

9.1

儲存器中的 *IBM MQ*

**IBM**

## 附註

使用本資訊及其支援的產品之前，請先閱讀第 47 頁的『[注意事項](#)』中的資訊。

除非新版中另有指示，否則此版本適用於 IBM® MQ 9.1.1 版及所有後續版本與修訂版。

當您將資訊傳送至 IBM 時，您授與 IBM 非專屬權利，以任何其認為適當的方式使用或散佈資訊，而無需對您負責。

© Copyright International Business Machines Corporation 2007, 2024.

# 目錄

<b>儲存器中的 IBM MQ.....</b>	<b>5</b>
規劃儲存器中的 IBM MQ.....	5
選擇您要如何在儲存器中使用 IBM MQ.....	5
支援 IBM MQ 認證儲存器.....	6
支援建置您自己的 IBM MQ 儲存器映像檔及圖表.....	8
IBM MQ Advanced certified container 的儲存體考量.....	8
IBM MQ Advanced certified container 的高可用性.....	9
IBM MQ Advanced certified container 的使用者鑑別和授權.....	11
在 OpenShift 上安裝及解除安裝 IBM MQ 操作器.....	11
使用 OpenShift Web 主控台安裝 IBM MQ 操作器.....	11
使用 OpenShift CLI 安裝 IBM MQ 操作器.....	12
部署 IBM MQ 認證的儲存器.....	14
使用 OpenShift CLI 為 IBM MQ 準備 OpenShift 專案.....	15
使用 IBM Cloud Pak for Integration Platform Navigator 部署佇列管理程式.....	15
使用 OpenShift Web 主控台來部署佇列管理程式.....	16
使用 OpenShift CLI 部署佇列管理程式.....	17
與 IBM Cloud Pak for Integration 作業儀表板整合.....	18
使用 OpenShift CLI 來建置具有自訂 MQSC 和 INI 檔的映像檔.....	19
使用 Helm 部署 IBM MQ 認證儲存器.....	21
將 IBM MQ 的舊版 CD 部署至 IBM Cloud Private 叢集.....	24
將舊版 CD 的 IBM MQ 映像檔新增至 IBM Cloud Private 叢集.....	25
將舊版 CD 的 IBM MQ 映像檔新增至 IBM Cloud Kubernetes Service 叢集.....	26
連接至部署在 OpenShift 叢集裡的佇列管理程式.....	27
連接至部署在 OpenShift 叢集裡的 IBM MQ Console.....	28
使用 OpenShift CLI 備份及還原佇列管理程式配置.....	29
建置您自己的 IBM MQ 儲存器.....	30
使用儲存器規劃您自己的 IBM MQ 佇列管理程式映像檔.....	30
使用 Docker 建置範例 IBM MQ 佇列管理程式映像檔.....	31
在個別儲存器中執行本端連結應用程式.....	33
IBM MQ 運算子的 API 參考資料.....	35
mq.ibm.com/v1beta1 的 API 參考資料.....	35
<b>注意事項.....</b>	<b>47</b>
程式設計介面資訊.....	48
商標.....	48



儲存器可讓您將 IBM MQ 佇列管理程式或 IBM MQ 用戶端應用程式及其所有相依關係包裝成標準化單元，以進行軟體開發。

您可以在 IBM MQ Advanced 及 IBM MQ Advanced for Developers 中提供的預先包裝容器中執行 IBM MQ。此 IBM MQ Advanced certified container 提供支援的映像檔及 Helm 圖表，可用來將可正式作業的 IBM MQ 映像檔部署至 Red Hat® OpenShift®、IBM Cloud Private 或 IBM Cloud Kubernetes Service。

您也可以使用 IBM Cloud Pak for Integration 容器中執行 IBM MQ，或在您自行建置的容器中執行。

MQ Adv.

CD

如需 IBM MQ Advanced certified container 的相關資訊，請參閱下列鏈結。

在儲存器中規劃 IBM MQ 時，請考量 IBM MQ 提供的各種架構選項支援，例如如何管理高可用性，以及如何保護佇列管理程式的安全。

### 關於這項作業

在儲存器架構中規劃 IBM MQ 之前，您應該先熟悉基本 IBM MQ 概念 (請參閱 [IBM MQ 技術概觀](#)) 以及基本 Kubernetes/OpenShift 概念 (請參閱 [OpenShift Container Platform 架構](#))。

### 程序

- 第 5 頁的『選擇您要如何在儲存器中使用 IBM MQ』。
- 第 9 頁的『IBM MQ Advanced certified container 的高可用性』。
- 第 11 頁的『IBM MQ Advanced certified container 的使用者鑑別和授權』。

在容器中使用 IBM MQ 有多個選項：您可以選擇使用預先包裝的認證容器，也可以建置您自己的映像檔及部署程式碼。

### 使用 IBM MQ Advanced 認證儲存器

如果您計劃在 Red Hat OpenShift Container Platform 上部署，則可能想要使用已認證的儲存器。經認證的容器有三種：

- IBM MQ Advanced certified container for IBM Cloud Pak for Integration。這是個別 IBM 產品，其中包含已認證儲存器的版本。
- IBM MQ Advanced certified container
- IBM MQ Advanced for Developers 認證儲存器 (無保證)

IBM Cloud Private 及 IBM Cloud Kubernetes Service 也支援 IBM MQ 9.1.4 及更早版本 CD。

請注意，已認證容器正在快速發展，因此只有在 [Continuous Delivery](#) 版本下才支援。

認證的容器包括預先建置的容器映像檔，以及在 Red Hat OpenShift Container Platform 上執行的部署程式碼。從 IBM MQ 9.1.5 開始，佇列管理程式是使用「IBM MQ 操作員」來管理。舊版 IBM MQ (包括 9.1.5 版) 是使用 Helm 圖表進行管理。

使用已認證的儲存器時，不支援部分 IBM MQ 特性。如果您想要執行下列任何動作，則需要建置您自己的映像檔及圖表：

- 使用 REST API 進行管理或傳訊
- 使用下列任何 MQ 元件：

- Managed File Transfer 代理程式及其資源。不過，您可以使用認證的儲存器來提供一或多個「協調」、「指令」或「代理程式」佇列管理程式。
- AMQP
- IBM MQ Bridge to Salesforce
- IBM MQ Bridge to blockchain (在儲存器中不受支援)
- 使用 Helm 圖表部署時使用 Web 伺服器 (IBM Cloud Pak for Integration 除外)
- 自訂與 `crtmqm`、`strmqm` 及 `endmqm` 搭配使用的選項，例如配置回復日誌

## 建置您自己的影像和圖表

這是最具彈性的容器解決方案，但它需要您具備配置容器的強大技能，以及 "擁有" 產生的容器。如果您不打算使用 Red Hat OpenShift Container Platform，則需要建置您自己的映像檔及部署程式碼。

提供用於建置您自己的映像檔的範例。請參閱第 30 頁的『建置您自己的 IBM MQ 儲存器』。作為已認證儲存器的一部分提供的 Helm 圖表發佈在 GitHub 上，並且可以在您建置自己的映像檔時用作範例：

- [IBM MQ Advanced certified container 的 Helm 圖表](#)
- [IBM MQ Advanced for Developers 認證儲存器的 Helm Chart](#)

### 相關概念

第 6 頁的『支援 IBM MQ 認證儲存器』

IBM MQ 認證儲存器僅在特定 Kubernetes 環境中受支援

第 8 頁的『支援建置您自己的 IBM MQ 儲存器映像檔及圖表』

在 Linux 系統上使用儲存器時要考量的資訊。

## Linux 支援 IBM MQ 認證儲存器

IBM MQ 認證儲存器僅在特定 Kubernetes 環境中受支援

**V 9.1.4** **MQ Adv.** **CD** 若為 CD 版次 V9.1.4 以及更新版本，支援 IBM MQ Advanced certified container 與 Red Hat OpenShift 搭配使用。請參閱第 22 頁的『使用 Helm CLI 部署佇列管理程式』。

下列 Kubernetes 環境支援 V9.1.4 之前的 CD 版本：

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

如需 Kubernetes 的特定受支援版本，請參閱所下載 IBM MQ Advanced Helm Chart 內的檔案 `qualification.yaml` 及 `Chart.yaml`。這些版本會因不同版本而不同。

只有在使用 IBM MQ Operator 部署時，或使用下列其中一個 Helm 圖表時，才支援 IBM MQ Advanced certified container：

- `ibm-mqadvanced-server-prod`
- `ibm-mqadvanced-server-integration-prod` 在 IBM Cloud Pak for Integration

註：在 IBM MQ Operator 版本之後，已淘汰使用 Helm Chart。

因為儲存器技術正在快速發展，所以只有在此圖表發行時所支援平台的最新版本上才支援 IBM MQ Advanced certified container。如果您想要使用舊版平台，則可能需要使用舊版 IBM MQ Advanced certified container。

IBM MQ Advanced certified container 映像檔是以 IBM MQ Continuous Delivery (CD) 版本為基礎。這些支援最多一年，或兩個 CD 版次 (以較長者為準)。IBM MQ 的 Long Term Support 版次無法作為已認證儲存器使用。

從 IBM MQ Advanced certified container V4.0 開始，映像檔提供 Red Hat Universal Base Image (UBI) 上的 IBM MQ 安裝，其中包括 IBM MQ 使用的重要 Linux 程式庫及公用程式。在 Red Hat Enterprise Linux 主機

上執行時，Red Hat 支援 UBI。舊版 IBM MQ Advanced certified container 使用不受支援的 Ubuntu 基本映像檔。

## 相關概念

第 8 頁的『支援建置您自己的 IBM MQ 儲存器映像檔及圖表』  
在 Linux 系統上使用儲存器時要考量的資訊。

## Linux > MQ Adv. > CD IBM MQ Advanced certified container 的版本支援

一組表格，顯示 IBM MQ Advanced certified container、IBM MQ、IBM Cloud Kubernetes Service、IBM Cloud Pak for Integration 及 IBM Cloud Private 受支援版本之間的對映。

## 操作員 IBM MQ

V 9.1.5

支援 IBM MQ 操作器作為 IBM Cloud Pak for Integration 2020.2 版的一部分，或單獨與 IBM MQ 9.1.5 版及更高版本搭配使用。

Red Hat OpenShift Container Platform 4.4 版或更新版本支援 IBM MQ 操作器。

## IBM MQ Advanced certified container V 9.1.5 (Helm Chart)-已淘汰

包括 Helm 圖表 `ibm-mqadvanced-server-prod`。

V 9.1.5 從 IBM MQ Advanced certified container V5.0.x 開始，Helm 圖表、映像檔及修正程式透過 IBM Entitled Catalog and Registry 提供。舊版是透過 Passport Advantage 提供，而修正程式版本可從 IBM Fix Central 取得。

版本	IBM MQ 版本	結束支援	支援的平台
6.0.x	9.1.5 Continuous Delivery 版本	2021 年 3 月	<a href="#">詳細系統需求</a>
5.0.x	9.1.4 Continuous Delivery 版本	2020 年 12 月	<a href="#">詳細系統需求</a>
4.1.x	9.1.3 Continuous Delivery 版本	2020 年 7 月	<a href="#">詳細系統需求</a>

## IBM MQ Advanced certified container IBM Cloud Pak for Integration V 9.1.5 (Helm 圖表)-已淘汰的軟體

包括 Helm 圖表 `ibm-mqadvanced-server-integration-prod`。

版本	IBM MQ 版本	IBM Cloud Pak for Integration 版本
6.0.x	9.1.4 Continuous Delivery 版本	2020.1.1 (系統需求)
5.0.x	9.1.3 Continuous Delivery 版本	2019.4.1 (系統需求)
4.1.x	9.1.3 Continuous Delivery 版本	2019.3.2.2 (系統需求)

表 2: IBM Cloud Pak for Integration 的 IBM MQ Advanced certified container 軟體版本支援 (繼續)

版本	IBM MQ 版本	IBM Cloud Pak for Integration 版本
4.0.x	9.1.3 Continuous Delivery 版本	2019.3.2 (系統需求)
3.0.x	9.1.3 Continuous Delivery 版本	2019.3.1 (系統需求)

如需支援的版本資訊，請參閱 [IBM Cloud Pak for Integration 版本注意事項](#)。

## Linux 支援建置您自己的 IBM MQ 儲存器映像檔及圖表

在 Linux 系統上使用儲存器時要考量的資訊。

- 儲存器映像檔使用的基本映像檔必須使用受支援的 Linux 作業系統。
- 您必須使用 IBM MQ 安裝程式，在儲存器映像檔內安裝產品。
- 如需受支援套件的清單，請參閱 [Linux 系統的 IBM MQ rpm 元件](#)。
- **V 9.1.0** 不支援下列套件：
  - MQSeriesBCBridge
  - MQSeriesRDQM
- 佇列管理程式資料目錄 (依預設為 /var/mqm) 必須儲存在保持持續性狀態的容器磁區上。

**重要:** 您無法使用聯集檔案系統。

您必須將主機目錄裝載為資料磁區，或使用資料磁區儲存器。如需相關資訊，請參閱 [管理儲存器中的資料](#)。

- 您必須能夠在儲存器內執行 IBM MQ 控制指令，例如 **endmqm**。
- 您必須能夠從儲存器內取得檔案及目錄，以進行診斷。
- **V 9.1.0** 您可以使用名稱空間來與其他儲存器共用佇列管理程式的儲存器名稱空間，以便在本端將應用程式連結至在個別儲存器中執行的佇列管理程式。如需相關資訊，請參閱 [第 33 頁的『在個別儲存器中執行本端連結應用程式』](#)。

### 相關概念

[第 6 頁的『支援 IBM MQ 認證儲存器』](#)

IBM MQ 認證儲存器僅在特定 Kubernetes 環境中受支援

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

### 的儲存體考量

IBM MQ Advanced certified container 以兩種儲存模式執行：

- 當儲存器重新啟動時，可以刪除所有儲存器狀態時，會使用 **暫時儲存體**。這通常在建立環境以示範時使用，或在使用獨立式佇列管理程式進行開發時使用。
- **持續性儲存體** 是 IBM MQ 的一般配置，可確保在容器重新啟動時，現有配置、日誌及持續訊息在已重新啟動的容器中可用。

IBM MQ 操作器提供自訂儲存體性質的功能，視環境及想要的儲存體模式而定，儲存體性質可能會有很大差異。

### 臨時儲存空間

IBM MQ 是有狀態的應用程式，並將此狀態持續保存至儲存體，以在重新啟動時進行回復。如果使用暫時儲存體，重新啟動時將會遺失所有佇列管理程式狀態。這包括：

- 所有訊息
- 所有佇列管理程式至佇列管理程式的通訊狀態 (通道訊息序號)



- 佇列管理程式的 MQ 叢集身分
- 所有交易狀態
- 所有佇列管理程式配置
- 所有本端診斷資料

因此，您需要考量暫時儲存體是否適合正式作業、測試或開發實務範例。例如，已知所有訊息都是非持續性，且佇列管理程式不是「MQ 叢集」的成員。除了在重新啟動時刪除所有傳訊狀態之外，也會捨棄佇列管理程式配置。若要啟用完全暫時儲存器，必須將 IBM MQ 配置新增至儲存器映像檔本身(如需相關資訊，請參閱第 19 頁的『[使用 OpenShift CLI 來建置具有自訂 MQSC 和 INI 檔的映像檔](#)』)。如果未完成此作業，則每次容器重新啟動時都需要配置 IBM MQ。

例如，若要將 IBM MQ 配置為暫時儲存體，QueueManager 的儲存體類型應該包括下列：

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

## 持續性儲存體

IBM MQ 通常會使用持續性儲存體來執行，以確保在重新啟動之後佇列管理程式會保留其持續訊息及配置。因此，這是預設行為。由於各種儲存體提供者及每一個支援的不同功能，這通常表示需要自訂配置。下面概述在 v1beta1 API 中自訂 MQ 儲存體配置的一般欄位：

- `spec.queueManager.availability` 控制可用性模式。如果您使用 `SingleInstance`，則只需要 `ReadWriteOnce` 儲存體，而 `multiInstance` 需要具有正確檔案鎖定性質且支援 `ReadWriteMany` 的儲存類別。IBM MQ 提供 [支援陳述式](#) 及 [測試陳述式](#)。可用性模式也會影響持續性磁區佈置。如需相關資訊，請參閱：[第 9 頁的『IBM MQ Advanced certified container 的高可用性』](#)
- `spec.queueManager.storage` 控制個別儲存體設定。佇列管理程式可以配置成在一到四個持續性磁區之間使用

下列範例顯示使用單一實例佇列管理程式之簡式配置的 Snippet：

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

下列範例顯示多重實例佇列管理程式配置的 Snippet，其中含有非預設儲存類別，以及需要補充群組的檔案儲存體：

```
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 container 的高可用性

對於 IBM MQ Advanced certified container 的高可用性，您有兩個主要選項：**多重實例佇列管理程式**（這是使用共用網路檔案系統的主動-待命配對）和 **單一回復型佇列管理程式**（提供使用網路儲存體的 HA 簡單方法）。

您應該分別考量 **訊息** 和 **服務** 可用性。使用 IBM MQ for Multiplatforms，訊息只會儲存在一個佇列管理程式上。因此，如果該佇列管理程式變成無法使用，您會暫時無法存取它所保留的訊息。若要達到高 **訊息** 可用性，您需要能夠儘快回復佇列管理程式。您可以讓多個佇列實例供用戶端應用程式使用，例如使用 IBM MQ 統一叢集，以達到 **服務** 可用性。

佇列管理程式可以分為兩部分：儲存在磁碟上的資料，以及容許存取資料的執行中處理程序。任何佇列管理程式都可以移至不同的 Kubernetes 節點，只要它保留相同的資料（由 Kubernetes 持續性磁區提供），且仍可由用戶端應用程式在網路中定址。在 Kubernetes 中，「服務」是用來提供一致的網路身分。

IBM MQ 依賴於持續性磁區上資料的可用性。因此，提供持續性磁區的儲存體可用性對佇列管理程式可用性而言很重要，因為 IBM MQ 的可用程度不能超過它所使用的儲存體。如果您想要容忍整個可用性區域的中斷，則需要使用將磁碟寫入抄寫至另一個區域的磁區提供者。

## 多重實例佇列管理程式

多重實例佇列管理程式涉及一個 **作用中** 及一個 **待命** Kubernetes Pod，該 Pod 作為具有正好兩個副本及一組 Kubernetes 持續性磁區的 Kubernetes 有狀態集的一部分執行。使用共用檔案系統，將佇列管理程式交易日誌及資料保留在兩個持續性磁區上。

多重實例佇列管理程式需要 **作用中** 及 **待命** Pod 都具有持續性磁區的並行存取權。若要配置此項目，您可以使用 Kubernetes 持續性磁區，並將 **access mode** 設為 `ReadWriteMany`。磁區也必須符合 IBM MQ 共用檔案系統的需求，因為 IBM MQ 依賴自動釋放檔案鎖定來進行佇列管理程式失效接手。IBM MQ 會產生 已測試檔案系統的清單。

多重實例佇列管理程式的回復時間由下列因素控制：

1. 共用檔案系統在失敗之後釋放作用中實例最初所花費的鎖定所花費的時間。
2. 待命實例獲得鎖定然後啟動所需的時間。
3. Kubernetes Pod 就緒性探測偵測容器已備妥所花費的時間。這可在 Helm 圖表中配置。
4. IBM MQ 用戶端重新連接所花費的時間。

## 單一回復型佇列管理程式

單一回復型佇列管理程式是在單一 Kubernetes Pod 中執行之佇列管理程式的單一實例，其中 Kubernetes 會監視佇列管理程式，並視需要取代 Pod。

使用單一回復型佇列管理程式（租賃型鎖定除外）時，也適用 IBM MQ 共用檔案系統的需求，但您不需要使用共用檔案系統。您可以使用區塊儲存體，並在頂端安裝適當的檔案系統。例如，`xfs` 或 `ext4`。

單一回復型佇列管理程式的回復時間由下列因素控制：

1. 活性探測執行所花費的時間，以及它所容忍的失敗次數。這可在 Helm 圖表中配置。
2. Kubernetes 排程器將失敗 Pod 重新排定到新節點所需的時間。
3. 將儲存器映像檔下載至新「節點」所需的時間。如果您使用 **imagePullPolicy** 值 `IfNotPresent`，則映像檔可能已在該節點上可用。
4. 啟動新佇列管理程式實例所需的時間。
5. Kubernetes Pod 就緒性探測偵測容器已備妥所花費的時間。這可在 Helm 圖表中配置。
6. IBM MQ 用戶端重新連接所花費的時間。

### 重要：

雖然單一具復原力的佇列管理程式型樣提供一些好處，但您需要瞭解是否可以在「節點」失敗的相關限制下達成可用性目標。

在 Kubernetes 中，故障 Pod 通常會快速回復；但整個節點的故障會以不同方式處理。如果 Kubernetes 「主節點」失去與工作者節點的聯絡，則無法判斷節點是否失敗，或是否只是失去網路連線功能。因此，在此情況下，除非發生下列其中一個事件，否則 Kubernetes 不會採取 **任何動作**：

1. 節點會回復到 Kubernetes 主要節點可以與其通訊的狀態。
2. 已採取管理動作來明確刪除 Kubernetes 「主要節點」上的 Pod。這不一定會停止 Pod 執行，但只會從 Kubernetes 儲存庫中刪除它。因此，當局必須小心採取這項行政行動。

## 相關概念

### 高可用性配置

Linux

MQ Adv.

CD

## IBM MQ Advanced certified container 的使用者鑑

### 別和授權

IBM MQ 可以配置為使用 LDAP 使用者和群組進行授權。這是 IBM MQ Advanced certified container 的建議方法。

在多方承租戶容器化環境 (例如 Red Hat OpenShift Container Platform) 中，已設定安全限制，以防止潛在的安全問題。例如，在 Red Hat OpenShift Container Platform 中，預設 SecurityContextConstraints (稱為 `restricted`) 會使用隨機化使用者 ID，阻止任何儲存器本身本端的使用者。IBM MQ 通常會使用專用權提升來檢查使用者的密碼，在多租戶儲存器環境中也不建議這樣做。基於這些原因，在 IBM MQ 認證的儲存器中不支援使用在執行中儲存器內作業系統程式庫上定義的使用者。

您需要將佇列管理程式配置成使用 LDAP 進行使用者鑑別及授權。如需配置 IBM MQ 來執行此動作的相關資訊，請參閱 [連線鑑別: 使用者儲存庫](#) 和 [LDAP 授權](#)。

Linux

MQ Adv.

CD

V 9.1.5

## 在 OpenShift 上安裝及解除安裝

### IBM MQ 操作器

可以使用 Operator Hub 將 IBM MQ Operator 安裝到 OpenShift。

#### 開始之前

#### 程序

- [第 12 頁的『使用 OpenShift CLI 安裝 IBM MQ 操作器』](#)。
- [第 11 頁的『使用 OpenShift Web 主控台安裝 IBM MQ 操作器』](#)。

Linux

MQ Adv.

CD

V 9.1.5

## 使用 OpenShift Web 主控台安裝 IBM MQ

### 操作器

可以使用 Operator Hub 將 IBM MQ Operator 安裝到 OpenShift。

#### 開始之前

登入 OpenShift 叢集的 Web 主控台。

#### 程序

1. 將 IBM Common Services 操作員新增至可安裝操作員清單
  - a) 按一下加號圖示。您會看到 **匯入 YAML** 對話框。
  - b) 在對話框中貼上下列資源定義。

```
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) 按一下**建立**。

## 2. 將 IBM 操作器新增至可安裝操作器清單

- 按一下加號圖示。您會看到 **匯入 YAML** 對話框。
- 在對話框中貼上下列資源定義。

```
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
```

- 按一下**建立**。

## 3. 建立要用於 IBM MQ 操作器的名稱空間

IBM MQ 操作器可以安裝範圍限定為單一名稱空間或所有名稱空間。只有在您想要安裝至尚不存在的特定名稱空間時，才需要此步驟。

- 從導覽窗格中，按一下 **首頁 > 專案**。  
即會顯示「專案」頁面。
- 按一下**建立專案**。即會顯示「建立專案」區域。
- 輸入您要建立之名稱空間的詳細資料。例如，您可以指定 "ibm-mq" 作為名稱。
- 按一下**建立**。即會建立 IBM MQ 操作器的名稱空間。

## 4. 安裝 IBM MQ 運算子。

- 從導覽窗格中，按一下 **操作器 > OperatorHub**。  
即會顯示 OperatorHub 頁面。
- 在 **所有項目** 欄位中，輸入 "IBM MQ"。  
這時會顯示 IBM MQ 型錄項目。
- 選取 **IBM MQ**。  
即會顯示 IBM MQ 視窗。
- 按一下**安裝**。  
您會看到「建立操作員訂閱」頁面。
- 將安裝模式設為您建立的特定名稱空間或叢集範圍
- 按一下**訂閱**。  
您會在「已安裝的操作器」頁面上看到 IBM MQ。
- 在「已安裝的操作器」頁面上檢查「操作器」的狀態，當安裝完成時，狀態會變更為「成功」。

## 下一步

第 14 頁的『部署 IBM MQ 認證的儲存器』

Linux

MQ Adv.

CD

V 9.1.5

## 使用 OpenShift CLI 安裝 IBM MQ 操作器

可以使用 Operator Hub 將 IBM MQ Operator 安裝到 OpenShift。

## 開始之前

使用 **oc login** 登入 OpenShift 指令行介面 (CLI)。對於這些步驟，您將需要成為叢集管理者。

## 程序

- 為 IBM Common Services 操作員建立 OperatorSource

a) 建立定義 OperatorSource 資源的 YAML 檔

建立名為 "operator-source-cs.yaml" 的檔案，其中包含下列內容：

```
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) 將 OperatorSource 套用至伺服器。

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

2. 為 IBM 操作員建立 OperatorSource

a) 建立定義 OperatorSource 資源的 YAML 檔

建立名為 "operator-source-ibm.yaml" 的檔案，其中包含下列內容：

```
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) 將 OperatorSource 套用至伺服器。

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

3. 建立要用於 IBM MQ 操作器的名稱空間

IBM MQ 操作器可以安裝至單一名稱空間或所有名稱空間。只有在您想要安裝至尚不存在的特定名稱空間時，才需要此步驟。

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

4. 從 OperatorHub 檢視叢集可用的「運算子」清單

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

5. 檢查 IBM MQ 操作器，以驗證其支援的 InstallModes 及可用的通道

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

6. 建立 OperatorGroup 物件 YAML 檔案

OperatorGroup 是一個 OLM 資源，用於選取目標名稱空間，以在其中為與 OperatorGroup 相同的名稱空間中的所有操作員產生必要的 RBAC 存取權。

您訂閱「操作員」的名稱空間必須具有符合「操作員」的 InstallMode(AllNamespaces 或 SingleNamespace 模式)的 OperatorGroup。如果您想要安裝的操作器使用 AllNamespaces，則 openshift-operators 名稱空間已備妥適當的 OperatorGroup。

不過，如果「操作員」使用 SingleNamespace 模式，且您還沒有適當的 OperatorGroup，則必須建立一個。

a) 建立名為 "mq-operator-group.yaml" 的檔案，並包含下列內容：

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

b) 建立 OperatorGroup 物件

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

7. 建立訂閱物件 YAML 檔案，以訂閱「MQ 操作員」的名稱空間

a) 使用下列內容建立名為 "mq-sub.yaml" 的檔案：

```
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
```

對於 AllNamespaces **InstallMode** 用法，請指定 openshift-operators 名稱空間。否則，請指定 SingleNamespace **InstallMode** 用法的相關單一名稱空間。

b) 建立 Subscription 物件

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

8. 檢查操作員的狀態

在操作器安裝成功之後，Pod 狀態會顯示為 *Running*。對於 AllNamespaces **InstallMode** 用法，請指定 **openshift-operators** 作為名稱空間。否則，請指定 SingleNamespace **InstallMode** 用法的相關單一名稱空間。

## 下一步

[第 14 頁的『部署 IBM MQ 認證的儲存器』](#)

Linux

MQ Adv.

CD

## 部署 IBM MQ 認證的儲存器

IBM MQ 9.1.5 及更新版本可以使用 IBM MQ 操作器部署至 Red Hat OpenShift。IBM MQ 9.1.5 和 9.1.4 版可以使用 Helm 部署至 Red Hat OpenShift。您可以使用 Helm，將舊版 CD 部署至 IBM Cloud Private 叢集或 IBM Cloud Kubernetes Service 叢集。

## 關於這項作業

### 程序

- [第 22 頁的『使用 Helm CLI 部署佇列管理程式』](#)。
- [第 24 頁的『將 IBM MQ 的舊版 CD 部署至 IBM Cloud Private 叢集』](#)。
- [第 25 頁的『將舊版 CD 的 IBM MQ 映像檔新增至 IBM Cloud Private 叢集』](#)。
- [第 26 頁的『將舊版 CD 的 IBM MQ 映像檔新增至 IBM Cloud Kubernetes Service 叢集』](#)。



## 案

準備 Red Hat OpenShift Container Platform 叢集，以便它可以使用「IBM MQ 操作器」來部署佇列管理程式。此作業應由專案管理者完成。

### 開始之前

註: 如果您計劃在已安裝其他 IBM Cloud Pak for Integration 元件的專案中使用 IBM MQ，則可能不需要遵循這些指示。

使用 `cloudctl login` (適用於 IBM Cloud Pak for Integration) 或 `oc login` 登入叢集。

### 關於這項作業

IBM MQ Advanced certified container 映像檔是從執行授權檢查的容器登錄中取回。此檢查需要儲存在 `docker-registry` 取回密碼中的授權金鑰。如果您還沒有授權金鑰，請遵循下列指示來取得授權金鑰並建立取回密碼。

### 程序

1. 取得指派給您 ID 的授權金鑰。
  - a) 使用與授權軟體相關聯的 IBM ID 及密碼，登入 [MyIBM Container Software Library](#)。
  - b) 在**授權金鑰**區段，選取**複製金鑰**，以將該授權金鑰複製至剪貼簿。
2. 在您要部署佇列管理程式的專案中，建立包含授權金鑰的密鑰。

執行下列指令，其中 `<entitlement-key>` 是在步驟 1 中擷取的金鑰，`<user-email>` 是與授權軟體相關聯的 IBM ID。

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

### 下一步

第 17 頁的『[使用 OpenShift CLI 部署佇列管理程式](#)』

## Platform Navigator 部署佇列管理程式

使用 QueueManager 自訂資源，可使用 IBM Cloud Pak for Integration Platform Navigator 將佇列管理程式部署至 Red Hat OpenShift Container Platform 叢集。此作業應由專案管理者完成

### 開始之前

在瀏覽器中，啟動 IBM Cloud Pak for Integration Platform Navigator。

如果這是第一次將佇列管理程式部署至這個 Red Hat OpenShift 專案，請遵循 [第 15 頁的『使用 OpenShift CLI 為 IBM MQ 準備 OpenShift 專案』](#) 的步驟。

### 程序

1. 部署佇列管理程式。

下列範例會部署「快速入門」佇列管理程式，其使用暫時 (非持續性) 儲存體，並關閉 MQ 安全。在重新啟動佇列管理程式之後，將不會持續保存訊息。您可以調整配置來變更許多佇列管理程式設定。

  - a) 在 IBM Cloud Pak for Integration Platform Navigator 中，按一下 **執行時期及實例**。
  - b) 按一下 **建立實例**。

c) 選取 **佇列管理程式**，然後按 **下一步**。

即會顯示用來建立 QueueManager 實例的表單。

註: 您也可以按一下 **程式碼**，以檢視或變更 QueueManager 配置 YAML。

d) 在 **詳細資料** 區段中，檢查或更新 **名稱** 欄位，並指定要在其中建立佇列管理程式實例的 **名稱空間**。

e) 如果您接受 IBM Cloud Pak for Integration 授權合約，請將 **授權接受** 變更為 **開啟**。

您必須接受授權，才能部署佇列管理程式。

f) 在 **佇列管理程式配置** 區段中，檢查或更新基礎佇列管理程式的 **名稱**。

依預設，IBM MQ 用戶端應用程式使用的佇列管理程式名稱將與 QueueManager 的名稱相同，但會移除任何無效字元 (例如連字號)。如果您想要強制使用特定名稱，您可以在這裡編輯此名稱。

g) 按一下 **建立**

現在會顯示現行專案 (名稱空間) 中的佇列管理程式清單。新的 QueueManager 應該具有 Pending 狀態

2. 請檢查佇列管理程式是否在執行中

當 QueueManager 狀態為 Running 時，建立完成。

### 相關工作

第 27 頁的『[連接至部署在 OpenShift 叢集裡的佇列管理程式](#)』

一組配置範例，用於連接至部署在 Red Hat OpenShift 叢集中的佇列管理程式。

第 28 頁的『[連接至部署在 OpenShift 叢集裡的 IBM MQ Console](#)』

如何連接至已部署至 Red Hat OpenShift Container Platform 叢集之佇列管理程式的 IBM MQ Console。

Linux

MQ Adv.

CD

V 9.1.5

## 使用 OpenShift Web 主控台來部署佇列管

### 理程式

使用 QueueManager 自訂資源，利用 Red Hat OpenShift Web 主控台，將佇列管理程式部署至 Red Hat OpenShift Container Platform 叢集。此作業應由專案管理者完成

### 開始之前

登入 OpenShift 叢集的 Web 主控台。您需要選取要使用的現有專案 (名稱空間)，或建立新的專案 (名稱空間)。

如果這是第一次將佇列管理程式部署至這個 Red Hat OpenShift 專案，請遵循第 15 頁的『[使用 OpenShift CLI 為 IBM MQ 準備 OpenShift 專案](#)』的步驟。

### 程序

1. 部署佇列管理程式。

下列範例會部署「快速入門」佇列管理程式，其使用暫時 (非持續性) 儲存體，並關閉 MQ 安全。在重新啟動佇列管理程式之後，將不會持續保存訊息。您可以調整配置來變更許多佇列管理程式設定。

a) 在 OpenShift Web 主控台中，從導覽窗格中按一下 **操作器 > 已安裝的操作器**

b) 按一下 **IBM MQ**。

c) 按一下 **佇列管理程式** 標籤。

d) 按一下 **建立 QueueManager** 按鈕。

即會顯示 YAML 編輯器，其中包含 QueueManager 資源的範例 YAML。

註: 您也可以按一下 **編輯表單**，以檢視或變更 QueueManager 配置。

e) 如果您接受授權合約，請將 **授權接受** 變更為 **開啟**。

IBM MQ 在數個不同的授權下可用。如需有效授權的相關資訊，請參閱第 35 頁的『[mq.ibm.com/v1beta1 的授權參考手冊](#)』。您必須接受授權，才能部署佇列管理程式。

f) 按一下 **建立**



現在會顯示現行專案 (名稱空間) 中的佇列管理程式清單。新的 QueueManager 應該處於 Pending 狀態。

## 2. 請檢查佇列管理程式是否在執行中

當 QueueManager 狀態為 Running 時，建立完成。

### 相關工作

第 27 頁的『[連接至部署在 OpenShift 叢集裡的佇列管理程式](#)』

一組配置範例，用於連接至部署在 Red Hat OpenShift 叢集中的佇列管理程式。

第 28 頁的『[連接至部署在 OpenShift 叢集裡的 IBM MQ Console](#)』

如何連接至已部署至 Red Hat OpenShift Container Platform 叢集之佇列管理程式的 IBM MQ Console。

Linux

MQ Adv.

CD

V 9.1.5

## 使用 OpenShift CLI 部署佇列管理程式

使用 QueueManager 自訂資源，可使用指令行介面 (CLI) 將佇列管理程式部署至 Red Hat OpenShift Container Platform 叢集。此作業應由專案管理者完成

### 開始之前

您需要安裝 [Red Hat OpenShift Container Platform 指令行介面](#)。

使用 **cloudctl login** (適用於 IBM Cloud Pak for Integration) 或 **oc login** 登入叢集。

如果這是第一次將佇列管理程式部署至這個 Red Hat OpenShift 專案，請遵循 [第 15 頁的『使用 OpenShift CLI 為 IBM MQ 準備 OpenShift 專案』](#) 的步驟。

### 程序

#### 1. 部署佇列管理程式。

下列範例會部署「快速入門」佇列管理程式，其使用暫時 (非持續性) 儲存體，並關閉 MQ 安全。在重新啟動佇列管理程式之後，將不會持續保存訊息。您可以調整 YAML 的內容，以變更許多佇列管理程式設定。

##### a) 建立 QueueManager YAML 檔案

例如，若要在 IBM Cloud Pak for Integration 中安裝基本佇列管理程式，請建立具有下列內容的檔案 "mq-quickstart.yaml"：

```
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"
```

**重要事項:**如果您接受 IBM Cloud Pak for Integration 授權合約，請將 `accept: false` 變更為 `accept: true`。如需授權的詳細資料，請參閱 [第 35 頁的『mq.ibm.com/v1beta1 的授權參考手冊』](#)。

此範例還包括隨佇列管理程式一起部署的 Web 伺服器，其中已啟用 Web 主控台並使用 Cloud Pak Identity and Access Manager 啟用「單一登入」。

若要獨立於 IBM Cloud Pak for Integration 安裝基本佇列管理程式，請建立具有下列內容的檔案 "mq-quickstart.yaml"：

```
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"
```

**重要事項：**如果您接受 MQ 授權合約，請將 `accept: false` 變更為 `accept: true`。如需授權的詳細資料，請參閱第 35 頁的『mq.ibm.com/v1beta1 的授權參考手冊』。

#### b) 建立 QueueManager 物件

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

#### 2. 請檢查佇列管理程式是否在執行中

您可以執行下列指令來驗證部署：

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

，然後檢查狀態。

例如，執行

```
oc describe queuemanager quickstart
```

，並檢查 `status.Phase` 欄位是否指出 `Running`

#### 相關工作

第 27 頁的『[連接至部署在 OpenShift 叢集裡的佇列管理程式](#)』

一組配置範例，用於連接至部署在 Red Hat OpenShift 叢集中的佇列管理程式。

第 28 頁的『[連接至部署在 OpenShift 叢集裡的 IBM MQ Console](#)』

如何連接至已部署至 Red Hat OpenShift Container Platform 叢集之佇列管理程式的 IBM MQ Console。

## V 9.1.4 Linux MQ Adv. CD 與 IBM Cloud Pak for Integration 作業儀 表板整合

透過 IBM Cloud Pak for Integration 追蹤交易的能力由「作業儀表板」提供。

### 關於這項作業

啟用與「作業儀表板」的整合會將 MQ API 結束程式安裝至佇列管理程式。API 結束程式會將追蹤資料傳送至「作業儀表板」資料儲存庫，以瞭解流經佇列管理程式的訊息。

請注意，只會追蹤使用 MQ 用戶端連結傳送的訊息。

## 程序

### 1. 部署已啟用追蹤的佇列管理程式

依預設，會停用追蹤特性。

如果您使用 IBM Cloud Pak for Integration Platform Navigator 進行部署，則可以在部署時啟用追蹤，方法是將 **啟用追蹤** 設為 **開啟**，並將 **追蹤名稱空間** 設為已安裝「作業儀表板」的名稱空間。如需部署佇列管理程式的相關資訊，請參閱 [第 15 頁的『使用 IBM Cloud Pak for Integration Platform Navigator 部署佇列管理程式』](#)

如果您使用 [OpenShift CLI](#) 或 [OpenShift Web 主控台](#) 進行部署，則可以使用下列 YAML Snippet 來啟用追蹤：

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

如果您是使用 Helm 進行部署，則可以透過設定 `odTracingConfig.enabled=true` 和 `odTracingConfig.odTracingNamespace=<Operations_Dashboard_Namespace` 來啟用追蹤。如果您想要在現有佇列管理程式上啟用「作業儀表板」整合，則可以在升級 Helm 版次期間套用此設定。

**重要事項:** 在向「作業儀表板」登錄 MQ 之前，佇列管理程式將不會啟動（請參閱下一步）。

請注意，當啟用此特性時，除了佇列管理程式儲存器之外，它還會執行兩個 Sidecar 儲存器（「代理程式」和「收集器」）。這些 Sidecar 儲存器的映像檔將在與主要 MQ 映像檔相同的登錄中提供，並且將使用相同的取回原則和取回密碼。還有其他設定可用來配置 CPU 和記憶體限制。

### 2. 如果這是第一次在此名稱空間中部署具有「作業儀表板」整合的佇列管理程式，則您需要向「作業儀表板」登錄。

登錄會建立佇列管理程式 Pod 需要順利啟動的「密鑰」物件。

Linux

MQ Adv.

CD

V 9.1.5

## 使用 OpenShift CLI 來建置具有自訂

## MQSC 和 INI 檔的映像檔

使用「Red Hat OpenShift Container Platform 管線」來建立新的 IBM MQ 儲存器映像檔，並將您想要套用到使用此映像檔之佇列管理程式的 MQSC 及 INI 檔案。此作業應由專案管理者完成

### 開始之前

您需要安裝 [Red Hat OpenShift Container Platform 指令行介面](#)。

使用 `cloudctl login` (適用於 IBM Cloud Pak for Integration) 或 `oc login` 登入叢集。

如果您在 Red Hat OpenShift 專案中沒有 IBM Entitled Registry 的 OpenShift 密鑰，請遵循 [第 15 頁的『使用 OpenShift CLI 為 IBM MQ 準備 OpenShift 專案』](#) 的步驟。

## 程序

### 1. 建立 ImageStream

映像檔串流及其相關聯標籤提供抽象概念，可從 Red Hat OpenShift Container Platform 內參照儲存器映像檔。映像檔串流及其標籤可讓您查看可用的映像檔，並確保您正在使用您需要的特定映像檔，即使儲存庫中的映像檔變更也一樣。

```
oc create imagestream mymq
```

### 2. 為新映像檔建立 BuildConfig

BuildConfig 將容許對新映像檔進行建置，該映像檔將基於 IBM 正式映像檔，但將新增您要在容器啟動時執行的任何 MQSC 或 INI 檔案。

#### a) 建立定義 BuildConfig 資源的 YAML 檔

例如，使用下列內容建立稱為 "mq-build-config.yaml" 的檔案：

```

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'

```

您需要取代提及基本 IBM MQ 的兩個位置，以指向您要使用之版本及修正程式的正確基本映像檔。套用修正程式時，您需要重複這些步驟來重建映像檔。

此範例會根據 IBM 正式映像檔建立新的映像檔，並將稱為 "my.mqsc" 及 "my.ini" 的檔案新增至 /etc/mqm 目錄。儲存器會在啟動時套用在此目錄中找到的任何 MQSC 或 INI 檔。INI 檔案會使用 **crtmqm -ii** 選項來套用，並與現有的 INI 檔案合併。MQSC 檔案按字母順序套用。

MQSC 指令必須可重複，因為每次佇列管理程式啟動時都會執行這些指令。這通常表示在任何 DEFINE 指令上新增 REPLACE 參數，並將 IGNSTATE (YES) 參數新增至任何 START 或 STOP 指令。

- b) 將 BuildConfig 套用至伺服器。

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

### 3. 執行建置以建立映像檔

- a) 開始建置

```
oc start-build mymq
```

您應該會看到類似下列內容的輸出：

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

- b) 檢查建置的狀態

例如，您可以使用前一個步驟所傳回的建置 ID 來執行下列指令：

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

### 4. 使用新映像檔來部署佇列管理程式

遵循第 17 頁的『使用 OpenShift CLI 部署佇列管理程式』中說明的步驟，將新的自訂映像檔新增至 YAML。

您可以將下列 YAML Snippet 新增至一般 QueueManager YAML，其中 *my-namespace* 是您正在使用的 OpenShift 專案/名稱空間，而 *image* 是您先前建立的映像檔名稱 (例如 "mymq:latest-amd64")：

```

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

```

## 相關工作

第 17 頁的『使用 OpenShift CLI 部署佇列管理程式』

使用 QueueManager 自訂資源，可使用指令行介面 (CLI) 將佇列管理程式部署至 Red Hat OpenShift Container Platform 叢集。此作業應由專案管理者完成

從 IBM MQ 9.1.5.0 開始，建議使用 IBM MQ 操作器來部署佇列管理程式。IBM MQ 9.1.5.0 和舊版 CD 可以使用 Helm 來部署，並遵循下列指示。

## 關於這項作業

### 程序

- [第 21 頁的『使用 Helm 為 OpenShift 上的 IBM MQ 準備 OpenShift 叢集』](#)。
- [第 22 頁的『使用 Helm CLI 部署佇列管理程式』](#)。

## OpenShift 叢集

準備 Red Hat OpenShift Container Platform 叢集，以便它可以使 Helm 來部署佇列管理程式。此作業應該由叢集管理者完成。

### 開始之前

註: 如果您使用 IBM Cloud Pak for Integration，則安裝程式應該已準備好 OpenShift 專案 (名稱空間) 供您與 IBM MQ 搭配使用，因此您可能不需要遵循這些指示。

使用 `cloudctl login` (適用於 IBM Cloud Pak for Integration) 或 `oc login` 登入叢集。

### 程序

1. 確定您已將 IBM Helm 儲存庫新增至 Helm 的本端副本。  
例如，可以執行下列指令：

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

2. 確定您有 Helm 伺服器 (稱為 "Tiller") 已安裝在叢集上。  
遵循 [在 OpenShift 上開始使用 Helm](#) 中的指示，在叢集上安裝 Helm。
3. 確保 OpenShift 專案 (名稱空間) 中的「服務帳戶」已獲授權使用正確的「安全環境定義限制 (SCC)」。

#### V 9.1.5

IBM MQ 在預設 SCC "restricted" 下運作，因此通常可以跳過此步驟。

將變更套用至 SCC 需要由 OpenShift 叢集管理者完成。每一個 Helm Chart 版本都有 SCC 的不同需求，這些需求記錄在該 Helm Chart 的個別 README 檔中：

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

每一個 README 中都有設定 SCC 授權的指示。請注意，IBM MQ Helm 圖表會建立「服務帳戶」以供自己使用，這表示需要在「群組」層次套用 SCC 許可權 (適用於名稱空間中的所有「服務帳戶」)。

4. 請確定您具有有效的「映像檔取回密碼」，可從您選擇的容器登錄取回映像檔  
IBM MQ Advanced certified container 映像檔是從執行授權檢查的容器登錄中取回。此檢查需要儲存在 `docker-registry` 取回密碼中的授權金鑰。如果您還沒有授權金鑰，請遵循下列指示來取得授權金鑰並建立取回密碼。
  - a) 取得指派給您 ID 的授權金鑰。
    - i) 使用與授權軟體相關聯的 IBM ID 及密碼，登入 [MyIBM Container Software Library](#)。
    - ii) 在授權金鑰區段，選取**複製金鑰**，以將該授權金鑰複製至剪貼簿。
  - b) 在您要其中部署佇列管理程式的名稱空間中建立密碼。

- 執行下列指令，其中 `<entitlement-key>` 是在步驟 1 中擷取的金鑰，`<user-email>` 是與授權軟體相關聯的 IBM ID。

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

## 下一步

第 22 頁的『[使用 Helm CLI 部署佇列管理程式](#)』

V 9.1.4

Linux

MQ Adv.

CD

## 使用 Helm CLI 部署佇列管理程式

使用 Helm，將佇列管理程式部署至 Red Hat OpenShift Container Platform 叢集。此作業應由專案管理者完成。

## 開始之前

您需要安裝 [Helm V2](#) 及 Red Hat OpenShift Container Platform 指令行介面。如果您不是使用 IBM Cloud Pak for Integration，請遵循第 21 頁的『[使用 Helm 為 OpenShift 上的 IBM MQ 準備 OpenShift 叢集](#)』的步驟。

使用 `cloudctl login` (適用於 IBM Cloud Pak for Integration) 或 `oc login` 登入叢集。

## 程序

1. 確定您已將 IBM Helm 儲存庫新增至 Helm 的本端副本。  
例如，可以執行下列指令：

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

2. 檢閱佇列管理程式的配置選項

部署步驟包括安裝及配置步驟。佇列管理程式的部分設定必須在部署時設定，而變更它們需要重新部署。

您可以執行下列其中一個指令，來檢視 Helm Chart README，以取得所有可用部署選項的詳細資料：

- 若為 IBM Cloud Pak for Integration 中的 IBM MQ Advanced certified container：

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

- 若為 IBM MQ Advanced certified container：

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

您通常至少需要下列參數：

- a. 版本名稱。例如：`my-release`
  - b. 遠端 Helm 儲存庫。例如：`ibm-entitled-charts`
  - c. Helm 圖表：例如 `ibm-mqadvanced-server-prod` 或 `ibm-mqadvanced-server-integration-prod`
  - d. 映像檔取回密碼名稱。例如：`entitled-registry`。請注意，如果您要部署至 IBM Cloud Pak for Integration 中 MQ 的預先定義專案，則不需要這樣做。
3. 部署佇列管理程式。

請注意，依預設，Helm 圖表會假設您在 Red Hat OpenShift Container Platform 叢集裡已設定預設 [儲存類別](#)。



例如，若要在 IBM Cloud Pak for Integration 中安裝基本佇列管理程式，請執行下列指令：

```
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
```

您可以在 `tls.hostname` 欄位中輸入任何主機名稱 (這是必要欄位，但在此範例中不會使用，因為我們會產生新的自簽憑證)

若要獨立於 IBM Cloud Pak for Integration 安裝基本佇列管理程式，您可以執行下列指令：

```
helm install \  
--name my-release \  
ibm-entitled-charts/ibm-mqadvanced-server-prod \  
--set license=accept \  
--set image.pullSecret=ibm-entitlement-key
```

## 相關工作

第 27 頁的『[連接至部署在 OpenShift 叢集裡的佇列管理程式](#)』

一組配置範例，用於連接至部署在 Red Hat OpenShift 叢集中的佇列管理程式。

第 28 頁的『[連接至部署在 OpenShift 叢集裡的 IBM MQ Console](#)』

如何連接至已部署至 Red Hat OpenShift Container Platform 叢集之佇列管理程式的 IBM MQ Console。

## Linux > MQ Adv. > CD > V 9.1.5 使用 Helm CLI 來部署佇列管理程式與 IBM Cloud File Storage

使用 Helm 將佇列管理程式部署至 IBM Cloud 叢集上的 Red Hat OpenShift，並使用 IBM Cloud File Storage 的範例實務。此作業應由專案管理者完成

## 開始之前

您需要安裝 Helm V2 及 Red Hat OpenShift Container Platform 指令行介面。如果您不是使用 IBM Cloud Pak for Integration，請遵循第 21 頁的『[使用 Helm 為 OpenShift 上的 IBM MQ 準備 OpenShift 叢集](#)』的步驟。

使用 **cloudctl login** (適用於 IBM Cloud Pak for Integration) 或 **oc login** 登入叢集。

## 程序

1. 確定您已將 IBM Helm 儲存庫新增至 Helm 的本端副本。

例如，可以執行下列指令：

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

2. 部署佇列管理程式。

使用 IBM Cloud File Storage 時，您通常會看到使用 `ibmc-file-gold-gid` 儲存類別的最佳結果。此儲存類別會啟用使用者在正確檔案系統群組中可以寫入的儲存體。

例如，若要在 IBM Cloud Pak for Integration 中安裝基本佇列管理程式，請執行下列指令：

```
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}
```

您可以在 `tls.hostname` 欄位中輸入任何主機名稱 (這是必要欄位, 但在這裡未使用, 因為在此範例中, 我們會產生新的自簽憑證)。

若要獨立於 IBM Cloud Pak for Integration 安裝基本佇列管理程式, 您可以執行下列指令:

```
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}
```

## 相關工作

第 27 頁的『[連接至部署在 OpenShift 叢集裡的佇列管理程式](#)』

一組配置範例, 用於連接至部署在 Red Hat OpenShift 叢集中的佇列管理程式。

第 28 頁的『[連接至部署在 OpenShift 叢集裡的 IBM MQ Console](#)』

如何連接至已部署至 Red Hat OpenShift Container Platform 叢集之佇列管理程式的 IBM MQ Console。

## Linux MQ Adv. CD 將 IBM MQ 的舊版 CD 部署至 IBM Cloud Private 叢集

若為 9.1.4 之前的 IBM MQ CD 版本, 請使用 IBM Cloud Private 管理主控台, 將佇列管理程式部署至 IBM Cloud Private。

## 開始之前



**小心:** **V 9.1.4** IBM MQ 9.1.4 或更新版本不支援此部署。

此作業假設您已將 IBM MQ 映像檔新增至 IBM Cloud Private 叢集。

Helm Chart README.md 檔可從 IBM Cloud Private 型錄項目取得 (在您完成 這個子步驟之後會顯示), 或從指令行 [新增 IBM Cloud Private 的 local-charts 儲存庫](#) 作為遠端 Helm 儲存庫, 並執行下列指令:

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

您必須具有支援必要安全環境定義的 PodSecurity 原則或 SecurityContext 限制 (適用於 Red Hat OpenShift 上的 IBM Cloud Private)。您可以從 Helm Chart README.md 檔案中找到詳細資料 (包括範例)。

您也可以從 Helm chart README.md 檔案中找到如何配置 Helm 版本的詳細資料。

### 註:

- 如果您要部署至預設不支援必要安全設定的 IBM Cloud Private 環境, 請遵循 IBM Cloud Private 產品說明文件中 [部署在非預設名稱空間中需要較高專用權的 Helm 圖表](#) 中的指示來啟用部署。
- 如果您使用 SELinux, 則必須符合 [IBM MQ 支援 Red Hat Enterprise Linux](#) 中說明的 IBM MQ 需求。

## 關於這項作業

IBM Cloud Private 提供一個平台來管理內部部署儲器化應用程式。將 IBM MQ 映像檔新增至 IBM Cloud Private 叢集之後, 您可以使用 IBM Cloud Private 管理主控台或指令行來部署佇列管理程式。

## 程序

- 使用 IBM Cloud Private 管理主控台
  - a) 在 Web 瀏覽器中開啟 IBM Cloud Private 管理主控台, 然後按一下 **型錄**。  
請參閱 IBM Cloud Private 產品說明文件中的 [使用管理主控台來存取 IBM Cloud Private 叢集](#)。
  - b) 從清單中選取 `ibm-mqadvanced-server-prod` 圖表。
  - c) 選取 **配置**, 然後完成下列配置步驟:



- a. 輸入版次名稱。
- b. 閱讀並接受授權合約。
- c. 在 **dataPVC** 區段下，將 **storageclass** 設為您想要的儲存空間類別。保留空白以選取預設儲存類別。
- d. 在 **image** 區段下，將儲存庫設為完整映像檔路徑。例如：

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

- e. 在 **image** 區段下，將標籤設為 image 標籤。例如：

```
9.1.3.0-r1
```

- f. 如果您需要 Kubernetes 取回密碼才能存取映像檔登錄，請將它新增為 **pullSecret**。
- g. 在 **queueManager** 區段下，設定佇列管理程式的名稱。

- d) 按一下 **安裝**，將佇列管理程式部署為 *Helm* 版次。
- 使用指令行

- a) 配置 **cloudctl** 以存取 IBM Cloud Private 叢集。

請參閱 IBM Cloud Private 產品說明文件中的 [安裝 IBM Cloud Private CLI](#)。

- b) 確定您已 [新增 IBM Cloud Private 的 local-charts 儲存庫作為遠端 Helm 儲存庫](#)。

- c) 安裝圖表。

執行下列指令，並指定這些參數：

- a. 版本名稱 (例如 `my-release`)
- b. 包含 `ibm-mqadvanced-server-prod` 圖表的遠端 Helm 儲存庫名稱 (例如 `my-repo`)
- c. 映像檔儲存庫 (例如 `mycluster.icp:8500/namespace_name/ibm-mqadvanced-server-prod`)
- d. 映像檔標籤 (例如 `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
```

## 相關工作

第 22 頁的『[使用 Helm CLI 部署佇列管理程式](#)』

使用 Helm，將佇列管理程式部署至 Red Hat OpenShift Container Platform 叢集。此作業應由專案管理者完成。

第 25 頁的『[將舊版 CD 的 IBM MQ 映像檔新增至 IBM Cloud Private 叢集](#)』

若為 9.1.4 之前的 IBM MQ CD 版本，請準備 IBM Cloud Private 叢集，以部署 IBM MQ 的可正式作業映像檔。

第 26 頁的『[將舊版 CD 的 IBM MQ 映像檔新增至 IBM Cloud Kubernetes Service 叢集](#)』

若為 9.1.4 之前的 IBM MQ CD 版本，請將 IBM MQ 的可正式作業映像檔匯入至 IBM Cloud Kubernetes Service。

Linux

MQ Adv.

CD

## 將舊版 CD 的 IBM MQ 映像檔新增至 IBM Cloud

### Private 叢集

若為 9.1.4 之前的 IBM MQ CD 版本，請準備 IBM Cloud Private 叢集，以部署 IBM MQ 的可正式作業映像檔。

### 關於這項作業



小心： **V 9.1.4** IBM MQ 9.1.4 或更新版本不支援此匯入。

您可以從 Passport Advantage 下載 IBM MQ 映像檔，並將它匯入至 IBM Cloud Private 儲存器。

## 程序

1. 從 [Passport Advantage](#) 和 [Passport Advantage Express](#) 網站下載最新 IBM MQ 映像檔。  
如需可用下載的詳細資料，請跳至 [下載 IBM MQ 9.1](#)，然後按一下您要下載之版本的標籤。表格中列出要下載之組件的名稱及號碼。
2. 將下載的保存檔匯入至 IBM Cloud Private。  
請參閱 [IBM Cloud Private 產品說明文件中的將 IBM 軟體新增至 IBM Cloud Private 型錄](#)。

## 下一步

現在，您已準備好 [將佇列管理程式部署至 IBM Cloud Private](#)。

### 相關工作

第 22 頁的『[使用 Helm CLI 部署佇列管理程式](#)』

使用 Helm，將佇列管理程式部署至 Red Hat OpenShift Container Platform 叢集。此作業應由專案管理者完成。

第 24 頁的『[將 IBM MQ 的舊版 CD 部署至 IBM Cloud Private 叢集](#)』

若為 9.1.4 之前的 IBM MQ CD 版本，請使用 IBM Cloud Private 管理主控台，將佇列管理程式部署至 IBM Cloud Private。

第 26 頁的『[將舊版 CD 的 IBM MQ 映像檔新增至 IBM Cloud Kubernetes Service 叢集](#)』

若為 9.1.4 之前的 IBM MQ CD 版本，請將 IBM MQ 的可正式作業映像檔匯入至 IBM Cloud Kubernetes Service。

## Linux > MQ Adv. > CD 將舊版 CD 的 IBM MQ 映像檔新增至 IBM Cloud Kubernetes Service 叢集

若為 9.1.4 之前的 IBM MQ CD 版本，請將 IBM MQ 的可正式作業映像檔匯入至 IBM Cloud Kubernetes Service。

### 關於這項作業



小心: **V 9.1.4** IBM MQ 9.1.4 或更新版本不支援此匯入。

您可以從 Passport Advantage 下載 IBM MQ 映像檔，並將它匯入至 IBM Cloud Kubernetes Service 叢集。

## 程序

1. 從 [Passport Advantage](#) 和 [Passport Advantage Express](#) 網站下載最新 IBM MQ 映像檔。  
如需可用下載的詳細資料，請跳至 [下載 IBM MQ 9.1](#)，然後按一下您要下載之版本的標籤。表格中列出要下載之組件的名稱及號碼。
2. 將下載的保存檔匯入至 IBM Cloud Kubernetes Service。  
請參閱 [在公用 Kubernetes 容器中執行 IBM Cloud Private 映像檔](#)。

### 相關工作

第 22 頁的『[使用 Helm CLI 部署佇列管理程式](#)』

使用 Helm，將佇列管理程式部署至 Red Hat OpenShift Container Platform 叢集。此作業應由專案管理者完成。

第 24 頁的『[將 IBM MQ 的舊版 CD 部署至 IBM Cloud Private 叢集](#)』

若為 9.1.4 之前的 IBM MQ CD 版本，請使用 IBM Cloud Private 管理主控台，將佇列管理程式部署至 IBM Cloud Private。

第 25 頁的『[將舊版 CD 的 IBM MQ 映像檔新增至 IBM Cloud Private 叢集](#)』

若為 9.1.4 之前的 IBM MQ CD 版本，請準備 IBM Cloud Private 叢集，以部署 IBM MQ 的可正式作業映像檔。

## V 9.1.4 Linux MQ Adv. CD 連接至部署在 OpenShift 叢集裡的佇列管理程式

一組配置範例，用於連接至部署在 Red Hat OpenShift 叢集中的佇列管理程式。

### 關於這項作業

您需要 [OpenShift 路徑](#)，才能從 Red Hat OpenShift 叢集外部將應用程式連接至 IBM MQ 佇列管理程式。

您必須在 IBM MQ 佇列管理程式及用戶端應用程式上啟用 TLS，因為 伺服器名稱指示 (SNI) 只能在 TLS 通訊協定中使用。「Red Hat OpenShift Container Platform 路由器」會使用 SNI 將要求遞送至 IBM MQ 佇列管理程式。

OpenShift 路徑的必要配置取決於用戶端應用程式的 SNI 行為。

若要將 SNI 標頭設為 TLS 1.2 或更高版本，CipherSpec 或 CipherSuite 必須用於 TLS 通訊。

如果符合下列條件，則 SNI 會設為 MQ 通道：

- IBM MQ C 用戶端是 V8 或更新版本。
- Java/JMS 用戶端是 V9.1.1 或更新版本，Java 安裝支援 `javax.net.ssl.SNIHostName` 類別。
- 「.NET 用戶端」處於未受管理的模式。

如果提供主機名稱作為連線名稱，且符合下列條件，則 SNI 會設為「主機名稱」：

- .NET 用戶端處於受管理模式。
- 使用 AMQP 或 XR 用戶端。
- Java/JMS 用戶端與設為 NO 的 **AllowOutboundSNI** 搭配使用。

在下列情況下，SNI 未設定且為空白：

- IBM MQ C 用戶端是 V7.5 或更舊版本。
- IBM MQ C 用戶端與設為 NO 的 **AllowOutboundSNI** 搭配使用。
- Java/JMS 用戶端與不支援 `javax.net.ssl.SNIHostName` 類別的 Java 安裝搭配使用。

### 範例

#### 主機名稱型 OpenShift 路徑：適用於將 SNI 設為主機名稱的用戶端應用程式

下列 Helm 圖表會自動建立主機名稱型 OpenShift 路徑，以將應用程式連接至 IBM MQ 佇列管理程式。將 SNI 設為主機名稱的用戶端應用程式可以使用此 OpenShift 路徑。

- `ibm-mqadvanced-server-dev`
- `ibm-mqadvanced-server-prod`
- IBM Cloud Pak for Integration 中的 `ibm-mqadvanced-server-integration-prod`。

如果您不使用這些圖表，並且需要建立您自己的主機名稱型 OpenShift 路徑，則可以在叢集裡套用下列 `yaml`：

```
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
```

### MQ 通道型 OpenShift 路徑: 適用於將 SNI 設為 MQ 通道的用戶端應用程式

將 SNI 設為 MQ 通道的用戶端應用程式需要為您要連接的每一個通道建立新的 OpenShift 路徑。您也必須在 Red Hat OpenShift 叢集中使用唯一通道名稱，以容許遞送至正確的佇列管理程式。

若要判定每一個新的 OpenShift 路徑所需的主機名稱，您需要將每一個通道名稱對映至 SNI 位址，如這裡所記載: <https://www.ibm.com/support/pages/ibm-websphere-mq-how-does-mq-provide-multiple-certificates-certlabl-capability>

然後，您必須在叢集裡套用下列 yam1，以建立新的 OpenShift 路徑 (適用於每一個通道):

```
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
```

### 配置用戶端應用程式連線詳細資料

您可以執行下列指令來決定要用於用戶端連線的主機名稱:

```
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}"
```

用戶端連線的埠應該設為 OpenShift Container Platform (OCP) Router 所使用的埠-通常是 443。

### 相關工作

第 22 頁的『使用 Helm CLI 部署佇列管理程式』

使用 Helm，將佇列管理程式部署至 Red Hat OpenShift Container Platform 叢集。此作業應由專案管理者完成。

第 28 頁的『連接至部署在 OpenShift 叢集裡的 IBM MQ Console』

如何連接至已部署至 Red Hat OpenShift Container Platform 叢集之佇列管理程式的 IBM MQ Console。

V 9.1.4

Linux

MQ Adv.

CD

## 連接至部署在 OpenShift 叢集裡的

## IBM MQ Console

如何連接至已部署至 Red Hat OpenShift Container Platform 叢集之佇列管理程式的 IBM MQ Console。

### 關於這項作業

如果您使用 IBM MQ 操作器，則可以在 OpenShift Web 主控台或 IBM Cloud Pak for Integration Platform Navigator 中的 QueueManager 詳細資料頁面上找到 IBM MQ Console URL。或者，可以透過執行下列指令，從 OpenShift CLI 中找到它:

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

### 範例

下列 Helm 圖表會自動建立 OpenShift 路徑，以存取 IBM MQ Console

- ibm-mqadvanced-server-dev

- IBM Cloud Pak for Integration 中的 `ibm-mqadvanced-server-integration-prod`。

您可以執行下列指令，以取得 OpenShift 路徑的主機名稱：

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

您可以使用下列 URL 來存取 IBM MQ Console：

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

## 相關工作

第 22 頁的『使用 Helm CLI 部署佇列管理程式』

使用 Helm，將佇列管理程式部署至 Red Hat OpenShift Container Platform 叢集。此作業應由專案管理者完成。

第 27 頁的『連接至部署在 OpenShift 叢集裡的佇列管理程式』

一組配置範例，用於連接至部署在 Red Hat OpenShift 叢集中的佇列管理程式。

# Linux > MQ Adv. > CD 使用 OpenShift CLI 備份及還原佇列管理程式配置

如果佇列管理程式配置遺失，備份佇列管理程式配置可協助您從其定義重建佇列管理程式。此程序不會備份佇列管理程式日誌資料。由於訊息的暫時性，在還原時，歷程日誌資料可能不相關。

## 開始之前

使用 `cloudctl login` (適用於 IBM Cloud Pak for Integration) 或 `oc login` 登入叢集。

## 程序

- 備份佇列管理程式配置。

您可以使用 `dmpmqcfg` 指令來傾出 IBM MQ 佇列管理程式的配置。

- a) 取得佇列管理程式的 Pod 名稱。

例如，如果您使用「操作員」，則可以執行下列指令，其中 `queue_manager_name` 是 QueueManager 資源的名稱：

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

例如，如果您使用 Helm，則可以執行下列指令，其中 `release_name` 是 Helm 版本的名稱。

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

- b) 在 Pod 上執行 `dmpmqcfg` 指令，將輸出導向本端機器上的檔案。

`dmpmqcfg` 會輸出佇列管理程式的 MQSC 配置。

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

- 還原佇列管理程式配置。

遵循前一個步驟所概述的備份程序之後，您應該有一個包含佇列管理程式配置的 `backup.mqsc` 檔。您可以將此檔案套用至新的佇列管理程式來還原配置。

- a) 取得佇列管理程式的 Pod 名稱。

例如，如果您使用「操作員」，則可以執行下列指令，其中 `queue_manager_name` 是 QueueManager 資源的名稱：

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

例如，如果您使用 Helm，則可以執行下列指令，其中 `release_name` 是 Helm 版本的名稱。

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

b) 在 Pod 上執行 `runmqsc` 指令，並在 `backup.mqsc` 檔案的內容中導向。

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

## 建置您自己的 IBM MQ 儲存器

開發自行建置的容器，先前稱為 "Docker 容器映像檔"。這是最具彈性的容器解決方案，但它需要您具備配置容器的強大技能，以及 "擁有" 產生的容器。

### 開始之前

在開發您自己的容器之前，請考量您是否可以改用 IBM 所提供的其中一個預先包裝容器。請參閱 [儲存器中的 IBM MQ](#)

### 關於這項作業

當您將 IBM MQ 包裝成容器映像檔時，應用程式的變更可以快速且輕鬆地部署至測試和暫置系統。這可能是企業持續交付的主要好處。

### 程序

- 如需如何使用 Docker 來建置 IBM MQ 儲存器映像檔的相關資訊，請參閱下列子主題：
  - [Linux](#) 第 8 頁的『支援建置您自己的 IBM MQ 儲存器映像檔及圖表』
  - 第 30 頁的『使用儲存器規劃您自己的 IBM MQ 佇列管理程式映像檔』
  - 第 31 頁的『使用 Docker 建置範例 IBM MQ 佇列管理程式映像檔』
  - 第 33 頁的『在個別儲存器中執行本端連結應用程式』

### 相關概念

[儲存器中的 IBM MQ](#)

## 使用儲存器規劃您自己的 IBM MQ 佇列管理程式映像檔

在儲存器中執行 IBM MQ 佇列管理程式時，有數個需求需要考量。範例容器映像檔提供處理這些需求的方法，但如果您想要使用自己的映像檔，則需要考量如何處理這些需求。

### 程序監督

當您執行容器時，基本上是執行單一處理程序 (容器內的 PID 1)，它可以在稍後大量產生子處理程序。

如果主要處理程序結束，則儲存器執行時期會停止儲存器。IBM MQ 佇列管理程式需要多個處理程序在背景中執行。

因此，您需要確定只要佇列管理程式在執行中，您的主要處理程序就會保持作用中。例如，透過執行管理查詢來檢查佇列管理程式在此處理程序中是否處於作用中狀態，是很好的作法。

### 移入資料 /var/mqm

儲存器必須以 `/var/mqm` 作為磁區來配置。

當您這麼做時，當儲存器第一次啟動時，磁區的目錄是空的。此目錄通常在安裝時移入，但在使用儲存器時，安裝與執行時期是個別環境。

**V9.1.0** 若要解決此問題，當容器啟動時，您可以使用 `crtmqdir` 指令，在 `/var/mqm` 第一次執行時移入它。



## 使用 Docker 建置範例 IBM MQ 佇列管理程式映像檔

使用此資訊來建置範例儲存器映像檔，以在儲存器中執行 IBM MQ 佇列管理程式。

### 關於這項作業

首先，您建置基本映像檔，其中包含 Red Hat 通用基本映像檔檔案系統及 IBM MQ 的全新安裝。

其次，您在基本程式之上建置另一個容器映像檔層，這會新增一些 IBM MQ 配置，以容許基本使用者 ID 及密碼安全。

最後，您可以使用此映像檔作為其檔案系統來執行儲存器，並使用主機檔案系統上儲存器特定磁區所提供的 `/var/mqm` 內容。

### 程序

- 如需如何建置範例儲存器映像檔以在儲存器中執行 IBM MQ 佇列管理程式的相關資訊，請參閱下列子主題：
  - [第 31 頁的『建置範例基本 IBM MQ 佇列管理程式映像檔』](#)
  - [第 31 頁的『建置已配置的範例 IBM MQ 佇列管理程式映像檔』](#)

### 建置範例基本 IBM MQ 佇列管理程式映像檔

為了在您自己的容器映像檔中使用 IBM MQ，您一開始需要使用全新的 IBM MQ 安裝來建置基本映像檔。下列步驟顯示如何使用在 GitHub 上管理的範例程式碼來建置範例基本映像檔。

### 程序

- 使用 `mq-container` GitHub 儲存庫中提供的 `make` 檔來建置正式作業儲存器映像檔。  
遵循 GitHub 上 [建置容器映像檔](#) 中的指示。

### 結果

您現在具有已安裝 IBM MQ 的基本容器映像檔。

### 建置已配置的範例 IBM MQ 佇列管理程式映像檔

建置通用基本 IBM MQ 儲存器映像檔之後，您需要套用自己的配置，以容許安全存取。若要這樣做，請使用一般映像檔作為母項來建立您自己的容器映像檔層。

### 開始之前

對於 IBM MQ 9.1 映像檔，您無法使用 Red Hat OpenShift Container Platform "受限" 安全環境定義限制 (SCC) 來配置安全存取。"restricted" SCC 會使用隨機使用者 ID，並透過變更為不同的使用者來防止專用權升級。IBM MQ 9.1 RPM 型安裝程式依賴於 `mqm` 使用者和群組，並且也使用可執行程式上的 `setuid` 位元。

在 IBM MQ 9.2 中已移除此限制。

### 程序

1. 建立新的目錄，並新增名為 `config.mqsc` 的檔案，其內容如下：

```
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)
```

請注意，前述範例使用簡式使用者 ID 和密碼鑑別。不過，您可以套用企業需要的任何安全配置。

2. 建立名為 `Dockerfile` 的檔案，其內容如下：

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

其中：

- `johndoe` 是您要新增的使用者 ID
- `passw0rd` 是原始密碼

3. 使用下列指令來建置自訂容器映像檔：

```
sudo docker build -t mymq .
```

其中 `.` 是包含您剛建立的兩個檔案的目錄。

然後，`Docker` 會使用該映像檔建立暫時儲存器，並執行其餘指令。

`RUN` 指令會新增名為 `johndoe` 且密碼為 `passw0rd` 的使用者，而 `COPY` 指令會將 `config.mqsc` 檔案新增至母項映像檔已知的特定位置。

註：在 Red Hat Enterprise Linux (RHEL) 上，您使用指令 `docker` (RHEL V7) 或 `podman` (RHEL V7 或 RHEL V8)。若為 `podman`，指令開頭不需要 `sudo`。

4. 執行新的自訂映像檔，以使用您剛才建立的磁碟映像檔來建立新的容器。

您的新映像檔層未指定任何要執行的特定指令，因此已從母項映像檔繼承。母項的進入點 (程式碼可在 GitHub 上找到)：

- 建立佇列管理程式
- 啟動佇列管理程式
- 建立預設接聽器
- 然後從 `/etc/mqm/config.mqsc` 執行任何 MQSC 指令

發出下列指令，以執行新的自訂映像檔：

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

其中：

#### 第一個 `env` 參數

將環境變數傳遞至儲存器，以確認您接受 IBM WebSphere MQ 的授權。您也可以設定要檢視的 `LICENSE` 變數，以檢視授權。

如需 IBM MQ 授權的進一步詳細資料，請參閱 [IBM MQ 授權資訊](#)。

#### 第二個 `env` 參數

設定您正在使用的佇列管理程式名稱。

#### 磁區參數

告訴儲存器，MQ 寫入 `/var/mqm` 的任何內容實際上都應該寫入主機上的 `/var/example`。

此選項表示您稍後可以輕鬆地刪除儲存器，並仍然保留任何持續資料。此選項也可讓您更容易檢視日誌檔。

#### 發佈參數

將主機系統上的埠對映至儲存器中的埠。依預設，容器會使用其自己的內部 IP 位址來執行，這表示您需要特別對映您要公開的任何埠。

在此範例中，這表示將主機上的埠 1414 對映至儲存器中的埠 1414。



## 分離參數

在背景中執行儲存器。

## 結果

您已建置已配置的儲存器映像檔，並且可以使用 `docker ps` 指令來檢視執行中的儲存器。您可以使用 `docker top` 指令來檢視在容器中執行的 IBM MQ 處理程序。



### 小心：

您可以使用 `docker logs ${CONTAINER_ID}` 指令來檢視容器的日誌。

## 下一步

- 如果您使用 Docker `ps` 指令時未顯示容器，則容器可能已失敗。您可以使用 `docker ps -a` 指令來查看失敗的儲存器。
- 當您使用 Docker `ps -a` 指令時，會顯示儲存器 ID。當您發出 `docker run` 指令時，也會列印這個 ID。
- 您可以使用 `docker logs ${CONTAINER_ID}` 指令來檢視容器的日誌。
- 您可以使用指令 `sysctl fs.file-max=524288` 來設定開啟檔案的數目上限。

## V 9.1.0 在個別儲存器中執行本端連結應用程式

使用 Docker 中儲存器之間的程序名稱空間共用，您可以在 IBM MQ 佇列管理程式的個別儲存器中執行需要本端連結連線至 IBM MQ 的應用程式。

## 關於這項作業

此功能在 IBM MQ 9.0.3 以及更新版本的佇列管理程式中受支援。

您必須遵守下列限制：

- 您必須使用 `--pid` 引數來共用儲存器 PID 名稱空間。
- 您必須使用 `--ipc` 引數來共用儲存器 IPC 名稱空間。
- 您必須：
  1. 使用 `--uts` 引數與主機共用儲存器 UTS 名稱空間，或
  2. 使用 `-h` 或 `--hostname` 引數，確保儲存器具有相同的主機名稱。
- 您必須在可供 `/var/mqm` 目錄下所有儲存器使用的磁區中裝載 IBM MQ 資料目錄。

在已安裝 Docker 的 Linux 系統上完成下列步驟，即可試用此功能。

下列範例使用範例 IBM MQ 儲存器映像檔。您可以在 [Github](#) 上找到此映像檔的詳細資料。

## 程序

1. 發出下列指令，以建立暫存目錄來作為您的磁區：

```
mkdir /tmp/dockerVolume
```

2. 發出下列指令，在儲存器中建立名為 `sharedNamespace` 的佇列管理程式 (QM1)：

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

3. 發出下列指令，以啟動稱為 `secondaryContainer` 的第二個儲存器 (以 `ibmcom/mq` 為基礎)，但不建立佇列管理程式：

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

4. 發出下列指令，在第二個儲存器上執行 **dspmq** 指令，以查看兩個佇列管理程式的狀態：

```
docker exec secondaryContainer dspmq
```

5. 執行下列指令，以針對在其他儲存器上執行的佇列管理程式處理 MQSC 指令：

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

## 結果

現在，您已在個別儲存器中執行本端應用程式，而且現在可以從次要儲存器順利執行 **dspmq**、**amqspout**、**amqsget** 及 **runmqsc** 等指令作為 QM1 佇列管理程式的本端連結。

如果您未看到預期的結果，請參閱第 34 頁的『對名稱空間應用程式進行疑難排解』以取得相關資訊。

### V 9.1.0 對名稱空間應用程式進行疑難排解

使用共用名稱空間時，您必須確保共用所有名稱空間 (IPC、PID 及 UTS/hostname) 及裝載磁區，否則您的應用程式將無法運作。

如需您必須遵循的限制清單，請參閱第 33 頁的『在個別儲存器中執行本端連結應用程式』。

如果您的應用程式不符合列出的所有限制，您可能會在儲存器啟動時遇到問題，但您預期的功能無法運作。

下列清單概述一些常見原因，以及您在忘記符合其中一項限制時可能看到的行為。

- 如果您忘記共用名稱空間 (UTS/PID/IPC) 或容器的主機名稱，並裝載磁區，則容器將能夠看到佇列管理程式，但無法與佇列管理程式互動。
  - 對於 **dspmq** 指令，您會看到下列：

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

- 對於 **runmqsc** 指令，或嘗試連接至佇列管理程式的其他指令，您可能會收到 AMQ8146 錯誤訊息：

```
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
```

- 如果您共用所有必要的名稱空間，但未將共用磁區裝載至 `/var/mqm` 目錄，且您具有有效的 IBM MQ 資料路徑，則您的指令也會收到 AMQ8146 錯誤訊息。

不過，**dspmq** 完全無法看到您的佇列管理程式，而是傳回空白回應：

```
docker exec container dspmq
```

- 如果您共用所有必要的名稱空間，但未將共用磁區裝載至 `/var/mqm` 目錄，且您沒有有效的 IBM MQ 資料路徑 (或沒有 IBM MQ 資料路徑)，則會看到各種錯誤，因為資料路徑是 IBM MQ 安裝架構的重要元件。如果沒有資料路徑，則 IBM MQ 無法運作。

如果您執行下列任何指令，且看到類似這些範例中所示的回應，則應該驗證您已裝載目錄或建立 IBM MQ 資料目錄：

```
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

```

Linux MQ Adv. CD V 9.1.5 **IBM MQ 運算子的 API 參考資料**

IBM MQ 提供 Kubernetes 操作器，它提供與 OpenShift Container Platform 的原生整合。

Linux MQ Adv. CD V 9.1.5 **mq.ibm.com/v1beta1 的 API 參考資料**

v1beta1 API 可用來建立及管理 QueueManager 資源。

Linux MQ Adv. CD V 9.1.5 **mq.ibm.com/v1beta1 的授權參考手冊**

spec.license.license 欄位必須包含您接受之授權的授權 ID。有效值如下：

下者的值： <b>spec.license.license</b>	下者的值： <b>spec.license.use</b>	授權資訊
L-RJON-BN7PN3	Production 或 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 或 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>

請注意，已指定授權 *version*，不一定與 IBM MQ 的版本相同。

Linux MQ Adv. CD V 9.1.5 **QueueManager 的 API 參考資料 (mq.ibm.com/v1beta1)**

### QueueManager

QueueManager 是 IBM MQ 伺服器，可為應用程式提供佇列作業及發佈/訂閱服務。

欄位	說明
apiVersion 字串	APIVersion 定義這個物件表示法的版本化綱目。伺服器應該將可辨識的綱目轉換成最新的內部值，且可能會拒絕無法辨識的值。進一步資訊: <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> 。

欄位	說明
kind 字串	Kind 是一個字串值，代表此物件所代表的 REST 資源。伺服器可能會從用戶端向其提交要求的端點推斷此情況。無法更新。在 CamelCase 中。進一步資訊： <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-tind">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-tind</a> 。
metadata	
spec 第 39 頁的『QueueManager 規格』	所需的 QueueManager 狀態。
status 第 40 頁的『QueueManagerStatus』	QueueManager 的觀察狀態。

## 可用性

佇列管理程式的可用性設定，例如是否使用作用中-待命配對。

出現在：

- 第 37 頁的『QueueManager 配置』

欄位	說明
type 字串	要使用的可用性類型。針對單一 Pod 使用 "SingleInstance"，Kubernetes 會自動重新啟動 (在某些情況下)。針對一對 Pods 使用 "MultiInstance"，其中一個是「作用中」佇列管理程式，另一個是待命。請參閱最新版本 IBM MQ 中的 <a href="#">儲存器中 IBM MQ 的高可用性</a> 。

## 授權

這些設定可控制您是否接受授權，以及要使用哪些授權標準。

出現在：

- 第 39 頁的『QueueManager 規格』

欄位	說明
use 字串	控制軟體使用方式的設定，其中授權支援多種用途。如需有效值，請參閱 <a href="https://ibm.biz/BdqvCF">https://ibm.biz/BdqvCF</a> 。
accept 布林值	您是否接受與此軟體相關聯的授權 (必要)。
license 字串	您接受的授權 ID。這必須是您使用之 MQ 版本的正確授權 ID。如需有效值，請參閱 <a href="https://ibm.biz/BdqvCF">https://ibm.biz/BdqvCF</a> 。
metric 字串	指定要使用的授權標準的設定。例如，"ProcessorValueUnit"、"VirtualProcessorCore" 或 "ManagedVirtualServer"。

## 限制

QueueManagerResourceList 定義 CPU 和記憶體設定。

出現在：

- 第 42 頁的『資源』

欄位	說明
cpu	
memory	

## LocalObject 參照

LocalObjectReference 包含足夠資訊，可讓您在相同名稱空間內找到所參照的物件。

出現在：

- [第 39 頁的『QueueManager 規格』](#)

欄位	說明
name 字串	參照的名稱。進一步資訊： <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> 待辦事項：新增其他有用欄位。apiVersion, kind, uid?。

## PKI

公開金鑰基礎架構設定，用於定義與傳輸層安全 (TLS) 或 MQ Advanced Message Security (AMS) 搭配使用的金鑰和憑證。

出現在：

- [第 39 頁的『QueueManager 規格』](#)

欄位	說明
keys <a href="#">第 37 頁的『PKISource』陣列</a>	要新增至佇列管理程式金鑰儲存庫的私密金鑰。
trust <a href="#">第 37 頁的『PKISource』陣列</a>	要新增至佇列管理程式金鑰儲存庫的憑證。

## PKISource

PKISource 定義「公開金鑰基礎架構」資訊的來源，例如金鑰或憑證。

出現在：

- [第 37 頁的『PKI』](#)

欄位	說明
name 字串	名稱用作金鑰或憑證的標籤。必須是小寫英數字串。
secret <a href="#">第 43 頁的『密碼』</a>	使用 Kubernetes 密鑰來提供金鑰。

## QueueManager 配置

QueueManagerConfig 可定義「佇列管理程式」儲存器及基礎「佇列管理程式」的設定。

出現在：

- [第 39 頁的『QueueManager 規格』](#)

欄位	說明
logFormat 字串	要用於此儲存器的日誌格式。對於儲存器中 JSON 格式的日誌，請使用 "JSON"。對於文字格式的訊息，請使用「基本」。
metrics <a href="#">第 38 頁的『QueueManager 度量值』</a>	Prometheus 樣式度量值的設定。
readinessProbe <a href="#">第 39 頁的『QueueManagerReadinessProbe』</a>	控制就緒性探測的設定。

欄位	說明
resources 第 42 頁的『資源』	控制資源需求的設定。
storage 第 41 頁的『QueueManager 儲存體』	用於控制佇列管理程式使用持續性磁區及儲存類別的儲存體設定。
availability 第 36 頁的『可用性』	佇列管理程式的可用性設定，例如是否使用作用中-待命配對。
imagePullPolicy 字串	控制 kubelet 何時嘗試取回指定映像檔的設定。
livenessProbe 第 38 頁的『QueueManagerLivenessProbe』	控制存活性探測的設定。
debug 布林值	是否將來自儲存器特定程式碼的除錯訊息記載至儲存器日誌。
image 字串	將使用的儲存器映像檔。
name 字串	基礎「MQ 佇列管理程式」的名稱 (如果不同於 metadata.name)。如果您想要的「佇列管理程式」名稱不符合名稱的 Kubernetes 規則 (例如，包含大寫字母的名稱)，請使用此欄位。

### QueueManagerLivenessProbe

控制存活性探測的設定。

出現在:

- 第 37 頁的『QueueManager 配置』

欄位	說明
failureThreshold 整數	在成功之後，將探測視為失敗的連續失敗次數下限。
initialDelaySeconds 整數	在儲存器啟動之後，起始活性探測之前的秒數。進一步資訊: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probe">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probe</a> 。
periodSeconds 整數	執行探測的頻率 (以秒為單位)。
successThreshold 整數	在失敗之後將探測視為成功的連續成功數下限。
timeoutSeconds 整數	探測逾時之前經歷的秒數。進一步資訊: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probe">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probe</a> 。

### QueueManager 度量值

Prometheus 樣式度量值的設定。

出現在:

- 第 37 頁的『QueueManager 配置』

欄位	說明
enabled 布林值	是否啟用 Prometheus 相容度量端點。

### QueueManagerOptionalVolume

MQ 回復日誌的 PersistentVolume 詳細資料。使用多重實例佇列管理程式時需要。

出現在:

- [第 41 頁的『QueueManager 儲存體』](#)

欄位	說明
class 字串	要用於此磁區的儲存類別。只有在 "type" 是 "persistent-claim" 時才有效。
enabled 布林值	此磁區是否應該啟用為個別磁區，或放置在預設 "queueManager" 磁區上。
size 字串	要傳遞至 Kubernetes 的 PersistentVolume 大小。只有在 "type" 是 "persistent-claim" 時才有效。
sizeLimit 字串	使用「暫時」磁區時的大小限制。檔案仍會寫入暫存目錄，因此您可以使用此選項來限制大小。僅當 type 為 "ephemeral" 時才有效。
type 字串	要使用的磁區類型。選擇 ephemeral 以建立非持續性 "emptyDir" 磁區，或選擇 persistent-claim 以使用持續性磁區。

## QueueManagerReadinessProbe

控制就緒性探測的設定。

出現在：

- [第 37 頁的『QueueManager 配置』](#)

欄位	說明
failureThreshold 整數	在成功之後，將探測視為失敗的連續失敗次數下限。
initialDelaySeconds 整數	在儲存器啟動之後，起始活性探測之前的秒數。進一步資訊： <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probe">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probe</a> 。
periodSeconds 整數	執行探測的頻率（以秒為單位）。
successThreshold 整數	在失敗之後將探測視為成功的連續成功數下限。
timeoutSeconds 整數	探測逾時之前經歷的秒數。進一步資訊： <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probe">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probe</a> 。

## QueueManager 規格

所需的 QueueManager 狀態。

出現在：

- [第 35 頁的『QueueManager』](#)

欄位	說明
license <a href="#">第 36 頁的『授權』</a>	這些設定可控制您是否接受授權，以及要使用哪些授權標準。
pki <a href="#">第 37 頁的『PKI』</a>	公開金鑰基礎架構設定，用於定義與傳輸層安全 (TLS) 或 MQ Advanced Message Security (AMS) 搭配使用的金鑰和憑證。
queueManager <a href="#">第 37 頁的『QueueManager 配置』</a>	QueueManagerConfig 可定義「佇列管理程式」儲存器及基礎「佇列管理程式」的設定。
securityContext <a href="#">第 43 頁的『SecurityContext』</a>	要新增至佇列管理程式 Pod 的 securityContext 的安全設定。
tracing <a href="#">第 44 頁的『TracingConfig』</a>	與 Cloud Pak for Integration 作業儀表板進行追蹤整合的設定。



欄位	說明
version 字串	控制將使用之 MQ 版本的設定 (必要)。例如: "9.1.5.0-r2" 會使用儲存器映像檔的第二個修訂版來指定 MQ 9.1.5.0 版。通常會在修訂中套用儲存器特定修正式，例如基本映像檔的修正式。
affinity	標準 Kubernetes 親緣性規則。如需相關資訊，請參閱 <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> 。
imagePullSecrets 第 37 頁的『LocalObject 參照』陣列	相同名稱空間中密鑰的選用參照清單，用於取回此 QueueManager 所使用的任何映像檔。如果指定的話，這些密鑰會傳遞給個別的拉取器實作，以供它們使用。例如，如果是 Docker，則只允許使用 DockerConfig 類型密碼。如需相關資訊，請參閱 <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> 。
template 第 43 頁的『範本』	Kubernetes 資源的進階範本。範本可讓使用者置換 IBM MQ 如何產生基礎 Kubernetes 資源，例如 StatefulSet、Pod 及服務。這僅適用於進階使用者，因為如果使用不正確，可能會中斷 MQ 的正常作業。範本中的設定將置換 QueueManager 資源中的任何其他指定值。
terminationGracePeriod Seconds 整數	Pod 需要溫和終止的選用持續時間 (以秒為單位)。值必須是非負整數。值零表示立即刪除。嘗試結束佇列管理程式的目標時間，會提升應用程式斷線的階段。必要的話，會中斷必要的佇列管理程式維護作業。
web 第 45 頁的『WebServer 配置』	MQ Web 伺服器的設定。

## QueueManagerStatus

QueueManager 的觀察狀態。

出現在：

- 第 35 頁的『QueueManager』

欄位	說明
endpoints 第 41 頁的『QueueManagerStatusEndpoint』陣列	此「佇列管理程式」公開之端點 (例如 API 或使用者介面端點) 的相關資訊。
name 字串	佇列管理程式的名稱。
versions 第 41 頁的『QueueManagerStatusVersion』	正在使用的 MQ 版本，以及 IBM Entitled Registry 中提供的其他版本。
adminUiUrl 字串	管理使用者介面的 URL。
conditions 第 40 頁的『QueueManagerStatusCondition』陣列	條件代表佇列管理程式狀態的最新可用觀察值。

## QueueManagerStatusCondition

QueueManagerStatusCondition 會定義「佇列管理程式」的條件。

出現在：

- 第 40 頁的『QueueManagerStatus』

欄位	說明
message 字串	人類可讀的訊息，指出前次轉移的詳細資料。



欄位	說明
type 字串	條件類型。
lastTransitionTime 字串	前次條件從某個狀態轉移至另一個狀態的時間。

### QueueManagerStatusEndpoint

QueueManagerStatusEndpoint 會定義 QueueManager 的端點。

出現在:

- [第 40 頁的『QueueManagerStatus』](#)

欄位	說明
name 字串	端點的名稱。
type 字串	端點的類型，例如使用者介面端點的 'UI'、API 端點的 'API'、API 文件的 'OpenAPI'。
uri 字串	端點的 URI。

### QueueManagerStatusVersion

正在使用的 MQ 版本，以及 IBM Entitled Registry 中提供的其他版本。

出現在:

- [第 40 頁的『QueueManagerStatus』](#)

欄位	說明
available <a href="#">第 41 頁的『QueueManagerStatusVersion 可用』</a>	IBM Entitled Registry 中提供其他版本的 MQ 。
reconciled 字串	正在使用 IBM MQ 的特定版本。如果指定自訂映像檔，則這可能不符合實際使用的 MQ 版本。

### QueueManagerStatusVersion 可用

IBM Entitled Registry 中提供其他版本的 MQ 。

出現在:

- [第 41 頁的『QueueManagerStatusVersion』](#)

欄位	說明
channels 陣列	可用於自動更新 MQ 版本的通道。
versions <a href="#">第 45 頁的『版本』</a> 陣列	可用的 MQ 特定版本。

### QueueManager 儲存體

用於控制佇列管理程式使用持續性磁區及儲存類別的儲存體設定。

出現在:

- [第 37 頁的『QueueManager 配置』](#)

欄位	說明
<a href="#">persistedData 第 38 頁的『QueueManagerOptionalVolume』</a>	MQ 持續保存資料 (包括配置、佇列及訊息) 的 PersistentVolume 詳細資料。使用多重實例佇列管理程式時需要。
<a href="#">queueManager 第 42 頁的『QueueManager 磁區』</a>	通常位於 /var/mqm 下之任何資料的預設 PersistentVolume。如果未指定其他磁區，則將包含所有持續保存的資料及回復日誌。
<a href="#">recoveryLogs 第 38 頁的『QueueManagerOptionalVolume』</a>	MQ 回復日誌的 PersistentVolume 詳細資料。使用多重實例佇列管理程式時需要。

## QueueManager 磁區

通常位於 /var/mqm 下之任何資料的預設 PersistentVolume。如果未指定其他磁區，則將包含所有持續保存的資料及回復日誌。

出現在:

- [第 41 頁的『QueueManager 儲存體』](#)

欄位	說明
class 字串	要用於此磁區的儲存類別。只有在 "type" 是 "persistent-claim" 時才有效。
size 字串	要傳遞至 Kubernetes 的 PersistentVolume 大小。只有在 "type" 是 "persistent-claim" 時才有效。
sizeLimit 字串	使用「暫時」磁區時的大小限制。檔案仍會寫入暫存目錄，因此您可以使用此選項來限制大小。僅當 type 為 "ephemeral" 時才有效。
type 字串	要使用的磁區類型。選擇 ephemeral 以建立非持續性 "emptyDir" 磁區，或選擇 persistent-claim 以使用持續性磁區。

## 要求

QueueManagerResourceList 定義 CPU 和記憶體設定。

出現在:

- [第 42 頁的『資源』](#)

欄位	說明
memory	
cpu	

## 資源

控制資源需求的設定。

出現在:

- [第 37 頁的『QueueManager 配置』](#)

欄位	說明
<a href="#">limits 第 36 頁的『限制』</a>	QueueManagerResourceList 定義 CPU 和記憶體設定。
<a href="#">requests 第 42 頁的『要求』</a>	QueueManagerResourceList 定義 CPU 和記憶體設定。

## 密碼

使用 Kubernetes 密鑰來提供金鑰。

出現在:

- [第 37 頁的『PKISource』](#)

欄位	說明
items 陣列	Kubernetes 密鑰內的金鑰，應該新增至「佇列管理程式」儲存器。
secretName 字串	Kubernetes 密鑰的名稱。

## SecurityContext

要新增至佇列管理程式 Pod 的 securityContext 的安全設定。

出現在:

- [第 39 頁的『QueueManager 規格』](#)

欄位	說明
supplementalGroups 陣列	除了容器的主要 GID 之外，還在每一個容器中執行的第一個處理程序所套用的群組清單。如果未指定，則不會將任何群組新增至任何儲存器。
fsGroup 整數	套用至 Pod 中所有容器的特殊增補群組。部分磁區類型容許 Kubelet 將該磁區的所有權變更為 Pod 所擁有: 1. 擁有的 GID 將是 FSGroup 2. setgid 位元已設定 (在磁區中建立的新檔案將由 FSGroup 擁有) 3. 許可權位元是 OR 'd with rw-rw---- 如果取消設定， Kubelet 將不會修改任何磁區的所有權和許可權。
initVolumeAsRoot 布林值	這會影響起始設定 PersistentVolume 的儲存器所使用的 securityContext。如果您使用的儲存體提供者需要您是 root 使用者才能存取新佈建的磁區，請將此設為 "true"。將此設為 "true" 會影響您可以使用的「安全環境定義限制 (SCC)」物件，如果您未獲授權使用容許 root 使用者的 SCC，則「佇列管理程式」可能無法啟動。如需相關資訊，請參閱 <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> 。

## 範本

Kubernetes 資源的進階範本。範本可讓使用者置換 IBM MQ 如何產生基礎 Kubernetes 資源，例如 StatefulSet、Pod 及服務。這僅適用於進階使用者，因為如果使用不正確，可能會中斷 MQ 的正常作業。範本中的設定將置換 QueueManager 資源中的任何其他指定值。

出現在:

- [第 39 頁的『QueueManager 規格』](#)

欄位	說明
pod	置換用於 Pod 的範本。請參閱 <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> 。

## TracingAgent

僅在 Cloud Pak for Integration 中，您可以配置選用 Tracing Agent 的設定。

出現在:

- [第 44 頁的『TracingConfig』](#)

欄位	說明
image 字串	將使用的儲存器映像檔。

欄位	說明
imagePullPolicy 字串	控制 kubelet 何時嘗試取回指定映像檔的設定。
livenessProbe 第 44 頁的『TracingProbe』	控制存活性探測的設定。
readinessProbe 第 44 頁的『TracingProbe』	控制就緒性探測的設定。

## TracingCollector

僅在 Cloud Pak for Integration 中，您可以配置選用「追蹤收集器」的設定。

出現在：

- 第 44 頁的『TracingConfig』

欄位	說明
image 字串	將使用的儲存器映像檔。
imagePullPolicy 字串	控制 kubelet 何時嘗試取回指定映像檔的設定。
livenessProbe 第 44 頁的『TracingProbe』	控制存活性探測的設定。
readinessProbe 第 44 頁的『TracingProbe』	控制就緒性探測的設定。

## TracingConfig

與 Cloud Pak for Integration 作業儀表板進行追蹤整合的設定。

出現在：

- 第 39 頁的『QueueManager 規格』

欄位	說明
agent 第 43 頁的『TracingAgent』	僅在 Cloud Pak for Integration 中，您可以配置選用 Tracing Agent 的設定。
collector 第 44 頁的『TracingCollector』	僅在 Cloud Pak for Integration 中，您可以配置選用「追蹤收集器」的設定。
enabled 布林值	是否透過追蹤來啟用與 Cloud Pak for Integration 作業儀表板的整合。
namespace 字串	Cloud Pak for Integration 作業儀表板安裝所在的名稱空間。

## TracingProbe

控制就緒性探測的設定。

出現在：

- 第 44 頁的『TracingCollector』

欄位	說明
failureThreshold 整數	在成功之後，將探測視為失敗的連續失敗次數下限。
initialDelaySeconds 整數	在儲存器啟動之後，起始活性探測之前的秒數。進一步資訊： <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probe">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probe</a> 。

欄位	說明
periodSeconds 整數	執行探測的頻率（以秒為單位）。
successThreshold 整數	在失敗之後將探測視為成功的連續成功數下限。
timeoutSeconds 整數	探測逾時之前經歷的秒數。進一步資訊: <a href="https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probe">https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probe</a> 。

## 版本

QueueManagerStatusVersion 會定義 MQ 的版本。

出現在:

- [第 41 頁的『QueueManagerStatusVersion 可用』](#)

欄位	說明
name 字串	此版本 QueueManager 的版本 "name"。這些是 spec.version 欄位的有效值。

## WebServer 配置

MQ Web 伺服器的設定。

出現在:

- [第 39 頁的『QueueManager 規格』](#)

欄位	說明
enabled 布林值	是否啟用 Web 伺服器。





## 注意事項

本資訊係針對 IBM 在美國所提供之產品與服務所開發。

在其他國家中，IBM 可能不會提供本書中所提的各項產品、服務或功能。請洽當地 IBM 業務代表，以取得當地目前提供的產品和服務之相關資訊。這份文件在提及 IBM 的產品、程式或服務時，不表示或暗示只能使用 IBM 的產品、程式或服務。只要未侵犯 IBM 的智慧財產權，任何功能相當的產品、程式或服務都可以取代 IBM 的產品、程式或服務。不過，任何非 IBM 的產品、程式或服務，使用者必須自行負責作業的評估和驗證責任。

本文件所說明之主題內容，IBM 可能擁有其專利或專利申請案。提供本文件不代表提供這些專利的授權。您可以書面提出授權查詢，來函請寄到：

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

如果是有關雙位元組 (DBCS) 資訊的授權查詢，請洽詢所在國的 IBM 智慧財產部門，或書面提出授權查詢，來函請寄到：

智慧財產權授權  
法務部與智慧財產權法律  
IBM Japan, Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japan

**下列段落不適用於英國，若與任何其他國家之法律條款抵觸，亦不適用於該國：** International Business Machines Corporation 只依 "現況" 提供本出版品，不提供任何明示或默示之保證，其中包括且不限於不侵權、可商用性或特定目的之適用性的隱含保證。有些地區在特定交易上，不允許排除明示或暗示的保證，因此，這項聲明不一定適合您。

這項資訊中可能有技術上或排版印刷上的訛誤。因此，IBM 會定期修訂；並將修訂後的內容納入新版中。IBM 隨時會改進及/或變更本出版品所提及的產品及/或程式，不另行通知。

本資訊中任何對非 IBM 網站的敘述僅供參考，IBM 對該網站並不提供任何保證。這些網站所提供的資料不是 IBM 本產品的資料內容，如果要使用這些網站的資料，您必須自行承擔風險。

IBM 得以各種適當的方式使用或散布由您提供的任何資訊，無需對您負責。

如果本程式的獲授權人為了 (i) 在個別建立的程式和其他程式（包括本程式）之間交換資訊，以及 (ii) 相互使用所交換的資訊，因而需要相關的資訊，請洽詢：

IBM Corporation  
軟體交互作業能力協調程式，部門 49XA  
3605 公路 52 N  
Rochester, MN 55901  
U.S.A.

在適當條款與條件之下，包括某些情況下（支付費用），或可使用此類資訊。

IBM 基於雙方之 IBM 客戶合約、IBM 國際程式授權合約或任何同等合約之條款，提供本資訊所提及的授權程式與其所有適用的授權資料。

本文件中所含的任何效能資料都是在受管制的環境下判定。因此不同作業環境之下所得的結果，可能會有很大的差異。有些測定已在開發階段系統上做過，不過這並不保證在一般系統上會出現相同結果。甚至有部分的測量，是利用插補法而得的估計值，實際結果可能有所不同。本書的使用者應依自己的特定環境，查證適用的資料。

本文件所提及之非 IBM 產品資訊，取自產品的供應商，或其發佈的聲明或其他公開管道。IBM 並未測試過這些產品，也無法確認這些非 IBM 產品的執行效能、相容性或任何對產品的其他主張是否完全無誤。有關非 IBM 產品的性能問題應直接洽詢該產品供應商。

有關 IBM 未來方針或目的之所有聲明，僅代表 IBM 的目標與主旨，隨時可能變更或撤銷，不必另行通知。

這份資訊含有日常商業運作所用的資料和報告範例。為了要使它們儘可能完整，範例包括個人、公司、品牌和產品的名稱。這些名稱全屬虛構，如與實際公司的名稱和住址雷同，純屬巧合。

著作權授權：

本資訊含有原始語言之範例應用程式，用以說明各作業平台中之程式設計技術。您可以基於研發、使用、銷售或散布符合作業平台（撰寫範例程式的作業平台）之應用程式介面的應用程式等目的，以任何形式複製、修改及散布這些範例程式，而不必向 IBM 付費。這些範例並未在所有情況下完整測試。因此，IBM 不會保證或暗示這些程式的穩定性、服務能力或功能。

若貴客戶正在閱讀本項資訊的電子檔，可能不會有照片和彩色說明。

## 程式設計介面資訊

---

程式設計介面資訊 (如果有提供的話) 旨在協助您建立與此程式搭配使用的應用軟體。

本書包含預期程式設計介面的相關資訊，可讓客戶撰寫程式以取得 WebSphere MQ 的服務。

不過，本資訊也可能包含診斷、修正和調整資訊。提供診斷、修正和調整資訊，是要協助您進行應用軟體的除錯。

**重要：**請勿使用此診斷、修改及調整資訊作為程式設計介面，因為它可能會變更。

## 商標

---

IBM、IBM 標誌 [ibm.com](http://www.ibm.com) 是 IBM Corporation 在全球許多適用範圍的商標。IBM 商標的最新清單可在 Web 的 "Copyright and trademark information" [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml) 中找到。其他產品和服務名稱，可能是 IBM 或其他公司的商標。

Microsoft 及 Windows 是 Microsoft Corporation 在美國及/或其他國家或地區的商標。

UNIX 是 The Open Group 在美國及/或其他國家/地區的註冊商標。

Linux 是 Linus Torvalds 在美國及/或其他國家或地區的註冊商標。

本產品包含 Eclipse Project (<http://www.eclipse.org/>) 所開發的軟體。

Java 和所有以 Java 為基礎的商標及標誌是 Oracle 及/或其子公司的商標或註冊商標。





產品編號:

(1P) P/N: