9.3

儲存器中的 *IBM MQ*

附註

使用本資訊及其支援的產品之前，請先閱讀第 197 頁的『注意事項』中的資訊。

目錄

Multi IBM MQ 在儲存器及 IBM Cloud Pak for Integration 中

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 Red Hat® OpenShift®上使用 IBM MQ Operator 來執行 IBM MQ 。 這可以使用 IBM Cloud Pak for Integration、 IBM MQ Advanced 或 IBM MQ Advanced for Developers 來完成。

您也可以在自己建置的容器中執行 IBM MQ 。

MQ Adv. CD 如需 IBM MQ Operator 的相關資訊，請參閱下列鏈結。

Multi 規劃儲存器中的 IBM MQ

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/Red Hat 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 6 頁的『在儲存器中支援 IBM MQ』.
- 第 19 頁的『IBM MQ Operator 的儲存體考量』.
- 第 22 頁的『儲存器中 IBM MQ 的高可用性』.
- 第 24 頁的『儲存器中 IBM MQ 的災難回復』.
- 第 25 頁的『儲存器中 IBM MQ 的使用者鑑別和授權』.

選擇您要如何在儲存器中使用 IBM MQ

在容器中使用 IBM MQ 有多個選項: 您可以選擇使用 IBM MQ Operator，它會使用預先包裝的容器映像檔，或者您可以建置自己的映像檔及部署程式碼。

使用 IBM MQ Operator

OpenShift CP4I

如果您計劃在 Red Hat OpenShift Container Platform 上部署，則可能想要使用 IBM MQ Operator。

IBM MQ Operator 延伸 Red Hat OpenShift Container Platform API 以新增 `QueueManager` 自訂資源。 操作員監看新的佇列管理程式定義，然後將它們轉換成必要的低階資源，例如 `StatefulSet` 和 `Service` 資源。 如果是原生 HA，操作員也可以執行佇列管理程式實例的複式漸進式更新。 請參閱 第 194 頁的『執行您自己的原生 HA 佇列管理程式漸進式更新的考量』

使用 IBM MQ Operator 時，不支援部分 IBM MQ 特性。 如需使用 IBM MQ Operator 時支援的項目的詳細資料，請參閱 第 6 頁的『在儲存器中支援 IBM MQ』 。

請注意， IBM MQ Operator 不支援在具有多架構計算機器的 OpenShift 叢集上安裝。

建置您自己的映像檔及部署程式碼

Multi

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 187 頁的『建置您自己的 IBM MQ 容器及部署程式碼』。

如需建置您自己的映像檔及部署程式碼時所支援內容的詳細資料，請參閱 第 6 頁的『在儲存器中支援 IBM MQ』 。

相關參考

第 6 頁的『在儲存器中支援 IBM MQ』
並非所有 IBM MQ 特性都以相同方式在儲存器中提供及支援。

OpenShift CP4I-LTS CP4I CD

在儲存器中支援 IBM MQ

並非所有 IBM MQ 特性都以相同方式在儲存器中提供及支援。

下面是一個表格，詳細顯示 IBM MQ Operator 如何支援 IBM MQ 特性，或當您建置自己的儲存器及部署程式碼時如何支援。

註: 只有在與 IBM MQ Operator 搭配使用時，才支援 IBM Container Registry 上預先建置的 IBM MQ 儲存器映像檔 (icr.io 及 cp.icr.io) ，且適用於修正程式。

無法將預先建置 IBM MQ Advanced for Developers 映像檔的授權「升級」至不同的授權。 IBM MQ Operator 將部署不同的映像檔，視選取的授權而定。

在此表格中，適用下列術語:

"儲存器啟用碼"
執行檔 **runmqserver**、 **runmqintegrationserver**、 **chkmqhealthy**、 **chkmqready** 和 **chkmqstarted**。 此程式碼作為範例提供，且只有在與 IBM MQ Operator 搭配使用時才支援作為預先建置儲存器的一部分。

	使用 IBM MQ Operator 和 IBM Cloud Pak for Integration 授權	使用 IBM MQ Operator 和 IBM MQ Advanced 授權	使用 IBM MQ Operator 和 IBM MQ Advanced for Developers 授權	預先建置 IBM MQ Advanced for Developers 映像檔	建置-您自己的容器
支援的平台	僅在 Red Hat OpenShift Container Platform 上受支援。 一旦 Red Hat 停止支援， IBM MQ 就不再支援 Red Hat OpenShift Container Platform 的版次。 如需詳細資料，請參閱 第 10 頁的『IBM MQ Operator 的版本支援』。		僅適用於 Red Hat OpenShift Container Platform ， 但不受支援。	在任何 Docker、containerd 或 cri-o 平台上工作，但不受支援。 如需詳細資料，請參閱 IBM MQ 的系統需求。	任何 Docker、containerd 或 cri-o platform。 如需詳細資料，請參閱 IBM MQ 的系統需求。 原生 HA 僅在 Kubernetes 或 Red Hat OpenShift Container Platform 上受支援。 範例儲存器映像檔使用 Red Hat Universal Base Image (UBI) ，其中包括 IBM MQ 所使用的 Linux® 程式庫及公用程式。 在 Red Hat OpenShift 上執行時， Red Hat 支援 UBI。 不支援 儲存器啟用碼 。
CPU 架構	在 amd64、 s390x z/Linux 及 ppc64le Power Systems 上受支援。		在 amd64、 s390x z/Linux 及 ppc64le Power Systems 上可用，但不受支援。		根據 IBM MQ 軟體。

	使用 IBM MQ Operator 和 IBM Cloud Pak for Integration 授權	使用 IBM MQ Operator 和 IBM MQ Advanced 授權	使用 IBM MQ Operator 和 IBM MQ Advanced for Developers 授權	預先建置 IBM MQ Advanced for Developers 映像檔	建置-您自己的容器
支援持續時間	IBM Cloud Pak for Integration - Long Term Support 或 Continuous Delivery。[1] 在下一個 IBM Cloud Pak for Integration CD 或 CP4I-LTS 版次之前，都支援 CD 操作員及佇列管理程式。 在下一個 IBM Cloud Pak for Integration CP4I-LTS 版本之前，支援 CP4I-LTS 操作員及佇列管理程式，外加容許升級的寬限期。	僅限 Continuous Delivery 串流，適用於 IBM MQ Operator 及佇列管理程式。 只有在下一個 CD 或 LTS 版次之前，才支援每一個 IBM MQ Operator 及佇列管理程式版本。	不支援		根據 IBM MQ 軟體。 請參閱 IBM MQ 長期支援及 Continuous Delivery 版本的常見問題 (FAQ)。 不支援 儲存器啟用碼 。
安全修正程式可用性	在 IBM Container Registry 上可作為容器映像檔使用的定期修正程式				IBM MQ 軟體的修正程式在 Fix Central 上以軟體形式提供。 不支援 儲存器啟用碼 。
臨時修正程式可用性	佇列管理程式修正程式可作為軟體使用，且需要自訂映像檔建置。 IBM MQ Operator 修正程式無法作為臨時修正程式使用。		沒有可用的臨時修正程式。		IBM MQ 軟體的修正程式在 Fix Central 上以軟體形式提供，或透過 IBM 支援中心提供。 不支援 儲存器啟用碼 。

[1] 支援 IBM MQ Operator 作為 IBM MQ CD 版本或 CP4I-LTS 版本:

- 隨 IBM MQ Operator 2.0.x 一起部署的 IBM MQ 9.3.0.x 儲存器映像檔 (當用作 IBM Cloud Pak for Integration 2022.2.1 的一部分時) 適用於 CP4I-LTS 支援。 IBM MQ Operator 的最新 Long Term Support (LTS) 版本是 2.0.23，而最新 LTS 儲存器映像檔是 9.3.0.17-r3。
- 隨 IBM MQ Operator 3.1.x 一起部署的 IBM MQ 9.3.5 儲存器映像檔 (當用作 IBM Cloud Pak for Integration 2023.4.1 的一部分時) 適用於 CD 支援。 IBM MQ Operator 的最新 Continuous Delivery (CD) 版本是 3.1.3，而最新 CD 儲存器映像檔是 9.3.5.1-r2。

	使用 IBM MQ Operator 和 IBM Cloud Pak for Integration 授權	使用 IBM MQ Operator 和 IBM MQ Advanced 授權	使用 IBM MQ Operator 和 IBM MQ Advanced for Developers 授權	預先建置 IBM MQ Advanced for Developers 映像檔	建置-您自己的容器
特性: Advanced Message Security	受支援。 請注意，使用伺服器端加密並不容易，因為 IBM MQ Operator 不直接容許您為 Advanced Message Security 指定自己的金鑰儲存庫。		可用但不受支援。		根據 IBM MQ 軟體支援，但沒有可用的範例。
特性: Managed File Transfer	無法使用且不受支援。 不過，您可以使用 IBM MQ Operator 來提供一或多個「協調」、「指令」或「代理程式」佇列管理程式。			無法使用且不受支援。	根據 IBM MQ 軟體，支援代理程式的 sample 。
特性: MQTT	無法使用且不受支援。				根據 IBM MQ 軟體支援，但沒有可用的範例。
特性 :AMQP	無法使用且不受支援。				根據 IBM MQ 軟體支援，但沒有可用的範例。
特性: REST API	從 IBM MQ Operator 3.0 及 IBM MQ 9.3.4 開始提供並支援。 在此之前，不支援 REST API。	可用且受支援。 從 IBM MQ Operator 3.0 及 IBM MQ 9.3.4 開始即可輕鬆配置。	可從 IBM MQ Operator 3.0 及 IBM MQ 9.3.4 以上版本取得並支援，但不受支援。 在此之前， REST API 無法使用。	從 IBM MQ 9.3.4 開始提供且受支援，但不受支援。 在此之前， REST API 無法使用。	根據 IBM MQ 軟體提供並支援。
特性: 抄寫的資料佇列管理程式	無法使用且不受支援。 抄寫的「資料佇列管理程式」與 Linux 核心緊密連結，且在儲存器中不受支援。				
特性: 原生 HA	可用且受支援。		可用，但不受支援。		僅適用於 Kubernetes 和 Red Hat OpenShift Container Platform。 根據 IBM MQ 軟體支援。
特性: 多重實例佇列管理程式	可用且受支援。		可用，但不受支援。		根據 IBM MQ 軟體提供並支援。
特性: 回復日誌的類型	僅限循環式記載或抄寫的日誌。 不支援線性記載。				根據 IBM MQ 軟體提供並支援。 您需要配置 **crtmqm** 選項。
特性: 指定 crtmqdir、crtmqm、strmqm 和 endmqm 的自訂指令行選項	無法使用且不受支援。 大部分選項都可以使用 INI 檔案來配置，但部分選項無法配置，例如使用線性記載。				選用，視您如何實作 容器啟用碼而定。

	使用 IBM MQ Operator 和 IBM Cloud Pak for Integration 授權	使用 IBM MQ Operator 和 IBM MQ Advanced 授權	使用 IBM MQ Operator 和 IBM MQ Advanced for Developers 授權	預先建置 IBM MQ Advanced for Developers 映像檔	建置-您自己的容器
特性: 作業系統 (OS) 使用者	無法使用且不受支援。				根據 IBM MQ 軟體，如果您使用 RPM 來安裝 IBM MQ，但沒有可用的範例，則可能且受支援。 由於安全風險而不建議使用。
特性: IBM MQ Bridge to blockchain	無法使用且不受支援。 從 IBM MQ 9.3.2 開始，完全從 IBM MQ 移除。				
特性: IBM MQ Bridge to Salesforce	無法使用且不受支援。				根據 IBM MQ 軟體支援，但從 IBM MQ 9.3.1 開始已淘汰。

註: 「根據 IBM MQ 軟體支援」詞組表示 IBM 技術支援僅限於在儲存器內執行的核心 IBM MQ 軟體。

相關概念

IBM MQ 長期支援及持續交付版次的常見問題

相關參考

IBM Cloud Pak for Integration 軟體支援中心生命週期附錄

OpenShift CP4I-LTS CP4I CD IBM MQ Operator 的版本支援

IBM MQ、 OpenShift Container Platform 和 IBM Cloud Pak for Integration 受支援版本之間的對映。

註:

IBM MQ Operator 僅支援 OpenShift Container Platform 的 Extended Update Support (EUS) 版本。 如需此包括哪些發行的相關資訊，請參閱「 Red Hat OpenShift Container Platform 生命週期原則」網頁上的 生命週期階段 。

- 第 10 頁的『可用的 IBM MQ 版本』
- 第 11 頁的『相容的 Red Hat OpenShift Container Platform 版本』
- 第 12 頁的『IBM Cloud Pak for Integration 版本』
- 第 12 頁的『舊版運算子中的可用 IBM MQ 版本』
- 第 12 頁的『舊版運算子的相容 OpenShift Container Platform 版本』

可用的 IBM MQ 版本

操作員通道	Operator 版本	IBM MQ 版本									
		9.2.0 EUS	9.2.3	9.2.4	9.2.5	9.3.0	9.3.1	9.3.2	9.3.3	9.3.4	9.3.5
v2.0	2.0	→	⚠	●	●	●□					
v2.1	2.1	→	⚠	⚠	⚠	→	●				

操作員通道	Operator 版本	IBM MQ 版本									
		9.2.0 EUS	9.2.3	9.2.4	9.2.5	9.3.0	9.3.1	9.3.2	9.3.3	9.3.4	9.3.5
v2.2	2.2	→	⚠	⚠	⚠	→	●				
v2.3	2.3	→	⚠	⚠	⚠	→	⚠	●			
v2.4	2.4	→	⚠	⚠	⚠	→	⚠	⚠	●		
v3.0	3.0					→	⚠	⚠	⚠	●	
v3.1	3.1					→	⚠	⚠	⚠	⚠	●

索引鍵:

●
可用的 Continuous Delivery 支援

▯
IBM Cloud Pak for Integration - Long Term Support 可用的

→
僅在從 IBM Cloud Pak for Integration - Long Term Support 運算元移轉至 Continuous Delivery 運算元期間可用。

⚠
Deprecated當 IBM MQ 版本不再支援時，它們可能仍可在操作器中配置，但不再符合支援資格，並可能在未來版本中移除。

如需每一個版本的完整資料，包括每一個版本中的詳細特性、變更及修正程式，請參閱 第 30 頁的『IBM MQ Operator 的發行歷程』 。

相容的 Red Hat OpenShift Container Platform 版本

操作員通道	Operator 版本	OpenShift Container Platform 版本 [2]		
		4.10	4.12	4.14
v2.0	2.0.0-2.0.15	→	▯	
	2.0.16		▯	
	2.0.17 及以上版本		▯	▯
v2.1	2.1	→	●	
v2.2	2.2	→	●	
v2.3	2.3	→	●	
v2.4	2.4.0-2.4.3	→	●	
	2.4.4		●	
	2.4.5 及更新版本		●	●
v3.0	3.0.0 以上		●	●
v3.1	3.1.0 以上		●	●

索引鍵:

[2] OpenShift Container Platform 版本取決於其自己的支援日期。 如需相關資訊，請參閱 OpenShift Container Platform 生命週期原則 。

●
可用的 Continuous Delivery 支援

▯
IBM Cloud Pak for Integration - Long Term Support 可用的

→
不再支援。 請移轉至更新的 OpenShift Container Platform 版本。

IBM Cloud Pak for Integration 版本

支援用作 IBM Cloud Pak for Integration 2022.2.1 版的一部分，或單獨使用:

- IBM MQ Operator 2.0.x
- IBM MQ Operator 2.1.x

支援用作 IBM Cloud Pak for Integration 2022.4.1 版的一部分，或獨立使用:

- IBM MQ Operator 2.2.x
- IBM MQ Operator 2.3.x

支援用作 IBM Cloud Pak for Integration 2023.2.1 版的一部分，或單獨使用:

- IBM MQ Operator 2.4.x

支援用作 IBM Cloud Pak for Integration 2023.4.1 版的一部分，或單獨使用:

- IBM MQ Operator 3.0.x
- IBM MQ Operator 3.1.x

舊版運算子中的可用 IBM MQ 版本

請參閱 IBM MQ 9.2 說明文件中的 可用的 IBM MQ 版本 。

舊版運算子的相容 OpenShift Container Platform 版本

請參閱 IBM MQ 9.2 文件中的 相容 OpenShift Container Platform 版本 。

規劃儲存器中的 IBM MQ 授權

儲存器授權可讓您只授權個別 IBM MQ 儲存器的可用容量，而不需要您授權執行儲存器的整個伺服器。 若要利用儲存器授權，必須使用 IBM License Service 來追蹤授權使用情況，並判定您需要的授權。

相關資訊

IBM 儲存器授權
儲存器授權常見問題 (FAQ)
安裝 License Service
檢視及追蹤授權使用情況

Linux 建置您自己的 IBM MQ 儲存器映像檔時的授權註釋

授權註釋可讓您根據儲存器上定義的限制 (而非基礎機器上定義的限制) 來追蹤使用情形。 您可以將用戶端配置為使用特定註釋來部署儲存器，然後 IBM License Service 會使用這些註釋來追蹤使用情形。

部署自行建置 IBM MQ 儲存器映像檔時，有兩種常見的授權方法:

- 授權執行儲存器的整個機器。
- 根據相關聯的限制來授權儲存器。

這兩個選項都可供用戶端使用，您可以在 Passport Advantage 上的 IBM 儲存器授權」頁面 中找到進一步詳細資料。

如果要根據儲存器限制來授權 IBM MQ 儲存器，則需要安裝 IBM License Service 以追蹤使用情形。 如需受支援環境及安裝指示的相關資訊，請參閱 GitHub 上的 ibm-licensing-operator 頁面。

IBM License Service 安裝在部署 IBM MQ 儲存器的 Kubernetes 叢集上，並使用 Pod 註釋來追蹤使用情形。因此，用戶端需要使用 IBM License Service 隨後使用的特定註釋來部署 Pod。 根據您在儲存器內部署的授權及功能，使用下列一或多個註釋。

註: 許多註釋包含下列其中一行或兩行:

```
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productMetric: "PROCESSOR_VALUE_UNIT" |  "VIRTUAL_PROCESSOR_CORE"
```

在使用註釋之前，您必須先編輯下列行:

- 對於 `productChargedContainers`，您必須選擇 `"All"`，或替換儲存器的實際名稱。
- 對於 `productMetric`，您必須選擇所提供的其中一個值。

與 IBM MQ 產品授權搭配使用的註釋

如果您具有 IBM MQ 產品授權，請在下面選取與您已購買且想要使用的授權相符的註釋。

- 第 14 頁的『IBM MQ』
- 第 15 頁的『IBM MQ 進階』
- 第 15 頁的『IBM MQ for Non-Production Environment』
- 第 15 頁的『IBM MQ Advanced for Non-Production Environment』
- 第 15 頁的『IBM MQ Advanced for Developers』

與「IBM MQ 多重實例高可用性」配置搭配使用的 IBM MQ 註釋如下。 另請參閱第 13 頁的『選取高可用性配置的正確註釋』。

- 第 15 頁的『IBM MQ 儲存器多重實例』
- 第 15 頁的『IBM MQ 進階儲存器多重實例』
- 第 15 頁的『IBM MQ Container Multi Instance for Non-Production Environment』
- 第 16 頁的『IBM MQ Advanced Container Multi Instance for Non-Production Environment』

與 CP4I 產品授權搭配使用的註釋

如果您具有 IBM Cloud Pak for Integration (CP4I) 授權，請選取下面符合您已購買且想要使用的授權的註釋。

- 第 16 頁的『具有 CP4I 授權的 IBM MQ』
- 第 16 頁的『IBM MQ Advanced with CP4I 授權』
- 第 16 頁的『IBM MQ for Non-Production Environment with CP4I 授權』
- 第 16 頁的『IBM MQ Advanced for Non-Production Environment with CP4I 授權』

與「IBM MQ 多重實例高可用性」配置搭配使用的 CP4I 註釋如下。 另請參閱第 13 頁的『選取高可用性配置的正確註釋』。

- 第 16 頁的『IBM MQ Container Multi Instance with CP4I 授權』
- 第 17 頁的『IBM MQ Advanced Container Multi Instance with CP4I 授權』
- 第 17 頁的『IBM MQ Container Multi Instance for Non-Production Environment with CP4I 授權』
- 第 17 頁的『IBM MQ Advanced Container Multi Instance for Non-Production Environment，含 CP4I 授權』

選取高可用性配置的正確註釋

IBM MQ 多重實例

當您在 IBM MQ 多重實例高可用性配置中部署一對佇列管理程式時，您應該在兩個實例上使用相同的註釋。視購買的授權而定，應該選取下列其中一個註釋:

- IBM MQ 或 IBM MQ Advanced 獨立式授權

  - 第 15 頁的『IBM MQ 儲存器多重實例』
  - 第 15 頁的『IBM MQ 進階儲存器多重實例』
  - 第 15 頁的『IBM MQ Container Multi Instance for Non-Production Environment』
  - 第 16 頁的『IBM MQ Advanced Container Multi Instance for Non-Production Environment』
- IBM Cloud Pak for Integration 授權 (entitlement)
  - 第 16 頁的『IBM MQ Container Multi Instance with CP4I 授權』
  - 第 17 頁的『IBM MQ Advanced Container Multi Instance with CP4I 授權』
  - 第 17 頁的『IBM MQ Container Multi Instance for Non-Production Environment with CP4I 授權』
  - 第 17 頁的『IBM MQ Advanced Container Multi Instance for Non-Production Environment , 含 CP4I 授權』

與 IBM Cloud Pak for Integration 授權搭配使用時，註釋中的授權比例可確保記錄正確的授權耗用。 與獨立式 IBM MQ 或 IBM MQ Advanced 授權搭配使用時，每個實例的 License Service 中所報告的註釋需要對映至 IBM MQ 授權組件，如下所示:

- IBM MQ Advanced container 多重實例
  - 1 x IBM MQ Advanced **及** 1 x IBM MQ Advanced 高可用性抄本 **或**
  - 2 x IBM MQ Advanced[3]
- IBM MQ Advanced container 適用於非正式作業環境的多重實例
  - 1 x IBM MQ Advanced **及** 1 x IBM MQ Advanced 高可用性抄本 **或**
  - 2 x IBM MQ Advanced 適用於非正式作業環境)[3]
- IBM MQ 儲存器多重實例
  - 1 x IBM MQ **及** 1 x IBM MQ 高可用性抄本 **或**
  - 2 x IBM MQ[3]
- IBM MQ Container Multi Instance for Non-Production Environment
  - 1 x IBM MQ **及** 1 x IBM MQ 高可用性抄本 **或**
  - 2 x IBM MQ 適用於非正式作業環境) [3]

IBM MQ 原生 HA

如果您在「原生 HA 仲裁」中部署三個佇列管理程式，則只有作用中實例會耗用授權。 所有實例都應該具有相同的註釋。 視所購買的授權而定，應該選取下列其中一項:

- IBM MQ 或 IBM MQ Advanced 獨立式授權
  - 第 15 頁的『IBM MQ 進階』
  - 第 15 頁的『IBM MQ Advanced for Non-Production Environment』
- IBM Cloud Pak for Integration 授權 (entitlement)
  - 第 16 頁的『IBM MQ Advanced with CP4I 授權』
  - 第 16 頁的『IBM MQ Advanced for Non-Production Environment with CP4I 授權』

註釋

本主題的其餘部分詳述每一個註釋的內容。

IBM MQ

```
productID: "c661609261d5471fb4ff8970a36bccea"
productName: "IBM MQ"
```

[3] 此授權選項不是最佳選項，僅應在沒有相關「高可用性抄本」部分的授權可用時使用。

```
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

IBM MQ 進階

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

IBM MQ for Non-Production Environment

```
productID: "151bec68564a4a47a14e6fa99266deff"
productName: "IBM MQ for Non-Production Environment"
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

IBM MQ Advanced for Non-Production Environment

```
productID: "21dfe9a0f00f444f888756d835334909"
productName: "IBM MQ Advanced for Non-Production Environment"
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

IBM MQ Advanced for Developers

```
productID: "2f886a3eefbe4ccb89b2adb97c78b9cb"
productName: "IBM MQ Advanced for Developers (Non-Warranted)"
productMetric: "FREE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

IBM MQ 儲存器多重實例

```
productID: "2dea73b866b648b6b4abe2a85eb76964"
productName: "IBM MQ Container Multi Instance"
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

IBM MQ 進階儲存器多重實例

```
productID: "bd35bff411bb47c2a3f3a4590f33a8ef"
productName: "IBM MQ Advanced Container Multi Instance"
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

IBM MQ Container Multi Instance for Non-Production Environment

```
productID: "af11b093f16a4a26806013712b860b60"
productName: "IBM MQ Container Multi Instance for Non-Production Environment"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

IBM MQ Advanced Container Multi Instance for Non-Production Environment

```
productID: "31f844f7a96b49749130cd0708fdbb17"
productName: "IBM MQ Advanced Container Multi Instance for Non-Production Environment"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

具有 CP4I 授權的 IBM MQ

```
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
cloudpakName: "IBM Cloud Pak for Integration"
productID: "c661609261d5471fb4ff8970a36bccea"
productName: "IBM MQ"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productCloudpakRatio: "4:1"
```

IBM MQ Advanced with CP4I 授權

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

IBM MQ for Non-Production Environment with CP4I 授權

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

IBM MQ Advanced for Non-Production Environment with CP4I 授權

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

IBM MQ Container Multi Instance with CP4I 授權

```
productName: "IBM MQ Container Multi Instance"
productID: "2dea73b866b648b6b4abe2a85eb76964"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productCloudpakRatio: "10:3"
cloudpakName: "IBM Cloud Pak for Integration"
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

IBM MQ Advanced Container Multi Instance with CP4I 授權

```
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
cloudpakName: "IBM Cloud Pak for Integration"
productID: "bd35bff411bb47c2a3f3a4590f33a8ef"
productName: "IBM MQ Advanced Container Multi Instance"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productCloudpakRatio: "5:3"
```

IBM MQ Container Multi Instance for Non-Production Environment with CP4I 授權

```
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
cloudpakName: "IBM Cloud Pak for Integration"
productID: "af11b093f16a4a26806013712b860b60"
productName: "IBM MQ Container Multi Instance for Non-Production Environment"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productCloudpakRatio: "20:3"
```

IBM MQ Advanced Container Multi Instance for Non-Production Environment，含 CP4I 授權

```
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
cloudpakName: "IBM Cloud Pak for Integration"
productID: "31f844f7a96b49749130cd0708fdbb17"
productName: "IBM MQ Advanced Container Multi Instance for Non-Production Environments"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productCloudpakRatio: "10:3"
```

OpenShift CP4I IBM MQ Operator 的相依關係

從 IBM MQ Operator 3.0 開始，當您安裝 IBM MQ Operator 時，不會自動安裝其他操作器。 在舊版 IBM MQ Operator 中，對 IBM Cloud Pak foundational services 操作器，它也會安裝 IBM Operand Deployment Lifecycle Manager (ODLM) 操作器。

需要個別安裝 IBM Licensing Operator，以追蹤授權使用情況。 請參閱 IBM Cloud Pak for Integration 說明文件中的 部署 License Service 。

IBM MQ Operator 3.0 以上版本

V 9.3.4

當您使用 IBM Cloud Pak for Integration 授權來建立 QueueManager 時，您可以選擇是否要對 Keycloak 的 IBM Cloud Pak for Integration 實例使用單一登入。 依預設，使用 IBM Cloud Pak for Integration 授權會啟用 Keycloak，但如果未安裝， QueueManager 會進入「已封鎖」狀態，直到安裝正確的相依關係為止。 如需相依關係的詳細資料，請參閱 第 95 頁的『安裝 IBM MQ Operator』 。

舊版 IBM MQ Operator

當您安裝 舊版 IBM MQ Operator 時，會自動安裝 IBM Cloud Pak foundational services 操作器。 這些相依運算子具有較小的 CPU 及記憶體覆蓋區，並在某些情況下用來部署其他資源。

當您建立 QueueManager 時， IBM MQ Operator 會針對它需要的其他服務建立 OperandRequest 。 OperandRequest 由 ODLM 操作器履行，必要的話，將安裝並實例化必要的服務。 需要哪些服務取決於部署佇列管理程式時所接受的授權合約，以及要求哪些佇列管理程式元件。

- 如果您選擇 IBM MQ Advanced 或 IBM MQ Advanced for Developers 授權，則不會要求其他服務。 例如，在下列情況下，不會使用 IBM Cloud Pak foundational services :

```
spec:
  license:
```

```
    accept: true
    license: L-AMRD-XH6P3Q
    use: "Production"
```

- 如果您選擇 IBM Cloud Pak for Integration 授權並選擇啟用 Web 伺服器，則 IBM MQ Operator 也會實例化「IBM Identity and Access Management (IAM) 操作員」，以啟用單一登入。 如果您已安裝 IBM Cloud Pak for Integration 操作器，則 IAM 操作器將已可用。 例如：

```
spec:
  license:
    accept: true
    license: L-RJON-CD3JKX
    use: "Production"
```

不過，如果您停用 Web 伺服器，則不會要求任何 IBM Cloud Pak foundational services 。 例如：

```
spec:
  license:
    accept: true
    license: L-RJON-CD3JKX
    use: "Production"
  web:
    enabled: false
```

如需相依操作員的軟硬體需求明細，請參閱 基礎服務的硬體需求及建議。

您可以選擇佇列管理程式所使用的 CPU 及記憶體數量。 如需相關資訊，請參閱 第 173 頁的『.spec.queueManager.resources』。

相關參考

第 165 頁的『mq.ibm.com/v1beta1 的授權參考手冊』

OpenShift CP4I IBM MQ Operator 所需的叢集範圍許可權

IBM MQ Operator 需要以叢集為範圍的許可權，才能管理許可 Webhook 和範例，以及讀取儲存類別和叢集版本資訊。

IBM MQ Operator 需要下列叢集範圍的許可權：

- 管理許可 Webhook 的許可權。 這容許建立、擷取及更新特定 Webhook ，這些 Webhook 用於建立及管理「操作員」提供的儲存器的程序。
  - API 群組: **admissionregistration.k8s.io**
  - 資源: **validatingwebhookconfigurations**
  - verbs: **get, delete**
- 建立及管理在 Red Hat OpenShift 主控台中用來在建立自訂資源時提供範例和 Snippet 的資源的許可權。
  - API 群組: **console.openshift.io**
  - 資源: **consoleyamlsamples**
  - verbs: **create, get, update, delete**
- 讀取叢集版本的許可權。 這可讓「操作員」回復叢集環境的任何問題。
  - API 群組: **config.openshift.io**
  - 資源: **clusterversions**
  - verbs: **get, list, watch**
- 在叢集上讀取儲存類別的許可權。 這可讓「操作員」回復儲存器中所選取儲存類別的任何問題。
  - API 群組: **storage.k8s.io**
  - 資源: **storageclasses**
  - verbs: **get, list**

註: IBM MQ Operator 也需要以名稱空間為範圍的許可權。 如果 IBM MQ Operator 安裝在叢集範圍，則名稱空間範圍的許可權會呈現在所有名稱空間中。

OpenShift CP4I Kubernetes IBM MQ Operator 的儲存體考量

IBM MQ Operator 以兩種儲存模式執行:

- 當容器重新啟動時，可以捨棄容器的所有狀態資訊時，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 Operator 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，這些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，則在重新啟動時，佇列管理程式的所有狀態資訊都會遺失。 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的配置。 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 149 頁的『使用 Red Hat OpenShift CLI 以自訂 MQSC 及 INI 檔案建置映像檔』)。 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 。

OpenShift CP4I 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

OpenShift CP4I

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性訊息及配置。 這是預設行為。 因為有各種儲存體提供者，每一個都支援不同的功能，這通常表示需要自訂配置。 下列範例概述在 v1beta1 API 中自訂 IBM 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` 或 `NativeHA`，則只需要 `ReadWriteOnce` 儲存體。 對於 `multiInstance` ，您需要支援 `ReadWriteMany` 且具有正確檔案鎖定性質的儲存類別。 IBM MQ 提供 支援聲明 和 測試聲明。 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 22 頁的『儲存器中 IBM MQ 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:
        class: ibmc-file-gold-gid
      persistedData:
        enabled: true
        class: ibmc-file-gold-gid
      recoveryLogs:
        enabled: true
        class: ibmc-file-gold-gid
  securityContext:
     supplementalGroups: [65534] # Change to 99 for clusters with RHEL7 or earlier worker nodes
```

如需原生 HA 佇列管理程式儲存體考量的相關資訊，請參閱 第 130 頁的『原生 HA』。

註: 您也可以使用單一實例佇列管理程式來配置增補群組。

儲存體容量

OpenShift CP4I

當您使用 IBM MQ Operator 時，所要求的儲存體大小是固定的，且無法在建立佇列管理程式之後重新調整大小。 您必須確定磁區夠大，符合您的需求。

加密

OpenShift CP4I

IBM MQ 不會主動加密靜態資料。 因此，您應該使用被動加密儲存體及/或 IBM MQ Advanced Message Security 來加密訊息。 在 IBM Cloud 上，區塊和檔案儲存空間都有靜態被動加密。

OpenShift CP4I Kubernetes IBM MQ Advanced for Developers 容器映像檔

IBM MQ Advanced for Developers 可以使用預先建置的容器映像檔。 此映像檔可從 IBM Container Registry 取得。 此映像檔適合與 Docker、 Podman、 Kubernetes 及其他容器環境搭配使用。

註: Deprecated IBM MQ Advanced for Developers 映像檔先前可從 Docker Hub 取得，但已淘汰，且在 Docker Hub 上沒有進一步的更新項目。

可用的映像檔

IBM MQ 映像檔儲存在 IBM Container Registry 中:

- IBM MQ Advanced for Developers 9.3.0.17: `icr.io/ibm-messaging/mq:9.3.0.17-r3`
- IBM MQ Advanced for Developers 9.3.5.1: `icr.io/ibm-messaging/mq:9.3.5.1-r2`

快速參照

- 授權:
  - IBM MQ Advanced for Developers 和 Apache 授權 2.0。 請注意， IBM MQ Advanced for Developers 授權不允許進一步配送，且條款限制使用開發人員機器。
- 存檔問題的位置:
  - GitHub
- 適用於下列 CPU 架構:
  - amd64
  - s390x

– ppc64le

使用情形

在儲存器中執行 IBM MQ Advanced for Developers 。

如需如何執行容器的詳細資料，請參閱 使用文件 。

若要能夠使用映像檔，您必須透過設定 **LICENSE** 環境變數來接受 IBM MQ 授權條款。

支援的環境變數

LANG
設定您要用來列印授權的語言。

授權
設定 `accept` 以同意 IBM MQ Advanced for Developers 授權條件。

設定 **檢視** 以檢視授權條件。

Deprecated **log_format**
已淘汰: 由 第 21 頁的『[MQ 9.3.2 2023 年 2 月]MQ_LOGGING_CONSOLE_FORMAT』取代。

變更列印至儲存器 `stdout` 位置的日誌格式。

設定 `basic` ，以使用簡式人類可讀格式。 這是預設值。

設定 `json` 以使用 JSON 格式 (每行一個 JSON 物件)。

Deprecated **MQ_ADMIN_PASSWORD**
指定管理使用者的密碼。

長度必須至少為 8 個字元。

V 9.3.4 管理使用者沒有預設密碼。 對於 IBM MQ Operator 3.0.0 之前的版本，預設值為 `passw0rd`。

V 9.3.4 從 IBM MQ 9.3.4 開始，此變數已淘汰。 本主題中的範例 YAML 顯示如何自行建立此變數並使用密鑰保護其安全。

Deprecated **MQ_APP_PASSWORD**
指定應用程式使用者的密碼。

如果設定的話，這會使 **DEV.APP.SVRCONN** 通道變成安全的，且只容許提供有效使用者 ID 和密碼的連線。

長度必須至少為 8 個字元。

V 9.3.4 沒有應用程式使用者的預設密碼。 對於 IBM MQ Operator 3.0.0 之前的版本，對於 IBM MQ 用戶端，預設值為空白 (不需要密碼) ，對於 HTTP 用戶端，則為 `passw0rd` 。

V 9.3.4 從 IBM MQ 9.3.4 開始，此變數已淘汰。 本主題中的範例 YAML 顯示如何自行建立此變數並使用密鑰保護其安全。

MQ_DEV
設定 `false` 以停止正在建立的預設物件。

MQ_ENABLE_METRICS
設定 `true` ，以產生佇列管理程式的 Prometheus 度量值。

V 9.3.2 **MQ_LOGGING_CONSOLE_SOURCE**
指定鏡映至儲存器 **stdout** 位置之日誌的來源清單 (以逗點區隔)。

有效值為 `qmgr` 及 `web`。

預設值為 `qmgr` ， `web`。

V 9.3.2 **MQ_LOGGING_CONSOLE_FORMAT**
取代 第 21 頁的『[已淘汰]log_format』。

變更列印至儲存器 **stdout** 位置的日誌格式。

設定 `basic`，以使用簡式人類可讀格式。 這是預設值。

設定 `json` 以使用 JSON 格式 (每行一個 JSON 物件)。

V 9.3.2 **MQ_LOGGING_CONSOLE_EXCLUDE_ID**

指定已排除日誌訊息的訊息 ID 清單 (以逗點區隔)。

日誌訊息仍會出現在磁碟上的日誌檔中，但不會列印至儲存器的 **stdout** 位置。

預設值為 `AMQ5041I,AMQ5052I,AMQ5051I,AMQ5037I,AMQ5975I`。

MQ_QMGR_NAME

設定您要用來建立佇列管理程式的名稱。

如需 IBM MQ Advanced for Developers 映像檔所支援預設開發人員配置的相關資訊，請參閱 預設開發人員配置說明文件。

V 9.3.4 說明如何為 admin 和 app 使用者指定密碼的佇列管理程式 YAML 範例

從 IBM MQ 9.3.4 開始， **admin** 和 **app** 使用者 ID 不再具有預設密碼。 對於這些使用者，您必須在使用 `Development` 授權部署佇列管理程式時提供密碼。 以下是範例佇列管理程式 YAML，它顯示如何使用 IBM MQ Operator 來執行此動作。

下列指令會建立密鑰，其中包含 **admin** 及 **app** 使用者的密碼。

```
oc create secret generic my-mq-dev-passwords --from-literal=dev-admin-password=passw0rd --from-
literal=dev-app-password=passw0rd
```

下列 YAML 在部署佇列管理程式時使用這些密碼。

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: qm-dev
spec:
  license:
    accept: false
    license: L-AXAF-JLZ53A
    use: Development
  web:
    enabled: true
  template:
    pod:
      containers:
        - env:
            - name: MQ_DEV
              value: "true"
            - name: MQ_CONNAUTH_USE_HTP
              value: "true"
            - name: MQ_ADMIN_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: my-mq-dev-passwords
                  key: dev-admin-password
            - name: MQ_APP_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: my-mq-dev-passwords
                  key: dev-app-password
          name: qmgr
  queueManager:
    storage:
      queueManager:
        type: persistent-claim
    name: QUICKSTART
  version: 9.3.5.1-r2
```

OpenShift CP4I Kubernetes 儲存器中 IBM MQ 的高可用性

IBM MQ Operator 有三種高可用性選項: **原生 HA 佇列管理程式** (具有作用中抄本及兩個待命抄本)、 **多重實例佇列管理程式** (使用共用網路檔案系統的主動-待命配對) 或 **單一具復原力佇列管理程式** (提供使用網路儲存

體的 HA 簡單方法)。 後兩者依賴檔案系統來確保可回復資料的可用性，但原生 HA 不會。 因此，當不使用「原生 HA」時，檔案系統的可用性對佇列管理程式可用性而言很重要。 當資料回復很重要時，檔案系統應該透過抄寫來確保備援。

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。

原生 HA 佇列管理程式

CP4I MQ Adv.

原生 HA 佇列管理程式涉及一個 **作用中** 及兩個 **抄本** Kubernetes Pod，它們會作為 Kubernetes StatefulSet 的一部分執行，其中每一個都有三個抄本，各有自己的一組 Kubernetes 持續性磁區。 使用原生 HA 佇列管理程式 (租賃型鎖定除外) 時，共用檔案系統的 IBM MQ 需求也適用，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*。 原生 HA 佇列管理程式的回復時間由下列因素控制:

1. 抄本實例偵測作用中實例失敗所花費的時間。 這是可配置的。
2. Kubernetes Pod 就緒探測偵測備妥容器已變更並重新導向網路資料流量所需的時間。 這是可配置的。
3. IBM MQ 用戶端重新連接所需的時間。

如需相關資訊，請參閱 第 130 頁的『原生 HA』。

多重實例佇列管理程式

Multi

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 作為具有正好兩個抄本及一組 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 就緒探測偵測備妥容器已變更並重新導向網路資料流量所需的時間。 這是可配置的。
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

Multi

單一回復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。 這是可配置的。
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。 這是可配置的。
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。 當搭配使用有狀態工作量 (例如 IBM MQ 與 Kubernetes StatefulSet) 時，如果 Kubernetes 主要節點與工作者節點失去聯絡，則無法判定節點是否失敗，或是否僅失去網路連線功能。 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。

註: 透過 IBM MQ Operator 建立佇列管理程式時，不支援變更 IBM MQ 佇列管理程式的 StatefulSet 詳細資料 (包括抄本數目)。

相關概念

高可用性配置

相關工作

第 130 頁的『使用 IBM MQ Operator 來配置佇列管理程式的高可用性』

OpenShift CP4I Kubernetes 儲存器中 IBM MQ 的災難回復

您需要考量您正在準備的災難類型。 在雲端環境中，使用可用性區域可提供特定層次的災難容錯，而且更容易使用。 如果您有奇數資料中心 (用於仲裁) 和低延遲網路鏈結，則可能執行具有多個可用性區域的單一 Red Hat OpenShift Container Platform 或 Kubernetes 叢集，每個區域位於個別實體位置。 本主題討論無法符合這些準則的災難回復考量: 即偶數個資料中心或高延遲網路鏈結。

對於災難回復，您需要考量下列各項:

- 將 IBM MQ 資料 (保留在一或多個 `PersistentVolume` 資源中) 抄寫至災難回復位置
- 使用抄寫的資料重建佇列管理程式
- IBM MQ 用戶端應用程式及其他佇列管理程式可看見的佇列管理程式網路 ID。 例如，此 ID 可以是 DNS 項目。

持續資料需要同步或非同步抄寫至災難回復站台。 這通常是儲存體提供者特有的，但也可以使用 `VolumeSnapshot` 來完成。 如需磁區 Snapshot 的相關資訊，請參閱 CSI 磁區 Snapshot 。

從災難回復時，您需要使用抄寫的資料，在新的 Kubernetes 叢集上重建佇列管理程式實例。 如果您使用 IBM MQ Operator，則需要 `QueueManager` YAML 以及 YAML ，以用於其他支援資源，例如 `ConfigMap` 或 `Secret`。

相關資訊

ha_for_ctr.dita

OpenShift CP4I 規劃保護儲存器中的 IBM MQ 安全

在儲存器配置中規劃 IBM MQ 時的安全考量。

程序

- 第 25 頁的『儲存器中 IBM MQ 的使用者鑑別和授權』
  - 第 25 頁的『在儲存器中使用作業系統使用者的安全限制』
- 第 25 頁的『限制容器中 IBM MQ 的網路資料流量的考量』

儲存器中 IBM MQ 的使用者鑑別和授權

儲存器中的 IBM MQ 可以配置為透過 LDAP、相互 TLS 或自訂 MQ 外掛程式來鑑別使用者。

請注意，「IBM MQ 操作員」不容許在儲存器映像檔內使用作業系統使用者和群組。 如需相關資訊，請參閱第 25 頁的『在儲存器中使用作業系統使用者的安全限制』。

LDAP

如需配置 IBM MQ 以使用 LDAP 使用者儲存庫的相關資訊，請參閱 連線鑑別: 使用者儲存庫 及 LDAP 授權。

相互 TLS

如果您將佇列管理程式的送入連線配置為需要 TLS 憑證 (相互 TLS)，則可以將憑證的識別名稱對映至使用者名稱。 您需要執行兩件事:

- 使用 SSLPEER 來配置通道鑑別記錄，以建立與使用者名稱的對映。 如需相關資訊，請參閱 將 SSL 或 TLS 識別名稱對映至 MCAUSER 使用者 ID。
- 配置佇列管理程式，以容許您為系統不知道的使用者名稱定義權限記錄。 如需相關資訊，請參閱 qm.ini 檔案的服務段落。

JSON Web 記號

如需配置 IBM MQ 以使用「JSON Web 記號 (JWT)」的相關資訊，請參閱 使用鑑別記號。

自訂 MQ 外掛程式

這是一種先進的技術，需要更多的工作。 如需相關資訊，請參閱 使用自訂授權服務。

相關工作

第 125 頁的『範例: 配置具有相互 TLS 鑑別的佇列管理程式』
此範例使用 IBM MQ Operator 將佇列管理程式部署至 OpenShift Container Platform。 交互 TLS 用於鑑別，以從 TLS 憑證對映至佇列管理程式中的身分。

在儲存器中使用作業系統使用者的安全限制

不建議在儲存器中使用作業系統使用者，且「IBM MQ 操作器」禁止使用作業系統使用者。

在多租戶容器化環境中，通常會設定安全限制，以防止潛在的安全問題，例如:

- **防止在容器內使用 "root" 使用者**
- **強制使用隨機 UID**。 例如，在 Red Hat OpenShift Container Platform 中，預設 `SecurityContextConstraints` (稱為 `restricted`) 會針對每一個儲存器使用隨機化使用者 ID。
- **防止使用專用權升級**。 IBM MQ on Linux 會使用專用權提升來檢查使用者的密碼-它會使用 "setuid" 程式，以便成為執行此動作的「root」使用者。

OpenShift CP4I 為了確保符合這些安全措施， IBM MQ Operator 不容許使用儲存器內作業系統程式庫上定義的 ID。 儲存器中未定義任何 `mqm` 使用者 ID 或群組。

限制容器中 IBM MQ 的網路資料流量的考量

您可以在 OpenShift Container Platform 和 Kubernetes 中定義網路原則，以限制叢集裡 Pod 的資料流量。本主題說明網路原則如何套用至 IBM MQ 的部分考量。

對於佇列管理程式的網路入口，有數個埠需要考量:

- 埠 1414 用於佇列管理程式資料流量
- 埠 9414 (適用於原生 HA)
- 埠 9157 用於度量
- Web 主控台和 REST API 的埠 9443

網路輸出更為複雜。 您可能想要考量的網路輸出範例:

- DNS-如果您有通道或其他配置使用 DNS 名稱
- 其他佇列管理程式
- 線上憑證狀態通訊協定 (OCSP) 及憑證撤銷清冊 (CRL)-由憑證提供者決定。
- 鑑別提供者:
  - LDAP
  - 針對 IBM MQ Web 伺服器，開啟 ID Connect 或其他已配置的登入提供者。 這包括 IBM Cloud Pak Platform 使用者介面及 IBM Cloud Pak 基礎服務 IAM。
- 追蹤提供者:
  - Instana
  - Cloud Pak for Integration 作業儀表板 [4]

範例 Ingress NetworkPolicy

下列網路原則範例用來控制稱為 "myqm" 之佇列管理程式的入口，以在 Red Hat OpenShift Container Platform 上使用。

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

[4] 「作業儀表板」已從 IBM MQ 9.3.0 中淘汰，並在 IBM MQ 9.3.3 中移除。 請參閱第 142 頁的『與 IBM Cloud Pak for Integration 作業儀表板整合』。

儲存器中 IBM MQ 的 FIPS 相符性

在啟動時，儲存器中的 IBM MQ 會偵測儲存器啟動所在的作業系統是否符合 FIPS 標準，以及 (如果是的話) 是否自動配置 FIPS 支援。 這裡說明需求和限制。

聯邦資訊處理標準

美國政府針對 IT 系統和安全 (包括資料加密) 提供技術建議。 國家標準與技術機構 (NIST) 是負責 IT 系統和安全的政府機構。 NIST 會產生建議和標準，包括「聯邦資訊存取安全標準 (FIPS)」。

FIPS 140-2 是重要的 FIPS 標準，它需要使用強大的加密演算法。 FIPS 140-2 也指定雜湊演算法的需求，用來保護封包在傳輸中不受修改。

IBM MQ 提供 FIPS 140-2 支援 (如果已配置此支援)。

註: 在 AIX, Linux, and Windows 上， IBM MQ 透過 IBM Crypto for C (ICC) 加密模組提供 FIPS 140-2 相符性。 此模組的憑證已移至「歷程」狀態。 客戶應該檢視 IBM Crypto for C (ICC) 憑證 ，並注意 NIST 提供的任何建議。 目前正在進行取代 FIPS 140-3 模組，您可以在 處理程序清單中的 NIST CMVP 模組中搜尋它，以檢視其狀態。

<

如需與叢集設定及其他考量相關的需求，請參閱 FIPS Wall: 現行 IBM FIPS 相符性方法。

儲存器中的 IBM MQ 可以在 FIPS 140-2 相符性模式下執行。 在啟動期間，儲存器中的 IBM MQ (9.3.1.0 及更高版本) 會偵測儲存器啟動所在的主機作業系統是否符合 FIPS 標準。 如果主機作業系統符合 FIPS 標準，且已提供私密金鑰和憑證，則 IBM MQ 儲存器會配置佇列管理程式、 IBM MQ Web 伺服器，以及「原生高可用性」部署中節點之間的資料傳送，以在 FIPS 相符性模式下執行。

使用 IBM MQ Operator 來部署佇列管理程式時，操作器會建立終止類型為 **Passthrough** 的路徑。 這表示資料流量會直接傳送至目的地，而不需要路由器提供 TLS 終止。 在此情況下， IBM MQ 佇列管理程式和 IBM MQ Web 伺服器是目的地，它們已提供符合 FIPS 標準的安全通訊。

主要需求:

1. 在密鑰中提供給佇列管理程式和 Web 伺服器的私密金鑰和憑證，可讓外部用戶端安全連接至佇列管理程式和 Web 伺服器。
2. 在「原生高可用性」配置中，用於在不同節點之間進行資料傳送的私密金鑰和憑證。

限制

對於儲存器中符合 FIPS 標準的 IBM MQ 部署，請考量下列事項:

- 容器中的 IBM MQ 提供用於收集度量值的端點。 目前此端點僅為 HTTP。 您可以關閉度量值端點，使其餘的 IBM MQ 符合 FIPS 標準。
- 儲存器中的 IBM MQ 容許自訂映像檔置換。 也就是說，您可以使用 IBM MQ 儲存器映像檔作為基本映像檔來建置自訂映像檔。 FIPS 相符性可能不適用於這類自訂映像檔。
- 對於使用 IBM Instana 進行訊息追蹤， IBM MQ 與 IBM Instana 之間的通訊是 HTTP 或 HTTPS ，不符合 FIPS 標準。
- 對 IBM 身分及存取管理 (IAM) /Zen 服務的 IBM MQ Operator 存取權不符合 FIPS 標準。

如何偵測 *FIPS* 合規性並自動配置 *FIPS* 支援

如果容器啟動所在的作業系統符合 FIPS 標準，則會自動配置 FIPS 支援。

註: 在 AIX, Linux, and Windows 上， IBM MQ 透過 IBM Crypto for C (ICC) 加密模組提供 FIPS 140-2 相符性。 此模組的憑證已移至「歷程」狀態。 客戶應該檢視 IBM Crypto for C (ICC) 憑證 ，並注意 NIST 提供的任何建議。 目前正在進行取代 FIPS 140-3 模組，您可以在 處理程序清單中的 NIST CMVP 模組中搜尋它，以檢視其狀態。

在啟動期間，儲存器中的 IBM MQ 會偵測儲存器啟動所在的作業系統是否符合 FIPS 標準。 如果是，則會自動採取下列動作:

佇列管理程式
如果主機作業系統符合 FIPS 標準，且提供私密金鑰和憑證，則佇列管理程式屬性 **SSLFIPS** 會設為 YES。 否則， **SSLFIPS** 屬性會設為 NO。

IBM MQ Web 伺服器
IBM MQ Web 伺服器提供用於管理 IBM MQ 的 HTTP/HTTPS 介面。 如果主機作業系統符合 FIPS 標準，則會更新 JVM 選項，以讓 Web 伺服器使用符合 FIPS 標準的加密法。 若要能夠使用 FIPS，必須在儲存器啟動期間提供私密金鑰及憑證。

原生 HA
在節點之間抄寫的資料安全由 qm.ini 檔案的 **NativeHALocalInstance** 段落控制。 例如：

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

如果啟用 FIPS，則會將 **SSLFipsRequired** 屬性新增至段落，並將值設為 Yes:

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

如果儲存器在沒有 FIPS 支援的 OpenShift 叢集中執行，則佇列管理程式、 IBM MQ Web 伺服器及原生 HA 元件不會自動啟用其 FIPS 支援。 FIPS 的 OpenShift 平台目前僅支援 x86-64 架構。 對於 Power 和 Linux for IBM Z 架構， OpenShift 不提供 FIPS 支援。 若要在這些架構的 IBM MQ 元件中明確啟用 FIPS 支援，請在佇列管理程式 YAML 中，將 *MQ_ENABLE_FIPS* 環境變數設為 true 。 下列 YAML Snippet 說明 *MQ_ENABLE_FIPS* 環境變數的用法:

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

在儲存器中置換 *IBM MQ* 的自動 *FIPS* 模式
使用環境變數 *MQ_ENABLE_FIPS* 來明確啟用或停用儲存器中 IBM MQ 元件的 FIPS 模式。

開始之前
註: 在 AIX, Linux, and Windows 上， IBM MQ 透過 IBM Crypto for C (ICC) 加密模組提供 FIPS 140-2 相符性。 此模組的憑證已移至「歷程」狀態。 客戶應該檢視 IBM Crypto for C (ICC) 憑證，並注意 NIST 提供的任何建議。 目前正在進行取代 FIPS 140-3 模組，您可以在 處理程序清單中的 NIST CMVP 模組中搜尋它，以檢視其狀態。

關於這項作業
MQ_ENABLE_FIPS 支援三個值:

自動
這是預設值。

如果主機作業系統已啟用 FIPS，則所有元件 (佇列管理程式、 IBM MQ Web 伺服器及原生 HA) 都以 FIPS 模式執行。

如果主機作業系統未啟用 FIPS，則所有元件都不會以 FIPS 模式執行。

true
此值會針對儲存器中選取的元件開啟 FIPS。

即使儲存器中的 IBM MQ 在不符合 FIPS 標準的主機作業系統上執行，佇列管理程式屬性 **SSLFIPS** 也會設為 YES 。 亦即，如果 IBM MQ 佇列管理程式、Web 伺服器及「原生 HA」符合 FIPS 標準，但儲存器的作業系統不符合 FIPS 標準。

false
此值會關閉 FIPS 相符性。

即使容器中的 IBM MQ 在符合 FIPS 標準的主機上執行，佇列管理程式屬性 **SSLFIPS** 也會設為 NO。 不過，如果提供私密金鑰和憑證， IBM MQ 仍會保護連線安全。

不會更新 IBM MQ Web 伺服器的 JVM 選項。 不過，如果提供私密金鑰和憑證， IBM MQ Web 伺服器仍會執行 HTTP 端點。

原生 HA 中的資料抄寫不使用 FIPS 加密法。

範例

以下是佇列管理程式 YAML 範例，說明針對佇列管理程式元件啟用 TLS 及 FIPS:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  namespace: ibm-mq-fips
  name: ibm-mq-qm-ppcle
spec:
  license:
    accept: true
    license: L-AMRD-XH6P3Q
    use: Production
  queueManager:
    name: PPCLEQM
    storage:
      queueManager:
        type: ephemeral
  template:
    pod:
      containers:
        - env:
            - name: MQ_ENABLE_FIPS
              value: "true"
          name: qmgr
  version: 9.3.5.1-r2
  web:
    enabled: false
  pki:
    keys:
      - name: ibm-mq-tls-certs
        secret:
          secretName: ibm-mq-tls-secret
          items:
          - tls.key
          - tls.crt
```

Multi 規劃儲存器中 IBM MQ 的可調整性及效能

在大部分情況下，儲存器中 IBM MQ 的調整大小及效能與 IBM MQ for Multiplatforms 相同。 不過，儲存器平台還有一些額外的限制。

關於這項作業

在儲存器中規劃 IBM MQ 的可調整性及效能時，請考量下列選項:

程序

- **限制執行緒及處理程序的數目**。

  IBM MQ 使用執行緒來管理並行。 在 Linux 中，執行緒會實作為處理程序，因此您可能會遇到儲存器平台或作業系統對處理程序數目上限所強制的限制。 從 Red Hat OpenShift Container Platform 4.11 開始，每個容器的預設限制為 4096 個處理程序。 對於舊版 OpenShift Container Platform ，限制為 1024 個處理程序。 如需 IBM MQ Operator 版本與 OpenShift Container Platform 版本的相容性，請參閱 第 11 頁

的『相容的 Red Hat OpenShift Container Platform 版本』。 雖然這適用於絕大多數的實務範例，但在某些情況下，這可能會影響佇列管理程式的用戶端連線數目。

叢集管理者可以使用 kubelet 配置設定 **podPidsLimit** 來配置 Kubernetes 中的程序限制。 請參閱 Kubernetes 文件中的 處理程序 ID 限制和保留 。 在 Red Hat OpenShift Container Platform 中，您也可以 建立 **ContainerRuntimeConfig** 自訂資源以編輯 CRI-O 參數。

在 IBM MQ 配置中，您也可以設定佇列管理程式的用戶端連線數目上限。 請參閱 伺服器連線通道限制，以瞭解將限制套用至個別伺服器連線通道，以及 MAXCHANNELS INI 屬性，以瞭解將限制套用至整個佇列管理程式。

- **限制磁區數目**。

  在雲端及儲存器系統中，通常會使用網路連接儲存磁區。 可以連接至 Linux 節點的磁區數目有一些限制。 例如， AWS EC2 限制為每個 VM 不超過 30 個磁區。 Red Hat OpenShift Container Platform 有類似的限制，與 Microsoft Azure 和 Google Cloud Platform 一樣。

  「原生 HA」佇列管理程式需要三個實例中的每一個實例都有一個磁區，並強制實例分散在節點之間。 不過，您可以將佇列管理程式配置成每個實例使用三個磁區 (佇列管理程式資料、回復日誌及持續保存資料)。

- **使用 IBM MQ 調整大小技術**。

  使用 IBM MQ 調整大小技術 (例如 IBM MQ 統一叢集) 來執行多個具有相同配置的佇列管理程式，會有助於取代少數大型佇列管理程式。 這會增加減少單一容器重新啟動 (例如，作為容器平台維護的一部分) 的影響的好處。

OpenShift CP4I-LTS CP4I CD 使用 IBM MQ Operator for Red Hat OpenShift

IBM MQ Operator 會將 IBM MQ 部署並管理為 IBM Cloud Pak for Integration 的一部分，或在 Red Hat OpenShift Container Platform 上獨立部署並管理

程序

- 第 30 頁的『IBM MQ Operator 的發行歷程』.
- 第 73 頁的『將 IBM MQ 移轉至 IBM Cloud Pak for Integration』.
- 第 95 頁的『安裝 IBM MQ Operator』.
- 第 110 頁的『升級 IBM MQ Operator 及佇列管理程式』.
- 第 106 頁的『將佇列管理程式部署至 Red Hat OpenShift Container Platform 叢集』.
- 第 155 頁的『使用 IBM MQ Operator 操作 IBM MQ』.
- 第 165 頁的『IBM MQ Operator 的 API 參考資料』.

OpenShift CP4I-LTS CP4I CD IBM MQ Operator 的發行歷程

附註:

- 如需舊版 IBM MQ 操作器的相關資訊，請參閱 IBM MQ 9.2 文件中的 IBM MQ Operator 的版本歷程 。
- 如需未來 IBM MQ 更新項目的相關資訊，請參閱整體 IBM MQ 計劃維護版本日期 頁面。

IBM MQ Operator 3.1.3

CD

IBM Cloud Pak for Integration 版本

IBM Cloud Pak for Integration 2023.4.1

操作員通道

v3.1

.spec.version 容許的值
9.3.5.1-r2

移轉期間容許的 .spec.version 值
9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3,9.3.3.2-r3 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1, 9.3.0.16-r2, 9.3.0.17-r1, 9.3.0.17-r2, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3, 9.3.3.3-r1, 9.3.3.3-r2, 9.3.4.0-r1, 9.3.4.1-r1, 9.3.5.0-r1, 9.3.5.0-r2, 9.3.5.1-r1

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.12 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.14 及 4.16。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 4.3 版及更新版本 (選用安裝)。

變更的內容

- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 3.1.2

CD

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2023.4.1

操作員通道
v3.1

.spec.version 容許的值
9.3.5.1-r1

移轉期間容許的 .spec.version 值
9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3,9.3.3.2-r3 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1, 9.3.0.16-r2, 9.3.0.17-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3, 9.3.3.3-r1, 9.3.3.3-r2, 9.3.4.0-r1, 9.3.4.1-r1, 9.3.5.0-r1, 9.3.5.0-r2,

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.12 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.14 及 4.16。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 4.3 版及更新版本 (選用安裝)。

變更的內容

- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 3.1.1

CD

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2023.4.1

操作員通道
v3.1

.spec.version 容許的值
9.3.5.0-r2

移轉期間容許的 .spec.version 值
9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3,9.3.3.2-r3 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1, 9.3.0.16-r2, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3, 9.3.3.3-r1, 9.3.3.3-r2, 9.3.4.0-r1, 9.3.4.1-r1, 9.3.5.0-r1

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.12 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.14 及 4.16。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 4.3 版及更新版本 (選用安裝)。

變更的內容

- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 3.1.0

CD

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2023.4.1

操作員通道
v3.1

.spec.version 容許的值
9.3.5.0-r1

移轉期間容許的 .spec.version 值
9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3,9.3.3.2-r3 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3, 9.3.3.3-r1, 9.3.3.3-r2, 9.3.4.0-r1, 9.3.4.1-r1

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.12 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.14 及 4.16。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 4.3 版及更新版本 (選用安裝)。

變更的內容

- 這些「安全佈告欄」中詳述已解決的漏洞:
  - https://www.ibm.com/support/pages/node/7126571.
  - https://www.ibm.com/support/pages/node/7137570.

IBM MQ Operator 3.0.1

CD

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2023.4.1

操作員通道
v3.0

.spec.version 容許的值
9.3.4.1-r1

移轉期間容許的 .spec.version 值
9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3,9.3.3.2-r3 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2,

9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3, 9.3.3.3-r1, 9.3.4.0-r1

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.12 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.14 及 4.16。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 4.3 版及更新版本 (選用安裝)。

變更的內容

- 在 第 33 頁的『IBM MQ Operator 3.0.0』上建置的僅安全更新。
- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 3.0.0

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2023.4.1

操作員通道
`v3.0`

`.spec.version` 容許的值
9.3.4.0-r1

移轉期間容許的 `.spec.version` 值
9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.12 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.14 及 4.16。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 4.3 版及更新版本 (選用安裝)。

新增功能

- 您可以使用新的 `manualConfig` YAML 內容 (需要 IBM MQ 9.3.4 或更新版本)，將 `mqwebuser.xml` 檔案新增至 ConfigMap 或「密鑰」來配置 IBM MQ Web 伺服器。
- 現在支援 administrative REST API 。 您可以透過上述 ConfigMap 或「密鑰」來配置 (需要 IBM MQ 9.3.4 或更新版本)。 不過，請注意 Web 伺服器仍未被視為存活性探測的重要服務，因此如果失敗，儲存器將不會自動重新啟動。
- 當使用 IBM Cloud Pak for Integration 授權時，您可以選擇「手動」鑑別和授權 (需要 IBM MQ 9.3.4 或更高版本) 來停用單一登入
- 您可以在儲存器內啟用唯讀 root 檔案系統。 這可防止在執行時期寫入儲存器內的大部分檔案 (需要 IBM MQ 9.3.4 或更高版本)，從而增進安全。 `readOnlyRootFilesystem` 選項隨附其他選項，可配置裝載以容許寫入暫存檔之 "scratch" 及 "tmp" 磁區的大小。 請參閱 第 152 頁的『使用唯讀根檔案系統執行 IBM MQ 儲存器』

變更的內容

- 已移除 (先前已淘汰) 版本: IBM MQ 9.2.0 EUS、 9.2.3、 9.2.4、 9.2.5。 重要事項: 在升級 IBM MQ Operator 之前，請確定您沒有任何已移除版本的佇列管理程式。 升級之後，除了升級至支援內版本之外，您將無法再編輯 QueueManager 資源，因為 IBM MQ Operator 不再辨識舊版本。
- 操作器安裝及生命週期
  - Red Hat OpenShift Container Platform 4.14 版現在支援 IBM MQ Operator 。

  - IBM MQ Operator 不再自動安裝 IBM Cloud Pak foundational services 。 如果您部署使用 IBM Cloud Pak for Integration 授權並配置單一登入 (具有該授權之佇列管理程式的預設值) 的 QueueManager ，如果尚未安裝必要的相依關係，則 QueueManager 會進入「已封鎖」狀態。 不會自動安裝其他操作員。
- 安全變更
  - IBM Cloud Pak for Integration 2023.4.1 使用 Keycloak 進行單一登入及授權，而非 IBM Cloud Pak Identity and Access Manager。
  - IBM Cloud Pak for Integration 「快速入門」範本不再使用 *MQSNAUT* 停用安全。 您需要配置鑑別。 請參閱第 25 頁的『儲存器中 IBM MQ 的使用者鑑別和授權』。
  - 從 9.3.4 版開始，在 IBM MQ Advanced for Developers 中已停用預設使用者。 依預設會停用預設使用者 ("admin" 和 "app") 以及作為 IBM MQ Advanced for Developers 一部分提供的其他配置。
- IBM MQ Operator Pod 的次要變更:
  - IBM MQ Operator 不再部署起始設定儲存器
  - IBM MQ Operator 儲存器名稱現在是 *manager*
  - IBM MQ Operator Pod 字首為 *ibm-mq-operator*
- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 2.4.8

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2023.2.1

操作員通道
`v2.4`

`.spec.version` 容許的值
9.3.3.3-r2

移轉期間容許的 `.spec.version` 值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3, 9.3.3.3-r1

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.12 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.14 及 4.16。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 版本 3.19 至 3.24 (含)。

變更的內容

- 在 第 38 頁的『IBM MQ Operator 2.4.0』上建置的僅安全更新。
- 這些「安全佈告欄」中詳述已解決的漏洞:
  - https://www.ibm.com/support/pages/node/7126571.
  - https://www.ibm.com/support/pages/node/7137570.

IBM MQ Operator 2.4.7

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2023.2.1

操作員通道
`v2.4`

.spec.version 容許的值
9.3.3.3-r1

移轉期間容許的 .spec.version 值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.12 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.14 及 4.16。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 版本 3.19 至 3.24 (含)。

變更的內容

- 在 第 38 頁的『IBM MQ Operator 2.4.0』上建置的僅安全更新。
- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 2.4.6

CD

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2023.2.1

操作員通道
v2.4

.spec.version 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.12 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.14 及 4.16。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 版本 3.19 至 3.24 (含)。

變更的內容

- 在 第 38 頁的『IBM MQ Operator 2.4.0』上建置的僅安全更新。
- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 2.4.5

CD

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2023.2.1

操作員通道
v2.4

.spec.version 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1,

9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.12 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.14 及 4.16。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 版本 3.19 至 3.24 (含)。

變更的內容

- 在 第 38 頁的『IBM MQ Operator 2.4.0』上建置的僅安全更新。
- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 2.4.4

CD

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2023.2.1

操作員通道
v2.4

`.spec.version` 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.12 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.14 及 4.16。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 版本 3.19 至 3.24 (含)。

變更的內容

- 在 第 38 頁的『IBM MQ Operator 2.4.0』上建置的僅安全更新。
- 此 Security Bulletin 中詳述已解決的漏洞。
- OpenShift Container Platform 4.10 不再測試或支援 IBM MQ Operator 。

IBM MQ Operator 2.4.3

CD

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2023.2.1

操作員通道
v2.4

`.spec.version` 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.10 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.10 及 4.12。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 版本 3.19 至 3.24 (含)。

變更的內容

- 在 第 38 頁的『IBM MQ Operator 2.4.0』 上建置的僅安全更新。
- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 2.4.2

CD

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2023.2.1

操作員通道
v2.4

.spec.version 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.10 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.10 及 4.12。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 版本 3.19 至 3.24 (含)。

變更的內容

- 在 第 38 頁的『IBM MQ Operator 2.4.0』 上建置的僅安全更新。
- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 2.4.1

CD

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2023.2.1

操作員通道
v2.4

.spec.version 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.10 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.10 及 4.12。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 版本 3.19 至 3.24 (含)。

變更的內容

- 在 第 38 頁的『IBM MQ Operator 2.4.0』上建置的僅安全更新。
- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 2.4.0

CD

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2023.2.1

操作員通道
v2.4

`.spec.version` 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.10 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.10 及 4.12。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 版本 3.19 至 3.24 (含)。

新增功能

- 已移除作業儀表板整合。
- 已新增 **`LogFilePages`** 的 IBM MQ Operator 支援。

變更的內容

- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 2.3.3

CD

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.4.1

操作員通道
v2.3

`.spec.version` 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.10 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.10 及 4.12。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 版本 3.19 至 3.24 (含)。

變更的內容

- 以 第 39 頁的『IBM MQ Operator 2.3.0』為建置基礎的僅安全更新
- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 2.3.2

CD

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.4.1

操作員通道
v2.3

`.spec.version` 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.10 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.10 及 4.12。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 版本 3.19 至 3.24 (含)。

變更的內容

- 以 第 39 頁的『IBM MQ Operator 2.3.0』 為建置基礎的僅安全更新
- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 2.3.1

CD

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.4.1

操作員通道
v2.3

`.spec.version` 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.10 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.10 及 4.12。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 版本 3.19 至 3.24 (含)。

新增功能

- 從 2023 年 3 月開始， IBM MQ Operator 和 IBM MQ 佇列管理程式儲存器映像檔會進行數位簽署。 IBM MQ Operator 2.3.1 和 IBM MQ 9.3.2.0-r2 映像檔已使用此版本簽署。 請參閱第 73 頁的『驗證映像檔簽章』。

變更的內容

- 以 第 39 頁的『IBM MQ Operator 2.3.0』 為建置基礎的僅安全更新
- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 2.3.0

CD

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.4.1

操作員通道
v2.3

.spec.version 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.10 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.10 及 4.12。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 版本 3.19 至 3.24 (含)。

新增功能

- 從 IBM MQ Operator 2.3.0 開始，可以配置 FIPS 140-2 支援。 請參閱 第 27 頁的『儲存器中 IBM MQ 的 FIPS 相符性』。
- 從 IBM MQ Operator 2.3.0 開始， IBM MQ 9.3.1 已淘汰。

變更的內容

- 從 IBM MQ Operator 2.3.0 開始， 第 130 頁的『原生 HA』 在 IBM MQ Advanced 或 IBM MQ Advanced for Developers 授權下可用
- IBM MQ Operator 所需的許可權集已減少。
- 從 IBM MQ Operator 2.3.0 開始， **ibm-automation-core** 會從針對 IBM Cloud Pak for Integration 部署所建立的 OperandRequest 中移除。
- 從 IBM MQ Operator 2.3.0 開始， IBM MQ Operator 部署指定 IfNotPresent 的 **imagePullPolicy**。
- 這些安全公告中詳述已解決的漏洞:
  - CVE-2022-47629 和 CVE-2022-35737 的公佈欄
  - CVE-2023-26284

IBM MQ Operator 2.2.2

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.4.1

操作員通道
v2.2

.spec.version 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.10 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.10 及 4.12。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 版本 3.19 至 3.24 (含)。

變更的內容

- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 2.2.1

CD

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.4.1

操作員通道
v2.2

.spec.version 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.10 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.10 及 4.12。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 版本 3.19 至 3.24 (含)。

變更的內容

- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 2.2.0

CD

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.4.1

操作員通道
v2.2

.spec.version 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.1.0-r1, 9.3.1.0-r2

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.10 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.10 及 4.12。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 版本 3.19 至 3.24 (含)。

新增功能

- 從 IBM MQ Operator 2.2.0 (CD) 開始，原生支援 IBM Instana 追蹤。 9.3.1.0-r2 (CD) IBM MQ 佇列管理程式儲存器映像檔中提供支援。 9.3.1.0-r2 包含 IBM Instana MQ 結束程式 2.4.0 版 (2022.4.0)。若要啟用 IBM Instana 追蹤，請參閱 第 143 頁的『整合 IBM MQ 與 IBM Instana 追蹤』。

變更的內容

- 此 Security Bulletin 中詳述已解決的漏洞。
- 從 IBM MQ Operator 2.2.0 開始，「作業儀表板」已淘汰，將不會收到進一步更新。 不應開始新的「作業儀表板」使用。 IBM MQ 2.0.x 操作員繼續支援「作業儀表板」。

IBM MQ Operator 2.1.0

CD

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.2.1

操作員通道
v2.1

.spec.version 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.1.0-r1

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.10 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.10 及 4.12。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 3.X，但至少 3.19

新增功能

- 新增 IBM MQ 9.3.1。
- 新增選項可讓使用者 停用佇列管理程式規格的預設值更新。
- 新增狀態條件，以淘汰 IBM MQ 9.3.1 之前的所有 IBM MQ 版本。
- 新增狀態條件，以警告使用此 CD 版本 IBM MQ Operator 的 LTS 運算元的使用者。

變更的內容

- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 2.0.23 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.2.1

操作員通道
v2.0

.spec.version 容許的值
9.3.0.17-r3

移轉期間容許的 .spec.version 值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1, 9.3.0.16-r2, 9.3.0.17-r1, 9.3.0.17-r2

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.12 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.14 及 4.16。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 3.19

變更的內容

- 在 第 52 頁的『IBM MQ Operator 2.0.0』 上建置的安全更新
- IBM MQ 型錄映像檔已從 SQLite 資料庫格式移至檔案型型錄格式。
- 已解決的漏洞在此 Security Bulletin 中有詳細說明。

IBM MQ Operator 2.0.22 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.2.1

操作員通道
v2.0

.spec.version 容許的值
9.3.0.17-r2

移轉期間容許的 .spec.version 值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1, 9.3.0.16-r2, 9.3.0.17-r1

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.12 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.14 及 4.16。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 3.19

變更的內容

- 以 第 52 頁的『IBM MQ Operator 2.0.0』 為建置基礎的僅安全更新
- 已解決的漏洞在此 Security Bulletin 中有詳細說明。

IBM MQ Operator 2.0.21 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.2.1

操作員通道
v2.0

.spec.version 容許的值
9.3.0.17-r1

移轉期間容許的 .spec.version 值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1, 9.3.0.16-r2

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.12 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.14 及 4.16。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 3.19

變更的內容

- 以 第 52 頁的『IBM MQ Operator 2.0.0』 為建置基礎的僅安全更新
- 已解決的漏洞在此 Security Bulletin 中有詳細說明。

IBM MQ Operator 2.0.20 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.2.1

操作員通道
v2.0

.spec.version 容許的值
9.3.0.16-r2

移轉期間容許的 .spec.version 值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.12 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.14 及 4.16。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 3.19

變更的內容

- 以 第 52 頁的『IBM MQ Operator 2.0.0』 為建置基礎的僅安全更新
- 已解決的漏洞在此 Security Bulletin 中有詳細說明。

IBM MQ Operator 2.0.19 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.2.1

操作員通道
`v2.0`

.spec.version 容許的值
9.3.0.16-r1

移轉期間容許的 .spec.version 值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.12 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.14 及 4.16。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 3.19

變更的內容

- 以 第 52 頁的『IBM MQ Operator 2.0.0』 為建置基礎的僅安全更新
- 這些「安全佈告欄」中詳述已解決的漏洞:
  - https://www.ibm.com/support/pages/node/7126571.
  - https://www.ibm.com/support/pages/node/7137570.

IBM MQ Operator 2.0.18 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.2.1

操作員通道
`v2.0`

.spec.version 容許的值
9.3.0.15-r1

移轉期間容許的 .spec.version 值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.12 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.14 及 4.16。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 3.19

變更的內容

- 以 第 52 頁的『IBM MQ Operator 2.0.0』 為建置基礎的僅安全更新
- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 2.0.17 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.2.1

操作員通道
v2.0

.spec.version 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.12 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.14 及 4.16。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 3.19

變更的內容

- 以 第 52 頁的『IBM MQ Operator 2.0.0』 為建置基礎的僅安全更新
- 此 Security Bulletin 中詳述已解決的漏洞。
- OpenShift Container Platform 4.10 不再測試或支援 IBM MQ Operator 。

IBM MQ Operator 2.0.16 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.2.1

操作員通道
v2.0

.spec.version 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4,

9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.12 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.14 及 4.16。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 3.19

變更的內容

- 以 第 52 頁的『IBM MQ Operator 2.0.0』 為建置基礎的僅安全更新
- 此 Security Bulletin 中詳述已解決的漏洞。
- OpenShift Container Platform 4.10 不再測試或支援 IBM MQ Operator 。

IBM MQ Operator 2.0.15 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.2.1

操作員通道
`v2.0`

`.spec.version` 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.10 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.10 及 4.12。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 3.19

變更的內容

- 以 第 52 頁的『IBM MQ Operator 2.0.0』 為建置基礎的僅安全更新
- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 2.0.14 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.2.1

操作員通道
`v2.0`

`.spec.version` 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.10 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.10 及 4.12。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 3.19

變更的內容

- 以 第 52 頁的『IBM MQ Operator 2.0.0』 為建置基礎的僅安全更新
- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 2.0.13 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.2.1

操作員通道
`v2.0`

`.spec.version` 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.10 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.10 及 4.12。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 3.19

變更的內容

- 以 第 52 頁的『IBM MQ Operator 2.0.0』 為建置基礎的僅安全更新
- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 2.0.12 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.2.1

操作員通道
`v2.0`

`.spec.version` 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.10 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.10 及 4.12。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 3.19

變更的內容

- 以 第 52 頁的『IBM MQ Operator 2.0.0』 為建置基礎的僅安全更新
- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 2.0.11 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.2.1

操作員通道
v2.0

.spec.version 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.10 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.10 及 4.12。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 3.19

變更的內容

- 以 第 52 頁的『IBM MQ Operator 2.0.0』 為建置基礎的僅安全更新
- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 2.0.10 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.2.1

操作員通道
v2.0

.spec.version 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.10 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.10 及 4.12。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 3.19

變更的內容

- 以 第 52 頁的『IBM MQ Operator 2.0.0』 為建置基礎的僅安全更新
- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 2.0.9 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.2.1

操作員通道
v2.0

.spec.version 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.10 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.10 及 4.12。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 3.19

新增功能

- 從 2023 年 3 月開始， IBM MQ Operator 和 IBM MQ 佇列管理程式儲存器映像檔會進行數位簽署。 IBM MQ Operator 2.0.9 及 IBM MQ 9.3.0.4-r2 映像檔已使用此版本簽署。 請參閱 第 73 頁的『驗證映像檔簽章』)

變更的內容

- 以 第 52 頁的『IBM MQ Operator 2.0.0』 為建置基礎的僅安全更新
- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 2.0.8 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.2.1

操作員通道
`v2.0`

`.spec.version` 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.10 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.10 及 4.12。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 3.19

變更的內容

- 以 第 52 頁的『IBM MQ Operator 2.0.0』 為建置基礎的僅安全更新
- 這些安全公告中詳述已解決的漏洞:
  - CVE-2022-47629 和 CVE-2022-35737 的公佈欄
  - CVE-2023-26284

IBM MQ Operator 2.0.7 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.2.1

操作員通道
`v2.0`

`.spec.version` 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.10 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.10 及 4.12。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 3.19

變更的內容

- 以 第 52 頁的『IBM MQ Operator 2.0.0』 為建置基礎的僅安全更新
- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 2.0.6 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.2.1

操作員通道
`v2.0`

`.spec.version` 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.10 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.10 及 4.12。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 3.19

變更的內容

- 以 第 52 頁的『IBM MQ Operator 2.0.0』 為建置基礎的僅安全更新
- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 2.0.5 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.2.1

操作員通道
`v2.0`

`.spec.version` 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.10 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.10 及 4.12。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 3.19

變更的內容

- 以 第 52 頁的『IBM MQ Operator 2.0.0』 為建置基礎的僅安全更新
- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 2.0.4

CP4I-LTS

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.2.1

操作員通道
`v2.0`

`.spec.version` 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1, 9.3.0.1-r2

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.10 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.10 及 4.12。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 3.19

變更的內容

- 以 第 52 頁的『IBM MQ Operator 2.0.0』 為建置基礎的僅安全更新
- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 2.0.3

CP4I-LTS CD

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.2.1

操作員通道
`v2.0`

`.spec.version` 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.0.6-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-r1

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.10 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.10 及 4.12。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 3.19

變更的內容

- 以 第 52 頁的『IBM MQ Operator 2.0.0』 為建置基礎的僅安全更新
- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 2.0.2

CP4I-LTS CD

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.2.1

操作員通道
`v2.0`

`.spec.version` 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.0.6-r2-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.10 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.10 及 4.12。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 3.19

變更的內容

- 以 第 52 頁的『IBM MQ Operator 2.0.0』 為建置基礎的僅安全更新
- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 2.0.1

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.2.1

操作員通道
`v2.0`

`.spec.version` 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.0.6-r1-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1, 9.3.0.0-r2

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.10 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.10 及 4.12。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 3.19

變更的內容

- 以 第 52 頁的『IBM MQ Operator 2.0.0』 為建置基礎的僅安全更新
- 此 Security Bulletin 中詳述已解決的漏洞。

IBM MQ Operator 2.0.0

IBM Cloud Pak for Integration 版本
IBM Cloud Pak for Integration 2022.2.1

操作員通道
`v2.0`

`.spec.version` 容許的值
9.2.0.1-r1-eus, 9.2.0.2-r1-eus, 9.2.0.2-r2-eus, 9.2.0.4-r1-eus, 9.2.0.5-r1-eus, 9.2.0.5-r2-eus, 9.2.0.5-r3-eus, 9.2.3.0-r1, 9.2.4.0-r1, 9.2.5.0-r1, 9.2.5.0-r2, 9.2.5.0-r3, 9.3.0.0-r1

Red Hat OpenShift Container Platform 版本
OpenShift Container Platform 4.10 及以上版本。 **附註**: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版次，即偶數次要版次，例如 4.10 及 4.12。

IBM Cloud Pak foundational services 版本
IBM Cloud Pak foundational services 3.19

新增功能

- 新增 IBM MQ 9.3.0。
- 新增 POWER (ppc64le) 的支援。

變更的內容

- 現在需要 Red Hat OpenShift Container Platform 4.10 。 請參閱 第 10 頁的『IBM MQ Operator 的版本支援』。
- Deprecated 已淘汰版本: IBM MQ 9.2.3。 未來版本的 IBM MQ Operator 可能不會核對這些版本。
- Removed 已移除 (先前已淘汰) 持續交付版本: IBM MQ 9.1.5、 9.2.0 CD、 9.2.1、 9.2.2
- 現在 Operator Lifecycle Manager (OLM) 已安裝 IBM MQ Operator 驗證 Web 連結鉤。 OLM 現在會管理 Web 連結鉤的憑證。
- 已修正先前在 IBM MQ Console 記載中產生使用者喜好設定警告的錯誤。
- 這些「安全佈告欄」中詳述已解決的漏洞:
  - https://www.ibm.com/support/pages/node/6602255
  - https://www.ibm.com/support/pages/node/6602259

OpenShift CP4I-LTS CP4I CD 與 IBM MQ Operator 搭配使用的佇列管理程式儲存器映像檔的版本歷程

註: 如需舊版「佇列管理程式儲存器」映像檔的相關資訊，請參閱 IBM MQ 9.2 說明文件中的 IBM MQ Operator 的發行歷程 。

9.3.5.1-r2

CD

必要的操作員版本
3.1.3 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.5.1-r2`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.5.1-r2`
- `icr.io/ibm-messaging/mq:9.3.5.1-r2`

新增功能

- IBM MQ 9.3.5 的新增功能

變更的內容

- IBM MQ 9.3.5 中的變更內容
- 根據 Red Hat Universal Base Image 8.9-1161.1715068733
- golang.org/x/net 程式庫已升級，以重新修補所報告的漏洞

9.3.5.1-r1

CD

必要的操作員版本
3.1.2 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.5.1-r1`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.5.1-r1`

- icr.io/ibm-messaging/mq:9.3.5.1-r1

新增功能

- IBM MQ 9.3.5 的新增功能

變更的內容

- IBM MQ 9.3.5 中的變更內容
- 以 Red Hat Universal Base 映像檔 8.9-1161 為基礎
- 已解決 "dependabot" 報告的安全漏洞

9.3.5.0-r2

CD

必要的操作員版本
3.1.1 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.5.0-r2
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.5.0-r2
- icr.io/ibm-messaging/mq:9.3.5.0-r2

新增功能

- IBM MQ 9.3.5 的新增功能

變更的內容

- IBM MQ 9.3.5 中的變更內容
- 以 Red Hat Universal Base 映像檔 8.9-1137 為基礎
- 如果您已啟用「作業儀表板」，則只需要擷取新的 9.3.5.0-r2 映像檔。

9.3.5.0-r1

CD

必要的操作員版本
3.1.0 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.5.0-r1
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.5.0-r1
- icr.io/ibm-messaging/mq:9.3.5.0-r1

新增功能

- IBM MQ 9.3.5 的新增功能

變更的內容

- IBM MQ 9.3.5 中的變更內容
- 以 Red Hat Universal Base 映像檔 8.9-1137 為基礎
- 提供給 /var/mam 的符號鏈結將在其中複製 mqwebuser.xml 中未加密的認證。
- golang.org/x/crypto 程式庫已升級以重新修補 CVE-2023-48795 漏洞。
- 使用更安全的 SHA512 演算法 (而非 SHA256)，以在 Web 金鑰儲存庫中建立自簽憑證。

- 現在，與 IBM MQ Web 伺服器搭配使用的 PKCS#12 金鑰儲存庫是使用 **Pkcs12.Modern.Encode** 函數產生的，該函數使用 SHA-2 加密 (先前使用舊式 SHA-1 加密產生)。
- 已修正 **PathTraversal** 方法用法所報告的漏洞。

9.3.4.1-r1

必要的操作員版本
3.0.1 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.4.1-r1`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.4.1-r1`
- `icr.io/ibm-messaging/mq:9.3.4.1-r1`

新增功能

- IBM MQ 9.3.4 的新增功能

變更的內容

- IBM MQ 9.3.4 中的變更內容
- 以 Red Hat Universal Base 映像檔 8.9-1108 為基礎

9.3.4.0-r1

必要的操作員版本
3.0.0 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.4.0-r1`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.4.0-r1`
- `icr.io/ibm-messaging/mq:9.3.4.0-r1`

新增功能

- IBM MQ 9.3.4 的新增功能

變更的內容

- IBM MQ 9.3.4 中的變更內容
- 以 Red Hat Universal Base 映像檔 8.9-1029 為基礎
- 已改良對 IBM MQ Web 伺服器的支援-依預設， IBM MQ Web 伺服器日誌現在會出現在儲存器日誌中。 Web 伺服器的 `messages.log` 檔案現在會自動鏡映至儲存器日誌輸出。 作為此變更的一部分，現在寫入磁碟的 `messages.log` 檔案一律為 JSON 格式，不過儲存器日誌仍以 JSON 或人類可讀的 "basic" 格式提供。
- 已修正佇列管理程式儲存器映像檔內的信號處理，以便在啟動完成之前由 Red Hat OpenShift Container Platform 終止儲存器時正確處理控制信號。

9.3.3.3-r2

必要的操作員版本
2.4.8 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.3.3-r2
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.3.3-r2
- icr.io/ibm-messaging/mq:9.3.3.3-r2

新增功能

- IBM MQ 9.3.3 的新增功能

變更的內容

- IBM MQ 9.3.3 中的變更內容
- 以 Red Hat Universal Base 映像檔 8.9-1137 為基礎
- golang.org/x/crypto 程式庫已升級以重新修補 CVE-2023-48795 漏洞。
- 使用更安全的 SHA512 演算法 (而非 SHA256)， 以在 Web 金鑰儲存庫中建立自簽憑證。
- 現在，與 IBM MQ Web 伺服器搭配使用的 PKCS#12 金鑰儲存庫是使用 **Pkcs12.Modern.Encode** 函數產生的，該函數使用 SHA-2 加密 (先前使用舊式 SHA-1 加密產生)。
- 已修正 **PathTraversal** 方法用法所報告的漏洞。

9.3.3.3-r1

必要的操作員版本
2.4.7 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.3.3-r1
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.3.3-r1
- icr.io/ibm-messaging/mq:9.3.3.3-r1

新增功能

- IBM MQ 9.3.3 的新增功能

變更的內容

- IBM MQ 9.3.3 中的變更內容
- 以 Red Hat Universal Base 映像檔 8.9-1108 為基礎

包括 IBM MQ APAR

- IT44961
- IT44821
- IT44954

9.3.3.2-r3

CD

必要的操作員版本
2.4.6 或更新版本

支援的架構
amd64, s390x, ppc64le

映像檔

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.3.2-r3
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.3.2-r3
- icr.io/ibm-messaging/mq:9.3.3.2-r3

新增功能

- IBM MQ 9.3.3 的新增功能

變更的內容

- IBM MQ 9.3.3 中的變更內容
- 以 Red Hat Universal Base 映像檔 8.9-1029 為基礎

9.3.3.2-r2

CD

必要的操作員版本
2.4.5 或更新版本

支援的架構
amd64, s390x, ppc64le

映像檔

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.3.2-r2
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.3.2-r2
- icr.io/ibm-messaging/mq:9.3.3.2-r2

新增功能

- IBM MQ 9.3.3 的新增功能

變更的內容

- IBM MQ 9.3.3 中的變更內容
- 根據 Red Hat 通用基本映像檔 8.8-1072.1697626218
- IBM MQ 佇列管理程式儲存器映像檔 9.3.3.2-r2 包含 Instana MQ 結束程式 3.1.7 版 (2023.4.0)。

9.3.3.2-r1

CD

必要的操作員版本
2.4.4 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.3.2-r1
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.3.2-r1
- icr.io/ibm-messaging/mq:9.3.3.2-r1

新增功能

- IBM MQ 9.3.3 的新增功能

變更的內容
[5]

[5] 這個主題的舊版錯誤地指出 IBM MQ 佇列管理程式儲存器映像檔 9.3.3.2-r1 包含 3.1.7 版 (2023.4.0) Instana MQ 結束程式。

- IBM MQ 9.3.3 中的變更內容
- 根據 Red Hat 通用基本映像檔 8.8-1072.1697626218
- 將 libcurl 的層次更新為 8.4.0。

包括 IBM MQ APAR

- IT41871
- IT44585
- IT44623
- IT44762

9.3.3.1-r2

CD

必要的操作員版本
2.4.3 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.3.1-r2`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.3.1-r2`
- `icr.io/ibm-messaging/mq:9.3.3.1-r2`

新增功能

- IBM MQ 9.3.3 的新增功能

變更的內容

- IBM MQ 9.3.3 中的變更內容
- 以 IBM MQ 9.3.3.1-r1 為建置基礎的僅安全更新
- 以 Red Hat Universal Base 映像檔 8.8-1037 為基礎

9.3.3.1-r1

CD

必要的操作員版本
2.4.2 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.3.1-r1`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.3.1-r1`
- `icr.io/ibm-messaging/mq:9.3.3.1-r1`

新增功能

- IBM MQ 9.3.3 的新增功能

變更的內容

- IBM MQ 9.3.3 中的變更內容
- 以 Red Hat Universal Base Image 8.8-1037 為基礎。

9.3.3.0-r2

CD

必要的操作員版本
2.4.1 或更新版本

支援的架構
amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.3.0-r2`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.3.0-r2`
- `icr.io/ibm-messaging/mq:9.3.3.0-r2`

新增功能

- IBM MQ 9.3.3 的新增功能

變更的內容

- IBM MQ 9.3.3 中的變更內容
- 以 Red Hat Universal Base 映像檔 8.8-1014 為基礎。

9.3.3.0-r1

CD

必要的操作員版本
2.4.0 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.3.0-r1`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.3.0-r1`
- `icr.io/ibm-messaging/mq:9.3.3.0-r1`

新增功能

- IBM MQ 9.3.3 的新增功能

變更的內容

- IBM MQ 9.3.3 中的變更內容
- 以 Red Hat Universal Base 映像檔 8.8-860 為基礎。
- IBM MQ 佇列管理程式儲存器映像檔 9.3.3.0-r1 包含 3.1.2 版 (2023.2.0) Instana MQ 結束程式。

9.3.2.1-r2

CD

必要的操作員版本
2.3.3 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.2.1-r2`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.2.1-r2`
- `icr.io/ibm-messaging/mq:9.3.2.1-r2`

新增功能

- IBM MQ 9.3.2 的新增功能

變更的內容

- IBM MQ 9.3.2 中的變更內容
- 以 Red Hat Universal Base Image 8.7-1107 為基礎。

9.3.2.1-r1

CD

必要的操作員版本
2.3.2 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.2.1-r1`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.2.1-r1`
- `icr.io/ibm-messaging/mq:9.3.2.1-r1`

新增功能

- IBM MQ 9.3.2 的新增功能

變更的內容

- IBM MQ 9.3.2 中的變更內容
- 以 Red Hat Universal Base Image 8.7-1107 為基礎。

9.3.2.0-r2

CD

必要的操作員版本
2.3.1 或更新版本

支援的架構
amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.2.0-r2`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.2.0-r2`
- `icr.io/ibm-messaging/mq:9.3.2.0-r2`

新增功能

- IBM MQ 9.3.2 的新增功能

變更的內容

- IBM MQ 9.3.2 中的變更內容
- 以 Red Hat Universal Base Image 8.7-1085 為基礎。

9.3.2.0-r1

CD

必要的操作員版本
2.3.0 或更新版本

支援的架構
amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.2.0-r1`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.2.0-r1`
- `icr.io/ibm-messaging/mq:9.3.2.0-r1`

新增功能

- IBM MQ 9.3.2 的新增功能
- 現在已設定環境變數 *MQ_LOGGING_CONSOLE_FORMAT*，這會取代已淘汰的 *LOG_FORMAT* 變數。

變更的內容

- IBM MQ 9.3.2 中的變更內容
- 不支援與發證者 (CA) 憑證具有相同「主旨識別名稱 (DN)」的佇列管理程式憑證。 憑證必須具有唯一的「主旨識別名稱」。
- 以 Red Hat 通用基本映像檔 8.7-1049.1675784874 為基礎。

9.3.1.1-r1

CD

必要的操作員版本
2.2.2 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.1.1-r1`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.1.1-r1`
- `icr.io/ibm-messaging/mq:9.3.1.1-r1`

新增功能

- IBM MQ 9.3.1 的新增功能

變更的內容

- IBM MQ 9.3.1 中的變更內容
- 以 Red Hat Universal Base Image 8.7-1031 為基礎。
- IBM MQ 佇列管理程式儲存器映像檔 9.3.1.1-r1 包括 IBM Instana MQ 結束程式 2.4.3 版 (2022.4.3)。

9.3.1.0-r3

CD

必要的操作員版本
2.2.1 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.1.0-r3`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.1.0-r3`
- `icr.io/ibm-messaging/mq:9.3.1.0-r3`

新增功能

- IBM MQ 9.3.1 的新增功能

變更的內容

- IBM MQ 9.3.1 中的變更內容
- 以 Red Hat 通用基本映像檔 8.7-923.1669829893 為基礎。
- IBM MQ 佇列管理程式儲存器映像檔 9.3.1.0-r3 包括 2.4.3 版 (2022.4.3) IBM Instana MQ 結束程式。

9.3.1.0-r2

CD

必要的操作員版本
2.2.0 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.1.0-r2`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.1.0-r2`
- `icr.io/ibm-messaging/mq:9.3.1.0-r2`

新增功能

- IBM MQ 9.3.1 的新增功能
- 從 9.3.1.0-r2 (CD) IBM MQ 佇列管理程式儲存器映像檔中，原生支援 IBM Instana 追蹤。 IBM MQ 9.3.1.0-r2 版包括 2.4.0 版 (2022.4.0) IBM Instana MQ 結束程式。 若要啟用 IBM Instana 追蹤，請參閱 第 143 頁的『整合 IBM MQ 與 IBM Instana 追蹤』。

變更的內容

- IBM MQ 9.3.1 中的變更內容
- 以 Red Hat Universal Base Image 8.7-923 為基礎。
- 如果未提供金鑰和憑證，佇列管理程式屬性 **SSLKEYR** 現在會設為空白，而不是設為 "`/run/runmqserver/tls/key`"。

9.3.1.0-r1

CD

必要的操作員版本
2.1.0 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.1.0-r1`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.1.0-r1`
- `icr.io/ibm-messaging/mq:9.3.1.0-r1`

新增功能

- IBM MQ 9.3.1 的新增功能

變更的內容

- IBM MQ 9.3.1 中的變更內容
- 以 Red Hat Universal Base 映像檔 8.6-941 為基礎。

9.3.0.17-r3

CP4I-LTS

必要的操作員版本

2.0.22 或更高版本

支援的架構

amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.17-r3`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.17-r3`
- `icr.io/ibm-messaging/mq:9.3.0.17-r3`

新增功能

- IBM MQ 9.3.0 的新增功能

變更的內容

- IBM MQ 9.3.0 中的變更內容
- 以 IBM MQ 9.3.0.0-r1 為建置基礎的僅安全更新
- 以 Red Hat 通用基本映像檔 9.4-949.1716471857 為基礎

9.3.0.17-r2

CP4I-LTS

必要的操作員版本

2.0.22 或更高版本

支援的架構

amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.17-r2`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.17-r2`
- `icr.io/ibm-messaging/mq:9.3.0.17-r2`

新增功能

- IBM MQ 9.3.0 的新增功能

變更的內容

- IBM MQ 9.3.0 中的變更內容
- 以 IBM MQ 9.3.0.0-r1 為建置基礎的安全更新
- 根據 Red Hat Universal Base Image 8.9-1161.1715068733
- golang.org/x/net 程式庫已升級，以重新修補所報告的漏洞

9.3.0.17-r1

CP4I-LTS

必要的操作員版本

2.0.21 或更新版本

支援的架構

amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.17-r1`

- cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.17-r1
- icr.io/ibm-messaging/mq:9.3.0.17-r1

新增功能

- IBM MQ 9.3.0 的新增功能

變更的內容

- IBM MQ 9.3.0 中的變更內容
- 以 IBM MQ 9.3.0.0-r1 為建置基礎的安全更新
- 以 Red Hat Universal Base 映像檔 8.9-1161 為基礎
- 已解決 "dependabot" 報告的安全漏洞。

9.3.0.16-r2

CP4I-LTS

必要的操作員版本
2.0.20 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.16-r2
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.16-r2
- icr.io/ibm-messaging/mq:9.3.0.16-r2

新增功能

- IBM MQ 9.3.0 的新增功能

變更的內容

- IBM MQ 9.3.0 中的變更內容
- 以 IBM MQ 9.3.0.0-r1 為建置基礎的安全更新
- 以 Red Hat Universal Base 映像檔 8.9-1137 為基礎
- 如果您已啟用「作業儀表板」，則只需要擷取新的 9.3.0.16-r2 映像檔。

9.3.0.16-r1

CP4I-LTS

必要的操作員版本
2.0.19 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.16-r1
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.16-r1
- icr.io/ibm-messaging/mq:9.3.0.16-r1

新增功能

- IBM MQ 9.3.0 的新增功能

變更的內容

- IBM MQ 9.3.0 中的變更內容
- 以 IBM MQ 9.3.0.0-r1 為建置基礎的安全更新

- 以 Red Hat Universal Base 映像檔 8.9-1137 為基礎
- golang.org/x/crypto 程式庫已升級以重新修補 CVE-2023-48795 漏洞。
- 使用更安全的 SHA512 演算法 (而非 SHA256)，以在 Web 金鑰儲存庫中建立自簽憑證。
- 現在，與 IBM MQ Web 伺服器搭配使用的 PKCS#12 金鑰儲存庫是使用 **Pkcs12.Modern.Encode** 函數產生的，該函數使用 SHA-2 加密 (先前使用舊式 SHA-1 加密產生)。
- 已修正 **PathTraversal** 方法用法所報告的漏洞。

9.3.0.15-r1

CP4I-LTS

必要的操作員版本
2.0.18 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.15-r1`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.15-r1`
- `icr.io/ibm-messaging/mq:9.3.0.15-r1`

新增功能

- IBM MQ 9.3.0 的新增功能

變更的內容

- IBM MQ 9.3.0 中的變更內容
- 在 IBM MQ 9.3.0.0-r1 上建置的僅安全更新
- 以 Red Hat Universal Base 映像檔 8.9-1108 為基礎

9.3.0.11-r2

CP4I-LTS

必要的操作員版本
2.0.17 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.11-r2`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.11-r2`
- `icr.io/ibm-messaging/mq:9.3.0.11-r2`

新增功能

- IBM MQ 9.3.0 的新增功能

變更的內容

- IBM MQ 9.3.0 中的變更內容
- 在 IBM MQ 9.3.0.0-r1 上建置的僅安全更新
- 以 Red Hat Universal Base Image 8.9-1029 為基礎。

9.3.0.11-r1

CP4I-LTS

必要的操作員版本
2.0.16 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.11-r1`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.11-r1`
- `icr.io/ibm-messaging/mq:9.3.0.11-r1`

新增功能

- IBM MQ 9.3.0 的新增功能

變更的內容

- IBM MQ 9.3.0 中的變更內容
- 在 IBM MQ 9.3.0.0-r1 上建置的僅安全更新
- 以 Red Hat Universal Base Image 8.8-1072.1697626218 為基礎。
- 將 libcurl 的層次更新至 8.4.0

9.3.0.10-r2

CP4I-LTS

必要的操作員版本
2.0.15 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.10-r2`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.10-r2`
- `icr.io/ibm-messaging/mq:9.3.0.10-r2`

新增功能

- IBM MQ 9.3.0 的新增功能

變更的內容

- IBM MQ 9.3.0 中的變更內容
- 在 IBM MQ 9.3.0.0-r1 上建置的僅安全更新
- 以 Red Hat Universal Base Image 8.8-1037 為基礎。

9.3.0.10-r1

CP4I-LTS

必要的操作員版本
2.0.14 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.10-r1`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.10-r1`
- `icr.io/ibm-messaging/mq:9.3.0.10-r1`

新增功能

- IBM MQ 9.3.0 的新增功能

變更的內容

- IBM MQ 9.3.0 中的變更內容
- 在 IBM MQ 9.3.0.0-r1 上建置的僅安全更新
- 以 Red Hat Universal Base Image 8.8-1037 為基礎。

9.3.0.6-r1

CP4I-LTS

必要的操作員版本
2.0.13 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.6-r1`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.6-r1`
- `icr.io/ibm-messaging/mq:9.3.0.6-r1`

新增功能

- IBM MQ 9.3.0 的新增功能

變更的內容

- IBM MQ 9.3.0 中的變更內容
- 在 IBM MQ 9.3.0.0-r1 上建置的僅安全更新
- 以 Red Hat Universal Base 映像檔 8.8-1014 為基礎。

9.3.0.5-r3

CP4I-LTS

必要的操作員版本
2.0.12 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.5-r3`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.5-r3`
- `icr.io/ibm-messaging/mq:9.3.0.5-r3`

新增功能

- IBM MQ 9.3.0 的新增功能

變更的內容

- IBM MQ 9.3.0 中的變更內容
- 在 IBM MQ 9.3.0.0-r1 上建置的僅安全更新
- 以 Red Hat Universal Base 映像檔 8.8-860 為基礎。

9.3.0.5-r2

CP4I-LTS

必要的操作員版本
2.0.11 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.5-r2`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.5-r2`
- `icr.io/ibm-messaging/mq:9.3.0.5-r2`

新增功能

- IBM MQ 9.3.0 的新增功能

變更的內容

- IBM MQ 9.3.0 中的變更內容
- 在 IBM MQ 9.3.0.0-r1 上建置的僅安全更新
- 以 Red Hat Universal Base Image 8.7-1107 為基礎。

重要: 適用於 IBM MQ LTS 佇列管理程式儲存器映像檔 9.3.0.5-r2 上作業儀表板的使用者

啟用「作業儀表板」時， IBM MQ LTS 佇列管理程式儲存器映像檔 9.3.0.5-r2 會部署作業儀表板代理程式及收集器映像檔，這些映像檔不包含通用版時可用的最新安全修正程式。

緩解: 升級至至少 9.3.0.5-r3 所有 IBM MQ LTS 佇列管理程式儲存器 9.3.0.5-r2 映像檔，並啟用「作業儀表板」。 請參閱第 120 頁的『使用 Red Hat OpenShift 升級 IBM MQ 佇列管理程式』。

9.3.0.5-r1

CP4I-LTS

必要的操作員版本
2.0.10 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.5-r1`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.5-r1`
- `icr.io/ibm-messaging/mq:9.3.0.5-r1`

新增功能

- IBM MQ 9.3.0 的新增功能

變更的內容

- IBM MQ 9.3.0 中的變更內容
- 在 IBM MQ 9.3.0.0-r1 上建置的僅安全更新
- 以 Red Hat Universal Base Image 8.7-1107 為基礎。

重要: 適用於 IBM MQ LTS 佇列管理程式儲存器映像檔 9.3.0.5-r1 上作業儀表板的使用者

當啟用「作業儀表板」時， IBM MQ LTS 佇列管理程式儲存器映像檔 9.3.0.5-r1 會部署作業儀表板代理程式及收集器映像檔，這些映像檔不包含通用版時可用的最新安全修正程式。

緩解: 升級至至少已啟用「作業儀表板」的 9.3.0.5-r3 所有 IBM MQ LTS 佇列管理程式儲存器 9.3.0.5-r1 映像檔。 請參閱第 120 頁的『使用 Red Hat OpenShift 升級 IBM MQ 佇列管理程式』。

9.3.0.4-r2

CP4I-LTS

必要的操作員版本

2.0.9 或更高版本

支援的架構

amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.4-r2`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.4-r2`
- `icr.io/ibm-messaging/mq:9.3.0.4-r2`

新增功能

- IBM MQ 9.3.0 的新增功能

變更的內容

- IBM MQ 9.3.0 中的變更內容
- 在 IBM MQ 9.3.0.0-r1 上建置的僅安全更新
- 以 Red Hat Universal Base Image 8.7-1085 為基礎。

9.3.0.4-r1

CP4I-LTS

必要的操作員版本

2.0.8 或更高版本

支援的架構

amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.4-r1`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.4-r1`
- `icr.io/ibm-messaging/mq:9.3.0.4-r1`

新增功能

- IBM MQ 9.3.0 的新增功能

變更的內容

- IBM MQ 9.3.0 中的變更內容
- 在 IBM MQ 9.3.0.0-r1 上建置的僅安全更新
- 以 Red Hat 通用基本映像檔 8.7-1049.1675784874 為基礎。

9.3.0.3-r1

CP4I-LTS

必要的操作員版本

2.0.7 或更高版本

支援的架構

amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.3-r1`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.3-r1`

- icr.io/ibm-messaging/mq:9.3.0.3-r1

新增功能

- IBM MQ 9.3.0 的新增功能

變更的內容

- IBM MQ 9.3.0 中的變更內容
- 在 IBM MQ 9.3.0.0-r1 上建置的僅安全更新
- 以 Red Hat Universal Base Image 8.7-1031 為基礎。

9.3.0.1-r4

CP4I-LTS

必要的操作員版本
2.0.6 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.1-r4
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.1-r4
- icr.io/ibm-messaging/mq:9.3.0.1-r4

新增功能

- IBM MQ 9.3.0 的新增功能

變更的內容

- IBM MQ 9.3.0 中的變更內容
- 在 IBM MQ 9.3.0.0-r1 上建置的僅安全更新
- 以 Red Hat 通用基本映像檔 8.7-923.1669829893 為基礎。

9.3.0.1-r3

CP4I-LTS

必要的操作員版本
2.0.5 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.1-r3
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.1-r3
- icr.io/ibm-messaging/mq:9.3.0.1-r3

新增功能

- IBM MQ 9.3.0 的新增功能

變更的內容

- IBM MQ 9.3.0 中的變更內容
- 在 IBM MQ 9.3.0.0-r1 上建置的僅安全更新
- 以 Red Hat Universal Base Image 8.7-923 為基礎。

9.3.0.1-r2

CP4I-LTS

必要的操作員版本

2.0.4 或更高版本

支援的架構

amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.1-r2`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.1-r2`
- `icr.io/ibm-messaging/mq:9.3.0.1-r2`

新增功能

- IBM MQ 9.3.0 的新增功能

變更的內容

- IBM MQ 9.3.0 中的變更內容
- 在 IBM MQ 9.3.0.0-r1 上建置的僅安全更新
- 以 Red Hat Universal Base 映像檔 8.6-941 為基礎。

9.3.0.1-r1

CP4I-LTS CD

必要的操作員版本

2.0.3 或更高版本

支援的架構

amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.1-r1`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.1-r1`
- `icr.io/ibm-messaging/mq:9.3.0.1-r1`

新增功能

- IBM MQ 9.3.0 的新增功能

變更的內容

- IBM MQ 9.3.0 中的變更內容
- 在 IBM MQ 9.3.0.0-r1 上建置的僅安全更新
- 以 Red Hat Universal Base 映像檔 8.6-941 為基礎。

9.3.0.0-r3

CP4I-LTS CD

必要的操作員版本

2.0.2 或更高版本

支援的架構

amd64, s390x, ppc64le

映像檔

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.0-r3`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.0-r3`

- icr.io/ibm-messaging/mq:9.3.0.0-r3

新增功能

- IBM MQ 9.3.0 的新增功能

變更的內容

- IBM MQ 9.3.0 中的變更內容
- 在 IBM MQ 9.3.0.0-r1 上建置的僅安全更新
- 以 Red Hat Universal Base Image 8.6-902 為基礎。

9.3.0.0-r2

必要的操作員版本
2.0.1 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.0-r2
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.0-r2
- icr.io/ibm-messaging/mq:9.3.0.0-r2

新增功能

- IBM MQ 9.3.0 的新增功能

變更的內容

- IBM MQ 9.3.0 中的變更內容
- 在 IBM MQ 9.3.0.0-r1 上建置的僅安全更新
- 以 Red Hat Universal Base 映像檔 8.6-854 為基礎。

9.3.0.0-r1

必要的操作員版本
2.0.0 或更高版本

支援的架構
amd64, s390x, ppc64le

映像檔

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.3.0.0-r1
- cp.icr.io/cp/ibm-mqadvanced-server:9.3.0.0-r1
- icr.io/ibm-messaging/mq:9.3.0.0-r1

新增功能

- IBM MQ 9.3.0 的新增功能

變更的內容

- IBM MQ 9.3.0 中的變更內容
- MQ Advanced for Developers 映像檔中的預設開發人員配置現在使用 ANY_TLS12_OR_HIGHER。
- 已修正 IBM MQ Web 伺服器的問題，該問題導致日誌中發生錯誤，因為遺漏 Java 喜好設定。
- 以 Red Hat Universal Base Image 8.6-751.1655117800 為基礎。

OpenShift CP4I 驗證映像檔簽章

從 2023 年 3 月開始， IBM MQ Operator 和 IBM MQ 佇列管理程式儲存器映像檔會進行數位簽署。

要簽署的第一個 IBM MQ 操作員:

- 2.3.1 (CD)
- 2.0.9 (LTS)

要簽署的第一個 IBM MQ 佇列管理程式儲存器映像檔:

- 9.3.2.0-r2 (CD)
- 9.3.0.4-r2 (LTS)

關於這項作業

數位簽章為內容消費者提供一種方法，可確保他們下載的內容既真實 (源自預期來源)，又具有完整性 (我們預期的內容)。

程序

- 驗證 IBM MQ Operator 及 IBM MQ 佇列管理程式儲存器映像檔的簽章:
  - Operator3.0.0 V 9.3.4 若為 3.0.0 或更新版本的 IBM MQ Operator，或 9.3.4.0-r1 或更新版本的 IBM MQ 佇列管理程式儲存器映像檔，請參閱 IBM Cloud Pak for Integration (CP4I) 2023.4 說明文件中的 驗證映像檔簽章 。
  - Operator2.4.0 如需 2.4.x 中的 IBM MQ Operator，或 9.3.3.x 中的 IBM MQ 佇列管理程式儲存器映像檔，請參閱 CP4I 2023.2 說明文件中的 驗證映像檔簽章 。
  - 如需 2.4.0 之前的 IBM MQ Operator，或 9.3.3.0-r1 之前的 IBM MQ 佇列管理程式儲存器映像檔，請參閱 CP4I 2022.4 說明文件中的 驗證映像檔簽章 。

OpenShift CP4I-LTS CD 將 IBM MQ 移轉至 IBM Cloud Pak for Integration

這組主題說明使用 IBM Cloud Pak for Integration 中的 IBM MQ Operator，將現有 IBM MQ 佇列管理程式移轉至儲存器環境的主要步驟。

關於這項作業

在 Red Hat OpenShift 上部署 IBM MQ 的用戶端可以分成下列實務範例:

1. 在 Red Hat OpenShift 中為新的應用程式建立新的 IBM MQ 部署。
2. 針對 Red Hat OpenShift 中的新應用程式，將 IBM MQ 網路延伸至 Red Hat OpenShift 。
3. 將 IBM MQ 部署移至 Red Hat OpenShift 以繼續支援現有應用程式。

僅適用於實務範例 3，您需要移轉 IBM MQ 配置。 其他實務範例視為新的部署。

這組主題著重於實務範例 3，並說明使用 IBM MQ Operator 將現有 IBM MQ 佇列管理程式移轉至儲存器環境的主要步驟。 由於 IBM MQ 的彈性及廣泛使用，有數個選用步驟。 每一個包括「我是否需要執行此動作」區段。 驗證您的需求應該可以在移轉期間節省您的時間。

您也需要考量要移轉哪些資料:

1. 移轉具有相同配置但沒有任何現有佇列訊息的 IBM MQ 。
2. 移轉具有相同配置及現有訊息的 IBM MQ 。

一般版本至版本移轉可以使用任一方法。 在移轉時的一般 IBM MQ 佇列管理程式中，如果有任何訊息儲存在佇列中，則很少會讓選項 1 適用於許多情況。 在移轉至容器平台的情況下，更常見的情況是使用選項 1，以減少移轉的複雜性並容許藍色綠色部署。 因此，指示著重於此實務範例。

此實務範例的目標是在儲存器環境中建立符合現有佇列管理程式定義的佇列管理程式。 這可讓現有的網路連接應用程式只重新配置成指向新的佇列管理程式，而不變更任何其他配置或應用程式邏輯。

在這項移轉期間，您會產生多個配置檔，以套用至新的佇列管理程式。 若要簡化這些檔案的管理，您應該建立一個目錄，並將它們產生到該目錄中。

程序

1. 第 74 頁的『檢查必要功能是否可用』
2. 第 74 頁的『擷取佇列管理程式配置』
3. 選擇性的: 第 75 頁的『選用項目: 擷取並獲得佇列管理程式金鑰及憑證』
4. 選擇性的: 第 77 頁的『選用項目: 配置 LDAP』
5. 選擇性的: 第 84 頁的『選用項目: 變更 IBM MQ 配置中的 IP 位址和主機名稱』
6. 第 85 頁的『更新儲存器環境的佇列管理程式配置』
7. 第 88 頁的『選取在儲存器中執行之 IBM MQ 的目標 HA 架構』
8. 第 89 頁的『建立佇列管理程式的資源』
9. 第 90 頁的『在 Red Hat OpenShift 上建立新的佇列管理程式』
10. 第 93 頁的『驗證新的儲存器部署』

OpenShift CP4I-LTS CD 檢查必要功能是否可用

IBM MQ Operator 不包含 IBM MQ Advanced 內可用的所有特性，您必須驗證這些特性並非必要。 其他特性部分受支援，並且可以重新配置以符合儲存器中可用的特性。

開始之前

這是 第 73 頁的『將 IBM MQ 移轉至 IBM Cloud Pak for Integration』中的第一個步驟。

程序

1. 請驗證目標儲存器映像檔包含所有必要的功能。

   如需最新資訊，請參閱 第 5 頁的『選擇您要如何在儲存器中使用 IBM MQ』。
2. IBM MQ Operator 具有單一 IBM MQ 資料流量埠，稱為接聽器。 如果您有多個接聽器，請將此簡化為在儲存器中使用單一接聽器。 因為這不是一般實務範例，所以未詳細記載此修改。
3. 如果使用 IBM MQ 結束程式，請在 IBM MQ 結束程式二進位檔中分層將它們移轉至儲存器。 這是進階移轉實務範例，因此不包含在這裡。 如需步驟的大綱，請參閱 第 149 頁的『使用 Red Hat OpenShift CLI 以自訂 MQSC 及 INI 檔案建置映像檔』。
4. 如果 IBM MQ 系統包括「高可用性」，請檢閱可用的選項。

   請參閱 第 22 頁的『儲存器中 IBM MQ 的高可用性』。

下一步

現在您已準備好 擷取佇列管理程式配置。

OpenShift CP4I-LTS CD 擷取佇列管理程式配置

大部分配置在佇列管理程式之間是可攜的。 例如，應用程式與之互動的事物，例如佇列、主題及通道的定義。 請利用這項作業，從現有的 IBM MQ 佇列管理程式擷取配置。

開始之前

這項作業假設您已 檢查必要的功能是否可用。

程序

1. 使用現有的 IBM MQ 安裝來登入機器。

2. 備份配置。

   請執行下列指令：

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

   此指令的使用注意事項：

   - 此指令會將備份儲存在 tmp 目錄中。 您可以將備份儲存在另一個位置，但此實務範例假設後續指令的 tmp 目錄。
   - 將 *QMGR_NAME* 取代為環境中的佇列管理程式名稱。 如果您不確定該值，請執行 **dspmq** 指令，以檢視機器上可用的佇列管理程式。 以下是名為 qm1 之佇列管理程式的 **dspmq** 指令輸出範例：

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

     dspmq 指令需要啟動 IBM MQ 佇列管理程式，否則您會收到下列錯誤：

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

     必要的話，請執行下列指令來啟動佇列管理程式：

     ```
     strmqm QMGR_NAME
     ```

下一步

現在您已準備好 擷取並獲得佇列管理程式金鑰及憑證。

OpenShift CP4I-LTS CD 選用項目: 擷取並獲得佇列管理程式金鑰及憑證

可以使用 TLS 來配置 IBM MQ，以加密進入佇列管理程式的資料流量。 使用此作業來驗證佇列管理程式是否使用 TLS、擷取金鑰及憑證，以及在移轉的佇列管理程式上配置 TLS。

開始之前

這項作業假設您已 擷取佇列管理程式配置。

關於這項作業

我需要這麼做嗎?

IBM MQ 可以配置為加密進入佇列管理程式的資料流量。 此加密是使用佇列管理程式上所配置的金鑰儲存庫來完成。 然後， IBM MQ 通道會啟用 TLS 通訊。 如果您不確定它是否已在環境中配置，請執行下列指令來驗證：

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

如果找不到任何結果，則不會使用 TLS。 不過，這並不表示不應在移轉的佇列管理程式中配置 TLS。 您可能想要變更此行為有數個原因：

- 與前一個環境相比， Red Hat OpenShift 環境上的安全方法應該得到加強。
- 如果您需要從 Red Hat OpenShift 環境外部存取已移轉的佇列管理程式，則需要 TLS 才能通過 Red Hat OpenShift 路徑。

註: V 9.3.2 不支援與發證者 (CA) 憑證具有相同「主旨識別名稱 (DN)」的佇列管理程式憑證。 憑證必須具有唯一的「主旨識別名稱」。 產品現在會檢查 DN 是否不同。

程序

1. 從現有儲存庫中擷取任何授信憑證。

如果目前在佇列管理程式上使用 TLS，則佇列管理程式可能會儲存一些授信憑證。 這些需要擷取並複製到新的佇列管理程式。 請完成下列其中一個選用步驟:

- 若要簡化憑證的擷取，請在本端系統上執行下列 Script:

```
#!/bin/bash

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

執行 Script 時，請指定 IBM MQ 備份的位置作為引數，並擷取憑證。 例如，如果 Script 稱為 `extractCert.sh`，且 IBM MQ 備份位於 `/tmp/backup.mqsc`，則執行下列指令:

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

- 或者，依顯示的順序執行下列指令:

  a. 識別 TLS 儲存庫的位置:

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

  範例輸出:

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

  金鑰儲存庫位於 `/run/runmqserver/tls/key.kdb`

  b. 根據此位置資訊，查詢金鑰儲存庫以判定儲存的憑證:

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

  範例輸出:

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

  c. 擷取每一個列出的憑證。 執行下列指令來執行此動作:

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

  在先前顯示的範例中，這等同於下列:

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

2. 獲得佇列管理程式的新金鑰和憑證

若要在移轉的佇列管理程式上配置 TLS，您可以產生新的金鑰及憑證。 然後會在部署期間使用此項目。 在許多組織中，這表示聯絡您的安全團隊以要求金鑰及憑證。 在部分組織中無法使用此選項，並使用自簽憑證。

下列範例會產生自簽憑證，其中期限設為 10 年:

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

會建立兩個新檔案:

- `qmgr.key` 是佇列管理程式的私密金鑰
- `qmgr.crt` 是公用憑證

下一步

現在您已準備好 配置 LDAP。

OpenShift CP4I-LTS CD 選用項目: 配置 LDAP

IBM MQ Operator 可以配置成使用數種不同的安全方法。 一般而言， LDAP 對企業部署最有效，且 LDAP 用於此移轉實務範例。

開始之前

這項作業假設您已 擷取並獲得佇列管理程式金鑰和憑證。

關於這項作業

我需要這麼做嗎?

如果您已使用 LDAP 進行鑑別及授權，則不需要進行任何變更。

如果您不確定是否正在使用 LDAP，請執行下列指令:

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

範例輸出:

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

輸出中有兩個特別感興趣的屬性:

AUTHTYPE
如果此值為 `IDPWLDAP`，則您是使用 LDAP 進行鑑別。

如果值為空白或另一個值，則不會配置 LDAP。 在此情況下，請檢查 AUTHORMD 屬性，以查看是否使用 LDAP 使用者進行授權。

AUTHORMD
如果此值為 OS，則您不會使用 LDAP 進行授權。

若要修改授權及鑑別以使用 LDAP ，請完成下列作業:

程序

1. 更新 LDAP 伺服器的 IBM MQ 備份。
2. 更新 IBM MQ 備份以取得 LDAP 授權資訊。

OpenShift CP4I-LTS CD

LDAP 第 *1* 部分: 更新 *LDAP* 伺服器的 *IBM MQ* 備份

如何設定 LDAP 的綜合性說明不在此實務範例的範圍內。 本主題提供程序、範例及進一步資訊參照的摘要。

開始之前

這項作業假設您已 擷取並獲得佇列管理程式金鑰和憑證。

關於這項作業

我需要這麼做嗎?

如果您已使用 LDAP 進行鑑別及授權，則不需要進行任何變更。 如果您不確定是否正在使用 LDAP ，請參閱第 77 頁的『選用項目: 配置 LDAP』。

設定 LDAP 伺服器有兩個部分:

1. 定義 LDAP 配置。
2. 建立 LDAP 配置與佇列管理程式定義的關聯。

可協助您使用此配置的進一步資訊:

- 使用者儲存庫概觀
- AUTHINFO 指令參考手冊

程序

1. 定義 LDAP 配置。

   編輯 backup.mqsc 檔案，為 LDAP 系統定義新的 **AUTHINFO** 物件。 例如:

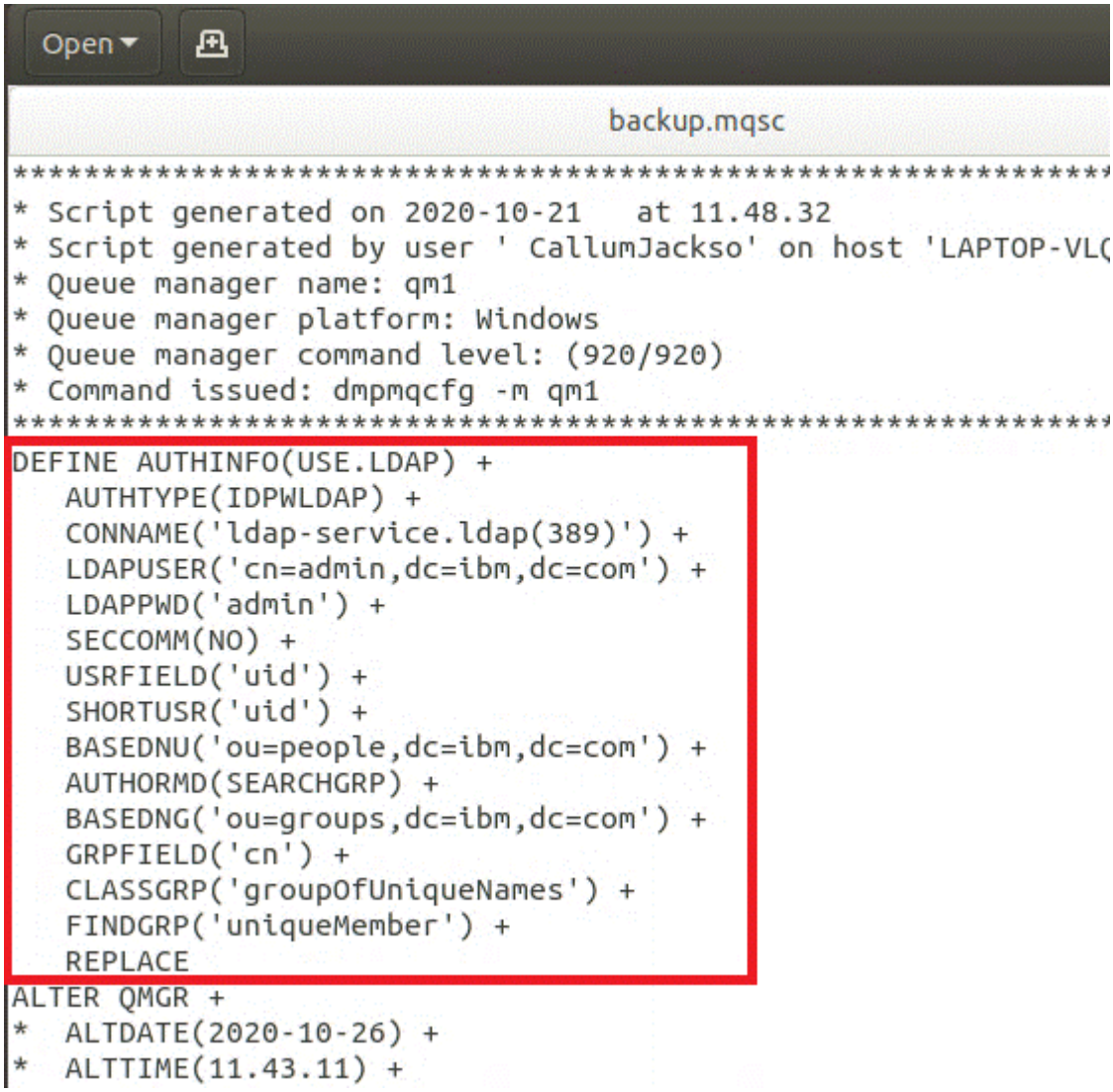
```
DEFINE AUTHINFO(USE.LDAP) +
   AUTHTYPE(IDPWLDAP) +
   CONNAME('ldap-service.ldap(389)') +
   LDAPUSER('cn=admin,dc=ibm,dc=com') +
   LDAPPWD('admin') +
   SECCOMM(NO) +
   USRFIELD('uid') +
   SHORTUSR('uid') +
   BASEDNU('ou=people,dc=ibm,dc=com') +
   AUTHORMD(SEARCHGRP) +
   BASEDNG('ou=groups,dc=ibm,dc=com')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

   其中

   - **CONNAME** 是對應於 LDAP 伺服器的主機名稱及埠。 如果有多個位址用於備援，則可以使用逗點區隔清單來配置這些位址。
   - **LDAPUSER** 是對應於 IBM MQ 在連接至 LDAP 以查詢使用者記錄時所使用之使用者的識別名稱。
   - **LDAPPWD** 是對應於 **LDAPUSER** 使用者的密碼。

- **SECCOM** 指定與 LDAP 伺服器的通訊是否應該使用 TLS。 可能的值如下:
  - YES: 使用 TLS，並由 IBM MQ 伺服器提供憑證。
  - ANON: 使用 TLS 時沒有 IBM MQ 伺服器提供的憑證。
  - NO: 在連線期間不使用 TLS。
- **USRFIELD** 指定 LDAP 記錄中要比對所呈現使用者名稱的欄位。
- **SHORTUSR** 是 LDAP 記錄中長度不超過 12 個字元的欄位。 如果鑑別成功，則此欄位內的值是主張的身分。
- **BASEDNU** 是應該用於搜尋 LDAP 的基本 DN。
- **BASEDNG** 是 LDAP 內群組的基本 DN。
- **AUTHORMD** 定義用來解析使用者群組成員資格的機制。 有四個選項:
  - OS: 查詢作業系統中與簡稱相關聯的群組。
  - SEARCHGRP: 在 LDAP 中搜尋已鑑別使用者的群組項目。
  - SEARCHUSR: 在已鑑別使用者記錄中搜尋群組成員資格資訊。
  - SRCHGRPSN: 在 LDAP 中的群組項目中搜尋已鑑別使用者的簡短使用者名稱 (由 SHORTUSR 欄位定義)。
- **GRPFIELD** 是 LDAP 群組記錄內對應於簡式名稱的屬性。 如果指定的話，這可用來定義授權記錄。
- **CLASSUSR** 是對應於使用者的 LDAP 物件類別。
- **CLASSGRP** 是對應於群組的 LDAP 物件類別。
- **FINDGRP** 是 LDAP 記錄內對應於群組成員資格的屬性。

新項目可以放置在檔案內的任何位置，不過您可能會發現在檔案開頭具有任何新項目會很有用:



2. 將 LDAP 配置與佇列管理程式定義相關聯。

   您需要將 LDAP 配置與佇列管理程式定義相關聯。 緊接在 DEFINE AUTHINFO 項目下方的是 ALTER QMGR 項目。 修改 CONNAUTH 項目以對應新建立的 AUTHINFO 名稱。 例如，在前一個範例中已定義 AUTHINFO(USE.LDAP)，表示名稱為 USE.LDAP。 因此，將 CONNAUTH('SYSTEM.DEFAULT.AUTHINFO.IDPWOS') 變更為 CONNAUTH('USE.LDAP'):

若要立即切換至 LDAP，請在 ALTER QMGR 指令之後立即新增一行，以呼叫 REFRESH SECURITY 指令:

*backup.mqsc

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

下一步

現在，您已準備好 更新 IBM MQ LDAP 授權備份資訊。

OpenShift CP4I-LTS CD

LDAP 第 *2* 部分*:* 更新 *LDAP* 授權資訊的 *IBM MQ* 備份

IBM MQ 提供細部授權規則，可控制對 IBM MQ 物件的存取權。 如果您將鑑別及授權變更為 LDAP，則授權規則可能無效且需要更新。

開始之前

這項作業假設您已 更新 LDAP 伺服器的備份。

關於這項作業

我需要這麼做嗎?

如果您已使用 LDAP 進行鑑別及授權，則不需要進行任何變更。 如果您不確定是否正在使用 LDAP，請參閱第 77 頁的『選用項目: 配置 LDAP』。

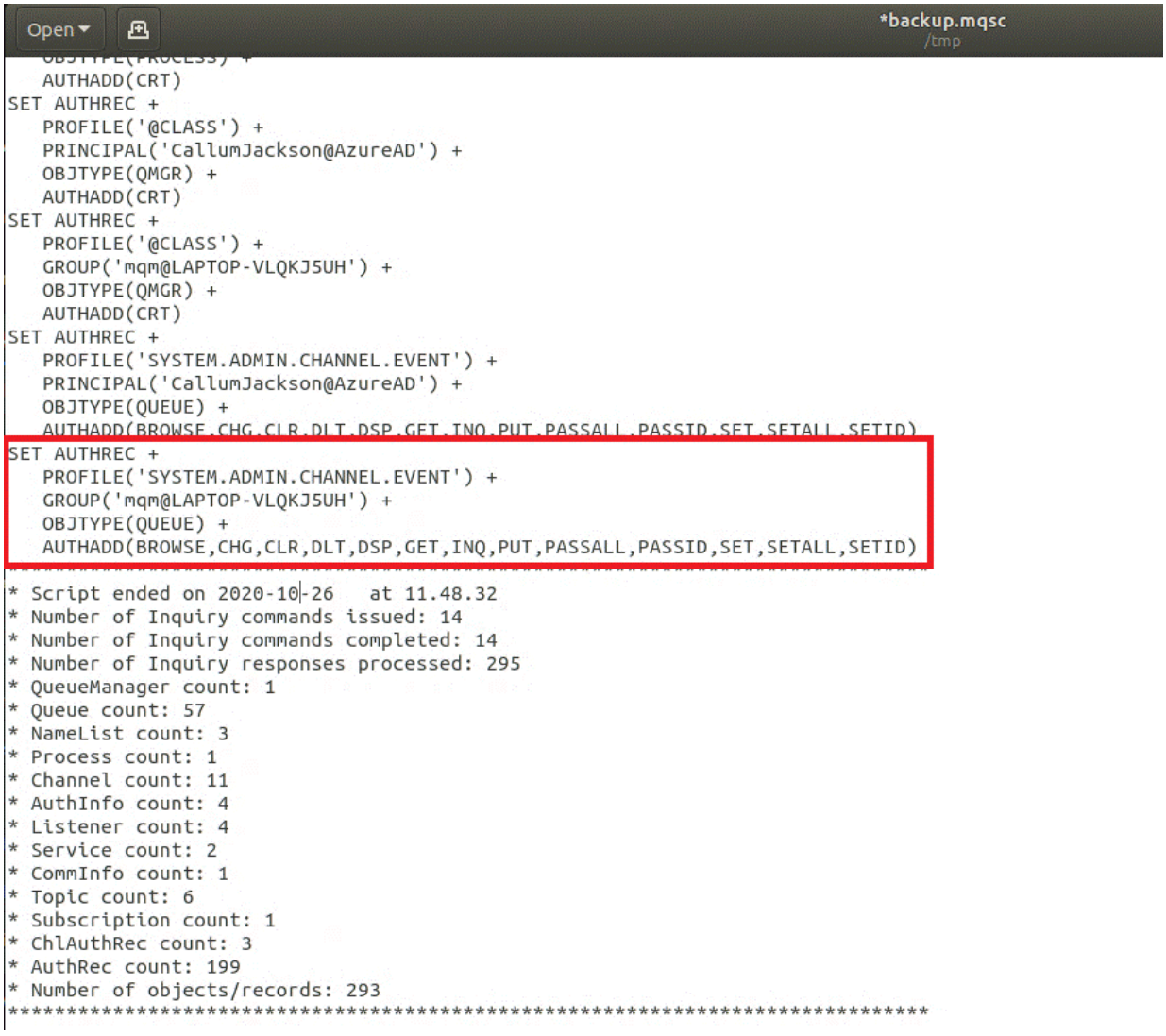
更新 LDAP 授權資訊有兩個部分:

1. 從檔案中移除所有現有的授權。
2. 定義 LDAP 的新授權資訊。

程序

1. 從檔案中移除所有現有的授權。

   在備份檔中，接近檔案結尾時，您應該會看到數個以 SET AUTHREC 開頭的項目:



   尋找現有項目並刪除它們。 最直接明確的方法是移除所有現有的 SET AUTHREC 規則，然後根據 LDAP 項目來建立新項目。
2. 定義 LDAP 的新授權資訊

視您的佇列管理程式配置及資源和群組數目而定，這可能是耗時或直接明確的活動。 下列範例假設您的佇列管理程式只有一個稱為 Q1 的佇列，且您想要容許 LDAP 群組 apps 具有存取權。

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

第一個 AUTHREC 指令會新增存取佇列管理程式的許可權，第二個指令會提供存取佇列的權限。 如果需要存取第二個佇列，則需要第三個 AUTHREC 指令，除非您決定使用萬用字元來提供更通用的存取權。

以下是另一個範例。 如果管理者群組 (稱為 admins) 需要佇列管理程式的完整存取權，請新增下列指令:

```
SET AUTHREC PROFILE('*') OBJTYPE(QUEUE) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(TOPIC) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(CHANNEL) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(CLNTCONN) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(AUTHINFO) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(LISTENER) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(NAMELIST) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(PROCESS) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(SERVICE) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(QMGR) GROUP('admins') AUTHADD(ALL)
```

下一步

現在，您已準備好 變更 IBM MQ 配置中的 IP 位址和主機名稱。

OpenShift CP4I-LTS CD 選用項目: 變更 IBM MQ 配置中的 IP 位址和主機名稱

IBM MQ 配置可能指定了 IP 位址和主機名稱。 在某些狀況下，這些可以保留，而在其他狀況下則需要更新。

開始之前

這項作業假設您已 配置 LDAP。

關於這項作業

我需要這麼做嗎?

首先，除了前一節所定義的 LDAP 配置之外，請判斷您是否已指定任何 IP 位址或主機名稱。 若要執行此動作，請執行下列指令:

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

範例輸出:

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

在此範例中，搜尋會傳回三個結果。 一個結果對應於先前定義的 LDAP 配置。 這可以忽略，因為 LDAP 伺服器的主機名稱仍然相同。 另外兩個結果是空的連線項目，因此也可以忽略這些連線項目。 如果您沒有任何其他項目，則可以跳過本主題的其餘部分。

程序

1. 瞭解傳回的項目。

   IBM MQ 可以在配置的許多方面包含 IP 位址、主機名稱和埠。 我們可以將這些分類為兩個種類:

   a. **此佇列管理程式的位置**: 此佇列管理程式使用或發佈的位置資訊，可供 IBM MQ 網路內的其他佇列管理程式或應用程式用於連線功能。

   b. **佇列管理程式相依關係的位置**: 此佇列管理程式需要知道的其他佇列管理程式或系統的位置。

   因為此實務範例僅聚焦於此佇列管理程式配置的變更，所以我們只會處理種類 (a) 的配置更新。 不過，如果其他佇列管理程式或應用程式參照此佇列管理程式位置，則其配置可能需要更新，以符合此佇列管理程式的新位置。

   有兩個金鑰物件可能包含需要更新的資訊:

   - 接聽器: 這些代表 IBM MQ 正在接聽的網址。
   - 叢集接收端通道: 如果佇列管理程式是 IBM MQ 叢集的一部分，則此物件存在。 它指定其他佇列管理程式可以連接的網址。

2. 在 `grep 'CONNAME\|LOCLADDR\|IPADDRV' -B 3 backup.mqsc` 指令的原始輸出中，識別是否定義了任何 CLUSTER RECEIVER 通道。 如果是，請更新 IP 位址。

   若要識別是否已定義任何 CLUSTER RECEIVER 通道，請在原始輸出中尋找具有 CHLTYPE (CLUSRCVR) 的任何項目:

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

   如果項目確實存在，請使用 IBM MQ Red Hat OpenShift 路徑更新 CONNAME 。 此值基於 Red Hat OpenShift 環境，並使用可預測的語法:

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

   例如，如果在 `cp4i` 名稱空間內佇列管理程式部署命名為 `qm1` ，且 *openshift_app_route_hostname* 為 `apps.callumj.icp4i.com`，則路徑 URL 如下:

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

   路徑的埠號通常是 443。 除非 Red Hat OpenShift 管理者以不同方式告訴您，否則這通常是正確的值。 使用此資訊，更新 CONNAME 欄位。 例如:

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

   在 `grep 'CONNAME\|LOCLADDR\|IPADDRV' -B 3 backup.mqsc` 指令的原始輸出中，驗證 LOCLADDR 或 IPADDRV 是否存在任何項目。 如果有的話，請刪除它們。 它們在儲存器環境中不相關。

下一步

現在，您已準備好 更新儲存器環境的佇列管理程式配置。

OpenShift CP4I-LTS CD 更新儲存器環境的佇列管理程式配置

在儲存器中執行時，儲存器會定義某些配置層面，且可能與匯出的配置衝突。

開始之前

這項作業假設您已 變更 IBM MQ IP 位址和主機名稱的配置。

關於這項作業

儲存器定義下列配置層面:

- 接聽器定義 (對應於公開的埠)。
- 任何潛在 TLS 儲存庫的位置。

因此，您需要更新匯出的配置:

1. 移除任何接聽器定義。
2. 定義 TLS 金鑰儲存庫的位置。

程序

1. 移除任何接聽器定義。

   在備份配置中，搜尋 `DEFINE LISTENER`。 這應該介於 `AUTHINFO` 與 `SERVICE` 定義之間。 強調顯示區域，並刪除它。

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

2. 定義 TLS 金鑰儲存庫的位置。

   佇列管理程式備份包含原始環境的 TLS 配置。 這與儲存器環境不同，因此需要一些更新項目:

   - 將 **CERTLABL** 項目變更為 `default`
   - 將 TLS 金鑰儲存庫 (**SSLKEYR**) 的位置變更為: `/run/runmqserver/tls/key`

   若要尋找檔案中 **SSLKEYR** 屬性的位置，請搜尋 **SSLKEYR**。 通常只會找到一個項目。 如果找到多個項目，請確認您正在編輯 **QMGR** 物件，如下圖所示:

*backup.mqsc

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

下一步

現在，您已準備好 選取在儲存器中執行之 IBM MQ 的目標架構。

OpenShift CP4I-LTS CD 選取在儲存器中執行之 IBM MQ 的目標 HA 架構

在單一實例 (單一 Kubernetes Pod) 與多重實例 (兩個 Pod) 之間進行選擇，以符合您的高可用性需求。

開始之前

此作業假設您已 更新儲存器環境的佇列管理程式配置。

關於這項作業

IBM MQ Operator 提供兩個高可用性選項:

- **單一實例**: 單一容器 (Pod) 已啟動，且在失敗時由 Red Hat OpenShift 負責重新啟動。 由於 Kubernetes 內有狀態集的性質，在數個狀況下，此失效接手可能需要較長的時間，或需要完成管理動作。
- **多重實例**: 啟動兩個容器 (每個位於個別 Pod 中)，一個處於作用中模式，另一個處於待命模式。 此拓蹼可啟用更快速的失效接手。 它需要符合 IBM MQ 需求的「讀寫多次」檔案系統。

在此作業中，您只選擇目標 HA 架構。 在此實務範例的後續作業中，會說明配置您所選擇架構的步驟 (第 90 頁的『在 Red Hat OpenShift 上建立新的佇列管理程式』)。

程序

1. 請檢閱兩個選項。

   如需這兩個選項的綜合性說明，請參閱 第 22 頁的『儲存器中 IBM MQ 的高可用性』。
2. 選取目標 HA 架構。

   如果您不確定要選擇哪個選項，請從 **單一實例** 選項開始，並驗證這是否符合您的高可用性需求。

下一步

現在您已準備好 建立佇列管理程式資源。

OpenShift CP4I-LTS CD 建立佇列管理程式的資源

將 IBM MQ 配置以及 TLS 憑證和金鑰匯入 Red Hat OpenShift 環境。

開始之前

這項作業假設您已 選取在儲存器中執行 IBM MQ 的目標架構。

關於這項作業

在先前的區段中，您已擷取、更新並定義兩個資源:

- IBM MQ 配置
- TLS 憑證及金鑰

在部署佇列管理程式之前，您需要將這些資源匯入至 Red Hat OpenShift 環境。

程序

1. 將 IBM MQ 配置匯入至 Red Hat OpenShift。

   下列指示假設您在現行目錄中具有稱為 `backup.mqsc` 的檔案中的 IBM MQ 配置。 否則，您需要根據環境來自訂檔名。

   a) 使用 `oc login` 登入叢集。

   b) 將 IBM MQ 配置載入至 `configmap`。

   請執行下列指令:

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

   c) 驗證已順利載入檔案。

   請執行下列指令:

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

2. 匯入 IBM MQ TLS 資源

   如 第 75 頁的『選用項目: 擷取並獲得佇列管理程式金鑰及憑證』中所述，佇列管理程式部署可能需要 TLS。 如果是這樣，您應該已有一些檔案以 `.crt` 和 `.key` 結尾。 您需要將這些密鑰新增至 Kubernetes 密鑰，以供佇列管理程式在部署時參照。

   例如，如果您具有佇列管理程式的金鑰及憑證，則可能會呼叫它們:

   - `qmgr.crt`
   - `qmgr.key`

   若要匯入這些檔案，請執行下列指令:

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

   當您匯入相符的公開和私密金鑰時， Kubernetes 會提供這個有用的公用程式。 如果您有其他憑證要新增 (例如，新增至佇列管理程式信任儲存庫)，請執行下列指令:

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

   例如，如果要匯入的檔案是 `trust1.crt`、 `trust2.crt` 和 `trust3.crt`，則指令如下:

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

下一步

現在您已準備好 在 Red Hat OpenShift 上建立新的佇列管理程式。

OpenShift CP4I-LTS CD 在 Red Hat OpenShift 上建立新的佇列管理程式

在 Red Hat OpenShift 上部署單一實例或多重實例佇列管理程式。

開始之前

這項作業假設您已 建立佇列管理程式資源，且 已將 IBM MQ Operator 安裝至 Red Hat OpenShift。

關於這項作業

如 第 88 頁的『選取在儲存器中執行之 IBM MQ 的目標 HA 架構』中所述，有兩種可能的部署拓蹼。 因此，本主題提供兩個不同的範本:

- 部署單一實例佇列管理程式。
- 部署多重實例佇列管理程式。

重要: 根據您偏好的拓蹼，只完成兩個範本中的一個。

程序

- 部署單一實例佇列管理程式。

  移轉的佇列管理程式會使用 YAML 檔案部署至 Red Hat OpenShift 。 以下是根據先前主題中使用的名稱的範例:

  ```
  apiVersion: mq.ibm.com/v1beta1
  kind: QueueManager
  metadata:
    name: qm1
  spec:
    version: 9.3.5.1-r2
    license:
      accept: true
      license: L-VTPK-22YZPK
  ```

```
    use: "Production"
  pki:
    keys:
    - name: default
      secret:
        secretName: my-tls-migration
        items:
          - tls.key
          - tls.crt
  web:
    enabled: true
  queueManager:
    name: QM1
    mqsc:
      - configMap:
          name: my-mqsc-migrated
          items:
            - backup.mqsc
```

視您執行的步驟而定，可能需要自訂先前的 YAML。 為了協助您解決此問題，以下是此 YAML 的說明:

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

這會定義 Kubernetes 物件、類型及名稱。 唯一需要自訂作業的欄位是 name 欄位。

```
 spec:
  version: 9.3.5.1-r2
  license:
    accept: true
    license: L-VTPK-22YZPK
    use: "Production"
```

這對應於部署的版本和授權資訊。 如果您需要自訂此項，請使用 第 165 頁的『mq.ibm.com/v1beta1 的授權參考手冊』中提供的資訊。

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

若要將佇列管理程式配置為使用 TLS，它必須參照相關憑證及金鑰。 `secretName` 欄位會參照在 匯入 IBM MQ TLS 資源 區段內建立的 Kubernetes 密鑰，並且項目清單 (`tls.key` 和 `tls.crt`) 是 Kubernetes 使用 `oc create secret tls` 語法時指派的標準名稱。 如果您有其他憑證要新增至信任儲存庫，則可以用類似方式新增這些憑證，但這些項目是匯入期間使用的對應檔名。 例如，下列程式碼可用來建立信任儲存庫憑證:

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

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

重要: 如果不需要 TLS，請刪除 YAML 的 TLS 區段。

```
  web:
    enabled: true
```

這會啟用部署的 Web 主控台

```
  queueManager:
    name: QM1
```

這會將佇列管理程式的名稱定義為 QM1。 佇列管理程式會根據您的需求自訂，例如原始佇列管理程式名稱。

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

前一個程式碼會取回在 匯入 IBM MQ 配置 區段中匯入的佇列管理程式配置。 如果您使用不同的名稱，則需要修改 `my-mqsc-migrated` 和 `backup.mqsc`。

請注意，範例 YAML 假設 Red Hat OpenShift 環境的預設儲存類別定義為 RWX 或 RWO 儲存類別。 如果環境中未定義預設值，則您需要指定要使用的儲存類別。 您可以透過延伸 YAML 來執行此動作，如下所示:

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

新增強調顯示的文字，並自訂類別屬性以符合您的環境。 若要探索環境內的儲存類別名稱，請執行下列指令:

```
oc get storageclass
```

以下是此指令傳回的範例輸出:

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

下列程式碼顯示如何參照在 匯入 IBM MQ 配置 區段中匯入的 IBM MQ 配置。 如果您使用不同的名稱，則需要修改 `my-mqsc-migrated` 和 `backup.mqsc`。

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

您已部署單一實例佇列管理程式。 這會完成範本。 現在，您已準備好 驗證新的儲存器部署。

- 部署多重實例佇列管理程式。

  移轉的佇列管理程式會使用 YAML 檔案部署至 Red Hat OpenShift 。 下列範例基於前幾節中使用的名稱。

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: qm1mi
spec:
  version: 9.3.5.1-r2
  license:
    accept: true
    license: L-VTPK-22YZPK
    use: "Production"
  pki:
    keys:
    - name: default
```

```
      secret:
        secretName: my-tls-migration
        items:
          - tls.key
          - tls.crt
  web:
    enabled: true
  queueManager:
    name: QM1
    availability: MultiInstance
    storage:
      defaultClass: aws-efs
      persistedData:
        enabled: true
      queueManager:
        enabled: true
      recoveryLogs:
        enabled: true
    mqsc:
      - configMap:
          name: my-mqsc-migrated
          items:
            - backup.mqsc
```

以下是此 YAML 的說明。 大部分配置遵循與 部署單一實例佇列管理程式相同的方法，因此這裡只會說明佇列管理程式可用性及儲存體方面。

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

這會將佇列管理程式名稱指定為 QM1，並將部署設為 MultiInstance，而不是預設單一實例。

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

IBM MQ 多重實例佇列管理程式取決於 RWX 儲存體。 依預設，佇列管理程式會以單一實例模式部署，因此變更為多重實例模式時需要其他儲存體選項。 在先前的 YAML 範例中，會定義三個儲存體持續性磁區及一個持續保存磁區類別。 這個持續保存的磁區類別必須是 RWX 儲存類別。 如果您不確定環境內的儲存類別名稱，則可以執行下列指令來探索它們:

```
oc get storageclass
```

以下是此指令傳回的範例輸出:

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

下列程式碼顯示如何參照在 匯入 IBM MQ 配置 區段中匯入的 IBM MQ 配置。 如果您使用不同的名稱，則需要修改 my-mqsc-migrated 和 backup.mqsc。

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

您已部署多重實例佇列管理程式。 這會完成範本。 現在，您已準備好 驗證新的儲存器部署。

OpenShift CP4I-LTS CD 驗證新的儲存器部署

既然 IBM MQ 已部署在 Red Hat OpenShift 上，您可以使用 IBM MQ 範例來驗證環境。

開始之前

這項作業假設您已 在 Red Hat OpenShift 上建立新的佇列管理程式。

重要: 此作業假設未在佇列管理程式中啟用 TLS。

關於這項作業

在這項作業中，您可以從移轉的佇列管理程式儲存器內執行 IBM MQ 範例。 不過，您可能偏好使用您自己從另一個環境執行的應用程式。

您需要下列資訊:

- LDAP 使用者名稱
- LDAP 密碼
- IBM MQ 通道名稱
- 佇列名稱

此程式碼範例使用下列設定。 請注意，您的設定將會不同。

- LDAP 使用者名稱 :mqapp
- LDAP 密碼 :mqapp
- IBM MQ 通道名稱: DEV.APP.SVRCONN
- 佇列名稱: Q1

程序

1. 對執行中 IBM MQ 儲存器執行 exec 指令。

   使用下列指令:

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

   其中 `qm1-ibm-mq-0` 是我們在 第 90 頁的『在 Red Hat OpenShift 上建立新的佇列管理程式』中部署的 Pod。 如果您呼叫不同的部署，則自訂此值。
2. 傳送訊息。

   執行下列指令:

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

   系統會提示您輸入密碼，然後您可以傳送訊息。
3. 請驗證已順利收到訊息。

   執行 GET 範例:

   ```
   ./amqsgetc Q1 QM1
   ```

結果

您已完成 第 73 頁的『將 IBM MQ 移轉至 IBM Cloud Pak for Integration』。

下一步

請使用下列資訊來協助您處理更複雜的移轉實務範例:

移轉排入佇列的訊息
若要移轉現有的佇列訊息，請遵循下列主題中的指引，在新的佇列管理程式就緒之後匯出及匯入訊息: 在兩個系統之間使用 dmpmqmsg 公用程式。

從 Red Hat OpenShift 環境外部連接至 IBM MQ
已部署的佇列管理程式可以向 Red Hat OpenShift 環境外部的 IBM MQ 用戶端及佇列管理程式公開。 此處理程序視連接至 Red Hat OpenShift 環境的 IBM MQ 版本而定。 請參閱第 140 頁的『配置路徑以從 Red Hat OpenShift 叢集外部連接至佇列管理程式』。

OpenShift CP4I 安裝 IBM MQ Operator

IBM MQ Operator 可以使用 OpenShift 主控台或指令行介面 (CLI) 安裝至 Red Hat OpenShift 。

開始之前

為了確保安裝儘可能順利進行，在開始安裝之前，請確定您已瞭解所有必要條件及需求。 請參閱第 5 頁的『規劃儲存器中的 IBM MQ』。

重要: V 9.3.4 在安裝 IBM MQ Operator 之前，請先檢閱 建構部署 的指引。

關於這項作業

下列步驟代表安裝 IBM MQ Operator 的一般作業流程:

1. 安裝 Red Hat OpenShift Container Platform。
2. 配置儲存體。
3. 鏡映映像檔 (僅限氣隙)。
4. 新增 IBM 操作器型錄並準備叢集。
5. 安裝 IBM MQ Operator。
6. 建立授權金鑰密碼 (僅限線上安裝)。
7. 選用項目: 安裝 IBM Cloud Pak for Integration (CP4I) 及其相依關係。
8. 部署 License Service。
9. 部署佇列管理程式。

程序

1. 安裝 Red Hat OpenShift Container Platform。

   如需安裝 OpenShift 的詳細步驟，請參閱 安裝 Red Hat 軟體 4.6 或更新版本。

   重要: 請確保安裝受支援版本的 OpenShift Container Platform。 例如，若要使用 IBM MQ Operator 2.0 或更新版本，您必須安裝 OpenShift Container Platform 4.12 或更新版本。 另請注意，僅支援 OpenShift Container Platform Extended Update Support (EUS) 版本，這些版本是偶數次要版本，例如 4.14 和 4.16。 如需相關資訊，請參閱 IBM Cloud Pak 和 Red Hat OpenShift Container Platform 相容性。

   對於使用 Red Hat OpenShift Container Platform CLI 的任何步驟，您必須使用 `oc login` 登入 OpenShift 叢集。 若要安裝 CLI，請參閱 開始使用 OpenShift CLI。

   安裝 OpenShift 之後，您可以使用您在 建立授權金鑰密碼中建立的 IBM 授權金鑰，來驗證並取得儲存器軟體的存取權。
2. 配置儲存。

   您必須在 Red Hat OpenShift Container Platform 中定義儲存空間類別，並設定儲存空間配置以滿足調整大小需求。

   重要: IBM MQ 單一實例及原生 HA 佇列管理程式可以使用 RWO 存取模式，而多重實例佇列管理程式則需要 RWX，如 第 19 頁的『IBM MQ Operator 的儲存體考量』中所述。 IBM MQ 多重實例佇列管理程式需要特定的檔案系統性質，您可以使用 測試 IBM MQ 的共用檔案系統的指示來驗證這些檔案系統性質。

在 IBM MQ 檔案系統的測試陳述式中，可以找到已知相容及不相容檔案系統的清單，以及其他限制的注意事項。

建議的儲存體提供者位於 CP4I 儲存體考量 頁面上。

3. V 9.3.4
鏡映映像檔 (僅限氣隙)。

如果叢集位於受限 (氣隙) 網路環境中，則必須鏡映 IBM MQ 映像檔。 視您的配置而定，您可能還需要鏡映一些其他元件。 請閱讀下列資訊，然後視需要鏡映映像檔。

- 您必須鏡映 IBM MQ 映像檔。 請使用下列值:

```
export OPERATOR_PACKAGE_NAME=ibm-mq
export OPERATOR_VERSION=3.1.3
```

- 如果您想要部署至少一個 **所有** 下列陳述式均為 true 的佇列管理程式，則還必須鏡映部分其他必要元件:
  - 您使用 IBM MQ 9.3.4 或更新版本。
  - 您正在使用 CP4I 授權。
  - 已啟用 IBM MQ Console 。
  - 您使用 IBM Cloud Pak for Integration Keycloak 服務進行 IBM MQ Console 單一登入 (SSO) 鑑別及授權 (預設值)。

  如果先前的陳述式為 true ，則 Keycloak 會提供 SSO ，且您必須鏡映下列每一個元件:
  - IBM Cloud Pak foundational services
  - Certificate Manager. 如果您已安裝早於 4.4 版的 IBM Cloud Pak foundational services 操作器版本，則必須鏡映 Certificate Manager。[6]
  - IBM Cloud Pak for Integration
  - Keycloak (Red Hat OpenShift 運算子)

若要建立鏡映映像檔，請參閱 氣隙叢集的鏡映映像檔。

4. 新增 IBM MQ Operator 型錄來源。

新增型錄來源，讓操作器可供叢集使用。 請參閱 第 97 頁的『新增 IBM MQ Operator 型錄來源』。

5. 安裝「IBM MQ Operator」。

選擇下列兩個選項之一 (使用主控台或使用 CLI):

- 選項 1: 使用 OpenShift 主控台安裝 IBM MQ Operator。
- 選項 2: 使用 OpenShift CLI 安裝 IBM MQ Operator。

6. 建立授權金鑰密碼 (僅限線上安裝)。

IBM MQ Operator 會部署從執行授權檢查的容器登錄取回的佇列管理程式映像檔。 此檢查需要儲存在 `docker-registry` 取回密碼中的授權金鑰。 如果您在將安裝佇列管理程式的名稱空間中還沒有授權金鑰，請遵循下列指示來取得授權金鑰並建立取回密鑰。

註: 如果只部署 IBM MQ Advanced for Developers (非 Warranted) 佇列管理程式，則不需要授權金鑰。

您可以使用 OpenShift 主控台或 CLI 來建立授權金鑰密碼。 下列範例使用 CLI:

a. 取得指派給 IBM ID 的授權金鑰。 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。

c. 從 OpenShift CLI ，執行下列指令，以建立稱為 `ibm-entitlement-key` 的映像檔取回密碼。

```
oc create secret docker-registry ibm-entitlement-key \
--docker-server=cp.icr.io \
--docker-username=cp \
```

[6] 從 IBM Cloud Pak foundational services 的 4.4 版開始，不再需要此鏡映。

```
--docker-password=<entitlement-key> \
--docker-email=<user-email>
\--namespace=<namespace>
```

其中 *<entitlement-key>* 是您在步驟 b 中複製的授權金鑰， *<user-email>* 是與授權軟體相關聯的 IBM ID， *<namespace>* 是您在其中安裝 IBM MQ Operator 的名稱空間。

7. 選擇性的: 安裝 CP4I 及其相依關係。

   當您部署至少一個佇列管理程式時，其中 **所有** 下列陳述式均為 true 時，會有一些額外的必要元件:

   - 您使用 IBM MQ 9.3.4 或更新版本。
   - 您正在使用 CP4I 授權。
   - 已啟用 IBM MQ Console 。
   - 您使用 CP4I Keycloak 服務進行 IBM MQ Console 單一登入 (SSO) 鑑別及授權 (預設值)。

   如果所有先前陳述式都為 true ，則 Keycloak 會提供 SSO ，且您必須完成下列其他步驟:

   - 以與 CP4I 操作器相同的安裝模式安裝 IBM Cloud Pak foundational services 操作器。 如需支援的版本，請參閱 此版本的操作器通道版本
   - 如果您已安裝 4.4 版之前的 IBM Cloud Pak foundational services 操作器版本，請 安裝 Red Hat OpenShift Container Platform 的 cert-manager 操作器。[7]
   - 安裝 CP4I 操作器。
   - 選用項目: 部署平台使用者介面。

     a. 建立 `ibm-common-services` 名稱空間。 透過 CLI 登入 OpenShift 叢集時，請執行下列指令:

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

     b. 部署平台使用者介面。

8. 部署 License Service。

   這是監視佇列管理程式授權使用情況的必要項目。 遵循 部署 License Service 中的指示。

9. 部署佇列管理程式。

   如需部署範例 "快速入門" 佇列管理程式的指示，請參閱 第 106 頁的『將佇列管理程式部署至 Red Hat OpenShift Container Platform 叢集』。

相關工作

第 109 頁的『解除安裝 IBM MQ Operator』
您可以使用 Red Hat OpenShift 主控台或 CLI ，從 Red Hat OpenShift 解除安裝 IBM MQ Operator 。

相關參考

第 101 頁的『在氣隙環境中安裝 IBM MQ Operator 2.x』
本指導教學引導您將 IBM MQ Operator 2.x 安裝至沒有網際網路連線功能的 Red Hat OpenShift 叢集。 您可以使用可攜式儲存裝置或使用防禦機器，在氣隙環境中安裝 IBM MQ Operator 。

OpenShift CP4I 新增 IBM MQ Operator 型錄來源

將型錄來源新增至 OpenShift 叢集，會將 IBM 操作器新增至您可以安裝的操作器清單。

開始之前

這項作業假設您已完成 第 95 頁的『安裝 IBM MQ Operator』的前 3 個步驟。

此作業必須由叢集管理者執行。

[7] 從 IBM Cloud Pak foundational services 4.4 版開始，不再需要這樣做。

關於這項作業

IBM MQ Operator 型錄是運算子的索引，可用來延伸 Red Hat OpenShift Container Platform 叢集的 API，以啟用 IBM 軟體產品。

完成 **選項 A: 氣隙** 或 **選項 B: 網際網路**，視您的叢集是在受限 (氣隙) 網路環境中，還是您的叢集可以存取網際網路而定。

程序

- V 9.3.4

  選項 A: 氣隙 在氣隙網路環境中新增型錄來源。

  a) 新增 IBM MQ Operator 型錄來源。

  遵循 將型錄來源新增至叢集中的指示。

  註: 因為您已完成操作員安裝步驟 鏡映映像檔 (僅限氣隙)，所以您只需要完成套用型錄來源的步驟。例如：

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

  b) 新增其他必要元件的型錄來源。

  當您部署至少一個佇列管理程式時，其中 **所有** 下列陳述式均為 true 時，會有一些額外的必要元件：

  - 您使用 IBM MQ 9.3.4 或更新版本。
  - 您正在使用 IBM Cloud Pak for Integration 授權。
  - 已啟用 IBM MQ Console 。
  - 您使用 IBM Cloud Pak for Integration Keycloak 服務進行 IBM MQ Console 單一登入 (SSO) 鑑別及授權 (預設值)。

  如果所有先前陳述式都為 true，則由 Keycloak 提供 SSO。 因此，對於 IBM MQ Operator 型錄來源，您也必須針對下列每一個其他必要元件，遵循 將型錄來源新增至叢集 中的步驟：

  - IBM Cloud Pak foundational services
  - Certificate Manager. 如果您已安裝早於 4.4 版的 IBM Cloud Pak foundational services 操作器版本，則必須鏡映 Certificate Manager。[8]
  - IBM Cloud Pak for Integration

- **選項 B: 網際網路** 在可存取網際網路的環境中新增型錄來源。

  使用 OpenShift CLI 來建立 `CatalogSource` 。

  將下列 YAML 檔套用至 Red Hat OpenShift Container Platform 叢集，以新增型錄。

  a) 建立 CatalogSource YAML。

  將下列資源定義儲存為稱為 `catalog_source.yaml` 的檔案。

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

  b) 套用 `CatalogSource` YAML。

[8] 從 IBM Cloud Pak foundational services 的 4.4 版開始，不再需要此鏡映。

透過按一下 "+" 按鈕或使用指令行，從 Red Hat OpenShift Container Platform Web 主控台執行此動作。

例如，執行下列指令來套用檔案:

```
oc apply -f catalog_source.yaml -n openshift-marketplace
```

c) 驗證已順利建立 CatalogSource

請執行下列指令:

```
oc get CatalogSources ibm-operator-catalog -n openshift-marketplace
```

您會在成功時收到此輸出:

```
NAME                    DISPLAY                TYPE   PUBLISHER    AGE
ibm-operator-catalog    IBM operator Catalog   grpc   IBM          28s
```

結果

現在，您已準備好完成 安裝 IBM MQ Operator 的步驟 5。

OpenShift CP4I 使用 OpenShift 主控台安裝 IBM MQ Operator

可以使用 OperatorHub 將 IBM MQ Operator 安裝至 Red Hat OpenShift 。

開始之前

這項作業假設您已完成 第 95 頁的『安裝 IBM MQ Operator』的步驟 1-4。

程序

1. 登入 Red Hat OpenShift 叢集主控台。
2. 從導覽窗格中，按一下 **操作器** > **OperatorHub**。

   即會顯示 OperatorHub 頁面。
3. 在 **所有項目** 欄位中，輸入 "IBM MQ"。

   這時會顯示 IBM MQ 型錄項目。
4. 選取 **IBM MQ**。

   即會顯示 IBM MQ 視窗。
5. 按一下**安裝**。

   即會顯示「安裝操作器」頁面。
6. 輸入下列值:

   a) 將 **通道** 設為您選擇的版本。

   檢閱 第 10 頁的『IBM MQ Operator 的版本支援』，以判定要選擇哪個操作員通道。

   b) 將 **安裝模式** 設為「叢集上的特定名稱空間」(您可以在下一步中建立)，或整個叢集範圍。

   建議選擇叢集層面範圍，因為在不同名稱空間中安裝不同版本的操作員可能會導致問題。 操作員設計為控制平面的延伸。

   c) 選擇性的: 如果您選擇「叢集上的特定名稱空間」，請將 **名稱空間** 設為您要在其中安裝操作器的專案 (名稱空間) 值。

   註: 使用主控台來安裝操作器時，您可以使用現有名稱空間 (操作器提供的預設名稱空間)，或建立新的名稱空間。 如果您要建立新的名稱空間，您可以從這個表單建立它，如下所示: 從導覽窗格中，按一下 **首頁** > **專案**，選取 **建立專案**，指定您要建立之專案 (名稱空間) 的 **名稱**，然後按一下 **建立**。

   d) 將 **核准策略** 設為「自動」。
7. 按一下 **安裝**，並等待您的操作器安裝。

   當安裝完成時，會為您提供確認。

如果要驗證安裝，請導覽至 **操作器 > 已安裝的操作器**，然後從 **專案** 下拉清單中選取您的專案。 當安裝完成時，操作器的狀態會變更為「成功」。

下一步

現在，您已準備好 建立授權金鑰密碼 (第 95 頁的『安裝 IBM MQ Operator』的步驟 6)。

OpenShift CP4I 使用 Red Hat OpenShift CLI 安裝 IBM MQ Operator

IBM MQ Operator 可以使用指令行介面 (CLI) 安裝至 Red Hat OpenShift 。

開始之前

這項作業假設您已完成 第 95 頁的『安裝 IBM MQ Operator』的步驟 1-4。

程序

1. 使用 **oc login** 登入 Red Hat OpenShift 指令行介面 (CLI)。
2. 選擇性的: 建立要用於 IBM MQ Operator 的名稱空間。

   IBM MQ Operator 可以安裝範圍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。

   若要在 CLI 中建立新的名稱空間，請執行下列指令:

   ```
   oc create namespace <namespace_name>
   ```

   其中 *<namespace_name>* 是您要建立之名稱空間的名稱。
3. 從 OperatorHub 檢視可供叢集使用的運算子清單:

   ```
   oc get packagemanifests -n openshift-marketplace
   ```
4. 檢查 IBM MQ Operator ， 以驗證其受支援的 **InstallModes** 及可用的 **Channels**。

   ```
   oc describe packagemanifests ibm-mq -n openshift-marketplace
   ```
5. 選擇性的: 建立 **OperatorGroup**。

   OperatorGroup 是一種 OLM 資源，它會選取目標名稱空間，以針對與 **OperatorGroup** 相同的名稱空間中的所有操作員產生必要的 RBAC 存取權。

   您訂閱操作器的名稱空間必須具有符合操作器 **InstallMode** 的 **OperatorGroup** ， 可以是 `AllNamespaces` 或 `SingleNamespace` 模式。

   如果您要安裝的操作器使用 `AllNamespaces` 模式，則 `openshift-operators` 名稱空間已備妥適當的 **OperatorGroup** ， 您可以跳過此步驟。

   如果操作器使用 `SingleNamespace` 模式，且您還沒有適當的 **OperatorGroup** ， 請執行下列指令來建立一個:

   ```
   cat << EOF | oc apply -f -
   apiVersion: operators.coreos.com/v1
   kind: OperatorGroup
   metadata:
     name: <operatorgroup_name>
     namespace: <namespace_name>
   spec:
     targetNamespaces:
     - <namespace_name>
   EOF
   ```
6. 檢閱 第 10 頁的『IBM MQ Operator 的版本支援』， 以判定要選擇哪個操作員通道。
7. 安裝操作器。

使用下列指令，變更 *<ibm-mq-operator-channel>* 以符合您要安裝之 IBM MQ 操作器版本的通道，並將 *<namespace_name>* 變更為 **openshift-operators** (如果您使用 "AllNamespaces" 模式)，或變更為您要將 IBM MQ 操作器部署至其中的名稱空間 (如果您使用 "SingleNamespace" 模式)。

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

8. 幾分鐘之後，即會安裝操作器。 執行下列指令，以驗證所有元件都處於「成功」狀態:

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

其中 *<namespace_name>* 是 **openshift-operators** (如果您使用 "AllNamespaces" 模式)，或專案 (名稱空間) 名稱 (如果您使用 "SingleNamespace" 模式)。

下一步

現在，您已準備好 建立授權金鑰密碼 (第 95 頁的『安裝 IBM MQ Operator』的步驟 6)。

OpenShift CP4I Linux 在氣隙環境中安裝 IBM MQ Operator 2.x

本指導教學引導您將 IBM MQ Operator 2.x 安裝至沒有網際網路連線功能的 Red Hat OpenShift 叢集。 您可以使用可攜式儲存裝置或使用防禦機器，在氣隙環境中安裝 IBM MQ Operator 。

開始之前

這些指示適用於在氣隙環境中安裝 IBM MQ Operator 2.x 版本。 若要安裝 IBM MQ Operator 3.0.0 以及更新版本，請參閱 第 95 頁的『安裝 IBM MQ Operator』，並特別注意氣隙特定步驟。

使用可攜式儲存裝置在氣隙環境中安裝 IBM MQ Operator

如需完成安裝的步驟，請參閱 IBM Cloud Pak for Integration 說明文件中的 使用可攜式儲存裝置鏡映映像檔 。 如果您只安裝 IBM MQ，請將下列環境變數的所有出現項目取代為這裡提供的值:

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

其中 *version_number* 是您要用來執行氣隙安裝的案例版本。 如需可用案例版本的清單，請參閱 https://github.com/IBM/cloud-pak/tree/master/repo/case/ibm-mq。 檢閱 第 10 頁的『IBM MQ Operator 的版本支援』，以判定要選擇哪個操作員通道。

使用防禦機器在氣隙環境中安裝 IBM MQ Operator

1. 第 102 頁的『必要條件』
2. 第 102 頁的『準備 Docker 登錄』
3. 第 102 頁的『準備防禦主機』
4. 第 103 頁的『建立安裝程式和映像檔庫存的環境變數』
5. 第 103 頁的『下載 IBM MQ 安裝程式及映像檔庫存』
6. 第 103 頁的『以叢集管理者身分登入 OpenShift Container Platform 叢集』
7. 第 103 頁的『建立 IBM MQ Operator 的 Kubernetes 名稱空間』
8. 第 104 頁的『鏡映映像檔並配置叢集』

9. 第 105 頁的『安裝「IBM MQ Operator」。』
10. 第 106 頁的『部署 IBM MQ 佇列管理程式』

必要條件

1. 必須安裝 OpenShift Container Platform 叢集。 如需支援的 OpenShift Container Platform 版本，請參閱第 10 頁的『IBM MQ Operator 的版本支援』。
2. Docker 登錄必須可用。 如需相關資訊，請參閱 第 102 頁的『準備 Docker 登錄』。
3. 必須配置防禦伺服器。 如需相關資訊，請參閱 第 102 頁的『準備防禦主機』。

準備 Docker 登錄

本端 Docker 登錄用來儲存本端環境中的所有映像檔。 您必須建立此類登錄，且必須確保它符合下列需求:

- 支援 Docker 資訊清單 V2 綱目 2。
- 支援多架構映像檔。
- 可從防禦伺服器及 OpenShift Container Platform 叢集節點存取。
- 具有可以從防禦主機寫入目標登錄的使用者名稱及密碼。
- 具有可以從 Red Hat OpenShift 叢集節點上的目標登錄讀取之使用者的使用者名稱及密碼。
- 容許在映像檔名稱中使用路徑分隔字元。

建立 Docker 登錄之後，您必須配置登錄:

- Red Hat OpenShift 說明文件中的 建立鏡映登錄以在受限網路中安裝 包含簡式登錄的範例。
- 請確認每一個名稱空間都符合下列需求:
  - 支援自動建立儲存庫。
  - 具有可以寫入及建立儲存庫之使用者的認證。 防禦主機會使用這些認證。
  - 具有可以讀取所有儲存庫之使用者的認證。 OpenShift Container Platform 叢集使用這些認證。

準備防禦主機

準備可以存取 OpenShift Container Platform 叢集、本端 Docker 登錄及網際網路的防禦主機。 防禦主機必須位於具有 IBM Cloud Pak CLI 及 OpenShift Container Platform CLI 支援之任何作業系統的 Linux for x86-64 平台上。

在防禦主機節點上完成下列步驟:

1. 安裝 OpenSSL 1.11.1 版或更新版本。
2. 在防禦主機節點上安裝 Docker 或 Podman 。
   - 若要安裝 Docker，請執行下列指令:

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

   - 若要安裝 Podman，請參閱 Podman 安裝指示
3. 在防禦主機節點上安裝 skopeo 1.x.x 版。 若要安裝 skopeo ，請執行下列指令:

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

4. 安裝 IBM Cloud Pak CLI。 為您的平台安裝最新版本的二進位檔。 如需相關資訊，請參閱 cloud-pak-cli。
   a. 下載該二進位檔。

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

例如：

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

b. 解壓縮該二進位檔。

```
tar -xf binary-file-name
```

c. 執行下列指令來修改及移動檔案

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

d. 確認已安裝 `cloudctl`：

```
cloudctl --help
```

5. 安裝 oc OpenShift Container Platform CLI 工具。

   如需相關資訊，請參閱 OpenShift Container Platform CLI 工具

6. 建立一個目錄，用來作為離線儲存庫。

   下列是範例目錄。 此範例在後續步驟中使用。

```
mkdir $HOME/offline
```

   附註: 此離線儲存庫必須持續保存，以避免多次傳送資料。 持續性也有助於多次執行或依排程執行鏡映程序。

建立安裝程式和映像檔庫存的環境變數

使用安裝程式映像檔名稱及映像檔庫存來建立下列環境變數:

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

其中 *version_number* 是您要用來執行氣隙安裝的案例版本。 如需可用案例版本的清單，請參閱 https://github.com/IBM/cloud-pak/tree/master/repo/case/ibm-mq。 檢閱 IBM MQ Operator 的版本支援，以判定要選擇的操作員通道。

下載 IBM MQ 安裝程式及映像檔庫存

將 `ibm-mq` 安裝程式及映像檔庫存下載至防禦主機:

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

以叢集管理者身分登入 OpenShift Container Platform 叢集

下列是登入 OpenShift Container Platform 叢集的範例指令:

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

建立 IBM MQ Operator 的 Kubernetes 名稱空間

建立具有名稱空間的環境變數以安裝 IBM MQ Operator，然後建立名稱空間:

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

鏡映映像檔並配置叢集

完成下列步驟，以鏡映映像檔並配置叢集:

註: 請勿在任何指令中的雙引號內使用波狀符號。 例如，請勿使用 `args "--registry registry --user registry_userid --pass registry_password --inputDir ~/offline"`。 波狀符號不會展開，您的指令可能會失敗。

1. 儲存所有來源 Docker 登錄的鑑別認證。

   所有 IBM Cloud Platform Common Services、 IBM MQ Operator 映像檔和 IBM MQ Advanced Developer 映像檔都儲存在不需要鑑別的公用登錄中。 不過， IBM MQ Advanced Server (非開發人員)、其他產品和協力廠商元件需要一或多個鑑別登錄。 下列登錄需要鑑別:

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

   如需這些登錄的相關資訊，請參閱 建立登錄名稱空間。

   您必須執行下列指令，為所有需要鑑別的登錄配置認證。 針對每一個此類登錄分別執行指令:

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

   此指令會將登錄認證儲存並快取在檔案系統上位於 `$HOME/.airgap/secrets` 位置的檔案中。

2. 使用本端 Docker 登錄連線資訊來建立環境變數。

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

   附註: Docker 登錄使用標準埠，例如 80 或 443。 如果 Docker 登錄使用非標準埠，請使用語法 *host*:*port* 來指定埠。 例如:

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

3. 配置本端 Docker 登錄的鑑別密碼。

   附註: 此步驟只需要執行一次。

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

   此指令會將登錄認證儲存並快取在檔案系統上位於 `$HOME/.airgap/secrets` 位置的檔案中。

4. 配置廣域映像檔取回密鑰和 **ImageContentSourcePolicy**。

   a. 檢查是否需要重新啟動節點。

   - 在 OpenShift Container Platform 4.4 版及更新版本中，以及在使用氣隙的 IBM MQ Operator 新安裝上，此步驟會重新啟動所有叢集節點。 可能要到套用新的取回密鑰之後，叢集資源才可供使用。
   - 在 IBM MQ Operator 1.8 中， CASE 已更新為包含映像檔的其他鏡映來源。 因此，當您從舊版 IBM MQ Operator 升級至 1.8 版或更高版本時，會觸發節點重新啟動。
   - 若要檢查此步驟是否需要重新啟動節點，請將 `--dry-run` 選項新增至此步驟的程式碼。 這會產生最新的 **ImageContentSourcePolicy** ，並將它顯示在主控台視窗 (**stdout**) 中。 如果此

ImageContentSourcePolicy 不同於配置的叢集 **ImageContentSourcePolicy**，則會進行重新啟動。

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

b. 若要配置廣域映像檔取回密碼及 **ImageContentSourcePolicy**，請在沒有 `--dry-run` 選項的情況下執行此步驟的程式碼:

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

5. 確認 **ImageContentSourcePolicy** 資源已建立。

```
oc get imageContentSourcePolicy
```

6. 選用項目: 如果您使用不安全的登錄，則必須將本端登錄新增至叢集 **insecureRegistries** 清單。

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

7. 驗證叢集節點狀態。

```
oc get nodes
```

在套用 **imageContentsourcePolicy** 及廣域映像檔取回密鑰之後，您可能會看到節點狀態為 **Ready**、**Scheduling** 或 **Disabled**。 請等待所有節點都顯示 **Ready** 狀態。

8. 將映像檔鏡映至本端登錄。

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

安裝「IBM MQ Operator」。

1. 登入 Red Hat OpenShift 叢集 Web 主控台。
2. 建立型錄來源。 請使用執行先前步驟的相同終端機。

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

3. 驗證已為 Common Services 安裝程式操作員建立 **CatalogSource** 。

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

4. 使用 OLM 來安裝 IBM MQ Operator 。

a. 從導覽窗格中，按一下 **操作器** > **OperatorHub**。

即會顯示 **OperatorHub** 頁面。

b. 在 **所有項目** 欄位中，輸入 IBM MQ。

即會顯示 IBM MQ 型錄項目。

c. 選取 **IBM MQ**。

即會顯示 **IBM MQ** 視窗。

d. 按一下**安裝**。

即會顯示「**建立操作員訂閱**」頁面。

e. 檢閱 第 10 頁的『IBM MQ Operator 的版本支援』，以判定要選擇哪個操作員通道。

f. 將 **安裝模式** 設為您建立的特定名稱空間或叢集範圍。

g. 按一下**訂閱**。

IBM MQ 會新增至「**已安裝的操作器**」頁面。

h. 在「**已安裝的操作器**」頁面上檢查操作器的狀態。 安裝完成時，狀態會變更為 **Succeeded**。

部署 IBM MQ 佇列管理程式

若要在已安裝的操作器下建立新的佇列管理程式，請參閱 第 106 頁的『將佇列管理程式部署至 Red Hat OpenShift Container Platform 叢集』。

相關工作

第 115 頁的『[已淘汰]在氣隙環境中準備升級至最新 IBM MQ 2.x 操作員或佇列管理程式』
在沒有網際網路連線功能的 Red Hat OpenShift 叢集中，在升級「IBM MQ 2.x 操作員」或佇列管理程式之前，您需要採取一些準備步驟。

OpenShift CP4I 將佇列管理程式部署至 Red Hat OpenShift Container Platform 叢集

此範例會部署「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 IBM MQ 安全。 在重新啟動佇列管理程式之後，訊息不會持續保存。 您可以調整配置來變更許多佇列管理程式設定。

關於這項作業

此作業提供 3 個選項，用於將佇列管理程式部署至 OpenShift:

1. 使用 OpenShift 主控台部署佇列管理程式。
2. 使用 OpenShift CLI 部署佇列管理程式。
3. 使用 IBM Cloud Pak for Integration Platform UI 部署佇列管理程式。

程序

- **選項 1: 使用 OpenShift 主控台部署佇列管理程式。**

a) 部署佇列管理程式。

a. 使用 Red Hat OpenShift Container Platform 叢集管理者認證登入 OpenShift 主控台。

b. 將 **專案** 變更為 IBM MQ Operator 安裝所在的名稱空間。 從 **專案** 下拉清單中選取名稱空間。

c. 在導覽窗格中，按一下 **操作器** > **已安裝的操作器**。

d. 在「已安裝的操作器」畫面的清單中，尋找並按一下 **IBM MQ**。

e. 按一下 **佇列管理程式** 標籤。

f. 按一下 **建立 QueueManager** 按鈕。 即會顯示實例建立畫面，並提供兩種方法來配置資源: **表單視圖** 及 **YAML 視圖**。 依預設會選取 **表單視圖**。

b) 配置佇列管理程式。

步驟 2 選項 1: 在 **表單視圖**中配置。

表單視圖 會開啟一個表單，您可以用來檢視或修改資源配置。

a. 按一下 **授權**旁邊的箭頭，以展開授權接受區段。

b. 如果您接受授權合約，請將 **授權接受** 設為 **true** 。

c. 按一下箭頭以開啟下拉清單，然後選取授權。 IBM MQ 可在數個不同的授權下使用。 如需有效授權的相關資訊，請參閱 第 165 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。

d. 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。

步驟 2 選項 2: 在 **YAML 視圖**中配置。

YAML 視圖 會開啟編輯器，其中包含 QueueManager 的範例 YAML 檔案。 遵循下列步驟來更新檔案中的值。

a. 將 metadata.namespace 變更為您的專案 (名稱空間) 名稱。

b. 將 spec.license.license 的值變更為符合您需求的授權字串。 如需授權詳細資料，請參閱 第 165 頁的『mq.ibm.com/v1beta1 的授權參考手冊』。

c. 如果您接受授權合約，請將 spec.license.accept 變更為 true 。

d. 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。

c) 驗證佇列管理程式建立。

您可以完成下列步驟來驗證是否已建立「佇列管理程式」：

a. 確保您位於您在其中建立 IBM MQ Operator 的名稱空間中。

b. 從「 **首頁** 」畫面中，按一下 **操作器** > **已安裝的操作器**，然後選取您為其建立佇列管理程式的已安裝 IBM MQ Operator 。

c. 按一下 **佇列管理程式** 標籤。 當 QueueManager 狀態為 Running 時，建立完成。

- **選項 2: 使用 OpenShift CLI 部署佇列管理程式**。

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: 9.3.5.1-r2
  license:
    accept: false
    license: L-VTPK-22YZPK
    use: NonProduction
  web:
    enabled: true
  queueManager:
    name: "QUICKSTART"
    storage:
      queueManager:
        type: ephemeral
```

重要: 如果您接受授權合約，請將 accept: false 變更為 accept: true。 如需授權的詳細資料，請參閱 第 165 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，並在 IBM Cloud Pak for Integration 內啟用了單一登入的 Web 主控台。 V 9.3.4 從 IBM Cloud Pak for Integration 2023.4.1 開始，若要讓「單一登入」運作，您必須先 安裝其他 IBM Cloud Pak for Integration 元件。

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: 9.3.5.1-r2
  license:
    accept: false
    license: L-AMRD-XH6P3Q
  web:
    enabled: true
  queueManager:
    name: "QUICKSTART"
    storage:
      queueManager:
        type: ephemeral
```

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 165 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
```

c) 驗證佇列管理程式建立。

完成下列步驟，以驗證您已建立佇列管理程式:

a. 驗證部署:

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

b. 檢查狀態:

```
oc describe queuemanager quickstart
```

- **選項 3: 使用 IBM Cloud Pak for Integration Platform UI 部署佇列管理程式**。

a) 在瀏覽器中，啟動 IBM Cloud Pak for Integration Platform UI。

b) 在 IBM Cloud Pak for Integration Platform UI 中，按一下 **建立實例**。

c) 選取 **傳訊**，然後按 **下一步**。

即會顯示實例建立畫面，並提供兩種方法來配置資源: **表單視圖** 及 **YAML 視圖**。 依預設會選取 **表單視圖** 。

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**。

如需授權的詳細資料，請參閱 第 165 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) 在 **佇列管理程式** 區段中，檢查或更新基礎佇列管理程式的 **名稱** 。 在舊版 IBM Cloud Pak for Integration Platform UI 中，請使用 **佇列管理程式配置** 區段。

依預設， IBM MQ 用戶端應用程式使用的佇列管理程式名稱與 `QueueManager` 的名稱相同，但移除了任何無效字元 (例如連字號)。

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

h) 驗證佇列管理程式建立。

當 `QueueManager` 狀態為 `Running` 時，建立完成。

相關工作

第 140 頁的『配置路徑以從 Red Hat OpenShift 叢集外部連接至佇列管理程式』
您需要 Red Hat 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 ，因為只有在使用 TLS 1.2 或更高版本的通訊協定時，才能在 TLS 通訊協定中使用 SNI。 Red Hat OpenShift Container Platform Router 使用 SNI 將要求遞送至 IBM MQ 佇列管理程式。

第 155 頁的『連接至 Red Hat OpenShift 叢集中部署的 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 。

第 122 頁的『配置佇列管理程式的範例』
可以透過調整 QueueManager 自訂資源的內容來配置佇列管理程式。

OpenShift CP4I 解除安裝 IBM MQ Operator

您可以使用 Red Hat OpenShift 主控台或 CLI ， 從 Red Hat OpenShift 解除安裝 IBM MQ Operator 。

程序

- 選項 1: 使用 OpenShift 主控台解除安裝 IBM MQ Operator 。

  註: 如果 IBM MQ Operator 安裝在叢集上的所有專案/名稱空間中，請針對您要刪除佇列管理程式的每一個專案，重複下列程序的步驟 2-6。

  a) 使用 Red Hat OpenShift Container Platform 叢集管理者認證登入 Red Hat OpenShift Container Platform Web 主控台。

  b) 將 **專案** 變更為您要從中解除安裝 IBM MQ Operator 的名稱空間。 從 **專案** 下拉清單中選取名稱空間。

  c) 在導覽窗格中，按一下 **操作器** > **已安裝的操作器**。

  d) 按一下 **IBM MQ** 運算子。

  e) 按一下 **佇列管理程式** 標籤，以檢視此「 IBM MQ Operator」所管理的佇列管理程式。

  f) 刪除一或多個佇列管理程式。

  請注意，雖然這些佇列管理程式會繼續執行，但如果沒有 IBM MQ Operator，它們可能無法如預期般運作。

  g) 選擇性的: 適當的話，針對您要刪除佇列管理程式的每一個專案，重複步驟 2-6。

  h) 回到 **操作器** > **已安裝的操作器**。

  i) 在 **IBM MQ** 操作器旁邊，按一下三個點功能表，然後選取 **解除安裝操作器**。

- 選項 2: 使用 OpenShift CLI 解除安裝 IBM MQ Operator

  a) 使用 `oc login` 登入 Red Hat OpenShift 叢集。

  b) 如果 IBM MQ Operator 安裝在單一名稱空間中，請完成下列子步驟:

  a. 請確定您位於包含要解除安裝之 IBM MQ Operator 的專案中:

  ```
  oc project <project_name>
  ```

  b. 檢視專案中安裝的佇列管理程式:

  ```
  oc get qmgr
  ```

  c. 刪除一或多個佇列管理程式:

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

  請注意，雖然這些佇列管理程式會繼續執行，但如果沒有 IBM MQ Operator，它們可能無法如預期般運作。

  d. 檢視 **ClusterServiceVersion** 實例:

  ```
  oc get csv
  ```

  e. 刪除 IBM MQ **ClusterServiceVersion**:

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

  f. 檢視訂閱:

  ```
  oc get subscription
  ```

g. 刪除所有訂閱:

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

h. 如果沒有其他項目使用共用服務，則您可能想要解除安裝共用服務操作器，並刪除操作器群組:

i) 遵循 IBM Cloud Pak foundational services 產品說明文件中 解除安裝基礎服務 的指示，來解除安裝共用服務操作器。

ii) 檢視操作員群組:

```
oc get operatorgroup
```

iii) 刪除操作員群組:

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

c) 如果 IBM MQ Operator 已安裝且可用於叢集上的所有名稱空間，請完成下列子步驟:

a. 檢視所有已安裝的佇列管理程式:

```
oc get qmgr -A
```

b. 刪除一或多個佇列管理程式:

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

請注意，雖然這些佇列管理程式會繼續執行，但如果沒有 IBM MQ Operator，它們可能無法如預期般運作。

c. 檢視 **ClusterServiceVersion** 實例:

```
oc get csv -A
```

d. 從叢集中刪除 IBM MQ **ClusterServiceVersion** :

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

e. 檢視訂閱:

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

f. 刪除訂閱:

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

g. 選用項目: 如果沒有其他項目使用共用服務，您可能想要解除安裝共用服務操作器。 若要這樣做，請遵循 IBM Cloud Pak foundational services 產品說明文件中 解除安裝基礎服務 的指示。

OpenShift CP4I Operator2.0.0 升級 IBM MQ Operator 及佇列管理程式

IBM MQ Operator 的 Continuous Delivery (CD) 和 Long Term Support (LTS) 版本有不同的升級程序。 完成部署類型的升級步驟。

關於這項作業

若要升級 IBM MQ Operator 及佇列管理程式，請完成下列其中一個步驟:

程序

- 選項 1: **在現行操作器通道上將部署升級至最新版本**。

  若要將 IBM MQ Operator 的部署升級至現行操作器通道上的最新版本，請參閱 第 111 頁的『升級至 IBM MQ Operator 通道最新安全版本』。

- 選項 2: **升級 CD 部署**。

若要將先前 CD 部署的 IBM MQ Operator 升級至最新 CD 版本的 IBM MQ Operator (3.1.3 版) ，請參閱 第 112 頁的『移轉至 IBM MQ Operator 的現行 CD 通道』。

註:

2.0.x 版同時發行為 CD 和 LTS 版本，因此您可以使用 第 112 頁的『移轉至 IBM MQ Operator 的現行 CD 通道』 中的程序，從任何 2.0.x IBM MQ Operator 版升級至最新 CD 版本的 IBM MQ Operator。

OpenShift CP4I 升級至 IBM MQ Operator 通道最新安全版本

升級 IBM MQ Operator 可讓您升級佇列管理程式。

開始之前

重要: 本主題用於將 IBM MQ Operator 的部署升級至部署通道上的最新 Security Release。 如果這不適用於您的部署，請參閱 第 110 頁的『升級 IBM MQ Operator 及佇列管理程式』中說明的替代升級路徑。

若要在沒有網際網路連線功能的 Red Hat OpenShift 叢集裡部署 IBM MQ Operator ，請遵循 第 115 頁的『[已淘汰]在氣隙環境中準備升級至最新 IBM MQ 2.x 操作員或佇列管理程式』中的程序。

程序

1. 將 IBM MQ Operator 升級至更新版本。

   如果您已設定自動升級，則在發行新的「安全版本」時， IBM MQ Operator 會完成升級。

   如果您未設定自動升級，請手動核准 IBM MQ Operator 升級:

   - 如果有可用的升級， **Upgrade Status** 可能是「**升級可用**」。
   - 在此情況下，您可能可以使用可用的控制項來核准升級 IBM MQ Operator 的 **InstallPlan** 。
2. 將 IBM MQ 佇列管理程式升級至較新版本。

   下表說明每一個作用中「操作員」通道的 IBM MQ 佇列管理程式最新版本。 使用相關版本，請遵循 第 120 頁的『使用 Red Hat OpenShift 升級 IBM MQ 佇列管理程式』中的程序。

操作員通道	最新 IBM MQ 佇列管理程式
Operator2.0.0 2.0 版 (LTS)	9.3.0.17-r3
Operator3.0.0 3.1 版 (CD)	9.3.5.1-r2

OpenShift CP4I Operator2.0.0 移轉至 IBM MQ Operator 的 LTS 通道

升級 IBM MQ Operator 可讓您升級佇列管理程式。

開始之前

重要: 本主題適用於將 1.3.x Long Term Support (LTS) IBM MQ Operator 的部署升級至 LTS IBM MQ Operator 2.0.x 的串流 **僅限**。 如果這不適用於您的部署，請參閱 升級 IBM MQ Operator 和佇列管理程式中說明的替代升級路徑。

若要在沒有網際網路連線功能的 Red Hat OpenShift 叢集裡部署 IBM MQ Operator ，請遵循 第 115 頁的『[已淘汰]在氣隙環境中準備升級至最新 IBM MQ 2.x 操作員或佇列管理程式』中的程序。

重要: IBM MQ Operator 2.0.x 需要:

- Red Hat OpenShift Container Platform 4.12.

  若要升級，請遵循 升級 Red Hat OpenShift 中的程序。
- IBM Cloud Pak foundational services 3.19.x

當您從 IBM MQ Operator 1.3.x (2020.4) 升級時，多重實例佇列管理程式的 兩個 實例會同步重新啟動。 當您將 IBM MQ 的版本變更為 9.2.0.5-r3-eus 時，會發生此情況。 當您從 IBM MQ Operator 1.3.x 升級至

2.0.x 時，會漸進式更新 IBM MQ 佇列管理程式。 如果您已安裝 IBM Cloud Pak for Integration Platform UI，則在將 IBM Cloud Pak for Integration Platform UI 版本變更為 2020.4.1-8-eus 及 2022.2.1-0 時，會額外重新啟動 IBM MQ 。

程序

1. 在遵循步驟 2 中的鏈結之前，您必須先閱讀升級的下列必要資訊:
   - 您應該省略尚未安裝之元件的所有子步驟。 如果您未安裝此項目，則這包括 IBM Cloud Pak for Integration Platform UI 。
   - 步驟 2 會將您帶至 IBM Cloud Pak for Integration 說明文件。 在升級程序期間，您會回到下列 IBM MQ 主題，以升級 IBM MQ 作業: 升級 IBM MQ 佇列管理程式。
   - 建議所有 IBM MQ 使用者使用步驟 2 中鏈結的指示，以及任何適用於您環境的其他作業，至少完成下列作業:
     - 修補 IBM MQ Operator 及運算元 (修補 2020.4):
       - 在 v1.3-eus Operator Channel 中，將 IBM MQ Operator 升級至至少 1.3.5 版。
       - 將 IBM MQ Operand (佇列管理程式儲存器映像檔) 升級至至少 9.2.0.5-r3-eus 版。

         註: 建議您至少將 IBM MQ Operand 更新至這個版本，但這不是必要的。
     - 升級相依關係:
       - 升級 IBM Cloud Pak foundational services。
       - 升級 OpenShift Container Platform。
     - 升級操作器:
       - 將 IBM MQ Operator 升級至 2.0.23。
     - 升級功能:
       - 將 IBM MQ Operand (佇列管理程式儲存器映像檔) 升級至最新的 9.3.0 版本 (9.3.0.17-r3)，以接收最新的安全修正程式。
2. 完成 Upgrade from IBM MQ Operator 1.3-eus (IBM Cloud Pak for Integration 2020.4)，以升級 IBM MQ Operator 及佇列管理程式。

相關工作

第 115 頁的『[已淘汰]在氣隙環境中準備升級至最新 IBM MQ 2.x 操作員或佇列管理程式』
在沒有網際網路連線功能的 Red Hat OpenShift 叢集中，在升級「IBM MQ 2.x 操作員」或佇列管理程式之前，您需要採取一些準備步驟。

第 118 頁的『使用 Red Hat OpenShift 升級 IBM MQ Operator』
您可以使用 Red Hat OpenShift Web 主控台或 CLI 來升級 IBM MQ Operator 。

第 120 頁的『使用 Red Hat OpenShift 升級 IBM MQ 佇列管理程式』

OpenShift CP4I CD 移轉至 IBM MQ Operator 的現行 CD 通道

從舊版 IBM MQ Operator 升級至 3.1.3 版。 升級操作器可讓您升級佇列管理程式。

開始之前

本主題適用於將 IBM MQ Operator 3.1.0 版之前的 Continuous Delivery (CD) 部署升級至 3.1.3 **僅**版。 如果這不適用於您的部署，請參閱 升級 IBM MQ Operator 和佇列管理程式中說明的替代升級路徑。

若要升級至 IBM MQ Operator 3.1.3，您必須執行 Red Hat OpenShift Container Platform 4.12 或更新版本。 若要升級平台，請參閱 升級 Red Hat OpenShift。

註: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版本，即偶數次要版本，例如 4.14 和 4.16。

程序

1. 選擇性的: **升級目前為 CD 2.0.0 之前版本的 IBM MQ Operator** 。

   如果您的 IBM MQ Operator 目前是 1.x CD 版本，請先遵循 第 113 頁的『將 1.x CD IBM MQ Operator 移轉至 2.0.x 版』中的程序，然後回到這裡以升級至最新 CD 版本。
2. 選擇性的: **將目前為 2.2.x 或 2.3.x 版的 IBM MQ Operator 升級至 2.4.x**。

   如果您的 IBM MQ Operator 目前是 2.2.x 或 2.3.x 版本，請遵循 第 114 頁的『移轉至 IBM MQ Operator 的 v2.4 通道』中的相關步驟，然後回到這裡以升級至最新 CD 版本。 請注意，這是升級至 3.1.3 版之前必要的必要步驟。
3. **升級元件**。

   選擇下列其中一個選項:

   - **選項 1**: 如果您是 CP4I 使用者，或者您已使用 CP4I 授權來部署至少一個佇列管理程式，請遵循相關步驟，透過產生的升級計劃來 **升級所有元件**，包括 IBM MQ Operator 及佇列管理程式:
     - 若要從 2023.2 版升級，請參閱 透過產生升級方案從 2023.2 升級。
     - 若要從 2022.2 版升級，請參閱 透過產生升級方案從 2022.2。
   - **選項 2**: 適用於所有其他使用者:

     a. **鏡映映像檔 (僅限氣隙)**。

        您必須鏡映 IBM MQ 映像檔。 僅使用下列值，完成下列鏈結中的步驟:

        ```
        export OPERATOR_PACKAGE_NAME=ibm-mq
        export OPERATOR_VERSION=3.1.3
        ```

        您應該省略 3.5 小節「配置叢集」，因為在先前安裝或升級期間應該已設定映像檔登錄的連線。

        鏈結: 氣隙叢集的鏡映映像檔。

     b. **將 IBM MQ Operator 升級至 3.1.3**。

        請參閱 第 118 頁的『使用 Red Hat OpenShift 升級 IBM MQ Operator』。

     c. **升級實例**。

        若要接收最新特性及安全修正程式，請將 IBM MQ Operand (佇列管理程式儲存器映像檔) 升級至最新 CD 版本 (9.3.5.1-r2)。 請參閱 第 120 頁的『使用 Red Hat OpenShift 升級 IBM MQ 佇列管理程式』。
4. 選擇性的: **將 Red Hat OpenShift Container Platform 4.12 升級至 4.14**。

   從 IBM MQ Operator 3.0.0 開始，需要 Red Hat OpenShift Container Platform 4.12 。 請注意，您可以選擇性地選擇進一步升級至 Red Hat OpenShift 4.14。 若要驗證每一個 IBM MQ Operator 通道的相容版本，請參閱 第 11 頁的『相容的 Red Hat OpenShift Container Platform 版本』。 若要升級，請參閱 升級 Red Hat OpenShift。
5. 選擇性的: **固定 IBM MQ Operator 的特定型錄來源**。

   如果您要升級的安裝使用 IBM MQ Operator 型錄，則應該固定 IBM MQ Operator 的特定型錄來源。 請參閱 移至每一個運算子的特定型錄來源。

相關工作

第 115 頁的『[已淘汰]在氣隙環境中準備升級至最新 IBM MQ 2.x 操作員或佇列管理程式』
在沒有網際網路連線功能的 Red Hat OpenShift 叢集中，在升級「IBM MQ 2.x 操作員」或佇列管理程式之前，您需要採取一些準備步驟。

OpenShift CP4I Operator2.2.0 **將 *1.x CD IBM MQ Operator* 移轉至 *2.0.x* 版**

升級 IBM MQ Operator 可讓您升級佇列管理程式。

開始之前

重要: 本主題適用於將 IBM MQ Operator 2.0.x 版之前的 Continuous Delivery (CD) 部署升級至 2.0.x 版 (僅限)。 如果這不適用於您的部署，請參閱 升級 IBM MQ Operator 和佇列管理程式中說明的替代升級路徑。

若要在沒有網際網路連線功能的 Red Hat OpenShift 叢集裡部署 IBM MQ Operator ，請遵循 第 115 頁的『[已淘汰]在氣隙環境中準備升級至最新 IBM MQ 2.x 操作員或佇列管理程式』中的程序。

若要完成此升級，必須符合 IBM MQ Operator 2.0.0 的下列需求:

- Red Hat OpenShift Container Platform 4.12.

  若要升級，請遵循 升級 Red Hat OpenShift 中的程序。
- IBM Cloud Pak foundational services 3.19

程序

1. 在遵循步驟 2 中的鏈結之前，您必須先閱讀升級的下列必要資訊:
   - 省略尚未安裝之元件的所有子步驟。 如果您未安裝此項目，則這包括 IBM Cloud Pak for Integration Platform UI 。
   - 步驟 2 會將您帶至 IBM Cloud Pak for Integration 說明文件。 在升級程序期間，您會回到下列 IBM MQ 主題，以升級 IBM MQ 作業: 升級 IBM MQ 佇列管理程式。
   - 建議所有 IBM MQ 使用者使用步驟 2 中鏈結的指示，以及任何適用於您環境的其他作業，至少完成下列作業:
     - 修補 IBM MQ Operator 及運算元 (修補 2021.4):
       - 在 v1.8 操作器通道中，將 IBM MQ Operator 升級至至少 1.8.0 版。
       - 將 IBM MQ Operand (佇列管理程式儲存器映像檔) 升級至至少 9.2.5.0-r3 版。

         註: 建議您將 IBM MQ Operand 更新至現行版本 (9.3.0.17-r3) ，但這不是必要項目。
     - 升級相依關係:
       - 升級 IBM Cloud Pak foundational services。
       - 升級 OpenShift Container Platform。
     - 升級操作器:
       - 將 IBM MQ Operator 升級至 2.0.23。
     - 升級功能:
       - 將 IBM MQ Operand (佇列管理程式儲存器映像檔) 升級至最新的 9.3.0 版本 (9.3.0.17-r3) ，以接收最新的安全修正程式。
2. 完成 從 IBM MQ Operator 1.8 (IBM Cloud Pak for Integration 2021.4) 或更舊版本 CD IBM MQ Operator 升級 IBM MQ Operator 和佇列管理程式。

下一步

現在，您已準備好將 IBM MQ Operator 和佇列管理程式升級至最新的 CD 版本 (3.1.3)。 請參閱 第 112 頁的『移轉至 IBM MQ Operator 的現行 CD 通道』。

OpenShift Operator2.4.0 CP4I 移轉至 IBM MQ Operator 的 v2.4 通道

升級 IBM MQ Operator 可讓您升級佇列管理程式。

開始之前

重要: 本主題適用於將 IBM MQ Operator 2.4.0 版之前的 Continuous Delivery (CD) 部署升級至 2.4.8 **版**。 這是升級至 IBM MQ Operator 最新 CD 版本的中間步驟; v2.4 通道不會接收安全更新。 如果這不適用於您的部署，請參閱 升級 IBM MQ Operator 和佇列管理程式中說明的替代升級路徑。

若要在沒有網際網路連線功能的 Red Hat OpenShift 叢集裡部署 IBM MQ Operator，請遵循 第 115 頁的『[已淘汰]在氣隙環境中準備升級至最新 IBM MQ 2.x 操作員或佇列管理程式』 中的程序。

若要完成此升級，必須符合 IBM MQ Operator 2.4.8 的下列需求:

- Red Hat OpenShift Container Platform 4.12.

  若要升級，請遵循 升級 Red Hat OpenShift 中的程序。

  註: 僅支援 OpenShift Container Platform Extended Update Support (EUS) 版本，即偶數次要版本，例如 4.14 和 4.16。
- IBM Cloud Pak foundational services 3.19 至 3.24 (含)。

程序

1. 選擇性的: **升級目前為 CD 2.0.0 之前版本的 IBM MQ Operator**

   如果您的 IBM MQ Operator 目前是 1.x CD 版本，請先遵循 第 113 頁的『將 1.x CD IBM MQ Operator 移轉至 2.0.x 版』 中的程序，然後回到這裡以升級至最新 2.4 版本。
2. **將 CD 2.x.x 版的 IBM MQ Operator 升級至最新 2.4 版本 (2.4.8)**。

   請遵循 第 118 頁的『使用 Red Hat OpenShift 升級 IBM MQ Operator』 中的程序。
3. 選擇性的: **升級 IBM Cloud Pak for Integration 的其他元件**。

   如果您是 IBM Cloud Pak for Integration 的使用者，則可能有您想要升級的其他元件。 如需升級其他元件的步驟，請根據您的部署，參閱下面的相關步驟:

   - 選項 1: 從 IBM MQ 操作員升級 2.0.x/2.1.x (IBM Cloud Pak for Integration 2022.2)。
   - 選項 2: 從 IBM MQ 升級 2.2.x/2.3.x (IBM Cloud Pak for Integration 2022.4)。
4. 選擇性的: **升級 IBM Cloud Pak foundational services**。

   如果您是 IBM Cloud Pak for Integration 的使用者，則可能想要將 IBM Cloud Pak foundational services 從 3.19.x 版升級至 3.24.x 版。 如需完成此升級的步驟，請參閱 升級 IBM Cloud Pak foundational services。

相關工作

第 115 頁的『[已淘汰]在氣隙環境中準備升級至最新 IBM MQ 2.x 操作員或佇列管理程式』
在沒有網際網路連線功能的 Red Hat OpenShift 叢集中，在升級「IBM MQ 2.x 操作員」或佇列管理程式之前，您需要採取一些準備步驟。

第 118 頁的『使用 Red Hat OpenShift 升級 IBM MQ Operator』
您可以使用 Red Hat OpenShift Web 主控台或 CLI 來升級 IBM MQ Operator。

第 120 頁的『使用 Red Hat OpenShift 升級 IBM MQ 佇列管理程式』

OpenShift CP4I Linux Deprecated 在氣隙環境中準備升級至最新 IBM MQ 2.x 操作員或佇列管理程式

在沒有網際網路連線功能的 Red Hat OpenShift 叢集中，在升級「IBM MQ 2.x 操作員」或佇列管理程式之前，您需要採取一些準備步驟。

開始之前

註: 這些指示適用於在氣隙環境中升級至 IBM MQ Operator 2.x 版本。 若要升級至 IBM MQ Operator 3.0.0 以及更新版本，請參閱 第 110 頁的『升級 IBM MQ Operator 及佇列管理程式』，並特別注意氣隙特定步驟。

本主題假設您已配置本端映像檔登錄，其中鏡映先前發行的 IBM Cloud Pak for Integration 映像檔。

關於這項作業

您必須先鏡映最新的 IBM Cloud Pak for Integration 映像檔，然後才能在氣隙環境中升級 IBM MQ Operator 或佇列管理程式。

請注意，此作業中的前四個步驟與您在 第 101 頁的『在氣隙環境中安裝 IBM MQ Operator 2.x』時採取的步驟相同。

程序

1. 建立安裝程式及映像檔庫存的環境變數。

   使用安裝程式映像檔名稱及映像檔庫存來建立下列環境變數:

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

   其中 *version_number* 是您要用來執行氣隙安裝的案例版本。 如需可用案例版本的清單，請參閱 https://github.com/IBM/cloud-pak/tree/master/repo/case/ibm-mq。 檢閱 IBM MQ Operator 的版本支援，以判定要選擇的操作員通道。

2. 下載 IBM MQ 安裝程式及映像檔庫存。

   將 `ibm-mq` 安裝程式及映像檔庫存下載至防禦主機:

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

3. 以叢集管理者身分登入 OpenShift Container Platform 叢集。

   下列是登入 OpenShift Container Platform 叢集的範例指令:

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

4. 鏡映映像檔並配置叢集。

   完成下列步驟，以鏡映映像檔並配置叢集:

   註: 請勿在任何指令中的雙引號內使用波狀符號。 例如，請勿使用 `args "--registry registry --user registry_userid --pass registry_password --inputDir ~/offline"`。 波狀符號不會展開，您的指令可能會失敗。

   a. 儲存所有來源 Docker 登錄的鑑別認證。

      所有 IBM Cloud Platform Common Services、 IBM MQ Operator 映像檔和 IBM MQ Advanced Developer 映像檔都儲存在不需要鑑別的公用登錄中。 不過， IBM MQ Advanced Server (非開發人員)、其他產品和協力廠商元件需要一或多個鑑別登錄。 下列登錄需要鑑別:

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

      如需這些登錄的相關資訊，請參閱 建立登錄名稱空間。

      您必須執行下列指令，為所有需要鑑別的登錄配置認證。 針對每一個此類登錄分別執行指令:

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

      此指令會將登錄認證儲存並快取在檔案系統上位於 `$HOME/.airgap/secrets` 位置的檔案中。

b. 使用本端 Docker 登錄連線資訊來建立環境變數。

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

附註: Docker 登錄使用標準埠，例如 80 或 443。 如果 Docker 登錄使用非標準埠，請使用語法 *host*:*port* 來指定埠。 例如:

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

c. 配置本端 Docker 登錄的鑑別密碼。

附註: 此步驟只需要執行一次。

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

此指令會將登錄認證儲存並快取在檔案系統上位於 $HOME/.airgap/secrets 位置的檔案中。

d. 配置廣域映像檔取回密鑰和 **ImageContentSourcePolicy**。

i) 檢查是否需要重新啟動節點。

- 在 OpenShift Container Platform 4.4 版及更新版本中，以及在使用氣隙的 IBM MQ Operator 新安裝上，此步驟會重新啟動所有叢集節點。 可能要到套用新的取回密鑰之後，叢集資源才可供使用。
- 在 IBM MQ Operator 1.8 中， CASE 已更新為包含映像檔的其他鏡映來源。 因此，當您從舊版 IBM MQ Operator 升級至 1.8 版或更高版本時，會觸發節點重新啟動。
- 若要檢查此步驟是否需要重新啟動節點，請將 --dry-run 選項新增至此步驟的程式碼。 這會產生最新的 **ImageContentSourcePolicy** ，並將它顯示在主控台視窗 (**stdout**) 中。 如果此 **ImageContentSourcePolicy** 不同於配置的叢集 **ImageContentSourcePolicy**，則會進行重新啟動。

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

ii) 若要配置廣域映像檔取回密碼及 **ImageContentSourcePolicy**，請在沒有 --dry-run 選項的情況下執行此步驟的程式碼:

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

e. 確認 **ImageContentSourcePolicy** 資源已建立。

```
oc get imageContentSourcePolicy
```

f. 選用項目: 如果您使用不安全的登錄，則必須將本端登錄新增至叢集 **insecureRegistries** 清單。

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

g. 驗證叢集節點狀態。

```
oc get nodes
```

在套用 **imageContentsourcePolicy** 及廣域映像檔取回密鑰之後，您可能會看到節點狀態為 **Ready**、**Scheduling** 或 **Disabled**。 請等待所有節點都顯示 **Ready** 狀態。

h. 將映像檔鏡映至本端登錄。

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

5. 升級型錄來源。

請使用執行先前步驟的相同終端機。

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

下一步

若要完成 IBM Cloud Pak for Integration 升級，您可能需要回到 IBM Cloud Pak for Integration 說明文件。

否則，您現在已準備好完成下列其中一項作業，來升級 IBM MQ Operator 和佇列管理程式:

- 第 118 頁的『使用 Red Hat OpenShift 升級 IBM MQ Operator』
- 第 120 頁的『使用 Red Hat OpenShift 升級 IBM MQ 佇列管理程式』

OpenShift CP4I 使用 Red Hat OpenShift 升級 IBM MQ Operator

您可以使用 Red Hat OpenShift Web 主控台或 CLI 來升級 IBM MQ Operator 。

程序

若要使用 Red Hat OpenShift 升級 IBM MQ Operator ，請完成下列其中一項作業:

- 第 118 頁的『使用 Red Hat OpenShift Web 主控台升級 IBM MQ Operator』
- 第 119 頁的『使用 Red Hat OpenShift CLI 升級 IBM MQ Operator』

OpenShift CP4I 使用 *Red Hat OpenShift Web* 主控台升級 *IBM MQ Operator*

可以使用 Operator Hub 來升級 IBM MQ Operator 。

開始之前

註: IBM MQ Operator 的最新 CD 版本是 3.1.3。 IBM MQ Operator 的最新 LTS 版本是 2.0.23。 如需最新的 IBM MQ Operator 版本注意事項，請參閱 第 30 頁的『IBM MQ Operator 的發行歷程』。

登入 Red Hat OpenShift 叢集 Web 主控台。

程序

1. 檢閱 第 10 頁的『IBM MQ Operator 的版本支援』，以判定要升級至哪個操作員通道。
2. 套用最新的型錄來源。

   如果您使用 IBM MQ 特定的型錄來源 (所有氣隙安裝)，而不是 `ibm-operator-catalog`，則必須套用 IBM MQ 版本的型錄來源。

遵循 將型錄來源新增至叢集中的指示。

註: 如果您已完成氣隙 鏡映映像檔 (僅限氣隙)的操作器安裝步驟，則只需要完成套用型錄來源的步驟。 例如:

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

3. 升級 IBM MQ Operator。 新的主要/次要 IBM MQ Operator 版本會透過新的「訂閱通道」來遞送。 若要將操作器升級至新的主要/次要版本，您將需要在 IBM MQ Operator Subscription 中更新選取的通道。
   a) 從導覽窗格中，按一下 **操作器 > 已安裝的操作器**。
      會顯示指定專案中所有已安裝的操作員。
   b) 選取 **IBM MQ 操作器**
   c) 導覽至 **訂閱** 標籤
   d) 按一下 **通道**
      即會顯示「 **變更訂閱更新通道** 」視窗。
   e) 選取所需的通道，然後按一下 **儲存**。
      操作員將升級至新通道可用的最新版本。 請參閱 第 10 頁的『IBM MQ Operator 的版本支援』。

OpenShift CP4I 使用 *Red Hat OpenShift CLI* 升級 *IBM MQ Operator*

可以從指令行升級 IBM MQ Operator 。

開始之前

註: IBM MQ Operator 的最新 CD 版本是 3.1.3。 IBM MQ Operator 的最新 LTS 版本是 2.0.23。 如需最新的 IBM MQ Operator 版本注意事項，請參閱 第 30 頁的『IBM MQ Operator 的發行歷程』。

使用 **oc login** 登入叢集。

您必須先鏡映最新的 IBM Cloud Pak for Integration 映像檔，然後才能在氣隙環境中升級 IBM MQ Operator 。 若要升級至 IBM MQ Operator 3.0 或更高版本， 移轉至 IBM MQ Operator 的現行 CD 通道 包括氣隙特定步驟。 若要升級至舊版 IBM MQ 操作器，請參閱 Deprecated 在氣隙環境中準備升級至最新 IBM MQ 2.x 操作器或佇列管理程式。

程序

1. 檢閱 第 10 頁的『IBM MQ Operator 的版本支援』 ， 以判定要升級至哪個操作員通道。
2. 套用最新的型錄來源。

   如果您使用 IBM MQ 特定的型錄來源 (所有氣隙安裝) ， 而不是 `ibm-operator-catalog`，則必須套用 IBM MQ 版本的型錄來源。

   遵循 將型錄來源新增至叢集中的指示。

   註: 如果您已完成氣隙 鏡映映像檔 (僅限氣隙)的操作器安裝步驟，則只需要完成套用型錄來源的步驟。 例如:

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

3. 升級 IBM MQ Operator。 新的主要/次要 IBM MQ Operator 版本會透過新的「訂閱通道」來遞送。 若要將您的操作器升級至新的主要或次要版本，您需要在 IBM MQ Operator Subscription 中更新選取的頻道。
   a) 請確定必要的「 IBM MQ Operator 升級通道」可用。

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

   b) 修補 `Subscription` 以移至所需的更新通道 (其中 *vX。Y* 是前一個步驟中所識別的所需更新通道。

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

OpenShift CP4I 使用 Red Hat OpenShift 升級 IBM MQ 佇列管理程式

開始之前

在升級 IBM MQ 佇列管理程式的程序中，您可能已從 IBM Cloud Pak for Integration 文件傳送至這個主題。

程序

若要使用 Red Hat OpenShift 來升級 IBM MQ 佇列管理程式，請完成下列其中一項作業:

- 第 120 頁的『使用 Red Hat OpenShift Web 主控台升級 IBM MQ 佇列管理程式』
- 第 121 頁的『使用 Red Hat OpenShift CLI 升級 IBM MQ 佇列管理程式』
- 第 121 頁的『使用平台使用者介面在 Red Hat OpenShift 中升級 IBM MQ 佇列管理程式』

下一步

若要完成 IBM Cloud Pak for Integration 升級，您可能需要回到 IBM Cloud Pak for Integration 說明文件。

OpenShift CP4I 使用 *Red Hat OpenShift Web* 主控台升級 *IBM MQ* 佇列管理程式

可以使用 Operator Hub 在 Red Hat OpenShift 中升級使用 IBM MQ Operator 部署的 IBM MQ 佇列管理程式。

開始之前

註: 最新 CD 版本的 IBM MQ 佇列管理程式為 9.3.5.1-r2。 最新 LTS 版本的 IBM MQ 佇列管理程式為 9.3.0.17-r3。 如需最新的 IBM MQ 佇列管理程式版本注意事項，請參閱 第 53 頁的『與 IBM MQ Operator 搭配使用的佇列管理程式儲存器映像檔的版本歷程』。

- 登入 Red Hat OpenShift 叢集 Web 主控台。
- 確定您的 IBM MQ Operator 正在使用想要的「更新通道」。 請參閱 第 118 頁的『使用 Red Hat OpenShift 升級 IBM MQ Operator』。

您必須先鏡映最新的 IBM Cloud Pak for Integration 映像檔，然後才能在氣隙環境中升級佇列管理程式。 若要升級至 IBM MQ Operator 3.0 或更高版本， 移轉至 IBM MQ Operator 的現行 CD 通道 包括氣隙特定步驟。 若要升級至舊版 IBM MQ 操作器，請參閱 Deprecated 在氣隙環境中準備升級至最新 IBM MQ 2.x 操作器或佇列管理程式。

程序

1. 從導覽窗格中，按一下 **操作器 > 已安裝的操作器**。
   會顯示指定專案中所有已安裝的操作員。
2. 選取 **IBM MQ 操作器**。
   即會顯示「**IBM MQ 操作員**」視窗。
3. 導覽至 **佇列管理程式** 標籤。
   即會顯示「**佇列管理程式詳細資料**」視窗。
4. 選取您要升級的佇列管理程式。
5. 導覽至 YAML 標籤。
6. 必要的話，請更新下列欄位，以符合所需的 IBM MQ 佇列管理程式版本升級。
   - spec.version
   - spec.license.licence

   如需 IBM MQ Operator 版本與 IBM MQ 佇列管理程式儲存器映像檔的對映，請參閱 第 53 頁的『與 IBM MQ Operator 搭配使用的佇列管理程式儲存器映像檔的版本歷程』。

7. 儲存更新的佇列管理程式 YAML。

OpenShift CP4I 使用 *Red Hat OpenShift CLI* 升級 *IBM MQ* 佇列管理程式

使用 IBM MQ Operator 部署的 IBM MQ 佇列管理程式可以使用指令行在 Red Hat OpenShift 中升級。

開始之前

註: 最新 CD 版本的 IBM MQ 佇列管理程式為 9.3.5.1-r2。 最新 LTS 版本的 IBM MQ 佇列管理程式為 9.3.0.17-r3。 如需最新的 IBM MQ 佇列管理程式版本注意事項，請參閱 第 53 頁的『與 IBM MQ Operator 搭配使用的佇列管理程式儲存器映像檔的版本歷程』。

您必須是叢集管理者，才能完成這些步驟。

- 使用 `oc login` 登入 Red Hat OpenShift 指令行介面 (CLI)。
- 確定您的 IBM MQ Operator 正在使用想要的「更新通道」。 請參閱 第 110 頁的『升級 IBM MQ Operator 及佇列管理程式』。

您必須先鏡映最新的 IBM Cloud Pak for Integration 映像檔，然後才能在氣隙環境中升級佇列管理程式。 若要升級至 IBM MQ Operator 3.0 或更高版本， 移轉至 IBM MQ Operator 的現行 CD 通道 包括氣隙特定步驟。 若要升級至舊版 IBM MQ 操作器，請參閱 Deprecated 在氣隙環境中準備升級至最新 IBM MQ 2.x 操作器或佇列管理程式。

程序

必要的話，編輯 **QueueManager** 資源以更新下列欄位，以符合所需的 IBM MQ 佇列管理程式版本升級。

- spec.version
- spec.license.licence

如需通道與 IBM MQ Operator 版本及 IBM MQ 佇列管理程式版本的對映，請參閱 第 10 頁的『IBM MQ Operator 的版本支援』。

使用下列指令:

```
oc edit queuemanager my_qmgr
```

其中 *my_qmgr* 是您要升級的 QueueManager 資源的名稱。

CP4I 使用平台使用者介面在 *Red Hat OpenShift* 中升級 *IBM MQ* 佇列管理程式

使用 IBM MQ Operator 部署的 IBM MQ 佇列管理程式，可以使用 IBM Cloud Pak for Integration Platform UI (previously the Platform Navigator)在 Red Hat OpenShift 中升級。

開始之前

註: 最新 CD 版本的 IBM MQ 佇列管理程式為 9.3.5.1-r2。 最新 LTS 版本的 IBM MQ 佇列管理程式為 9.3.0.17-r3。 如需最新的 IBM MQ 佇列管理程式版本注意事項，請參閱 第 53 頁的『與 IBM MQ Operator 搭配使用的佇列管理程式儲存器映像檔的版本歷程』。

- 登入包含您要升級之佇列管理程式的名稱空間中的 IBM Cloud Pak for Integration Platform UI 。
- 確定您的 IBM MQ Operator 正在使用想要的「更新通道」。 請參閱第 110 頁的『升級 IBM MQ Operator 及佇列管理程式』。

您必須先鏡映最新的 IBM Cloud Pak for Integration 映像檔，然後才能在氣隙環境中升級佇列管理程式。 若要升級至 IBM MQ Operator 3.0 或更高版本， 移轉至 IBM MQ Operator 的現行 CD 通道 包括氣隙特定步驟。 若要升級至舊版 IBM MQ 操作器，請參閱 Deprecated 在氣隙環境中準備升級至最新 IBM MQ 2.x 操作器或佇列管理程式。

程序

1. 從 IBM Cloud Pak for Integration Platform UI (previously the Platform Navigator) 首頁中，按一下 **執行時期** 標籤。
2. 具有可用升級項目的佇列管理程式在 版旁邊具有藍色 **i** 。 按一下 **i** 以顯示 **可用的新版本**。
3. 按一下您要升級之佇列管理程式最右邊的三個點，然後按一下 **變更版本**。
4. 在 **選取新通道或版本**下，選取所需的升級版本。
5. 按一下 **變更版本**。

結果
佇列管理程式已升級。

OpenShift 使用 IBM MQ Operator 來配置佇列管理程式

配置範例; 配置 HA; 從 OpenShift 叢集外部連接; 與 CP4i 儀表板整合; 與 Instana 追蹤整合; 使用自訂 MQSC 和 INI 檔案建置映像檔; 新增自訂註釋和標籤。

關於這項作業

程序

- 第 122 頁的『配置佇列管理程式的範例』.
- 第 130 頁的『使用 IBM MQ Operator 來配置佇列管理程式的高可用性』.
- 第 140 頁的『配置路徑以從 Red Hat OpenShift 叢集外部連接至佇列管理程式』.
- 第 142 頁的『與 IBM Cloud Pak for Integration 作業儀表板整合』.
- 第 143 頁的『整合 IBM MQ 與 IBM Instana 追蹤』.
- 第 149 頁的『使用 Red Hat OpenShift CLI 以自訂 MQSC 及 INI 檔案建置映像檔』.
- 第 151 頁的『將自訂註釋和標籤新增至佇列管理程式資源』.
- 第 151 頁的『停用執行時期 Webhook 檢查』.
- 第 152 頁的『停用佇列管理程式規格的預設值更新』.

OpenShift CP4I 配置佇列管理程式的範例

可以透過調整 QueueManager 自訂資源的內容來配置佇列管理程式。

關於這項作業
請使用下列範例，以協助您使用 QueueManager YAML 檔案來配置佇列管理程式。

程序

- 第 122 頁的『範例: 提供 MQSC 及 INI 檔案』
- 第 125 頁的『範例: 配置具有相互 TLS 鑑別的佇列管理程式』

OpenShift CP4I 範例: 提供 *MQSC* 及 *INI* 檔案

此範例會建立 Kubernetes ConfigMap，其中包含兩個 MQSC 檔及一個 INI 檔。 然後會部署佇列管理程式來處理這些 MQSC 及 INI 檔案。

關於這項作業
部署佇列管理程式時，可以提供 MQSC 及 INI 檔。 MQSC 及 INI 資料必須定義在一或多個 Kubernetes ConfigMaps 及 Secrets 中。 這些必須建立在您將在其中部署佇列管理程式的名稱空間 (專案) 中。

註: 當 MQSC 或 INI 檔案包含機密資料時，應該使用 Kubernetes 密鑰。

範例

下列範例會建立 Kubernetes ConfigMap，其中包含兩個 MQSC 檔及一個 INI 檔。 然後會部署佇列管理程式來處理這些 MQSC 及 INI 檔案。

範例 ConfigMap -在叢集裡套用下列 YAML:

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

範例 QueueManager -使用指令行或 Red Hat OpenShift Container Platform Web 主控台，以下列配置部署佇列管理程式:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: mqsc-ini-qm
spec:
  version: 9.3.5.1-r2
  license:
    accept: false
    license: L-AMRD-XH6P3Q
    use: Production
  web:
    enabled: true
  queueManager:
    name: "MQSCINI"
    mqsc:
    - configMap:
        name: mqsc-ini-example
        items:
        - example1.mqsc
        - example2.mqsc
    ini:
    - configMap:
        name: mqsc-ini-example
        items:
        - example.ini
    storage:
      queueManager:
        type: ephemeral
```

重要: 如果您接受 IBM MQ Advanced 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 mq.ibm.com/v1beta1 的授權參考資料 。

其他資訊:

- 佇列管理程式可以配置為使用單一 Kubernetes ConfigMap 或「密鑰」(如本範例所示) 或多個 ConfigMaps 及「密鑰」。
- 您可以選擇使用 Kubernetes ConfigMap 或 Secret 中的所有 MQSC 及 INI 資料 (如本範例所示)，或將每一個佇列管理程式配置為只使用可用檔案的子集。
- MQSC 及 INI 檔案會根據其索引鍵按字母順序進行處理。 因此，不論 `example1.mqsc` 在佇列管理程式配置中出現的順序為何，一律會在 `example2.mqsc` 之前處理。
- 如果多個 MQSC 或 INI 檔案在多個 Kubernetes ConfigMaps 或「密鑰」之間具有相同的索引鍵，則會根據這些檔案在佇列管理程式配置中的定義順序來處理這組檔案。
- 當佇列管理程式 Pod 執行時，不會挑選對 Kubernetes ConfigMap 所做的任何變更，因為 IBM MQ Operator 不知道該變更。 如果您對 ConfigMap 進行變更 (例如對 MQSC 指令或 INI 檔案進行變更)，則必須手動重新啟動佇列管理程式以取得這些變更。 對於單一實例佇列管理程式，請刪除 Pod 以觸發必要的重

新啟動。 若為原生 HA 部署，請先刪除備用 Pod，以重新啟動它們。 當它們再次處於執行中狀態時，請刪除作用中 Pod 以重新啟動它。 此重新啟動順序可確保佇列管理程式的關閉時間下限。

OpenShift CP4I 使用 *OpenSSL* 建立自簽 *PKI*

IBM MQ 可讓您使用交互 TLS 進行鑑別，其中連線兩端提供憑證，並使用憑證中的詳細資料來建立與佇列管理程式的身分。 這個主題呈現如何使用 OpenSSL 指令行工具來建立範例「公開金鑰基礎架構 (PKI)」，並建立兩個憑證，以在其他範例中使用。

開始之前

確定已安裝 OpenSSL 指令行工具。

安裝 IBM MQ client，並將 `samp/bin` 及 `bin` 新增至 *PATH*。 您需要 **runmqicred** 指令，該指令可以安裝為 IBM MQ client 的一部分，如下所示:

- Windows Linux Windows 和 Linux: 從 https://ibm.biz/mq93redistclients 安裝適用於您作業系統的 IBM MQ 可重新配送用戶端。
- macOS Mac: 下載並設定 IBM MQ MacOS Toolkit: https://developer.ibm.com/tutorials/mq-macos-dev/

關於這項作業

重要: 這裡說明的範例不適用於正式作業環境，僅作為快速進行的範例。 憑證管理是進階使用者的複雜主題。 對於正式作業，您必須考量替換、撤銷、金鑰長度、災難回復等事項。

這些步驟已使用 OpenSSL 3.1.4 進行測試。

程序

1. 建立要用於內部憑證管理中心的私密金鑰

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

   內部憑證管理中心的私密金鑰建立在稱為 *ca.key* 的檔案中。 此檔案應該保持安全且機密-它將用來簽署內部憑證管理中心的憑證。
2. 發出內部憑證管理中心的自簽憑證

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

   `-days` 指定主要 CA 憑證的有效天數。

   憑證建立在稱為 *ca.crt* 的檔案中。 此憑證包含內部憑證管理中心的公用資訊，且可自由共用。
3. 建立佇列管理程式的私密金鑰和憑證

   a) 建立佇列管理程式的私密金鑰和憑證簽署要求

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

   在稱為 *example-qm.key* 的檔案中建立私密金鑰，並在稱為 *example-qm.csr* 的檔案中建立憑證簽署要求。

   b) 使用內部憑證管理中心簽署佇列管理程式金鑰

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

   `-days` 指定憑證的有效天數。

   已簽章的憑證建立在稱為 *example-qm.crt* 的檔案中

   c) 使用佇列管理程式金鑰及憑證建立 Kubernetes 密鑰

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

即會建立稱為 *example-qm-tls* 的 Kubernetes 密鑰。 此密鑰包含佇列管理程式、公用憑證及 CA 憑證的私密金鑰。

4. 建立應用程式的私密金鑰和憑證

a) 建立應用程式的私密金鑰和憑證簽署要求

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

在稱為 *example-app1.key* 的檔案中建立私密金鑰，並在稱為 *example-app1.csr* 的檔案中建立憑證簽署要求。

b) 使用內部憑證管理中心簽署佇列管理程式金鑰

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

`-days` 指定憑證的有效天數。

在稱為 *example-app1.crt* 的檔案中建立已簽章的憑證。

c) 使用應用程式的金鑰和憑證來建立 PKCS#12 金鑰儲存庫

IBM MQ 使用金鑰資料庫，而不是個別金鑰檔。 儲存器化佇列管理程式會從「密鑰」建立佇列管理程式的金鑰資料庫，但對於用戶端應用程式，您需要手動建立金鑰資料庫。

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

將 *<PASSWORD>* 取代為您自己選擇的密碼。

金鑰儲存庫建立在稱為 *example-app1.p12* 的檔案中。 應用程式的金鑰及憑證儲存在內，其「標籤」或「一般名稱」為 "example-app1" ， 以及 CA 憑證。

d) 如果您是使用 arm64 Apple Mac ，則需要配置另一個結合應用程式與 CA 憑證的檔案。
例如：

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

相關工作

第 125 頁的『範例: 配置具有相互 TLS 鑑別的佇列管理程式』
此範例使用 IBM MQ Operator 將佇列管理程式部署至 OpenShift Container Platform 。 交互 TLS 用於鑑別，以從 TLS 憑證對映至佇列管理程式中的身分。

第 133 頁的『範例: 使用 IBM MQ Operator 來配置原生 HA』
此範例會使用 IBM MQ Operator，將使用原生高可用性特性的佇列管理程式部署至 OpenShift Container Platform 。 交互 TLS 用於鑑別，以從 TLS 憑證對映至佇列管理程式中的身分。

第 138 頁的『使用 IBM MQ Operator 來配置多重實例佇列管理程式』
此範例會使用 IBM MQ Operator，將使用的多重實例佇列管理程式部署至 OpenShift Container Platform 。 交互 TLS 用於鑑別，以從 TLS 憑證對映至佇列管理程式中的身分。

OpenShift CP4I Linux 範例: 配置具有相互 *TLS* 鑑別的佇列管理程式

此範例使用 IBM MQ Operator 將佇列管理程式部署至 OpenShift Container Platform 。 交互 TLS 用於鑑別，以從 TLS 憑證對映至佇列管理程式中的身分。

開始之前

若要完成此範例，您必須先完成下列必要條件:

- 針對此範例建立 OpenShift Container Platform (OCP) 專案/名稱空間。
- 在指令行上，登入 OCP 叢集，並切換至上述名稱空間。
- 請確定上述名稱空間中已安裝 IBM MQ Operator 且可供使用。

關於這項作業

此範例提供自訂資源 YAML，定義要部署至 OpenShift Container Platform 的佇列管理程式。 它也會詳細說明在啟用 TLS 的情況下部署佇列管理程式所需的其他步驟。

程序

1. 建立憑證配對，如 第 124 頁的『使用 OpenSSL 建立自簽 PKI』 中所述。
2. 建立包含 MQSC 指令及 INI 檔案的配置對映

   建立包含 MQSC 指令的 Kubernetes ConfigMap，以建立新的佇列及 SVRCONN 通道，以及新增容許存取通道的通道鑑別記錄。

   請確定您位於先前建立的名稱空間中 (請參閱 開始之前)，然後在 OCP Web 主控台或使用指令行輸入下列 YAML。

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

   MQSC 會定義一個稱為 *MTLS.SVRCONN* 的通道，以及一個稱為 *EXAMPLE.QUEUE*。 通道配置為只容許存取提供「通用名稱」為 *example-app1* 之憑證的用戶端。 這是在步驟 第 126 頁的『1』 中建立的其中一個憑證中使用的通用名稱。 此通道上具有此通用名稱的連線會對映至 *app1* 的使用者 ID，該使用者 ID 獲授權連接至佇列管理程式，以及存取範例佇列。 INI 檔案會啟用安全原則，這表示 *app1* 使用者 ID 不需要存在於外部使用者登錄中-它只在此配置中作為名稱存在。
3. 部署佇列管理程式

   使用下列自訂資源 YAML 建立新的佇列管理程式。 請確定您在開始此作業之前所建立的名稱空間中，然後在 OCP Web 主控台中輸入下列 YAML，或使用指令行。 請檢查是否指定正確的授權，並將 `false` 變更為 `true` 以接受授權。

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: exampleqm
spec:
  license:
    accept: false
    license: L-AMRD-XH6P3Q
    use: Production
  queueManager:
    name: EXAMPLEQM
    mqsc:
    - configMap:
        name: example-tls-configmap
        items:
        - example-tls.mqsc
    ini:
    - configMap:
        name: example-tls-configmap
        items:
        - example-tls.ini
    storage:
      queueManager:
        type: ephemeral
```

```
  version: 9.3.5.1-r2
  pki:
    keys:
      - name: default
        secret:
          secretName: example-qm-tls
          items:
            - tls.key
            - tls.crt
            - ca.crt
```

請注意，已在步驟 第 126 頁的『1』中建立密鑰 *example-qm-tls* ，並在步驟 第 126 頁的『2』中建立 ConfigMap *example-tls-configmap*

4. 確認佇列管理程式正在執行中

   現在正在部署佇列管理程式。 請先確認它處於 Running 狀態，然後再繼續。 例如：

   ```
   oc get qmgr exampleqm
   ```

5. 測試佇列管理程式的連線

   若要確認佇列管理程式已配置相互 TLS 通訊，請遵循 第 127 頁的『從筆記型電腦測試與佇列管理程式的相互 TLS 連線』中的步驟。

結果

恭喜您，您已順利部署已啟用 TLS 的佇列管理程式，並使用 TLS 憑證中提供的詳細資料向佇列管理程式進行鑑別並提供身分。

OpenShift CP4I Linux 從筆記型電腦測試與佇列管理程式的相互 *TLS* 連線

在使用「IBM MQ Operator」建立佇列管理程式之後，您可以透過連接至佇列管理程式並放置及取得訊息，來測試佇列管理程式是否在運作中。 這項作業會引導您完成如何使用 IBM MQ 範例程式來連接，方法是在 Kubernetes 叢集以外的機器上執行它們，例如您的筆記型電腦。

開始之前

若要完成此範例，您必須先完成下列必要條件：

- 安裝「IBM MQ client」。 您需要 **amqsputc** 和 **amqsgetc** 指令，這些指令可以安裝成 IBM MQ client 的一部分，如下所示：
  - Windows Linux Windows 和 Linux：從 https://ibm.biz/mq93redistclients 安裝適用於您作業系統的 IBM MQ 可重新配送用戶端。
  - macOS Mac：下載並設定 IBM MQ MacOS Toolkit：https://developer.ibm.com/tutorials/mq-macos-dev/
- 請確定您已將必要的金鑰和憑證檔案下載至機器上的目錄，且您知道金鑰儲存庫密碼。 例如，這些檔案建立在 第 124 頁的『使用 OpenSSL 建立自簽 PKI』中：
  - `example-app1.p12`
  - `example-app1-chain.crt` (僅當您使用 arm64 Apple Mac 時)
- 將配置為使用 TLS 的佇列管理程式部署至 OCP 叢集，例如，遵循 第 125 頁的『範例: 配置具有相互 TLS 鑑別的佇列管理程式』中的步驟

關於這項作業

此範例使用在 Kubernetes 叢集外部機器上執行的 IBM MQ 範例程式 (例如筆記型電腦) ，以連接至配置有 TLS 的 QueueManager ，並放置及取得訊息。

程序

1. 確認佇列管理程式正在執行中

現在正在部署佇列管理程式。 請先確認它處於 Running 狀態，然後再繼續。 例如：

```
oc get qmgr exampleqm
```

2. 尋找佇列管理程式主機名稱

   使用下列指令，使用自動建立的路徑，從 OCP 叢集外部尋找佇列管理程式的佇列管理程式完整主機名稱：`exampleqm-ibm-mq-qm`：

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

3. 建立 IBM MQ 用戶端通道定義表 (CCDT)

   建立一個稱為 `ccdt.json` 的檔案，其內容如下：

   ```
   {
       "channel":
       [
           {
               "name": "MTLS.SVRCONN",
               "clientConnection":
               {
                   "connection":
                   [
                       {
                           "host": "<hostname from previous step>",
                           "port": 443
                       }
                   ],
                   "queueManager": "EXAMPLEQM"
               },
               "transmissionSecurity":
               {
                 "cipherSpecification": "ANY_TLS13",
                 "certificateLabel": "example-app1"
               },
               "type": "clientConnection"
           }
      ]
   }
   ```

   連線使用埠 443 ，因為這是 Red Hat OpenShift Container Platform 路由器接聽的埠。 資料流量將轉遞至埠 1414 上的佇列管理程式。

   如果您使用不同的通道名稱，則也需要調整該名稱。 相互 TLS 範例使用名為 *MTLS.SVRCONN* 的通道

   如需詳細資料，請參閱 配置 JSON 格式 CCDT

4. 建立用戶端 INI 檔以配置連線詳細資料

   在現行目錄中建立稱為 `mqclient.ini` 的檔案。 此檔案將由 **amqsputc** 和 **amqsgetc** 讀取。

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

   請務必將 *SSLKeyRepository* 密碼 更新為您在建立 PKCS#12 檔案時選擇的密碼。 還有其他方法可以設定金鑰儲存庫密碼，包括使用加密密碼。 如需相關資訊，請參閱 在 AIX, Linux, and Windows 上提供 IBM MQ MQI client 的金鑰儲存庫密碼

   請注意， Red Hat OpenShift Container Platform Router 會使用 SNI 將要求遞送至 IBM MQ 佇列管理程式。 *OutboundSNI= HOSTNAME* 屬性可確保 IBM MQ 用戶端包含必要的資訊，以供路由器使用 IBM MQ Operator 所配置的預設路徑。 如需相關資訊，請參閱 第 140 頁的『配置路徑以從 Red Hat OpenShift 叢集外部連接至佇列管理程式』。

5. 如果您使用 arm64 Apple Mac，則需要配置其他環境變數。

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

   此檔案包含完整憑證鏈，包括應用程式及 CA 憑證。

6. 將訊息放入佇列
   請執行下列指令:

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

   如果佇列管理程式連線成功，則會輸出下列回應:

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

   透過輸入一些文字，然後每次按 **Enter** 鍵，將數個訊息放入佇列。

   若要完成，請按 **Enter** 鍵兩次。
7. 從佇列擷取訊息
   請執行下列指令:

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

   您在前一個步驟中新增的訊息已耗用且為輸出。 在幾秒鐘之後，指令結束。

結果

恭喜您，您已順利測試已啟用 TLS 的佇列管理程式連線，並顯示您可以從用戶端安全地放置訊息並取得訊息至佇列管理程式。

OpenShift CP4I 範例: 自訂授權服務註釋

IBM MQ Operator 會自動將 IBM License Service 註釋新增至已部署的資源。 這些由 IBM License Service 監視，並產生對應於所需授權的報告。

關於這項作業

IBM MQ Operator 所新增的註釋是標準狀況中所預期的註釋，並以部署佇列管理程式期間所選取的授權值為基礎。

範例

如果 **License** 設為 `L-RJON-BZFQU2 (IBM Cloud Pak for Integration 2021.2.1)`，且 **Use** 設為 `NonProduction`，則會套用下列註釋:

- cloudpakId: c8b82d189e7545f0892db9ef2731b90d
- cloudpakName: IBM Cloud Pak for Integration
- productCharged 儲存器 :qmgr
- productCloudpak 比例: '4: 1'
- productID: 21dfe9a0f00f444f888756d835334909
- productName: IBM MQ Advanced for Non-Production
- productMetric:VIRTUAL_PROCESSOR_CORE
- productVersion: 9.2.3.0

在 IBM Cloud Pak for Integration 內， IBM App Connect Enterprise 的部署包括 IBM MQ 的受限授權。 在這些狀況下，需要置換這些註釋，以確保 IBM License Service 擷取正確的用法。 若要執行此動作，請使用 第 151 頁的『將自訂註釋和標籤新增至佇列管理程式資源』中說明的方法。

例如，如果在 IBM App Connect Enterprise 授權下部署 IBM MQ ，請使用下列程式碼片段中顯示的方法:

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

```
  annotations:
    productMetric: FREE
```

授權註釋可能需要修改的其他兩個常見原因:

1. IBM MQ Advanced 包含在另一個 IBM 產品的授權中。
   - 在此狀況下，請使用先前針對 IBM App Connect Enterprise 所說明的方法。
2. IBM MQ 是在 IBM Cloud Pak for Integration 授權下部署。
   - 如果您具有 IBM Cloud Pak for Integration 授權，則可以決定以 IBM MQ 或 IBM MQ Advanced 比例來部署佇列管理程式。 如果您在 IBM MQ 比例下部署，則必須確保您未使用任何進階功能，例如原生 HA 或 Advanced Message Security。
   - 在此狀況下，請使用下列註釋供正式作業使用:

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

   - 將下列註釋用於非正式作業用途:

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

OpenShift MQ Adv. 使用 IBM MQ Operator 來配置佇列管理程式的高可用性

關於這項作業

程序

- 第 130 頁的『原生 HA』.
- 第 133 頁的『範例: 使用 IBM MQ Operator 來配置原生 HA』.
- 第 138 頁的『使用 IBM MQ Operator 來配置多重實例佇列管理程式』.

OpenShift MQ Adv. 原生 *HA*

原生 HA 是適用於 IBM MQ 的原生 (內建) 高可用性解決方案，適合與雲端區塊儲存體搭配使用。

「原生 HA」配置提供高可用性佇列管理程式，其中可回復的 MQ 資料 (例如，訊息) 會在多組儲存體之間抄寫，以防止因儲存體故障而流失。 佇列管理程式由多個執行中的實例組成，其中一個實例是主導者，其他實例則準備好在失敗時快速接管，以最大化對佇列管理程式及其訊息的存取權。

原生 HA 配置由三個 Kubernetes Pod 組成，每一個 Pod 都有一個佇列管理程式實例。 一個實例是作用中佇列管理程式，處理訊息並寫入其回復日誌。 每當寫入回復日誌時，作用中佇列管理程式就會將資料傳送至另外兩個實例，稱為抄本。 每一個抄本都會寫入自己的回復日誌，確認資料，然後從抄寫的回復日誌更新自己的佇列資料。 如果執行作用中佇列管理程式的 Pod 失敗，則佇列管理程式的其中一個抄本實例會接管作用中角色，並具有可操作的現行資料。

日誌類型稱為「抄寫日誌」。 抄寫的日誌本質上是線性日誌，已啟用自動日誌管理及自動媒體映像檔。 請參閱 記載類型。 您可以使用相同的技術來管理用於管理線性日誌的抄寫日誌。

Kubernetes Service 用來將 TCP/IP 用戶端連線遞送至現行作用中實例，該實例被識別為備妥可供網路資料流量使用的唯一 Pod。 這會發生，而不需要用戶端應用程式知道不同的實例。

3 個 Pod 被用來大大減少出現腦分裂情況的可能性。 在雙 Pod 高可用性中，當兩個 Pod 之間的連線功能中斷時，可能會發生核心分裂。 沒有連線功能，兩個 Pod 可以同時執行佇列管理程式，累計不同的資料。 當連線回復時，會有兩個不同版本的資料 ('spit-brain ') ，且需要人為介入來決定要保留哪些資料集，以及要捨棄哪些資料集。

原生 HA 使用具有額定的三個 Pod 系統，以避免核心分裂狀況。 至少可以與其中一個其他 Pod 通訊的 Pod 會形成仲裁。 佇列管理程式只能在具有仲裁的 Pod 上變成作用中實例。 佇列管理程式無法在未連接至至少一個其他 Pod 的 Pod 上變成作用中，因此永遠不會同時有兩個作用中實例:

- 如果單一 Pod 失敗，則其他兩個 Pod 中的其中一個 Pod 上的佇列管理程式可以接管。 如果兩個 Pod 失敗，則佇列管理程式無法變成其餘 Pod 上的作用中實例，因為該 Pod 沒有仲裁 (其餘 Pod 無法指出其他兩個 Pod 是否失敗，或它們仍在執行中且失去連線功能)。
- 如果單一 Pod 失去連線功能，則佇列管理程式無法在此 Pod 上變成作用中，因為 Pod 沒有仲裁。 其餘兩個 Pod 中的其中一個 Pod 上的佇列管理程式可以接管具有額定的佇列管理程式。 如果所有 Pod 都失去連線功能，則佇列管理程式無法在任何 Pod 上變成作用中，因為所有 Pod 都沒有仲裁。

如果作用中 Pod 失敗並隨後回復，則可以在抄本角色中重新結合群組。

為了效能和可靠性，建議 RWO (ReadWriteOnce) 持續性儲存體與原生 HA 配置搭配使用。 如果任何儲存體提供者中的 RWO 磁區符合下列條件，則支援這些磁區:

- 從區塊儲存體提供者取得。
- 格式化為 ext4 或 XFS (確保 POSIX 合規)。
- 支援動態磁區供應和 "volumeBindingMode: WaitForFirstConsumer"。

明確禁止下列提供者:

- NFS
- GlusterFS
- 其他非區塊提供者。

下圖顯示一般部署，其中有三個佇列管理程式實例部署在三個儲存器中。

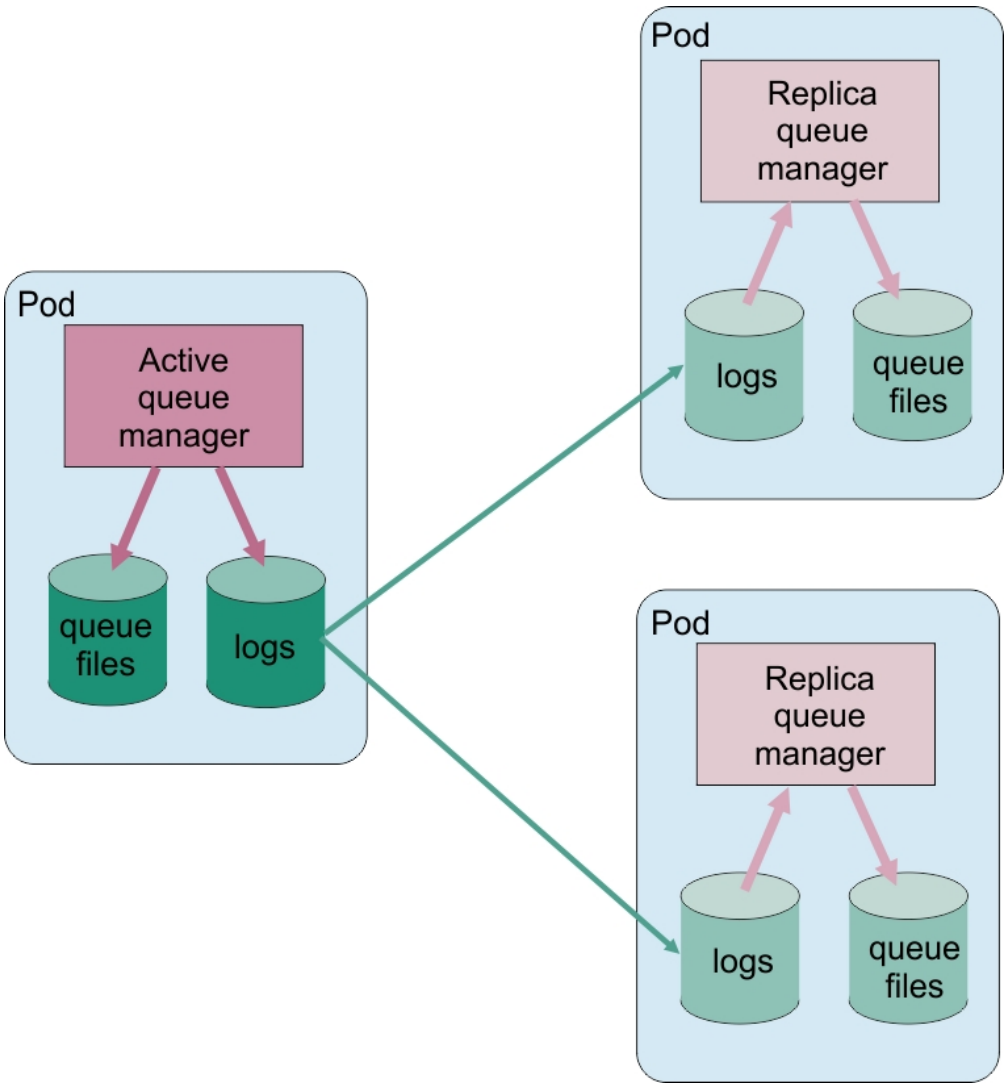


圖 *1:* 原生 *HA* 配置範例

OpenShift MQ Adv. 使用 *IBM MQ Operator* 來配置原生 *HA*

原生 HA 是使用 QueueManager API 來配置，而進階選項是使用 INI 檔案來提供。

原生 HA 是使用 QueueManager API 的 .spec.queueManager.availability 來配置，例如:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
name: nativeha-example
spec:
  license:
    accept: false
    license: L-AMRD-XH6P3Q
    use: Production
  queueManager:
    availability:
      type: NativeHA
  version: 9.3.5.1-r2
```

.spec.queueManager.availability.type 欄位必須設為 NativeHA。

在 .spec.queueManager.availability 下，您也可以配置 TLS 密鑰及密碼，以在抄寫時在佇列管理程式實例之間使用。 強烈建議這樣做，並在 第 133 頁的『範例: 使用 IBM MQ Operator 來配置原生 HA』 中提供逐步手冊。

相關工作

第 133 頁的『範例: 使用 IBM MQ Operator 來配置原生 HA』
此範例會使用 IBM MQ Operator，將使用原生高可用性特性的佇列管理程式部署至 OpenShift Container Platform 。 交互 TLS 用於鑑別，以從 TLS 憑證對映至佇列管理程式中的身分。

範例: 使用 *IBM MQ Operator* 來配置原生 *HA*

此範例會使用 IBM MQ Operator，將使用原生高可用性特性的佇列管理程式部署至 OpenShift Container Platform 。 交互 TLS 用於鑑別，以從 TLS 憑證對映至佇列管理程式中的身分。

開始之前

若要完成此範例，您必須先完成下列必要條件:

- 針對此範例建立 OpenShift Container Platform (OCP) 專案/名稱空間。
- 在指令行上，登入 OCP 叢集，並切換至上述名稱空間。
- 請確定上述名稱空間中已安裝 IBM MQ Operator 且可供使用。

關於這項作業

此範例提供自訂資源 YAML ，定義要部署至 OpenShift Container Platform 的佇列管理程式。 它也會詳細說明在啟用 TLS 的情況下部署佇列管理程式所需的其他步驟。

程序

1. 建立憑證配對，如 第 124 頁的『使用 OpenSSL 建立自簽 PKI』中所述。
2. 建立包含 MQSC 指令及 INI 檔案的配置對映

   建立包含 MQSC 指令的 Kubernetes ConfigMap ，以建立新的佇列及 SVRCONN 通道，以及新增容許存取通道的通道鑑別記錄。

   請確定您位於先前建立的名稱空間中 (請參閱 開始之前) ，然後在 OCP Web 主控台或使用指令行輸入下列 YAML。

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

   MQSC 會定義一個稱為 *MTLS.SVRCONN* 的通道，以及一個稱為 *EXAMPLE.QUEUE*。 通道配置為只容許存取提供「通用名稱」為 *example-app1* 之憑證的用戶端。 這是在步驟 第 133 頁的『1』中建立的其中一個憑證中使用的通用名稱。 此通道上具有此通用名稱的連線會對映至 *app1* 的使用者 ID ，該使用者 ID 獲授權連接至佇列管理程式，以及存取範例佇列。 INI 檔案會啟用安全原則，這表示 *app1* 使用者 ID 不需要存在於外部使用者登錄中-它只在此配置中作為名稱存在。
3. 部署佇列管理程式

使用下列自訂資源 YAML 建立新的佇列管理程式。 請確定您在開始此作業之前所建立的名稱空間中，然後在 OCP Web 主控台中輸入下列 YAML，或使用指令行。 請檢查是否指定正確的授權，並將 `false` 變更為 `true` 以接受授權。

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: exampleqm
spec:
  license:
    accept: false
    license: L-AMRD-XH6P3Q
    use: Production
  queueManager:
    name: EXAMPLEQM
    availability:
      type: NativeHA
      tls:
        secretName: example-qm-tls
    mqsc:
    - configMap:
        name: example-nativeha-configmap
        items:
        - example-tls.mqsc
    ini:
    - configMap:
        name: example-nativeha-configmap
        items:
        - example-tls.ini
    storage:
      queueManager:
        type: persistent-claim
  version: 9.3.5.1-r2
  pki:
    keys:
      - name: default
        secret:
          secretName: example-qm-tls
          items:
            - tls.key
            - tls.crt
            - ca.crt
```

請注意，已在步驟 第 133 頁的『1』中建立密鑰 *example-qm-tls*，並在步驟 第 133 頁的『2』中建立 ConfigMap *example-nativeha-configmap*

可用性類型設為 *NativeHA*，並且已選取持續性儲存體。 將使用 Kubernetes 叢集中配置的預設儲存空間類別。 如果您未將儲存類別配置為預設值，或想要使用不同的儲存類別，請在 `spec.queueManager.storage` 下新增 `defaultClass: <storage_class_name>`。

原生 HA 佇列管理程式中的三個 Pod 會透過網路抄寫資料。 依預設不會加密此鏈結，但此範例會使用佇列管理程式的憑證來加密資料流量。 您可以指定不同的憑證，以取得額外的安全。「原生 HA TLS 密鑰」必須是具有特定結構 (例如，私密金鑰必須稱為 *tls.key*) 的 Kubernetes TLS 密鑰。

4. 確認佇列管理程式正在執行中

   現在正在部署佇列管理程式。 請先確認它處於 `Running` 狀態，然後再繼續。 例如：

   ```
   oc get qmgr exampleqm
   ```

5. 測試佇列管理程式的連線

   若要確認佇列管理程式已配置且可用，請遵循 第 127 頁的『從筆記型電腦測試與佇列管理程式的相互 TLS 連線』中的步驟。

6. 強制作用中 Pod 失敗

   若要驗證佇列管理程式的自動回復，請模擬 Pod 失敗：

   a) 檢視作用中及待命 Pod

   請執行下列指令：

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

   請注意，在 **READY** 欄位中，作用中 Pod 會傳回值 `1/1`，而抄本 Pod 會傳回值 `0/1`。

b) 刪除作用中 Pod

執行下列指令，並指定作用中 Pod 的完整名稱:

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

c) 再次檢視 Pod 狀態

請執行下列指令:

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

d) 檢視佇列管理程式狀態

執行下列指令，並指定其中一個其他 Pod 的完整名稱:

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

您應該會看到狀態顯示作用中實例已變更，例如:

```
QMNAME(EXAMPLEQM) ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
```

e) 重新測試佇列管理程式的連線

若要確認佇列管理程式已回復，請遵循 第 127 頁的『從筆記型電腦測試與佇列管理程式的相互 TLS 連線』中的步驟。

結果

恭喜您，您已順利部署具有原生高可用性及交互 TLS 鑑別的佇列管理程式，並驗證它在作用中 Pod 失敗時自動回復。

檢視 *IBM MQ* 儲存器的原生 *HA* 佇列管理程式狀態

若為 IBM MQ 容器，您可以在其中一個執行中 Pod 內執行 **dspmq** 指令，以檢視「原生 HA」實例的狀態。

關於這項作業

您可以在其中一個執行中 Pod 中使用 **dspmq** 指令，以檢視佇列管理程式實例的作業狀態。 傳回的資訊視實例是作用中還是抄本而定。 作用中實例所提供的資訊是明確的，抄本節點的資訊可能已過期。

您可以執行下列動作:

- 檢視現行節點上的佇列管理程式實例是作用中還是抄本。
- 檢視現行節點上實例的原生 HA 作業狀態。
- 檢視原生 HA 配置中所有三個實例的作業狀態。

下列狀態欄位用來報告原生 HA 配置狀態:

角色
指定實例的現行角色，並且是 Active、 Replica 或 Unknown 之一。

實例
使用 **crtmqm** 指令的 **-lr** 選項建立此佇列管理程式實例時，為其提供的名稱。

INSYNC
指出實例是否可以在必要時接管作為作用中實例。

仲裁
以 *number_of_instances_in-sync*/*number_of_instances_configured* 格式報告仲裁狀態。

REPLADDR
佇列管理程式實例的抄寫位址。

CONNACTV
指出節點是否連接至作用中實例。

BACKLOG
指出實例落後的 KB 數。

CONNINST
指出指定的實例是否連接至此實例。

ALTDATE
指出前次更新此資訊的日期 (如果從未更新過，則為空白)。

ALTTIME
指出前次更新此資訊的時間 (如果從未更新過，則為空白)。

程序

- 尋找屬於佇列管理程式的 Pod。

```
oc get pod --selector app.kubernetes.io/instance=nativeha-qm
```

- 在其中一個 Pod 中執行 dspmq

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

```
oc rsh Pod
```

適用於互動式 Shell，您可以在其中直接執行 dspmq。

- 若要判定佇列管理程式實例是作為作用中實例還是抄本執行，請執行下列動作:

```
oc exec -t Pod dspmq -o status -m QMgrName
```

名為 BOB 之佇列管理程式的作用中實例會報告下列狀態:

```
QMNAME(BOB)                STATUS(Running)
```

名為 BOB 之佇列管理程式的抄本實例會報告下列狀態:

```
QMNAME(BOB)                STATUS(Replica)
```

非作用中實例會報告下列狀態:

```
QMNAME(BOB)                STATUS(Ended Immediately)
```

- 若要判定所指定 Pod 中實例的原生 HA 作業狀態，請執行下列動作:

```
oc exec -t Pod dspmq -o nativeha -m QMgrName
```

名為 BOB 之佇列管理程式的作用中實例可能報告下列狀態:

```
QMNAME(BOB)              ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
```

佇列管理程式 BOB 的抄本實例可能會報告下列狀態:

```
QMNAME(BOB)              ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
```

名為 BOB 之佇列管理程式的非作用中實例可能會報告下列狀態:

```
QMNAME(BOB)              ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
```

- 若要判定「原生 HA」配置中所有實例的「原生 HA」作業狀態，請執行下列動作:

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

如果您在執行佇列管理程式 BOB 作用中實例的節點上發出此指令，則可能會收到下列狀態:

```
QMNAME(BOB)            ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
 INSTANCE(inst1) ROLE(Active)  REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
 INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
 INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
```

如果您在執行佇列管理程式 BOB 抄本實例的節點上發出此指令，則可能會收到下列狀態，指出其中一個抄本落後:

```
QMNAME(BOB)            ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
 INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
 INSTANCE(inst1) ROLE(Active)  REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
 INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(No)  BACKLOG(435)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
```

如果您在執行佇列管理程式 BOB 非作用中實例的節點上發出此指令，則可能會收到下列狀態:

```
QMNAME(BOB)            ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
 INSTANCE(inst1) ROLE(Unknown) REPLADDR(9.20.123.45) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATE() ALTTIME()
 INSTANCE(inst2) ROLE(Unknown) REPLADDR(9.20.123.46) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATE() ALTTIME()
 INSTANCE(inst3) ROLE(Unknown) REPLADDR(9.20.123.47) CONNACTV(No) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATE() ALTTIME()
```

如果您在實例仍在協議作用中及抄本時發出指令，則會收到下列狀態:

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

相關工作

第 133 頁的『範例: 使用 IBM MQ Operator 來配置原生 HA』
此範例會使用 IBM MQ Operator，將使用原生高可用性特性的佇列管理程式部署至 OpenShift Container Platform 。 交互 TLS 用於鑑別，以從 TLS 憑證對映至佇列管理程式中的身分。

相關參考

dspmq (顯示佇列管理程式) 指令

OpenShift MQ Adv. 原生 *HA* 的進階調整

調整計時及間隔的進階設定。 除非已知預設值不符合您系統的需求，否則應該不需要使用這些設定。

用於配置原生 HA 的基本選項是使用 `QueueManager` API 來處理， IBM MQ Operator 會使用該 API 來為您配置基礎佇列管理程式 INI 檔。 在 NativeHALocal 實例段落下，有一些其他進階選項只能使用 INI 檔案來配置。 如需如何配置 INI 檔案的相關資訊，請參閱 第 122 頁的『範例: 提供 MQSC 及 INI 檔案』 。

`HeartbeatInterval`
活動訊號間隔定義原生 HA 佇列管理程式的作用中實例傳送網路活動訊號的頻率 (毫秒)。 活動訊號間隔值的有效範圍是 500 (0.5 秒) 至 60000 (1 分鐘) ，超出此範圍的值會導致佇列管理程式無法啟動。 如果省略此屬性，則會使用預設值 5000 (5 秒)。 每一個實例都必須使用相同的活動訊號間隔。

`HeartbeatTimeout`
活動訊號逾時值定義原生 HA 佇列管理程式的抄本實例在決定作用中實例無回應之前等待的時間。 活動訊號間隔逾時值的有效範圍是 500 (0.5 秒) 至 120000 (2 分鐘)。 活動訊號逾時值必須大於或等於活動訊號間隔。

無效值會導致佇列管理程式無法啟動。 如果省略此屬性，則抄本會等待 2 x `HeartbeatInterval` ，然後再啟動處理程序來選取新的作用中實例。 每一個實例都必須使用相同的活動訊號逾時。

`RetryInterval`
重試間隔定義原生 HA 佇列管理程式應該重試失敗抄寫鏈結的頻率 (毫秒)。 重試間隔的有效範圍是 500 (0.5 秒) 至 120000 (2 分鐘)。 如果省略此屬性，在重試失敗的抄寫鏈結之前，抄本會等待 2 x `HeartbeatInterval` 。

OpenShift MQ Adv. 正在結束原生 *HA* 佇列管理程式

您可以使用 **endmqm** 指令來結束屬於「原生 HA」群組的作用中或抄本佇列管理程式。

程序

- 若要結束佇列管理程式的作用中實例，請參閱本文件「配置」一節中的 結束原生 HA 佇列管理程式 。

OpenShift CP4I MQ Adv. Kubernetes **使用 *IBM MQ Operator* 來配置多重實例佇列管理程式**

此範例會使用 IBM MQ Operator，將使用的多重實例佇列管理程式部署至 OpenShift Container Platform 。 交互 TLS 用於鑑別，以從 TLS 憑證對映至佇列管理程式中的身分。

開始之前

若要完成此範例，您必須先完成下列必要條件:

- 針對此範例建立 OpenShift Container Platform (OCP) 專案/名稱空間。
- 在指令行上，登入 OCP 叢集，並切換至上述名稱空間。
- 請確定上述名稱空間中已安裝 IBM MQ Operator 且可供使用。

關於這項作業

此範例提供自訂資源 YAML ，定義要部署至 OpenShift Container Platform 的佇列管理程式。 它也會詳細說明在啟用 TLS 的情況下部署佇列管理程式所需的其他步驟。

程序

1. 決定適當的儲存類別

   可以使用多個 持續性磁區存取模式來存取 Kubernetes 叢集中的儲存體。 多重實例佇列管理程式會建立多個持續性磁區: 每一個佇列管理程式各一個，以及至少一個共用磁區。 多重實例佇列管理程式的共用磁區必須使用 `ReadWriteMany` 儲存類別。 Kubernetes 叢集中的預設儲存類別通常用於 `ReadWriteOnce` 儲存類別 (區塊儲存體)。 例如，如果您使用 Red Hat OpenShift Data Foundation，儲存類別 *ocs-storagecluster-cephfs* 會提供適合的共用檔案系統。 選擇檔案系統非常重要，因為並非所有共用檔案系統都以相同方式處理檔案鎖定。 請參閱 Multiplatforms 上的規劃檔案系統支援 及 IBM MQ 多重實例佇列管理程式檔案系統的測試陳述式。
2. 建立憑證配對，如 第 124 頁的『使用 OpenSSL 建立自簽 PKI』 中所述。
3. 建立包含 MQSC 指令及 INI 檔案的配置對映

   建立包含 MQSC 指令的 Kubernetes ConfigMap ，以建立新的佇列及 SVRCONN 通道，以及新增容許存取通道的通道鑑別記錄。

   請確定您位於先前建立的名稱空間中 (請參閱 開始之前) ，然後在 OCP Web 主控台或使用指令行輸入下列 YAML。

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

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

MQSC 會定義一個稱為 *MTLS.SVRCONN* 的通道，以及一個稱為 *EXAMPLE.QUEUE*。 通道配置為只容許存取提供「通用名稱」為 *example-app1* 之憑證的用戶端。 這是在步驟 第 138 頁的『2』中建立的其中一個憑證中使用的通用名稱。 此通道上具有此通用名稱的連線會對映至 *app1* 的使用者 ID，該使用者 ID 獲授權連接至佇列管理程式，以及存取範例佇列。 INI 檔案會啟用安全原則，這表示 *app1* 使用者 ID 不需要存在於外部使用者登錄中-它只在此配置中作為名稱存在。

4. 部署佇列管理程式

   使用下列自訂資源 YAML 建立新的佇列管理程式。 請確定您在開始此作業之前所建立的名稱空間中，然後在 OCP Web 主控台中輸入下列 YAML，或使用指令行。 請檢查是否指定正確的授權，並將 `false` 變更為 `true` 以接受授權。

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: exampleqm
spec:
  license:
    accept: false
    license: L-AMRD-XH6P3Q
    use: Production
  queueManager:
    name: EXAMPLEQM
    availability:
      type: MultiInstance
    mqsc:
    - configMap:
        name: example-miqm-configmap
        items:
        - example-tls.mqsc
    ini:
    - configMap:
        name: example-miqm-configmap
        items:
        - example-tls.ini
    storage:
      defaultClass: <STORAGE CLASS>
  version: 9.3.5.1-r2
  pki:
    keys:
      - name: default
        secret:
          secretName: example-qm-tls
          items:
            - tls.key
            - tls.crt
            - ca.crt
```

   Change <STORAGE CLASS> to the storage class you identified in Step 第 138 頁的『1』.

   請注意，已在步驟 第 138 頁的『2』中建立密鑰 *example-qm-tls*，並在步驟 第 138 頁的『3』 中建立 ConfigMap *example-miqm-configmap*

   可用性類型設為 *MultiInstance*，這會自動選取持續性儲存體。

5. 確認佇列管理程式正在執行中

   現在正在部署佇列管理程式。 請先確認它處於 `Running` 狀態，然後再繼續。 例如：

```
oc get qmgr exampleqm
```

6. 測試佇列管理程式的連線

   若要確認佇列管理程式已配置且可用，請遵循 第 127 頁的『從筆記型電腦測試與佇列管理程式的相互 TLS 連線』中的步驟。

7. 強制作用中 Pod 失敗

   若要驗證佇列管理程式的自動回復，請模擬 Pod 失敗：

   a) 檢視作用中及待命 Pod

      請執行下列指令：

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

請注意，在 **READY** 欄位中，作用中 Pod 會傳回值 1/1，而待命 Pod 會傳回值 0/1。

b) 刪除作用中 Pod

執行下列指令，並指定作用中 Pod 的完整名稱:

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

c) 再次檢視 Pod 狀態

請執行下列指令:

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

d) 檢視佇列管理程式狀態

執行下列指令，並指定其他 Pod 的完整名稱:

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

您應該會看到狀態顯示作用中實例已變更，例如:

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

e) 重新測試佇列管理程式的連線

若要確認佇列管理程式已回復，請遵循 第 127 頁的『從筆記型電腦測試與佇列管理程式的相互 TLS 連線』中的步驟。

結果

恭喜您，您已順利部署具有交互 TLS 鑑別的多重實例佇列管理程式，並驗證它在作用中 Pod 失敗時自動回復。

OpenShift CP4I V9.3.0 CD 配置路徑以從 Red Hat OpenShift 叢集外部連接至佇列管理程式

您需要 Red Hat 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，因為只有在使用 TLS 1.2 或更高版本的通訊協定時，才能在 TLS 通訊協定中使用 SNI。 Red Hat OpenShift Container Platform Router 使用 SNI 將要求遞送至 IBM MQ 佇列管理程式。

關於這項作業

Red Hat OpenShift 路徑 的必要配置取決於用戶端應用程式的 伺服器名稱指示 (SNI) 行為。 IBM MQ 支援兩種不同的 SNI 標頭設定，視配置和用戶端類型而定。 「SNI 標頭」設為用戶端目的地的主機名稱，或設為 IBM MQ 通道名稱。 如需 IBM MQ 如何將通道名稱對映至主機名稱的相關資訊，請參閱 IBM MQ 如何提供多個憑證功能。

SNI 標頭是設為 IBM MQ 通道名稱，還是使用 **OutboundSNI** 屬性來控制主機名稱。 可能的值為 OutboundSNI=CHANNEL (預設值) 或 OutboundSNI=HOSTNAME。 如需相關資訊，請參閱 用戶端配置檔的 SSL 段落。 請注意， CHANNEL 和 HOSTNAME 是您使用的確切值; 它們不是您取代為實際通道名稱或主機名稱的變數名稱。

具有不同 OutboundSNI 設定的用戶端行為

如果 **OutboundSNI** 設為 HOSTNAME，則只要在連線名稱中提供主機名稱，下列用戶端就會設定主機名稱 SNI:

- C 用戶端
- 未受管理模式中的 .NET 用戶端

- Java/JMS 用戶端

如果 **OutboundSNI** 設為 HOSTNAME，且在連線名稱中使用 IP 位址，則下列用戶端會傳送空白 SNI 標頭:

- C 用戶端
- 未受管理模式中的 .NET 用戶端
- Java/JMS 用戶端 (無法對主機名稱執行反向 DNS 查閱)

如果 **OutboundSNI** 設為 CHANNEL，則不論使用的是主機名稱還是 IP 位址連線名稱，都會改用 IBM MQ 通道名稱並一律傳送。

下列用戶端類型不支援將 SNI 標頭設為 IBM MQ 通道名稱，因此不論 **OutboundSNI** 設定為何，一律嘗試將 SNI 標頭設為主機名稱:

- AMQP 用戶端
- XR 用戶端
- 受管理模式中的 .NET 用戶端 (IBM MQ 9.3.0 之前)

從 IBM MQ 9.3.0 開始，如果 **OutboundSNI** 內容設為 HOSTNAME，其容許 IBM MQ 受管理 .NET 用戶端使用 Red Hat OpenShift 路徑連接至佇列管理程式，則已更新 IBM MQ 受管理 .NET 用戶端，將 SERVERNAME 設為個別的主機名稱。

如果用戶端應用程式透過 IBM MQ Internet Pass-Thru (MQIPT) 連接至部署在 Red Hat OpenShift 叢集中的佇列管理程式，則可以使用路徑定義中的 SSLClientOutboundSNI 內容，將 MQIPT 配置為將 SNI 設為主機名稱。

OutboundSNI、多個憑證及 Red Hat OpenShift 路徑

IBM MQ 使用 SNI 標頭來提供多個憑證功能。 如果應用程式連接至透過 CERTLABL 欄位配置為使用不同憑證的 IBM MQ 通道，則應用程式必須使用 CHANNEL 的 **OutboundSNI** 設定進行連接。

如果 Red Hat OpenShift Route 配置需要 HOSTNAME SNI，則您無法使用 IBM MQ 的多個憑證功能，也無法在任何 IBM MQ 通道物件上設定 CERTLABL 設定。

如果 **OutboundSNI** 設定不是 CHANNEL 的應用程式連接至已配置憑證標籤的通道，則會拒絕該應用程式，並在佇列管理程式錯誤日誌中列印 AMQ9673 訊息。

如需 IBM MQ 如何提供多個憑證功能的相關資訊，請參閱 IBM MQ 如何提供多個憑證功能 。

範例

將 SNI 設為 MQ 通道的用戶端應用程式需要針對您要連接的每一個通道建立新的 Red Hat OpenShift 路徑。您也必須在 Red Hat OpenShift Container Platform 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。

MQ 通道名稱不能以小寫字母結尾很重要，因為 IBM MQ 會將通道名稱對映至 SNI 標頭。

若要判定每一個新的 Red Hat 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。如需相關資訊，請參閱 IBM MQ 如何提供多個憑證功能 。

然後，您必須在叢集裡套用下列 yaml，為每一個通道建立新的 Red Hat OpenShift 路徑:

```
  apiVersion: route.openshift.io/v1
  kind: Route
  metadata:
    name: <provide a unique name for the Route>
    namespace: <the namespace of your MQ deployment>
  spec:
    host: <SNI address mapping for the channel>
    to:
      kind: Service
      name: <the name of the Kubernetes Service for your MQ deployment (for example "<Queue
Manager Name>-ibm-mq")>
    port:
      targetPort: 1414
    tls:
      termination: passthrough
```

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 "<Queue Manager Name>-ibm-mq-qm")>
-n <namespace of your MQ deployment> -o jsonpath="{.spec.host}"
```

用戶端連線的埠應該設為「Red Hat OpenShift Container Platform 路由器」所使用的埠-通常是 443。

相關工作

第 155 頁的『連接至 Red Hat OpenShift 叢集中部署的 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 。

CP4I Deprecated Removed 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。

開始之前

小心:

V 9.3.0 Deprecated V 9.3.0 從 IBM MQ Operator 2.0.0 開始，「作業儀表板」已淘汰，將不會收到進一步更新。 不應建立「作業儀表板」的新用途。

V 9.3.3 Removed 從 IBM MQ Operator 2.4.0 ，會移除「作業儀表板」。 請注意，如果在支援該佇列管理程式儲存器映像檔的 IBM MQ Operator 上，「作業儀表板」仍可用於 9.3.3.0-r1 之前的現有佇列管理程式。 如需 IBM MQ Operator 的版本支援，請參閱 第 10 頁的『可用的 IBM MQ 版本』。

「作業儀表板」的支援將於 2024 年 6 月 30 日結束。 如需相關資訊，請參閱 軟體撤銷及/或支援停止。

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 UI (previously the 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 使用 IBM Cloud Pak for Integration Platform UI 部署佇列管理程式

   如果您使用 Red Hat OpenShift CLI 或 Red Hat 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
   ```

   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。

OpenShift CP4I Operator2.2.0 整合 IBM MQ 與 IBM Instana 追蹤

IBM Instana 可用來追蹤 IBM Cloud Pak for Integration 內的交易。

開始之前

本文件涵蓋 IBM Instana 追蹤，這是透過系統來追蹤訊息的程序。 它不涵蓋 IBM Instana 監視，在其中會擷取 IBM MQ 佇列管理程式狀態的詳細資料。 如需 IBM Instana 監視 IBM MQ 的相關資訊，請參閱 監視 IBM MQ。 如需已鑑別監視的詳細指示，請參閱 第 144 頁的『配置使用 TLS 進行鑑別的 IBM Instana 監視』。

註:

- 此特性只能與 IBM MQ 操作器 2.2.0 版以及更新版本搭配使用。 此特性僅在 IBM MQ 9.3.1.0-r2 版或更新版本的運算元上受支援。
- 您可以在舊版「IBM MQ 操作員」及佇列管理程式版本上執行 IBM Instana 追蹤，但不能原生執行。 請參閱 IBM Instana 說明文件中的 配置 IBM MQ 追蹤 。

您必須先部署 IBM Instana 後端及 IBM Instana 代理程式，然後才能使用「IBM MQ 操作員」執行 IBM Instana 追蹤。 依預設， IBM MQ 佇列管理程式會與部署在與佇列管理程式 Pod 相同節點上的 IBM Instana 代理程式進行通訊。

關於這項作業

啟用與 IBM Instana 的整合會導致 IBM MQ API 結束程式安裝在佇列管理程式中。 API 結束程式會針對流經佇列管理程式的訊息，將追蹤資料傳送至 IBM Instana 代理程式。

API 結束程式會將 RFH2 標頭新增至每一則訊息。 這些標頭包含追蹤資訊。

IBM Instana 代理程式負責將追蹤資料傳送至 IBM Instana 後端。

如需部署 IBM Instana 後端及 IBM Instana 代理程式的相關資訊，請參閱 IBM Instana 說明文件中的 在 CP4I 平台使用者介面中啟用 IBM Instana 監視 。

程序

標準部署

- 在啟用 IBM Instana 追蹤的情況下部署佇列管理程式。

  依預設，會停用 IBM Instana 追蹤。

  如果您使用 IBM Cloud Pak for Integration Platform UI (previously the Platform Navigator) 或 OpenShift Web 主控台:

  1. 按一下 **遙測** > **追蹤** > **Instana**。
  2. 將 **啟用 Instana 追蹤** 切換參數設為 `true`。

  如果您透過 YAML 進行部署，請使用下列 Snippet:

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

進階部署

- 透過 https 與 IBM Instana 代理程式通訊。

依預設， IBM MQ 的 IBM Instana 結束程式會透過 http 與 IBM Instana 代理程式進行通訊。 代理程式的主機位址設為執行佇列管理程式之節點的 IP 位址。 這符合 IBM Instana 說明文件中 啟用 IBM Instana 監視 所說明的配置，其中 IBM Instana 代理程式由 IBM Instana Agent Operator 部署為 daemonset。

目前 IBM MQ 的 IBM Instana 結束程式與 IBM Instana 代理程式之間的通訊支援 http 或 https 通訊協定。 若要使用 https， IBM Instana 代理程式必須先配置為使用 TLS 加密。 請參閱 IBM Instana 說明文件中的 設定代理程式端點的 TLS 加密 。 然後，可以將通訊協定設為 :https ， 如下所示: https:

如果您使用 OpenShift Web 主控台:

1. 按一下 **遙測** > **Instana**。
2. 展開 **進階配置** 下拉清單。
3. 將 **Instana 代理程式通訊協定** 設為 。

如果您透過 YAML 進行部署，請使用下列 Snippet:

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

- 設定 **agentHost**

  如果 IBM Instana 代理程式尚未部署為執行佇列管理程式之 Openshift 叢集上的常駐程式，則您必須將 **agentHost** 值設為 IBM Instana 代理程式執行所在的主機名稱或 IP 位址。 **agentHost** 值不應包含通訊協定或埠。

  如果您使用 OpenShift Web 主控台:

  1. 按一下 **遙測** > **Instana**。
  2. 展開 **進階配置** 下拉清單。
  3. 在 **Instana 代理程式主機** 文字框中輸入主機名稱。

  如果您透過 YAML 進行部署，請使用下列 Snippet:

```
spec:
  telemetry:
    instana:
      enabled: true
      agentHost: 9.9.9.9
```

下一步

另請參閱第 106 頁的『將佇列管理程式部署至 Red Hat OpenShift Container Platform 叢集』。

OpenShift CP4I Operator2.2.0 配置使用 TLS 進行鑑別的 IBM Instana 監視

若要能夠透過 IBM Instana 代理程式監視佇列管理程式，您必須同時配置代理程式及佇列管理程式。

開始之前

IBM Instana 文件中的 「監視 IBM MQ」 的「配置」區段 提供關於 IBM Instana 監視配置的一般資訊。 不過，它不包含配置佇列管理程式的詳細資料。

您必須先部署 IBM Instana 後端及 IBM Instana 代理程式，然後才能使用「 IBM MQ 操作員」執行 IBM Instana 追蹤。 若要這麼做，請參閱 IBM Instana 說明文件中的 在 CP4I 平台使用者介面中啟用 IBM Instana 監視 。

程序

1. 產生憑證。
2. 配置 IBM Instana 代理程式。

3. 配置佇列管理程式。
4. 驗證及除錯。

相關工作

第 143 頁的『整合 IBM MQ 與 IBM Instana 追蹤』
IBM Instana 可用來追蹤 IBM Cloud Pak for Integration 內的交易。

OpenShift CP4I Operator2.2.0 **產生 *IBM Instana* 代理程式及佇列管理程式的憑證及金鑰**

對於 IBM Instana 代理程式與佇列管理程式之間的 TLS 通訊，兩者都必須具有憑證及對應的私密金鑰。

開始之前

這是四項作業中的第一項，用來 配置使用 TLS 進行鑑別的 IBM Instana 監視。

註: 在產生這些憑證時所使用的值是為了示範。 在正式作業環境中部署時，請確保憑證的主旨和期限是適當的。

程序

IBM MQ 佇列管理程式

若要透過 TLS 與 IBM Instana 代理程式進行通訊，佇列管理程式必須具有憑證及對應的私密金鑰。 如果您已有這些，請跳過此區段。

1. 產生佇列管理程式的憑證和私密金鑰。

   請執行下列指令:

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

IBM Instana 代理程式

若要讓代理程式執行與 IBM MQ 佇列管理程式的 TLS 通訊，代理程式必須具有憑證及對應的私密金鑰。 如果您在想要使用的 JKS 金鑰儲存庫中已有私密金鑰和憑證，請跳過本節。

2. 產生 IBM Instana 代理程式的憑證及私密金鑰。

   請執行下列指令:

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

3. 將憑證和私密金鑰儲存在 PKCS12 金鑰儲存庫中。

   執行下列指令，將 *your_password* 取代為您要用來保護金鑰儲存庫安全的密碼。 在所有後續步驟中執行此取代。

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

4. 將 PKCS12 金鑰儲存庫轉換成 JKS 金鑰儲存庫。

   請執行下列指令:

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

5. 標示憑證。

   請執行下列指令：

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

6. 將佇列管理程式憑證匯入至金鑰儲存庫。

   請執行下列指令：

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

下一步

您現在已準備好 配置代理程式以進行 IBM Instana 監視。

OpenShift CP4I Operator2.2.0

Instana 監視: 配置代理程式

將金鑰儲存庫裝載至 IBM Instana 代理程式，然後配置特定佇列管理程式的監視。

開始之前

此作業假設您已 產生 IBM Instana 代理程式及佇列管理程式的憑證及金鑰。

程序

將金鑰儲存庫裝載至 IBM Instana 代理程式

1. 從 IBM Instana 代理程式名稱空間中的 JKS 金鑰儲存庫建立密鑰。

   執行下列指令，將 *keystore_secret_name* 取代為您要使用的名稱。 在所有後續步驟中執行此取代。

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

2. 在 instana-agent 名稱空間中，使用 `oc edit daemonset instana-agent` 指令來編輯 instana-agent daemonset ，以包括下列其他 volumeMount 及磁區：

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

配置監視特定佇列管理程式

3. 在 instana-agent 名稱空間中，使用 `oc edit configmap instana-agent` 指令來編輯 instana-agent configmap。
4. 在 `configuration.yaml: |` 下新增下列區段。 如果您已定義此區段，則只要將新的佇列管理程式新增至清單即可。

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

   其中

   - *your_password* 是 JKS 金鑰儲存庫的密碼

- *QUEUE_MANAGER_NAME* 是要部署的基礎 IBM MQ 佇列管理程式名稱，而不是佇列管理程式運算元的名稱。

註: 如果 *QUEUE_MANAGER_NAME* 未設為基礎佇列管理程式名稱，而是設為 Operand，則監視將無法運作。 基礎名稱定義在「佇列管理程式作業」的 `spec.queuemanager.name` 中。

5. 刪除 instana-agent 名稱空間中的 instana-agent Pod。 這會導致它們重新啟動，並以新設定開始監視。

下一步

現在，您已準備好 配置佇列管理程式以進行 IBM Instana 監視。

OpenShift CP4I Operator2.2.0 *Instana* 監視: 配置佇列管理程式

設定使用 TLS 與 IBM Instana 代理程式進行通訊的佇列管理程式。此連線的鑑別是使用 SSLPEERMAP 來完成。

開始之前

此作業假設您已 配置代理程式以進行 IBM Instana 監視。

程序

1. 透過 MQSC 及 INI 來配置佇列管理程式。

   MQSC 用於設定啟用 TLS 的新通道，然後配置該通道以鑑別連接的 IBM Instana 代理程式 (如果它具有具有必要欄位的憑證)。 在此情況下，我們會將具有包含 `CN=instana-agent,OU=app team1` 欄位的憑證的任何連接用戶端對映至使用者 `app1`。 然後， MQSC 會授與使用者 `app1` 執行 IBM Instana 監視所需作業的許可權。

   INI 檔案用來授與許可權給我們的外部使用者 `app1`。

   下列 configmap 包含必要的 MQSC 及 INI 設定。 將它部署至佇列管理程式名稱空間。

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

   其中 *your-queue-manager-namespace* 是將在其中部署佇列管理程式的名稱空間。

   註: 如果您正在監視使用者定義的佇列，則必須將其他行新增至 configmap MQSC，並授與 DSP、CHG 及 GET 許可權給那些佇列。 例如:

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

此範例使用 MQSC 及 INI 資料的 configmap，但如果您所做的任何新增都是機密，則可以使用密鑰。 如需使用 MQSC 及 INI 進行部署的一般資訊，請參閱 第 122 頁的『範例: 提供 MQSC 及 INI 檔案』。

2. 若要建立 TLS 連線，佇列管理程式必須信任 IBM Instana 代理程式的憑證。 若要達到此目的，請建立僅包含 IBM Instana 代理程式憑證的密鑰:

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

3. 佇列管理程式必須提供自己的憑證來進行 TLS 信號交換，且需要存取相關聯的私密金鑰。 部署密鑰，其中包含您先前建立或已擁有的金鑰和憑證:

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

建立 configmap 和密碼之後，您已準備好建立佇列管理程式本身。

4. 確保佇列管理程式 YAML 未在佇列管理程式儲存器中設定環境變數 **MQSNOAUT** 。

否則，在啟用之後，鑑別機制將無法運作。 在部署之後移除變數不會導致重新啟用機制，且必須重建佇列管理程式。

5. 將下列區段新增至佇列管理程式定義，其中 *MYQM* 是佇列管理程式的名稱:

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

規格的已標示區段說明如下:

a. 請確定您已為基礎佇列管理程式提供唯一名稱。 如果基礎佇列管理程式沒有唯一名稱，則監視可能無法如預期般運作。 此名稱必須符合先前編輯的 IBM Instana 代理程式 configmap 中的名稱。

b. 寫入 configmap 的 INI 資訊會新增至佇列管理程式。

c. 寫入 configmap 的 MQSC 資訊會新增至佇列管理程式。

d. 佇列管理程式憑證和私密金鑰會新增至佇列管理程式金鑰儲存庫。

e. IBM Instana 代理程式憑證會新增至佇列管理程式信任儲存庫。

6. 選擇性的: 在受監視佇列管理程式上啟用 IBM Instana 追蹤。

如果您想要這麼做，請參閱 第 143 頁的『整合 IBM MQ 與 IBM Instana 追蹤』。

7. 部署佇列管理程式。

下一步

現在，您已準備好 驗證及除錯 IBM Instana 監視。

OpenShift CP4I Operator2.2.0 *Instana* 監視: 驗證及除錯

若要能夠透過 IBM Instana 代理程式監視佇列管理程式，您必須同時配置代理程式及佇列管理程式。

開始之前

此作業假設您已 配置佇列管理程式以進行 IBM Instana 監視。

程序

正在驗證

1. 若要驗證您已在部署中順利完成，請在 IBM Instana 儀表板中檢視佇列管理程式。

   在應用程式頁面的服務區段中，以及在「基礎架構」視圖中，應該可以看到佇列管理程式。

除錯

註: 這些除錯步驟假設以 daemonset 形式執行 IBM Instana 代理程式的 Openshift 部署。

如果您在「 IBM Instana 」儀表板中看不到佇列管理程式，則可能是您的佇列管理程式配置錯誤。 請使用下列步驟來調查。

2. 識別作用中佇列管理程式 Pod 執行所在的節點。

   在佇列管理程式名稱空間中執行下列指令:

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

3. 若要判定哪個 IBM Instana 代理程式 Pod 正在與佇列管理程式相同的節點上執行，請在 instana-agent 名稱空間中執行相同的指令:

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

4. 若要協助瞭解 IBM Instana 代理程式端的任何問題，請取得 IBM Instana 代理程式 Pod 的日誌，並尋找與 'mq' 或佇列管理程式名稱相關的項目。

   請執行下列指令:

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

5. 請檢查佇列管理程式日誌。

   如果代理程式已嘗試連接至佇列管理程式，則佇列管理程式日誌應該會指出連線失敗的原因。 請執行下列指令:

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

結果

您已完成所有四項作業，以 使用 TLS 來配置已鑑別的 IBM Instana 監視。

OpenShift CP4I 使用 Red Hat OpenShift CLI 以自訂 MQSC 及 INI 檔案建置映像檔

使用「 Red Hat OpenShift Container Platform 管線」來建立新的 IBM MQ 儲存器映像檔，並將您想要套用至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 Red Hat OpenShift 密碼，請遵循 建立授權金鑰密碼的步驟。

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:9.3.5.1-r2
         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:9.3.5.1-r2"
         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 (如需詳細資料，請參閱 第 30 頁的『IBM MQ Operator 的發行歷程』)。 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 106 頁的『將佇列管理程式部署至 Red Hat OpenShift Container Platform 叢集』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 Red Hat 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 106 頁的『將佇列管理程式部署至 Red Hat OpenShift Container Platform 叢集』
此範例會部署「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 IBM MQ 安全。 在重新啟動佇列管理程式之後，訊息不會持續保存。 您可以調整配置來變更許多佇列管理程式設定。

OpenShift CP4I 將自訂註釋和標籤新增至佇列管理程式資源

您可以將自訂註釋和標籤新增至 QueueManager meta 資料。

關於這項作業

自訂註釋和標籤會新增至 PVC 以外的所有資源。 如果自訂註釋或標籤符合現有的索引鍵，則會使用 IBM MQ Operator 所設定的值。

程序

- 新增自訂註釋。

  若要將自訂註釋新增至佇列管理程式資源 (包括 Pod)，請在 metadata 下新增註釋。 例如:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
  annotations:
    annotationKey: "value"
```

- 新增自訂標籤。

  若要將自訂標籤新增至佇列管理程式資源 (包括 Pod)，請在 metadata 下新增標籤。 例如:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
  labels:
    labelKey: "value"
```

OpenShift CP4I 停用執行時期 Webhook 檢查

執行時期 Webhook 檢查可確保儲存體類別對您的佇列管理程式而言是可行的。 您可以停用它們以增進效能，或因為它們不適用於您的環境。

關於這項作業

執行時期 Webhook 檢查是在佇列管理程式配置上完成。 它們會檢查儲存類別是否適合您選取的佇列管理程式類型。

您可以選擇停用這些檢查，以減少建立佇列管理程式所花費的時間，或因為這些檢查對您的特定環境無效。

註: 停用執行時期 Webhook 檢查之後，容許任何儲存類別值。 這可能會導致佇列管理程式岔斷。

程序

- 停用執行時期 Webhook 檢查。

  在 metadata 下新增下列註釋。 例如:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
  annotations:
    "com.ibm.cp4i/disable-webhook-runtime-checks" : "true"
```

OpenShift CP4I Operator2.1.0 停用佇列管理程式規格的預設值更新

IBM MQ Operator 會使用其預設值來更新佇列管理程式規格中任何未指定的值。 如果您想要避免對佇列管理程式規格進行任何修改，則可以停用此行為。 仍會更新佇列管理程式狀態欄位。

程序

- 停用佇列管理程式預設值更新。

  在 metadata 下新增下列註釋。 例如:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
  annotations:
    "com.ibm.mq/write-defaults-spec" : "false"
```

  註: 此特性只能與 MQ Operator 2.1.0 以及更新版本搭配使用。 從 IBM MQ Operator 2.1.0 開始，依預設會套用此註釋。

V 9.3.4 使用唯讀根檔案系統執行 IBM MQ 儲存器

從 IBM MQ Operator 3.0.0 和 IBM MQ 儲存器 9.3.4.0 開始，您可以將 IBM MQ 儲存器配置成以唯讀根檔案系統來執行。 這可防止攻擊者複製及執行儲存器中的惡意程式碼。

關於這項作業

啟用唯讀根檔案系統會使儲存器檔案不可變。 亦即，在儲存器檔案系統上，可以檢視但無法修改檔案，且無法建立任何新檔案。 只能在已裝載的檔案系統上修改或建立檔案。

當啟用唯讀根檔案系統時，會建立兩個暫時磁區 Scratch 和 Tmp ，並分別裝載在儲存器中的 /run 和 /tmp 目錄中。

- **暫存** 磁區包含用於配置佇列管理程式的檔案、金鑰儲存庫及其他檔案。
- Tmp 磁區包含診斷檔案，例如佇列管理程式 RAS 檔案。

因為這些磁區是暫時的，所以在 Pod 重新啟動時，這些磁區上的檔案會遺失。

為佇列管理程式資料建立的磁區類型取決於儲存體類型。 依預設，會裝載持續性磁區。 或者，如果儲存體類型為 ephemeral，則會裝載暫時磁區。 如果磁區中的資料大小超出指定給 **sizeLimit** 內容的值，則 Kubernetes 可以退出儲存器並建立新的儲存器。 在 IBM MQ Operator 3.0.0 之前，對佇列管理程式資料使用暫時儲存體時，未施行大小限制。

依預設，不會啟用唯讀根檔案系統。 若要啟用它，請完成下列步驟:

程序

1. 使用 spec.securityContext API 來啟用唯讀根檔案系統。

   針對您的佇列管理程式，將 第 177 頁的『.spec.securityContext』 中的 **readOnlyRootFilesystem** 內容設為 true。

IBM MQ Operator 會建立兩個暫時磁區: **暫存** 和 Tmp。

2. 選擇性的: 設定或變更佇列管理程式資料儲存體類型。

   依預設，持續性磁區要求裝載於 `/mnt/mqm`。 或者，如果 第 176 頁的『.spec.queueManager.storage.queueManager』中的 **type** 內容設為 `ephemeral` ，則會建立並裝載暫時磁區。

3. 對於每一個暫時磁區，請仔細考量資料可能成長的程度。 相應地設定 **sizeLimit** 內容的值，包括 SI 單位。

   - 對於 **暫存** 暫時磁區，請在 第 176 頁的『.spec.queueManager.storage.scratch』中設定 **sizeLimit** 內容。 預設值為 "100M"。
   - 對於 Tmp 暫時磁區，請在 第 177 頁的『.spec.queueManager.storage.tmp』中設定 **sizeLimit** 內容。 預設值為 "2Gi"。
   - 如果佇列管理程式磁區的 **type** 設為 `ephemeral`，請在 第 176 頁的『.spec.queueManager.storage.queueManager』中設定 **sizeLimit** 內容。 預設值為 "2Gi"。

OpenShift V 9.3.4 使用 IBM MQ Operator 使用基本登錄來配置 IBM MQ Console

若要登入「IBM MQ Console」，您可以提供您自己的配置給佇列管理程式。

開始之前

如果您要部署具有 IBM MQ Advanced for Developers 授權的佇列管理程式，則會內建簡式配置。 請參閱 第 22 頁的『[MQ 9.3.4 Dec 2023]說明如何為 admin 和 app 使用者指定密碼的佇列管理程式 YAML 範例』。

如果您要部署 IBM Cloud Pak for Integration 授權佇列管理程式，則可以啟用與 IBM Cloud Pak for Integration Keycloak 的整合，以使用單一登入來登入 IBM MQ Console 。 請參閱第 155 頁的『連接至 Red Hat OpenShift 叢集中部署的 IBM MQ Console』。

程序

1. **建立密碼並使用 securityUtility 進行加密**。

   `ConfigMap` 用來儲存您用來存取佇列管理程式的認證。 為了提高安全，您可以使用 securityUtility 指令來編碼這些認證。

   或者，您可以使用「密鑰」，以保護 Kubernetes 層中的認證。 不過，監視或疑難排解工具可能會不安全地公開基礎檔案。

2. 選擇性的: **登入 Red Hat OpenShift 指令行介面 (CLI)**。

   如果使用 OpenShift CLI，請使用 `oc login` 登入。

   或者，您可以使用 OpenShift 主控台。

3. **使用您的配置建立 ConfigMap** 。

   如需建立 XML 配置的說明，請參閱 IBM MQ Console 和 REST API 安全。

   下列範例會在群組 `MQWebAdminGroup` 內建立使用者。 `MQWebAdminGroup` 的成員會獲指派 `MQWebAdmin` 角色。 在此範例中:

   - 您 **必須** 將 *USERNAME* 及 *PASSWORD* 取代為您自己的值。 請注意，在範例中使用 *USERNAME* 兩次。

     您 **必須** 指定 *NAMESPACE* 作為部署 IBM MQ Operator 的位置，以及將部署或已部署佇列管理程式的位置。

   a) 使用 OpenShift 主控台或指令行來建立下列 `ConfigMap`:

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: mqwebuserconfigmap
  namespace: NAMESPACE
data:
```

```
  mqwebuser.xml: |
    <?xml version="1.0" encoding="UTF-8"?>
    <server>
      <featureManager>
        <feature>appSecurity-2.0</feature>
        <feature>basicAuthenticationMQ-1.0</feature>
      </featureManager>
      <enterpriseApplication id="com.ibm.mq.console">
        <application-bnd>
          <security-role name="MQWebAdmin">
            <group name="MQWebAdminGroup" realm="defaultRealm"/>
          </security-role>
        </application-bnd>
      </enterpriseApplication>
      <basicRegistry id="basic" realm="defaultRealm">
        <user name="USERNAME" password="PASSWORD"/>
        <group name="MQWebAdminGroup">
          <member name="USERNAME"/>
        </group>
      </basicRegistry>
      <sslDefault sslRef="mqDefaultSSLConfig"/>
    </server>
```

b) 選擇性的: 如果使用指令行，請套用 ConfigMap:

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

對於其餘步驟，請選擇下列其中一個選項:

- 使用配置來部署新的佇列管理程式，以存取 IBM MQ Console。
- 套用可讓 IBM MQ Console 存取現有佇列管理程式的配置。

4. 選擇性的: **使用配置來部署新的佇列管理程式，以存取 IBM MQ Console**。

a) 建立佇列管理程式。

透過下列其中一個選項，將鑑別和授權提供者設為 **手動** ，並提供新建立的 ConfigMap mqwebuserconfigmap :

- 選項 1: 透過佇列管理程式 YAML

  在佇列管理程式 YAML 的 web 區段下新增下列程式碼:

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

- 選項 2: 透過 OpenShift 主控台「表單」視圖:

  i) 在 OpenShift 主控台上，選取 **操作器** > **已安裝的操作器**。

  ii) 選取 IBM MQ Operator 的部署。

  iii) 選取 **佇列管理程式** ，然後按一下 **建立 QueueManager**。

  iv) 選取佇列管理程式的相關選項。

  v) 選取 **Web** ，並將 **啟用 Web 伺服器** 設為 true。

  vi) 開啟 **進階配置** 清單框。

  vii) 在 **主控台** 清單框下，將 **鑑別** 及 **授權** 的 **提供者** 設為 **手動**。

  viii) 開啟 **配置** 清單框。

  ix) 開啟 **ConfigMap** 清單框，並選取在步驟 第 153 頁的『3』中建立的 ConfigMap mqwebuserconfigmap 。

  x) 按一下**建立**。

現在，您可以透過在步驟 第 153 頁的『3』中建立的 ConfigMap 中指定的認證，來存取新佇列管理程式的 IBM MQ Console 。

5. 選擇性的: **套用針對現有佇列管理程式啟用 IBM MQ Console 的配置**。

   編輯您要啟用 IBM MQ Console 之佇列管理程式的 YAML:

   a. 在 OpenShift 主控台上，選取 **操作器 > 已安裝的操作器**。

   b. 選取 IBM MQ Operator 的部署。

   c. 選取 **佇列管理程式** ，然後選取佇列管理程式的名稱。

   d. 選取 **YAML**。

   e. 將佇列管理程式 YAML 的現有 web 區段取代為下列程式碼:

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

   f. 按一下**儲存**。

   現在，您可以透過在步驟 第 153 頁的『3』中建立的 ConfigMap 中指定的認證，來存取現有佇列管理程式的 IBM MQ Console 。

OpenShift CP4I 使用 IBM MQ Operator 操作 IBM MQ

程序

- 第 155 頁的『連接至 Red Hat OpenShift 叢集中部署的 IBM MQ Console』.
- 第 156 頁的『使用 IBM MQ Operator 時監視』.
- 第 162 頁的『使用 Red Hat OpenShift CLI 來備份及還原佇列管理程式配置』.

OpenShift CP4I 連接至 Red Hat OpenShift 叢集中部署的 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

您可以在 Red Hat OpenShift Web 主控台或 IBM Cloud Pak for Integration Platform UI (previously the 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 Red Hat OpenShift CLI 中找到它:

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

如果您使用 IBM Cloud Pak for Integration 授權:

- 若為 IBM MQ Operator 3.0.0 以及更新版本， IBM MQ Console 會使用 Keycloak 來進行身分及存取管理。 請參閱 IBM Cloud Pak for Integration 文件中的 Identity and Access Management 。
- 對於早於 3.0.0 版的 IBM MQ Operator 部署， IBM MQ Console 會使用 IBM Cloud Pak Identity and Access Manager (IAM)。 叢集管理者可能已設定 IAM 元件。 不過，如果這是第一次在 Red Hat OpenShift 叢集上使用 IAM ，則您需要擷取起始管理者密碼。 請參閱 取得起始管理者密碼。

如果您使用 IBM MQ 授權，則 IBM MQ Console 未預先配置，且您需要自行配置它。 如需相關資訊，請參閱配置使用者和角色。 如需範例，請參閱第 153 頁的『使用 IBM MQ Operator 使用基本登錄來配置 IBM MQ Console』。

相關工作

第 140 頁的『配置路徑以從 Red Hat OpenShift 叢集外部連接至佇列管理程式』
您需要 Red Hat 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，因為只有在使用 TLS 1.2 或更高版本的通訊協定時，才能在 TLS 通訊協定中使用 SNI。 Red Hat OpenShift Container Platform Router 使用 SNI 將要求遞送至 IBM MQ 佇列管理程式。

OpenShift CP4I 使用 *IBM Cloud Pak IAM* 授與 *IBM MQ Console* 的許可權

IBM MQ Console 的許可權是透過 IBM Cloud Pak Administration Hub 而非 IBM Cloud Pak for Integration Platform UI (previously the Platform Navigator)來管理。 IBM MQ 不會使用 IBM Cloud Pak for Integration 所提供的「自動化」許可權，而是使用 IBM Cloud Pak Identity and Access Manager (IAM) 所啟用的基本許可權。

程序

1. 開啟 IBM Cloud Pak 管理主控台。

   從 IBM Cloud Pak for Integration Platform UI 中，按一下工具列右上角的 Cloud Pak 切換器 (9 點圖示)，然後按一下 **IBM Cloud Pak 管理** 畫面。
2. 在左上角的導覽功能表中，選取 **身分和存取權**，然後選取 **團隊和服務 ID**。
3. 建立團隊，然後將使用者新增至其中。
   a) 選取 **建立團隊**。
   b) 輸入團隊名稱，然後選取您要管理之使用者的安全網域。
   c) 搜尋使用者。
      這些使用者必須已存在於您的身分提供者中。
   d) 當您找到每一個使用者時，請為他們提供一個角色。 這必須是「管理者」或「叢集管理者」，才能使用 IBM MQ Console 來管理 IBM MQ 。
4. 將每一個使用者新增至名稱空間。
   a) 選取要編輯的團隊。
   b) 選取 **資源 > 管理資源**。
   c) 選取您要此團隊管理的名稱空間。 這些可以是具有佇列管理程式的任何名稱空間。

OpenShift CP4I 使用 IBM MQ Operator 時監視

IBM MQ Operator 所管理的佇列管理程式可以產生與 Prometheus 相容的度量值。

您可以使用 Red Hat OpenShift Container Platform (OCP) 監視堆疊來檢視這些度量值。 開啟 OCP 中的 **度量** 標籤，然後按一下 **觀察 > 度量**。 依預設會啟用佇列管理程式度量，但可以透過將 **.spec.metrics.enabled** 設為 `false` 來停用。

Prometheus 是度量值的時間序列資料庫及規則評估引擎。 IBM MQ 容器會公開可由 Prometheus 查詢的度量值端點。 這些度量是從 MQ 系統主題產生，用於監視及活動追蹤。

OpenShift Container Platform 包括使用 Prometheus 伺服器的預先配置、預先安裝及自行更新監視堆疊。需要配置 OpenShift Container Platform 監視堆疊來監視使用者定義的專案。 如需相關資訊，請參閱 啟用使用者定義專案的監視。 當您建立已啟用度量值的 `QueueManager` 時， IBM MQ Operator 會建立 `ServiceMonitor`，然後 Prometheus 運算子可以探索該度量值。

在舊版 IBM Cloud Pak for Integration 中，您也可以改用 IBM Cloud Platform Monitoring 服務來提供 Prometheus 伺服器。

OpenShift CP4I 使用 *IBM MQ Operator* 時發佈的度量值

佇列管理程式儲存器可以發佈與 Red Hat OpenShift Monitoring 相容的度量值。

度量	類型	說明
ibmmq_qmgr_commit_total	counter	確定計數
ibmmq_qmgr_cpu_load_fifteen_minute_average_percentage	gauge	CPU 負載 - 十五分鐘平均值
ibmmq_qmgr_cpu_load_five_minute_average_percentage	gauge	CPU 負載 - 五分鐘平均值
ibmmq_qmgr_cpu_load_one_minute_average_percentage	gauge	CPU 負載 - 一分鐘平均值
ibmmq_qmgr_destructive_get_bytes_total	counter	間隔總計破壞性取得 - 位元組計數
ibmmq_qmgr_destructive_get_total	counter	間隔總計破壞性取得 - 計數
ibmmq_qmgr_durable_subscription_alter_total	counter	變更可延續訂閱計數
ibmmq_qmgr_durable_subscription_create_total	counter	建立可延續訂閱計數
ibmmq_qmgr_durable_subscription_delete_total	counter	刪除可延續訂閱計數
ibmmq_qmgr_durable_subscription_resume_total	counter	回復可延續訂閱計數
ibmmq_qmgr_errors_file_system_free_space_percentage	gauge	MQ 錯誤檔案系統 - 可用空間
ibmmq_qmgr_errors_file_system_in_use_bytes	gauge	MQ 錯誤檔案系統 - 使用中位元組數
ibmmq_qmgr_expired_message_total	counter	過期訊息計數
ibmmq_qmgr_failed_browse_total	counter	失敗瀏覽計數
ibmmq_qmgr_failed_mqcb_total	counter	失敗 MQCB 計數
ibmmq_qmgr_failed_mqclose_total	counter	失敗 MQCLOSE 計數
ibmmq_qmgr_failed_mqconn_mqconnx_total	counter	失敗 MQCONN/MQCONNX 計數

度量	類型	說明
ibmmq_qmgr_failed_mqget_total	counter	失敗 MQGET - 計數
ibmmq_qmgr_failed_mqinq_total	counter	失敗 MQINQ 計數
ibmmq_qmgr_failed_mqopen_total	counter	失敗 MQOPEN 計數
ibmmq_qmgr_failed_mqput1_total	counter	失敗 MQPUT1 計數
ibmmq_qmgr_failed_mqput_total	counter	失敗 MQPUT 計數
ibmmq_qmgr_failed_mqset_total	counter	失敗 MQSET 計數
ibmmq_qmgr_failed_mqsubrq_total	counter	失敗 MQSUBRQ 計數
ibmmq_qmgr_failed_subscription_create_alter_resume_total	counter	失敗建立/變更/回復訂閱計數
ibmmq_qmgr_failed_subscription_delete_total	counter	訂閱刪除失敗計數
ibmmq_qmgr_failed_topic_mqput_mqput1_total	counter	失敗主題 MQPUT/MQPUT1 計數
ibmmq_qmgr_fdc_files	gauge	MQ FDC 檔案計數
ibmmq_qmgr_log_file_system_in_use_bytes	gauge	日誌檔案系統 - 使用中位元組數
ibmmq_qmgr_log_file_system_max_bytes	gauge	日誌檔案系統 - 位元組數上限
ibmmq_qmgr_log_in_use_bytes	gauge	日誌 - 使用中位元組數
ibmmq_qmgr_log_logical_written_bytes_total	counter	日誌 - 寫入的邏輯位元組數
ibmmq_qmgr_log_max_bytes	gauge	日誌 - 位元組數上限
ibmmq_qmgr_log_occupied_by_reusable_extents_bytes	gauge	日誌 - 可重複使用的範圍所佔用的位元組數
ibmmq_qmgr_log_physical_written_bytes_total	counter	日誌 - 寫入的實體位元組數

度量	類型	說明
ibmmq_qmgr_log_primary_space_in_use_percentage	gauge	日誌 - 使用中的現行主要空間
ibmmq_qmgr_log_required_for_media_recovery_bytes	gauge	日誌 - 媒體回復所需的位元組數
ibmmq_qmgr_log_workload_primary_space_utilization_percentage	gauge	日誌 - 工作量主要空間使用率
ibmmq_qmgr_log_write_latency_seconds	gauge	日誌 - 寫入延遲
ibmmq_qmgr_log_write_size_bytes	gauge	日誌 - 寫入大小
ibmmq_qmgr_mqcb_total	counter	MQCB 計數
ibmmq_qmgr_mqclose_total	counter	MQCLOSE 計數
ibmmq_qmgr_mqconn_mqconnx_total	counter	MQCONN/MQCONNX 計數
ibmmq_qmgr_mqctl_total	counter	MQCTL 計數
ibmmq_qmgr_mqdisc_total	counter	MQDISC 計數
ibmmq_qmgr_mqinq_total	counter	MQINQ 計數
ibmmq_qmgr_mqopen_total	counter	MQOPEN 計數
ibmmq_qmgr_mqput_mqput1_bytes_total	counter	間隔總計 MQPUT/MQPUT1 位元組計數
ibmmq_qmgr_mqput_mqput1_total	counter	間隔總計 MQPUT/MQPUT1 計數
ibmmq_qmgr_mqset_total	counter	MQSET 計數
ibmmq_qmgr_mqstat_total	counter	MQSTAT 計數
ibmmq_qmgr_mqsubrq_total	counter	MQSUBRQ 計數
ibmmq_qmgr_non_durable_subscription_create_total	counter	建立不可延續訂閱計數
ibmmq_qmgr_non_durable_subscription_delete_total	counter	刪除不可延續訂閱計數

度量	類型	說明
ibmmq_qmgr_non_persistent_message_browse_bytes_total	counter	非持續訊息瀏覽 - 位元組計數
ibmmq_qmgr_non_persistent_message_browse_total	counter	非持續訊息瀏覽 - 計數
ibmmq_qmgr_non_persistent_message_destructive_get_total	counter	非持續訊息破壞性取得 - 計數
ibmmq_qmgr_non_persistent_message_get_bytes_total	counter	取得非持續訊息 - 位元組計數
ibmmq_qmgr_non_persistent_message_mqput1_total	counter	非持續訊息 MQPUT1 計數
ibmmq_qmgr_non_persistent_message_mqput_total	counter	非持續訊息 MQPUT 計數
ibmmq_qmgr_non_persistent_message_put_bytes_total	counter	放置非持續訊息 - 位元組計數
ibmmq_qmgr_non_persistent_topic_mqput_mqput1_total	counter	非持續 - 主題 MQPUT/MQPUT1 計數
ibmmq_qmgr_persistent_message_browse_bytes_total	counter	持續訊息瀏覽 - 位元組計數
ibmmq_qmgr_persistent_message_browse_total	counter	持續訊息瀏覽 - 計數
ibmmq_qmgr_persistent_message_destructive_get_total	counter	持續訊息破壞性取得 - 計數
ibmmq_qmgr_persistent_message_get_bytes_total	counter	取得持續訊息 - 位元組計數
ibmmq_qmgr_persistent_message_mqput1_total	counter	持續訊息 MQPUT1 計數
ibmmq_qmgr_persistent_message_mqput_total	counter	持續訊息 MQPUT 計數
ibmmq_qmgr_persistent_message_put_bytes_total	counter	放置持續訊息 - 位元組計數

度量	類型	說明
ibmmq_qmgr_persistent_topic_mqput_mqput1_total	counter	持續 - 主題 MQPUT/MQPUT1 計數
ibmmq_qmgr_published_to_subscribers_bytes_total	counter	已發佈至訂閱者 - 位元組計數
ibmmq_qmgr_published_to_subscribers_message_total	counter	已發佈至訂閱者 - 訊息計數
ibmmq_qmgr_purged_queue_total	counter	已清除佇列計數
ibmmq_qmgr_queue_manager_file_system_free_space_percentage	gauge	佇列管理程式檔案系統 - 可用空間
ibmmq_qmgr_queue_manager_file_system_in_use_bytes	gauge	佇列管理程式檔案系統 - 使用中位元組數
ibmmq_qmgr_ram_free_percentage	gauge	RAM 可用百分比
ibmmq_qmgr_ram_usage_estimate_for_queue_manager_bytes	gauge	RAM 位元組數總計 - 佇列管理程式的估計值
ibmmq_qmgr_rollback_total	counter	回復計數
ibmmq_qmgr_system_cpu_time_estimate_for_queue_manager_percentage	gauge	系統 CPU 時間 - 佇列管理程式的百分比估計值
ibmmq_qmgr_system_cpu_time_percentage	gauge	系統 CPU 時間百分比
ibmmq_qmgr_topic_mqput_mqput1_total	counter	主題 MQPUT/MQPUT1 間隔總計
ibmmq_qmgr_topic_put_bytes_total	counter	間隔總計主題放置位元組數
ibmmq_qmgr_trace_file_system_free_space_percentage	gauge	MQ 追蹤檔案系統 - 可用空間
ibmmq_qmgr_trace_file_system_in_use_bytes	gauge	MQ 追蹤檔案系統 - 使用中位元組數
ibmmq_qmgr_user_cpu_time_estimate_for_queue_manager_percentage	gauge	使用者 CPU 時間 - 佇列管理程式的百分比估計值

度量	類型	說明
ibmmq_qmgr_user_cpu_time_percentage	gauge	使用者 CPU 時間百分比

相關資訊

在系統主題上發佈的度量值

OpenShift CP4I 使用 Red Hat OpenShift CLI 來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，您可以執行下列指令，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
  ```

  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，您可以執行下列指令，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
  ```

  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
  ```

OpenShift CP4I 對 IBM MQ Operator 的問題進行疑難排解

如果您對 IBM MQ Operator 有問題，請使用說明的技術來協助您診斷及解決它們。

程序

- 第 163 頁的『收集使用 IBM MQ Operator 部署的佇列管理程式的疑難排解資訊』
- 第 164 頁的『疑難排解: 取得佇列管理程式資料的存取權』

OpenShift CP4I 收集使用 IBM MQ Operator 部署的佇列管理程式的疑難排解資訊

收集在提出新的支援案例時應該提供給「IBM 支援中心」的疑難排解資訊。

程序

1. 收集雲端提供者資訊。

   這是管理 Red Hat OpenShift 叢集的雲端提供者 (例如， IBM Cloud)。
2. 收集架構資訊。

   Red Hat OpenShift 叢集的架構是下列其中一項:

   - Linux for x86-64
   - Linux on Power Systems (ppc64le)
   - Linux for IBM Z
3. 收集 IBM MQ 部署資訊。

   a) 使用 bash/zsh Shell 登入 Red Hat OpenShift 叢集。

   b) 設定下列環境變數:

   ```
   export QM=QueueManager_name
   export QM_NAMESPACE=QueueManager_namespace
   export MQ_OPERATOR_NAMESPACE=mq_operator_namespace
   ```

   其中 *QueueManager_name* 是 QueueManager 資源的名稱， *QueueManager_namespace* 是部署它的名稱空間，而 *mq_operator_namespace* 是部署 IBM MQ Operator 的名稱空間。 這可能與 QueueManager 名稱空間相同。

   c) 執行下列指令，並將所有產生的輸出檔提供給 IBM 支援中心。

   ```
   # OCP / Kubernetes: Version
   oc version -o yaml > ocversion.yaml

   # QueueManager: YAML
   oc get qmgr $QM -n $QM_NAMESPACE -o yaml > "queue-manager-$QM.yaml"

   # MQ Queue Manager: Pods
   oc get pods -n $QM_NAMESPACE -o wide --selector "app.kubernetes.io/instance=$QM" > "qm-
   pods-$QM.txt"

   # MQ Queue Manager: Pod YAML
   oc get pods -n $QM_NAMESPACE -o yaml --selector "app.kubernetes.io/instance=$QM" > "qm-
   pods-$QM.yaml"

   # MQ Queue Manager: Pod Logs
   for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/
   instance=$QM" | cut -d ' ' -f 1); do oc logs -n $QM_NAMESPACE --previous "$p" > "qm-logs-
   previous-$p.txt"; oc logs -n $QM_NAMESPACE $p > "qm-logs-$p.txt";done

   # MQ Web UI: Console Log
   for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/
   instance=$QM" | cut -d ' ' -f 1); do oc cp -n $QM_NAMESPACE --retries=10 "$p:var/mqm/web/
   installations/Installation1/servers/mqweb/logs/console.log" "web-$p-console.log"; done

   # MQ Web UI: Messages Log
   for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/
   instance=$QM" | cut -d ' ' -f 1); do oc cp -n $QM_NAMESPACE --retries=10 "$p:var/mqm/web/
   installations/Installation1/servers/mqweb/logs/messages.log" "web-$p-messages.log"; done

   # MQ Queue Manager: routes defined by operator
   oc get routes -n $QM_NAMESPACE -o yaml --selector "app.kubernetes.io/instance=$QM" > "qm-
   routes-$QM.yaml"

   # MQ Queue Manager: routes to QM
   oc get routes -n $QM_NAMESPACE -o yaml --field-selector "spec.to.name=$QM-ibm-mq" > "qm-
   routes2-$QM.yaml"

   # MQ Queue Manager: stateful set
   oc get statefulset -n $QM_NAMESPACE -o yaml ${QM}-ibm-mq > "qm-statefulset-$QM.yaml"
   ```

```
# MQ Queue Manager: services
oc get services -n $QM_NAMESPACE -o yaml --selector "app.kubernetes.io/instance=$QM" >
"qm-services-$QM.yaml"

# MQ Queue Manager: PVCs
oc get pvc -n $QM_NAMESPACE -o yaml --selector "app.kubernetes.io/instance=$QM" > "qm-
pvcs-$QM.yaml"

# MQ Operator: Version
oc get csv -n $QM_NAMESPACE | grep "^ibm-mq\|NAME" > mq-operator-csv.txt

# Cloud Pak Foundational Services: Version
oc get csv -n $QM_NAMESPACE | grep "^ibm-common-service-operator\|NAME" > common-services-
csv.txt

# Cloud Pak for Integration: Version (if applicable)
oc get csv -n $QM_NAMESPACE | grep "^ibm-integration-platform-navigator\|NAME" > cp4i-
csv.txt

# Output from runmqras (this may take a while to execute)
for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/
instance=$QM" | cut -d ' ' -f 1); do timestamp=$(TZ=UTC date +"%Y%m%d_%H%M%S"); oc exec
-n $QM_NAMESPACE $p -- runmqras -workdirectory "/tmp/runmqras_$timestamp" -section
logger,mqweb,nativeha,trace; oc cp -n $QM_NAMESPACE --retries=10 "$p:tmp/
runmqras_$timestamp/" .; done

# MQ Operator: Pod Log
oc logs -n $MQ_OPERATOR_NAMESPACE $(oc get pods -n $MQ_OPERATOR_NAMESPACE --no-headers --
selector app.kubernetes.io/name=ibm-mq,app.kubernetes.io/managed-by=olm | cut -d ' ' -f
1) > mq-operator-log.txt
```

註:

這些指令大部分都需要存取佇列管理程式部署所在的名稱空間。 不過，如果 IBM MQ Operator 已安裝 **叢集範圍**，則收集 IBM MQ Operator 日誌可能還需要 **叢集管理者** 存取權。

相關工作

收集 IBM 支援中心的疑難排解資訊

OpenShift CP4I 疑難排解: 取得佇列管理程式資料的存取權

使用 PVC Inspector 工具來存取佇列管理程式 PVC 上的檔案，其中無法建立佇列管理程式 Pod 的遠端 Shell。 這可能是因為 Pod 處於 **Error** 或 **CrashLoopBackOff** 狀態。 此工具設計用於與 IBM MQ Operator 所部署的佇列管理程式搭配使用。

開始之前

使用 PVC Inspector 工具。 您必須具有佇列管理程式名稱空間的存取權。

關於這項作業

為了協助疑難排解，您可以存取儲存在與給定佇列管理程式相關聯之「持續性磁區要求 (PVC)」上的資料。 若要這樣做，您可以使用工具將 PVC 裝載至一組 Inspector Pod。 然後，您可以將遠端 Shell 放入任何 Inspector Pod 中，以讀取檔案。

視部署類型而定，會在一到三個 Inspector Pod 之間建立。 在相關聯的 PVC 檢查程式 Pod 上，提供 Native-HA 或 Multi-Instance 佇列管理程式的給定 Pod 特定的磁區。 共用磁區在所有檢查程式上都可用。 Inspector Pod 的名稱包含相關聯佇列管理程式 Pod 的名稱。

程序

1. 下載 MQ PVC 檢查程式工具。

   這裡提供此工具: https://github.com/ibm-messaging/mq-pvc-tool。
2. 請確定您已登入叢集。
3. 找出佇列管理程式的名稱，以及佇列管理程式執行所在的名稱空間。
4. 針對佇列管理程式執行檢查程式工具。

a) 執行下列指令，並指定佇列管理程式名稱及其名稱空間名稱。

```
./pvc-tool.sh queue_manager_name queue_manager_namespace_name
```

b) 工具完成之後，請執行下列指令，以檢視正在建立的 Inspector Pod。

```
oc get pods
```

5. 檢視裝載至 Inspector Pod 的檔案。

a) 每一個 PVC Inspector Pod 都與佇列管理程式 Pod 相關聯，因此可能有多個 Inspector Pod。 執行下列指令，以存取其中一個 Pod:

```
oc rsh pvc-inspector-pod-name
```

您會被放置在包含已裝載 PVC 目錄的目錄中。

b) 執行下列指令，以列出 PVC 目錄:

```
ls
```

c) 在遠端 Shell 階段作業之外執行下列指令，以查看 PVC 清單:

```
oc get pvc
```

d) 執行下列指令，以清除工具所建立的 Pod:

```
oc delete pods -l tool=mq-pvc-inspector
```

OpenShift CP4I IBM MQ Operator 的 API 參考資料

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

OpenShift CP4I mq.ibm.com/v1beta1 的 API 參考資料

v1beta1 API 可用來建立及管理 QueueManager 資源。

OpenShift CP4I-LTS CP4I CD *mq.ibm.com/v1beta1* 的授權參考手冊

現行授權版本

`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: **spec.license.license**	下者的值: **spec.license.use**	授權資訊	適用的 IBM MQ 版本
L-VTPK-22YZPK	Production 或 NonProduction	IBM Cloud Pak for Integration 2023.4.1	9.3.4 或 9.3.5
L-QYQF-8UFZBN	Production 或 NonProduction	IBM Cloud Pak for Integration Limited Edition 2023.4.1	9.3.4 或 9.3.5
L-AMRD-XH6P3Q	Production	IBM MQ Advanced 及 IBM MQ Advanced for Non-Production Environment 9.3 -05/2023	9.3.3、 9.3.4 或 9.3.5
L-AXAF-JLZ53A	Development	IBM MQ Advanced for Developers (非 Warranted) 9.3 -05/2023	9.3.3、 9.3.4 或 9.3.5
L-YBXJ-ADJNSM	Production 或 NonProduction	IBM Cloud Pak for Integration 2023.2.1	9.3.3

下者的值: **spec.license.license**	下者的值: **spec.license.use**	授權資訊	適用的 IBM MQ 版本
L-PYRA-849GYQ	Production 或 NonProduction	IBM Cloud Pak for Integration Limited Edition 2023.2.1	9.3.3
L-RJON-CJR2RX	Production 或 NonProduction	IBM Cloud Pak for Integration 2022.4.1	9.3.1 或 9.3.2
L-RJON-CJR2TC	Production 或 NonProduction	IBM Cloud Pak for Integration Limited Edition 2022.4.1	9.3.1 或 9.3.2
L-UPFX-8MW49T	Production	IBM MQ Advanced 及 IBM MQ Advanced for Non-Production Environment 9.3 -02/2023	9.3.2
L-APIG-CAUEQC	Development	IBM MQ Advanced for Developers (非 Warranted) 9.3	9.3.0、 9.3.1 或 9.3.2
L-RJON-CD3JKX	Production 或 NonProduction	IBM Cloud Pak for Integration 2022.2.1	9.3.0 或 9.3.1
L-RJON-CD3JJU	Production 或 NonProduction	IBM Cloud Pak for Integration Limited Edition 2022.2.1	9.3.0 或 9.3.1
L-APIG-CAUEBE	Production	IBM MQ Advanced 和 IBM MQ Advanced for Non-Production Environment 9.3	9.3.0 或 9.3.1

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

較舊的授權版本

請參閱 IBM MQ 9.2 說明文件中的 舊版授權 。

OpenShift CP4I *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。 IBM MQ 說明文件: https://ibm.biz/BdPZqj。 授權參照: https://ibm.biz/BdPZfq.。

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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources。
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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-tind。
metadata	
spec QueueManager 規格	所需的 QueueManager 狀態。
status QueueManager 狀態	QueueManager 的觀察狀態。

.spec

所需的 QueueManager 狀態。

出現在:

- 第 166 頁的『QueueManager』

欄位	說明
affinity	標準 Kubernetes 親緣性規則。 如需相關資訊，請參閱 https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#affinity-v1-core。
annotations 註釋	註釋欄位用作 Pod 註釋的透通。 使用者可以將任何註釋新增至此欄位，並讓它套用至 Pod。 這裡的註釋會改寫預設註釋 (如果有提供的話)。 需要 MQ Operator 1.3.0 或更高版本。
imagePullSecrets LocalObjectReference 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 https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod。
labels 標籤	標籤欄位用作 Pod 標籤的透通。 使用者可以將任何標籤新增至此欄位，並讓它套用至 Pod。 這裡的標籤會改寫預設標籤 (如果有提供的話)。 需要 MQ Operator 1.3.0 或更高版本。
license 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。
pki 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。
queueManager QueueManager 配置	「佇列管理程式」儲存器及基礎「佇列管理程式」的設定。
securityContext SecurityContext	要新增至佇列管理程式 Pod 的 securityContext 的安全設定。
telemetry 遙測	Open Telemetry 配置的設定。 需要 MQ 操作器 2.2.0 或更高版本。
template 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。
terminationGracePeriodSeconds 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，會岔斷必要的佇列管理程式維護作業。 預設值為 30 秒。
tracing TracingConfig	與 Cloud Pak for Integration 作業儀表板進行追蹤整合的設定。
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程式，例如基本映像檔的修正程式。
web WebServer 配置	MQ Web 伺服器的設定。

.spec.annotations

註釋欄位用作 Pod 註釋的透通。 使用者可以將任何註釋新增至此欄位，並讓它套用至 Pod。 這裡的註釋會改寫預設註釋 (如果有提供的話)。 需要 MQ Operator 1.3.0 或更高版本。

出現在:

- 第 167 頁的『.spec』

.spec.imagePullSecrets

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 167 頁的『.spec』

欄位	說明
name 字串	參照的名稱。 進一步資訊: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names 待辦事項: 新增其他有用欄位。 apiVersion， kind， uid?。

.spec.labels

標籤欄位用作 Pod 標籤的透通。 使用者可以將任何標籤新增至此欄位，並讓它套用至 Pod。 這裡的標籤會改寫預設標籤 (如果有提供的話)。 需要 MQ Operator 1.3.0 或更高版本。

出現在:

- 第 167 頁的『.spec』

.spec.license

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 167 頁的『.spec』

欄位	說明
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 https://ibm.biz/BdPZfq 。
metric 字串	指定要使用的授權標準的設定。 例如， ProcessorValueUnit、VirtualProcessorCore 或 ManagedVirtualServer。 使用 MQ 授權時預設為 ProcessorValueUnit， 使用 Cloud Pak for Integration 授權時預設為 VirtualProcessorCore 。
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 https://ibm.biz/BdPZfq 。

.spec.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 167 頁的『.spec』

欄位	說明
keys PKISource 陣列	要新增至佇列管理程式金鑰儲存庫的私密金鑰。
trust PKISource 陣列	要新增至佇列管理程式金鑰儲存庫的憑證。

.spec.pki.keys

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 168 頁的『.spec.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 密鑰	使用 Kubernetes 密鑰來提供金鑰。

.spec.pki.keys.secret

使用 Kubernetes 密鑰來提供金鑰。

出現在:

- 第 168 頁的『.spec.pki.keys』

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。

.spec.pki.trust

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 168 頁的『.spec.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 密鑰	使用 Kubernetes 密鑰來提供金鑰。

.spec.pki.trust.secret

使用 Kubernetes 密鑰來提供金鑰。

出現在:

- 第 169 頁的『.spec.pki.trust』

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。

.spec.queueManager

「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 167 頁的『.spec』

欄位	說明
availability 可用性	佇列管理程式的可用性設定，例如是否使用主動-待命配對或原生高可用性。
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。 預設為 false。

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。 預設為 IfNotPresent。
ini INISource 陣列	為佇列管理程式提供 INI 的設定。 需要 MQ Operator 1.1.0 或更高版本。
livenessProbe QueueManagerLivenessProbe	控制存活性探測的設定。
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 Basic 。 預設為 Basic。
metrics QueueManager 度量	Prometheus 樣式度量值的設定。
mqsc MQSCSource 陣列	為佇列管理程式提供 MQSC 的設定。 需要 MQ Operator 1.1.0 或更高版本。
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。
readinessProbe QueueManagerReadinessProbe	控制就緒性探測的設定。
recoveryLogs RecoveryLogs	MQ 回復日誌的設定。 需要 MQ Operator 2.4.0 或更高版本。
resources 資源	控制資源需求的設定。
route 路徑	佇列管理程式路徑的設定。 需要 MQ Operator 1.4.0 或更高版本。
startupProbe StartupProbe	控制啟動探測的設定。 僅適用於 MultiInstance 及 NativeHA 部署。 需要 MQ Operator 1.5.0 或更高版本。
storage QueueManager 儲存體	儲存體設定，用於控制「佇列管理程式」對持續性磁區及儲存類別的使用。

.spec.queueManager.availability

佇列管理程式的可用性設定，例如是否使用主動-待命配對或原生高可用性。

出現在:

- 第 169 頁的『.spec.queueManager』

欄位	說明
tls TLS	用於在 NativeHA 抄本之間配置安全通訊的選用 TLS 設定。 需要 MQ Operator 1.5.0 或更高版本。
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 Pod 配對使用 MultiInstance ，其中一個是 active 佇列管理程式，另一個是待命。 將 NativeHA 用於原生高可用性抄寫 (需要 MQ Operator 1.5.0 或更高版本)。 預設為 SingleInstance。 如需詳細資料，請參閱 http://ibm.biz/BdqAQa 。
updateStrategy 字串	用於 MultiInstance 及 NativeHA 佇列管理程式的更新策略。 每當佇列管理程式配置變更時，請使用 RollingUpdate 來啟用自動漸進式更新。 使用 OnDelete 來停用自動漸進式更新，只有在刪除 Pod (包括由外部因素觸發的 Pod 刪除) 時，才會套用「佇列管理程式」變更。 預設為 RollingUpdate。 需要 MQ Operator 1.6.0 或更高版本。

.spec.queueManager.availability.tls

用於在 NativeHA 抄本之間配置安全通訊的選用 TLS 設定。 需要 MQ Operator 1.5.0 或更高版本。

出現在:

- 第 170 頁的『.spec.queueManager.availability』

欄位	說明
cipherSpec 字串	NativeHA TLS 的 CipherSpec 名稱。
secretName 字串	Kubernetes 密鑰的名稱。

.spec.queueManager.ini

INI 配置檔的來源。

出現在:

- 第 169 頁的『.spec.queueManager』

欄位	說明
configMap ConfigMapINISource	ConfigMap 代表包含 INI 資訊的 Kubernetes ConfigMap 。
secret SecretINISource	密鑰代表包含 INI 資訊的 Kubernetes 密鑰。

.spec.queueManager.ini.configMap

ConfigMap 代表包含 INI 資訊的 Kubernetes ConfigMap 。

出現在:

- 第 171 頁的『.spec.queueManager.ini』

欄位	說明
items 陣列	Kubernetes 來源內應該套用的金鑰。
name 字串	Kubernetes 來源的名稱。

.spec.queueManager.ini.secret

密鑰代表包含 INI 資訊的 Kubernetes 密鑰。

出現在:

- 第 171 頁的『.spec.queueManager.ini』

欄位	說明
items 陣列	Kubernetes 來源內應該套用的金鑰。
name 字串	Kubernetes 來源的名稱。

.spec.queueManager.livenessProbe

控制存活性探測的設定。

出現在:

- 第 169 頁的『.spec.queueManager』

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。 預設值為 1。
initialDelaySeconds 整數	在啟動儲存器之後，起始探測之前的秒數。 SingleInstance 預設為 90 秒。 對於 MultiInstance 及 NativeHA 部署，預設為 0 秒。 進一步資訊: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes。
periodSeconds 整數	執行探測的頻率（以秒為單位）。 預設值為 10 秒。
successThreshold 整數	在失敗之後將探測視為成功的連續成功數下限。 預設值為 1。
timeoutSeconds 整數	探測逾時之前經歷的秒數。 預設值為 5 秒。 進一步資訊: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes。

.spec.queueManager.metrics

Prometheus 樣式度量值的設定。

出現在:

- 第 169 頁的『.spec.queueManager』

欄位	說明
enabled 布林值	是否啟用 Prometheus 相容度量值的端點。 預設為 true。

.spec.queueManager.mqsc

MQSC 配置檔的來源。

出現在:

- 第 169 頁的『.spec.queueManager』

欄位	說明
configMap ConfigMapMQSCSource	ConfigMap 代表包含 MQSC 資訊的 Kubernetes ConfigMap 。
secret SecretMQSCSource	Secret 代表包含 MQSC 資訊的 Kubernetes 「密鑰」。

.spec.queueManager.mqsc.configMap

ConfigMap 代表包含 MQSC 資訊的 Kubernetes ConfigMap 。

出現在:

- 第 172 頁的『.spec.queueManager.mqsc』

欄位	說明
items 陣列	Kubernetes 來源內應該套用的金鑰。
name 字串	Kubernetes 來源的名稱。

.spec.queueManager.mqsc.secret

Secret 代表包含 MQSC 資訊的 Kubernetes 「密鑰」。

出現在:

- 第 172 頁的『.spec.queueManager.mqsc』

欄位	說明
items 陣列	Kubernetes 來源內應該套用的金鑰。
name 字串	Kubernetes 來源的名稱。

.spec.queueManager.readinessProbe

控制就緒性探測的設定。

出現在:

- 第 169 頁的『.spec.queueManager』

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。 預設值為 1。
initialDelaySeconds 整數	在啟動儲存器之後，起始探測之前的秒數。 SingleInstance 預設為 10 秒。 對於 MultiInstance 和 NativeHA 部署，預設值為 0。 進一步資訊: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes。
periodSeconds 整數	執行探測的頻率（以秒為單位）。 預設值為 5 秒。
successThreshold 整數	在失敗之後將探測視為成功的連續成功數下限。 預設值為 1。
timeoutSeconds 整數	探測逾時之前經歷的秒數。 預設值為 3 秒。 進一步資訊: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes。

.spec.queueManager.recoveryLogs

MQ 回復日誌的設定。 需要 MQ Operator 2.4.0 或更高版本。

出現在:

- 第 169 頁的『.spec.queueManager』

欄位	說明
logFilePages 整數	回復日誌資料保留在一系列檔案中。 日誌檔大小以 4 KB 頁面為單位指定。

.spec.queueManager.resources

控制資源需求的設定。

出現在:

- 第 169 頁的『.spec.queueManager』

欄位	說明
limits 限制	CPU 及記憶體設定。
requests 要求	CPU 及記憶體設定。

.spec.queueManager.resources.limits

CPU 及記憶體設定。

出現在:

- 第 173 頁的『.spec.queueManager.resources』

欄位	說明
cpu	
memory	

.spec.queueManager.resources.requests

CPU 及記憶體設定。

出現在:

- 第 173 頁的『.spec.queueManager.resources』

欄位	說明
cpu	
memory	

.spec.queueManager.route

佇列管理程式路徑的設定。 需要 MQ Operator 1.4.0 或更高版本。

出現在:

- 第 169 頁的『.spec.queueManager』

欄位	說明
enabled 布林值	是否啟用路徑。 預設為 true。

.spec.queueManager.startupProbe

控制啟動探測的設定。 僅適用於 MultiInstance 及 NativeHA 部署。 需要 MQ Operator 1.5.0 或更高版本。

出現在:

- 第 169 頁的『.spec.queueManager』

欄位	說明
failureThreshold 整數	將探測視為失敗的連續失敗次數下限。 預設值為 24。
initialDelaySeconds 整數	在啟動儲存器之後，起始探測之前的秒數。 預設值為 0 秒。 進一步資訊: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes。
periodSeconds 整數	執行探測的頻率（以秒為單位）。 預設值為 5 秒。
successThreshold 整數	將探測視為成功的連續成功數下限。 預設值為 1。
timeoutSeconds 整數	探測逾時之前經歷的秒數。 預設值為 5 秒。 進一步資訊: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes。

.spec.queueManager.storage

儲存體設定，用於控制「佇列管理程式」對持續性磁區及儲存類別的使用。

出現在:

- 第 169 頁的『.spec.queueManager』

欄位	說明
defaultClass 字串	依預設套用至此佇列管理程式的所有持續性磁區的儲存類別。 特定持續性磁區可以定義自己的儲存類別，以置換此預設儲存類別設定。 如果 type of availability 是 SingleInstance 或 NativeHA，則儲存類別可以是 ReadWriteOnce 或 ReadWriteMany 類型。 如果 type of availability 是 MultiInstance，則儲存類別必須是 ReadWriteMany 類型。
defaultDeleteClaim 布林值	刪除「佇列管理程式」時是否應該刪除所有磁區。 特定持續性磁區可以為 deleteClaim 定義自己的值，這將置換此 defaultDeleteClaim 設定。 預設為 false。
persistedData QueueManagerOptionalVolume	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。
queueManager 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。
recoveryLogs QueueManagerOptionalVolume	MQ 回復日誌的持續性磁區詳細資料。 使用多重實例佇列管理程式時需要。
scratch 標為暫時刪除	佇列管理程式之暫存暫時磁區的設定。 此磁區將裝載為儲存器上的 '/run' 資料夾。 只有在 root 檔案系統設為唯讀時才適用。 需要 MQ Operator 3.0.0 或更高版本。
tmp 暫存	佇列管理程式的 Tmp 暫時磁區設定。 此磁區將裝載在儲存器上作為 '/tmp' 資料夾。 將在此磁區中建立診斷資料檔案 (例如 runmqras 指令產生的 zip 檔案)。 只有在 root 檔案系統設為唯讀時才適用。 需要 MQ Operator 3.0.0 或更高版本。

.spec.queueManager.storage.persistedData

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。

出現在:

- 第 174 頁的『.spec.queueManager.storage』

欄位	說明
class 字串	要用於此磁區的儲存類別。 僅當 type 為 persistent-claim 時才有效。 如果 type of availability 是 SingleInstance 或 NativeHA，則儲存類別可以是 ReadWriteOnce 或 ReadWriteMany 類型。 如果 type of availability 是 MultiInstance，則儲存類別必須是 ReadWriteMany 類型。
deleteClaim 布林值	刪除佇列管理程式時是否應刪除此磁區。
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。 預設為 false。
size 字串	要傳遞至 Kubernetes(包括 SI 單位) 的 PersistentVolume 大小。 僅當 type 為 persistent-claim 時才有效。 例如，2Gi。 預設為 2Gi。
sizeLimit 字串	使用 ephemeral 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。 只有在 type 是 ephemeral 且 root 檔案系統設為唯讀時才有效。 需要 MQ Operator 3.0.0 或更高版本。
type 字串	要使用的磁區類型。 選擇 ephemeral 以使用非持續性儲存體，或選擇 persistent-claim 以使用持續性磁區。 預設為 persistent-claim。

.spec.queueManager.storage.queueManager

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 174 頁的『.spec.queueManager.storage』

欄位	說明
class 字串	要用於此磁區的儲存類別。 僅當 type 為 persistent-claim 時才有效。 如果 type of availability 是 SingleInstance 或 NativeHA，則儲存類別可以是 ReadWriteOnce 或 ReadWriteMany 類型。 如果 type of availability 是 MultiInstance，則儲存類別必須是 ReadWriteMany 類型。
deleteClaim 布林值	刪除佇列管理程式時是否應刪除此磁區。
size 字串	要傳遞至 Kubernetes(包括 SI 單位) 的 PersistentVolume 大小。 僅當 type 為 persistent-claim 時才有效。 例如，2Gi。 預設為 2Gi。
sizeLimit 字串	使用 ephemeral 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。 只有在 type 是 ephemeral 且 root 檔案系統設為唯讀時才有效。 需要 MQ Operator 3.0.0 或更高版本。
type 字串	要使用的磁區類型。 選擇 ephemeral 以使用非持續性儲存體，或選擇 persistent-claim 以使用持續性磁區。 預設為 persistent-claim。

.spec.queueManager.storage.recoveryLogs

MQ 回復日誌的持續性磁區詳細資料。 使用多重實例佇列管理程式時需要。

出現在:

- 第 174 頁的『.spec.queueManager.storage』

欄位	說明
class 字串	要用於此磁區的儲存類別。 僅當 type 為 persistent-claim 時才有效。 如果 type of availability 是 SingleInstance 或 NativeHA，則儲存類別可以是 ReadWriteOnce 或 ReadWriteMany 類型。 如果 type of availability 是 MultiInstance，則儲存類別必須是 ReadWriteMany 類型。
deleteClaim 布林值	刪除佇列管理程式時是否應刪除此磁區。
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。 預設為 false。
size 字串	要傳遞至 Kubernetes(包括 SI 單位) 的 PersistentVolume 大小。 僅當 type 為 persistent-claim 時才有效。 例如，2Gi。 預設為 2Gi。
sizeLimit 字串	使用 ephemeral 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。 只有在 type 是 ephemeral 且 root 檔案系統設為唯讀時才有效。 需要 MQ Operator 3.0.0 或更高版本。
type 字串	要使用的磁區類型。 選擇 ephemeral 以使用非持續性儲存體，或選擇 persistent-claim 以使用持續性磁區。 預設為 persistent-claim。

.spec.queueManager.storage.scratch

佇列管理程式之暫存暫時磁區的設定。 此磁區將裝載為儲存器上的 '/run' 資料夾。 只有在 root 檔案系統設為唯讀時才適用。 需要 MQ Operator 3.0.0 或更高版本。

出現在:

- 第 174 頁的『.spec.queueManager.storage』

欄位	說明
`sizeLimit` 字串	暫時磁區的大小限制，包括 SI 單位。 例如，`2Gi`。 只有在 root 檔案系統設為唯讀時才有效。 需要 MQ Operator 3.0.0 或更高版本。

.spec.queueManager.storage.tmp

佇列管理程式的 Tmp 暫時磁區設定。 此磁區將裝載在儲存器上作為 '/tmp' 資料夾。 將在此磁區中建立診斷資料檔案 (例如 runmqras 指令產生的 zip 檔案)。 只有在 root 檔案系統設為唯讀時才適用。 需要 MQ Operator 3.0.0 或更高版本。

出現在:

- 第 174 頁的『.spec.queueManager.storage』

欄位	說明
`sizeLimit` 字串	暫時磁區的大小限制，包括 SI 單位。 例如，`2Gi`。 只有在 root 檔案系統設為唯讀時才有效。 需要 MQ Operator 3.0.0 或更高版本。

.spec.securityContext

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

出現在:

- 第 167 頁的『.spec』

欄位	說明
`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，而且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。 預設為 false。 如需相關資訊，請參閱 https://docs.openshift.com/container-platform/latest/authentication/managing-security-context-constraints.html。
`readOnlyRootFilesystem` 布林值	是否啟用「佇列管理程式」的唯讀 root 檔案系統設定。 預設為 false。 需要 MQ Operator 3.0.0 或更高版本。
`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。

.spec.telemetry

Open Telemetry 配置的設定。 需要 MQ 操作器 2.2.0 或更高版本。

出現在:

- 第 167 頁的『.spec』

欄位	說明
`tracing` 追蹤	「開啟遙測」追蹤的設定。

.spec.telemetry.tracing

「開啟遙測」追蹤的設定。

出現在:

- 第 177 頁的『.spec.telemetry』

欄位	說明
instana Instana	Instana 追蹤的設定。

.spec.telemetry.tracing.instana

Instana 追蹤的設定。

出現在:

- 第 178 頁的『.spec.telemetry.tracing』

欄位	說明
agentHost 字串	要將追蹤資料傳送至其中之 Instana 代理程式的主機名稱。 這不應包括通訊協定。
enabled 布林值	是否啟用 Instana 追蹤。 預設為 false。
protocol 字串	與 Instana 代理程式通訊時要使用的通訊協定。 支援 http 及 https。

.spec.template

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 167 頁的『.spec』

欄位	說明
pod	置換用於 Pod 的範本。 請參閱 https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#podspec-v1-core。

.spec.tracing

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

出現在:

- 第 167 頁的『.spec』

欄位	說明
agent TracingAgent	僅在 Cloud Pak for Integration 中，您可以配置選用 Tracing Agent 的設定。
collector 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。 預設為 false。
namespace 字串	Cloud Pak for Integration 作業儀表板安裝所在的名稱空間。

.spec.tracing.agent

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

出現在:

- 第 178 頁的『.spec.tracing』

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。 預設為 IfNotPresent。
livenessProbe TracingProbe	控制存活性探測的設定。
readinessProbe TracingProbe	控制就緒性探測的設定。

.spec.tracing.agent.livenessProbe

控制存活性探測的設定。

出現在:

- 第 179 頁的『.spec.tracing.agent』

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。 預設值為 1。
initialDelaySeconds 整數	在儲存器啟動之後，起始活性探測之前的秒數。 預設值為 10 秒。 進一步資訊: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes。
periodSeconds 整數	執行探測的頻率（以秒為單位）。 預設值為 10 秒。
successThreshold 整數	在失敗之後將探測視為成功的連續成功數下限。 預設值為 1。
timeoutSeconds 整數	探測逾時之前經歷的秒數。 預設值為 2 秒。 進一步資訊: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes。

.spec.tracing.agent.readinessProbe

控制就緒性探測的設定。

出現在:

- 第 179 頁的『.spec.tracing.agent』

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。 預設值為 1。
initialDelaySeconds 整數	在儲存器啟動之後，起始活性探測之前的秒數。 預設值為 10 秒。 進一步資訊: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes。
periodSeconds 整數	執行探測的頻率（以秒為單位）。 預設值為 10 秒。
successThreshold 整數	在失敗之後將探測視為成功的連續成功數下限。 預設值為 1。
timeoutSeconds 整數	探測逾時之前經歷的秒數。 預設值為 2 秒。 進一步資訊: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes。

.spec.tracing.collector

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 178 頁的『.spec.tracing』

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。 預設為 IfNotPresent。
livenessProbe TracingProbe	控制存活性探測的設定。
readinessProbe TracingProbe	控制就緒性探測的設定。

.spec.tracing.collector.livenessProbe

控制存活性探測的設定。

出現在:

- 第 180 頁的『.spec.tracing.collector』

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。 預設值為 1。
initialDelaySeconds 整數	在儲存器啟動之後，起始活性探測之前的秒數。 預設值為 10 秒。 進一步資訊: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes。
periodSeconds 整數	執行探測的頻率（以秒為單位）。 預設值為 10 秒。
successThreshold 整數	在失敗之後將探測視為成功的連續成功數下限。 預設值為 1。
timeoutSeconds 整數	探測逾時之前經歷的秒數。 預設值為 2 秒。 進一步資訊: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes。

.spec.tracing.collector.readinessProbe

控制就緒性探測的設定。

出現在:

- 第 180 頁的『.spec.tracing.collector』

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。 預設值為 1。
initialDelaySeconds 整數	在儲存器啟動之後，起始活性探測之前的秒數。 預設值為 10 秒。 進一步資訊: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes。
periodSeconds 整數	執行探測的頻率（以秒為單位）。 預設值為 10 秒。
successThreshold 整數	在失敗之後將探測視為成功的連續成功數下限。 預設值為 1。
timeoutSeconds 整數	探測逾時之前經歷的秒數。 預設值為 2 秒。 進一步資訊: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes。

.spec.web

MQ Web 伺服器的設定。

出現在:

- 第 167 頁的『.spec』

欄位	說明
`console` 主控台	MQ Web 主控台的設定。 需要 MQ Operator 3.0.0 或更高版本。
`enabled` 布林值	是否啟用 Web 伺服器。 預設為 false。
`manualConfig` ManualConfig	提供 Web 伺服器 XML 配置的設定。 需要 MQ Operator 3.0.0 或更高版本。

.spec.web.console

MQ Web 主控台的設定。 需要 MQ Operator 3.0.0 或更高版本。

出現在:

- 第 181 頁的『.spec.web』

欄位	說明
`authentication` 鑑別	MQ Web 主控台的鑑別設定。 需要 MQ Operator 3.0.0 或更高版本。
`authorization` 授權	MQ Web 主控台的授權設定。 需要 MQ Operator 3.0.0 或更高版本。

.spec.web.console.authentication

MQ Web 主控台的鑑別設定。 需要 MQ Operator 3.0.0 或更高版本。

出現在:

- 第 181 頁的『.spec.web.console』

欄位	說明
`provider` 字串	用於 MQ Web 主控台的鑑別提供者。 使用 `integration-keycloak`，以使用 Cloud Pak for Integration Platform 使用者介面 (Keycloak) 進行單一登入。 如果您使用 Cloud Pak for Integration 授權，則預設為 `integration-keycloak`；如果您使用 MQ 授權，則預設為 `manual`。 如果您想要提供自己的配置，請使用 `manual`。

.spec.web.console.authorization

MQ Web 主控台的授權設定。 需要 MQ Operator 3.0.0 或更高版本。

出現在:

- 第 181 頁的『.spec.web.console』

欄位	說明
`provider` 字串	用於 MQ Web 主控台的授權提供者。 使用 `integration-keycloak` 來使用 Cloud Pak for Integration Keycloak 所提供的角色。 如果您想要提供自己的配置，請使用 `manual`。 如果您使用 Cloud Pak for Integration 授權，則預設為 `integration-keycloak`；如果您使用 MQ 授權，則預設為 `manual`。

.spec.web.manualConfig

提供 Web 伺服器 XML 配置的設定。 需要 MQ Operator 3.0.0 或更高版本。

出現在:

- 第 181 頁的『.spec.web』

欄位	說明
configMap ConfigMap	ConfigMap 代表包含 Web 伺服器 XML 配置的 Kubernetes ConfigMap 。
secret 密鑰	Secret 代表包含 Web 伺服器 XML 配置的 Kubernetes 密鑰。 使用「密鑰」可保護 Kubernetes 層中的任何認證，但監視或疑難排解工具可能會不安全地公開基礎檔案。 為了提高安全性，請使用 "securityUtility" 來編碼認證。

.spec.web.manualConfig.configMap

ConfigMap 代表包含 Web 伺服器 XML 配置的 Kubernetes ConfigMap 。

出現在:

- 第 181 頁的『.spec.web.manualConfig』

欄位	說明
name 字串	Kubernetes 來源的名稱。

.spec.web.manualConfig.secret

Secret 代表包含 Web 伺服器 XML 配置的 Kubernetes 密鑰。 使用「密鑰」可保護 Kubernetes 層中的任何認證，但監視或疑難排解工具可能會不安全地公開基礎檔案。 為了提高安全性，請使用 "securityUtility" 來編碼認證。

出現在:

- 第 181 頁的『.spec.web.manualConfig』

欄位	說明
name 字串	Kubernetes 來源的名稱。

.status

QueueManager 的觀察狀態。

出現在:

- 第 166 頁的『QueueManager』

欄位	說明
adminUiUrl 字串	管理使用者介面的 URL 。
availability 可用性	佇列管理程式的可用性狀態。
conditions QueueManagerStatusCondition 陣列	條件代表佇列管理程式狀態的最新可用觀察值。
endpoints 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。
metadata meta 資料	meta 資料代表「佇列管理程式」的其他資訊，包括整合-Keycloak 狀態。
name 字串	佇列管理程式的名稱。
phase 字串	佇列管理程式狀態的階段。

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

.status.availability

佇列管理程式的可用性狀態。

出現在:

- 第 182 頁的『.status』

欄位	說明
initialQuorumEstablished 布林值	是否已為 NativeHA 建立起始仲裁。

.status.conditions

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

出現在:

- 第 182 頁的『.status』

欄位	說明
lastTransitionTime 字串	前次條件從某個狀態轉移至另一個狀態的時間。
message 字串	人類可讀的訊息，指出前次轉移的詳細資料。
reason 字串	此狀態前次轉移的原因。
status 字串	狀況的狀態。
type 字串	條件類型。

.status.endpoints

QueueManagerStatusEndpoint 會定義 QueueManager 的端點。

出現在:

- 第 182 頁的『.status』

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。

.status.metadata

meta 資料代表「佇列管理程式」的其他資訊，包括整合-Keycloak 狀態。

出現在:

- 第 182 頁的『.status』

欄位	說明
integrationKeycloak IntegrationKeycloak	QueueManagerStatusIntegrationKeycloak 定義 QueueManager 的整合-Keycloak 狀態。

.status.metadata.integrationKeycloak

QueueManagerStatusIntegrationKeycloak 定義 QueueManager 的整合-Keycloak 狀態。

出現在:

- 第 183 頁的『.status.metadata』

欄位	說明
clientName 字串	

.status.versions

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 182 頁的『.status』

欄位	說明
available QueueManagerStatusVersion 可用	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。

.status.versions.available

IBM Entitled Registry 中提供其他版本的 MQ 。

出現在:

- 第 184 頁的『.status.versions』

欄位	說明
channels 陣列	可用於自動更新 MQ 版本的通道。
versions 版本 陣列	可用的 MQ 特定版本。

.status.versions.available.versions

QueueManagerStatusVersion 會定義 MQ 的版本。

出現在:

- 第 184 頁的『.status.versions.available』

欄位	說明
licenses Licenses 陣列	適用於此版本 QueueManager 的授權。
name 字串	此版本 QueueManager 的 name 版。 這些是 spec.version 欄位的有效值。

.status.versions.available.versions.licenses

QueueManagerStatusLicense 會定義授權。

出現在:

- 第 184 頁的『.status.versions.available.versions』

欄位	說明
displayName 字串	授權的顯示名稱。
link 字串	鏈結至授權內容。
matchesCurrentType 布林值	授權是否符合目前使用的授權類型。
name 字串	授權的名稱。

OpenShift CP4I *QueueManager* 的狀態條件 *(mq.ibm.com/v1beta1)*

status.conditions 欄位會更新，以反映 QueueManager 資源的狀況。 一般而言，狀況會說明異常狀況。 處於健全且備妥狀態的佇列管理程式沒有 **Error** 或 **Pending** 狀況。 它可能具有一些諮詢 **Warning** 條件。

IBM MQ Operator 1.2 中引進了對條件的支援。

下列是針對 QueueManager 資源所定義的條件:

表 *1:* 佇列管理程式狀態條件			
元件	**條件類型**	**原因碼**	**訊息警告**
QueueManager[9]	已封鎖	OperatorDependency	若要安裝，此實例需要 [IBM Cloud Pak for Integration] 配置 Keycloak 。 此實例將保持 [擱置] 狀態，直到在此 QueueManager 的 Cp4iServicesBinding 資源中，將 Keycloak 報告為 [KeycloakReady] 為止。
			若要安裝，此實例需要操作器 [IBM IAM]。 此實例將保持 [已封鎖] 狀態，直到 [IBM Cloud Pak 基礎服務] 安裝操作器為止。
	擱置	建立中	正在部署 MQ 佇列管理程式
	擱置	OidcPending	MQ 佇列管理程式正在等待 OIDC 用戶端登錄
	錯誤	失敗	MQ 佇列管理程式無法部署
	警告	UnsupportedVersion	[10] 運算子已安裝在 OCP *<ocp_version>*版上不受支援的運算元。 不支援此運算元。
	警告	CP4I-LTS 支援	[11] 已安裝 CP4I-LTS 運算元 *<mq_version>* ，但由不符合延伸支援持續時間資格的運算子所管理。 此運算元不適用於延伸支援持續時間。
	警告	CP4I-LTS 支援	[12] 已安裝 CP4I-LTS 運算元 *<mq_version>* ，但 OCP 第 4 版 *<ocp_version>* 不符合延伸支援期間的資格。 此運算元不適用於延伸支援持續時間。
	警告	CP4I-LTS 支援	[13] 已安裝 CP4I-LTS 運算元 *<mq_version>* ，但 OCP 版本 *<ocp_version>* 不適用於延伸支援期間。 根據一般 CD 版次支援此運算元。
Pod[14]	擱置	PodPending	正在部署 MQ 佇列管理程式的 Pod
	錯誤	PodFailed	正在部署 MQ 佇列管理程式的 Pod

[9] 條件 `Creating` 及 `Failed` 會監視佇列管理程式部署的整體進度。 如果您使用 IBM Cloud Pak for Integration 授權且已啟用 Web 主控台，則在等待 OIDC 用戶端登錄完成 IAM 時， `OidcPending` 條件會記載佇列管理程式的狀態。

[10] 運算子 1.4.0 以及更新版本

[11] 運算子 1.4.0 以及更新版本

[12] 運算子 1.4.0 以及更新版本

[13] 僅限運算子 1.3.0

[14] 在部署佇列管理程式期間， Pod 條件會監視 Pod 的狀態。 如果您看到任何 `PodFailed` 條件，則整體佇列管理程式條件也會設為 `Failed`。

表 1: 佇列管理程式狀態條件 (繼續)			
元件	**條件類型**	**原因碼**	**訊息警告**
儲存體 [15]	擱置	StoragePending	正在供應 MQ 佇列管理程式的儲存體
	警告	StorageEphemeral	將暫時儲存體用於正式作業 MQ 佇列管理程式
	錯誤	StorageFailed	無法供應 MQ 佇列管理程式的儲存體

Multi 建置您自己的 IBM MQ 容器及部署程式碼

開發自行建置的儲存器。 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，請考量是否可以改用 IBM MQ Operator。 請參閱 第 5 頁的『選擇您要如何在儲存器中使用 IBM MQ』

關於這項作業

程序

- 第 187 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 188 頁的『建置範例 IBM MQ 佇列管理程式儲存器映像檔』
- 第 190 頁的『在個別儲存器中執行本端連結應用程式』
- 檢閱 IBM MQ 範例 Helm 圖表。

Multi 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。

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。

15 儲存體狀況會監視建立持續性儲存體磁區之要求的進度 (StoragePending 狀況)，並報告往回連結錯誤及其他失敗。 如果在儲存體供應期間發生任何錯誤，則 StorageFailed 條件會新增至條件清單，且整體佇列管理程式條件也會設為 Failed。

容器安全

為了將執行時期安全需求降至最低，範例儲存器映像檔是使用 IBM MQ 可解壓縮安裝來安裝。 這可確保未設定任何 setuid 位元，且儲存器不需要使用專用權提升。 部分儲存器系統會定義您可以使用的使用者 ID，而不可壓縮的安裝不會對可用的作業系統使用者做出任何假設。

Multi 建置範例 IBM MQ 佇列管理程式儲存器映像檔

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 188 頁的『建置範例基本 IBM MQ 佇列管理程式映像檔』
  - 第 188 頁的『建置已配置的範例 IBM MQ 佇列管理程式映像檔』

Multi 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。
- 選擇性的: 如果您計劃使用 Red Hat OpenShift Container Platform "受限" 安全環境定義限制 (SCC) 來配置安全存取，請使用其中一個 IBM MQ 非安裝映像檔。

  下載這些映像檔的鏈結可在 IBM MQ 下載的 Containers 區段中找到。

結果

您現在具有已安裝 IBM MQ 的基本容器映像檔。

現在，您已準備好 建置已配置的範例 IBM MQ 佇列管理程式映像檔。

Multi 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，您可以使用一般映像檔作為母項來建立自己的容器映像檔層。

開始之前

此作業假設當您 建置範例基本 IBM MQ 佇列管理程式映像檔時，已使用 "No-Install" IBM MQ 套件。 否則，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。 依預設使用的 "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

傳統 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。 在現行版本的 IBM MQ 中，當您使用 "No-Install" IBM MQ 套件時，不再有 mqm 使用者，也沒有 mqm 群組。

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 QLOCAL(EXAMPLE.QUEUE.1) REPLACE
```

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
COPY config.mqsc /etc/mqm/
```

3. 使用下列指令來建置自訂容器映像檔:

```
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。

註: 在 Red Hat Enterprise Linux (RHEL) 上，您使用指令 **docker** (RHEL V7) 或 **podman** (RHEL V7 或 RHEL V8)。 在 Linux 上，您需要在指令開頭執行 **docker** 指令與 **sudo** ，以取得額外專用權。

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:

```
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 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

您已建置已配置的容器映像檔，並且可以使用 **docker ps** 指令來檢視執行中容器。 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

- 如果在使用 **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，也會列印此 ID。
- 您可以使用 **docker logs ${CONTAINER_ID}** 指令來檢視容器的日誌。

Multi 在個別儲存器中執行本端連結應用程式

透過在儲存器之間共用程序名稱空間，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

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。

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，在儲存器中建立名稱為 `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**、 **amqsput**、**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 191 頁的『對名稱空間應用程式進行疑難排解』 以取得相關資訊。

Multi 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 190 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
```

MQ Adv. 建立原生 HA 群組 (如果建立您自己的儲存器)

您必須建立、配置及啟動三個佇列管理程式，才能建立「原生 HA」群組。

關於這項作業

建立原生 HA 解決方案的建議方法是使用 IBM MQ 運算子 (請參閱 原生 HA)。 或者，如果您建立自己的儲存器，則可以遵循下列指示。

若要建立原生 HA 群組，您可以在三個節點上建立三個佇列管理程式，並將其日誌類型設為 `log replication`。 然後，您可以編輯每一個佇列管理程式的 `qm.ini` 檔案，以新增三個節點中每一個節點的連線詳細資料，以便它們可以彼此抄寫日誌資料。

然後，您必須啟動這三個佇列管理程式，以便它們可以檢查這三個實例是否可以彼此通訊，並決定其中哪些將是作用中實例，哪些將是抄本。

註: 如果您是執行 Kubernetes 或 Red Hat OpenShift，則只能以此方式在您自己的容器中建立原生 HA 群組。

程序

1. 在三個節點中的每一個節點上，建立佇列管理程式，指定日誌類型的日誌抄本，並提供每一個日誌實例的唯一名稱。 每一個佇列管理程式都具有相同的名稱:

   ```
   crtmqm -lr instance_name qmname
   ```

   例如:

   ```
   node 1> crtmqm -lr qm1_inst1 qm1

   node 2> crtmqm -lr qm1_inst2 qm1

   node 3> crtmqm -lr qm1_inst3 qm1
   ```

2. 順利建立每一個佇列管理程式時，會將另一個名為 `NativeHALocalInstance` 的段落新增至佇列管理程式配置檔 `qm.ini`。 `Name` 屬性會新增至指定所提供實例名稱的段落。

   您可以選擇性地將下列屬性新增至 `qm.ini` 檔中的 `NativeHALocalInstance` 段落:

   KeyRepository
   金鑰儲存庫的位置，該金鑰儲存庫保留用於保護日誌抄寫資料流量的數位憑證。 位置以詞幹格式提供，亦即，它包括不含副檔名的完整路徑和檔名。 如果省略 `KeyRepository` 段落屬性，則會以純文字在實例之間交換日誌抄寫資料。

   CertificateLabel
   憑證標籤，識別用於保護日誌抄寫資料流量的數位憑證。 如果提供 `KeyRepository` ，但省略 `CertificateLabel` ，則會使用預設值 `ibmwebspheremq`*`queue_manager`* 。

   CipherSpec
   用來保護日誌抄寫資料流量的 MQ CipherSpec 。 如果提供此段落屬性，則也必須提供 `KeyRepository` 。 如果提供 `KeyRepository` ，但省略 `CipherSpec` ，則會使用預設值 `ANY` 。

LocalAddress
接受日誌抄寫資料流量的本端網路介面位址。 如果提供此段落屬性，則會使用 "[addr] [(port)]" 格式來識別本端網路介面及/或埠。 網址可以指定為主機名稱、 IPv4 帶點十進位或 IPv6 十六進位格式。 如果省略此屬性，佇列管理程式會嘗試連結至所有網路介面，它會使用 ReplicationAddress 中符合本端實例名稱之 NativeHAInstances 段落中指定的埠。

HeartbeatInterval
活動訊號間隔定義原生 HA 佇列管理程式的作用中實例傳送網路活動訊號的頻率 (毫秒)。 活動訊號間隔值的有效範圍是 500 (0.5 秒) 至 60000 (1 分鐘) ，超出此範圍的值會導致佇列管理程式無法啟動。 如果省略此屬性，則會使用預設值 5000 (5 秒)。 每一個實例都必須使用相同的活動訊號間隔。

HeartbeatTimeout
活動訊號逾時值定義原生 HA 佇列管理程式的抄本實例在決定作用中實例無回應之前等待的時間。 活動訊號間隔逾時值的有效範圍是 500 (0.5 秒) 至 120000 (2 分鐘)。 活動訊號逾時值必須大於或等於活動訊號間隔。

無效值會導致佇列管理程式無法啟動。 如果省略此屬性，則抄本會等待 2 x HeartbeatInterval ，然後再啟動處理程序來選取新的作用中實例。 每一個實例都必須使用相同的活動訊號逾時。

RetryInterval
重試間隔定義原生 HA 佇列管理程式應該重試失敗抄寫鏈結的頻率 (毫秒)。 重試間隔的有效範圍是 500 (0.5 秒) 至 120000 (2 分鐘)。 如果省略此屬性，在重試失敗的抄寫鏈結之前，抄本會等待 2 x HeartbeatInterval 。

3. 編輯每一個佇列管理程式的 qm.ini 檔，並新增連線詳細資料。 您可以新增三個 NativeHAInstance 段落，「原生 HA」群組 (包括本端實例) 中的每一個佇列管理程式實例各一個。 新增下列屬性:

   名稱
   指定您建立佇列管理程式實例時所使用的實例名稱。

   ReplicationAddress
   指定實例的主機名稱 IPv4 帶點十進位或 IPv6 十六進位格式位址。 您可以將位址指定為主機名稱、IPv4 帶點十進位或 IPv6 十六進位格式位址。 抄寫位址必須可解析且可從群組中的每一個實例遞送。 用於日誌抄寫的埠號必須以方括弧 ([]) 指定，例如:

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

   註: NativeHAInstance 段落在每個實例上都相同，且可以使用自動配置 (**crtmqm -ii**) 來提供。

4. 啟動三個實例中的每一個:

   ```
   strmqm QMgrName
   ```

   當實例啟動時，它們會進行通訊以檢查這三個實例是否都在執行中，然後決定這三個實例中的哪一個是作用中實例，而另外兩個實例則繼續以抄本形式執行。

範例

下列範例顯示 qm.ini 檔案的區段，其中指定三個實例之一的必要「原生 HA」詳細資料:

```
NativeHALocalInstance:
  LocalName=node-1

NativeHAInstance:
  Name=node-1
  ReplicationAddress=host1.example.com(4444)
NativeHAInstance:
  Name=node-2
  ReplicationAddress=host2.example.com(4444)
NativeHAInstance:
  Name=node-3
  ReplicationAddress=host3.example.com(4444)
```

MQ Adv. 執行您自己的原生 HA 佇列管理程式漸進式更新的考量

對原生 HA 佇列管理程式的 IBM MQ 版本或 Pod 規格進行任何更新都需要您執行佇列管理程式實例的漸進式更新。 IBM MQ Operator 會自動為您處理此問題，但如果您要建置自己的部署程式碼，則有一些重要考量。

註: 範例 Helm Chart 包含 Shell Script 來執行漸進式更新，但 Script **不** 適合正式作業使用，因為它未解決本主題中的考量。

Kubernetes 在 Kubernetes 中， `StatefulSet` 資源用來管理依序啟動及漸進式更新。 啟動程序的一部分是個別啟動每一個 Pod，等待它備妥，然後移至下一個 Pod。 這不適用於原生 HA，因為所有 Pod 都需要啟動，才能進行領導者選舉。 因此， `StatefulSet` 上的 `.spec.podManagementPolicy` 欄位需要設為 `Parallel`。 這也表示所有 Pod 也會平行更新，這尤其不需要。 因此， `StatefulSet` 也應該使用 `OnDelete` 更新策略。

無法使用 `StatefulSet` 漸進式更新程式碼會導致需要自訂漸進式更新程式碼，這應該考量下列各項:

- 一般漸進式更新程序
- 以最佳順序更新 Pod 以將關閉時間縮至最短
- 處理叢集狀態中的變更
- 處理錯誤
- 處理計時問題

一般漸進式更新程序

漸進式更新程式碼應該等待每一個實例從 `dspmq` 顯示 `REPLICA` 狀態。 這表示實例已執行某個層次的啟動 (例如，容器已啟動，且 MQ 處理程序正在執行中)，但它還不一定能夠與其他實例交談。 例如: Pod A 會重新啟動，只要它處於 `REPLICA` 狀態，就會重新啟動 Pod B。 一旦 Pod B 以新配置開始，它應該能夠與 Pod A 交談，並且可以形成仲裁， A 或 B 將變成新的作用中實例。

在此過程中，在每一個 Pod 都達到 `REPLICA` 狀態之後，如果要讓它連接至其對等節點並建立仲裁，則延遲很有用。

以最佳順序更新 Pod 以將關閉時間縮至最短

漸進式更新程式碼應該一次刪除一個 Pod，從處於已知錯誤狀態的 Pod 開始，接著是任何未順利啟動的 Pod。 作用中佇列管理程式 Pod 通常應該最後更新。

如果前次更新導致 Pod 進入已知錯誤狀態，則暫停刪除 Pod 也很重要。 這可防止跨所有 Pod 部署中斷的更新。 例如，如果 Pod 更新為使用無法存取 (或包含拼字錯誤) 的新容器映像檔，則會發生此情況。

處理叢集狀態中的變更

漸進式更新程式碼需要適當地回應叢集狀態中的即時變更。 例如，由於「節點」重新開機或「節點」壓力，可能會收回其中一個佇列管理程式的 Pod。 如果叢集忙碌，可能不會立即重新排定收回的 Pod。 在此情況下，在重新啟動任何其他 Pod 之前，漸進式更新程式碼需要適當地等待。

處理錯誤

當呼叫 Kubernetes API 及其他非預期的叢集行為時，漸進式更新程式碼必須健全而不會失敗。

此外，漸進式更新程式碼本身需要容忍重新啟動。 漸進式更新可能長時間執行，且可能需要重新啟動程式碼。

處理計時問題

漸進式更新程式碼需要檢查 Pod 的更新修訂，以便它可以確保 Pod 已重新啟動。 這可避免 Pod 可能指出其「已啟動」但實際上尚未終止的計時問題。

相關概念

第 5 頁的『選擇您要如何在儲存器中使用 IBM MQ』

在容器中使用 IBM MQ 有多個選項: 您可以選擇使用 IBM MQ Operator，它會使用預先包裝的容器映像檔，或者您可以建置自己的映像檔及部署程式碼。

MQ Adv. 檢視自訂建置儲存器的原生 HA 佇列管理程式狀態

對於自訂建置的儲存器，您可以使用 **dspmq** 指令來檢視原生 HA 實例的狀態。

關於這項作業

您可以使用 **dspmq** 指令來檢視節點上佇列管理程式實例的作業狀態。 傳回的資訊視實例是作用中還是抄本而定。 作用中實例所提供的資訊是明確的，抄本節點的資訊可能已過期。

您可以執行下列動作:

- 檢視現行節點上的佇列管理程式實例是作用中還是抄本。
- 檢視現行節點上實例的原生 HA 作業狀態。
- 檢視原生 HA 配置中所有三個實例的作業狀態。

下列狀態欄位用來報告原生 HA 配置狀態:

角色
指定實例的現行角色，並且是 `Active`、 `Replica` 或 `Unknown` 之一。

實例
使用 **crtmqm** 指令的 **-lr** 選項建立此佇列管理程式實例時，為其提供的名稱。

INSYNC
指出實例是否可以在必要時接管作為作用中實例。

仲裁
以 *number_of_instances_in-sync*/*number_of_instances_configured* 格式報告仲裁狀態。

REPLADDR
佇列管理程式實例的抄寫位址。

CONNACTV
指出節點是否連接至作用中實例。

BACKLOG
指出實例落後的 KB 數。

CONNINST
指出指定的實例是否連接至此實例。

ALTDATE
指出前次更新此資訊的日期 (如果從未更新過，則為空白)。

ALTTIME
指出前次更新此資訊的時間 (如果從未更新過，則為空白)。

程序

- 若要判定佇列管理程式實例是作為作用中實例還是抄本執行，請執行下列動作:

  ```
  dspmq -o status -m QMgrName
  ```

  名為 BOB 之佇列管理程式的作用中實例會報告下列狀態:

  ```
  QMNAME(BOB)                   STATUS(Running)
  ```

  名為 BOB 之佇列管理程式的抄本實例會報告下列狀態:

  ```
  QMNAME(BOB)                   STATUS(Replica)
  ```

  非作用中實例會報告下列狀態:

```
QMNAME(BOB)                 STATUS(Ended Immediately)
```

- 若要判定現行節點上實例的原生 HA 作業狀態，請執行下列動作:

  ```
  dspmq -o nativeha -m QMgrName
  ```

  名為 BOB 之佇列管理程式的作用中實例可能報告下列狀態:

  ```
  QMNAME(BOB)               ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
  ```

  佇列管理程式 BOB 的抄本實例可能會報告下列狀態:

  ```
  QMNAME(BOB)               ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
  ```

  名為 BOB 之佇列管理程式的非作用中實例可能會報告下列狀態:

  ```
  QMNAME(BOB)               ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
  ```

- 若要判定「原生 HA」配置中所有實例的「原生 HA」作業狀態，請執行下列動作:

  ```
  dspmq -o nativeha -x -m QMgrName
  ```

  如果您在執行佇列管理程式 BOB 作用中實例的節點上發出此指令，則可能會收到下列狀態:

  ```
  QMNAME(BOB)            ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
   INSTANCE(inst1) ROLE(Active)  REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
  CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
   INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
  CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
   INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
  CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
  ```

  如果您在執行佇列管理程式 BOB 抄本實例的節點上發出此指令，則可能會收到下列狀態，指出其中一個抄本落後:

  ```
  QMNAME(BOB)            ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
   INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
  CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
   INSTANCE(inst1) ROLE(Active)  REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
  CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
   INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(No)  BACKLOG(435)
  CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
  ```

  如果您在執行佇列管理程式 BOB 非作用中實例的節點上發出此指令，則可能會收到下列狀態:

  ```
  QMNAME(BOB)            ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
   INSTANCE(inst1) ROLE(Unknown) REPLADDR(9.20.123.45) CONNACTV(Unknown) INSYNC(Unknown)
  BACKLOG(Unknown) CONNINST(No) ALTDATE() ALTTIME()
   INSTANCE(inst2) ROLE(Unknown) REPLADDR(9.20.123.46) CONNACTV(Unknown) INSYNC(Unknown)
  BACKLOG(Unknown) CONNINST(No) ALTDATE() ALTTIME()
   INSTANCE(inst3) ROLE(Unknown) REPLADDR(9.20.123.47) CONNACTV(No) INSYNC(Unknown)
  BACKLOG(Unknown) CONNINST(No) ALTDATE() ALTTIME()
  ```

  如果您在實例仍在協議作用中及抄本時發出指令，則會收到下列狀態:

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

相關參考
dspmq (顯示佇列管理程式) 指令

MQ Adv. 正在結束原生 HA 佇列管理程式

您可以使用 **endmqm** 指令來結束屬於「原生 HA」群組的作用中或抄本佇列管理程式。

程序

- 若要結束佇列管理程式的作用中實例，請參閱本文件「配置」一節中的 結束原生 HA 佇列管理程式 。

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。

在其他國家或地區中，IBM 可能未提供本文件所提及的各項產品、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。 本文件在提及 IBM 產品、程式或服務時，不表示或暗示只能使用 IBM 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 之產品、程式或服務，使用者必須自行負責作業之評估和驗證責任。

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：

Intellectual Property Licensing
Legal and Intellectual Property Law
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。

本資訊中可能會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。 同時，IBM 得隨時改進並（或）變動本書中所提及的產品及（或）程式。

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，在其他作業環境下取得的結果可能大不相同。 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 未來動向的任何陳述，僅代表 IBM 的目標而已，並可能於未事先聲明的情況下有所變動或撤回。

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。 所有這些名稱都是虛構的，如有任何類似實際企業所用的名稱及地址之處，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 不保證或暗示這些程式的可靠性、服務性或功能。

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 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 中找到。 其他產品及服務名稱可能是 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 (https://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: