

9.1

컨테이너의 *IBM MQ*

**IBM**

#### 참고

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, [49 페이지의 『주의사항』](#)에 있는 정보를 확인하십시오.

이 개정판은 새 개정판에 별도로 명시하지 않는 한, IBM® MQ의 버전 9릴리스 1 및 모든 후속 릴리스와 수정에 적용됩니다.

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

© Copyright International Business Machines Corporation 2007년, 2024.

# 목차

<b>컨테이너의 IBM MQ</b> .....	<b>5</b>
컨테이너에 IBM MQ 계획.....	5
컨테이너에서 IBM MQ를 사용하는 방법 선택.....	5
IBM MQ 인증 컨테이너 지원.....	6
IBM MQ 컨테이너 이미지 및 차트 빌드 지원.....	8
IBM MQ Advanced certified container의 스토리지 고려사항.....	8
IBM MQ Advanced certified container에 대한 고가용성.....	10
IBM MQ Advanced certified container에 대한 사용자 인증 및 권한 부여.....	11
OpenShift에 IBM MQ Operator 설치 및 설치 제거.....	11
OpenShift 웹 콘솔을 사용하여 IBM MQ Operator 설치.....	11
OpenShift CLI를 사용하여 IBM MQ Operator 설치.....	13
IBM MQ 인증 컨테이너 배치.....	15
OpenShift CLI를 사용하여 IBM MQ용 OpenShift 프로젝트 준비.....	15
IBM Cloud Pak for Integration Platform Navigator를 사용하여 큐 관리자 배치.....	16
OpenShift 웹 콘솔을 사용하여 큐 관리자 배치.....	16
OpenShift CLI를 사용하여 큐 관리자 배치.....	17
IBM Cloud Pak for Integration 조작 대시보드와 통합.....	19
OpenShift CLI를 사용하여 사용자 정의 MQSC와 INI 파일을 사용하여 이미지 빌드.....	20
Helm 을 사용하여 IBM MQ 인증된 컨테이너 배치.....	21
Deploying previous CD releases of IBM MQ into an IBM Cloud Private cluster.....	25
IBM MQ 이미지의 이전 CD 릴리스를 IBM Cloud Private 클러스터에 추가.....	26
IBM MQ 이미지의 이전 CD 릴리스를 IBM Cloud Kubernetes Service 클러스터에 추가.....	27
OpenShift 클러스터에 배치된 큐 관리자에 연결.....	28
OpenShift 클러스터에 배치된 IBM MQ Console에 연결.....	29
OpenShift CLI를 사용하여 큐 관리자 구성 백업 및 복원.....	30
IBM MQ 컨테이너 빌드.....	31
컨테이너를 사용하여 IBM MQ 큐 관리자 이미지 계획.....	32
Docker 를 사용하여 샘플 IBM MQ 큐 관리자 이미지 빌드.....	32
별도의 컨테이너에서 로컬 바인딩 애플리케이션 실행.....	34
IBM MQ Operator용 API 참조.....	37
mq.ibm.com/v1beta1용 API 참조.....	37
<b>주의사항</b> .....	<b>49</b>
프로그래밍 인터페이스 정보.....	50
상표.....	50



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

IBM MQ Advanced 및 IBM MQ Advanced for Developers에 제공된 사전 패키징된 컨테이너에서 IBM MQ를 실행할 수 있습니다. 이 IBM MQ Advanced certified container에서는 지원되는 이미지 및 Helm 차트를 제공하며 제품 준비 IBM MQ 이미지를 Red Hat® OpenShift®, IBM Cloud® Private 또는 IBM Cloud Kubernetes Service에 배치하기 위해 사용될 수 있습니다.

IBM MQ를 IBM Cloud Pak® for Integration 컨테이너, 또는 직접 빌드한 컨테이너에서 실행할 수도 있습니다.

**MQ Adv.** **CD** IBM MQ Advanced certified container에 대한 자세한 정보는 다음 링크를 참조하십시오.

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

## 이 태스크 정보

Before you plan your IBM MQ in containers architecture, you should familiarize yourself with both the basic IBM MQ concepts (see [IBM MQ 기술 개요](#)) as well as basic Kubernetes/OpenShift concepts (see [OpenShift Container Platform 아키텍처](#)).

## 프로시저

- 5 페이지의 『컨테이너에서 IBM MQ를 사용하는 방법 선택』.
- 10 페이지의 『IBM MQ Advanced certified container에 대한 고가용성』.
- 11 페이지의 『IBM MQ Advanced certified container에 대한 사용자 인증 및 권한 부여』.

컨테이너에는 IBM MQ를 사용하기 위한 여러 옵션이 있습니다. 패키징되기 이전의 인증된 컨테이너 이미지를 사용하도록 선택하거나 자체 이미지와 개발 코드를 빌드할 수 있습니다.

## IBM MQ Advanced 인증 컨테이너 사용

Red Hat OpenShift Container Platform에 배치하려는 경우, 인증된 컨테이너를 사용하려고 할 수도 있습니다. 다음과 같은 세 가지 유형의 인증된 컨테이너가 있습니다.

- IBM MQ Advanced certified container for IBM Cloud Pak for Integration. 이는 인증된 컨테이너 버전을 포함하는 별도의 IBM 제품입니다.
- IBM MQ Advanced certified container
- IBM MQ Advanced for Developers 인증된 컨테이너(보증되지 않음)

IBM MQ 9.1.4 및 이전 CD 릴리스는 IBM Cloud Private 및 IBM Cloud Kubernetes Service에서도 지원됩니다.

인증된 컨테이너가 빠르게 발전하는 중이고 [Continuous Delivery](#) 릴리스만 지원된다는 것에 유의하십시오.

인증된 컨테이너에는 사전 빌드된 컨테이너 이미지와 Red Hat OpenShift Container Platform에서 실행하기 위한 배치 코드가 포함됩니다. IBM MQ 9.1.5 이상에서 큐 관리자는 IBM MQ 연산자를 사용하여 관리됩니다. IBM MQ의 이전 버전(버전 9.1.5포함)은 Helm 차트를 사용하여 관리됩니다.

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

- 관리 또는 메시징에 REST API 사용
- 다음 MQ 컴포넌트를 사용하십시오.
  - Managed File Transfer 에이전트 및 해당 자원. 그러나 인증된 컨테이너를 사용하여 하나 이상의 조정, 명령 또는 에이전트 큐 관리자를 제공할 수 있습니다.
  - AMQP
  - IBM MQ Bridge to Salesforce
  - IBM MQ Bridge to blockchain(컨테이너에서 지원되지 않음)
- Helm 도표를 사용하여 배치할 때 웹 서버 사용 (IBM Cloud Pak for Integration 제외)
- 복구 로그 구성과 같이 **crtmqm**, **strmqm** 및 **endmqm**과 함께 사용되는 옵션을 사용자 정의하십시오.

## 자체 이미지 및 차트 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. 31 페이지의 『IBM MQ 컨테이너 빌드』의 내용을 참조하십시오. 인증된 컨테이너의 일부로 제공된 Helm 차트가 GitHub에 발행되고 자체 이미지를 빌드할 때 샘플로 사용될 수 있습니다.

- [IBM MQ Advanced certified container용 Helm 차트](#)
- [IBM MQ Advanced for Developers 인증된 컨테이너용 Helm 차트](#)

### 관련 개념

6 페이지의 『IBM MQ 인증 컨테이너 지원』

IBM MQ 인증된 컨테이너는 특정 Kubernetes 환경에서만 지원됩니다.

8 페이지의 『IBM MQ 컨테이너 이미지 및 차트 빌드 지원』

Linux 시스템에서 컨테이너를 사용 중인 경우 고려해야 할 정보.

## Linux IBM MQ 인증 컨테이너 지원

IBM MQ 인증된 컨테이너는 특정 Kubernetes 환경에서만 지원됩니다.

**V 9.1.4** **MQ Adv.** **CD** CD 릴리스 V9.1.4 이상의 경우, IBM MQ Advanced certified container은 Red Hat OpenShift의 사용을 위해 지원됩니다. 22 페이지의 『Helm CLI를 사용하여 큐 관리자 배치』의 내용을 참조하십시오.

V9.1.4 이전의 CD 릴리스는 다음과 같은 Kubernetes 환경에서 지원되었습니다.

- IBM Cloud Kubernetes Service
- IBM Cloud Private
- Red Hat OpenShift(가) 있는 IBM Cloud Private

지원되는 특정 Kubernetes 버전의 경우 다운로드한 IBM MQ Advanced Helm 차트 내의 `qualification.yaml` 및 `Chart.yaml` 파일을 참조하십시오. 이 버전은 릴리스마다 다를 수 있습니다.

IBM MQ Advanced certified container 는 IBM MQ 연산자를 사용하거나 다음 Helm 도표 중 하나를 사용하는 경우에만 지원됩니다.

- `ibm-mqadvanced-server-prod`
- `ibm-mqadvanced-server-integration-prod`(IBM Cloud Pak for Integration에 있음)

**참고:** Helm 차트의 사용은 IBM MQ 연산자의 릴리스에 따라 더 이상 사용되지 않습니다.

컨테이너 기술은 신속하게 진화하기 때문에 IBM MQ Advanced certified container은 릴리스 시점에 이 차트에서 지원하는 최신 버전의 플랫폼에서만 지원됩니다. 이전 플랫폼 버전을 사용하려는 경우 이전 버전의 IBM MQ Advanced certified container을 사용해야 할 수도 있습니다.

IBM MQ Advanced certified container 이미지는 IBM MQ Continuous Delivery (CD) 릴리스를 기반으로 합니다. 최대 1년 또는 두 개의 CD 릴리스 동안 지원됩니다(둘 중 더 긴 기간). IBM MQ의 Long Term Support 릴리스는 인증된 컨테이너로 사용할 수 없습니다.

IBM MQ Advanced certified container V4.0 부터 이미지는 Red Hat Universal Base Image (UBI) 에 IBM MQ의 설치를 제공합니다. 여기에는 IBM MQ가 사용하는 키 Linux 라이브러리 및 유틸리티가 포함되어 있습니다. UBI는 Red Hat Enterprise Linux 호스트에서 실행될 때 Red Hat에서 지원됩니다. IBM MQ Advanced certified container의 이전 버전은 지원되지 않는 Ubuntu 기본 이미지를 사용합니다.

**관련 개념**

8 페이지의 『IBM MQ 컨테이너 이미지 및 차트 빌드 지원』  
Linux 시스템에서 컨테이너를 사용 중인 경우 고려해야 할 정보.

Linux MQ Adv. CD **IBM MQ Advanced certified container에 대한 버전 지원**

IBM MQ Advanced certified container, IBM MQ, IBM Cloud Kubernetes Service, IBM Cloud Pak for Integration, IBM Cloud Private의 지원 버전 간 매핑을 표시하는 표 세트를 제공합니다.

**IBM MQ 운영자**

V 9.1.5

IBM MQ 연산자는 IBM Cloud Pak for Integration 버전 2020.2의 일부로 또는 독립적으로 IBM MQ 버전 9.1.5 이상을 사용하도록 지원됩니다.

IBM MQ 연산자는 Red Hat OpenShift Container Platform 버전 4.4 이상에서 지원됩니다.

**IBM MQ Advanced certified container V 9.1.5 (Helm 도표) - 더 이상 사용되지 않음**

Helm 차트 `ibm-mqadvanced-server-prod`를 포함합니다.

V 9.1.5 IBM MQ Advanced certified container V5.0.x 이상에서 Helm 차트, 이미지 및 수정사항이 IBM 제목 카테고리 및 레지스트리를 통해 제공됩니다. 초기 버전이 Passport Advantage®를 통해 제공되고 IBM Fix Central에서 수정사항 릴리스를 사용할 수 있습니다.

표 1. IBM MQ Advanced certified container에 대한 지원			
버전	IBM MQ 버전	지원 종료	지원 플랫폼
6.0.x	9.1.5 Continuous Delivery 릴리스	2021년 3월	<a href="#">세부 시스템 요구사항</a>
5.0.x	9.1.4 Continuous Delivery 릴리스	2020년 12월	<a href="#">세부 시스템 요구사항</a>
4.1.x	9.1.3 Continuous Delivery 릴리스	2020년 7월	<a href="#">세부 시스템 요구사항</a>

**IBM Cloud Pak for Integration 소프트웨어 IBM MQ Advanced certified container**

V 9.1.5 (Helm 차트) - 더 이상 사용되지 않음

Helm 차트 `ibm-mqadvanced-server-integration-prod`를 포함합니다.

표 2. IBM Cloud Pak for Integration 용 IBM MQ Advanced certified container 소프트웨어에 대한 버전 지원		
버전	IBM MQ 버전	IBM Cloud Pak for Integration 버전
6.0.x	9.1.4 Continuous Delivery 릴리스	2020.1.1 (시스템 요구사항)
5.0.x	9.1.3 Continuous Delivery 릴리스	2019.4.1 (시스템 요구사항)
4.1.x	9.1.3 Continuous Delivery 릴리스	2019.3.2.2 (시스템 요구사항)
4.0.x	9.1.3 Continuous Delivery 릴리스	2019.3.2 (시스템 요구사항)
3.0.x	9.1.3 Continuous Delivery 릴리스	2019.3.1 (시스템 요구사항)

지원되는 버전 정보는 [IBM Cloud Pak for Integration 릴리스 노트](#)를 참조하십시오.

## Linux IBM MQ 컨테이너 이미지 및 차트 빌드 지원

Linux 시스템에서 컨테이너를 사용 중인 경우 고려해야 할 정보.

- 컨테이너 이미지에서 사용하는 기본 이미지는 지원되는 Linux 운영 체제를 사용해야 합니다.
- IBM MQ 설치 프로그램을 사용하여 컨테이너 이미지 내에 제품을 설치해야 합니다.
- 지원되는 패키지 목록은 [Linux 시스템에 대한 IBM MQ rpm 컴포넌트](#)의 내용을 참조하십시오.
- **V 9.1.0** 다음 패키지는 지원되지 않습니다.
  - MQSeriesBCBridge
  - MQSeriesRDQM
- 큐 관리자 데이터 디렉토리(기본적으로 /var/mqm)는 지속 상태를 유지하는 컨테이너 볼륨에 저장되어야 합니다.

**중요사항:** 유니온 파일 시스템을 사용할 수 없습니다.

데이터 볼륨으로 호스트 디렉토리를 마운트하거나 데이터 볼륨 컨테이너를 사용해야 합니다. 자세한 정보는 [컨테이너에서 데이터 관리](#)를 참조하십시오.

- 컨테이너 내에서 IBM MQ 제어 명령을 실행할 수 있어야 합니다(예: **endmqm**).
- 진단 목적으로 컨테이너 내에 있는 파일과 디렉토리를 가져올 수 있어야 합니다.
- **V 9.1.0** 로컬에서 애플리케이션을 별도의 컨테이너에서 실행 중인 큐 관리자에 바인딩할 수 있도록 네임 스페이싱을 사용하여 큐 관리자에 대한 컨테이너의 네임스페이스를 다른 컨테이너와 공유할 수 있습니다. 자세한 정보는 [34 페이지의 『별도의 컨테이너에서 로컬 바인딩 애플리케이션 실행』](#)의 내용을 참조하십시오.

### 관련 개념

6 페이지의 [『IBM MQ 인증 컨테이너 지원』](#)

IBM MQ 인증된 컨테이너는 특정 Kubernetes 환경에서만 지원됩니다.

## Linux MQ Adv. CD V 9.1.5 IBM MQ Advanced certified container의

### 스토리지 고려사항

IBM MQ Advanced certified container는 두 스토리지 모드로 실행됩니다.

- **임시 스토리지**는 모든 컨테이너 상태가 컨테이너 재시작 시 삭제될 수 있습니다. 이는 일반적으로 환경이 데모 용으로 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 연산자는 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 경우 모든 큐 관리자 상태가 재시작 시 손실됩니다. 여기에는 다음이 포함됩니다.

- 모든 메시지
- 모든 큐 관리자 대 큐 관리자 통신 상태(채널 메시지 순서 번호)
- 큐 매니저의 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 상태를 처리하는 것뿐만 아니라 큐 관리자 구성도 제거됩니다. 완전히 임시 컨테이너를 사용으로 설정하려면, IBM MQ 구성이 컨테이너 이미지 자체(자세한 정보는 20 페이지의 『OpenShift CLI를 사용하여 사용자 정의 MQSC와 INI 파일을 사용하여 이미지 빌드』의 내용 참조)에 추가되어야 합니다. 이를 완료하지 않으면 컨테이너를 다시 시작할 때마다 IBM MQ가 구성되어야 합니다.

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

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

## 지속 스토리지

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

- `spec.queueManager.availability`는 가용성 모드를 제어합니다. `ReadWriteOnce` 스토리지만 필요한 `SingleInstance`를 사용 중인 경우, `multiInstance`는 오라클 파일 잠금 특성을 가진 `ReadWriteMany`를 지원하는 스토리지 클래스를 필요로 합니다. IBM MQ는 지원 명령문 및 명령문 테스트를 제공합니다. 또한 가용성 모드는 지속 볼륨 레이아웃에 영향을 줍니다. 추가 정보는 10 페이지의 『IBM MQ Advanced certified container에 대한 고가용성』의 내용을 참조하십시오.
- `spec.queueManager.storage`는 개별 스토리지 설정을 제어합니다. 큐 관리자는 1~3개 사이의 지속 볼륨을 사용하도록 구성될 수 있습니다.

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

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

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

```
spec:
  queueManager:
    availability:
      type: MultiInstance
    storage:
      queueManager:
        enabled: true
        class: ibmc-file-gold-gid
        persistedData:
          enabled: true
          class: ibmc-file-gold-gid
        recoveryLogs:
```

```
enabled: true
class: ibmc-file-gold-gid
securityContext:
supplementalGroups: [99]
```

Linux

MQ Adv.

CD

## IBM MQ Advanced certified container에 대한 고가

### 용성

IBM MQ Advanced certified container에서 고가용성에 대해 두 가지 주요 선택사항이 제공됩니다. **멀티 인스턴스 큐 관리자**(네트워크로 연결된 공유 파일 시스템을 사용하는 활성 대기 쌍) 및 **단일 복원력 큐 관리자**(네트워크로 연결된 스토리지를 사용하는 HA를 위한 간단한 접근 방식 제공)가 이와 같습니다.

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

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

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

### 다중 인스턴스 큐 관리자

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

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

멀티 인스턴스 큐 관리자의 복구 시간은 다음 요소로 제어됩니다.

1. 실패 후 공유된 파일 시스템이 활성 인스턴스에서 처음에 설정한 잠금을 릴리스하는 데 걸리는 시간.
2. 대기 인스턴스가 잠금을 확보하고 시작하는 데 걸리는 시간.
3. Kubernetes 팟(Pod) 프로브가 컨테이너가 준비되었는지 감지하는 데 걸리는 시간. 이는 Helm 차트에서 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 ext4와 같습니다.

단일 복원력 큐 관리자의 복구 시간은 다음 요소로 제어됩니다.

1. 라이브 프로브를 실행하는 데 걸리는 시간과 허용하는 장애 수. 이는 Helm 차트에서 구성 가능합니다.
2. Kubernetes 스케줄러가 새 노드로 장애가 발생한 팟(Pod)을 다시 스케줄하는 데 걸리는 시간.

3. 컨테이너 이미지를 새 노드로 다운로드하는 데 걸리는 시간. IfNotPresent의 **imagePullPolicy** 값을 사용하는 경우 이미지는 해당 노드에서 이미 사용 가능할 수 있습니다.
4. 새 큐 관리자 인스턴스를 시작하는 데 걸리는 시간.
5. Kubernetes 팟(Pod) 프로브가 컨테이너가 준비되었는지 감지하는 데 걸리는 시간. 이는 Helm 차트에서 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 마스터 노드에서 작업자 노드와의 연결이 끊어지면 노드에서 장애가 발생했는지 또는 단순히 네트워크 연결이 끊어진 것인지를 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.

#### 관련 개념

[고가용성 구성](#)

## Linux > MQ Adv. > CD IBM MQ Advanced certified container에 대한 사용자 인증 및 권한 부여

권한 부여를 위해 LDAP 사용자 및 그룹을 사용하도록 IBM MQ를 구성할 수 있습니다. IBM MQ Advanced certified container의 경우 이 접근법을 권장합니다.

Red Hat OpenShift Container Platform와 같은 다중 테넌트 컨테이너화된 환경에서 보안 컨테이너는 잠재적인 보안 문제가 발생하지 않도록 보안상의 제한조건이 적용됩니다. 예를 들어, Red Hat OpenShift Container Platform에서 기본 SecurityContextConstraints(restricted라는)는 임의의 사용자 ID를 사용하여 컨테이너 자체에 로컬인 사용자는 권장하지 않습니다. IBM MQ는 일반적으로 권한 상승을 사용하여 사용자의 비밀번호를 확인하며, 또한 다중 테넌트 컨테이너 환경에서 권장되지 않습니다. 이러한 이유로 인해, 실행 중인 컨테이너 내의 운영 체제 라이브러리에 정의된 사용자들의 사용이 IBM MQ 인증된 컨테이너에서 지원되지 않습니다.

사용자 인증 및 권한 부여를 위해 LDAP를 사용하도록 큐 관리자를 구성해야 합니다. 이 작업을 수행하도록 IBM MQ를 구성하는 방법에 대한 정보는 [연결 인증: 사용자 저장소 및 LDAP 권한 부여](#)를 참조하십시오.

## Linux > MQ Adv. > CD > V 9.1.5 OpenShift에 IBM MQ Operator 설치 및 설치 제거

IBM MQ 연산자는 연산자 허브를 사용하여 OpenShift에 설치될 수 있습니다.

### 시작하기 전에

#### 프로시저

- 13 페이지의 『OpenShift CLI를 사용하여 IBM MQ Operator 설치』.
- 11 페이지의 『OpenShift 웹 콘솔을 사용하여 IBM MQ Operator 설치』.

## Linux > MQ Adv. > CD > V 9.1.5 OpenShift 웹 콘솔을 사용하여 IBM MQ Operator 설치

IBM MQ 연산자는 연산자 허브를 사용하여 OpenShift에 설치될 수 있습니다.

## 시작하기 전에

OpenShift 클러스터의 웹 콘솔에 로그인합니다.

## 프로시저

1. 설치 가능한 연산자 목록에 IBM Common Services 연산자 추가
  - a) 더하기 아이콘을 클릭하십시오. **YAML 가져오기** 대화 상자를 볼 수 있습니다.
  - b) 대화 상자에서 다음 자원 정의를 붙여넣으십시오.

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

- c) **작성**을 클릭하십시오.
2. IBM 연산자를 설치 가능한 연산자 목록에 추가
    - a) 더하기 아이콘을 클릭하십시오. **YAML 가져오기** 대화 상자를 볼 수 있습니다.
    - b) 대화 상자에서 다음 자원 정의를 붙여넣으십시오.

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

- c) **작성**을 클릭하십시오.
3. IBM MQ Operator에 사용할 네임스페이스 작성

IBM MQ 연산자는 단일 네임스페이스나 모든 네임스페이스로 범위를 지정하여 설치될 수 있습니다. 이 단계는 아직 존재하지 않는 특정 네임스페이스에 설치하려는 경우에만 필요합니다.

    - a) 탐색 패널에서 **홈 > 프로젝트**를 클릭하십시오.  
프로젝트 페이지가 표시됩니다.
    - b) **프로젝트 작성**을 클릭하십시오. 프로젝트 작성 범위가 표시됩니다.
    - c) 작성 중인 네임스페이스의 세부사항을 입력하십시오. 예를 들어, "ibm-mq"를 이름으로 지정할 수 있습니다.
    - d) **작성**을 클릭하십시오. IBM MQ Operator의 네임스페이스가 작성됩니다.
  4. IBM MQ 운영자를 설치하십시오.
    - a) 탐색 패널에서 **Operators > OperatorHub**를 클릭하십시오.  
OperatorHub 페이지가 표시됩니다.
    - b) **모든 항목** 필드에 "IBM MQ"을 입력하십시오.  
IBM MQ 카탈로그 항목이 표시됩니다.
    - c) **IBM MQ**를 선택하십시오.  
IBM MQ 창이 표시됩니다.
    - d) **설치**를 클릭하십시오.

Operator 구독 작성 페이지를 볼 수 있습니다.

e) 작성한 특정 네임스페이스 또는 클러스터 전체 범위로 설치 모드 설정

f) 구독을 클릭하십시오.

설치된 연산자 페이지에서 IBM MQ를 볼 수 있습니다.

g) 설치된 연산자 페이지에서 연산자의 상태를 확인하십시오. 설치가 완료되면 상태가 성공으로 변경됩니다.

## 다음에 수행할 작업

15 페이지의 『IBM MQ 인증 컨테이너 배치』

Linux

MQ Adv.

CD

V 9.1.5

OpenShift CLI를 사용하여 IBM MQ

## Operator 설치

IBM MQ 연산자는 연산자 허브를 사용하여 OpenShift에 설치될 수 있습니다.

### 시작하기 전에

**oc login**를 사용하여 OpenShift 명령행 인터페이스 (CLI) 에 로그인하십시오. 이 단계에서는 클러스터 관리자여야 합니다.

### 프로시저

#### 1. IBM Common Services 연산자용 OperatorSource 작성

##### a) OperatorSource 자원을 정의하는 YAML 파일 작성

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

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

##### b) 서버에 OperatorSource을(를) 적용하십시오.

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

#### 2. IBM 연산자용 OperatorSource 작성

##### a) OperatorSource 자원을 정의하는 YAML 파일 작성

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

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

##### b) 서버에 OperatorSource을(를) 적용하십시오.

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

### 3. IBM MQ Operator에 사용할 네임스페이스 작성

IBM MQ Operator는 단일 네임스페이스 또는 모든 네임스페이스 범위로 설치될 수 있습니다. 이 단계는 아직 존재하지 않는 특정 네임스페이스에 설치하려는 경우에만 필요합니다.

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

### 4. OperatorHub에서 클러스터에 사용 가능한 Operator의 목록 보기

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

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

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

### 6. OperatorGroup 오브젝트 YAML 파일 작성

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

Operator를 구독하는 네임스페이스에는 Operator의 InstallMode(AllNamespaces 또는 SingleNamespace 모드)와 일치하는 OperatorGroup이 있어야 합니다. 설치하려는 연산자가 AllNamespaces를 사용하는 경우, openshift-operators 네임스페이스에는 이미 적절한 OperatorGroup가 있습니다.

그러나 연산자가 SingleNamespace 모드를 사용하고 적절한 OperatorGroup이 없는 경우 이를 작성해야 합니다.

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

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

b) OperatorGroup 오브젝트를 작성하십시오.

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

### 7. 구독 오브젝트 YAML 파일을 작성하여 네임스페이스를 MQ Operator에 등록

a) 다음 콘텐츠를 사용하여 "mq-sub.yaml"이라는 파일 작성:

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

AllNamespaces **InstallMode** 사용의 경우, openshift-operators 네임스페이스를 지정하십시오. 그렇지 않으면, SingleNamespace **InstallMode** 사용에 대한 관련 단일 네임스페이스를 지정하십시오.

b) Subscription 오브젝트를 작성하십시오.

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

### 8. 운영자의 상태 확인

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

## 다음에 수행할 작업

15 페이지의 『IBM MQ 인증 컨테이너 배치』

Linux

MQ Adv.

CD

## IBM MQ 인증 컨테이너 배치

IBM MQ 버전 9.1.5 이상은 IBM MQ 연산자를 사용하여 Red Hat OpenShift 에 배치할 수 있습니다. IBM MQ 버전 9.1.5 및 9.1.4 는 Helm을 사용하여 Red Hat OpenShift 에 배치할 수 있습니다. 이전 CD 버전은 Helm을 사용하여 IBM Cloud Private 클러스터 또는 IBM Cloud Kubernetes Service 클러스터에 배치할 수 있습니다.

## 이 태스크 정보

### 프로시저

- [22 페이지의 『Helm CLI를 사용하여 큐 관리자 배치』](#).
- [25 페이지의 『Deploying previous CD releases of IBM MQ into an IBM Cloud Private cluster』](#).
- [26 페이지의 『IBM MQ 이미지의 이전 CD 릴리스를 IBM Cloud Private 클러스터에 추가』](#).
- [27 페이지의 『IBM MQ 이미지의 이전 CD 릴리스를 IBM Cloud Kubernetes Service 클러스터에 추가』](#).

Linux

MQ Adv.

CD

## OpenShift CLI를 사용하여 IBM MQ용 OpenShift 프로젝트 준비

### 로젝트 준비

Red Hat OpenShift 컨테이너 플랫폼 클러스터를 준비하십시오. 그러면 IBM MQ 연산자를 사용하여 큐 관리자를 배치할 준비가 됩니다. 이 태스크는 프로젝트 관리자에서 완료해야 합니다.

### 시작하기 전에

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

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

## 이 태스크 정보

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

### 프로시저

1. 사용자 ID에 지정된 인타이틀먼트 키를 가져오십시오.
  - a) 자격이 있는 소프트웨어와 연관된 IBM ID 및 비밀번호를 사용하여 [MyIBM](#) 에 로그인하십시오.
  - b) **인타이틀먼트 키** 섹션에서 **키 복사**를 선택하여 인타이틀먼트 키를 클립보드에 복사하십시오.
2. 큐 관리자를 배치할 프로젝트에서 인타이틀먼트 키를 포함하는 본인확인정보를 작성하십시오.

다음 명령을 실행하십시오. 여기서 **<entitlement-key>**는 1단계에서 검색된 키이고 **<user-email>**은 권한 부여된 소프트웨어와 연관된 IBM ID입니다.

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

## 다음에 수행할 작업

17 페이지의 『OpenShift CLI를 사용하여 큐 관리자 배치』

Linux

MQ Adv.

CD

V 9.1.5

## IBM Cloud Pak for Integration Platform

### Navigator를 사용하여 큐 관리자 배치

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

#### 시작하기 전에

브라우저에서 IBM Cloud Pak for Integration Platform Navigator를 시작하십시오.

큐 관리자를 이 Red Hat OpenShift 프로젝트에 배치하는 것이 처음이면 [15 페이지의 『OpenShift CLI를 사용하여 IBM MQ용 OpenShift 프로젝트 준비』](#)의 단계를 따르십시오.

#### 프로시저

##### 1. 큐 관리자를 배치하십시오.

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

a) IBM Cloud Pak for Integration Platform Navigator에서 **런타임 및 인스턴스**를 클릭하십시오.

b) **인스턴스 작성**을 클릭하십시오.

c) 큐 관리자를 선택한 다음, **다음**을 클릭하십시오.

QueueManager의 인스턴스를 작성할 양식이 표시됩니다.

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

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

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

f) **큐 관리자 구성** 섹션에서 기본 큐 관리자의 **이름**을 확인하거나 업데이트하십시오.

기본적으로, IBM MQ 클라이언트 애플리케이션에서 사용된 큐 관리자의 이름은 QueueManager의 이름과 동일하지만 올바르지 않은 문자(하이픈과 같이)가 제거됩니다. 강제로 특정 이름을 사용하도록 하려면 여기서 이를 편집할 수 있습니다.

g) **작성**을 클릭하십시오.

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

##### 2. 큐 관리자가 실행 중인지 확인

QueueManager 상태가 Running인 경우 작성이 완료됩니다.

#### 관련 태스크

[28 페이지의 『OpenShift 클러스터에 배치된 큐 관리자에 연결』](#)

Red Hat OpenShift 클러스터에 배치된 큐 관리자에 연결하기 위한 구성 예제 세트입니다.

[29 페이지의 『OpenShift 클러스터에 배치된 IBM MQ Console에 연결』](#)

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

Linux

MQ Adv.

CD

V 9.1.5

## OpenShift 웹 콘솔을 사용하여 큐 관리자 배

### 치

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



## 시작하기 전에

OpenShift 클러스터의 웹 콘솔에 로그인합니다. 사용할 기존 프로젝트(네임스페이스)를 선택하여 새 프로젝트를 작성해야 합니다.

큐 관리자를 이 Red Hat OpenShift 프로젝트에 배치하는 것이 처음이면 [15 페이지의 『OpenShift CLI를 사용하여 IBM MQ용 OpenShift 프로젝트 준비』](#)의 단계를 따르십시오.

## 프로시저

### 1. 큐 관리자를 배치하십시오.

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

a) OpenShift 웹 콘솔의 탐색 분할창에서 **연산자 > 설치된 연산자**를 클릭하십시오.

b) **IBM MQ**를 클릭하십시오.

c) 큐 관리자 탭을 클릭하십시오.

d) 큐 관리자 작성 단추를 클릭하십시오.

QueueManager 자원의 예 YAML을 포함한 YAML 편집기가 표시됩니다.

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

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

IBM MQ는 여러 다른 라이선스에서 사용 가능합니다. 유효한 라이선스에 대한 자세한 정보는 [37 페이지의 『mq.ibm.com/v1beta1에 대한 라이선스 부여 참조』](#)의 내용을 참조하십시오. 큐 관리자를 배치하려면 라이선스를 승인해야 합니다.

f) **작성**을 클릭하십시오.

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

### 2. 큐 관리자가 실행 중인지 확인

QueueManager 상태가 Running인 경우 작성이 완료됩니다.

## 관련 태스크

[28 페이지의 『OpenShift 클러스터에 배치된 큐 관리자에 연결』](#)

Red Hat OpenShift 클러스터에 배치된 큐 관리자에 연결하기 위한 구성 예제 세트입니다.

[29 페이지의 『OpenShift 클러스터에 배치된 IBM MQ Console에 연결』](#)

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

## Linux > MQ Adv. > CD > V 9.1.5 > OpenShift CLI를 사용하여 큐 관리자 배치

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

## 시작하기 전에

Red Hat OpenShift Container Platform 명령행 인터페이스를 설치해야 합니다.

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

큐 관리자를 이 Red Hat OpenShift 프로젝트에 배치하는 것이 처음이면 [15 페이지의 『OpenShift CLI를 사용하여 IBM MQ용 OpenShift 프로젝트 준비』](#)의 단계를 따르십시오.

## 프로시저

### 1. 큐 관리자를 배치하십시오.

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

a) QueueManager YAML 파일을 작성하십시오.

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

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
spec:
  version: 9.1.5.0-r2
  license:
    accept: false
    license: L-RJON-BN7PN3
    use: NonProduction
  web:
    enabled: true
  queueManager:
    name: "QUICKSTART"
  storage:
    queueManager:
      type: ephemeral
  template:
    pod:
      containers:
        - name: qmgr
          env:
            - name: MQSNOAUT
              value: "yes"
```

**중요:** IBM Cloud Pak for Integration 라이선스 계약을 승인하면 `accept: false`를 `accept: true`로 변경하십시오. 라이선스에 대한 자세한 내용은 37 페이지의 『mq.ibm.com/v1beta1에 대한 라이선스 부여 참조』의 내용을 참조하십시오.

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

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

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart
spec:
  version: 9.1.5.0-r2
  license:
    accept: false
    license: L-APIG-BM7GDH
    use: Development
  web:
    enabled: true
  queueManager:
    name: "QUICKSTART"
  storage:
    queueManager:
      type: ephemeral
  template:
    pod:
      containers:
        - name: qmgr
          env:
            - name: MQSNOAUT
              value: "yes"
```

**중요:** MQ 라이선스 계약에 동의하는 경우 `accept: false`을(를) `accept: true`(으)로 변경하십시오. 라이선스에 대한 자세한 내용은 37 페이지의 『mq.ibm.com/v1beta1에 대한 라이선스 부여 참조』의 내용을 참조하십시오.

b) QueueManager 오브젝트를 작성하십시오.

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

2. 큐 관리자가 실행 중인지 확인  
실행 중인 배치의 유효성을 검증할 수 있습니다.

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

을 실행하여 배치의 유효성을 검증한 다음 상태를 확인합니다.  
예를 들어,

```
oc describe queuemanager quickstart
```

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

### 관련 태스크

28 페이지의 『[OpenShift 클러스터에 배치된 큐 관리자에 연결](#)』

Red Hat OpenShift 클러스터에 배치된 큐 관리자에 연결하기 위한 구성 예제 세트입니다.

29 페이지의 『[OpenShift 클러스터에 배치된 IBM MQ Console에 연결](#)』

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

## Linux > V 9.1.4 > MQ Adv. > CD IBM Cloud Pak for Integration 조작 대시보드와 통합

IBM Cloud Pak for Integration을 통해 트랜잭션을 추적하는 기능이 조작 대시보드에서 제공됩니다.

### 이 태스크 정보

조작 대시보드로 통합하면 큐 관리자에 MQ API 엑시트를 설치합니다. API 엑시트는 큐 관리자를 통해 수행 중인 메시지에 대한 조작 대시보드 데이터 저장소로 추적 데이터를 전송합니다.

MQ 클라이언트 바인딩을 사용하여 송신되는 메시지만 추적됩니다.

### 프로시저

1. 추적이 사용 가능한 큐 관리자 배치

기본적으로 추적 기능은 사용 불가능합니다.

IBM Cloud Pak for Integration Platform Navigator를 사용하여 배치하는 경우, 배치하는 동안 추적을 사용 가능하게 하고, **추적 사용**을 **켄**으로 설정하고 조작 대시보드가 설치된 네임스페이스로 **네임스페이스 추적**을 설정하는 동안 추적을 사용할 수 있습니다. 큐 관리자 배치에 대한 자세한 정보는 16 페이지의 『[IBM Cloud Pak for Integration Platform Navigator를 사용하여 큐 관리자 배치](#)』의 내용을 참조하십시오.

[OpenShift CLI](#) 또는 [OpenShift 웹 콘솔](#)을 사용하여 배치하면, 다음 YAML 스니펫으로 추적을 사용할 수 있습니다.

```
spec:
  tracing:
    enabled: true
    namespace: <Operations_Dashboard_Namespace>
```

Helm을 사용하여 배치하는 경우 `odTracingConfig.enabled=true` 및 `odTracingConfig.odTracingNamespace=<Operations_Dashboard_Namespace>`를 설정하여 추적을 사용할 수 있습니다. 기존 큐 관리자에서 조작 대시보드 통합을 사용 가능하게 하려면 Helm 릴리스를 업그레이드하는 동안 이 설정을 적용할 수 있습니다.

**중요:** MQ가 조작 대시보드로 조작되어야 큐 관리자가 시작됩니다(다음 단계 참조).

이 기능이 사용되면 큐 관리자 컨테이너 뿐만 아니라 두 개의 사이드카 컨테이너("Agent" 및 "Collector")가 실행됩니다. 이러한 사이드카 컨테이너에 대한 이미지를 기본 MQ 이미지와 동일한 레지스트리에서 사용 가

능하고 동일한 PULL 정책 및 PULL 시크릿을 사용합니다. CPU 및 메모리 제한사항을 구성할 수 있는 추가 설정이 있습니다.

- 이 네임스페이스에 오퍼레이션 대시보드 통합이 있는 큐 관리자가 처음으로 배치된 경우 조작 대시보드를 사용하여 등록을 수행해야 합니다.

등록하면 큐 관리자 Pod가 성공적으로 시작해야 하는 시크릿 오브젝트를 작성합니다.

## Linux MQ Adv. CD V 9.1.5 OpenShift CLI를 사용하여 사용자 정의 MQSC와 INI 파일을 사용하여 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 프로젝트에서 IBM Entitled Registry용 OpenShift Secret가 없는 경우, [15 페이지의 『OpenShift CLI를 사용하여 IBM MQ용 OpenShift 프로젝트 준비』](#) 단계를 따르십시오.

### 프로시저

#### 1. 작성ImageStream

이미지 스트림 및 연관된 태그는 Red Hat OpenShift Container Platform 내 컨테이너 이미지를 참조하기 위한 추상화를 제공합니다. 이미지 스트림 및 해당 태그를 사용하면 사용 가능한 이미지를 확인하고 저장소의 이미지가 변경되는 경우에도 필요한 특정 이미지를 사용하고 있는지 확인할 수 있습니다.

```
oc create imagestream mymq
```

#### 2. 새 이미지에 대한 BuildConfig 작성

BuildConfig에서는 IBM 공식 이미지를 기반으로 하는 새 이미지에 대한 빌드를 허용하지만 컨테이너 시작 시 실행할 MQSC 또는 INI 파일을 추가합니다.

##### a) BuildConfig 자원을 정의하는 YAML 파일 작성

예를 들어, 다음 콘텐츠를 사용하여 "mq-build-config.yaml"이라는 파일을 작성합니다.

```
apiVersion: build.openshift.io/v1
kind: BuildConfig
metadata:
  name: mymq
spec:
  source:
    dockerfile: |-
      FROM cp.icr.io/cp/ibm-mqadvanced-server-integration:9.1.5.0-r2-amd64
      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:9.1.5.0-r2-amd64"
      pullSecret:
        name: ibm-entitlement-key
  output:
    to:
      kind: ImageStreamTag
      name: 'mymq:latest-amd64'
```

사용하려는 버전과 수정 프로그램에 대한 올바른 기본 이미지를 가리키도록 기본 IBM MQ가 언급된 두 지점을 바꿔야 합니다. 수정사항이 적용되면 이미지를 다시 빌드하려면 이 단계를 반복해야 합니다.

이 예제는 IBM 공식 이미지를 기반으로 하는 새 이미지를 작성하고 "my.mqsc" 및 "my.ini" 파일을 /etc/mqm 디렉토리에 추가합니다. 이 디렉토리에 있는 MQSC나 INI 파일들은 시작 시 컨테이너에 의해 적용됩니다. INI 파일은 `crtmqm -ii` 옵션을 사용하여 적용되며 기존 INI 파일과 병합됩니다. MQSC 파일은 알파벳 순서로 적용됩니다.

큐 관리자가 시작할 때마다 실행되기 때문에 MQSC 명령이 반복 가능하다는 것이 중요합니다. 이는 일반적으로 DEFINE 명령에 REPLACE 매개변수를 추가하고 START 또는 STOP 명령에 IGNSTATE (YES) 매개변수를 추가하는 것을 의미합니다.

- b) 서버에 BuildConfig을(를) 적용하십시오.

```
oc apply -f mq-build-config.yaml
```

3. 빌드를 실행하여 이미지 작성

- a) 빌드 시작

```
oc start-build mymq
```

다음과 유사한 출력이 표시되어야 합니다.

```
build.build.openshift.io/mymq-1 started
```

- b) 빌드 상태 확인

예를 들어, 이전 단계에서 리턴된 빌드 식별자를 사용하여 다음 명령을 실행할 수 있습니다.

```
oc describe build mymq-1
```

4. 새 이미지를 사용하여 큐 관리자 배치

17 페이지의 『[OpenShift CLI를 사용하여 큐 관리자 배치](#)』에서 설명된 단계에 따라 새 사용자 정의 이미지를 YAML에 추가.

다음 YAML의 스니펫을 일반 QueueManager YAML에 추가할 수 있으며, 여기서 `my-namespace`는 사용 중인 OpenShift 프로젝트/네임스페이스이며 `image`는 전에 작성된 이미지의 이름(예: "mymq:latest-amd64")입니다.

```
spec:
  queueManager:
    image: image-registry.openshift-image-registry.svc:5000/my-namespace/my-image
```

## 관련 태스크

17 페이지의 『[OpenShift CLI를 사용하여 큐 관리자 배치](#)』

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

Linux

MQ Adv.

CD

## Helm 을 사용하여 IBM MQ 인증된 컨테이너 배치

IBM MQ 9.1.5.0에서 큐 관리자를 배치하는 권장 방법은 IBM MQ 연산자를 사용하는 것입니다. IBM MQ 9.1.5.0 및 이전 CD 릴리스는 다음 지시사항을 사용하여 Helm을 사용하여 배치할 수 있습니다.

## 이 태스크 정보

### 프로시저

- 21 페이지의 『[Helm 을 사용하여 OpenShift 에서 IBM MQ 에 대한 OpenShift 클러스터 준비](#)』.
- 22 페이지의 『[Helm CLI를 사용하여 큐 관리자 배치](#)』.

Linux

MQ Adv.

CD

## Helm 을 사용하여 OpenShift 에서 IBM MQ 에 대한 OpenShift 클러스터 준비

Red Hat OpenShift 컨테이너 플랫폼 클러스터를 준비하여 Helm을 사용하여 큐 관리자를 배치할 준비가 되었는지 확인하십시오. 클러스터 관리자가 이 태스크를 완료해야 합니다.

## 시작하기 전에

**참고:** IBM Cloud Pak for Integration를 사용하는 경우, 설치 프로그램은 IBM MQ와 함께 사용할 OpenShift 프로젝트(네임스페이스)를 준비해야 하므로 다음 지시사항을 따를 필요가 없습니다.

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

## 프로시저

1. IBM Helm 저장소를 Helm의 로컬 사본에 추가해야 합니다.  
예를 들어, 다음 명령을 실행할 수 있습니다.

```
helm repo add ibm-entitled-charts https://raw.githubusercontent.com/IBM/charts/master/repo/entitled
```

2. Helm 서버("Tiller"라고 함)가 클러스터에 설치되었는지 확인하십시오.  
OpenShift에서 Helm 시작하기의 지시사항에 따라 클러스터에 Helm을 설치하십시오.
3. OpenShift 프로젝트(네임스페이스)의 서비스 계정이 올바른 보안 컨텍스트 제한조건(SCC)을 사용하도록 권한부여되어야 합니다.

**V 9.1.5** IBM MQ가 "제한된" 기본 SCC에서 작동됩니다. 따라서 이 단계는 보통 건너뛴 수 있습니다.

SCC에 변경사항을 적용하려면 OpenShift 클러스터 관리자로 수행되어야 합니다. 각 Helm 차트 버전에서는 SCC에 대한 요구 사항이 다르며, 해당 Helm 차트의 개별 README 파일에서 문서화됩니다.

```
helm inspect readme ibm-entitled-charts/ibm-mqadvanced-server-prod
```

각 README에는 SCC에 대한 권한을 설정하기 위한 지시사항이 있습니다. IBM MQ Helm 차트는 자체 사용을 위한 서비스 계정을 생성합니다. 이는 네임스페이스의 모든 서비스 계정의 경우 SCC 권한이 "그룹" 레벨에서 적용되어야 한다는 것을 의미합니다.

4. 선택된 컨테이너 레지스트리에서 이미지를 가져올 유효한 "이미지 PULL 시크릿"이 있어야 합니다.  
라이선스 부여 검사를 수행하는 컨테이너 레지스트리에서 IBM MQ Advanced certified container 이미지를 가져옵니다. 이 검사에는 `docker-registry pull` 시크릿에 저장되는 인타이틀먼트 키가 필요합니다. 아직 인타이틀먼트 키가 없는 경우 다음 지시사항에 따라 인타이틀먼트 키를 가져오고 풀 시크릿을 작성하십시오.
  - a) 사용자 ID에 지정된 인타이틀먼트 키를 가져오십시오.
    - i) 자격이 있는 소프트웨어와 연관된 IBM ID 및 비밀번호를 사용하여 [MyIBM](#)에 로그인하십시오.
    - ii) 인타이틀먼트 키 섹션에서 **키 복사**를 선택하여 인타이틀먼트 키를 클립보드에 복사하십시오.
  - b) 큐 관리자를 배치할 네임스페이스에 시크릿을 작성하십시오.
    - 다음 명령을 실행하십시오. 여기서 `<entitlement-key>`는 1단계에서 검색된 키이고 `<user-email>`은 권한 부여된 소프트웨어와 연관된 IBM ID입니다.

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

## 다음에 수행할 작업

22 페이지의 『Helm CLI를 사용하여 큐 관리자 배치』

**Linux** **V 9.1.4** **MQ Adv.** **CD** **Helm CLI를 사용하여 큐 관리자 배치**

Helm을 사용하여 Red Hat OpenShift Container Platform 클러스터에 큐 관리자를 배치할 수 있습니다. 이 태스크는 프로젝트 관리자에서 완료해야 합니다.

## 시작하기 전에

Helm V2 및 Red Hat OpenShift Container Platform 명령행 인터페이스를 설치해야 합니다. IBM Cloud Pak for Integration를 사용하지 않는 경우, 21 페이지의 『Helm 을 사용하여 OpenShift 에서 IBM MQ 에 대한 OpenShift 클러스터 준비』의 단계를 수행하십시오.

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

## 프로시저

1. IBM Helm 저장소를 Helm의 로컬 사본에 추가해야 합니다.  
예를 들어, 다음 명령을 실행할 수 있습니다.

```
helm repo add ibm-entitled-charts https://raw.githubusercontent.com/IBM/charts/master/repo/entitled
```

2. 큐 관리자에 대한 구성 옵션을 검토하십시오.

설치와 구성 단계 모두가 배치 단계에 포함됩니다. 배치 시간에 큐 관리자에 대한 일부 설정이 수행되어야 하고 이를 변경하려면 재배치해야 합니다.

다음 명령 중 하나를 실행하여 사용 가능한 모든 배치 옵션에 대한 세부사항을 확인하기 위해 Helm chart README를 참조할 수 있습니다.

- IBM Cloud Pak for Integration의 IBM MQ Advanced certified container 경우:

```
helm inspect readme ibm-entitled-charts/ibm-mqadvanced-server-integration-prod
```

- IBM MQ Advanced certified container의 경우:

```
helm inspect readme ibm-entitled-charts/ibm-mqadvanced-server-prod
```

최소한 다음 매개변수가 필요합니다.

- a. 릴리스 이름(예: my-release)
  - b. 원격 Helm 저장소(예: ibm-entitled-charts)
  - c. Helm chart(예: ibm-mqadvanced-server-prod 또는 ibm-mqadvanced-server-integration-prod)
  - d. 이미지 풀 시크릿 이름(예: entitled-registry). IBM Cloud Pak for Integration에서 MQ의 사전 정의된 프로젝트에 배치 중인 경우 해당 이름이 필요없습니다.
3. 큐 관리자를 배치하십시오.

기본적으로 Helm 차트는 Red Hat OpenShift 컨테이너 플랫폼 클러스터에 기본 스토리지 클래스를 설정한다고 가정합니다.

예를 들어, IBM Cloud Pak for Integration에 기본 큐 관리자를 설치하려면 다음 명령을 실행하십시오.

```
helm install \  
--tls \  
--name my-release \  
ibm-entitled-charts/ibm-mqadvanced-server-integration-prod \  
--set license=accept \  
--set tls.hostname=my.cluster \  
--set tls.generate=true
```

tls.hostname 필드에 호스트 이름을 입력할 수 있습니다(이는 필수 필드이지만 새로운 자체 사인된 인증서를 생성하는 예제에서 사용되지 않음)

IBM Cloud Pak for Integration의 기본 큐 관리자를 독립적으로 설치하기 위해 다음 명령을 실행할 수 있습니다.

```
helm install \  
--name my-release \  
ibm-entitled-charts/ibm-mqadvanced-server-prod \  

```

```
--set license=accept \  
--set image.pullSecret=ibm-entitlement-key
```

## 관련 태스크

28 페이지의 『[OpenShift 클러스터에 배치된 큐 관리자에 연결](#)』

Red Hat OpenShift 클러스터에 배치된 큐 관리자에 연결하기 위한 구성 예제 세트입니다.

29 페이지의 『[OpenShift 클러스터에 배치된 IBM MQ Console에 연결](#)』

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

Linux

MQ Adv.

CD

V 9.1.5

Helm CLI를 사용하여 IBM Cloud File

## Storage로 큐 관리자 배치

Helm 을 사용하여 IBM Cloud File Storage를 사용하여 IBM Cloud 클러스터의 Red Hat OpenShift 에 큐 관리자를 배치하는 예제 시나리오입니다. 프로젝트 관리자가 이 태스크를 완료해야 함

## 시작하기 전에

Helm V2 및 Red Hat OpenShift Container Platform 명령행 인터페이스를 설치해야 합니다. IBM Cloud Pak for Integration를 사용하지 않는 경우, 21 페이지의 『[Helm 을 사용하여 OpenShift 에서 IBM MQ 에 대한 OpenShift 클러스터 준비](#)』의 단계를 수행하십시오.

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

## 프로시저

1. IBM Helm 저장소를 Helm의 로컬 사본에 추가했는지 확인하십시오.  
예를 들어, 다음 명령을 실행할 수 있습니다.

```
helm repo add ibm-entitled-charts https://raw.githubusercontent.com/IBM/charts/master/rep/entitled
```

2. 큐 관리자를 배치하십시오.

IBM Cloud File Storage를 사용하는 경우 일반적으로 `ibmc-file-gold-gid` 스토리지 클래스를 사용하여 최상의 결과를 볼 수 있습니다. 이 스토리지 클래스를 사용하면 올바른 파일 시스템 그룹의 사용자가 작성할 수 있는 스토리지를 사용할 수 있습니다.

예를 들어, IBM Cloud Pak for Integration에 기본 큐 관리자를 설치하려면 다음 명령을 실행하십시오.

```
helm install \  
--tls \  
--name my-release \  
ibm-entitled-charts/ibm-mqadvanced-server-integration-prod \  
--set license=accept \  
--set tls.hostname=my.cluster \  
--set tls.generate=true \  
--set dataPVC.storageClassName=ibmc-file-gold-gid \  
--set security.context.supplementalGroups={99}
```

`tls.hostname` 필드에 호스트 이름을 입력할 수 있습니다 (이 예제에서는 새 자체 서명 인증서를 생성하기 때문에 필수 필드이지만 여기에서는 사용되지 않습니다).

IBM Cloud Pak for Integration의 기본 큐 관리자를 독립적으로 설치하기 위해 다음 명령을 실행할 수 있습니다.

```
helm install \  
--name my-release \  
ibm-entitled-charts/ibm-mqadvanced-server-prod \  
--set license=accept \  
--set image.pullSecret=ibm-entitlement-key \  
--set dataPVC.storageClassName=ibmc-file-gold-gid \  
--set security.context.supplementalGroups={99}
```



## 관련 태스크

28 페이지의 『[OpenShift 클러스터에 배치된 큐 관리자에 연결](#)』

Red Hat OpenShift 클러스터에 배치된 큐 관리자에 연결하기 위한 구성 예제 세트입니다.

29 페이지의 『[OpenShift 클러스터에 배치된 IBM MQ Console에 연결](#)』

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

## Linux > MQ Adv. > CD **Deploying previous CD releases of IBM MQ into an IBM Cloud Private cluster**

9.1.4이전의 IBM MQ 의 CD 버전의 경우, IBM Cloud Private 관리 콘솔을 사용하여 큐 관리자를 IBM Cloud Private에 배치하십시오.

### 시작하기 전에



**주의:** > **V 9.1.4** IBM MQ 9.1.4 이상 버전에서는 이 배치가 지원되지 않습니다.

이 태스크에서는 이미 [에서 IBM MQ 이미지를 IBM Cloud Private 클러스터에 추가했다고 가정합니다.](#)

Helm 차트 README.md 파일은 IBM Cloud Private 카탈로그 항목에서 사용 가능합니다. 이 카탈로그 항목은 [이하 단계를 완료하거나 명령행에서 IBM Cloud Private의 local-charts 저장소를 원격 Helm 저장소로 추가](#) 하고 다음 명령을 실행하여 표시됩니다.

```
helm inspect readme remote_repo_name/ibm-mqadvanced-server-prod
```

필요한 보안 컨텍스트를 지원하는 [PodSecurity](#) 정책 또는 [SecurityContext](#) 제한조건 ( Red Hat OpenShift의 경우 IBM Cloud Private ) 이 있어야 합니다. 예제를 포함한 세부사항은 Helm 차트 README.md 파일에서 찾을 수 있습니다.

Helm 릴리스를 구성하는 방법에 대한 세부사항도 Helm 차트 README.md 파일에서 찾을 수 있습니다.

### 참고:

- 기본적으로 필수 보안 설정을 지원하지 않는 IBM Cloud Private 환경에 배치하는 경우, IBM Cloud Private 제품 문서의 [기본이 아닌 네임스페이스에서 승격된 권한이 필요한 Helm 차트 배치](#) 의 지시사항에 따라 배치를 사용 가능하게 하십시오.
- SELinux를 사용하는 경우 [IBM MQ support for SELinux on Red Hat Enterprise Linux](#)에 설명된 IBM MQ 요구 사항을 충족해야 합니다.

### 이 태스크 정보

IBM Cloud Private에서는 온프레미스, 컨테이너화된 애플리케이션을 관리하기 위한 플랫폼을 제공합니다. IBM MQ 이미지를 IBM Cloud Private 클러스터에 추가한 후에 IBM Cloud Private 관리 콘솔 또는 명령행을 사용하여 큐 관리자를 배치할 수 있습니다.

### 프로시저

- IBM Cloud Private Management Console 사용
  - a) 웹 브라우저에서 IBM Cloud Private 관리 콘솔을 열고 [카탈로그](#)를 클릭하십시오.  
IBM Cloud Private 제품 문서의 [관리 콘솔을 사용하여 IBM Cloud Private 클러스터 액세스](#) 를 참조하십시오.
  - b) 목록에서 `ibm-mqadvanced-server-prod` 차트를 선택하십시오.
  - c) 구성을 선택한 후 다음 구성 단계를 완료하십시오.
    - a. 릴리스 이름을 입력하십시오.
    - b. 라이선스 계약을 읽고 승인하십시오.

- c. **dataPVC** 섹션 아래에서 **storageclass**를 원하는 스토리지 클래스로 설정하십시오. 기본 스토리지 클래스를 선택하려면 공백으로 두십시오.
- d. **이미지** 섹션 아래에서 저장소를 전체 이미지 경로로 설정하십시오. 예를 들면, 다음과 같습니다.

```
mycluster.icp:8500/namespace_name/ibm-mqadvanced-server-prod
```

- e. **이미지** 섹션 아래에서 태그를 이미지 태그로 설정하십시오. 예를 들면, 다음과 같습니다.

```
9.1.3.0-r1
```

- f. 이미지 레지스트리에 액세스하기 위해 Kubernetes 풀 시크릿이 필요한 경우 이를 **pullSecret**으로 추가하십시오.

- g. **queueManager** 섹션 아래에서 큐 관리자의 이름을 설정하십시오.

- d) 설치를 클릭하여 큐 관리자를 **Helm** 릴리스로 배치하십시오.
- 명령행 사용

- a) **cloudctl** 를 구성하여 IBM Cloud Private 클러스터에 액세스하십시오.

IBM Cloud Private 제품 문서에서 [IBM Cloud Private CLI 설치](#) 를 참조하십시오.

- b) IBM Cloud Private의 **local-charts** 저장소를 원격 Helm 저장소로 추가했는지 확인하십시오.

- c) 차트를 설치하십시오.

다음 매개변수를 지정하여 다음 명령을 실행하십시오.

- a. 릴리스 이름(예: my-release)
- b. ibm-mqadvanced-server-prod 차트를 포함하는 원격 helm 저장소 이름(예: my-repo)
- c. 이미지 저장소(예: mycluster.icp:8500/namespace\_name/ibm-mqadvanced-server-prod)
- d. 이미지 태그(예: 9.1.3.0-r1)

```
helm install --name my-release --repo my-repo ibm-mqadvanced-server-prod --set license=accept --set image.repository=mycluster.icp:8500/namespace_name/ibm-mqadvanced-server-prod --set image.tag=9.1.3.0-r1 --tls
```

## 관련 태스크

22 페이지의 [『Helm CLI를 사용하여 큐 관리자 배치』](#)

Helm을 사용하여 Red Hat OpenShift Container Platform 클러스터에 큐 관리자를 배치할 수 있습니다. 이 태스크는 프로젝트 관리자에서 완료해야 합니다.

26 페이지의 [『IBM MQ 이미지의 이전 CD 릴리스를 IBM Cloud Private 클러스터에 추가』](#)

For CD versions of IBM MQ earlier than 9.1.4, prepare your IBM Cloud Private cluster to deploy a production-ready image for IBM MQ.

27 페이지의 [『IBM MQ 이미지의 이전 CD 릴리스를 IBM Cloud Kubernetes Service 클러스터에 추가』](#)

9.1.4이전의 IBM MQ 의 CD 버전의 경우 IBM MQ 에 대한 프로덕션 준비 이미지를 IBM Cloud Kubernetes Service로 가져오십시오.

Linux

MQ Adv.

CD

## IBM MQ 이미지의 이전 CD 릴리스를 IBM Cloud Private 클러스터에 추가

For CD versions of IBM MQ earlier than 9.1.4, prepare your IBM Cloud Private cluster to deploy a production-ready image for IBM MQ.

### 이 태스크 정보



주의: **V 9.1.4** IBM MQ 9.1.4 이상 버전에서는 이 가져오기가 지원되지 않습니다.

Passport Advantage 에서 IBM MQ 이미지를 다운로드하여 IBM Cloud Private 컨테이너로 가져올 수 있습니다.

## 프로시저

1. [Passport Advantage 및 Passport Advantage Express® 웹 사이트](#)에서 최신 IBM MQ 이미지를 다운로드하십시오.

사용 가능한 다운로드에 대한 세부사항은 [IBM MQ 9.1 다운로드](#)로 이동한 후 다운로드하려는 릴리스의 탭을 클릭하십시오. 다운로드할 파트의 이름 및 번호가 테이블에 나열됩니다.

2. 다운로드된 아카이브 파일을 IBM Cloud Private로 가져오십시오.

IBM Cloud Private 제품 문서에서 [Adding IBM software to the IBM Cloud Private Catalog](#) 를 참조하십시오.

## 다음에 수행할 작업

이제 [IBM Cloud Private에 큐 관리자를 배치할 준비](#)가 되었습니다.

### 관련 태스크

22 페이지의 [『Helm CLI를 사용하여 큐 관리자 배치』](#)

Helm을 사용하여 Red Hat OpenShift Container Platform 클러스터에 큐 관리자를 배치할 수 있습니다. 이 태스크는 프로젝트 관리자에서 완료해야 합니다.

25 페이지의 [『Deploying previous CD releases of IBM MQ into an IBM Cloud Private cluster』](#)

9.1.4이전의 IBM MQ 의 CD 버전의 경우, IBM Cloud Private 관리 콘솔을 사용하여 큐 관리자를 IBM Cloud Private에 배치하십시오.

27 페이지의 [『IBM MQ 이미지의 이전 CD 릴리스를 IBM Cloud Kubernetes Service 클러스터에 추가』](#)

9.1.4이전의 IBM MQ 의 CD 버전의 경우 IBM MQ 에 대한 프로덕션 준비 이미지를 IBM Cloud Kubernetes Service로 가져오십시오.

Linux

MQ Adv.

CD

## IBM MQ 이미지의 이전 CD 릴리스를 IBM Cloud Kubernetes Service 클러스터에 추가

9.1.4이전의 IBM MQ 의 CD 버전의 경우 IBM MQ 에 대한 프로덕션 준비 이미지를 IBM Cloud Kubernetes Service로 가져오십시오.

### 이 태스크 정보



주의: **V 9.1.4** IBM MQ 9.1.4 이상 버전에서는 이 가져오기가 지원되지 않습니다.

Passport Advantage 에서 IBM MQ 이미지를 다운로드하여 IBM Cloud Kubernetes Service 클러스터로 가져올 수 있습니다.

## 프로시저

1. [Passport Advantage 및 Passport Advantage Express 웹 사이트](#)에서 최신 IBM MQ 이미지를 다운로드하십시오.

사용 가능한 다운로드에 대한 세부사항은 [IBM MQ 9.1 다운로드](#)로 이동한 후 다운로드하려는 릴리스의 탭을 클릭하십시오. 다운로드할 파트의 이름 및 번호가 테이블에 나열됩니다.

2. 다운로드된 아카이브 파일을 IBM Cloud Kubernetes Service로 가져오십시오.

공용 Kubernetes 컨테이너에서 [IBM Cloud Private 이미지 실행](#)을 참조하십시오.

### 관련 태스크

22 페이지의 [『Helm CLI를 사용하여 큐 관리자 배치』](#)

Helm을 사용하여 Red Hat OpenShift Container Platform 클러스터에 큐 관리자를 배치할 수 있습니다. 이 태스크는 프로젝트 관리자에서 완료해야 합니다.

25 페이지의 [『Deploying previous CD releases of IBM MQ into an IBM Cloud Private cluster』](#)

9.1.4이전의 IBM MQ 의 CD 버전의 경우, IBM Cloud Private 관리 콘솔을 사용하여 큐 관리자를 IBM Cloud Private에 배치하십시오.

26 페이지의 [『IBM MQ 이미지의 이전 CD 릴리스를 IBM Cloud Private 클러스터에 추가』](#)

For CD versions of IBM MQ earlier than 9.1.4, prepare your IBM Cloud Private cluster to deploy a production-ready image for IBM MQ.

## Linux > V 9.1.4 > MQ Adv. > CD **OpenShift 클러스터에 배치된 큐 관리자** **자에 연결**

Red Hat OpenShift 클러스터에 배치된 큐 관리자에 연결하기 위한 구성 예제 세트입니다.

### 이 태스크 정보

Red Hat OpenShift 클러스터 외부에서 IBM MQ 큐 관리자에 응용프로그램을 연결하려면 [OpenShift 라우트](#) 가 필요합니다.

서버 이름 표시 (SNI) 는 TLS 프로토콜에서만 사용할 수 있으므로 IBM MQ 큐 관리자 및 클라이언트 애플리케이션에서 TLS를 사용 가능하게 해야 합니다. Red Hat OpenShift Container Platform 라우터는 IBM MQ 큐 관리자로 요청을 라우팅하기 위해 SNI를 사용합니다.

OpenShift Route에 대한 필수 구성은 클라이언트 애플리케이션의 SNI 동작에 따라 달라집니다.

SNI 헤더를 TLS 1.2 이상으로 설정하려면 TLS 통신에 CipherSpec 또는 CipherSuite 를 사용해야 합니다.

SNI는 다음 조건이 충족되는 경우 MQ 채널로 설정됩니다.

- IBM MQ C 클라이언트는 V8 이상입니다.
- Java/JMS 클라이언트는 V9.1.1 이상이며 Java 설치에는 `javax.net.ssl.SNIHostName` 클래스를 지원합니다.
- .NET 클라이언트가 관리되지 않는 모드에 있습니다.

호스트 이름이 연결 이름으로 제공되고 다음 조건이 충족되면 SNI는 호스트 이름으로 설정됩니다.

- .NET 클라이언트가 관리 모드에 있다.
- AMQP 또는 XR 클라이언트가 사용된다.
- Java/JMS 클라이언트는 **AllowOutboundSNI** 에서 NO로 설정된 경우에 사용됩니다.

SNI는 설정되지 않고 다음 조건 하에서는 공백이다.

- IBM MQ C 클라이언트는 V7.5 이하입니다.
- IBM MQ C 클라이언트는 **AllowOutboundSNI** 에서 아니오로 설정된 경우에 사용됩니다.
- Java/JMS 클라이언트는 `javax.net.ssl.SNIHostName` 클래스를 지원하지 않는 Java 설치와 함께 사용됩니다.

### 예

#### **OpenShift Route 기반 호스트 이름 : SNI를 호스트 이름으로 설정하는 클라이언트 애플리케이션의 경우**

다음 Helm 차트는 애플리케이션을 IBM MQ 큐 관리자에 연결하기 위해 자동으로 OpenShift Route 기반 호스트 이름을 작성합니다. SNI를 호스트 이름으로 설정하는 클라이언트 애플리케이션은 이 OpenShift Route를 사용할 수 있습니다.

- `ibm-mqadvanced-server-dev`
- `ibm-mqadvanced-server-prod`
- `ibm-mqadvanced-server-integration-prod`(IBM Cloud Pak for Integration)

이러한 차트를 사용하지 않고 고유한 OpenShift Route 기반 호스트 이름을 작성해야 하는 경우 클러스터에서 다음 `yaml`을 적용할 수 있습니다.

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: <provide a unique name for the Route>
  namespace: <namespace of your MQ deployment>
spec:
```

```

to:
  kind: Service
  name: <name of the Kubernetes Service for your MQ deployment (for example "<Helm Release>-ibm-mq")>
  port:
    targetPort: 1414
  tls:
    termination: passthrough

```

### OpenShift Route 기반 MQ 채널 : SNI를 MQ 채널로 설정하는 클라이언트 애플리케이션

SNI를 MQ 채널로 설정하는 클라이언트 애플리케이션의 경우 연결할 각각의 채널에 대해 새 OpenShift Route를 작성해야 합니다. 또한 올바른 큐 관리자로 라우팅하기 위해 Red Hat OpenShift 클러스터 전체에서 고유한 채널 이름을 사용해야 합니다.

각각의 새 OpenShift Route에 필요한 호스트 이름을 판별하려면 <https://www.ibm.com/support/pages/ibm-websphere-mq-how-does-mq-provide-multiple-certificates-certlabl-capability>의 설명에 따라 각각의 채널 이름을 SNI 주소로 매핑해야 합니다.

그런 다음 클러스터에서 다음 yaml을 적용하여 새 OpenShift Route를 작성해야 합니다(각각의 채널에 대해).

```

apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: <provide a unique name for the Route>
  namespace: <the namespace of your MQ deployment>
spec:
  host: <SNI address mapping for the channel>
  to:
    kind: Service
    name: <the name of the Kubernetes Service for your MQ deployment (for example "<Helm Release>-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 "<Helm Release>-ibm-mq-qm")>
-n <namespace of your MQ deployment> -o jsonpath="{.spec.host}"

```

클라이언트 연결용 포트를 OCP(OpenShift Container Platform) Router에서 사용하는 포트로 설정하십시오(일반적으로 443).

### 관련 태스크

22 페이지의 『[Helm CLI를 사용하여 큐 관리자 배치](#)』

Helm을 사용하여 Red Hat OpenShift Container Platform 클러스터에 큐 관리자를 배치할 수 있습니다. 이 태스크는 프로젝트 관리자에서 완료해야 합니다.

29 페이지의 『[OpenShift 클러스터에 배치된 IBM MQ Console에 연결](#)』

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

## Linux > V 9.1.4 > MQ Adv. > CD > OpenShift 클러스터에 배치된 IBM MQ Console에 연결

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

## 이 태스크 정보

IBM MQ 연산자를 사용하는 경우, IBM MQ Console URL은 OpenShift 웹 콘솔 또는 IBM Cloud Pak for Integration Platform Navigator의 QueueManager 세부사항 페이지에서 찾을 수 있습니다. 또는 다음 명령을 실행하여 OpenShift CLI에서 찾을 수 있습니다.

```
oc get queuemanager <QueueManager Name> -n <namespace of your MQ deployment> --output jsonpath='{.status.adminUiUrl}'
```

## 예

다음 Helm 차트는 IBM MQ Console에 액세스하기 위해 자동으로 OpenShift Route를 작성합니다.

- ibm-mqadvanced-server-dev
- ibm-mqadvanced-server-integration-prod(IBM Cloud Pak for Integration)

다음 명령을 실행하여 OpenShift Route의 호스트 이름을 가져올 수 있습니다.

```
oc get route <Route Name (for example "<Helm Release>-ibm-mq-web")> -n <namespace of your MQ deployment> --output jsonpath='{.spec.host}'
```

다음 URL을 사용하여 IBM MQ Console에 액세스할 수 있습니다.

```
https://<Route Hostname>/ibmmq/console
```

## 관련 태스크

[22 페이지의 『Helm CLI를 사용하여 큐 관리자 배치』](#)

Helm을 사용하여 Red Hat OpenShift Container Platform 클러스터에 큐 관리자를 배치할 수 있습니다. 이 태스크는 프로젝트 관리자에서 완료해야 합니다.

[28 페이지의 『OpenShift 클러스터에 배치된 큐 관리자에 연결』](#)

Red Hat OpenShift 클러스터에 배치된 큐 관리자에 연결하기 위한 구성 예제 세트입니다.

Linux

MQ Adv.

CD

## OpenShift CLI를 사용하여 큐 관리자 구성 백업

### 및 복원

큐 관리자 구성을 백업하면 큐 관리자 구성이 유실된 경우 해당 정의에서 큐 관리자를 다시 빌드할 수 있습니다. 이 프로시저에서는 큐 관리자 로그 데이터를 백업하지 않습니다. 메시지의 임시 네이처 때문에 히스토리 로그 데이터는 복원 시점에서 무관할 수 있습니다.

## 시작하기 전에

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

## 프로시저

- 큐 관리자 구성을 백업하십시오.

**dmpmqcfg** 명령을 사용하여 IBM MQ 큐 관리자의 구성을 덤프할 수 있습니다.

- a) 큐 관리자의 팟(pod) 이름을 가져옵니다.

예를 들어, 연산자를 사용하는 경우 다음 명령을 실행할 수 있습니다. 여기서 *queue\_manager\_name* 은 QueueManager 자원의 이름입니다.

```
oc get pods --selector app.kubernetes.io/name=ibm-mq,app.kubernetes.io/instance=queue_manager_name
```

예를 들어, Helm을 사용하는 경우 다음 명령을 실행할 수 있습니다. 여기서 `release_name` 은 Helm 릴리스의 이름입니다.

```
oc get pods --selector release=release_name
```

b) 로컬 시스템의 파일로 출력 경로를 지정하여 팟(Pod)에서 `dmpmqcfig` 명령을 실행하십시오.

`dmpmqcfig`는 큐 관리자의 MQSC 구성을 출력합니다.

```
oc exec -it pod_name -- dmpmqcfig > backup.mqsc
```

• 큐 관리자 구성을 복원하십시오.

이전 단계에서 간략히 설명한 백업 프로시저에 따르는 경우 큐 관리자 구성을 포함하는 `backup.mqsc` 파일을 보유해야 합니다. 이 파일을 새 큐 관리자에 적용하여 구성을 복원할 수 있습니다.

a) 큐 관리자의 팟(`pod`) 이름을 가져옵니다.

예를 들어, 연산자를 사용하는 경우 다음 명령을 실행할 수 있습니다. 여기서 `queue_manager_name` 은 QueueManager 자원의 이름입니다.

```
oc get pods --selector app.kubernetes.io/name=ibm-mq,app.kubernetes.io/instance=queue_manager_name
```

예를 들어, Helm을 사용하는 경우 다음 명령을 실행할 수 있습니다. 여기서 `release_name` 은 Helm 릴리스의 이름입니다.

```
oc get pods --selector release=release_name
```

b) `backup.mqsc` 파일의 콘텐츠 경로를 지정하여 팟(Pod)에서 `runmqsc` 명령을 실행하십시오.

```
oc exec -i pod_name -- runmqsc < backup.mqsc
```

## IBM MQ 컨테이너 빌드

이전에는 "Docker 컨테이너 이미지"로 참조되었던 자체 빌드 컨테이너를 개발하십시오. 이 솔루션은 가장 유연한 컨테이너 솔루션이지만, 이 솔루션을 사용하려면 컨테이너를 구성하는 측면에서 뛰어난 기술을 보유하고 있어야 하며 결과 컨테이너를 "소유"하고 있어야 합니다.

### 시작하기 전에

자체 컨테이너를 개발하기 전에 IBM에서 제공하는 패키징되기 이전의 컨테이너 중 하나를 대신 사용할 수 있는지 여부를 고려하십시오. [컨테이너의 IBM MQ](#)를 참조하십시오.

### 이 태스크 정보

컨테이너 이미지로 IBM MQ를 패키징할 때, 애플리케이션에 대한 변경사항이 테스트와 스테이징 시스템에 빠르고 쉽게 배치될 수 있습니다. 이는 엔터프라이즈의 Continuous Delivery에 기본적으로 이익이될 수 있습니다.

### 프로시저

- Docker를 사용하여 IBM MQ 컨테이너 이미지를 빌드하는 방법에 대한 정보는 다음 하위 주제를 참조하십시오.
  - [Linux](#) 8 페이지의 『IBM MQ 컨테이너 이미지 및 차트 빌드 지원』
  - 32 페이지의 『컨테이너를 사용하여 IBM MQ 큐 관리자 이미지 계획』
  - 32 페이지의 『Docker 를 사용하여 샘플 IBM MQ 큐 관리자 이미지 빌드』
  - 34 페이지의 『별도의 컨테이너에서 로컬 바인딩 애플리케이션 실행』

### 관련 개념

[컨테이너의 IBM MQ](#)

## 컨테이너를 사용하여 IBM MQ 큐 관리자 이미지 계획

컨테이너에서 IBM MQ 큐 관리자를 실행할 때 고려할 여러 요구사항이 있습니다. 샘플 컨테이너 이미지에서는 이러한 요구사항을 처리할 방법을 제공하지만, 자체 이미지를 사용하려는 경우 이러한 요구사항이 처리하는 방법에 대해 고려해야 합니다.

### 프로세스 감시

컨테이너를 실행할 때, 단일 프로세스(컨테이너 내의 PID 1)를 실행 중이고, 이를 통해 이후에 하위 프로세스를 올바르게 실행시킬 수 있습니다.

기본 프로세스가 종료되면 컨테이너 런타임이 컨테이너를 중지합니다. IBM MQ 큐 관리자는 여러 프로세스가 백그라운드에서 실행 중이어야 합니다.

이 때문에, 큐 관리자가 실행하고 있는 동안에는 기본 프로세스가 활성 상태로 유지하도록 보장해야 합니다. 큐 관리자가 이 프로세스로부터 활성화 상태인지 확인하는 것이 좋습니다(예: 관리 쿼리 수행).

### /var/mqm 채우기

컨테이너는 /var/mqm을 볼륨으로 사용하여 구성되어야 합니다.

이를 수행할 때, 볼륨의 디렉토리는 컨테이너가 처음 시작할 때 비어 있습니다. 이 디렉토리는 주로 설치시 채워지지만 설치 및 런타임은 컨테이너 사용시 독립 환경입니다.

**V 9.1.0** 이를 해결하기 위해 컨테이너가 시작될 때 `crtmqdir` 명령을 사용하여 처음 실행될 때 /var/mqm 를 채울 수 있습니다.

## Docker 를 사용하여 샘플 IBM MQ 큐 관리자 이미지 빌드

이 정보를 사용하여 컨테이너에서 IBM MQ 큐 관리자를 실행하기 위한 샘플 컨테이너 이미지를 빌드하십시오.

### 이 태스크 정보

먼저, Red Hat Universal 기본 이미지 파일 시스템 및 IBM MQ의 깔끔한 설치를 포함하는 기본 이미지를 빌드하십시오.

두 번째로, 기본 맨 위에 있는 다른 컨테이너 이미지 계층을 빌드하고, 이는 일부 IBM MQ 구성을 기본 사용자 ID 및 비밀번호 보안을 허용하도록 추가됩니다.

마지막으로, 호스트 파일 시스템의 컨테이너 특정 볼륨에서 제공되는 /var/mqm의 콘텐츠와 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.
  - [32 페이지의 『샘플 기본 IBM MQ 큐 관리자 이미지 빌드』](#)
  - [33 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 저장소](#)에서 제공되는 make 파일을 사용하여 프로덕션 컨테이너 이미지를 빌드하십시오.  
GitHub에 있는 [Building a container image](#)의 지시사항을 따르십시오.



## 결과

IBM MQ가 있는 기본 컨테이너 이미지가 설치되어야 합니다.

## 샘플 구성된 IBM MQ 큐 관리자 이미지 빌드

일반 기본 IBM MQ 컨테이너 이미지를 빌드하고 나면 보안 액세스를 허용하도록 구성을 적용해야 합니다. 이를 적용하려면 일반 이미지를 상위로 사용하여 컨테이너 이미지 계층을 작성하십시오.

## 시작하기 전에

IBM MQ 9.1 이미지의 경우 Red Hat OpenShift Container Platform "제한된" 보안 컨텍스트 제한조건 (SCC) 을 사용하여 보안 액세스를 구성할 수 없습니다. "restricted" SCC는 무작위 사용자 ID를 사용하며 다른 사용자로 변경하여 권한 에스컬레이션을 방지합니다. IBM MQ 9.1 RPM 기반 설치 프로그램은 mqm 사용자 및 그룹에 의존하며 실행 가능 프로그램에서 `setuid` 비트도 사용합니다.

이 제한사항은 IBM MQ 9.2에서 제거됩니다.

## 프로시저

1. 새 디렉토리를 작성하고, 다음 콘텐츠를 갖는 `config.mqsc`라는 파일을 추가하십시오.

```
DEFINE CHANNEL(PASSWORD.SVRCONN) CHLTYPE(SVRCONN)
SET CHLAUTH(PASSWORD.SVRCONN) TYPE(BLOCKUSER) USERLIST('nobody') +
DESCR('Allow privileged users on this channel')
SET CHLAUTH('*') TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS) DESCR('BackStop rule')
SET CHLAUTH(PASSWORD.SVRCONN) TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(CHANNEL) CHCKCLNT(REQUIRED)
ALTER AUTHINFO(SYSTEM.DEFAULT.AUTHINFO.IDPWOS) AUTHTYPE(IDPWOS) ADOPTCTX(YES)
REFRESH SECURITY TYPE(CONNAUTH)
```

이전 예제는 단순 사용자 ID 및 비밀번호 인증을 사용한다는 사실에 유의하십시오. 그러나 엔터프라이즈가 요구하는 모든 보안 구성을 적용할 수 있습니다.

2. 다음 콘텐츠를 갖는 `Dockerfile`이라는 파일을 작성하십시오.

```
FROM mq
RUN useradd johndoe -G mqm && \
    echo johndoe:passw0rd | chpasswd
COPY config.mqsc /etc/mqm/
```

설명:

- johndoe는 추가하려는 사용자 ID입니다.
- passw0rd는 원래 비밀번호입니다.

3. 다음 명령을 사용하여 사용자 정의 컨테이너 이미지를 빌드하십시오.

```
sudo 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.

**RUN** 명령은 비밀번호가 `passw0rd`인 `johndoe`라는 사용자를 추가하고, **COPY** 명령은 상위 이미지에 알려진 특정 위치에 `config.mqsc` 파일을 추가합니다.

**참고:** Red Hat Enterprise Linux(RHEL)에서 **docker**(RHEL V7) 또는 **podman**(RHEL V7 또는 RHEL V8) 명령을 사용하십시오. **podman**의 경우, 명령 시작시 **sudo**가 필요하지 않습니다.

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.

```
sudo docker run \
  --env LICENSE=accept \
  --env MQ_QMGR_NAME=QM1 \
  --volume /var/example:/var/mqm \
  --publish 1414:1414 \
  --detach \
  mymq
```

여기서,

#### 첫 번째 env 매개변수

환경 변수를 컨테이너로 전달하는데, 이것은 IBM IBM WebSphere® MQ에 대한 라이선스의 허용을 수신 확인합니다. 또한 해당 라이선스를 보기 위해 LICENSE 변수를 view로 설정할 수도 있습니다.

IBM MQ 라이선스에 대한 자세한 정보는 [IBM MQ 라이선스 정보](#) 를 참조하십시오.

#### 두 번째 env 매개변수

사용 중인 큐 관리자 이름을 설정합니다.

#### Volume 매개변수

MQ가 `/var/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 매핑합니다. 컨테이너는 기본적으로 자체 내부 IP 주소를 갖고 실행하며, 이것은 노출시키려는 임의의 포트를 특정하게 매핑해야 함을 의미합니다.

이 예에서는 호스트의 포트 1414를 컨테이너의 포트 1414에 매핑함을 의미합니다.

#### Detach 매개변수

컨테이너를 백그라운드에서 실행합니다.

## 결과

구성된 컨테이너 이미지를 빌드하고 `docker ps` 명령을 사용하여 실행 중인 컨테이너를 확인할 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

- Docker `ps` 명령을 사용할 때 컨테이너가 표시되지 않는 경우 컨테이너가 실패했을 수 있습니다. `docker ps -a` 명령을 사용하여 실패한 컨테이너를 볼 수 있습니다.
- Docker `ps -a` 명령을 사용할 때, 컨테이너 ID가 표시됩니다. Docker `run` 명령을 실행할 때 이 ID가 인쇄됩니다.
- Docker `logs ${CONTAINER_ID}` 명령을 사용하여 컨테이너의 로그를 볼 수 있습니다.
- `sysctl fs.file-max=524288` 명령을 사용하여 열린 파일의 최대 수를 설정할 수 있습니다.

### V 9.1.0

## 별도의 컨테이너에서 로컬 바인딩 애플리케이션 실행

Docker의 컨테이너 간에 프로세스 네임스페이스 공유를 사용하면 IBM MQ 큐 관리자의 별도 컨테이너에서 IBM MQ에 로컬 바인딩이 필요한 애플리케이션을 실행할 수 있습니다.

## 이 태스크 정보

이 기능은 IBM MQ 9.0.3 이상 큐 관리자에서 지원됩니다.

다음 제한사항을 준수해야 합니다.

- --pid 인수를 사용하여 컨테이너 PID 네임스페이스를 공유해야 합니다.
- --ipc 인수를 사용하여 컨테이너 IPC 네임스페이스를 공유해야 합니다.
- 다음 중 하나를 수행해야 합니다.
  1. --uts 인수를 사용하여 컨테이너 UTS 네임스페이스를 호스트와 공유하십시오.
  2. -h 또는 --hostname 인수를 사용하여 컨테이너가 동일한 호스트 이름을 사용하는지 확인하십시오.
- /var/mqm 디렉토리 아래의 모든 컨테이너에서 사용할 수 있는 볼륨에 IBM MQ 데이터 디렉토리를 마운트해야 합니다.

이미 Docker 가 설치된 Linux 시스템에서 다음 단계를 완료하여 이 기능을 사용할 수 있습니다.

다음 예제는 샘플 IBM MQ 컨테이너 이미지를 사용합니다. [Github](#)에서 이 이미지에 대한 세부사항을 찾을 수 있습니다.

## 프로시저

1. 다음 명령을 발행하여 볼륨으로 사용할 임시 디렉토리를 작성하십시오.

```
mkdir /tmp/dockerVolume
```

2. 다음 명령을 발행하여 sharedNamespace라는 이름으로 컨테이너에 큐 관리자(QM1)를 작성하십시오.

```
docker run -d -e LICENSE=accept -e MQ_QMGR_NAME=QM1 --volume /tmp/dockerVol:/mnt/mqm --uts host --name sharedNamespace ibmcom/mq
```

3. secondaryContainer라는 두 번째 컨테이너를 시작하고, ibmcom/mq를 기반으로 하지만 다음 명령을 실행하여 큐 관리자를 작성하지 마십시오.

```
docker run --entrypoint /bin/bash --volumes-from sharedNamespace --pid container:sharedNamespace --ipc container:sharedNamespace --uts host --name secondaryContainer -it --detach ibmcom/mq
```

4. 두 번째 컨테이너에서 **dspmqs** 명령을 다음과 같이 실행하여 두 큐 관리자의 상태를 확인하십시오.

```
docker exec secondaryContainer dspmqs
```

5. 다음 명령을 실행하여 다른 컨테이너에서 실행되는 큐 관리자에 대해 MQSC 명령을 처리하십시오.

```
docker exec -it secondaryContainer runmqsc QM1
```

## 결과

이제 별도의 컨테이너에서 로컬 애플리케이션이 실행 중이며 이제 두 번째 컨테이너에서 **dspmqs**, **amqsput**, **amqsget** 및 **runmqsc**와 같은 명령을 QM1 큐 관리자에 대한 로컬 바인딩으로 성공적으로 실행할 수 있습니다.

예상한 결과를 얻지 못한 경우 자세한 정보는 35 페이지의 『네임스페이스 애플리케이션 문제점 해결』의 내용을 참조하십시오.

## ▶ V9.1.0 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 34 페이지의 『별도의 컨테이너에서 로컬 바인딩 애플리케이션 실행』의 내용을 참조하십시오.

애플리케이션이 나열된 제한사항을 모두 충족하지 못할 경우, 컨테이너는 시작되지만 예상 기능이 작동하지 않는 문제점이 발견될 수 있습니다.

다음 목록에는 몇 가지 공통 원인 및 제한사항 중 하나가 충족되지 않았는지 확인할 수 있는 동작이 설명되어 있습니다.

- 네임스페이스 (UTS/PID/IPC) 또는 컨테이너의 호스트 이름을 공유하는 것을 잊고 볼륨을 마운트하는 경우, 컨테이너는 큐 관리자를 볼 수 있지만 큐 매니저와는 상호 작용하지 않습니다.
  - **dspmq** 명령의 경우 다음이 표시됩니다.

```
docker exec container dspmq
QMNAME(QM1)                STATUS(Status not available)
```

- **runmqsc** 명령 또는 큐 관리자에 연결하려고 시도하는 다른 명령의 경우 AMQ8146 오류 메시지가 수신될 수 있습니다.

```
docker exec -it container runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2024.
Starting MQSC for queue manager QM1.
AMQ8146: IBM MQ queue manager not available
```

- 필요한 모든 네임스페이스를 공유하지만 공유 볼륨을 /var/mqm 디렉토리에 마운트하지 않은 경우 유효한 IBM MQ 데이터 경로가 있으면 해당 명령이 AMQ8146 오류 메시지도 수신합니다.

그러나 **dspmq**는 큐 관리자를 표시할 수 없어 빈 응답을 대신 리턴합니다.

```
docker exec container dspmq
```

- 필요한 모든 네임스페이스를 공유하지만 공유 볼륨을 /var/mqm 디렉토리에 마운트하지 않았으며 유효한 IBM MQ 데이터 경로가 없는 경우(또는 IBM MQ 데이터 경로가 없음) 데이터 경로는 IBM MQ 설치의 핵심 컴포넌트이므로 다양한 오류가 표시됩니다. 데이터 경로가 없으면 IBM MQ를 작동할 수 없습니다.

다음 명령을 실행하고 이러한 예제에 표시되는 것과 유사한 응답을 참조하는 경우, 디렉토리를 마운트하거나 IBM MQ 데이터 디렉토리를 작성했는지 확인해야 합니다.

```
docker exec container dspmq
'No such file or directory' from /var/mqm/mqs.ini
AMQ6090: IBM MQ was unable to display an error message FFFFFFFF.
AMQffff

docker exec container dspmqver
AMQ7047: An unexpected error was encountered by a command. Reason code is 0.

docker exec container mqrc
<file path>/mqrc.c[1152]
lpi0btainQMDetails --> 545261715

docker exec container crtmqm QM1
AMQ8101: IBM MQ error (893) has occurred.

docker exec container strmqm QM1
AMQ6239: Permission denied attempting to access filesystem location '/var/mqm'.
AMQ7002: An error occurred manipulating a file.

docker exec container endmqm QM1
AMQ8101: IBM MQ error (893) has occurred.

docker exec container dlmqm QM1
AMQ7002: An error occurred manipulating a file.

docker exec container strmqweb
<file path>/mqrc.c[1152]
lpi0btainQMDetails --> 545261715
```

IBM MQ는 Kubernetes Operator를 제공하며, OpenShift 컨테이너 플랫폼으로 기본 통합을 제공합니다.

v1beta1 API를 사용하여 QueueManager 자원을 작성하고 관리할 수 있습니다.

## mq.ibm.com/v1beta1에 대한 라이선스 부여 참조

spec.license.license 필드에는 승인하는 라이선스에 대한 라이선스 ID가 있어야 합니다. 유효값은 다음과 같습니다.

다음의 값 spec.license.license	다음의 값 spec.license.use	라이선스 정보
L-RJON-BN7PN3	Production 또는 NonProduction	<a href="#">IBM Cloud Pak for Integration 2020.2</a>
L-RJON-BPHL2Y		<a href="#">IBM Cloud Pak for Integration Limited Edition 2020.2</a>
L-APIG-BJAKBF	Production 또는 Development	<a href="#">IBM MQ Advanced 9.1.5</a>
L-APIG-BM7GDH	Development	<a href="#">IBM MQ Advanced for Developers 9.1.5</a>

라이선스 버전이 지정되며, 항상 IBM MQ의 버전과 동일하지는 않습니다.

## QueueManager용 API 참조 (mq.ibm.com/v1beta1)

### QueueManager

QueueManager는 애플리케이션에 큐잉 및 발행/구독 서비스를 제공하는 IBM MQ 서버입니다.

필드	설명
apiVersion 문자열	APIVersion은 오브젝트의 이 표현의 버전화된 스키마를 정의합니다. 서버는 인식된 스키마를 최신 내부 값으로 변환해야 하며 인식되지 않는 값을 거부할 수 있습니다. 추가 정보: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a> .
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: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a> .
metadata	
spec 41 페이지의 『QueueManager스펙』	원하는 상태의 QueueManager입니다.
status 42 페이지의 『큐 관리자 상태』	QueueManager의 관찰된 상태입니다.

### 가용성

활성 대기 쌍을 사용할지 여부와 같은 큐 관리자의 가용성 설정입니다.

다음과 같이 표시됩니다.

- 39 페이지의 『QueueManager구성』

필드	설명
type 문자열	사용할 가용성 유형입니다. Kubernetes에 의해 자동으로 재시작되는 단일 Pod에 대해 "SingleInstance" 를 사용하십시오. 한 쌍의 Pods에 "MultiInstance" 를 사용하십시오. 이 중 하나는 "활성" 큐 관리자이고 다른 하나는 대기입니다. IBM MQ의 최신 버전에서 <a href="#">컨테이너의 IBM MQ에 대한 고가용성</a> 을 참조하십시오.

## 라이선스

라이선스의 승인을 제어하는 설정 및 사용할 라이선스 메트릭입니다.

다음과 같이 표시됩니다.

- 41 페이지의 『QueueManager스펙』

필드	설명
use 문자열	소프트웨어가 사용될 방법을 제어하는 설정이며, 여기서 라이선스는 다중 사용을 지원합니다. 유효한 값은 <a href="https://ibm.biz/BdqvCF">https://ibm.biz/BdqvCF</a> 을 참조하십시오.
accept boolean	이 소프트웨어와 연관된 라이선스에 동의하는지 여부입니다(필수).
license 문자열	승인 중인 라이선스의 ID입니다. 사용 중인 MQ의 버전으로 올바른 라이선스 ID여야 합니다. 유효한 값은 <a href="https://ibm.biz/BdqvCF">https://ibm.biz/BdqvCF</a> 을 참조하십시오.
metric 문자열	사용할 라이선스 메트릭을 지정하는 설정입니다. (예: "ProcessorValueUnit", "VirtualProcessorCore" 또는 "ManagedVirtualServer").

## 한계

QueueManagerResourceList 는 CPU 및 메모리 설정을 정의합니다.

다음과 같이 표시됩니다.

- 44 페이지의 『자원』

필드	설명
cpu	
memory	

## LocalObject참조

LocalObjectReference는 동일한 네임스페이스 내에서 참조된 오브젝트를 찾을 수 있는 충분한 정보가 포함됩니다.

다음과 같이 표시됩니다.

- 41 페이지의 『QueueManager스펙』

필드	설명
name 문자열	참조 이름입니다. 추가 정보: <a href="https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names">https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names</a> TODO: 다른 유용한 필드를 추가합니다. apiVersion, kind, uid?.

## PKI

TLS(Transport Layer Security) 또는 AMS(Advanced Message Security)에서 사용할 키와 인증서를 정의하기 위한 공개 키 인프라 설정

다음과 같이 표시됩니다.

- [41 페이지의 『QueueManager스펙』](#)

필드	설명
<a href="#">keys 39 페이지의 『PKISource』 배열</a>	큐 관리자의 키 저장소에 추가할 개인 키입니다.
<a href="#">trust 39 페이지의 『PKISource』 배열</a>	큐 관리자의 키 저장소에 추가할 인증서입니다.

## PKISource

PKISource는 키 또는 인증서와 같은 공개 키 인프라 정보의 소스를 정의합니다.

다음과 같이 표시됩니다.

- [39 페이지의 『PKI』](#)

필드	설명
name 문자열	이름은 키 또는 인증서의 레이블로 사용됩니다. 소문자 영숫자 문자열이어야 합니다.
<a href="#">secret 45 페이지의 『시크릿』</a>	Kubernetes 본인확인정보를 사용하여 키를 제공합니다.

## QueueManager구성

QueueManager구성은 큐 관리자 컨테이너 및 기본 큐 관리자에 대한 설정을 정의합니다.

다음과 같이 표시됩니다.

- [41 페이지의 『QueueManager스펙』](#)

필드	설명
logFormat 문자열	이 컨테이너에 사용할 로그 형식입니다. 컨테이너에서 JSON 형식 로그에 대해 "JSON" 을 사용한다. 텍스트 형식 메시지에는 "기본" 을 사용하십시오.
<a href="#">metrics 40 페이지의 『QueueManager메트릭』</a>	Prometheus-스타일 메트릭의 설정입니다.
<a href="#">readinessProbe 41 페이지의 『QueueManagerReadinessProbe』</a>	준비 프로브를 제어하는 설정입니다.
<a href="#">resources 44 페이지의 『자원』</a>	자원 요구사항을 제어하는 설정입니다.
<a href="#">storage 43 페이지의 『QueueManager스토리지』</a>	큐 관리자의 지속적 볼륨 및 스토리지 클래스 사용을 제어하는 스토리지 설정입니다.
<a href="#">availability 37 페이지의 『가용성』</a>	활성 대기 쌍을 사용할지 여부와 같은 큐 관리자의 가용성 설정입니다.
imagePullPolicy 문자열	kubelet이 지정된 이미지를 풀하려고 시도할 때 제어하는 설정입니다.

필드	설명
<a href="#">livenessProbe</a> 40 페이지의 『 <a href="#">QueueManagerLivenessProbe</a> 』	라이브 프로브를 제어하는 설정입니다.
debug boolean	컨테이너 특정 코드에서 컨테이너 로그로 디버그 메시지를 로그할 지 여부입니다.
image 문자열	사용할 컨테이너 이미지입니다.
name 문자열	metadata.name과 다른 경우 기본 MQ Queue Manager의 이름입니다. 이름으로 Kubernetes 규칙을 준수하지 않는 Queue Manager 이름을 원하는 경우 이 필드를 사용하십시오(예를 들어, 대문자를 포함한 이름).

### QueueManagerLivenessProbe

라이브 프로브를 제어하는 설정입니다.

다음과 같이 표시됩니다.

- 39 페이지의 『[QueueManager구성](#)』

필드	설명
failureThreshold 정수	성공 후 프로브가 실패한 것으로 간주되는 최소 연속 실패입니다.
initialDelaySeconds 정수	라이브 프로브가 초기화되기 전에 컨테이너가 시작된 후의 시간(초)입니다. 추가 정보: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes</a> .
periodSeconds 정수	프로브를 수행하는 빈도(초)입니다.
successThreshold 정수	실패 후 성공한 것으로 간주되는 프로브의 최소 연속 성공입니다.
timeoutSeconds 정수	프로브가 제한시간을 초과한 시간(초)입니다. 추가 정보: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes</a> .

### QueueManager메트릭

Prometheus-스타일 메트릭의 설정입니다.

다음과 같이 표시됩니다.

- 39 페이지의 『[QueueManager구성](#)』

필드	설명
enabled boolean	프로메테우스 호환 가능 메트릭 엔드포인트를 사용할지 여부입니다.

### QueueManagerOptionalVolume

MQ 복구 로그에 대한 PersistentVolume 세부사항입니다. 다중 인스턴스 큐 관리자를 사용할 때 필요합니다.

다음과 같이 표시됩니다.

- 43 페이지의 『[QueueManager스토리지](#)』

필드	설명
class 문자열	이 볼륨에 사용할 스토리지 클래스입니다. "유형" 이 "지속적 청구" 인 경우에만 유효합니다.



필드	설명
enabled boolean	이 볼륨을 별도의 볼륨으로 사용하거나 기본 "queueManager" 볼륨에 배치해야 하는지 여부입니다.
size 문자열	Kubernetes에 전달할 PersistentVolume 의 크기입니다. "유형" 이 "지속적 청구" 인 경우에만 유효합니다.
sizeLimit 문자열	"임시" 볼륨을 사용하는 경우 크기 제한입니다. 파일은 여전히 임시 디렉토리에 기록되므로 이 옵션을 사용하여 크기를 제한할 수 있습니다. type 이 "ephemeral" 인 경우에만 유효합니다.
type 문자열	사용할 볼륨의 유형입니다. 비지속적 "emptyDir" 볼륨을 작성하려면 ephemeral 를 선택하고 지속적 볼륨을 사용하려면 persistent-claim 를 선택하십시오.

## QueueManagerReadinessProbe

준비 프로브를 제어하는 설정입니다.

다음과 같이 표시됩니다.

- 39 페이지의 『QueueManager구성』

필드	설명
failureThreshold 정수	성공 후 프로브가 실패한 것으로 간주되는 최소 연속 실패입니다.
initialDelaySeconds 정수	라이브 프로브가 초기화되기 전에 컨테이너가 시작된 후의 시간(초)입니다. 추가 정보: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes</a> .
periodSeconds 정수	프로브를 수행하는 빈도(초)입니다.
successThreshold 정수	실패 후 성공한 것으로 간주되는 프로브의 최소 연속 성공입니다.
timeoutSeconds 정수	프로브가 제한시간을 초과한 시간(초)입니다. 추가 정보: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes</a> .

## QueueManager스펙

원하는 상태의 QueueManager입니다.

다음과 같이 표시됩니다.

- 37 페이지의 『QueueManager』

필드	설명
license 38 페이지의 『라이센스』	라이센스의 승인을 제어하는 설정 및 사용할 라이선스 메트릭입니다.
pki 39 페이지의 『PKI』	TLS(Transport Layer Security) 또는 AMS(Advanced Message Security)에서 사용할 키와 인증서를 정의하기 위한 공개 키 인프라 설정
queueManager 39 페이지의 『QueueManager구성』	QueueManager구성은 큐 관리자 컨테이너 및 기본 큐 관리자에 대한 설정을 정의합니다.
securityContext 45 페이지의 『SecurityContext』	큐 관리자 팟(Pod)의 securityContext에 추가할 보안 설정입니다.
tracing 46 페이지의 『TracingConfig』	통합 운영 대시보드에 대한 클라우드 팩과의 통합을 위한 설정입니다.

필드	설명
version 문자열	사용할 MQ의 버전을 제어하는 설정입니다(필수). 예를 들어, "9.1.5.0-r2" 는 컨테이너 이미지의 두 번째 개정판을 사용하여 MQ 버전 9.1.5.0을 지정합니다. 컨테이너 특정 수정사항은 종종 기본 이미지에 대한 수정사항과 같은 개정판에 적용됩니다.
affinity	표준 Kubernetes 연관관계 규칙입니다. 자세한 정보는 <a href="https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#affinity-v1-core">https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#affinity-v1-core</a> 를 참조하십시오.
imagePullSecrets 38 페이지의 『LocalObject참조』 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 <a href="https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod">https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod</a> 를 참조하십시오.
template 45 페이지의 『템플릿』	Kubernetes 자원을 위한 고급 템플릿입니다. 템플릿을 사용하면 사용자가 IBM MQ가 StatefulSet, Pods 및 Services와 같이 기본 Kubernetes 자원을 생성하는 방법을 오버라이드할 수 있습니다. 이는 고급 사용자만을 위한 것이며 잘못 사용된 경우 MQ의 정상 작동을 방해할 가능성이 있기 때문입니다. QueueManager 자원의 다른 위치에 지정된 모든 값은 템플릿의 설정으로 대체됩니다.
terminationGracePeriod Seconds 정수	팟(Pod)가 적절하게 종료되는 선택적 기간(초)입니다. 값은 음이 아닌 정수여야 합니다. 0값은 즉시 삭제를 표시합니다. 큐 관리자 종료 시도가 시도되는 대상 시간이며, 애플리케이션 연결 중단의 단계를 에스컬레이션합니다. 필요한 경우 필수 큐 관리자 유지보수 태스크가 인터럽트됩니다.
web 47 페이지의 『WebServer구성』	MQ 웹 서버를 위한 설정입니다.

## 큐 관리자 상태

QueueManager의 관찰된 상태입니다.

다음과 같이 표시됩니다.

- 37 페이지의 『QueueManager』

필드	설명
endpoints 43 페이지의 『QueueManagerStatusEndpoint』 배열	API 또는 UI 엔드 포인트와 같이 이 Queue Manager가 노출하는 엔드포인트에 대한 정보입니다.
name 문자열	대기열 관리자의 이름입니다.
versions 43 페이지의 『QueueManagerStatusVersion』	사용 중인 MQ의 버전 및 IBM Entitled Registry에서 사용 가능한 기타 버전입니다.
adminUiUrl 문자열	Admin UI의 URL입니다.
conditions 42 페이지의 『QueueManagerStatusCondition』 배열	조건은 큐 관리자 상태의 사용 가능한 최신 관찰을 나타냅니다.

## QueueManagerStatusCondition

QueueManagerStatusCondition은 Queue Manager의 조건을 정의합니다.

다음과 같이 표시됩니다.

- [42 페이지의 『큐 관리자 상태』](#)

필드	설명
message 문자열	마지막 전이에 대한 세부사항을 표시하는 사용자가 읽을 수 있는 메시지입니다.
type 문자열	조건의 유형입니다.
lastTransitionTime 문자열	상태가 한 상태에서 다른 상태로 전이된 마지막 시간입니다.

### QueueManagerStatusEndpoint

QueueManagerStatusEndpoint는 QueueManager에 대한 엔드포인트를 정의합니다.

다음과 같이 표시됩니다.

- [42 페이지의 『큐 관리자 상태』](#)

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.

### QueueManagerStatusVersion

사용 중인 MQ의 버전 및 IBM Entitled Registry에서 사용 가능한 기타 버전입니다.

다음과 같이 표시됩니다.

- [42 페이지의 『큐 관리자 상태』](#)

필드	설명
available <a href="#">43 페이지의 『QueueManagerStatusVersion사용 가능』</a>	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.

### QueueManagerStatusVersion사용 가능

IBM Entitled Registry에서 사용 가능한 MQ의 다른 버전입니다.

다음과 같이 표시됩니다.

- [43 페이지의 『QueueManagerStatusVersion』](#)

필드	설명
channels 배열	MQ 버전을 자동으로 업데이트하는 데 사용할 수 있는 채널입니다.
versions <a href="#">47 페이지의 『버전』</a> 배열	사용할 수 있는 MQ의 특정 버전입니다.

### QueueManager스토리지

큐 관리자의 지속적 볼륨 및 스토리지 클래스 사용을 제어하는 스토리지 설정입니다.

다음과 같이 표시됩니다.

- 39 페이지의 『QueueManager구성』

필드	설명
<a href="#">persistedData 40 페이지의 『QueueManagerOptionalVolume』</a>	구성, 큐 및 메시지를 포함하는 MQ 지속 데이터에 대한 PersistentVolume 세부 사항입니다. 다중 인스턴스 큐 관리자를 사용할 때 필요합니다.
<a href="#">queueManager 44 페이지의 『QueueManager볼륨』</a>	일반적으로 /var/mqm 아래 데이터에 대한 기본 PersistentVolume입니다. 다른 볼륨이 지정되지 않은 경우 모든 지속 데이터 및 복구 로그를 포함합니다.
<a href="#">recoveryLogs 40 페이지의 『QueueManagerOptionalVolume』</a>	MQ 복구 로그에 대한 PersistentVolume 세부사항입니다. 다중 인스턴스 큐 관리자를 사용할 때 필요합니다.

### QueueManager볼륨

일반적으로 /var/mqm 아래 데이터에 대한 기본 PersistentVolume입니다. 다른 볼륨이 지정되지 않은 경우 모든 지속 데이터 및 복구 로그를 포함합니다.

다음과 같이 표시됩니다.

- 43 페이지의 『QueueManager스토리지』

필드	설명
class 문자열	이 볼륨에 사용할 스토리지 클래스입니다. "유형" 이 "지속적 청구" 인 경우에만 유효합니다.
size 문자열	Kubernetes에 전달할 PersistentVolume 의 크기입니다. "유형" 이 "지속적 청구" 인 경우에만 유효합니다.
sizeLimit 문자열	"임시" 볼륨을 사용하는 경우 크기 제한입니다. 파일은 여전히 임시 디렉토리에 기록되므로 이 옵션을 사용하여 크기를 제한할 수 있습니다. type 이 "ephemeral" 인 경우에만 유효합니다.
type 문자열	사용할 볼륨의 유형입니다. 비지속적 "emptyDir" 볼륨을 작성하려면 ephemeral 를 선택하고 지속적 볼륨을 사용하려면 persistent-claim 를 선택하십시오.

### 요청 수

QueueManagerResourceList 는 CPU 및 메모리 설정을 정의합니다.

다음과 같이 표시됩니다.

- 44 페이지의 『자원』

필드	설명
memory	
cpu	

### 자원

자원 요구사항을 제어하는 설정입니다.

다음과 같이 표시됩니다.

- 39 페이지의 『QueueManager구성』

필드	설명
limits 38 페이지의 『한계』	QueueManagerResourceList 는 CPU 및 메모리 설정을 정의합니다.
requests 44 페이지의 『요청 수』	QueueManagerResourceList 는 CPU 및 메모리 설정을 정의합니다.

## 시크릿

Kubernetes 본인확인정보를 사용하여 키를 제공합니다.

다음과 같이 표시됩니다.

- 39 페이지의 『PKISource』

필드	설명
items 배열	큐 관리자 컨테이너에 추가해야 하는 Kubernetes 본인확인정보 내 키입니다.
secretName 문자열	Kubernetes 본인확인정보의 이름입니다.

## SecurityContext

큐 관리자 팟(Pod)의 securityContext에 추가할 보안 설정입니다.

다음과 같이 표시됩니다.

- 41 페이지의 『QueueManager스펙』

필드	설명
supplementalGroups 배열	컨테이너의 기본 GID 외에 각 컨테이너에서 실행되는 첫 번째 프로세스에 적용된 그룹 목록입니다. 지정하지 않으면 어떤 그룹도 컨테이너에 추가되지 않습니다.
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 필요한 스토리지 제공자를 사용하는 경우 이를 "참" 으로 설정하십시오. 이것을 "true" 로 설정하면 사용할 수 있는 SCC (Security Context 제한조건) 오브젝트에 영향을 주고, 루트 사용자를 허용하는 SCC를 사용할 권한이 없는 경우 큐 관리자를 시작하는 데 실패할 수 있다. 자세한 정보는 <a href="https://docs.openshift.com/container-platform/latest/authentication/managing-security-context-constraints.html">https://docs.openshift.com/container-platform/latest/authentication/managing-security-context-constraints.html</a> 의 내용을 참조하십시오.

## 템플릿

Kubernetes 자원을 위한 고급 템플릿입니다. 템플릿을 사용하면 사용자가 IBM MQ가 StatefulSet, Pods 및 Services와 같이 기본 Kubernetes 자원을 생성하는 방법을 오버라이드할 수 있습니다. 이는 고급 사용자만을 위한 것이며 잘못 사용된 경우 MQ의 정상 작동을 방해할 가능성이 있기 때문입니다. QueueManager 자원의 다른 위치에 지정된 모든 값은 템플릿의 설정으로 대체됩니다.

다음과 같이 표시됩니다.

- 41 페이지의 『QueueManager스펙』

필드	설명
pod	팟(Pod)에 사용된 템플릿을 대체합니다. <a href="https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#podspec-v1-core">https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#podspec-v1-core</a> 을 참조하십시오.

## TracingAgent

통합용 클라우드 팩에서는 선택적 추적 에이전트에 대한 설정을 구성할 수 있습니다.

다음과 같이 표시됩니다.

- [46 페이지의 『TracingConfig』](#)

필드	설명
image 문자열	사용할 컨테이너 이미지입니다.
imagePullPolicy 문자열	kubelet이 지정된 이미지를 풀하려고 시도할 때 제어하는 설정입니다.
<a href="#">livenessProbe 47 페이지의 『TracingProbe』</a>	라이브 프로브를 제어하는 설정입니다.
<a href="#">readinessProbe 47 페이지의 『TracingProbe』</a>	준비 프로브를 제어하는 설정입니다.

## TracingCollector

통합용 클라우드 팩에서는 선택적 추적 콜렉터에 대한 설정을 구성할 수 있습니다.

다음과 같이 표시됩니다.

- [46 페이지의 『TracingConfig』](#)

필드	설명
image 문자열	사용할 컨테이너 이미지입니다.
imagePullPolicy 문자열	kubelet이 지정된 이미지를 풀하려고 시도할 때 제어하는 설정입니다.
<a href="#">livenessProbe 47 페이지의 『TracingProbe』</a>	라이브 프로브를 제어하는 설정입니다.
<a href="#">readinessProbe 47 페이지의 『TracingProbe』</a>	준비 프로브를 제어하는 설정입니다.

## TracingConfig

통합 운영 대시보드에 대한 클라우드 팩과의 통합을 위한 설정입니다.

다음과 같이 표시됩니다.

- [41 페이지의 『QueueManager스펙』](#)

필드	설명
<a href="#">agent 46 페이지의 『TracingAgent』</a>	통합용 클라우드 팩에서는 선택적 추적 에이전트에 대한 설정을 구성할 수 있습니다.
<a href="#">collector 46 페이지의 『TracingCollector』</a>	통합용 클라우드 팩에서는 선택적 추적 콜렉터에 대한 설정을 구성할 수 있습니다.
enabled boolean	추적을 통해 통합 운영 대시보드의 클라우드 팩과의 통합을 사용할지 여부입니다.
namespace 문자열	통합 운영 대시보드용 클라우드 팩이 설치된 네임스페이스입니다.

## TracingProbe

준비 프로브를 제어하는 설정입니다.

다음과 같이 표시됩니다.

- 46 페이지의 『TracingCollector』

필드	설명
failureThreshold 정수	성공 후 프로브가 실패한 것으로 간주되는 최소 연속 실패입니다.
initialDelaySeconds 정수	라이브 프로브가 초기화되기 전에 컨테이너가 시작된 후의 시간(초)입니다. 추가 정보: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes</a> .
periodSeconds 정수	프로브를 수행하는 빈도(초)입니다.
successThreshold 정수	실패 후 성공한 것으로 간주되는 프로브의 최소 연속 성공입니다.
timeoutSeconds 정수	프로브가 제한시간을 초과한 시간(초)입니다. 추가 정보: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes</a> .

## 버전

QueueManagerStatusVersion은 MQ의 버전을 정의합니다.

다음과 같이 표시됩니다.

- 43 페이지의 『QueueManagerStatusVersion사용 가능』

필드	설명
name 문자열	이 버전의 QueueManager에 대한 버전 "이름" spec.version 필드에 올바른 값입니다.

## WebServer구성

MQ 웹 서버를 위한 설정입니다.

다음과 같이 표시됩니다.

- 41 페이지의 『QueueManager스펙』

필드	설명
enabled boolean	웹 서버 사용 가능 여부입니다.





## 주의사항

이 정보는 미국에서 제공되는 제품 및 서비스용으로 작성된 것입니다.

IBM은 다른 국가에서 이 책에 기술된 제품, 서비스 또는 기능을 제공하지 않을 수도 있습니다. 현재 사용할 수 있는 제품 및 서비스에 대한 정보는 한국 IBM 담당자에게 문의하십시오. 이 책에서 IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산을 침해하지 않는 한, 기능상으로 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수도 있습니다. 그러나 비IBM 제품, 프로그램 또는 서비스의 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대해 특허를 보유하고 있거나 현재 특허 출원 중일 수 있습니다. 이 책을 제공한다고 해서 특허에 대한 라이선스까지 부여하는 것은 아닙니다. 라이선스에 대한 의문사항은 다음으로 문의하십시오.

150-945  
서울특별시 영등포구  
국제금융로 10, 3IFC  
한국 아이.비.엠 주식회사  
U.S.A.

2바이트(DBCS) 정보에 관한 라이선스 문의는 한국 IBM에 문의하거나 다음 주소로 서면 문의하시기 바랍니다.

지적 재산권 라이선스 부여  
2-31 Roppongi 3-chome, Minato-Ku  
IBM Japan, Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japan

**다음 단락은 현지법과 상충하는 영국이나 기타 국가에서는 적용되지 않습니다.** IBM은 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여(단, 이에 한하지 않음) 명시적 또는 묵시적인 일체의 보증 없이 이 책을 "현상태대로" 제공합니다. 일부 국가에서는 특정 거래에서 명시적 또는 묵시적 보증의 면책사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 정보는 주기적으로 변경되며, 변경된 사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및/또는 프로그램을 사전 통지 없이 언제든지 개선 및/또는 변경할 수 있습니다.

이 정보에서 언급되는 비IBM의 웹 사이트는 단지 편의상 제공된 것으로, 어떤 방식으로든 이들 웹 사이트를 옹호하고자 하는 것은 아닙니다. 해당 웹 사이트의 자료는 본 IBM 제품 자료의 일부가 아니므로 해당 웹 사이트 사용으로 인한 위험은 사용자 본인이 감수해야 합니다.

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

(i) 독립적으로 작성된 프로그램과 기타 프로그램(본 프로그램 포함) 간의 정보 교환 및 (ii) 교환된 정보의 상호 이용을 목적으로 본 프로그램에 관한 정보를 얻고자 하는 라이선스 사용자는 다음 주소로 문의하십시오.

서울특별시 영등포구  
서울특별시 강남구 도곡동 467-12,  
군인공제회관빌딩  
한국 아이.비.엠 주식회사  
U.S.A.

이러한 정보는 해당 조건(예를 들면, 사용료 지불 등)하에서 사용될 수 있습니다.

이 정보에 기술된 라이선스가 부여된 프로그램 및 프로그램에 대해 사용 가능한 모든 라이선스가 부여된 자료는 IBM이 IBM 기본 계약, IBM 프로그램 라이선스 계약(IPLA) 또는 이와 동등한 계약에 따라 제공한 것입니다.

본 문서에 포함된 모든 성능 데이터는 제한된 환경에서 산출된 것입니다. 따라서 다른 운영 환경에서 얻어진 결과는 상당히 다를 수 있습니다. 일부 성능은 개발 단계의 시스템에서 측정되었을 수 있으므로 이러한 측정치가 일반적으로 사용되고 있는 시스템에서도 동일하게 나타날 것이라고는 보증할 수 없습니다. 또한 일부 성능은 추정

통해 추측되었을 수도 있으므로 실제 결과는 다를 수 있습니다. 이 책의 사용자는 해당 데이터를 본인의 특정 환경에서 검증해야 합니다.

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개 자료 또는 기타 범용 소스로부터 얻은 것입니다. IBM에서는 이러한 제품들을 테스트하지 않았으므로, 비IBM 제품과 관련된 성능의 정확성, 호환성 또는 기타 청구에 대해서는 확신할 수 없습니다. 비IBM 제품의 성능에 대한 의문사항은 해당 제품의 공급업체에 문의하십시오.

IBM이 제시하는 방향 또는 의도에 관한 모든 언급은 특별한 통지 없이 변경될 수 있습니다.

이 정보에는 일상의 비즈니스 운영에서 사용되는 자료 및 보고서에 대한 예제가 들어 있습니다. 이들 예제에는 개념을 가능한 완벽하게 설명하기 위하여 개인, 회사, 상표 및 제품의 이름이 사용될 수 있습니다. 이들 이름은 모두 가공의 것이며 실제 기업의 이름 및 주소와 유사하더라도 이는 전적으로 우연입니다.

저작권 라이선스:

이 정보에는 여러 운영 플랫폼에서의 프로그래밍 기법을 보여주는 원어로 된 샘플 응용프로그램이 들어 있습니다. 귀하는 이러한 샘플 프로그램의 작성 기준이 된 운영 플랫폼의 응용프로그램 프로그래밍 인터페이스(API)에 부합하는 응용프로그램을 개발, 사용, 판매 또는 배포할 목적으로 IBM에 추가 비용을 지불하지 않고 이들 샘플 프로그램을 어떠한 형태로든 복사, 수정 및 배포할 수 있습니다. 이러한 샘플 프로그램은 모든 조건하에서 완전히 테스트된 것은 아닙니다. 따라서 IBM은 이들 샘플 프로그램의 신뢰성, 서비스 가능성 또는 기능을 보증하거나 진술하지 않습니다.

이 정보를 소프트웨어로 확인하는 경우에는 사진과 컬러 삽화가 제대로 나타나지 않을 수도 있습니다.

## 프로그래밍 인터페이스 정보

---

프로그래밍 인터페이스 정보는 본 프로그램과 함께 사용하기 위한 응용프로그램 소프트웨어 작성을 돕기 위해 제공됩니다.

이 책에는 고객이 프로그램을 작성하여 WebSphere MQ서비스를 얻을 수 있도록 하는 계획된 프로그래밍 인터페이스에 대한 정보가 포함되어 있습니다.

그러나 본 정보에는 진단, 수정 및 성능 조정 정보도 포함되어 있습니다. 진단, 수정 및 성능 조정 정보는 응용프로그램 소프트웨어의 디버거를 돕기 위해 제공된 것입니다.

**중요사항:** 이 진단, 수정 및 튜닝 정보는 변경될 수 있으므로 프로그래밍 인터페이스로 사용하지 마십시오.

## 상표

---

IBM, IBM 로고, [ibm.com](http://www.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](http://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 (<http://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: