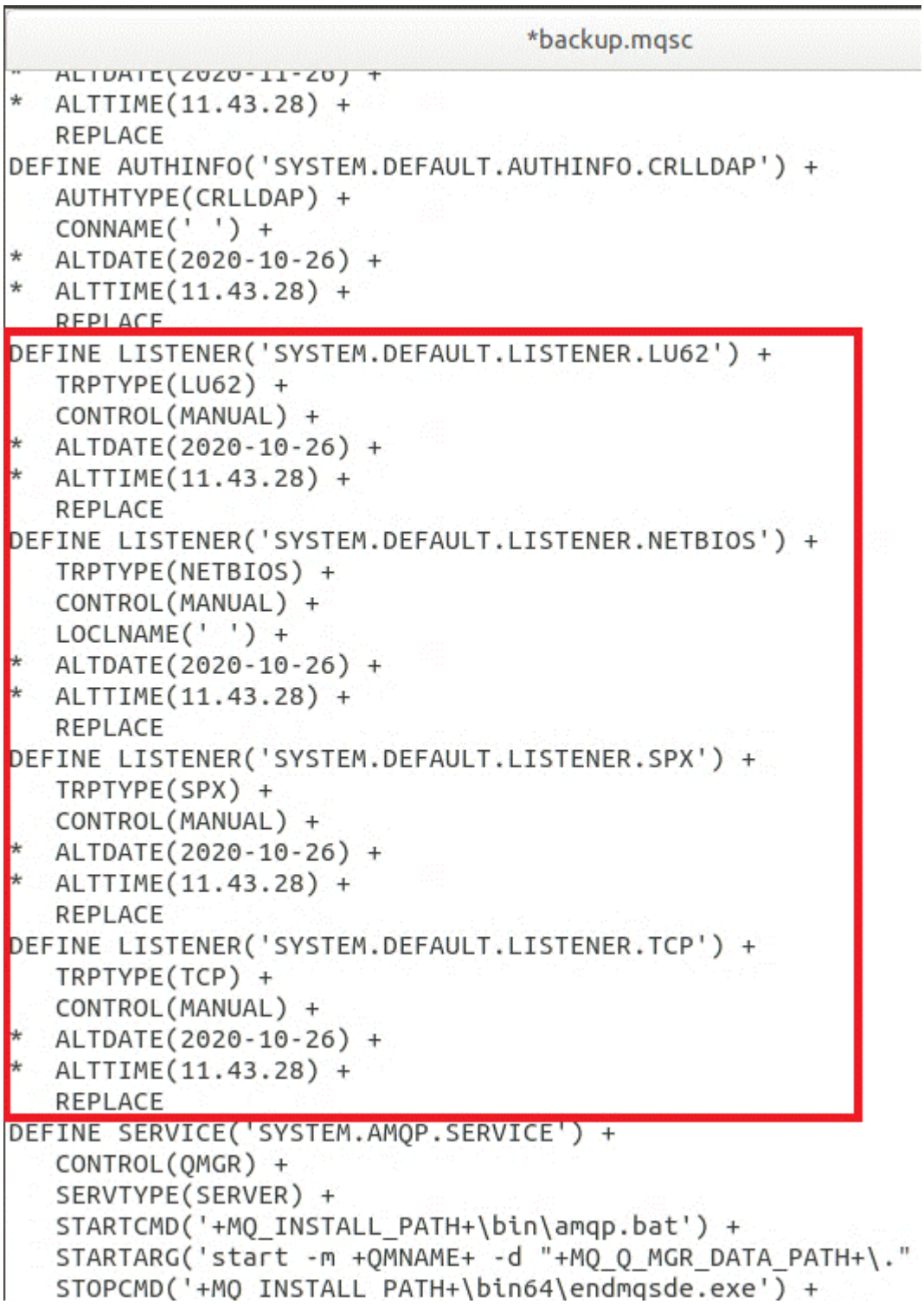
그러므로 내보낸 구성을 업데이트해야 합니다.

1. 리스너 정의 제거.
2. TLS 키 저장소의 위치 정의.

프로시저

1. 리스너 정의를 제거합니다.

   백업 구성에서 `DEFINE LISTENER`를 검색하십시오. 이는 `AUTHINFO`와 `SERVICE` 정의 사이에 있어야 합니다. 영역을 강조표시하고 삭제하십시오.



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 `default`로 변경하십시오.
   - TLS 키 저장소의 위치(**SSLKEYR**)를 `/run/runmqserver/tls/key`로 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) +
   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

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

OpenShift CD CP4I-SC2 컨테이너에서 실행 중인 IBM MQ의 대상 HA 아키텍처 선택

고가용성 요구사항을 충족시키기 위해 단일 인스턴스 (단일 Kubernetes 팟 (Pod)), 다중 인스턴스 (두 개의 팟 (Pod)) 및 원시 HA (하나의 활성 복제본 팟 (Pod) 및 두 개의 대기 복제본 팟 (Pod)) 중에서 선택하십시오.

시작하기 전에

이 태스크에서는 사용자가 컨테이너 환경의 큐 관리자 구성을 업데이트했다고 가정합니다.

이 태스크 정보

IBM MQ Operator 는 세 가지 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.
- **고유 HA**: 세 개의 컨테이너 (각각 별도의 팟 (Pod) 에 있음) 에 각각 큐 관리자의 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. 활성 큐 관리자를 실행하는 팟(Pod)이 실패하면, 큐 관리자의 복제본 인스턴스 중 하나가 활성 역할을 인계받고, 현재 데이터로 작업합니다.

이 태스크에서는 대상 HA 아키텍처만을 선택합니다. 선택한 아키텍처를 구성하는 단계에 대해서는 이 시나리오의 후속 태스크(112 페이지의 『Red Hat OpenShift에서 새 큐 관리자 작성』)에서 설명합니다.

프로시저

1. 세 가지 옵션을 검토하십시오.

   이러한 옵션에 대한 포괄적인 설명은 16 페이지의 『컨테이너의 IBM MQ 에 대한 고가용성 계획』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.

OpenShift CD CP4I-SC2 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 현재 디렉토리의 `backup.mqsc`파일에 IBM MQ 구성이 있다고 가정합니다. 그렇지 않으면 사용자 환경을 기반으로 파일 이름을 사용자 정의해야 합니다.

   a) `oc login`을 사용하여 클러스터에 로그인합니다.

   b) IBM MQ 구성을 `configmap`에 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.

97 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 `.key`로 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.crt`인 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.

OpenShift CD CP4I-SC2 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

110 페이지의 『컨테이너에서 실행 중인 IBM MQ의 대상 HA 아키텍처 선택』 에 설명된 대로 세 가지 가능한 배치 토폴로지가 있습니다. 따라서 이 주제에서는 세 가지 다른 템플리트를 제공합니다.

- 템플리트 1: 단일 인스턴스 큐 관리자 배치.
- 템플리트 2: 다중 인스턴스 큐 관리자 배치.
- 템플리트 3: 원시 HA큐 관리자 배치.

중요사항: 선호하는 토폴로지에 따라 세 개의 템플리트 중 하나만 완료하십시오.

프로시저

- **템플리트 1: 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: 9.4.0.0-r1
  license:
    accept: true
    license: L-BMSF-5YDSLR
    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: 9.4.0.0-r1
  license:
    accept: true
    license: L-BMSF-5YDSLR
    use: "Production"
```

  이는 배치를 위한 버전 및 라이센스 정보에 해당합니다. 이를 사용자 정의해야 하는 경우, 127 페이지의 『mq.ibm.com/v1beta1에 대한 라이센스 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 저

장소에 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 요구사항에 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`을(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-ebs   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`을(를) 수정해야 합니다.

```
  mqsc:
    - configMap:
```

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

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

- **템플리트 2: 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: 9.4.0.0-r1
  license:
    accept: true
    license: L-BMSF-5YDSLR
    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 `MultiInstance`이(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-ebs   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`을(를) 수정해야 합니다.

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

다중 인스턴스 큐 관리자를 배치했습니다. 이는 템플리트를 완성합니다. 이제는 새 컨테이너 배치 확인을 수행할 준비가 되었습니다.

- **템플리트 3: 원시 HA큐 관리자를 배치하십시오.**
  원시 HA큐 관리자를 작성하는 작업 예는 69 페이지의 『예제: IBM MQ Operator 를 사용하여 고유 HA 구성』의 내용을 참조하십시오.

OpenShift CD CP4I-SC2 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-0`은 112 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 MQSERVER=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

95 페이지의 『IBM MQ Operator 로 마이그레이션』를 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 토픽에 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. 76 페이지의 『Red Hat OpenShift 클러스터 외부에서 큐 관리자에 연결하도록 Route 구성』의 내용을 참조하십시오.

컨테이너에서 IBM MQ 작동

컨테이너에서 실행 중인 IBM MQ 큐 관리자를 조작하거나 상호작용해야 하는 경우 자세한 정보는 다음 주제를 참조하십시오.

프로시저

- 117 페이지의 『IBM MQ Operator 를 사용하여 IBM MQ 작동』.
- 125 페이지의 『고유 HA 큐 관리자의 상태 보기』.
- 126 페이지의 『수동으로 원시 HA큐 관리자 인스턴스 종료』.

OpenShift CP4I IBM MQ Operator 를 사용하여 IBM MQ 작동

프로시저

- 118 페이지의 『Red Hat OpenShift 클러스터에 배치된 IBM MQ Console에 연결』.
- 118 페이지의 『IBM MQ Operator 사용 시 모니터링』.
- 124 페이지의 『Red Hat OpenShift CLI를 사용하여 큐 관리자 구성 백업 및 복원』.

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 UI의 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 라이센스를 사용하는 경우 IBM MQ Console 는 ID및 액세스 관리를 위해 Keycloak 를 사용합니다. IBM Cloud Pak for Integration 문서의 ID및 액세스 관리 를 참조하십시오.

IBM MQ 라이센스를 사용하는 경우 IBM MQ Console 는 사전 구성되어 있지 않으므로 직접 구성해야 합니다. 자세한 정보는 사용자 및 역할 구성을 참조하십시오. 예에 대해서는 89 페이지의 『IBM MQ Operator 를 사용하여 기본 레지스트리로 IBM MQ Console 구성』 의 내용을 참조하십시오.

관련 태스크

76 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.

OpenShift CP4I *IBM MQ Console* 에 대한 권한 부여

IBM MQ Console 에 대한 권한은 라이센스 사용에 따라 다르게 관리됩니다.

이 태스크 정보

- IBM Cloud Pak for Integration 라이센스를 사용하는 경우 IBM MQ Console 는 ID및 액세스 관리를 위해 Keycloak 를 사용합니다.
  - IBM Cloud Pak for Integration 문서의 ID및 액세스 관리 를 참조하십시오.
  - 이전에 이전 버전의 IBM MQ Operator에서 IAM을 사용하여 사용자를 구성한 경우에는 IAM에서 Keycloak 로 사용자 마이그레이션을 참조하십시오.
- IBM MQ 라이센스를 사용하는 경우 IBM MQ Console 는 사전 구성되어 있지 않으므로 직접 구성해야 합니다.
  - 사용자 및 역할에 대한 자세한 정보는 사용자 및 역할 구성을 참조하십시오.
  - 간단한 예제는 89 페이지의 『IBM MQ Operator 를 사용하여 기본 레지스트리로 IBM MQ Console 구성』 의 내용을 참조하십시오.
  - 또는 이전에 설명한 대로 IBM Cloud Pak for Integration 운영자를 설치하여 Keycloak 를 구성할 수 있습니다.

OpenShift CP4I 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 사용으로 설정되지만 **.spec.metrics.enabled** 를 false로 설정하여 사용 안함으로 설정할 수 있습니다.

Prometheus는 메트릭을 위한 시계열 데이터베이스 및 규칙 평가 엔진입니다. IBM MQ 컨테이너는 Prometheus에서 조회할 수 있는 메트릭 엔드포인트를 공개합니다. 메트릭은 MQ 시스템 토픽에서 모니터링 및 활동 추적을 위해 생성합니다.

OpenShift Container Platform에는 Prometheus 서버를 사용하는 사전 구성되고, 사전 설치되고, 자체 업데이트되는 모니터링 스택이 포함되어 있습니다. OpenShift Container Platform 모니터링 스택은 사용자 정의 프로젝트를 모니터하도록 구성해야 합니다. 자세한 정보는 사용자 정의 프로젝트에 모니터링 사용을 참조하십시오. IBM MQ Operator은(는) Prometheus 운영자가 발견할 수 있는 메트릭을 사용하여 `QueueManager`을(를) 작성할 때 `ServiceMonitor`을(를) 작성합니다.

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	커미트 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
ibmmq_qmgr_expired_message_total	counter	만료된 메시지 수
ibmmq_qmgr_failed_browse_total	counter	실패한 찾아보기 수

메트릭	유형	설명
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
ibmmq_qmgr_log_logical_written_bytes_total	counter	로그 - 기록된 논리 바이트
ibmmq_qmgr_log_max_bytes	gauge	로그 - 최대 바이트

메트릭	유형	설명
ibmmq_qmgr_log_occupied_by_reusable_extents_bytes	gauge	로그 - 재사용 가능 범위에서 차지하는 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_required_for_media_recovery_bytes	gauge	로그 - 매체 복구에 필요한 바이트
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

관련 정보

시스템 주제에 게시된 메트릭

OpenShift CP4I 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

oc login을 사용하여 클러스터에 로그인합니다.

프로시저

- 큐 관리자 구성을 백업하십시오.

  dmpmqcfg 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 **dmpmqcfg** 명령을 실행하십시오.

  dmpmqcfg는 큐 관리자의 MQSC 구성을 출력합니다.

  ```
  oc exec -it pod_name -- dmpmqcfg > 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 컨텐츠 경로를 지정하여 팟(Pod)에서 **runmqsc** 명령을 실행하십시오.

  ```
  oc exec -i pod_name -- runmqsc < backup.mqsc
  ```

MQ Adv. 고유 HA 큐 관리자의 상태 보기

사용자 정의 빌드 컨테이너의 경우 **dspmq** 명령을 사용하여 고유 HA 인스턴스의 상태를 볼 수 있습니다.

이 태스크 정보

dspmq 명령을 사용하여 노드에서 큐 관리자 인스턴스의 운영 상태를 볼 수 있습니다. 리턴되는 정보는 인스턴스가 활성인지 또는 복제본인지에 따라 달라집니다. 활성 인스턴스에서 제공하는 정보는 결정적이며 복제본 노드의 정보는 이전 정보일 수 있습니다.

다음 조치를 수행할 수 있습니다.

- 현재 노드의 큐 관리자 인스턴스가 활성인지 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 활성 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 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 관리자 인스턴스가 활성 인스턴스로 실행 중인지 또는 복제본으로 실행 중인지를 판별합니다.

  ```
  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)
  ```

- 현재 노드에서 인스턴스의 원시 HA 운영 상태를 판별하려면 다음을 수행하십시오.

```
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.

```
dspmq -o nativeha -x -m QMgrName
```

큐 관리자 BOB의 활상 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) ALTDATE(2022-01-12) ALTTIME(12.03.44)
 INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
 INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(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) ALTDATE(2022-01-12) ALTTIME(12.03.44)
 INSTANCE(inst1) ROLE(Active)  REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
 INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(No)  BACKLOG(435)
CONNINST(Yes) ALTDATE(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) ALTDATE() ALTTIME()
 INSTANCE(inst2) ROLE(Unknown) REPLADDR(9.20.123.46) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATE() ALTTIME()
 INSTANCE(inst3) ROLE(Unknown) REPLADDR(9.20.123.47) CONNACTV(No) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATE() 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

MQ Adv. 수동으로 원시 HA큐 관리자 인스턴스 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 절에서 원시 HA큐 관리자 종료 를 참조하십시오.

OpenShift CP4I 컨테이너의 IBM MQ 에 대한 참조 정보

IBM MQ는 Red Hat OpenShift 컨테이너 플랫폼과의 기본 통합을 제공하는 Kubernetes Operator를 제공합니다.

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 CP4I-SC2 *mq.ibm.com/v1beta1*에 대한 라이센스 부여 참조

현재 라이센스 버전

`spec.license.license` 필드에는 승인하는 라이센스에 대한 라이센스 ID가 있어야 합니다. 유효값은 다음과 같습니다.

다음의 값 **spec.license.license**	다음의 값 **spec.license.use**	라이센스 정보	해당 IBM MQ 버전
L-JTPV-KYG8TF	Production 또는 NonProduction	IBM Cloud Pak for Integration 16.1.0	9.4.0
L-BMSF-5YDSLR	Production 또는 NonProduction	IBM Cloud Pak for Integration 제한 개정판 16.1.0	9.4.0
L-EHXT-MQCRN9	Production	IBM MQ Advanced 9.4	9.4.0
L-CLXQ-ADXTK3	Development	IBM MQ Advanced for Developers (무보증) 9.4	9.4.0

라이센스 버전이 지정되며, 항상 IBM MQ의 버전과 동일하지는 않습니다.

이전 라이센스 버전

IBM MQ 9.3 문서에서 이전 라이센스 버전 을 참조하십시오.

OpenShift CP4I *QueueManager*용 *API* 참조*(mq.ibm.com/v1beta1)*

QueueManager

QueueManager는 애플리케이션에 큐잉 및 발행/구독 서비스를 제공하는 IBM MQ 서버입니다. IBM MQ 문서: https://ibm.biz/BdPZqj. 라이센스 참조: https://ibm.biz/BdPZfq..

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.

필드	설명
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.

- 127 페이지의 『QueueManager』

필드	설명
affinity	표준 Kubernetes 연관관계 규칙입니다. 자세한 정보는https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#affinity-v1-core를 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-pod를 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 라이센스	라이센스의 승인을 제어하는 설정 및 사용할 라이센스 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.
securityContext SecurityContext	큐 관리자 팟(Pod)의 securityContext에 추가할 보안 설정입니다.
telemetry 텔레메트리	개방형 텔레메트리 구성에 대한 설정입니다. MQ 연산자 2.2.0 이상이 필요합니다.
template Template	Kubernetes 자원을 위한 고급 템플리트입니다. 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 템플리트의 설정으로 대체됩니다.

필드	설명
terminationGracePeriodSeconds 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 덮어씁니다. MQ Operator 1.3.0 이상이 필요합니다.

다음과 같이 표시됩니다.

- 128 페이지의 『.spec』

.spec.imagePullSecrets

LocalObjectReference는 동일한 네임스페이스 내에서 참조된 오브젝트를 찾을 수 있는 충분한 정보가 포함됩니다.

다음과 같이 표시됩니다.

- 128 페이지의 『.spec』

필드	설명
name 문자열	참조 이름입니다. 추가 정보: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names TODO: 다른 유용한 필드를 추가합니다. 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 덮어씁니다. MQ Operator 1.3.0 이상이 필요합니다.

다음과 같이 표시됩니다.

- 128 페이지의 『.spec』

.spec.license

라이센스의 승인을 제어하는 설정 및 사용할 라이센스 메트릭입니다.

다음과 같이 표시됩니다.

- 128 페이지의 『.spec』

필드	설명
accept boolean	이 소프트웨어와 연관된 라이센스에 동의하는지 여부입니다(필수).

필드	설명
`license` 문자열	승인 중인 라이센스의 ID입니다. 사용 중인 MQ의 버전으로 올바른 라이센스 ID여야 합니다. 올바른 값은 https://ibm.biz/BdPZfq 의 내용을 참조하십시오.
`metric` 문자열	사용할 라이센스 메트릭을 지정하는 설정입니다. 예를 들어, `ProcessorValueUnit`, `VirtualProcessorCore` 또는 `ManagedVirtualServer`입니다. MQ 라이센스를 사용하는 경우에는 기본적으로 `ProcessorValueUnit`으로, Cloud Pak for Integration 라이센스를 사용하는 경우에는 기본적으로 `VirtualProcessorCore`로 설정됩니다.
`use` 문자열	소프트웨어가 사용될 방법을 제어하는 설정이며, 여기서 라이센스는 다중 사용을 지원합니다. 올바른 값은 https://ibm.biz/BdPZfq 의 내용을 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.

- 128 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.

- 130 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.

- 130 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.

- 130 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.

- 130 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.

- 128 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 형식 메시지에 Basic 를 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.
name 문자열	metadata.name과 다른 경우 기본 MQ Queue Manager의 이름입니다. 이름에 대한 Kubernetes 규칙을 따르지 않는 큐 관리자 이름 (예: 대문자를 포함하는 이름) 을 원하는 경우 이 필드를 사용하십시오.

필드	설명
readinessProbe QueueManagerReadinessProbe	준비 프로브를 제어하는 설정입니다.
recoveryLogs RecoveryLogs	MQ 복구 로그 설정. MQ 연산자 2.4.0 이상이 필요합니다.
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.

- 131 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 Pods에 MultiInstance 를 사용하십시오. 이 중 하나는 active 큐 관리자이고 다른 하나는 대기입니다. 고유 고가용성 복제본의 경우 NativeHA를 사용합니다(MQ Operator 1.5.0 이상이 필요함). 기본적으로 SingleInstance로 설정됩니다. 세부사항은 http://ibm.biz/BdqAQa를 참조하십시오.
updateStrategy 문자열	MultiInstance 및 NativeHA 큐 관리자에 사용할 업데이트 전략입니다. 큐 관리자 구성이 변경될 때마다 자동 롤링 업데이트를 사용으로 설정하려면 RollingUpdate 를 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.

- 132 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.

- 131 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.

- 133 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.

- 133 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.

- 131 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.

- 131 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.

- 131 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.

- 134 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.

- 134 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.

- 131 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.recoveryLogs

MQ 복구 로그 설정. MQ 연산자 2.4.0 이상이 필요합니다.

다음과 같이 표시됩니다.

- 131 페이지의 『.spec.queueManager』

필드	설명
logFilePages 정수	복구 로그 데이터는 일련의 파일에 보유됩니다. 로그 파일 크기는 4KB 페이지 단위로 지정됩니다.

.spec.queueManager.resources

자원 요구사항을 제어하는 설정입니다.

다음과 같이 표시됩니다.

- 131 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.

- 135 페이지의 『.spec.queueManager.resources』

필드	설명
cpu	

필드	설명
memory	

.spec.queueManager.resources.requests

CPU & 메모리 설정입니다.

다음과 같이 표시됩니다.

- 135 페이지의 『.spec.queueManager.resources』

필드	설명
cpu	
memory	

.spec.queueManager.route

큐 관리자 라우트에 대한 설정입니다. MQ Operator 1.4.0 이상이 필요합니다.

다음과 같이 표시됩니다.

- 131 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.

- 131 페이지의 『.spec.queueManager』

필드	설명
failureThreshold 정수	프로브를 실패한 것으로 간주하는 최소 연속 실패 수입니다. 기본값은 24입니다.
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.

- 131 페이지의 『.spec.queueManager』

필드	설명
allowVolumeExpansion boolean	볼륨을 확장할 수 있는지 여부입니다.
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 boolean	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.
persistedData QueueManagerOptionalVolume	구성, 큐 및 메시지를 포함하는 MQ 지속 데이터에 대한 PersistentVolume 세부 사항입니다. 다중 인스턴스 큐 관리자를 사용할 때 필요합니다.
queueManager QueueManagerVolume	일반적으로 /var/mqm 아래 데이터에 대한 기본 PersistentVolume입니다. 다른 볼륨이 지정되지 않은 경우 모든 지속 데이터 및 복구 로그를 포함합니다.
recoveryLogs QueueManagerOptionalVolume	MQ 복구 로그에 대한 지속적 볼륨 세부사항입니다. 다중 인스턴스 큐 관리자를 사용할 때 필요합니다.
scratch 스크래치	큐 관리자의 스크래치 임시 볼륨에 대한 설정입니다. 이 볼륨은 컨테이너에서 '/run' 폴더로 마운트됩니다. 루트 파일 시스템이 읽기 전용으로 설정된 경우에만 적용 가능합니다. MQ Operator 3.0.0 이상이 필요합니다.
tmp Tmp	큐 관리자의 임시 볼륨에 대한 설정입니다. 이 볼륨은 컨테이너에서 '/tmp' 폴더로 마운트됩니다. runmqras 명령으로 생성된 zip 파일과 같은 진단 데이터 파일이 이 볼륨에 작성됩니다. 루트 파일 시스템이 읽기 전용으로 설정된 경우에만 적용 가능합니다. MQ Operator 3.0.0 이상이 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.

- 136 페이지의 『.spec.queueManager.storage』

필드	설명
class 문자열	이 볼륨에 사용할 스토리지 클래스입니다. type 이 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 boolean	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. type 이 persistent-claim인 경우에만 유효합니다. 예를 들어, 2Gi입니다. 기본적으로 2Gi로 설정됩니다.

필드	설명
sizeLimit 문자열	ephemeral 볼륨 사용 시 크기 제한. 파일은 여전히 임시 디렉토리에 기록되므로 이 옵션을 사용하여 크기를 제한할 수 있습니다. type 가 ephemeral 이고 루트 파일 시스템이 읽기 전용으로 설정된 경우에만 유효합니다. MQ Operator 3.0.0 이상이 필요합니다.
type 문자열	사용할 볼륨의 유형입니다. 비지속적 스토리지를 사용하려면 ephemeral를 선택하거나 지속적 볼륨을 사용하려면 persistent-claim을 선택하십시오. 기본적으로 persistent-claim으로 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.

- 136 페이지의 『.spec.queueManager.storage』

필드	설명
class 문자열	이 볼륨에 사용할 스토리지 클래스입니다. type 이 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 boolean	큐 관리자 삭제 시 이 볼륨을 삭제해야 하는지의 여부입니다.
size 문자열	SI 단위를 포함하여 Kubernetes로 전달할 PersistentVolume의 크기입니다. type 이 persistent-claim인 경우에만 유효합니다. 예를 들어, 2Gi입니다. 기본적으로 2Gi로 설정됩니다.
sizeLimit 문자열	ephemeral 볼륨 사용 시 크기 제한. 파일은 여전히 임시 디렉토리에 기록되므로 이 옵션을 사용하여 크기를 제한할 수 있습니다. type 가 ephemeral 이고 루트 파일 시스템이 읽기 전용으로 설정된 경우에만 유효합니다. MQ Operator 3.0.0 이상이 필요합니다.
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.

- 136 페이지의 『.spec.queueManager.storage』

필드	설명
class 문자열	이 볼륨에 사용할 스토리지 클래스입니다. type 이 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 boolean	큐 관리자 삭제 시 이 볼륨을 삭제해야 하는지의 여부입니다.

필드	설명
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. type이 persistent-claim인 경우에만 유효합니다. 예를 들어, 2Gi입니다. 기본적으로 2Gi로 설정됩니다.
sizeLimit 문자열	ephemeral 볼륨 사용 시 크기 제한. 파일은 여전히 임시 디렉토리에 기록되므로 이 옵션을 사용하여 크기를 제한할 수 있습니다. type 가 ephemeral 이고 루트 파일 시스템이 읽기 전용으로 설정된 경우에만 유효합니다. MQ Operator 3.0.0 이상이 필요합니다.
type 문자열	사용할 볼륨의 유형입니다. 비지속적 스토리지를 사용하려면 ephemeral를 선택하거나 지속적 볼륨을 사용하려면 persistent-claim을 선택하십시오. 기본적으로 persistent-claim으로 설정됩니다.

.spec.queueManager.storage.scratch

큐 관리자의 스크래치 임시 볼륨에 대한 설정입니다. 이 볼륨은 컨테이너에서 '/run' 폴더로 마운트됩니다. 루트 파일 시스템이 읽기 전용으로 설정된 경우에만 적용 가능합니다. MQ Operator 3.0.0 이상이 필요합니다.

다음과 같이 표시됩니다.

- 136 페이지의 『.spec.queueManager.storage』

필드	설명
sizeLimit 문자열	SI 단위를 포함한 임시 볼륨의 크기 한계입니다. 예를 들어, 2Gi입니다. 루트 파일 시스템이 읽기 전용으로 설정된 경우에만 유효합니다. MQ Operator 3.0.0 이상이 필요합니다.

.spec.queueManager.storage.tmp

큐 관리자의 임시 볼륨에 대한 설정입니다. 이 볼륨은 컨테이너에서 '/tmp' 폴더로 마운트됩니다. runmqras 명령으로 생성된 zip 파일과 같은 진단 데이터 파일이 이 볼륨에 작성됩니다. 루트 파일 시스템이 읽기 전용으로 설정된 경우에만 적용 가능합니다. MQ Operator 3.0.0 이상이 필요합니다.

다음과 같이 표시됩니다.

- 136 페이지의 『.spec.queueManager.storage』

필드	설명
sizeLimit 문자열	SI 단위를 포함한 임시 볼륨의 크기 한계입니다. 예를 들어, 2Gi입니다. 루트 파일 시스템이 읽기 전용으로 설정된 경우에만 유효합니다. MQ Operator 3.0.0 이상이 필요합니다.

.spec.securityContext

큐 관리자 팟(Pod)의 securityContext에 추가할 보안 설정입니다.

다음과 같이 표시됩니다.

- 128 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 `true`로 설정하십시오. 이를 `true`로 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.html의 내용을 참조하십시오.
`readOnlyRootFilesystem` boolean	큐 관리자에 대해 읽기 전용 루트 파일 시스템 설정을 사용할지 여부입니다. 기본적으로 false로 설정됩니다. MQ Operator 3.0.0 이상이 필요합니다.
`supplementalGroups` 배열	컨테이너의 기본 GID 외에 각 컨테이너에서 실행되는 첫 번째 프로세스에 적용된 그룹 목록입니다. 지정하지 않으면 어떤 그룹도 컨테이너에 추가되지 않습니다.

.spec.telemetry

개방형 텔레메트리 구성에 대한 설정입니다. MQ 연산자 2.2.0 이상이 필요합니다.

다음과 같이 표시됩니다.

- 128 페이지의 『.spec』

필드	설명
`tracing` 추적	텔레메트리 추적을 위한 설정입니다.

.spec.telemetry.tracing

텔레메트리 추적을 위한 설정입니다.

다음과 같이 표시됩니다.

- 140 페이지의 『.spec.telemetry』

필드	설명
`instana` 인스턴스	즉각적인 추적을 위한 설정입니다.

.spec.telemetry.tracing.instana

즉각적인 추적을 위한 설정입니다.

다음과 같이 표시됩니다.

- 140 페이지의 『.spec.telemetry.tracing』

필드	설명
`agentHost` 문자열	추적 데이터를 전송할 인스턴스화 에이전트의 호스트 이름입니다. 여기에는 프로토콜이 포함되지 않아야 합니다.
`enabled` boolean	인스턴스 추적을 사용할지 여부를 지정합니다. 기본적으로 false로 설정됩니다.

필드	설명
protocol 문자열	인스턴스 에이전트와 통신하는 데 사용할 프로토콜입니다. http및 https가 지원됩니다.

.spec.template

Kubernetes 자원을 위한 고급 템플리트입니다. 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 템플리트의 설정으로 대체됩니다.

다음과 같이 표시됩니다.

- 128 페이지의 『.spec』

필드	설명
pod	팟(Pod)에 사용된 템플리트를 대체합니다. https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#podspec-v1-core을 참조하십시오.

.spec.tracing

통합 운영 대시보드에 대한 클라우드 팩과의 통합을 위한 설정입니다.

다음과 같이 표시됩니다.

- 128 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.

- 141 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.

- 141 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.

- 141 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.

- 141 페이지의 『.spec.tracing』

필드	설명
image 문자열	사용할 컨테이너 이미지입니다.

필드	설명
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.

- 142 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.

- 142 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.web

MQ 웹 서버를 위한 설정입니다.

다음과 같이 표시됩니다.

- 128 페이지의 『.spec』

필드	설명
`console` 콘솔	MQ 웹 콘솔에 대한 설정입니다. MQ Operator 3.0.0 이상이 필요합니다.
`enabled` boolean	웹 서버 사용 가능 여부입니다. 기본적으로 false로 설정됩니다.
`manualConfig` ManualConfig	웹 서버 XML 구성을 제공하기 위한 설정입니다. MQ Operator 3.0.0 이상이 필요합니다.

.spec.web.console

MQ 웹 콘솔에 대한 설정입니다. MQ Operator 3.0.0 이상이 필요합니다.

다음과 같이 표시됩니다.

- 144 페이지의 『.spec.web』

필드	설명
`authentication` 인증	MQ 웹 콘솔에 대한 인증 설정입니다. MQ Operator 3.0.0 이상이 필요합니다.
`authorization` 권한 부여	MQ 웹 콘솔에 대한 권한 부여 설정입니다. MQ Operator 3.0.0 이상이 필요합니다.

.spec.web.console.authentication

MQ 웹 콘솔에 대한 인증 설정입니다. MQ Operator 3.0.0 이상이 필요합니다.

다음과 같이 표시됩니다.

- 144 페이지의 『.spec.web.console』

필드	설명
`provider` 문자열	MQ 웹 콘솔에 사용할 인증 제공자입니다. `integration-keycloak` 를 사용하여 Cloud Pak for Integration Platform UI (Keycloak) 에서 싱글 사인온을 사용하십시오. Cloud Pak for Integration 라이센스를 사용하는 경우 기본값은 `integration-keycloak` 이고, MQ 라이센스를 사용하는 경우 기본값은 `manual` 입니다. 사용자 고유의 구성을 제공하려면 `manual` 를 사용하십시오.

.spec.web.console.authorization

MQ 웹 콘솔에 대한 권한 부여 설정입니다. MQ Operator 3.0.0 이상이 필요합니다.

다음과 같이 표시됩니다.

- 144 페이지의 『.spec.web.console』

필드	설명
`provider` 문자열	MQ 웹 콘솔에 사용할 권한 제공자입니다. `integration-keycloak` 를 사용하여 Cloud Pak for Integration Keycloak에서 제공하는 역할을 사용하십시오. 사용자 고유의 구성을 제공하려면 `manual` 를 사용하십시오. Cloud Pak for Integration 라이센스를 사용하는 경우 기본값은 `integration-keycloak` 이고, MQ 라이센스를 사용하는 경우 기본값은 `manual` 입니다.

.spec.web.manualConfig

웹 서버 XML 구성을 제공하기 위한 설정입니다. MQ Operator 3.0.0 이상이 필요합니다.

다음과 같이 표시됩니다.

- 144 페이지의 『.spec.web』

필드	설명
`configMap` ConfigMap	ConfigMap 은 웹 서버 XML 구성을 포함하는 Kubernetes ConfigMap 을 나타냅니다.
`secret` Secret	시크릿은 웹 서버 XML 구성을 포함하는 Kubernetes 시크릿을 나타냅니다. 시크릿을 사용하면 Kubernetes 계층의 모든 신임 정보를 보호하지만 모니터링 또는 문제점 해결 도구가 기본 파일을 안전하지 않게 노출할 수 있습니다. 향상된 보안을 위해 "securityUtility를 사용하여 신임 정보를 인코딩하십시오.

.spec.web.manualConfig.configMap

ConfigMap 은 웹 서버 XML 구성을 포함하는 Kubernetes ConfigMap 을 나타냅니다.

다음과 같이 표시됩니다.

- 145 페이지의 『.spec.web.manualConfig』

필드	설명
`name` 문자열	Kubernetes 소스의 이름입니다.

.spec.web.manualConfig.secret

시크릿은 웹 서버 XML 구성을 포함하는 Kubernetes 시크릿을 나타냅니다. 시크릿을 사용하면 Kubernetes 계층의 모든 신임 정보를 보호하지만 모니터링 또는 문제점 해결 도구가 기본 파일을 안전하지 않게 노출할 수 있습니다. 향상된 보안을 위해 "securityUtility를 사용하여 신임 정보를 인코딩하십시오.

다음과 같이 표시됩니다.

- 145 페이지의 『.spec.web.manualConfig』

필드	설명
`name` 문자열	Kubernetes 소스의 이름입니다.

.status

QueueManager의 관찰된 상태입니다.

다음과 같이 표시됩니다.

- 127 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.

필드	설명
metadata 메타데이터	메타데이터는 통합-Keycloak 상태를 포함하여 큐 관리자에 대한 추가 정보를 나타냅니다.
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.

- 145 페이지의 『.status』

필드	설명
initialQuorumEstablished boolean	NativeHA에 대해 초기 쿼럼이 설정되었는지 여부입니다.

.status.conditions

QueueManagerStatusCondition은 Queue Manager의 조건을 정의합니다.

다음과 같이 표시됩니다.

- 145 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.

- 145 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.metadata

메타데이터는 통합-Keycloak 상태를 포함하여 큐 관리자에 대한 추가 정보를 나타냅니다.

다음과 같이 표시됩니다.

- 145 페이지의 『.status』

필드	설명
integrationKeycloak IntegrationKeycloak	QueueManagerStatusIntegrationKeycloak는 QueueManager에 대한 통합-Keycloak 상태를 정의합니다.

.status.metadata.integrationKeycloak

QueueManagerStatusIntegrationKeycloak는 QueueManager에 대한 통합-Keycloak 상태를 정의합니다.

다음과 같이 표시됩니다.

- 147 페이지의 『.status.metadata』

필드	설명
clientName 문자열	

.status.versions

사용 중인 MQ의 버전 및 IBM Entitled Registry에서 사용 가능한 기타 버전입니다.

다음과 같이 표시됩니다.

- 145 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.

- 147 페이지의 『.status.versions』

필드	설명
channels 배열	MQ 버전을 자동으로 업데이트하는 데 사용할 수 있는 채널입니다.
versions Versions 배열	사용할 수 있는 MQ의 특정 버전입니다.

.status.versions.available.versions

QueueManagerStatusVersion은 MQ의 버전을 정의합니다.

다음과 같이 표시됩니다.

- 147 페이지의 『.status.versions.available』

필드	설명
licenses 라이센스 배열	이 QueueManager버전에 적용할 수 있는 라이센스입니다.
name 문자열	이 버전의 QueueManager에 대한 버전 name . spec.version 필드에 올바른 값입니다.

.status.versions.available.versions.licenses

QueueManagerStatusLicense 는 라이센스를 정의합니다.

다음과 같이 표시됩니다.

- 147 페이지의 『.status.versions.available.versions』

필드	설명
displayName 문자열	라이센스에 대한 표시 이름입니다.
link 문자열	라이센스 컨텐츠에 링크합니다.
matchesCurrentType boolean	라이센스가 현재 사용되는 라이센스 유형과 일치하는지 여부.
name 문자열	라이센스의 이름입니다.

OpenShift CP4I *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.

QueueManager 자원에 대해서는 다음 상태가 정의되어 있습니다.

표 2. 큐 관리자 상태			
컴포넌트	**상태 유형**	**이유 코드**	**메시지 경고**
QueueManager[3]	차단됨	OperatorDependency	이 인스턴스를 설치하려면 [IBM Cloud Pak for Integration] 에서 Keycloak 을 구성해야 합니다. 이 인스턴스는 Keycloak 가 이 QueueManager에 대한 Cp4iServicesBinding 자원에서 [KeycloakReady] 로 보고될 때까지 [Pending] 상태로 유지됩니다.
			이 인스턴스를 설치하려면 [IBM IAM] 운영자가 필요합니다. 이 인스턴스는 [IBM Cloud Pak Foundational Services] 에서 운영자를 설치할 때까지 [차단됨] 상태로 유지됩니다.
	보류 중	작성 중	MQ 큐 관리자가 배치되는 중입니다.
	보류 중	OidcPending	MQ 큐 관리자가 OIDC 클라이언트 등록을 대기하고 있습니다.
	보류 중	중지됨	'mq.ibm.com/stop' 어노테이션이 있고 QueueManager 정의에서 'true' 로 설정되어 MQ 큐 관리자가 중지되었습니다. 중지되면 QueueManager StatefulSet 복제본 수가 0으로 설정되어 모든 MQ 큐 관리자 팟 (Pod) 을 제거합니다.
	오류	실패함	MQ 큐 관리자가 배치하는 데 실패했습니다.
	경고	UnsupportedVersion	OCP 버전 *<ocp_version>*에서 지원되지 않는 Operator가 Operand를 설치했습니다. 이 Operand는 지원되지 않습니다.
	경고	CP4I-LTS 지원	CP4I-LTS 피연산자 < *mq_version*> 이 (가) 설치되었지만 확장된 지원 지속 기간을 규정하지 않는 연산자에 의해 관리되고 있습니다. 이 Operand는 연장 지원 기간에 대해 적합하지 않습니다.
	경고	CP4I-LTS 지원	CP4I-LTS 피연산자 < *mq_version*> 이 (가) 설치되었지만 OCP 버전 < *ocp_version*> 이 (가) 확장된 지원 지속 기간을 규정하지 않습니다. 이 Operand는 연장 지원 기간에 대해 적합하지 않습니다.

[3] `Creating` 및 `Failed` 상태는 큐 관리자 배치의 전체 진행 상태를 모니터합니다. IBM Cloud Pak for Integration 라이센스를 사용 중이고 웹 콘솔이 사용으로 설정된 경우, `OidcPending` 조건은 OIDC 클라이언트 등록이 IAM 을 사용하여 완료될 때까지 대기하는 동안 큐 관리자의 상태를 로그합니다.

표 2. 큐 관리자 상태 (계속)			
컴포넌트	**상태 유형**	**이유 코드**	**메시지 경고**
팟(Pod)[4]	보류 중	PodPending	MQ 큐 관리자의 팟(Pod)이 배치되는 중입니다.
	오류	PodFailed	MQ 큐 관리자의 팟(Pod)이 배치되는 중입니다.
스토리지[5]	보류 중	StoragePending	MQ 큐 관리자의 스토리지가 프로비저닝되는 중입니다.
	경고	StorageEphemeral	프로덕션 MQ 큐 관리자에 대해 임시 스토리지를 사용하는 중입니다.
	경고	StorageExpansion보류 중	다음 PVC [< list of pvcs>] 에 대한 볼륨 확장이 보류 중입니다.
	경고	StorageMismatch	QueueManager 자원에 정의된 스토리지 크기가 하나 이상의 프로비저닝된 PVC [< list of pvcs>] 의 용량과 일치하지 않습니다. AllowVolume확장이 QueueManager 자원에서 false로 설정되므로 MQ 운영자가 이러한 차이를 조정하려고 시도하지 않습니다.
	오류	StorageFailed	MQ 큐 관리자의 스토리지를 프로비저닝하는 데 실패했습니다.

Linux IBM MQ 컨테이너 이미지를 직접 빌드하는 경우의 라이센스 어노테이션

라이센스 어노테이션은 기반 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 라이센스 부여에는 두 가지 접근법이 있습니다.

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

두 옵션 모두 클라이언트에서 사용할 수 있으며 추가 세부사항은 Passport Advantage®의 IBM 컨테이너 라이센스 페이지 에서 찾을 수 있습니다.

컨테이너 한계에 따라 IBM MQ 컨테이너에 라이센스가 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 배치되는 Kubernetes 클러스터에 설치되며 팟 (Pod) 어노테이션은 사용량을 추적하는 데 사용됩니다. 따라서 클라이언트는 IBM License Service에서 사용하는 특정 어노테이션과 함께 팟(Pod)을 배치해야 합니다. 컨테이너 내에 배치된 인타이틀먼트 및 기능에 따라 다음 어노테이션 중 하나 이상을 사용하십시오.

참고: 많은 어노테이션에는 다음 행 중 하나 또는 둘 다 포함되어 있습니다.

---

[4] Pod 상태는 큐 관리자 배치 중에 팟(Pod) 상태를 모니터합니다. `PodFailed` 상태가 표시되는 경우 전체 큐 관리자 상태도 `Failed`로 설정됩니다.

[5] Storage 상태는 지속적 스토리지의 볼륨을 작성하는 요청의 진행 상태(`StoragePending` 상태)를 모니터하고 바인딩 오류 및 기타 실패를 보고합니다. 스토리지 조건은 또한 볼륨 확장의 진행 상태를 모니터하고 큐 관리자 정의에 정의된 스토리지 크기와 배치된 PVC의 크기 간 불일치 경보를 모니터합니다. 스토리지 프로비저닝 중에 오류가 발생하면 `StorageFailed` 조건이 조건 목록에 추가되고 전체 큐 관리자 조건이 `Failed`로 설정됩니다.

```
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"
```

어노테이션을 사용하기 전에 다음 행을 편집해야 합니다.

- `productChargedContainers`의 경우 `"All"`를 선택하거나 컨테이너의 실제 이름을 대체해야 합니다.
- `productMetric`의 경우 제공된 값 중 하나를 선택해야 합니다.

IBM MQ 제품 인타이틀먼트와 함께 사용할 어노테이션

IBM MQ 제품 인타이틀먼트가 있는 경우 아래에서 구매하여 사용하려는 인타이틀먼트와 일치하는 어노테이션을 선택하십시오.

- 152 페이지의 『IBM MQ』
- 153 페이지의 『IBM MQ 고급』
- 153 페이지의 『비프로덕션 환경용 IBM MQ』
- 153 페이지의 『비프로덕션 환경용 IBM MQ Advanced』
- 153 페이지의 『개발자용 IBM MQ Advanced』

IBM MQ 다중 인스턴스 고가용성 구성에서 사용할 IBM MQ 어노테이션은 다음과 같습니다. 151 페이지의 『고가용성 구성에 대한 올바른 어노테이션 선택』도 참조하십시오.

- 153 페이지의 『IBM MQ 컨테이너 다중 인스턴스』
- 153 페이지의 『IBM MQ Advanced Container 다중 인스턴스』
- 153 페이지의 『IBM MQ 비프로덕션 환경의 컨테이너 다중 인스턴스』
- 154 페이지의 『비프로덕션 환경용 IBM MQ Advanced Container Multi Instance』

CP4I 제품 인타이틀먼트와 함께 사용할 어노테이션

IBM Cloud Pak for Integration (CP4I) 인타이틀먼트가 있는 경우 아래에서 구매하여 사용하려는 인타이틀먼트와 일치하는 어노테이션을 선택하십시오.

- 154 페이지의 『CP4I 인타이틀먼트가 있는 IBM MQ』
- 154 페이지의 『IBM MQ Advanced with CP4I 인타이틀먼트』
- 154 페이지의 『IBM MQ for Non-Production Environment with CP4I 권한』
- 154 페이지의 『IBM MQ Advanced for Non-Production Environment with CP4I 권한』

IBM MQ 다중 인스턴스 고가용성 구성에서 사용할 CP4I 어노테이션은 다음과 같습니다. 151 페이지의 『고가용성 구성에 대한 올바른 어노테이션 선택』도 참조하십시오.

- 154 페이지의 『IBM MQ CP4I 인타이틀먼트가 있는 컨테이너 다중 인스턴스』
- 155 페이지의 『IBM MQ Advanced Container Multi Instance with CP4I 인타이틀먼트』
- 155 페이지의 『IBM MQ CP4I 권한이 있는 비프로덕션 환경의 컨테이너 다중 인스턴스』
- 155 페이지의 『IBM MQ Advanced Container Multi Instance for Non-Production Environment with CP4I 권한』

고가용성 구성에 대한 올바른 어노테이션 선택

IBM MQ 다중 인스턴스

IBM MQ 다중 인스턴스 고가용성 구성에서 큐 관리자 쌍을 배치할 때 두 인스턴스 모두에서 동일한 어노테이션을 사용해야 합니다. 구입한 인타이틀먼트에 따라 다음 어노테이션 중 하나를 선택해야 합니다.

- IBM MQ 또는 IBM MQ Advanced 독립형 인타이틀먼트
  - 153 페이지의 『IBM MQ 컨테이너 다중 인스턴스』
  - 153 페이지의 『IBM MQ Advanced Container 다중 인스턴스』

  - 153 페이지의 『IBM MQ 비프로덕션 환경의 컨테이너 다중 인스턴스』
  - 154 페이지의 『비프로덕션 환경용 IBM MQ Advanced Container Multi Instance』
- IBM Cloud Pak for Integration 자격(entitlement)
  - 154 페이지의 『IBM MQ CP4I 인타이틀먼트가 있는 컨테이너 다중 인스턴스』
  - 155 페이지의 『IBM MQ Advanced Container Multi Instance with CP4I 인타이틀먼트』
  - 155 페이지의 『IBM MQ CP4I 권한이 있는 비프로덕션 환경의 컨테이너 다중 인스턴스』
  - 155 페이지의 『IBM MQ Advanced Container Multi Instance for Non-Production Environment with CP4I 권한』

IBM Cloud Pak for Integration 인타이틀먼트와 함께 사용하는 경우 어노테이션의 인타이틀먼트 비율은 올바른 인타이틀먼트 이용이 기록되는지 확인합니다. 독립형 IBM MQ 또는 IBM MQ Advanced 인타이틀먼트와 함께 사용하는 경우 각 인스턴스에 대해 License Service 에 보고된 어노테이션은 다음과 같이 IBM MQ 인타이틀먼트 파트에 맵핑되어야 합니다.

- IBM MQ Advanced container 다중 인스턴스
  - 1 x IBM MQ Advanced **및** 1 x IBM MQ Advanced 고가용성 복제본 **또는**
  - 2 x IBM MQ Advanced[6]
- IBM MQ Advanced container 비프로덕션 환경의 다중 인스턴스
  - 1 x IBM MQ Advanced **및** 1 x IBM MQ Advanced 고가용성 복제본 **또는**
  - 2 x IBM MQ Advanced (비프로덕션 환경의 경우)[6]
- IBM MQ 컨테이너 다중 인스턴스
  - 1 x IBM MQ **및** 1 x IBM MQ 고가용성 복제본 **또는**
  - 2 x IBM MQ[6]
- IBM MQ 비프로덕션 환경의 컨테이너 다중 인스턴스
  - 1 x IBM MQ **및** 1 x IBM MQ 고가용성 복제본 **또는**
  - 2 x IBM MQ (비프로덕션 환경의 경우) [6]

IBM MQ 기본 HA

고유 HA 쿼럼에 세 개의 큐 관리자를 배치하는 경우 활성 인스턴스만 인타이틀먼트를 이용합니다. 모든 인스턴스에는 동일한 어노테이션이 있어야 합니다. 구입한 인타이틀먼트에 따라 다음 중 하나를 선택해야 합니다.

- IBM MQ 또는 IBM MQ Advanced 독립형 인타이틀먼트
  - 153 페이지의 『IBM MQ 고급』
  - 153 페이지의 『비프로덕션 환경용 IBM MQ Advanced』
- IBM Cloud Pak for Integration 자격(entitlement)
  - 154 페이지의 『IBM MQ Advanced with CP4I 인타이틀먼트』
  - 154 페이지의 『IBM MQ Advanced for Non-Production Environment with CP4I 권한』

어노테이션

이 주제의 나머지 부분에서는 각 어노테이션의 컨텐츠에 대해 자세히 설명합니다.

IBM MQ

```
productID: "c661609261d5471fb4ff8970a36bccea"
productName: "IBM MQ"
```

[6] 이 인타이틀먼트 옵션은 차선이며 관련 고가용성 복제본 파트의 인타이틀먼트를 사용할 수 없는 경우에만 사용해야 합니다.

```
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

IBM MQ 고급

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

비프로덕션 환경용 IBM MQ

```
productID: "151bec68564a4a47a14e6fa99266deff"
productName: "IBM MQ for Non-Production Environment"
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

비프로덕션 환경용 IBM MQ Advanced

```
productID: "21dfe9a0f00f444f888756d835334909"
productName: "IBM MQ Advanced for Non-Production Environment"
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

개발자용 IBM MQ Advanced

```
productID: "2f886a3eefbe4ccb89b2adb97c78b9cb"
productName: "IBM MQ Advanced for Developers (Non-Warranted)"
productMetric: "FREE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

IBM MQ 컨테이너 다중 인스턴스

```
productID: "2dea73b866b648b6b4abe2a85eb76964"
productName: "IBM MQ Container Multi Instance"
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

IBM MQ Advanced Container 다중 인스턴스

```
productID: "bd35bff411bb47c2a3f3a4590f33a8ef"
productName: "IBM MQ Advanced Container Multi Instance"
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

IBM MQ 비프로덕션 환경의 컨테이너 다중 인스턴스

```
productID: "af11b093f16a4a26806013712b860b60"
productName: "IBM MQ Container Multi Instance for Non-Production Environment"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

비프로덕션 환경용 IBM MQ Advanced Container Multi Instance

```
productID: "31f844f7a96b49749130cd0708fdbb17"
productName: "IBM MQ Advanced Container Multi Instance for Non-Production Environment"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

CP4I 인타이틀먼트가 있는 IBM MQ

```
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
cloudpakName: "IBM Cloud Pak for Integration"
productID: "c661609261d5471fb4ff8970a36bccea"
productName: "IBM MQ"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productCloudpakRatio: "4:1"
```

IBM MQ Advanced with CP4I 인타이틀먼트

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

IBM MQ for Non-Production Environment with CP4I 권한

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

IBM MQ Advanced for Non-Production Environment with CP4I 권한

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

IBM MQ CP4I 인타이틀먼트가 있는 컨테이너 다중 인스턴스

```
productName: "IBM MQ Container Multi Instance"
productID: "2dea73b866b648b6b4abe2a85eb76964"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productCloudpakRatio: "10:3"
cloudpakName: "IBM Cloud Pak for Integration"
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

IBM MQ Advanced Container Multi Instance with CP4I 인타이틀먼트

```
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
cloudpakName: "IBM Cloud Pak for Integration"
productID: "bd35bff411bb47c2a3f3a4590f33a8ef"
productName: "IBM MQ Advanced Container Multi Instance"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productCloudpakRatio: "5:3"
```

IBM MQ CP4I 권한이 있는 비프로덕션 환경의 컨테이너 다중 인스턴스

```
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
cloudpakName: "IBM Cloud Pak for Integration"
productID: "af11b093f16a4a26806013712b860b60"
productName: "IBM MQ Container Multi Instance for Non-Production Environment"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productCloudpakRatio: "20:3"
```

IBM MQ Advanced Container Multi Instance for Non-Production Environment with CP4I 권한

```
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
cloudpakName: "IBM Cloud Pak for Integration"
productID: "31f844f7a96b49749130cd0708fdbb17"
productName: "IBM MQ Advanced Container Multi Instance for Non-Production Environment"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productCloudpakRatio: "10:3"
```

IBM MQ Advanced for Developers 컨테이너 이미지

IBM MQ Advanced for Developers에 대해 사전 빌드된 컨테이너 이미지를 사용할 수 있습니다. 이 이미지는 IBM Container Registry에서 사용 가능합니다. 이 이미지는 Docker, Podman, Kubernetes및 기타 컨테이너 환경에서 사용하기에 적합합니다.

사용 가능한 이미지

IBM MQ 이미지는 IBM Container Registry에 저장됩니다.

- IBM MQ Advanced for Developers 9.4.0.0: `icr.io/ibm-messaging/mq:9.4.0.0-r1`

빠른 참조

- 라이센스:
  - mq.ibm.com/v1beta1 및 Apache License 2.0에 대한 라이센싱 참조. IBM MQ Advanced for Developers 라이센스는 추가 배포를 허용하지 않으며, 조항은 개발자 머신에 대한 사용을 제한합니다.
- 문제를 보관할 위치:
  - GitHub
- 다음 CPU 아키텍처에 사용 가능합니다.
  - amd64
  - s390x
  - ppc64le

사용법

컨테이너에서 IBM MQ Advanced for Developers 를 실행하십시오.

컨테이너 실행 방법에 대한 세부사항은 사용 문서 를 참조하십시오.

이미지를 사용할 수 있으려면 **LICENSE** 환경 변수를 설정하여 IBM MQ 라이센스의 이용 약관에 동의해야 합니다.

지원되는 환경 변수

LANG
라이센스를 인쇄할 언어를 설정하십시오.

라이센스
IBM MQ Advanced for Developers 라이센스 조건에 동의하도록 동의 를 설정하십시오.

라이센스 조건을 보려면 보기 를 설정하십시오.

Deprecated **MQ_ADMIN_PASSWORD**
관리 사용자의 비밀번호를 지정하십시오.

8자이상이어야 합니다.

관리 사용자에 대한 기본 비밀번호가 없습니다.

V 9.4.0 V 9.4.0 IBM MQ 9.4.0부터 이 변수는 더 이상 제공되지 않습니다. 이 주제의 YAML예 에서는 이 변수를 직접 작성하고 시크릿으로 보안하는 방법을 보여줍니다.

Deprecated **MQ_APP_PASSWORD**
앱 사용자의 비밀번호를 지정하십시오.

이 옵션을 설정하면 **DEV.APP.SVRCONN** 채널이 보안되고 올바른 사용자 ID및 비밀번호를 제공하는 연결만 허용합니다.

8자이상이어야 합니다.

앱 사용자의 기본 비밀번호가 없습니다.

V 9.4.0 V 9.4.0 IBM MQ 9.4.0부터 이 변수는 더 이상 제공되지 않습니다. 이 주제의 YAML예 에서는 이 변수를 직접 작성하고 시크릿으로 보안하는 방법을 보여줍니다.

MQ_DEV
작성 중인 기본 오브젝트를 중지하려면 `false` 를 설정하십시오.

MQ_ENABLE_METRICS
큐 관리자에 대한 Prometheus 메트릭을 생성하려면 `true` 를 설정하십시오.

MQ_LOGGING_CONSOLE_SOURCE
컨테이너의 **stdout** 위치에 미러링되는 로그의 쉼표로 구분된 소스 목록을 지정하십시오.

올바른 값은 `qmgr`, `web` 및 `mqsc`입니다.

기본값은 `qmgr, web`입니다.

선택적 값은 `mqsc`입니다. 이 옵션을 사용하여 컨테이너 로그에서 `autocfgmqsc.LOG` 의 컨텐츠를 반영할 수 있습니다.

MQ_LOGGING_CONSOLE_FORMAT
컨테이너의 **stdout** 위치에 인쇄되는 로그의 형식을 변경하십시오.

사용자가 읽을 수 있는 단순 형식을 사용하려면 `basic` 을 설정하십시오. 이는 기본값입니다.

JSON 형식 (각 행에 하나의 JSON 오브젝트) 을 사용하도록 `json` 을 설정하십시오.

MQ_LOGGING_CONSOLE_EXCLUDE_ID
제외되는 로그 메시지에 대한 쉼표로 구분된 메시지 ID 목록을 지정하십시오.

로그 메시지는 여전히 디스크의 로그 파일에 표시되지만 컨테이너의 **stdout** 위치에는 인쇄되지 않습니다.

기본값은 `AMQ5041I,AMQ5052I,AMQ5051I,AMQ5037I,AMQ5975I`입니다.

mq_qmgr_name
큐 관리자를 작성하는 데 사용할 이름을 설정하십시오.

IBM MQ Advanced for Developers 이미지에서 지원하는 기본 개발자 구성에 대한 자세한 정보는 기본 개발자 구성 문서를 참조하십시오.

admin 및 app 사용자의 비밀번호를 지정하는 방법을 설명하는 예제 큐 관리자 YAML

admin 및 **app** 사용자 ID 사용자의 경우 Development 라이센스를 사용하여 큐 관리자를 배치할 때 비밀번호를 제공해야 합니다. 다음은 IBM MQ Operator에서 이를 수행하는 방법을 보여주는 큐 관리자 YAML예입니다.

다음 명령은 **admin** 및 **app** 사용자의 비밀번호를 포함하는 시크릿을 작성합니다.

```
oc create secret generic my-mq-dev-passwords --from-literal=dev-admin-password=passw0rd --from-
literal=dev-app-password=passw0rd
```

다음 YAML은 큐 관리자를 배치할 때 이러한 비밀번호를 사용합니다.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: qm-dev
spec:
  license:
    accept: false
    license: L-CLXQ-ADXTK3
    use: Development
  web:
    enabled: true
  template:
    pod:
      containers:
        - env:
            - name: MQ_DEV
              value: "true"
            - name: MQ_CONNAUTH_USE_HTP
              value: "true"
            - name: MQ_ADMIN_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: my-mq-dev-passwords
                  key: dev-admin-password
            - name: MQ_APP_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: my-mq-dev-passwords
                  key: dev-app-password
          name: qmgr
  queueManager:
    storage:
      queueManager:
        type: persistent-claim
    name: QUICKSTART
  version: 9.4.0.0-r1
```

컨테이너에서 IBM MQ 문제점 해결

컨테이너에서 IBM MQ 를 실행하는 데 문제가 있는 경우 여기에 설명된 기술을 사용하여 문제를 진단하고 해결할 수 있습니다.

프로시저

- 157 페이지의 『컨테이너에서 IBM MQ 의 계획되지 않은 다시 시작 문제점 해결』.
- 159 페이지의 『IBM MQ Operator 의 문제점 해결』.

OpenShift CP4I Kubernetes 컨테이너에서 IBM MQ 의 계획되지 않은 다시 시작 문제점 해결

대부분의 컨테이너 관리 시스템 (예: Red Hat OpenShift Container Platform 및 Kubernetes) 에서 컨테이너는 일반적으로 다시 시작됩니다. 컨테이너가 오래 지속되는 것은 정상적이지 않습니다. 이 주제에서는 컨테이너 라이프사이클, 다시 시작을 조사하는 방법 및 계획되지 않은 컨테이너 다시 시작 이면의 이유에 대해 설명합니다.

IBM MQ 배치에 문제가 없고 예상대로 계속 실행되는 경우 솔루션이 의도한 대로 수행되고 있을 수 있습니다. 컨테이너 로그에서 다음과 같은 로그 메시지를 볼 수 있습니다.

```
Signal received: terminated
```

이는 SIGTERM 신호가 MQ 컨테이너로 전송되어 종료하도록 요청함을 의미합니다. Linux 컨테이너는 작동을 트리거하기 위해 프로그램으로 전송되는 표준화된 메시지인 POSIX 신호에 응답할 책임이 있습니다.

IBM MQ 컨테이너가 SIGTERM 신호를 수신하면 `endmqm -w -r -tp` 명령을 실행하여 큐 관리자를 중지합니다. 큐 관리자가 중지되면 컨테이너가 중지됩니다. 큐 관리자를 중지하는 데 시간이 오래 걸리면 SIGKILL 신호가 전송되어 Linux 프로세스를 즉시 종료합니다. SIGTERM과 SIGKILL 사이의 시간은 Kubernetes에서 "종료 유예 기간" 으로 알려져 있으며 `QueueManager` 자원 (IBM MQ Operator를 사용하는 경우) 또는 팟 (Pod) 자원에서 직접 구성할 수 있습니다. 기본값은 30초이며, 이 중 1초는 컨테이너를 종료하기 위해 예약되고 나머지는 IBM MQ에 제공됩니다. 예를 들어, 기본적인 경우에는 `endmqm -w -tp 29` 가 발행되며, 이는 큐 관리자에게 종료하는 데 29초가 소요되었음을 알려줍니다.

팟 (Pod) 제거 이유

SIGTERM 신호는 Kubernetes (및 Red Hat OpenShift Container Platform) 에서 팟 (Pod) 을 단계적으로 종료하는 데 사용됩니다. Kubernetes 문서에서 팟 (Pod) 종료 를 참조하십시오. Kubernetes 는 노드의 팟 (Pod) 이 자발적으로 또는 비자발적으로 종료되는 프로세스에 대해 "팟 (Pod) 중단 및" 축출 " 용어를 사용합니다. 다음을 포함하여 팟 (Pod) 이 축출될 수 있는 여러 가지 이유가 있습니다.

- **kubelet에 의한 종료**. 이는 다음과 같은 여러 가지 이유 때문일 수 있습니다.
  - 노드가 종료 중이므로 (롤링 클러스터 업데이트의 일부로) 팟 (Pod) 을 종료할 수 있습니다.
  - 팟 (Pod) 은 노드 " 압력 (여기서 kubelet은 노드에서 리소스를 재확보하기 위해 사전에 팟 (Pod) 을 종료함) 으로 인해 종료될 수 있습니다. Kubernetes 클러스터 관리자는 클러스터 간에 다를 수 있는 제거 임계값을 구성할 수 있습니다.
  - 팟 (Pod) 이 활성 상태 프로브에 실패했으므로 팟 (Pod) 을 종료할 수 있습니다. Kubernetes 에서 활성 상태 프로브를 구성하여 팟 (Pod) 이 여전히 양호한 상태인지 확인할 수 있습니다. IBM MQ Operator 는 **dspmq** 명령을 호출하여 올바른 실행 상태를 확인하는 큐 관리자 활성 상태 프로브를 설정합니다. 큐 관리자가 양호한 상태가 아니거나 프로브 자체를 실행하는 데 시간이 너무 오래 걸리는 경우 kubelet은 해당 실패를 고려합니다. 허용할 실패 수에 대한 임계값은 `QueueManager` 자원 (IBM MQ Operator를 사용하는 경우) 또는 팟 (Pod) 자원에서 직접 구성할 수 있습니다.
- **Kubernetes 스케줄러에 의한 선점**. 이는 Kubernetes 스케줄러가 더 높은 우선순위 팟 (Pod) 을 실행해야 하는 경우에 발생할 수 있습니다.
- **감염된 노드**. 노드는 "감염" 될 수 있으며 감염을 허용하지 않는 팟 (Pod) 은 제거됩니다. 감염은 Kubernetes 관리자가 특정 노드에서 팟 (Pod) 을 "제거" 하는 데 사용됩니다. 예를 들어, IBM MQ 팟 (Pod) 이 이제 다른 워크로드에 예약된 특수 하드웨어가 있는 노드에서 더 이상 실행되지 않아야 합니다.
- **축출 API를 통한 요청**. 이는 관리자가 팟 (Pod) 을 제거하기 위해 호출할 수 있습니다.
- **팟 (Pod) 가비지 콜렉션**. 이는 노드가 서비스되지 않거나 Kubernetes API를 통해 제거되는 경우에 발생할 수 있습니다.

큐 관리자 팟 (Pod) 이 제거된 이유 판별

팟 (Pod) 이 제거된 이유를 이해하는 데 도움이 되는 잠재적 정보 소스는 다음과 같습니다.

- **클러스터 이벤트**. 예를 들어, OpenShift Container Platform의 시스템 이벤트 정보 보기 클러스터입니다.
- **클러스터 감사 이벤트**. Red Hat OpenShift Container Platform에서 감사 로그 보기를 참조하십시오.
- **압력이 있는 노드**. CPU, 네트워크 또는 메모리 압력 아래에서 노드를 찾으십시오. 노드 상태에서 이를 볼 수 있습니다. 노드를 보게 되면 노드에 더 이상 압력이 가해지지 않을 수 있습니다.
- **Red Hat OpenShift Container Platform Monitoring** 또는 기타 모니터링 메트릭은 디스크 대기 시간 문제와 같은 것을 표시할 수 있습니다. 유용한 Prometheus 메트릭은 `ibmmq_qmgr_log_write_latency_seconds`입니다. 이 정보는 MQ 통계 토픽에서 제공됩니다.

관련 정보

스케줄링, 선점 및 축출에 대한 Kubernetes 문서

OpenShift CP4I IBM MQ Operator 의 문제점 해결

IBM MQ Operator에 문제가 있는 경우에는 설명된 기술을 사용하여 문제를 진단하고 해결하십시오.

프로시저

- 159 페이지의 『IBM MQ Operator 로 배치된 큐 관리자에 대한 문제점 해결 정보 수집』
- 161 페이지의 『문제점 해결: 큐 관리자 데이터에 대한 액세스 수집』

OpenShift CP4I IBM MQ Operator 로 배치된 큐 관리자에 대한 문제점 해결 정보 수집

새 지원 케이스를 제기할 때 IBM 지원 센터에 제공해야 하는 문제점 해결 정보 수집.

프로시저

1. 클라우드 제공자 정보를 수집합니다.

   이는 Red Hat OpenShift 클러스터를 호스팅하는 클라우드 제공자입니다 (예: IBM Cloud).
2. 아키텍처 정보를 수집하십시오.

   Red Hat OpenShift 클러스터의 아키텍처는 다음 중 하나입니다.

   - Linux for x86-64
   - Linux on Power Systems (ppc64le)
   - Linux for IBM Z
3. IBM MQ 배치 정보를 수집하십시오.

   a) bash/zsh 쉘을 사용하여 Red Hat OpenShift 클러스터에 로그온하십시오.

   b) 다음 환경 변수를 설정하십시오.

```
export QM=QueueManager_name
export QM_NAMESPACE=QueueManager_namespace
export MQ_OPERATOR_NAMESPACE=mq_operator_namespace
```

   여기서 *QueueManager_name* 은 QueueManager 자원의 이름이고, *QueueManager_namespace* 는 배치된 네임스페이스이며, *mq_operator_namespace* 는 IBM MQ Operator 가 배치된 네임스페이스입니다. 이는 QueueManager 네임스페이스와 동일할 수 있습니다.

   c) 다음 명령을 실행하고 모든 결과 출력 파일을 IBM 지원 센터에 제공하십시오.

```
# OCP / Kubernetes: Version
oc version -o yaml > ocversion.yaml

# QueueManager: YAML
oc get qmgr $QM -n $QM_NAMESPACE -o yaml > "queue-manager-$QM.yaml"

# MQ Queue Manager: Pods
oc get pods -n $QM_NAMESPACE -o wide --selector "app.kubernetes.io/instance=$QM" > "qm-
pods-$QM.txt"

# MQ Queue Manager: Pod YAML
oc get pods -n $QM_NAMESPACE -o yaml --selector "app.kubernetes.io/instance=$QM" > "qm-
pods-$QM.yaml"

# MQ Queue Manager: Pod Logs
for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/
instance=$QM" | cut -d ' ' -f 1); do oc logs -n $QM_NAMESPACE --previous "$p" > "qm-logs-
previous-$p.txt"; oc logs -n $QM_NAMESPACE $p > "qm-logs-$p.txt";done

# MQ Queue Manager: Describe Pods
for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/
instance=$QM" | cut -d ' ' -f 1); do oc describe pod $p -n $QM_NAMESPACE > "qm-pod-
```

```
describe-$p.txt"; done

# MQ Web UI: Console Log
for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/
instance=$QM" | cut -d ' ' -f 1); do oc cp -n $QM_NAMESPACE --retries=10 "$p:var/mqm/web/
installations/Installation1/servers/mqweb/logs/console.log" "web-$p-console.log"; done

# MQ Web UI: Messages Log
for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/
instance=$QM" | cut -d ' ' -f 1); do oc cp -n $QM_NAMESPACE --retries=10 "$p:var/mqm/web/
installations/Installation1/servers/mqweb/logs/messages.log" "web-$p-messages.log"; done

# MQ Queue Manager: routes defined by operator
oc get routes -n $QM_NAMESPACE -o yaml --selector "app.kubernetes.io/instance=$QM" > "qm-
routes-$QM.yaml"

# MQ Queue Manager: routes to QM
oc get routes -n $QM_NAMESPACE -o yaml --field-selector "spec.to.name=$QM-ibm-mq" > "qm-
routes2-$QM.yaml"

# MQ Queue Manager: stateful set
oc get statefulset -n $QM_NAMESPACE -o yaml ${QM}-ibm-mq > "qm-statefulset-$QM.yaml"

# MQ Queue Manager: revisions of the stateful set
oc get controllerrevisions.apps -o yaml -n $QM_NAMESPACE --selector "app.kubernetes.io/
instance=$QM" > "qm-statefulset-revisions-$QM.yaml"

# MQ Queue Manager: Pod events
for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/
instance=$QM" | cut -d ' ' -f 1); do oc get -o custom-columns="LAST
SEEN:.lastTimestamp,TYPE:.type,REASON:.reason,KIND:.involvedObject.kind,NAME:.involvedObje
ct.name,MESSAGE:.message" event -n $QM_NAMESPACE --field-selector
involvedObject.name="$p" > "qm-pod-events-$p.txt"; done

# MQ Queue Manager: StatefulSet events
oc get events -n $QM_NAMESPACE -o custom-columns="LAST
SEEN:.lastTimestamp,TYPE:.type,REASON:.reason,KIND:.involvedObject.kind,NAME:.involvedObje
ct.name,MESSAGE:.message" --field-selector involvedObject.name="${QM}-ibm-mq" > "qm-
statefulset-events-$QM.txt"

# MQ Queue Manager: services
oc get services -n $QM_NAMESPACE -o yaml --selector "app.kubernetes.io/instance=$QM" >
"qm-services-$QM.yaml"

# MQ Queue Manager: PVCs
oc get pvc -n $QM_NAMESPACE -o yaml --selector "app.kubernetes.io/instance=$QM" > "qm-
pvcs-$QM.yaml"

# MQ Operator: Version
oc get csv -n $QM_NAMESPACE | grep "^ibm-mq\|NAME" > mq-operator-csv.txt

# Cloud Pak Foundational Services: Version
oc get csv -n $QM_NAMESPACE | grep "^ibm-common-service-operator\|NAME" > common-services-
csv.txt

# Cloud Pak for Integration: Version (if applicable)
oc get csv -n $QM_NAMESPACE | grep "^ibm-integration-platform-navigator\|NAME" > cp4i-
csv.txt

# Output from runmqras (this may take a while to execute)
for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/
instance=$QM" | cut -d ' ' -f 1); do timestamp=$(TZ=UTC date +"%Y%m%d_%H%M%S"); oc exec
-n $QM_NAMESPACE $p -- runmqras -workdirectory "/tmp/runmqras_$timestamp" -section
logger,mqweb,nativeha,trace; oc cp -n $QM_NAMESPACE --retries=10 "$p:tmp/
runmqras_$timestamp/" .; done

# MQ Operator: Pod Log
oc logs -n $MQ_OPERATOR_NAMESPACE $(oc get pods -n $MQ_OPERATOR_NAMESPACE --no-headers --
selector app.kubernetes.io/name=ibm-mq,app.kubernetes.io/managed-by=olm | cut -d ' ' -f
1) > mq-operator-log.txt
```

참고:

이러한 명령의 대부분은 큐 관리자가 배치된 네임스페이스에 대한 액세스가 필요합니다. 그러나 IBM MQ Operator 가 설치된 **클러스터 범위**인 경우 IBM MQ Operator 로그를 수집하려면 **클러스터 관리자** 액세스 권한이 추가로 필요할 수 있습니다.

관련 태스크

IBM 지원 센터에 대한 문제점 해결 정보 수집

OpenShift CP4I 문제점 해결: 큐 관리자 데이터에 대한 액세스 수집

PVC 검사기 도구를 사용하여 큐 관리자 팟 (Pod) 에 원격 쉘을 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 쉘을 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-tool에서 사용할 수 있습니다.
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 rsh pvc-inspector-pod-name
   ```

   마운트된 PVC 디렉토리를 포함하는 디렉토리에 배치됩니다.
   b) 다음 명령을 실행하여 PVC 디렉토리를 나열하십시오.

   ```
   ls
   ```

   c) 원격 쉘 세션 외부에서 다음 명령을 실행하여 PVC의 목록을 참조하십시오.

   ```
   oc get pvc
   ```

   d) 다음 명령을 실행하여 도구에서 작성된 팟 (Pod) 을 정리하십시오.

   ```
   oc delete pods -l tool=mq-pvc-inspector
   ```

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. IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산권을 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 라이센스까지 부여하는 것은 아닙니다. 라이센스에 대한 의문사항은 다음으로 문의하십시오.

07326
서울특별시 영등포구
국제금융로 10, 3IFC
한국 아이.비.엠 주식회사
U.S.A.

2바이트(DBCS) 정보에 관한 라이센스 문의는 한국 IBM에 문의하거나 다음 주소로 서면 문의하시기 바랍니다.

Intellectual Property Licensing
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 라이센스 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 라이센스가 부여된 프로그램 및 프로그램에 대해 사용 가능한 모든 라이센스가 부여된 자료는 IBM이 IBM 기본 계약, IBM 프로그램 라이센스 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 추정을

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 라이센스:

이 정보에는 여러 운영 플랫폼에서의 프로그래밍 기법을 보여주는 원어로 된 샘플 응용프로그램이 들어 있습니다. 귀하는 이러한 샘플 프로그램의 작성 기준이 된 운영 플랫폼의 애플리케이션 프로그래밍 인터페이스(API)에 부합하는 애플리케이션을 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 소프트카피로 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 IBM MQ의 서비스를 얻기 위해 프로그램을 작성할 수 있도록 하는 의도된 프로그래밍 인터페이스에 대한 정보가 들어 있습니다.

그러나 본 정보에는 진단, 수정 및 성능 조정 정보도 포함되어 있습니다. 진단, 수정 및 성능 조정 정보는 응용프로그램 소프트웨어의 디버그를 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: