

9.3

コンテナ内の *IBM MQ*

IBM

注記

本書および本書で紹介する製品をご使用になる前に、[221 ページの『特記事項』](#)に記載されている情報をお読みください。

本書は、IBM® MQ バージョン 9 リリース 3、および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

お客様が IBM に情報を送信する場合、お客様は IBM に対し、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で情報を使用または配布する非独占的な権利を付与します。

© Copyright International Business Machines Corporation 2007 年, 2024.

目次


コンテナ内の IBM MQ と IBM Cloud Pak for Integration.....	5
コンテナ内の IBM MQ の計画.....	5
コンテナ内の IBM MQ の使用方法の選択.....	5
コンテナでの IBM MQ のサポート.....	6
コンテナでの IBM MQ のライセンス交付の計画.....	14
IBM MQ Operator の依存関係.....	19
IBM MQ Operator に必要なクラスター・スコープ許可.....	20
IBM MQ Operator のストレージに関する考慮事項.....	21
IBM MQ Advanced for Developers コンテナ・イメージ.....	23
コンテナ内の IBM MQ の高可用性.....	25
コンテナ内の IBM MQ の災害復旧.....	27
コンテナ内の IBM MQ を保護するための計画.....	28
コンテナ内の IBM MQ のスケーラビリティとパフォーマンスの計画.....	33
IBM MQ Operator の使用.....	34
IBM MQ Operator のリリース履歴.....	35
イメージ署名の検証.....	79
IBM Cloud Pak for Integration への IBM MQ のマイグレーション.....	79
IBM MQ Operator のインストール.....	103
エア・ギャップ環境での IBM MQ Operator 2.x のインストール.....	110
Red Hat OpenShift Container Platform クラスターにキュー・マネージャーをデプロイする方法.....	116
IBM MQ Operator のアンインストール.....	119
IBM MQ Operator とキュー・マネージャーのアップグレード.....	121
IBM MQ Operator を使用したキュー・マネージャーの構成.....	135
IBM MQ Operator を使用した IBM MQ.....	172
IBM MQ Operator に関する問題のトラブルシューティング.....	180
IBM MQ Operator の API リファレンス.....	183
独自の IBM MQ コンテナおよびデプロイメント・コードのビルド.....	208
コンテナを使用する独自の IBM MQ キュー・マネージャー・イメージの計画.....	208
サンプルの IBM MQ キュー・マネージャーのコンテナ・イメージのビルド.....	209
別々のコンテナでのローカル・バインディング・アプリケーションの実行.....	212
独自のコンテナを作成する場合のネイティブ HA グループの作成.....	214
特記事項.....	221
プログラミング・インターフェース情報.....	222
商標.....	222

Multi コンテナ内の IBM MQ と IBM Cloud Pak for Integration

コンテナを使用すると、IBM MQ キュー・マネージャーや IBM MQ クライアント・アプリケーションをすべての依存関係とともに、ソフトウェア開発のために標準化された単位でパッケージ化できます。

Red Hat® OpenShift®で IBM MQ Operator を使用して IBM MQ を実行できます。このテストは、IBM Cloud Pak® for Integration、IBM MQ Advanced、または IBM MQ Advanced for Developers を使用して行うことができます。

また、IBM MQ をお客様が作成した独自のコンテナで実行することもできます。

  IBM MQ Operator について詳しくは、以下のリンクを参照してください。

Multi コンテナ内の IBM MQ の計画

コンテナ内の IBM MQ の計画を立てるときには、高可用性の実現方法、キュー・マネージャーの保護方法など、さまざまなアーキテクチャー・オプションのために IBM MQ が提供しているサポートについて考慮してください。

このタスクについて

コンテナ・アーキテクチャーで IBM MQ を計画する前に、基本的な IBM MQ 概念 (IBM MQ 技術の概要を参照)、および基本的な Kubernetes/Red Hat OpenShift の概念を理解しておく必要があります ([OpenShift Container Platform のアーキテクチャー](#) を参照してください)。

手順

- [5 ページの『コンテナ内の IBM MQ の使用方法の選択』](#)。
- [6 ページの『コンテナでの IBM MQ のサポート』](#)。
- [21 ページの『IBM MQ Operator のストレージに関する考慮事項』](#)。
- [25 ページの『コンテナ内の IBM MQ の高可用性』](#)。
- [27 ページの『コンテナ内の IBM MQ の災害復旧』](#)。
- [28 ページの『コンテナ内の IBM MQ でのユーザーの認証と許可』](#)。

コンテナ内の IBM MQ の使用方法の選択

コンテナ内の IBM MQ の使用方法としては、いくつかの選択肢があります。プリパッケージされているコンテナ・イメージを使用する IBM MQ Operator を使用することも、独自のイメージとデプロイメント・コードをビルドすることもできます。

IBM MQ Operator の使用

Red Hat OpenShift Container Platform にデプロイする場合は、おそらく、IBM MQ Operator を使用したいはずです。

IBM MQ Operator は、Red Hat OpenShift Container Platform API を拡張して、新しい QueueManager カスタム・リソースを追加します。オペレーターは、新しいキュー・マネージャー定義を監視し、それを StatefulSet リソースや Service リソースなどの必要な下位リソースに変換します。Native HA の場合、オペレーターはキュー・マネージャー・インスタンスの複雑なローリング更新を実行することもできます。以下を参照してください。 [216 ページの『ネイティブ HA キュー・マネージャーの独自ローリング更新を実行する場合の考慮事項』](#)

IBM MQ の機能の中には、IBM MQ Operator を使用する場合にはサポートされないものもあります。IBM MQ Operator の使用時にサポートされる内容については、[6 ページの『コンテナでの IBM MQ のサポート』](#)を参照してください。

IBM MQ Operator は、マルチアーキテクチャー・コンピュート・マシンを使用する OpenShift クラスタへのインストールをサポートしていないことに注意してください。

独自のイメージおよびデプロイメント・コードのビルド

Multi

これは最も柔軟なコンテナソリューションですが、コンテナの設定に強いスキルが必要であり、結果としてのコンテナを"所有する"必要があります。Red Hat OpenShift Container Platform を使用しない場合は、独自のイメージとデプロイメント・コードをビルドする必要があります。

独自のイメージをビルドするためのサンプルが用意されています。[208 ページの『独自の IBM MQ コンテナおよびデプロイメント・コードのビルド』](#)を参照してください。

独自のイメージおよびデプロイメント・コードのビルド時にサポートされる内容については、[6 ページの『コンテナでの IBM MQ のサポート』](#)を参照してください。

関連資料

[6 ページの『コンテナでの IBM MQ のサポート』](#)

コンテナでは、すべての IBM MQ 機能が同じ方法で使用可能およびサポートされているわけではありません。

OpenShift CP4I CD CP4I-LTS コンテナでの IBM MQ のサポート

コンテナでは、すべての IBM MQ 機能が同じ方法で使用可能およびサポートされているわけではありません。

以下の表は、IBM MQ Operator で IBM MQ 機能がどのようにサポートされているか、または独自のコンテナとデプロイメント・コードをビルドするときどのようにサポートされているかを示しています。

注：IBM Container Registry (icr.io および cp.icr.io) 上の事前作成された IBM MQ コンテナ・イメージは、IBM MQ Operator で使用される場合にのみサポートされ、フィックスに適格です。

事前作成された IBM MQ Advanced for Developers イメージのライセンスを別のライセンスに「アップグレード」することはできません。IBM MQ Operator は、選択されているライセンスに応じて異なるイメージをデプロイします。

この表では、以下の用語が適用されます。

"コンテナ・イネーブルメント・コード"

実行可能ファイル `runmqserver`、`runmqintegrationserver`、`chkmqhealthy`、`chkmqready` および `chkmqstarted`。このコードはサンプルとして提供されており、IBM MQ Operator で使用される場合に事前作成コンテナの一部としてのみサポートされます。

	IBM MQ Operator および IBM Cloud Pak for Integration ライセンスの使用	IBM MQ Operator および IBM MQ Advanced ライセンスの使用	IBM MQ Operator および IBM MQ Advanced for Developers ライセンスの使用	事前作成 IBM MQ Advanced for Developers イメージ	Build-your-own コンテナ
サポートされるプラットフォーム	Red Hat OpenShift Container Platform でのみサポート。Red Hat OpenShift Container Platform のリリースは、Red Hat がサポートを停止すると IBM MQ でサポートされなくなります。 詳しくは、11 ページの『 IBM MQ Operator のバージョン・サポート 』を参照してください。		Red Hat OpenShift Container Platform でのみ使用可能ですが、サポートされていません。	すべての Docker、containerd、または cri-o プラットフォームで機能しますが、サポートされません。詳細は IBM MQ のシステム要件 を参照してください。	任意の Docker、containerd または cri-o プラットフォーム。詳細は IBM MQ のシステム要件 を参照してください。ネイティブ HA は、Kubernetes または Red Hat OpenShift Container Platform でのみサポートされます。サンプル・コンテナ・イメージは、Red Hat Universal Base Image (UBI) を使用します。これには、IBM MQ で使用される Linux® ライブラリーとユーティリティーが含まれています。UBI は、Red Hat OpenShift での実行時に Red Hat によってサポートされます。コンテナ・イネーブルメント・コードはサポートされていません。
CPU アーキテクチャー	amd64、s390x z/Linux、および ppc64le Power® Systems でサポートされます。		amd64、s390x z/Linux、および ppc64le Power Systems で使用可能ですが、サポートされていません。		IBM MQ ソフトウェアに従います。

	IBM MQ Operator および IBM Cloud Pak for Integration ライセンスの使用	IBM MQ Operator および IBM MQ Advanced ライセンスの使用	IBM MQ Operator および IBM MQ Advanced for Developers ライセンスの使用	事前作成 IBM MQ Advanced for Developers イメージ	Build-your-own コンテナ
サポート期間	<p>IBM Cloud Pak for Integration - Long Term Support または Continuous Delivery。¹</p> <p>CD オペレーターおよびキュー・マネージャーは、次の IBM Cloud Pak for Integration CD または CP4I-LTS リリースまでサポートされます。</p> <p>CP4I-LTS オペレーターおよびキュー・マネージャーは、次の IBM Cloud Pak for Integration CP4I-LTS リリースまで、およびアップグレードを許可する猶予期間までサポートされます。</p>	<p>Continuous Delivery ストリームのみ (IBM MQ Operator と キュー・マネージャーの両方)。</p> <p>各 IBM MQ Operator およびキュー・マネージャーのバージョンは、次の CD または LTS リリースまでのみサポートされます。</p>	サポート対象外		<p>IBM MQ ソフトウェアに従います。<u>長期サポートおよび継続的デリバリーのリリースに関する IBM MQ の FAQ</u> を参照してください。コンテナ・イメージ・コードはサポートされていません。</p>

¹ IBM MQ Operator は、IBM MQ CD リリースとして、または CP4I-LTS リリースとしてサポートされます。

- IBM MQ Operator 2.0.x を使用してデプロイされた IBM MQ 9.3.0.x コンテナ・イメージは、IBM Cloud Pak for Integration 2022.2.1 の一部として使用される場合、CP4I-LTS サポートの対象となります。IBM MQ Operator の最新の Long Term Support (LTS) リリースは 2.0.23 であり、最新の LTS コンテナ・イメージは 9.3.0.17-r3 です。
- IBM MQ Operator 3.1.x を使用してデプロイされた IBM MQ 9.3.5 コンテナ・イメージは、IBM Cloud Pak for Integration 2023.4.1 の一部として使用される場合、CD サポートの対象となります。IBM MQ Operator の最新の Continuous Delivery (CD) リリースは 3.1.3 であり、最新の CD コンテナ・イメージは 9.3.5.1-r2 です。

	IBM MQ Operator および IBM Cloud Pak for Integration ライセンスの使用	IBM MQ Operator および IBM MQ Advanced ライセンスの使用	IBM MQ Operator および IBM MQ Advanced for Developers ライセンスの使用	事前作成 IBM MQ Advanced for Developers イメージ	Build-your-own コンテナ
セキュリティ・フィックスの可用性	IBM Container Registry でコンテナ・イメージとして使用可能な定期フィックス				IBM MQ ソフトウェアに対するフィックスは、 Fix Central 上でソフトウェアとして入手できます。コンテナ・イネーブルメント・コードはサポートされていません。
暫定修正の可用性	ソフトウェアとして使用可能なキュー・マネージャーのフィックス、およびカスタム・イメージのビルドが必要です。 IBM MQ Operator フィックスは、インテリム・フィックスとしては使用できません。	使用可能なインテリム・フィックスがありません。			IBM MQ ソフトウェアに対するフィックスは、 Fix Central または IBM サポートからソフトウェアとして入手できます。コンテナ・イネーブルメント・コードはサポートされていません。
機能: Advanced Message Security	サポート対象。サーバー・サイドの暗号化を使用することは容易ではないことに注意してください。これは、IBM MQ Operator では、Advanced Message Security に独自の鍵ストアを直接指定することはできないためです。		使用可能ですが、サポートされていません。		IBM MQ ソフトウェアによってサポートされますが、使用可能なサンプルはありません。
機能: Managed File Transfer	使用不可で、サポートされていません。ただし、IBM MQ Operator を使用して、1 つ以上の調整キュー・マネージャー、コマンド・キュー・マネージャー、またはエージェント・キュー・マネージャーを提供することができます。		使用不可で、サポートされていません。		エージェント用のサンプルを使用して、IBM MQ ソフトウェアに従ってサポートされます。
機能: MQTT	使用不可で、サポートされていません。				IBM MQ ソフトウェアによってサポートされますが、使用可能なサンプルはありません。

	IBM MQ Operator および IBM Cloud Pak for Integration ライセンスの使用	IBM MQ Operator および IBM MQ Advanced ライセンスの使用	IBM MQ Operator および IBM MQ Advanced for Developers ライセンスの使用	事前作成 IBM MQ Advanced for Developers イメージ	Build-your-own コンテナ
機能: AMQP	使用不可で、サポートされていません。				IBM MQ ソフトウェアによってサポートされますが、使用可能なサンプルはありません。
機能: REST API	IBM MQ Operator 3.0 および IBM MQ 9.3.4 以降で使用可能であり、サポートされています。それより前は、REST API はサポートされていませんでした。	使用可能で、サポートされています。IBM MQ Operator 3.0 および IBM MQ 9.3.4 以降から簡単に構成できます。	IBM MQ Operator 3.0 および IBM MQ 9.3.4 以降で使用可能で、サポートされていますが、サポートされていません。それ以前は、REST API は使用できませんでした。	IBM MQ 9.3.4 以降で使用でき、サポートされていますが、サポートされていません。それより前は、REST API は使用できませんでした。	IBM MQ ソフトウェアによって提供され、サポートされます。
機能: 複製データ・キュー・マネージャー	使用不可で、サポートされていません。複製データ・キュー・マネージャーは、Linux カーネルと緊密に結合されており、コンテナではサポートされていません。				
機能: ネイティブ HA	使用可能で、サポートされています。		使用可能ですが、サポートされていません。		Kubernetes および Red Hat OpenShift Container Platform でのみ使用可能です。IBM MQ ソフトウェアに従ってサポートされます。
機能: 複数インスタンス・キュー・マネージャー	使用可能で、サポートされています。		使用可能ですが、サポートされていません。		IBM MQ ソフトウェアによって提供され、サポートされます。
機能: リカバリー・ログのタイプ	循環ログまたは複製ログのみ。リニア・ログはサポートされていません。				IBM MQ ソフトウェアによって提供され、サポートされます。 crtmqm オプションを構成する必要があります。

	IBM MQ Operator および IBM Cloud Pak for Integration ライセンスの使用	IBM MQ Operator および IBM MQ Advanced ライセンスの使用	IBM MQ Operator および IBM MQ Advanced for Developers ライセンスの使用	事前作成 IBM MQ Advanced for Developers イメージ	Build-your-own コンテナ
機能: crtmqdir 、 crtmqm 、 strmqm 、および endmqm のカスタム・コマンド行オプションの指定	使用不可で、サポートされていません。ほとんどのオプションは INI ファイルを使用して構成できますが、リニア・ロギングの使用など、一部のオプションは構成できません。				オプション。コンテナ使用可能化コードの実装方法によって異なります。
機能: オペレーティング・システム (OS) ユーザー	使用不可で、サポートされていません。				RPM を使用して IBM MQ をインストールしたが、使用可能なサンプルがない場合は、IBM MQ ソフトウェアによって可能となり、サポートされます。セキュリティ・リスクのため、推奨されません。
機能: IBM MQ Bridge to blockchain	使用不可で、サポートされていません。IBM MQ 9.3.2 以降では、IBM MQ から完全に削除されています。				
機能: IBM MQ Bridge to Salesforce	使用不可で、サポートされていません。				IBM MQ ソフトウェアに従ってサポートされますが、IBM MQ 9.3.1 以降では非推奨です。

注: 「supported as per IBM MQ software」という語句は、IBM 技術サポートが、コンテナ内で実行されているコア IBM MQ ソフトウェアに制限されていることを意味します。

関連概念

[IBM MQ FAQ for Long Term Support and Continuous Delivery リリース](#)

関連資料

[IBM Cloud Pak for Integration ソフトウェア・サポート・ライフサイクルの補足](#)

OpenShift > CP4I > CD > CP4I-LTS IBM MQ Operator のバージョン・サポート

IBM MQ、OpenShift Container Platform および IBM Cloud Pak for Integration のサポート対象バージョンの対応関係。

注:

IBM MQ Operator は、Extended Update Support (EUS) バージョンの OpenShift Container Platform のみをサポートします。これに含まれるリリースについて詳しくは、Red Hat OpenShift Container Platform Life Cycle Policy Web ページの「[Life Cycle Phases](#)」を参照してください。

- [12 ページの『使用可能な IBM MQ のバージョン』](#)
- [13 ページの『互換性のある Red Hat OpenShift Container Platform のバージョン』](#)
- [13 ページの『IBM Cloud Pak for Integration のバージョン』](#)
- [14 ページの『以前ので使用可能な IBM MQ バージョン』](#)
- [14 ページの『以前のオペレーターの互換性のある OpenShift Container Platform バージョン』](#)

使用可能な IBM MQ のバージョン

オペレーター・チャンネル	Operator バージョン	IBM MQ のバージョン									
		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	→	⚠	⚠	⚠	→	●				
v2.2	2.2	→	⚠	⚠	⚠	→	●				
v2.3	2.3	→	⚠	⚠	⚠	→	⚠	●			
v2.4	2.4	→	⚠	⚠	⚠	→	⚠	⚠	●		
v3.0	3.0					→	⚠	⚠	⚠	●	
v3.1	3.1					→	⚠	⚠	⚠	⚠	●

キー



使用可能な Continuous Delivery サポート



使用可能な IBM Cloud Pak for Integration - Long Term Support



IBM Cloud Pak for Integration - Long Term Support オペランドから Continuous Delivery オペランドへのマイグレーション時にのみ使用可能です。



Deprecated IBM MQ リリースはサポート対象外になるため、オペレーターで引き続き構成できる可能性があります。サポート対象ではなくなり、将来のリリースで削除される可能性があります。

各バージョンの詳細機能、変更点、フィックスなど、各バージョンの全詳細については、[35 ページの『IBM MQ Operator のリリース履歴』](#)を参照してください。

互換性のある Red Hat OpenShift Container Platform のバージョン

オペレーター・チャネル	Operator バージョン	OpenShift Container Platform のバージョン ²		
		4.10	4.12	4.14
v2.0	2.0.0-2.0.15	→	□	
	2.0.16		□	
	2.0.17 以降		□	□
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 以降		●	●
v3.0	3.0.0 以降		●	●
v3.1	3.1.0 以降		●	●

キー



使用可能な Continuous Delivery サポート



使用可能な IBM Cloud Pak for Integration - Long Term Support



現在はサポートされていません。新しい OpenShift Container Platform バージョンにマイグレーションしてください。

IBM Cloud Pak for Integration のバージョン

IBM Cloud Pak for Integration バージョン 2022.2.1 の一部として、または単独での使用がサポートされています。

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

IBM Cloud Pak for Integration バージョン 2022.4.1 の一部として、または単独での使用がサポートされています。

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

IBM Cloud Pak for Integration バージョン 2023.2.1 の一部として使用することも、単独で使用することもできます。

- IBM MQ Operator 2.4.x

IBM Cloud Pak for Integration バージョン 2023.4.1 の一部として、または単独での使用がサポートされます。

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

² OpenShift Container Platform のバージョンは、それぞれのサポート日に影響を受けます。詳しくは、[OpenShift Container Platform ライフ・サイクル・ポリシー](#) を参照してください。

以前ので使用可能な IBM MQ バージョン

IBM MQ 9.2 資料の [使用可能な IBM MQ バージョン](#) を参照してください。

以前のオペレーターの互換性のある OpenShift Container Platform バージョン

IBM MQ 9.2 資料の [互換性のある OpenShift Container Platform バージョン](#) を参照してください。

コンテナでの IBM MQ のライセンス交付の計画

コンテナ・ライセンスにより、コンテナが実行されているサーバー全体のライセンス交付を必要とするのではなく、個々の IBM MQ コンテナの使用可能な容量のみのライセンス交付を受けることができます。コンテナ・ライセンスを利用するには、IBM License Service を使用してライセンス使用状況を追跡し、必要なライセンスを判別する必要があります。

関連情報

[IBM コンテナ・ライセンス](#)

[コンテナのライセンス交付に関する FAQ](#)

[ライセンス・サービスのインストール](#)

[ライセンス使用の表示および追跡](#)

独自の IBM MQ コンテナ・イメージを作成する時のライセンス・アノテーション

ライセンス・アノテーションを使用すると、基礎になっているマシンではなくコンテナで定義した制限に基づいて使用量を追跡管理できます。クライアントで特定のアノテーションを付けてコンテナをデプロイするための構成を行うと、IBM License Service はそのアノテーションに基づいて使用量を追跡管理します。

独自に作成した IBM MQ コンテナ・イメージをデプロイする場合は、ライセンス交付に関して以下の 2 つの一般的なアプローチがあります。

- コンテナを実行するマシン全体のライセンスを取得します。
- 関連する制限に基づいてコンテナのライセンスを取得します。

どちらのオプションもお客様が使用できます。詳細については、Passport Advantage® の「[IBM Container Licenses](#)」ページを参照してください。

コンテナの制限に基づいて IBM MQ コンテナのライセンスを取得する場合は、使用量を追跡管理するために IBM License Service をインストールする必要があります。サポートされている環境やインストール手順の詳細については、GitHub の [ibm-licensing-operator](#) のページを参照してください。

IBM MQ コンテナがデプロイされている Kubernetes クラスターに IBM License Service がインストールされ、ポッドのアノテーションを使用して使用量が追跡されます。そのためクライアントで、IBM License Service がその後使用する特定のアノテーションを付けてポッドをデプロイする必要があります。コンテナ内にデプロイされた資格および機能に基づいて、以下の 1 つ以上のアノテーションを使用します。

注：アノテーションの多くには、以下の行のいずれかまたは両方が含まれています。

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

アノテーションを使用する前に、以下の行を編集する必要があります。

- `productChargedContainers` の場合は、"All" を選択するか、コンテナの実際の名前を置き換える必要があります。
- `productMetric` の場合は、提供されている値のいずれかを選択する必要があります。

IBM MQ 製品ライセンスで使用するアノテーション

IBM MQ 製品ライセンスがある場合は、購入した使用する資格に一致する以下のアノテーションを選択します。

- [16 ページの『IBM MQ』](#)
- [16 ページの『IBM MQ 拡張』](#)
- [17 ページの『IBM MQ \(非実稼働環境\)』](#)
- [17 ページの『IBM MQ Advanced for Non-Production Environment \(非実稼働環境向け拡張\)』](#)
- [17 ページの『IBM MQ Advanced for Developers』](#)

IBM MQ マルチインスタンス高可用性構成で使用する IBM MQ アノテーションは、以下のとおりです。[15 ページの『高可用性構成の正しいアノテーションの選択』](#) も参照してください。

- [17 ページの『IBM MQ コンテナ・マルチインスタンス』](#)
- [17 ページの『IBM MQ 拡張コンテナ・マルチインスタンス』](#)
- [17 ページの『IBM MQ Container Multi Instance for Non-Production Environment』](#)
- [17 ページの『IBM MQ Advanced Container Multi Instance for Non-Production Environment』](#)

CP4I 製品ライセンスで使用するアノテーション

IBM Cloud Pak for Integration (CP4I) 資格がある場合は、購入して使用する資格に一致する以下の注釈を選択します。

- [18 ページの『IBM MQ with CP4I ライセンス』](#)
- [18 ページの『IBM MQ Advanced with CP4I ライセンス』](#)
- [18 ページの『IBM MQ for Non-Production Environment with CP4I の使用許諾』](#)
- [18 ページの『IBM MQ Advanced for Non-Production Environment with CP4I の使用許諾』](#)

IBM MQ マルチインスタンス高可用性構成で使用する CP4I アノテーションは、以下のとおりです。[15 ページの『高可用性構成の正しいアノテーションの選択』](#) も参照してください。

- [18 ページの『IBM MQ Container Multi Instance with CP4I のライセンス』](#)
- [18 ページの『IBM MQ Advanced Container Multi Instance with CP4I のライセンス』](#)
- [19 ページの『IBM MQ Container Multi Instance for Non-Production Environment with CP4I の使用許諾』](#)
- [19 ページの『IBM MQ Advanced Container Multi Instance for Non-Production Environment with CP4I のライセンス』](#)

高可用性構成の正しいアノテーションの選択

IBM MQ マルチインスタンス

IBM MQ 複数インスタンスの高可用性構成でキュー・マネージャーのペアをデプロイする場合は、両方のインスタンスで同じアノテーションを使用する必要があります。購入したライセンスに応じて、以下のいずれかのアノテーションを選択する必要があります。

- IBM MQ または IBM MQ Advanced スタンドアロン・ライセンス
 - [17 ページの『IBM MQ コンテナ・マルチインスタンス』](#)
 - [17 ページの『IBM MQ 拡張コンテナ・マルチインスタンス』](#)
 - [17 ページの『IBM MQ Container Multi Instance for Non-Production Environment』](#)
 - [17 ページの『IBM MQ Advanced Container Multi Instance for Non-Production Environment』](#)
- IBM Cloud Pak for Integration 資格
 - [18 ページの『IBM MQ Container Multi Instance with CP4I のライセンス』](#)
 - [18 ページの『IBM MQ Advanced Container Multi Instance with CP4I のライセンス』](#)
 - [19 ページの『IBM MQ Container Multi Instance for Non-Production Environment with CP4I の使用許諾』](#)

- [19 ページの『IBM MQ Advanced Container Multi Instance for Non-Production Environment with CP4I のライセンス』](#)

IBM Cloud Pak for Integration 資格で使用する場合、アノテーション内の資格比率により、正しい資格使用量が記録されます。スタンドアロンの IBM MQ または IBM MQ Advanced の資格で使用する場合は、以下のように、インスタンスごとに License Service で報告されるアノテーションを IBM MQ 資格パーツにマップする必要があります。

- IBM MQ Advanced container マルチインスタンス
 - 1 x IBM MQ Advanced および 1 x IBM MQ Advanced 高可用性レプリカ または
 - 2 つの IBM MQ Advanced³
- 非実稼働環境用の IBM MQ Advanced container マルチインスタンス
 - 1 x IBM MQ Advanced および 1 x IBM MQ Advanced 高可用性レプリカ または
 - 2 x IBM MQ Advanced (非実稼働環境の場合)³
- IBM MQ コンテナ・マルチインスタンス
 - 1 x IBM MQ および 1 x IBM MQ 高可用性レプリカ または
 - 2 つの IBM MQ³
- IBM MQ Container Multi Instance for Non-Production Environment
 - 1 x IBM MQ および 1 x IBM MQ 高可用性レプリカ または
 - 2 x IBM MQ (非実稼働環境の場合)³

IBM MQ ネイティブ HA

ネイティブ HA クォーラムに 3 つのキュー・マネージャーをデプロイする場合、アクティブ・インスタンスのみがライセンスを消費します。すべてのインスタンスに同じアノテーションを付ける必要があります。購入したライセンスに応じて、以下のいずれかを選択する必要があります。

- IBM MQ または IBM MQ Advanced スタンドアロン・ライセンス
 - [16 ページの『IBM MQ 拡張』](#)
 - [17 ページの『IBM MQ Advanced for Non-Production Environment \(非実稼働環境向け拡張\)』](#)
- IBM Cloud Pak for Integration 資格
 - [18 ページの『IBM MQ Advanced with CP4I ライセンス』](#)
 - [18 ページの『IBM MQ Advanced for Non-Production Environment with CP4I の使用許諾』](#)

アノテーション

このトピックの残りの部分では、各アノテーションの内容について詳しく説明します。

IBM MQ

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

IBM MQ 拡張

```
productID: "208423bb063c43288328b1d788745b0c"  
productName: "IBM MQ Advanced"
```

³ このライセンス・オプションは最適ではありません。関連する高可用性レプリカ・パーツのライセンスがない場合のみ使用してください。


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

IBM MQ (非実稼働環境)

```
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 (非実稼働環境向け拡張)

```
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 コンテナ・マルチインスタンス

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

IBM MQ 拡張コンテナ・マルチインスタンス

```
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"
```

IBM MQ with CP4I ライセンス

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

IBM MQ Advanced with CP4I ライセンス

```
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 の使用許諾

```
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 の使用許諾

```
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 のライセンス

```
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 のライセンス

```
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 の使用許諾

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

IBM MQ Advanced Container Multi Instance for Non-Production Environment with CP4I のライセンス

```
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 の依存関係

IBM MQ Operator 3.0 以降では、IBM MQ Operator のインストール時に他のオペレーターが自動的にインストールされることはありません。旧バージョンの IBM MQ Operator では、IBM Cloud Pak foundational services Operator に依存します。これにより、IBM Operand Deployment Lifecycle Manager (ODLM) Operator もインストールされます。

IBM Licensing Operator は、ライセンス使用量を追跡するために別個にインストールする必要があります。IBM Cloud Pak for Integration 資料の [License Service](#) を参照してください。

IBM MQ Operator 3.0 以降

V 9.3.4

IBM Cloud Pak for Integration ライセンスを使用して QueueManager を作成する場合、Keycloak の IBM Cloud Pak for Integration インスタンスでシングル・サインオンを使用するかどうかを選択できます。Keycloak の使用は、IBM Cloud Pak for Integration ライセンスではデフォルトで有効になっていますが、インストールされていない場合、QueueManager は正しい依存関係がインストールされるまで「Blocked」状態になります。依存関係について詳しくは、[103 ページの『IBM MQ Operator のインストール』](#)を参照してください。

旧バージョンの IBM MQ Operator

IBM Cloud Pak foundational services オペレーターは、IBM MQ Operator の旧バージョンをインストールすると自動的にインストールされます。これらの依存オペレーターは、CPU とメモリのフットプリントが小さく、状況によっては追加リソースを配置するために使用されます。

QueueManager を作成すると、IBM MQ Operator は、必要な追加サービス用に OperandRequest を作成します。OperandRequest は ODLM オペレーターによって実行され、必要に応じて必要なサービスをインストールしてインスタンス化します。どのサービスが必要かは、キュー・マネージャーの導入時に受け入れたライセンス契約と、どのキュー・マネージャー・コンポーネントが要求されるかに基づいて決定されます。

- 「IBM MQ Advanced」または「IBM MQ Advanced for Developers」ライセンスを選択した場合、追加のサービスは要求されません。例えば、以下の場合、IBM Cloud Pak foundational services は使用されません。

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

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

- IBM Cloud Pak for Integration ライセンスを選択し、Web サーバーを使用可能にすることを選択した場合、IBM MQ Operator は、IBMID およびアクセス管理 (IAM) オペレーターのインスタンスを生成して、シングル・サインオンを可能にします。IBM Cloud Pak for Integration Operator がインストール済みの場合、IAM Operator はすでに使用可能です。以下に例を示します。

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

ただし、ウェブサーバーを使用不可に設定すると、IBM Cloud Pak foundational services は要求されません。以下に例を示します。

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

従属オペレーターのハードウェア要件およびソフトウェア要件の詳細については、[ファウンデーションサービスのためのハードウェア要件と推奨事項](#)を参照してください。

キュー・マネージャーが使用する CPU とメモリーの量を選択できます。詳しくは、[192 ページの『spec.queueManager.resources』](#)を参照してください。

関連資料

[183 ページの『mq.ibm.com/v1beta1 のライセンスのリファレンス』](#)

OpenShift

CP4I

IBM MQ Operator に必要なクラスター・スコープ許可

IBM MQ Operator には、アドミッション Webhook やサンプルを管理したり、ストレージ・クラスやクラスター・バージョンの情報を読み取ったりするためのクラスター・スコープの権限が必要です。

IBM MQ Operator は、以下のクラスター・スコープ許可を必要とします。

- アドミッション Webhook を管理する権限。オペレーターが提供するコンテナの作成と管理のプロセスで使用される特定の Webhook の作成、取得、更新が可能になります。
 - API グループ: **admissionregistration.k8s.io**
 - リソース: **validatingwebhookconfigurations**
 - verbs: **get, delete**
- カスタム・リソースの作成時にサンプルおよびスニペットを提供するために、Red Hat OpenShift コンソールで使用されるリソースを作成および管理する権限です。
 - API グループ: **console.openshift.io**
 - リソース: **consoleyamlsamples**
 - verbs: **create, get, update, delete**
- クラスター・バージョンを読み取る権限。オペレーターがクラスター環境に関する問題をフィードバックすることが可能になります。
 - API グループ: **config.openshift.io**
 - リソース: **clusterversions**
 - verbs: **get, list, watch**
- クラスター上のストレージ・クラスを読み取る権限。オペレーターがコンテナ内の選択したストレージ・クラスに関する問題をフィードバックすることが可能になります。

- API グループ: **storage.k8s.io**
- リソース: **storageclasses**
- verbs: **get, list**

注: IBM MQ Operator には、名前空間を有効範囲とする許可も必要です。IBM MQ Operator がクラスター・スコープでインストールされている場合、名前空間スコープの許可はすべての名前空間に存在します。

OpenShift

CP4I

Kubernetes

IBM MQ Operator のストレージに関する考慮事項

IBM MQ Operator は、次の 2 つのストレージ・モードで稼働します。

- **一時ストレージ**は、コンテナの再始動時にコンテナのすべての状態情報を破棄してよい場合に使用します。これは、デモンストレーション用の環境を作成する場合や、スタンドアロンのキュー・マネージャーを使用して開発する場合によく使用されます。
- **永続ストレージ**は IBM MQ の一般的な構成であり、コンテナが再始動されても、既存の構成、ログ、永続メッセージを再始動後のコンテナで使用することができます。

IBM MQ Operator は、環境によってかなり異なるものになることがあるストレージ特性と、必要なストレージ・モードをカスタマイズする機能を備えています。

一時ストレージ

IBM MQ はステートフル・アプリケーションであるため、再始動時にリカバリーできるように、自身の状態をストレージに保存します。一時ストレージを使用している場合は、キュー・マネージャーのすべての状態情報が再始動時に失われます。これには、以下が含まれます。

- すべてのメッセージ
- すべてのキュー・マネージャー間通信の状態 (チャネルのメッセージ・シーケンス番号)
- キュー・マネージャーの MQ クラスター ID
- すべてのトランザクション状態
- すべてのキュー・マネージャー構成
- ローカルにあるすべての診断データ

このため、実稼働、テスト、または開発のシナリオにとって一時ストレージが適したアプローチであるかどうかを検討する必要があります。例えば、すべてのメッセージが非永続メッセージであると認識され、キュー・マネージャーが MQ クラスターのメンバーでない場合は、再始動時にすべてのメッセージング状態が廃棄されるだけでなく、キュー・マネージャーの構成も廃棄されます。完全に一時的であるコンテナを有効にするには、コンテナ・イメージ自体に IBM MQ 構成を追加する必要があります (詳しくは、[166 ページの『Red Hat OpenShift CLI を使用した、カスタム MQSC および INI ファイルを使用したイメージの作成』](#)を参照してください)。これを行わない場合は、コンテナが再始動するたびに IBM MQ を構成する必要があります。

OpenShift

CP4I

例えば、IBM MQ に一時ストレージを構成するには、QueueManager のストレージ・タイプに以下を指定する必要があります。

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

永続ストレージ

OpenShift

CP4I

IBM MQ は通常、永続ストレージを使用して実行され、キュー・マネージャーが再始動後も永続メッセージと構成を保持するようにします。これはデフォルトの動作です。さまざまなストレージ・プロバイダーがあり、それぞれが異なる機能をサポートしているため、多くの場合、構成のカスタマイズが必要になりま

す。以下の例は、v1beta1 API の IBM MQ ストレージ構成をカスタマイズする共通フィールドの概要を示しています。

- **spec.queueManager.availability** は、可用性モードを制御します。SingleInstance または NativeHA を使用している場合は、ReadWriteOnce ストレージのみが必要です。multiInstance には、正しいファイル・ロック特性を持つ ReadWriteMany をサポートするストレージ・クラスが必要です。IBM MQ は、[サポートに関するステートメント](#)と[テストに関するステートメント](#)を提示しています。可用性モードは、永続ボリュームのレイアウトにも影響します。詳しくは、[25 ページの『コンテナ内の IBM MQ の高可用性』](#)を参照してください。
- **spec.queueManager.storage** は、個々のストレージ設定を制御します。キュー・マネージャーは、1 つから 4 つの永続ボリュームを使用するように構成できます。

次の例は、単一インスタンスのキュー・マネージャーを使用する単純な構成のスニペットを示しています。

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

次の例は、マルチインスタンスのキュー・マネージャー構成のスニペットを示しており、デフォルトではないストレージ・クラスを指定し、補助グループを必要とするファイル・ストレージを指定しています。

```
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
```

ネイティブ HA キュー・マネージャーのストレージに関する考慮事項については、[144 ページの『ネイティブ HA』](#)を参照してください。

注：また、単一インスタンス・キュー・マネージャーを使用して、補足グループを構成することもできます。

ストレージ容量



IBM MQ Operator を使用する場合、要求されるストレージのサイズは固定され、キュー・マネージャーの作成後にサイズ変更することはできません。ボリュームが必要に応じて十分な大きさであることを確認する必要があります。

暗号化



IBM MQ は、保存データをアクティブに暗号化しません。したがって、メッセージを暗号化するには、パッシブに暗号化されたストレージまたは IBM MQ Advanced Message Security、あるいはその両方を使用する必要があります。IBM Cloud® では、ブロック・ストレージとファイル・ストレージの両方で、保存時にパッシブ暗号化を使用できます。

メージ

IBM MQ Advanced for Developers には、事前ビルドされたコンテナ・イメージが用意されています。このイメージは、IBM Container Registry から入手できます。このイメージは、Docker、Podman、Kubernetes、およびその他のコンテナ環境での使用に適しています。

注: **Deprecated** IBM MQ Advanced for Developers イメージは以前は Docker Hub から入手できましたが、これは非推奨であり、Docker Hub ではこれ以上の更新は提供されません。

使用可能なイメージ

IBM MQ イメージは、IBM Container Registry に保管されます。

- 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`

クイック・リファレンス

- ライセンス:
 - [IBM MQ Advanced for Developers](#) および [Apache License 2.0](#)。IBM MQ Advanced for Developers ライセンスでは、これ以上の配布は許可されず、条件により開発者マシンへの使用が制限されることに注意してください。
- 問題のファイリング先:
 - [GitHub](#)
- 以下の CPU アーキテクチャーで使用可能です。
 - amd64
 - s390x
 - ppc64le

使用法

コンテナで [IBM MQ Advanced for Developers](#) を実行します。

コンテナの実行方法について詳しくは、[使用法の資料](#) を参照してください。

イメージを使用できるようにするには、**LICENSE** 環境変数を設定して IBM MQ ライセンスの条項に同意する必要があります。

サポートされる環境変数

LANG

ライセンスの印刷に使用する言語を設定します。

LICENSE

IBM MQ Advanced for Developers ライセンス条件に同意するには、`accept` を設定します。

ライセンス条件を表示するには、`view` を設定します。

Deprecated log_format

非推奨: 24 ページの『[\[MQ 9.3.2 2023 年 2 月\]MQ_LOGGING_CONSOLE_FORMAT](#)』に置き換えられました。

コンテナの `stdout` ロケーションに出力されるログのフォーマットを変更します。

単純な人間が理解できる形式を使用するには、`basic` を設定します。これがデフォルト値です。

JSON 形式 (各行に 1 つの JSON オブジェクト) を使用するように `json` を設定します。

Deprecated **MQ_ADMIN_PASSWORD**

admin ユーザーのパスワードを指定します。

長さは 8 文字以上でなければなりません。

V 9.3.4 admin ユーザーのデフォルト・パスワードはありません。3.0.0 より前の IBM MQ Operator バージョンの場合、デフォルト値は `passwd` です。

V 9.3.4 IBM MQ 9.3.4 以降、この変数は推奨されなくなりました。[このトピックの YAML の例](#)は、この変数を自分で作成し、シークレットで保護する方法を示しています。

Deprecated **MQ_APP_PASSWORD**

アプリケーション・ユーザーのパスワードを指定します。

これを設定すると、**DEV.APP.SVRCONN** チャンネルが保護され、有効なユーザー ID とパスワードを提供する接続のみが許可されます。

長さは 8 文字以上でなければなりません。

V 9.3.4 アプリケーション・ユーザーのデフォルト・パスワードはありません。3.0.0 より前のバージョンの IBM MQ Operator の場合、IBM MQ クライアントではデフォルト値はブランク (パスワードは不要) で、HTTP クライアントでは `passwd` です。

V 9.3.4 IBM MQ 9.3.4 以降、この変数は推奨されなくなりました。[このトピックの YAML の例](#)は、この変数を自分で作成し、シークレットで保護する方法を示しています。

MQ_DEV

作成中のデフォルト・オブジェクトを停止するには、`false` に設定します。

MQ_ENABLE_METRICS

`true` に設定すると、キュー・マネージャーの Prometheus メトリックが生成されます。

V 9.3.2 **MQ_LOGGING_CONSOLE_SOURCE**

コンテナの **stdout** ロケーションにミラーリングされるログのソースのコンマ区切りリストを指定します。

有効な値は `qmgr` および `web` です。

デフォルト値は `qmgr, web` です。

V 9.3.2 **MQ_LOGGING_CONSOLE_FORMAT**

23 ページの『[\[非推奨\]log_format](#)』を置き換えます。

コンテナの **stdout** ロケーションに出力されるログのフォーマットを変更します。

単純な人間が理解できる形式を使用するには、`basic` を設定します。これがデフォルト値です。

JSON 形式 (各行に 1 つの JSON オブジェクト) を使用するように `json` を設定します。

V 9.3.2 **MQ_LOGGING_CONSOLE_EXCLUDE_ID**

除外するログ・メッセージのメッセージ ID のコンマ区切りリストを指定します。

ログ・メッセージは引き続きディスク上のログ・ファイルに表示されますが、コンテナの **stdout** ロケーションには出力されません。

デフォルト値は `AMQ5041I, AMQ5052I, AMQ5051I, AMQ5037I, AMQ5975I` です。

mq_qmgr_name

キュー・マネージャーの作成に使用する名前を設定します。

IBM MQ Advanced for Developers イメージでサポートされるデフォルトの開発者構成について詳しくは、[デフォルトの開発者構成の資料](#)を参照してください。

V 9.3.4 admin および app ユーザーのパスワードを指定する方法を説明するキュー・マネージャー YAML の例

IBM MQ 9.3.4 以降、**admin** および **app** のユーザー ID にデフォルトのパスワードが設定されなくなりました。これらのユーザーについては、Development ライセンスを使用してキュー・マネージャーをデプロイするときにパスワードを指定する必要があります。IBM MQ Operator でこれを行う方法を示すキュー・マネージャー YAML の例を以下に示します。

以下のコマンドは、**admin** および **app** ユーザーのパスワードを含むシークレットを作成します。

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

以下の YAML は、キュー・マネージャーのデプロイ時にこれらのパスワードを使用します。

```
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_HTTP
              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 コンテナ内の IBM MQ の高可用性

IBM MQ Operator での高可用性には 3 つの選択肢があります。ネイティブ HA キュー・マネージャー (アクティブなレプリカと 2 つのスタンバイ・レプリカを持つ)、複数インスタンス・キュー・マネージャー (共有ネットワーク・ファイル・システムを使用するアクティブ/スタンバイ・ペア)、または単一回復力キュー・マネージャー (ネットワーク・ストレージを使用する HA の単純なアプローチを提供する) です。後者の 2 つは、リカバリー可能データを確実に使用できるかどうかは、ファイル・システムが鍵を握りますが、ネイティブ HA ではそうではありません。そのため、ネイティブ HA を使用しない場合、ファイル・システムの可用性はキュー・マネージャーの可用性にとって非常に重要となります。データ・リカバリーが重要となる場合は、複製を行ってファイル・システムの冗長性を確保する必要があります。

メッセージの可用性とサービスの可用性は分けて考える必要があります。IBM MQ for [Multiplatforms](#) を使用する場合、メッセージは厳密に 1 つのキュー・マネージャーに保管されます。そのため、そのキュー・マネージャーが使用不可になると、その中に保管されているメッセージに一時的にアクセスできなくなります。メッセージの可用性を高めるためには、できるだけ速やかにキュー・マネージャーを復旧できなければなりません。サービスの可用性を高めるには、IBM MQ 均一クラスターを使用するなど、クライアント・アプリケーションが使用するキューのインスタンスを複数用意しておくことができます。

キュー・マネージャーは、ディスク上に保管されるデータとそのデータへのアクセスを可能にする実行プロセスの2つの部分に分けて考えることができます。キュー・マネージャーは、同じデータ ([Kubernetes Persistent Volumes](#) によって提供されたもの) を保持し、クライアント・アプリケーションによってネットワーク上で引き続きアドレス可能である限り、別の Kubernetes ノードに移動することができます。Kubernetes では、ネットワークにおける同一性を維持するために1つのサービスが一貫して使用されま

す。

IBM MQ は、永続ボリュームのデータの可用性に依存しています。このため、IBM MQ の可用性は使用するストレージの可用性を上回ることができないので、永続ボリュームを提供するストレージの可用性はキュー・マネージャーの可用性にとって非常に重要となります。可用性ゾーン全体の障害を許容する場合は、ディスク書き込みを別のゾーンに複製するボリューム・プロバイダーを使用することが必要です。

ネイティブ HA キュー・マネージャー

CP4I

MQ Adv.

ネイティブ HA キュー・マネージャーには、1つの **アクティブ** と2つの **レプリカ** Kubernetes ポッドが含まれます。これらのポッドは、Kubernetes StatefulSet の一部として実行され、それぞれに独自の Kubernetes 永続ボリューム・セットを持つ正確に3つのレプリカが含まれます。ネイティブ HA キュー・マネージャーを使用する場合、IBM MQ での共有ファイル・システムの要件も適用されますが(リース・ベースのロックを除く)、共有ファイル・システムを使用する必要はありません。上部に適切なファイル・システムを配置することで、ブロック・ストレージを使用できます。例えば、*xfs* や *ext4* を配置します。ネイティブ HA キュー・マネージャーが復旧に要する時間は、以下の要因によって左右されます。

1. アクティブ・インスタンスに障害が発生したことをレプリカ・インスタンスが検出するのにどれほどの時間がかかるか。これは構成可能です。
2. 作動可能コンテナが変更されてネットワーク・トラフィックがリダイレクトされたことを Kubernetes ポッドの Readiness Probe が検出するまでにかかる時間。これは構成可能です。
3. IBM MQ クライアントが再接続するまでにかかる時間。

詳細については、[144 ページの『ネイティブ HA』](#)を参照してください。

複数インスタンス・キュー・マネージャー

Multi

複数インスタンス・キュー・マネージャーには**アクティブ**で**スタンバイ状態**の Kubernetes ポッドが必要で、これらは厳密に2つのレプリカと Kubernetes 永続ボリューム一式と共に Kubernetes ステートフル・セットの一部として稼働します。キュー・マネージャーのトランザクション・ログとトランザクション・データは、共有ファイル・システムを使用して、2つの永続ボリュームに保管されます。

複数インスタンス・キュー・マネージャーには、永続ボリュームへの同時アクセスを可能にするために、**アクティブ**なポッドと**スタンバイ状態**のポッドの両方が必要です。これを構成するには、**access mode** を `ReadWriteMany` に設定した Kubernetes 永続ボリュームを使用します。これらのボリュームは、IBM MQ の共有ファイル・システムの要件も満たしていなければなりません。IBM MQ がキュー・マネージャー・フェイルオーバーの実施をファイル・ロックの自動解除に依存しているからです。IBM MQ は [テスト対象ファイル・システムのリスト](#)を作成します。

複数インスタンス・キュー・マネージャーが復旧に要する時間は、以下の要因によって左右されます。

1. 障害が発生した後、もともとアクティブ・インスタンスによって実行されたロックを共有ファイル・システムが解除するためにかかる時間。
2. スタンバイ状態のインスタンスがロックを取得してから起動するまでにかかる時間。
3. 作動可能コンテナが変更されてネットワーク・トラフィックがリダイレクトされたことを Kubernetes ポッドの Readiness Probe が検出するまでにかかる時間。これは構成可能です。
4. IBM MQ クライアントが再接続するまでにかかる時間。

シングル・レジリエント・キュー・マネージャー

Multi

シングル・レジリエント・キュー・マネージャーは、1つの Kubernetes ポッド内で実行されるキュー・マネージャーの単一のインスタンスのことで、ここで Kubernetes はキュー・マネージャーをモニタリングし、必要に応じてこのポッドを置き換えます。

シングル・レジリエント・キュー・マネージャーを使用する場合、IBM MQ での共有ファイル・システムの要件も適用されますが (リース・ベースのロックを除く)、共有ファイル・システムを使用する必要はありません。上部に適切なファイル・システムを配置することで、ブロック・ストレージを使用できます。例えば、`xfs` や `ext4` を配置します。

シングル・レジリエント・キュー・マネージャーが復旧に要する時間は、以下の要因によって左右されます。

1. Liveness プロブの実行にかかる時間、および Liveness プロブが許容する失敗の数。これは構成可能です。
2. Kubernetes スケジューラーが失敗したポッドを新規ノードに再スケジュールするためにかかる時間。
3. コンテナ・イメージを新規ノードにダウンロードするためにかかる時間。IfNotPresent に `imagePullPolicy` 値を使用している場合は、すでにそのノードで画像が利用可能になっている可能性があります。
4. 新規キュー・マネージャー・インスタンスが起動するのにかかる時間。
5. コンテナが作動可能であることを Kubernetes ポッドの Readiness Probe が検出するまでにかかる時間。これは構成可能です。
6. IBM MQ クライアントが再接続するまでにかかる時間。

重要:

シングル・レジリエント・キュー・マネージャー・パターンにはいくつかの利点がありますが、ノードの障害に関連した制限がある状態で、目標とする可用性に達するかどうかについて十分に理解しておく必要があります。

Kubernetes では、障害が発生したポッドは通常迅速に復旧しますが、ノード全体で障害が発生した場合は対応が異なります。IBM MQ のようなステートフル・ワークロードを Kubernetes StatefulSet で使用する場合、Kubernetes マスター・ノードは、ワーカー・ノードとの接続を失うと、ノードに障害が発生したかどうか、または単にネットワーク接続が失われたかどうかを判別できません。そのため、このような場合、次のいずれかのイベントが発生するまで、Kubernetes は何も対応しません。

1. ワーカー・ノードが Kubernetes マスター・ノードと通信できる状態に復旧する。
2. 管理上の処置として、Kubernetes マスター・ノード上のポッドを明示的に削除する。この場合、必ずしもポッドの実行を停止するわけではなく、単に Kubernetes ストアから削除します。したがって、この管理上の処置は慎重に行う必要があります。

注: IBM MQ Operator を使用してキュー・マネージャーを作成する場合、IBM MQ キュー・マネージャーの StatefulSet の詳細 (レプリカの数を含む) の変更はサポートされません。

関連概念

高可用性の構成

関連タスク

144 ページの『IBM MQ Operator を使用したキュー・マネージャーの高可用性の構成』

OpenShift

CP4I

Kubernetes

コンテナ内の IBM MQ の災害復旧

どのような種類の災害に備えるのかを検討する必要があります。クラウド環境では、複数のアベイラビリティ・ゾーンを使用すると、災害に対して一定レベルの耐障害性を確保できる上、利用方法も簡単です。奇数個のデータ・センター (クォーラムの場合) があり、ネットワーク・リンクの待ち時間が短い場合は、単一の Red Hat OpenShift Container Platform クラスターまたは Kubernetes クラスターを、別々の物理的な場所にある複数のアベイラビリティ・ゾーンで実行することもできます。このトピックでは、これらの基準を満たすことができない場合 (つまり、データ・センターが偶数個の場合やネットワーク・リンクの待ち時間が長い場合) の災害復旧に関する考慮事項を説明します。

災害復旧では、以下の点を考慮する必要があります。

- 災害復旧ロケーションへの IBM MQ データ (1 つ以上の PersistentVolume リソースで保持されている) の複製
- 複製されたデータを使用してキュー・マネージャーを再作成すること
- IBM MQ クライアント・アプリケーションおよびその他のキュー・マネージャーに表示されるキュー・マネージャーのネットワーク ID。例えば、この ID は DNS 項目になります。

永続データを災害復旧サイトに同期的または非同期的に複製する必要があります。通常、これはストレージ・プロバイダーに固有ですが、VolumeSnapshot を使用して行うこともできます。ボリュームのスナップショットについて詳しくは、[CSI volume snapshots](#) を参照してください。

災害から復旧する場合には、複製されたデータを使用して、新しい Kubernetes クラスター上でキュー・マネージャー・インスタンスを再作成する必要があります。IBM MQ Operator を使用している場合は、QueueManager YAML が必要なほか、ConfigMap や Secret など、他のサポート・リソース用の YAML も必要になります。

関連情報

[ha_for_ctr.dita](#)

OpenShift

CP4I

コンテナ内の IBM MQ を保護するための計画

コンテナ構成で IBM MQ を計画する際のセキュリティに関する考慮事項。

手順

- [28 ページの『コンテナ内の IBM MQ でのユーザーの認証と許可』](#)
 - [29 ページの『コンテナでのオペレーティング・システム・ユーザーの使用に関するセキュリティ制約』](#)
- [29 ページの『コンテナ内の IBM MQ へのネットワーク・トラフィックを制限する際の考慮事項』](#)

コンテナ内の IBM MQ でのユーザーの認証と許可

コンテナ内の IBM MQ は、LDAP、相互 TLS、またはカスタム MQ プラグインを使用してユーザーを認証するように構成できます。

IBM MQ Operator では、コンテナ・イメージ内でのオペレーティング・システムのユーザーおよびグループの使用は許可されないことに注意してください。詳細については、[29 ページの『コンテナでのオペレーティング・システム・ユーザーの使用に関するセキュリティ制約』](#) を参照してください。

LDAP

LDAP ユーザー・リポジトリを使用するように IBM MQ を構成する方法については、[接続認証: ユーザー・リポジトリ](#) および [LDAP 許可](#) を参照してください。

相互 TLS

TLS 証明書 (相互 TLS) を必要とするようにキュー・マネージャーへの着信接続を構成する場合は、証明書の識別名をユーザー名にマップできます。以下の 2 つの必要があることを行います。

- SSLPEER を使用して、ユーザー名へのマッピングを作成するようにチャンネル認証レコードを構成します。詳しくは、[MCAUSER ユーザー ID への SSL または TLS 識別名のマッピング](#) を参照してください。
- キュー・マネージャーを構成して、システムに認識されていないユーザー名の権限レコードを定義できるようにします。詳しくは、[qm.ini ファイルの Service スタンザ](#) を参照してください。

JSON Web トークン

JSON Web トークン (JWT) を使用するように IBM MQ を構成する方法については、[認証トークンの処理](#) を参照してください。

カスタム MQ プラグイン

これは高度な手法であり、さらに多くの作業を必要とします。詳しくは、[カスタム許可サービスの使用](#)を参照してください。

関連タスク

[139 ページの『例: 相互 TLS 認証を使用したキュー・マネージャーの構成』](#)


この例では、IBM MQ Operator を使用して、キュー・マネージャーを OpenShift Container Platform にデプロイします。相互 TLS は、認証に使用され、TLS 証明書からキュー・マネージャー内の ID にマップされます。

コンテナでのオペレーティング・システム・ユーザーの使用に関するセキュリティー制約

コンテナでのオペレーティング・システム・ユーザーの使用は推奨されておらず、IBM MQ Operator では禁止されています。

マルチテナントのコンテナ環境では、潜在的なセキュリティー問題を防ぐために、通常は以下の例のようなセキュリティー制約が適用されます。

- **コンテナ内での「root」ユーザーの使用の防止**
- **ランダム UID の使用の強制。** 例えば Red Hat OpenShift Container Platform では、SecurityContextConstraints のデフォルト (つまり restricted) を使用すると、コンテナごとにランダムなユーザー ID が使用されます。
- **特権エスカレーションの使用の防止。** IBM MQ on Linux は、特権エスカレーションを使用してユーザーのパスワードを検査します。つまり、「root」ユーザーになるために「setuid」プログラムを使用します。

 これらのセキュリティー手段に確実に準拠するために、IBM MQ Operator は、コンテナ内のオペレーティング・システム・ライブラリーで定義されている ID の使用を許可しません。コンテナ内に mqm というユーザー ID やグループは定義されていません。

コンテナ内の IBM MQ へのネットワーク・トラフィックを制限する際の考慮事項

ネットワーク・ポリシーを定義して、[OpenShift Container Platform](#) および [Kubernetes](#) のクラスター内のポッドにトラフィックを制限できます。このトピックでは、ネットワーク・ポリシーを IBM MQ に適用する方法に関するいくつかの考慮事項について説明します。

キュー・マネージャーへのネットワーク入口の場合、考慮すべきポートがいくつかあります。

- キュー・マネージャー・トラフィック用のポート 1414
- ネイティブ HA 用のポート 9414
- メトリック用のポート 9157
- Web コンソールおよび REST API 用のポート 9443

ネットワーク出口はより複雑です。考慮すべきネットワーク Egress の例を以下に示します。

- DNS-DNS 名を使用するチャンネルまたはその他の構成がある場合
- 他のキュー・マネージャー
- Online Certificate Status Protocol (OCSP) および証明書失効リスト (CRL)-証明書プロバイダーによって決定されます。
- 認証プロバイダー:
 - LDAP
 - IBM MQ Web サーバー用の Open ID Connect またはその他の構成済みログイン・プロバイダー。これには、IBM Cloud Pak Platform UI と IBM Cloud Pak Foundational Services IAM が含まれます。
- トレース・プロバイダー:
 - Instana

Ingress NetworkPolicy の例

Red Hat OpenShift Container Platform で使用するために、「myqm」というキュー・マネージャーの Ingress を制御するネットワーク・ポリシーの例を以下に示します。

```
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
```

コンテナ内の IBM MQ の FIPS 準拠

始動時に、コンテナ内の IBM MQ は、コンテナが始動するオペレーティング・システムが FIPS に準拠しているかどうかを検出し、準拠している場合は、FIPS サポートを自動的に構成します。ここでは、要件と制限について説明します。

連邦情報処理標準

米国政府は、データ暗号化など、IT システムおよびセキュリティーに関する技術的助言を行っています。米国連邦情報・技術局 (NIST) は、IT システムとセキュリティーに関する政府機関です。NIST は、連邦情報処理標準 (FIPS) などの勧告や規格を策定しています。

重要な FIPS 標準は FIPS 140-2 であり、強力な暗号アルゴリズムを使用する必要があります。FIPS 140-2 では、転送中のパケットが変更されることを防ぐためにハッシュ・アルゴリズムを使用することも規定しています。

IBM MQ は、FIPS 140-2 サポートを提供します (そのように構成されている場合)。

注: AIX®, Linux, and Windows では、IBM MQ は IBM Crypto for C (ICC) 暗号モジュールを介して FIPS 140-2 準拠を提供します。このモジュールの証明書は「履歴」ステータスに移動されました。お客様は、

⁴ Operations Dashboard は IBM MQ 9.3.0 から非推奨になり、IBM MQ 9.3.3 で削除されました。157 ページの『IBM Cloud Pak for Integration Operations Dashboard との統合』を参照してください。

[IBM Crypto for C \(ICC\) 証明書](#) を表示し、NIST から提供されたアドバイスに注意する必要があります。交換用の FIPS 140-3 モジュールが現在進行中であり、その状況を表示するには、[「NIST CMVP modules in process list」](#) でそのモジュールを検索します。

要件

クラスターのセットアップおよびその他の考慮事項に関連する要件については、[FIPS Wall: Current IBM approach to FIPS compliance](#) を参照してください。

コンテナ内の IBM MQ は、FIPS 140-2 準拠モードで実行できます。始動時に、コンテナ (9.3.1.0 以上) 内の IBM MQ は、コンテナが開始されているホスト・オペレーティング・システムが FIPS に準拠しているかどうかを検出します。ホスト・オペレーティング・システムが FIPS 準拠であり、秘密鍵と証明書が提供されている場合、IBM MQ コンテナは、キュー・マネージャー、IBM MQ Web サーバー、およびネイティブ高可用性デプロイメント内のノード間のデータ転送を FIPS 準拠モードで実行するように構成します。

IBM MQ Operator を使用してキュー・マネージャーをデプロイする場合、オペレーターは終了タイプ **Passthrough** の経路を作成します。これは、ルーターが TLS 終端を提供することなく、トラフィックが宛先に直接送信されることを意味します。この場合、IBM MQ キュー・マネージャーと IBM MQ Web サーバーは宛先であり、既に FIPS 準拠のセキュア通信を提供しています。

主な要件:

1. キュー・マネージャーおよび Web サーバーにシークレットで提供される秘密鍵および証明書。これにより、外部クライアントはキュー・マネージャーおよび Web サーバーに安全に接続できます。
2. ネイティブ高可用性構成内の異なるノード間でのデータ転送用の秘密鍵および証明書。

制限

コンテナ内の IBM MQ の FIPS 準拠デプロイメントの場合は、以下の点を考慮してください。

- コンテナ内の IBM MQ は、メトリックを収集するためのエンドポイントを提供します。現在、このエンドポイントは HTTP のみです。メトリック・エンドポイントをオフにして、残りの IBM MQ を FIPS 準拠にすることができます。
- コンテナ内の IBM MQ は、カスタム・イメージのオーバーライドを許可します。つまり、IBM MQ コンテナ・イメージをベース・イメージとして使用してカスタム・イメージをビルドできます。FIPS 準拠は、このようなカスタマイズされたイメージには適用されない可能性があります。
- IBM Instana を使用したメッセージ・トラッキングの場合、IBM MQ と IBM Instana の間の通信は HTTP または HTTPS であり、FIPS に準拠していません。
- IBM の ID およびアクセス管理 (IAM)/禅サービスへの IBM MQ Operator アクセスは、FIPS 準拠ではありません。

FIPS 準拠が検出され、FIPS サポートが自動的に構成される方法

コンテナが開始されるオペレーティング・システムが FIPS 準拠である場合、FIPS サポートは自動的に構成されます。

注: AIX, Linux, and Windows では、IBM MQ は IBM Crypto for C (ICC) 暗号モジュールを介して FIPS 140-2 準拠を提供します。このモジュールの証明書は「履歴」ステータスに移動されました。お客様は、[IBM Crypto for C \(ICC\) 証明書](#) を表示し、NIST から提供されたアドバイスに注意する必要があります。交換用の FIPS 140-3 モジュールが現在進行中であり、その状況を表示するには、[「NIST CMVP modules in process list」](#) でそのモジュールを検索します。

始動時に、コンテナ内の IBM MQ は、コンテナが開始されているオペレーティング・システムが FIPS に準拠しているかどうかを検出します。その場合、以下のアクションが自動的に実行されます。

キュー・マネージャー

ホスト・オペレーティング・システムが FIPS に準拠しており、秘密鍵と証明書が提供されている場合、キュー・マネージャー属性 **SSLFIPS** は YES に設定されます。それ以外の場合、**SSLFIPS** 属性は NO に設定されます。

IBM MQ Web サーバー

IBM MQ Web サーバーは、IBM MQ を管理するための HTTP/HTTPS インターフェースを提供します。ホスト・オペレーティング・システムが FIPS 準拠の場合、Web サーバーが FIPS 準拠の暗号方式を使用するように JVM オプションが更新されます。FIPS を使用できるようにするには、コンテナの開始時に秘密鍵と証明書を指定する必要があります。

ネイティブ HA

ノード間で複製されるデータのセキュリティは、qm.ini ファイルの **NativeHALocalInstance** スタンザによって制御されます。以下に例を示します。

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

FIPS が有効になっている場合、**SSLFipsRequired** 属性がスタンザに追加され、値が **Yes** に設定されます。

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

FIPS がサポートされていない OpenShift クラスタでコンテナが実行されている場合、キュー・マネージャー、IBM MQ Web サーバー、およびネイティブ HA コンポーネントでは、FIPS が自動的にサポートされません。現在、FIPS 用の OpenShift プラットフォームでは、x86-64 アーキテクチャーのみがサポートされています。Power および Linux for IBM Z[®] アーキテクチャーの場合、OpenShift は FIPS サポートを提供しません。これらのアーキテクチャーの IBM MQ コンポーネントで FIPS サポートを明示的に有効にするには、キュー・マネージャー YAML で **MQ_ENABLE_FIPS** 環境変数を **true** に設定します。以下の YAML スニペットは、**MQ_ENABLE_FIPS** 環境変数の使用法を説明しています。

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

コンテナ内の IBM MQ の自動 FIPS モードのオーバーライド

コンテナ内の IBM MQ コンポーネントに対して FIPS モードを明示的に有効または無効にするには、環境変数 **MQ_ENABLE_FIPS** を使用します。

始める前に

注：AIX, Linux, and Windows では、IBM MQ は IBM Crypto for C (ICC) 暗号モジュールを介して FIPS 140-2 準拠を提供します。このモジュールの証明書は「履歴」ステータスに移動されました。お客様は、**IBM Crypto for C (ICC) 証明書** を表示し、NIST から提供されたアドバイスに注意する必要があります。交換用の FIPS 140-3 モジュールが現在進行中であり、その状況を表示するには、「[NIST CMVP modules in process list](#)」でそのモジュールを検索します。

このタスクについて

MQ_ENABLE_FIPS は、以下の 3 つの値をサポートします。

auto

これがデフォルト値です。

ホスト・オペレーティング・システムで FIPS が有効になっている場合は、すべてのコンポーネント (キュー・マネージャー、IBM MQ Web サーバー、およびネイティブ HA) が FIPS モードで実行されます。

ホスト・オペレーティング・システムで FIPS が有効になっていない場合、すべてのコンポーネントが FIPS モードで実行されるわけではありません。

true

この値は、コンテナ内の選択されたコンポーネントに対して FIPS をオンにします。

コンテナ内の IBM MQ が FIPS 準拠でないホスト・オペレーティング・システムで実行されている場合でも、キュー・マネージャー属性 **SSLFIPS** は YES に設定されます。つまり、IBM MQ キュー・マネージャー、Web サーバー、およびネイティブ HA は FIPS 準拠ですが、コンテナのオペレーティング・システムは FIPS 準拠ではありません。

false

この値は、FIPS 準拠をオフにします。

コンテナ内の IBM MQ が FIPS 準拠のホスト・マシンで実行されている場合でも、キュー・マネージャー属性 **SSLFIPS** は NO に設定されます。ただし、秘密鍵と証明書が提供されている場合でも、IBM MQ は接続を保護します。

IBM MQ Web サーバーの JVM オプションは更新されません。ただし、秘密鍵と証明書が指定されている場合、IBM MQ Web サーバーは引き続き HTTPS エンドポイントを実行します。

ネイティブ HA でのデータ複製では、FIPS 暗号化は使用されません。

例

キュー・マネージャー・コンポーネントの TLS および FIPS を有効にする方法を説明するサンプル・キュー・マネージャー YAML を以下に示します。

```
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

コンテナ内の IBM MQ のスケーラビリティとパフォーマンスの計

画

ほとんどの場合、コンテナ内の IBM MQ のスケーリングとパフォーマンスは、IBM MQ for Multiplatforms と同じです。ただし、コンテナ・プラットフォームによって課される可能性がある追加の制限がいくつかあります。

このタスクについて

コンテナ内の IBM MQ のスケーラビリティとパフォーマンスを計画する場合は、以下のオプションを考慮してください。

手順

- スレッドおよびプロセスの数を制限します。

IBM MQ はスレッドを使用して並行性を管理します。Linux では、スレッドはプロセスとして実装されるため、コンテナ・プラットフォームまたはオペレーティング・システムによって課される、プロセスの最大数に関する制限を検出することができます。Red Hat OpenShift Container Platform 4.11 以降、コンテナ当たりのプロセス数にはデフォルトの 4096 個の制限があります。旧バージョンの OpenShift Container Platform の場合、この制限は 1024 プロセスです。IBM MQ Operator のバージョンと OpenShift Container Platform のバージョンとの互換性については、[13 ページの『互換性のある Red Hat OpenShift Container Platform のバージョン』](#)を参照してください。これは大部分のシナリオに適していますが、キュー・マネージャーのクライアント接続の数の影響を与える可能性がある場合があります。

Kubernetes のプロセス制限は、クラスター管理者が kubelet 構成設定 `podPidsLimit` を使用して構成できます。Kubernetes 資料の [プロセス ID の制限と予約](#) を参照してください。Red Hat OpenShift Container Platform では、[ContainerRuntimeConfig](#) カスタム・リソースを作成して CRI-O パラメーターを編集することもできます。

IBM MQ 構成では、キュー・マネージャーのクライアント接続の最大数を設定することもできます。個々のサーバー接続チャンネルに制限を適用する場合は [サーバー接続チャンネルの制限](#) を、キュー・マネージャー全体に制限を適用する場合は [MAXCHANNELS INI 属性](#) を参照してください。

- ボリューム数を制限します。

クラウド・システムおよびコンテナ・システムでは、Network Attached Storage ボリュームが一般的に使用されます。Linux ノードに接続できるボリュームの数には制限があります。例えば、[AWS EC2](#) は、[VM 当たりのボリューム数を 30 以下に制限](#)します。Red Hat OpenShift Container Platform 類似した制限がある (Microsoft Azure および Google Cloud Platform と同様)。

ネイティブ HA キュー・マネージャーは、3 つのインスタンスごとに 1 つのボリュームを必要とし、インスタンスをノード間に分散させます。ただし、インスタンスごとに 3 つのボリューム (キュー・マネージャー・データ、リカバリー・ログ、および永続データ) を使用するようキュー・マネージャーを構成できます。

- IBM MQ のスケーリング手法を使用します。

少数の大規模なキュー・マネージャーではなく、IBM MQ 均一クラスターなどの IBM MQ スケーリング手法を使用して、同じ構成で複数のキュー・マネージャーを実行することをお勧めします。これにより、単一コンテナの再始動 (例えば、コンテナ・プラットフォームの保守の一環として) の影響が軽減されるという利点が追加されます。

OpenShift CP4I CD CP4I-LTS Red Hat OpenShift での IBM MQ Operator の使用

IBM MQ Operator は、IBM MQ を IBM Cloud Pak for Integration の一部としてデプロイして管理するか、Red Hat OpenShift Container Platform で単体としてデプロイして管理します。

手順

- [35 ページの『IBM MQ Operator のリリース履歴』](#)。
- [79 ページの『IBM Cloud Pak for Integration への IBM MQ のマイグレーション』](#)。
- [103 ページの『IBM MQ Operator のインストール』](#)。
- [121 ページの『IBM MQ Operator とキュー・マネージャーのアップグレード』](#)。
- [116 ページの『Red Hat OpenShift Container Platform クラスターにキュー・マネージャーをデプロイする方法』](#)。
- [172 ページの『IBM MQ Operator を使用した IBM MQ』](#)。
- [183 ページの『IBM MQ Operator の API リファレンス』](#)。

注:

- 以前の IBM MQ オペレーターについては、IBM MQ 9.2 資料の [IBM MQ Operator のリリース履歴](#) を参照してください。
- 今後の IBM MQ の更新については、[IBM MQ の計画保守リリース日](#) のページ全体を参照してください。

IBM MQ Operator 3.1.3

CD

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2023.4.1

オペレーター・チャンネル

v3.1

.spec.version に許可される値

[9.3.5.1-r2](#)

マイグレーション中に許可される .spec.version の値

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 のバージョン

OpenShift Container Platform 4.12 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これらのリリースは、偶数番号のマイナー・リリース (4.14 や 4.16 など) です。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services バージョン 4.3 以上 (オプションのインストール)。

変更点

- 対処された脆弱性の詳細については、この[セキュリティ情報](#)を参照してください。

IBM MQ Operator 3.1.2

CD

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2023.4.1

オペレーター・チャンネル

v3.1

.spec.version に許可される値

[9.3.5.1-r1](#)

マイグレーション中に許可される .spec.version の値

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 のバージョン

OpenShift Container Platform 4.12 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これらのリリースは、偶数番号のマイナー・リリース (4.14 や 4.16 など) です。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services バージョン 4.3 以上 (オプションのインストール)。

変更点

- 対処された脆弱性の詳細については、この[セキュリティ情報](#)を参照してください。

IBM MQ Operator 3.1.1

CD

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2023.4.1

オペレーター・チャンネル

v3.1

.spec.version に許可される値

[9.3.5.0-r2](#)

マイ그레이ション中に許可される .spec.version の値

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 のバージョン

OpenShift Container Platform 4.12 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これらのリリースは、偶数番号のマイナー・リリース (4.14 や 4.16 など) です。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services バージョン 4.3 以上 (オプションのインストール)。

変更点

- 対処された脆弱性の詳細については、この[セキュリティ情報](#)を参照してください。

IBM MQ Operator 3.1.0

CD

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2023.4.1

オペレーター・チャンネル

v3.1

.spec.version に許可される値

[9.3.5.0-r1](#)

マイ그레이ション中に許可される .spec.version の値

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 のバージョン

OpenShift Container Platform 4.12 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これらのリリースは、偶数番号のマイナー・リリース (4.14 や 4.16 など) です。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services バージョン 4.3 以上 (オプションのインストール)。

変更点

- 対処された脆弱性について詳しくは、以下のセキュリティー情報を参照してください。
 - <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 バージョン

IBM Cloud Pak for Integration 2023.4.1

オペレーター・チャンネル

v3.0

.spec.version に許可される値

[9.3.4.1-r1](#)

マイ그레이ション中に許可される .spec.version の値

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 のバージョン

OpenShift Container Platform 4.12 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これらのリリースは、偶数番号のマイナー・リリース (4.14 や 4.16 など) です。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services バージョン 4.3 以上 (オプションのインストール)。

変更点

- [37 ページの『IBM MQ Operator 3.0.0』](#)に基づいて作成されたセキュリティーのみの更新。
- 対処された脆弱性の詳細については、この[セキュリティー情報](#)を参照してください。

IBM MQ Operator 3.0.0

CD

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2023.4.1

オペレーター・チャンネル

v3.0

.spec.version に許可される値

[9.3.4.0-r1](#)

マイ그레이ション中に許可される .spec.version の値

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 のバージョン

OpenShift Container Platform 4.12 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これらのリリースは、偶数番号のマイナー・リリース (4.14 や 4.16 など) です。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services バージョン 4.3 以上 (オプションのインストール)。

新機能

- 新しい manualConfig YAML プロパティを使用して、mqwebuser.xml ファイルを ConfigMap またはシークレットに追加することで、IBM MQ Web サーバーを構成できます (IBM MQ 9.3.4 以上が必要です)。
- administrative REST API がサポートされるようになりました。これは、上記のように ConfigMap または Secret を使用して構成できます (IBM MQ 9.3.4 以上が必要です)。ただし、Web サーバーは依然として Liveness プローブの重要なサービスとは見なされないため、失敗した場合でもコンテナは自動的に再始動されないことに注意してください。
- 「手動」認証および許可を選択して、IBM Cloud Pak for Integration ライセンスの使用時にシングル・サインオンを無効にする (IBM MQ 9.3.4 以上が必要)
- コンテナ内で読み取り専用のルート・ファイル・システムを有効にすることができます。これにより、実行時にコンテナ内のほとんどのファイルへの書き込みが防止されるため、セキュリティが向上します (IBM MQ 9.3.4 以上が必要です)。readOnlyRootFilesystem オプションには、一時ファイルの書き込みを可能にするためにマウントされる「スクラッチ」ボリュームおよび「tmp」ボリュームのサイズを構成するための追加オプションが付随しています。[169 ページの『読み取り専用ルート・ファイル・システムを使用した IBM MQ コンテナの実行』](#)を参照してください。

変更点

- 削除された (以前に非推奨になった) リリース: IBM MQ 9.2.0 EUS、9.2.3、9.2.4、9.2.5。重要: IBM MQ Operator をアップグレードする前に、削除されたバージョンのキュー・マネージャーがないことを確認してください。アップグレード後、IBM MQ Operator は古いバージョンを認識しなくなるため、サポート中のバージョンにアップグレードする以外に、QueueManager リソースを編集することはできなくなります。
- オペレーターのインストールおよびライフサイクル
 - IBM MQ Operator は、Red Hat OpenShift Container Platform バージョン 4.14 でサポートされるようになりました。
 - IBM MQ Operator は、IBM Cloud Pak foundational services を自動的にインストールしなくなりました。IBM Cloud Pak for Integration ライセンスを使用し、シングル・サインオン (そのライセンスを持つキュー・マネージャーのデフォルト) を構成する QueueManager をデプロイする場合、必要な依存関係がまだインストールされていないければ、QueueManager は「Blocked」状態になります。他のオペレーターは自動的にインストールされません。
- セキュリティーの変更
 - IBM Cloud Pak for Integration 2023.4.1 は、IBM Cloud Pak Identity and Access Manager ではなく、シングル・サインオンおよび許可に Keycloak を使用します。
 - IBM Cloud Pak for Integration の「クイック・スタート」テンプレートは、MQSNOAUTでのセキュリティを無効にしなくなりました。認証を構成する必要があります。[28 ページの『コンテナ内の IBM MQ でのユーザーの認証と許可』](#)を参照してください。
 - バージョン 9.3.4 以降では、IBM MQ Advanced for Developers のデフォルト・ユーザーは無効になっています。デフォルト・ユーザー («admin» および «app»), および IBM MQ Advanced for Developers の一部として提供されているその他の構成は、デフォルトで無効になっています。
- IBM MQ Operator ポッドに対する軽微な変更:
 - IBM MQ Operator は init コンテナをデプロイしなくなりました。
 - IBM MQ Operator コンテナ名は *manager* になりました。
 - IBM MQ Operator ポッドの接頭部は *ibm-mq-operator* です。
- 対処された脆弱性の詳細については、この[セキュリティ情報](#)を参照してください。

IBM MQ Operator 2.4.8

IBM Cloud Pak for Integration バージョン
IBM Cloud Pak for Integration 2023.2.1

オペレーター・チャンネル
v2.4

.spec.version に許可される値

9.3.3.3-r2

マイ그레이ション中に許可される **.spec.version** の値

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 のバージョン

OpenShift Container Platform 4.12 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これらのリリースは、偶数番号のマイナー・リリース (4.14 や 4.16 など) です。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services バージョン 3.19 から 3.24 まで。

変更点

- 42 ページの『[IBM MQ Operator 2.4.0](#)』に基づいて作成されたセキュリティーのみの更新。
- 対処された脆弱性について詳しくは、以下のセキュリティー情報を参照してください。
 - <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 バージョン

IBM Cloud Pak for Integration 2023.2.1

オペレーター・チャンネル
v2.4

.spec.version に許可される値

9.3.3.3-r1

マイ그레이ション中に許可される **.spec.version** の値

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 のバージョン

OpenShift Container Platform 4.12 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これらのリリースは、偶数番号のマイナー・リリース (4.14 や 4.16 など) です。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services バージョン 3.19 から 3.24 まで。

変更点

- 42 ページの『[IBM MQ Operator 2.4.0](#)』に基づいて作成されたセキュリティーのみの更新。
- 対処された脆弱性の詳細については、この[セキュリティー情報](#)を参照してください。

IBM MQ Operator 2.4.6

CD

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2023.2.1

オペレーター・チャンネル

v2.4

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.12 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これらのリリースは、偶数番号のマイナー・リリース (4.14 や 4.16 など) です。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services バージョン 3.19 から 3.24 まで。

変更点

- 42 ページの『[IBM MQ Operator 2.4.0](#)』に基づいて作成されたセキュリティーのみの更新。
- 対処された脆弱性の詳細については、この[セキュリティー情報](#)を参照してください。

IBM MQ Operator 2.4.5

CD

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2023.2.1

オペレーター・チャンネル

v2.4

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.12 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これらのリリースは、偶数番号のマイナー・リリース (4.14 や 4.16 など) です。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services バージョン 3.19 から 3.24 まで。

変更点

- 42 ページの『[IBM MQ Operator 2.4.0](#)』に基づいて作成されたセキュリティーのみの更新。
- 対処された脆弱性の詳細については、この[セキュリティー情報](#)を参照してください。

IBM MQ Operator 2.4.4

CD

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2023.2.1

オペレーター・チャンネル

v2.4

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.12 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これらのリリースは、偶数番号のマイナー・リリース (4.14 や 4.16 など) です。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services バージョン 3.19 から 3.24 まで。

変更点

- [42 ページの『IBM MQ Operator 2.4.0』](#)に基づいて作成されたセキュリティーのみの更新。
- 対処された脆弱性の詳細については、この[セキュリティー情報](#)を参照してください。
- IBM MQ Operator は、OpenShift Container Platform 4.10 ではテストもサポートもされなくなりました。

IBM MQ Operator 2.4.3

CD

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2023.2.1

オペレーター・チャンネル

v2.4

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.10 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これは偶数番号のマイナー・リリースです (例: 4.10 および 4.12)。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services バージョン 3.19 から 3.24 まで。

変更点

- [42 ページの『IBM MQ Operator 2.4.0』](#)に基づいて作成されたセキュリティーのみの更新。
- 対処された脆弱性の詳細については、この[セキュリティー情報](#)を参照してください。

IBM MQ Operator 2.4.2

CD

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2023.2.1

オペレーター・チャンネル

v2.4

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.10 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これは偶数番号のマイナー・リリースです (例: 4.10 および 4.12)。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services バージョン 3.19 から 3.24 まで。

変更点

- 42 ページの『[IBM MQ Operator 2.4.0](#)』に基づいて作成されたセキュリティのみの更新。
- 対処された脆弱性の詳細については、この[セキュリティ情報](#)を参照してください。

IBM MQ Operator 2.4.1

CD

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2023.2.1

オペレーター・チャンネル

v2.4

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.10 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これは偶数番号のマイナー・リリースです (例: 4.10 および 4.12)。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services バージョン 3.19 から 3.24 まで。

変更点

- 42 ページの『[IBM MQ Operator 2.4.0](#)』に基づいて作成されたセキュリティのみの更新。
- 対処された脆弱性の詳細については、この[セキュリティ情報](#)を参照してください。

IBM MQ Operator 2.4.0

CD

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2023.2.1

オペレーター・チャンネル

v2.4

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.10 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これは偶数番号のマイナー・リリースです (例: 4.10 および 4.12)。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services バージョン 3.19 から 3.24 まで。

新機能

- Operations Dashboard の統合を削除しました。
- **LogFilePages** の IBM MQ Operator サポートが追加されました。

変更点

- 対処された脆弱性の詳細については、この[セキュリティ情報を参照してください](#)。

IBM MQ Operator 2.3.3

CD

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.4.1

オペレーター・チャンネル

v2.3

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.10 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これは偶数番号のマイナー・リリースです (例: 4.10 および 4.12)。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services バージョン 3.19 から 3.24 まで。

変更点

- 44 ページの『[IBM MQ Operator 2.3.0](#)』上に構築されたセキュリティのみの更新
- 対処された脆弱性の詳細については、この[セキュリティ情報を参照してください](#)。

IBM MQ Operator 2.3.2

CD

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.4.1

オペレーター・チャンネル

v2.3

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.10 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これは偶数番号のマイナー・リリースです (例: 4.10 および 4.12)。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services バージョン 3.19 から 3.24 まで。

変更点

- 44 ページの『[IBM MQ Operator 2.3.0](#)』上に構築されたセキュリティーのみの更新
- 対処された脆弱性の詳細については、この[セキュリティー情報](#)を参照してください。

IBM MQ Operator 2.3.1

CD

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.4.1

オペレーター・チャンネル

v2.3

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.10 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これは偶数番号のマイナー・リリースです (例: 4.10 および 4.12)。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services バージョン 3.19 から 3.24 まで。

新機能

- 2023 年 3 月以降、IBM MQ Operator および IBM MQ キュー・マネージャーのコンテナ・イメージはデジタル署名されます。IBM MQ Operator 2.3.1 および IBM MQ 9.3.2.0-r2 イメージは、このリリースで署名されています。79 ページの『[イメージ署名の検証](#)』を参照してください。

変更点

- 44 ページの『[IBM MQ Operator 2.3.0](#)』上に構築されたセキュリティーのみの更新
- 対処された脆弱性の詳細については、この[セキュリティー情報](#)を参照してください。

IBM MQ Operator 2.3.0

CD

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.4.1

オペレーター・チャンネル

v2.3

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.10 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これは偶数番号のマイナー・リリースです (例: 4.10 および 4.12)。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services バージョン 3.19 から 3.24 まで。

新機能

- IBM MQ Operator 2.3.0 以降、FIPS 140-2 サポートを構成できます。30 ページの『[コンテナ内の IBM MQ の FIPS 準拠](#)』を参照してください。
- IBM MQ Operator 2.3.0 以降、IBM MQ 9.3.1 は非推奨になりました。

変更点

- IBM MQ Operator 2.3.0 以降、144 ページの『[ネイティブ HA](#)』は IBM MQ Advanced ライセンスまたは IBM MQ Advanced for Developers ライセンスで使用できます。
- IBM MQ Operator に必要な許可のセットが削減されます。
- IBM MQ Operator 2.3.0 以降、IBM Cloud Pak for Integration デプロイメント用に作成された OperandRequest から **ibm-automation-core** が削除されました。
- IBM MQ Operator 2.3.0 以降、IBM MQ Operator デプロイメントでは **imagePullPolicy** として **IfNotPresent** が指定されています。
- 対処された脆弱性について詳しくは、以下のセキュリティ情報を参照してください。
 - [CVE-2022-47629 および CVE-2022-35737 の広報](#)
 - [CVE-2023-26284 の掲示板](#)

IBM MQ Operator 2.2.2

CD

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.4.1

オペレーター・チャンネル

v2.2

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.10 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これは偶数番号のマイナー・リリースです (例: 4.10 および 4.12)。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services バージョン 3.19 から 3.24 まで。

変更点

- 対処された脆弱性の詳細については、この[セキュリティ情報](#)を参照してください。

IBM MQ Operator 2.2.1

CD

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.4.1

オペレーター・チャンネル

v2.2

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.10 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これは偶数番号のマイナー・リリースです (例: 4.10 および 4.12)。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services バージョン 3.19 から 3.24 まで。

変更点

- ・対処された脆弱性の詳細については、この[セキュリティ情報](#)を参照してください。

IBM MQ Operator 2.2.0

CD

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.4.1

オペレーター・チャンネル

v2.2

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.10 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これは偶数番号のマイナー・リリースです (例: 4.10 および 4.12)。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services バージョン 3.19 から 3.24 まで。

新機能

- ・ IBM MQ Operator 2.2.0 (CD) 以降では、IBM Instana トレースはネイティブでサポートされます。9.3.1.0-r2 (CD) IBM MQ キュー・マネージャー・コンテナ・イメージでサポートを利用できます。9.3.1.0-r2 には IBM Instana MQ 出口のバージョン 2.4.0 (2022.4.0)が含まれます。IBM Instana トレースを使用可能にするには、[158 ページの『IBM MQ と IBM Instana トレースの統合』](#)を参照してください。

変更点

- ・対処された脆弱性の詳細については、この[セキュリティ情報](#)を参照してください。
- ・ IBM MQ Operator 2.2.0 以降、Operations Dashboard は非推奨になり、今後更新されることはありません。Operations Dashboard の新しい使用を開始してはなりません。IBM MQ 2.0.x オペレーターは引き続き Operations Dashboard をサポートします。

IBM MQ Operator 2.1.0

CD

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.2.1

オペレーター・チャンネル

v2.1

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.10 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これは偶数番号のマイナー・リリースです (例: 4.10 および 4.12)。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services 3.X、ただし少なくとも 3.19

新機能

- IBM MQ 9.3.1 を追加します。
- キュー・マネージャー仕様のデフォルト値更新を無効にすることをユーザーに許可する新しいオプションが追加されました。
- IBM MQ 9.3.1 より前のすべての IBM MQ バージョンを非推奨にする新しい状況条件を追加します。
- この CD バージョンの IBM MQ Operator で LTS オペランドを使用しているユーザーに警告する新しい状況条件を追加します。

変更点

- 対処された脆弱性の詳細については、この セキュリティ情報 を参照してください。

IBM MQ Operator 2.0.23 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.2.1

オペレーター・チャンネル

v2.0

.spec.version に許可される値

9.3.0.17-r3

マイ그레이ション中に許可される .spec.version の値

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 のバージョン

OpenShift Container Platform 4.12 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これらのリリースは、偶数番号のマイナー・リリース (4.14 や 4.16 など) です。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services 3.19

変更点

- 58 ページの『IBM MQ Operator 2.0.0』 上に構築されたセキュリティ更新
- IBM MQ カタログ・イメージが SQLite データベース・フォーマットからファイル・ベースのカタログ・フォーマットに移動されました。
- 対処された脆弱性の詳細については、この Security Bulletin を参照してください。

IBM MQ Operator 2.0.22 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.2.1

オペレーター・チャンネル

v2.0

.spec.version に許可される値

9.3.0.17-r2

マイ그레이ション中に許可される **.spec.version** の値

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 のバージョン

OpenShift Container Platform 4.12 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これらのリリースは、偶数番号のマイナー・リリース (4.14 や 4.16 など) です。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services 3.19

変更点

- 58 ページの『[IBM MQ Operator 2.0.0](#)』上に構築されたセキュリティーのみの更新
- 対処された脆弱性の詳細については、この [Security Bulletin](#) を参照してください。

IBM MQ Operator 2.0.21 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.2.1

オペレーター・チャンネル

v2.0

.spec.version に許可される値

9.3.0.17-r1

マイ그레이ション中に許可される **.spec.version** の値

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 のバージョン

OpenShift Container Platform 4.12 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これらのリリースは、偶数番号のマイナー・リリース (4.14 や 4.16 など) です。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services 3.19

変更点

- 58 ページの『[IBM MQ Operator 2.0.0](#)』上に構築されたセキュリティーのみの更新
- 対処された脆弱性の詳細については、この [Security Bulletin](#) を参照してください。

IBM MQ Operator 2.0.20 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.2.1

オペレーター・チャンネル

v2.0

.spec.version に許可される値

9.3.0.16-r2

マイグレーション中に許可される .spec.version の値

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 のバージョン

OpenShift Container Platform 4.12 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これらのリリースは、偶数番号のマイナー・リリース (4.14 や 4.16 など) です。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services 3.19

変更点

- 58 ページの『[IBM MQ Operator 2.0.0](#)』上に構築されたセキュリティーのみの更新
- 対処された脆弱性の詳細については、この [Security Bulletin](#) を参照してください。

IBM MQ Operator 2.0.19 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.2.1

オペレーター・チャンネル

v2.0

.spec.version に許可される値

9.3.0.16-r1

マイグレーション中に許可される .spec.version の値

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 のバージョン

OpenShift Container Platform 4.12 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これらのリリースは、偶数番号のマイナー・リリース (4.14 や 4.16 など) です。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services 3.19

変更点

- 58 ページの『[IBM MQ Operator 2.0.0](#)』上に構築されたセキュリティーのみの更新
- 対処された脆弱性について詳しくは、以下のセキュリティー情報を参照してください。
 - <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 バージョン

IBM Cloud Pak for Integration 2022.2.1

オペレーター・チャンネル

v2.0

.spec.version に許可される値

9.3.0.15-r1

マイ그레이ション中に許可される .spec.version の値

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 のバージョン

OpenShift Container Platform 4.12 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これらのリリースは、偶数番号のマイナー・リリース (4.14 や 4.16 など) です。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services 3.19

変更点

- [58 ページの『IBM MQ Operator 2.0.0』](#) 上に構築されたセキュリティーのみの更新
- 対処された脆弱性の詳細については、この[セキュリティー情報](#)を参照してください。

IBM MQ Operator 2.0.17 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.2.1

オペレーター・チャンネル

v2.0

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.12 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これらのリリースは、偶数番号のマイナー・リリース (4.14 や 4.16 など) です。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services 3.19

変更点

- [58 ページの『IBM MQ Operator 2.0.0』](#) 上に構築されたセキュリティーのみの更新
- 対処された脆弱性の詳細については、この[セキュリティー情報](#)を参照してください。
- IBM MQ Operator は、OpenShift Container Platform 4.10 ではテストもサポートもされなくなりました。

IBM MQ Operator 2.0.16 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.2.1

オペレーター・チャンネル

v2.0

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.12 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これらのリリースは、偶数番号のマイナー・リリース (4.14 や 4.16 など) です。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services 3.19

変更点

- [58 ページの『IBM MQ Operator 2.0.0』](#) 上に構築されたセキュリティーのみの更新
- 対処された脆弱性の詳細については、この[セキュリティー情報](#)を参照してください。
- IBM MQ Operator は、OpenShift Container Platform 4.10 ではテストもサポートもされなくなりました。

IBM MQ Operator 2.0.15 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.2.1

オペレーター・チャンネル

v2.0

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.10 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これは偶数番号のマイナー・リリースです (例: 4.10 および 4.12)。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services 3.19

変更点

- [58 ページの『IBM MQ Operator 2.0.0』](#) 上に構築されたセキュリティーのみの更新
- 対処された脆弱性の詳細については、この[セキュリティー情報](#)を参照してください。

IBM MQ Operator 2.0.14 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.2.1

オペレーター・チャンネル

v2.0

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.10 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これは偶数番号のマイナー・リリースです (例: 4.10 および 4.12)。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services 3.19

変更点

- [58 ページの『IBM MQ Operator 2.0.0』](#) 上に構築されたセキュリティーのみの更新
- 対処された脆弱性の詳細については、この[セキュリティー情報](#)を参照してください。

IBM MQ Operator 2.0.13 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.2.1

オペレーター・チャンネル

v2.0

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.10 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これは偶数番号のマイナー・リリースです (例: 4.10 および 4.12)。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services 3.19

変更点

- [58 ページの『IBM MQ Operator 2.0.0』](#) 上に構築されたセキュリティーのみの更新
- 対処された脆弱性の詳細については、この[セキュリティー情報](#)を参照してください。

IBM MQ Operator 2.0.12 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.2.1

オペレーター・チャンネル

v2.0

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.10 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これは偶数番号のマイナー・リリースです (例: 4.10 および 4.12)。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services 3.19

変更点

- 58 ページの『[IBM MQ Operator 2.0.0](#)』上に構築されたセキュリティーのみの更新
- 対処された脆弱性の詳細については、この[セキュリティー情報](#)を参照してください。

IBM MQ Operator 2.0.11 (LTS)

▶ CP4I-LTS

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.2.1

オペレーター・チャンネル

v2.0

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.10 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これは偶数番号のマイナー・リリースです (例: 4.10 および 4.12)。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services 3.19

変更点

- 58 ページの『[IBM MQ Operator 2.0.0](#)』上に構築されたセキュリティーのみの更新
- 対処された脆弱性の詳細については、この[セキュリティー情報](#)を参照してください。

IBM MQ Operator 2.0.10 (LTS)

▶ CP4I-LTS

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.2.1

オペレーター・チャンネル

v2.0

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.10 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これは偶数番号のマイナー・リリースです (例: 4.10 および 4.12)。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services 3.19

変更点

- [58 ページ](#)の『IBM MQ Operator 2.0.0』上に構築されたセキュリティーのみの更新
- 対処された脆弱性の詳細については、この[セキュリティー情報](#)を参照してください。

IBM MQ Operator 2.0.9 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.2.1

オペレーター・チャンネル

v2.0

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.10 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これは偶数番号のマイナー・リリースです (例: 4.10 および 4.12)。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services 3.19

新機能

- 2023 年 3 月以降、IBM MQ Operator および IBM MQ キュー・マネージャーのコンテナ・イメージはデジタル署名されます。IBM MQ Operator 2.0.9 および IBM MQ 9.3.0.4-r2 イメージは、このリリースで署名されています。[79 ページ](#)の『イメージ署名の検証』を参照

変更点

- [58 ページ](#)の『IBM MQ Operator 2.0.0』上に構築されたセキュリティーのみの更新
- 対処された脆弱性の詳細については、この[セキュリティー情報](#)を参照してください。

IBM MQ Operator 2.0.8 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.2.1

オペレーター・チャンネル

v2.0

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.10 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これは偶数番号のマイナー・リリースです (例: 4.10 および 4.12)。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services 3.19

変更点

- 58 ページの『[IBM MQ Operator 2.0.0](#)』上に構築されたセキュリティーのみの更新
- 対処された脆弱性について詳しくは、以下のセキュリティー情報を参照してください。
 - [CVE-2022-47629](#) および [CVE-2022-35737](#) の広報
 - [CVE-2023-26284](#) の掲示板

IBM MQ Operator 2.0.7 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.2.1

オペレーター・チャンネル

v2.0

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.10 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これは偶数番号のマイナー・リリースです (例: 4.10 および 4.12)。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services 3.19

変更点

- 58 ページの『[IBM MQ Operator 2.0.0](#)』上に構築されたセキュリティーのみの更新
- 対処された脆弱性の詳細については、この[セキュリティー情報](#)を参照してください。

IBM MQ Operator 2.0.6 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.2.1

オペレーター・チャンネル

v2.0

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.10 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これは偶数番号のマイナー・リリースです (例: 4.10 および 4.12)。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services 3.19

変更点

- 58 ページの『[IBM MQ Operator 2.0.0](#)』上に構築されたセキュリティーのみの更新
- 対処された脆弱性の詳細については、この[セキュリティー情報](#)を参照してください。

IBM MQ Operator 2.0.5 (LTS)

CP4I-LTS

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.2.1

オペレーター・チャンネル

v2.0

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.10 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これは偶数番号のマイナー・リリースです (例: 4.10 および 4.12)。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services 3.19

変更点

- [58 ページの『IBM MQ Operator 2.0.0』](#) 上に構築されたセキュリティのみの更新
- 対処された脆弱性の詳細については、この[セキュリティ情報](#)を参照してください。

IBM MQ Operator 2.0.4

CP4I-LTS

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.2.1

オペレーター・チャンネル

v2.0

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.10 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これは偶数番号のマイナー・リリースです (例: 4.10 および 4.12)。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services 3.19

変更点

- [58 ページの『IBM MQ Operator 2.0.0』](#) 上に構築されたセキュリティのみの更新
- 対処された脆弱性の詳細については、この[セキュリティ情報](#)を参照してください。

IBM MQ Operator 2.0.3

CD

CP4I-LTS

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.2.1

オペレーター・チャンネル

v2.0

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.10 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これは偶数番号のマイナー・リリースです (例: 4.10 および 4.12)。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services 3.19

変更点

- 58 ページの『[IBM MQ Operator 2.0.0](#)』上に構築されたセキュリティーのみの更新
- 対処された脆弱性の詳細については、この[セキュリティー情報](#)を参照してください。

IBM MQ Operator 2.0.2

CD CP4I-LTS

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.2.1

オペレーター・チャンネル
v2.0

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.10 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これは偶数番号のマイナー・リリースです (例: 4.10 および 4.12)。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services 3.19

変更点

- 58 ページの『[IBM MQ Operator 2.0.0](#)』上に構築されたセキュリティーのみの更新
- 対処された脆弱性の詳細については、この[セキュリティー情報](#)を参照してください。

IBM MQ Operator 2.0.1

CD CP4I-LTS

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.2.1

オペレーター・チャンネル
v2.0

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.10 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これは偶数番号のマイナー・リリースです (例: 4.10 および 4.12)。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services 3.19

変更点

- 58 ページの『IBM MQ Operator 2.0.0』上に構築されたセキュリティーのみの更新
- 対処された脆弱性の詳細については、この[セキュリティー情報](#)を参照してください。

IBM MQ Operator 2.0.0

CD CP4I-LTS

IBM Cloud Pak for Integration バージョン

IBM Cloud Pak for Integration 2022.2.1

オペレーター・チャンネル

v2.0

.spec.version に許可される値

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 のバージョン

OpenShift Container Platform 4.10 以上。注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これは偶数番号のマイナー・リリースです (例: 4.10 および 4.12)。

IBM Cloud Pak foundational services のバージョン

IBM Cloud Pak foundational services 3.19

新機能

- IBM MQ 9.3.0 を追加します。
- POWER (ppc64le) のサポートが追加されました。

変更点

- Red Hat OpenShift Container Platform 4.10 が必要になりました。 [11 ページの『IBM MQ Operator のバージョン・サポート』](#)を参照してください。
- **Deprecated** 非推奨バージョン: IBM MQ 9.2.3。これらのバージョンは、IBM MQ Operator の将来のバージョンでは調整されない可能性があります。
- **Removed** 削除された (以前は非推奨) 継続的デリバリー・リリース: IBM MQ 9.1.5、9.2.0 CD、9.2.1、9.2.2
- IBM MQ Operator 検証 Web フックは、Operator Lifecycle Manager (OLM) によってインストールされるようになりました。OLM は、Web フックの証明書を管理するようになりました。
- 以前に IBM MQ Console ログインでユーザー設定の警告が生成されていたバグが修正されました。
- 対処された脆弱性について詳しくは、以下のセキュリティー情報を参照してください。
 - <https://www.ibm.com/support/pages/node/6602255>
 - <https://www.ibm.com/support/pages/node/6602259>

OpenShift CP4I CD CP4I-LTS IBM MQ Operator で使用するキュー・マネージャー・コンテナー・イメージのリリース履歴

注: 以前のキュー・マネージャー・コンテナー・イメージについては、IBM MQ 9.2 資料の [IBM MQ Operator のリリース履歴](#) を参照してください。

9.3.5.1-r2

CD

必要なオペレーター・バージョン

3.1.3 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- 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

新機能

- [IBM MQ 9.3.5 の新機能](#)

変更点

- [IBM MQ 9.3.5 の変更内容](#)
- ベース [Red Hat Universal Base Image 8.9-1161.1715068733](#)
- golang.org/x/net ライブラリーは、報告された脆弱性を修復するためにアップグレードされました。

9.3.5.1-r1

CD

必要なオペレーター・バージョン

3.1.2 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- 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

新機能

- [IBM MQ 9.3.5 の新機能](#)

変更点

- [IBM MQ 9.3.5 の変更内容](#)
- [Red Hat Universal Base Image 8.9-1161](#) に基づく
- 「dependabot」がセキュリティーの脆弱性に対処したと報告

9.3.5.0-r2

CD

必要なオペレーター・バージョン

3.1.1 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- 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

新機能

- [IBM MQ 9.3.5 の新機能](#)

変更点

- [IBM MQ 9.3.5 の変更内容](#)
- [Red Hat Universal Base Image 8.9-1137](#) に基づく
- Operations Dashboard が有効になっている場合にのみ、新しい 9.3.5.0-r2 イメージをピックアップする必要があります。

9.3.5.0-r1



必要なオペレーター・バージョン

[3.1.0](#) 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- `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`

新機能

- [IBM MQ 9.3.5 の新機能](#)

変更点

- [IBM MQ 9.3.5 の変更内容](#)
- [Red Hat Universal Base Image 8.9-1137](#) に基づく
- シンボリック・リンクが `/var/mam` に提供され、`mqwebuser.xml` 内の暗号化されていない資格情報がコピーされます。
- [golang.org/x/crypto](#) ライブラリーは、CVE-2023-48795 の脆弱性を修復するためにアップグレードされました。
- Web 鍵ストアに自己署名証明書を作成するために、SHA256 の代わりによりセキュアな SHA512 アルゴリズムが使用されます。
- IBM MQ Web サーバーで使用する PKCS#12 鍵ストアは、**Pkcs12.Modern.Encode** 機能を使用して生成されるようになりました。この機能は、SHA-2 暗号化を使用します (以前は従来の SHA-1 暗号化を使用して生成されていました)。
- **PathTraversal** メソッドの使用に関して報告される脆弱性が修正されました。

9.3.4.1-r1



必要なオペレーター・バージョン

[3.0.1](#) 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- `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`

新機能

- [IBM MQ 9.3.4 の新機能](#)

変更点

- [IBM MQ 9.3.4 の変更内容](#)
- [Red Hat Universal Base Image 8.9-1108](#) に基づく

9.3.4.0-r1



必要なオペレーター・バージョン

3.0.0 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- `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`

新機能

- [IBM MQ 9.3.4 の新機能](#)

変更点

- [IBM MQ 9.3.4 の変更内容](#)
- [Red Hat Universal Base Image 8.9-1029](#) に基づく
- IBM MQ Web サーバーのサポートの向上- IBM MQ Web サーバー・ログがコンテナ・ログにデフォルトで表示されるようになりました。Web サーバーの `messages.log` ファイルが、コンテナ・ログ出力に自動的にミラーリングされるようになりました。この変更の一環として、ディスクに書き込まれる `messages.log` ファイルは常に JSON 形式になりましたが、コンテナ・ログは引き続き JSON または人間が判読可能な「基本」形式として使用可能です。
- キュー・マネージャー・コンテナ・イメージ内のシグナル処理が修正され、始動が完了する前にコンテナが Red Hat OpenShift Container Platform によって終了された場合に制御シグナルが正しく処理されるようになりました。

9.3.3.3-r2

必要なオペレーター・バージョン

2.4.8 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- `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`

新機能

- [IBM MQ 9.3.3 の新機能](#)

変更点

- [IBM MQ 9.3.3 の変更内容](#)
- [Red Hat Universal Base Image 8.9-1137](#) に基づく

- golang.org/x/crypto ライブラリーは、CVE-2023-48795 の脆弱性を修復するためにアップグレードされました。
- Web 鍵ストアに自己署名証明書を作成するために、SHA256 の代わりによりセキュアな SHA512 アルゴリズムが使用されます。
- IBM MQ Web サーバーで使用する PKCS#12 鍵ストアは、**Pkcs12.Modern.Encode** 機能を使用して生成されるようになりました。この機能は、SHA-2 暗号化を使用します (以前は従来の SHA-1 暗号化を使用して生成されていました)。
- **PathTraversal** メソッドの使用に関して報告される脆弱性が修正されました。

9.3.3.3-r1

必要なオペレーター・バージョン

[2.4.7](#) 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- 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

新機能

- [IBM MQ 9.3.3 の新機能](#)

変更点

- [IBM MQ 9.3.3 の変更内容](#)
- [Red Hat Universal Base Image 8.9-1108](#) に基づく

組み込まれている **IBM MQ APAR**

- IT44961
- IT44821
- IT44954

9.3.3.2-r3

CD

必要なオペレーター・バージョン

[2.4.6](#) 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- 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

新機能

- [IBM MQ 9.3.3 の新機能](#)

変更点

- [IBM MQ 9.3.3 の変更内容](#)
- [Red Hat Universal Base Image 8.9-1029](#) に基づく

9.3.3.2-r2

CD

必要なオペレーター・バージョン

2.4.5 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- `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`

新機能

- [IBM MQ 9.3.3 の新機能](#)

変更点

- [IBM MQ 9.3.3 の変更内容](#)
- [Red Hat Universal Base Image 8.8-1072.1697626218](#) に基づく
- IBM MQ キュー・マネージャー・コンテナ・イメージ 9.3.3.2-r2 には、[Instana MQ Exit](#) のバージョン 3.1.7 (2023.4.0) が含まれています。

9.3.3.2-r1

CD

必要なオペレーター・バージョン

2.4.4 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- `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`

新機能

- [IBM MQ 9.3.3 の新機能](#)

変更点

5

- [IBM MQ 9.3.3 の変更内容](#)
- [Red Hat Universal Base Image 8.8-1072.1697626218](#) に基づく
- libcurl のレベルを 8.4.0 に更新しました。

組み込まれている **IBM MQ APAR**

- IT41871
- IT44585
- IT44623
- IT44762

⁵ このトピックの前のバージョンでは、IBM MQ キュー・マネージャー・コンテナ・イメージ 9.3.3.2-r1 に [Instana MQ Exit](#) のバージョン 3.1.7 (2023.4.0) が含まれていることが誤って示されていました。

9.3.3.1-r2

CD

必要なオペレーター・バージョン

[2.4.3](#) 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- `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`

新機能

- [IBM MQ 9.3.3 の新機能](#)

変更点

- [IBM MQ 9.3.3 の変更内容](#)
- [IBM MQ 9.3.3.1-r1](#) で作成されたセキュリティーのみの更新
- [Red Hat Universal Base Image 8.8-1037](#) に基づいています。

9.3.3.1-r1

CD

必要なオペレーター・バージョン

[2.4.2](#) 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- `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`

新機能

- [IBM MQ 9.3.3 の新機能](#)

変更点

- [IBM MQ 9.3.3 の変更内容](#)
- [Red Hat Universal Base Image 8.8-1037](#) に基づいています。

9.3.3.0-r2

CD

必要なオペレーター・バージョン

[2.4.1](#) 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- `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`

新機能

- [IBM MQ 9.3.3 の新機能](#)

変更点

- [IBM MQ 9.3.3 の変更内容](#)
- [Red Hat Universal Base Image 8.8-1014](#) に基づいています。

9.3.3.0-r1

CD

必要なオペレーター・バージョン

[2.4.0](#) 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- `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`

新機能

- [IBM MQ 9.3.3 の新機能](#)

変更点

- [IBM MQ 9.3.3 の変更内容](#)
- [Red Hat Universal Base Image 8.8-860](#) に基づいています。
- IBM MQ キュー・マネージャー・コンテナー・イメージ 9.3.3.0-r1 には、[Instana MQ Exit のバージョン 3.1.2 \(2023.2.0\)](#)が含まれています。

9.3.2.1-r2

CD

必要なオペレーター・バージョン

[2.3.3](#) 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- `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`

新機能

- [IBM MQ 9.3.2 の新機能](#)

変更点

- [IBM MQ 9.3.2 の変更内容](#)
- [Red Hat Universal Base Image 8.7-1107](#) に基づいています。

9.3.2.1-r1

CD

必要なオペレーター・バージョン

2.3.2 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- 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

新機能

- [IBM MQ 9.3.2 の新機能](#)

変更点

- [IBM MQ 9.3.2 の変更内容](#)
- [Red Hat Universal Base Image 8.7-1107](#) に基づいています。

9.3.2.0-r2

CD

必要なオペレーター・バージョン

2.3.1 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- 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

新機能

- [IBM MQ 9.3.2 の新機能](#)

変更点

- [IBM MQ 9.3.2 の変更内容](#)
- [Red Hat Universal Base Image 8.7-1085](#) に基づいています。

9.3.2.0-r1

CD

必要なオペレーター・バージョン

2.3.0 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- 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

新機能

- [IBM MQ 9.3.2 の新機能](#)

- 環境変数 `MQ_LOGGING_CONSOLE_FORMAT` が設定されました。これは推奨されない `LOG_FORMAT` 変数に代わるものです。

変更点

- [IBM MQ 9.3.2 の変更内容](#)
- 発行者 (CA) 証明書と同じサブジェクト識別名 (DN) を持つキュー・マネージャー証明書はサポートされません。証明書には固有のサブジェクト識別名が必要です。
- [Red Hat Universal Base Image 8.7-1049.1675784874](#) に基づいています。

9.3.1.1-r1

CD

必要なオペレーター・バージョン

[2.2.2](#) 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- `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`

新機能

- [IBM MQ 9.3.1 の新機能](#)

変更点

- [IBM MQ 9.3.1 の変更内容](#)
- [Red Hat Universal Base Image 8.7-1031](#) に基づいています。
- IBM MQ キュー・マネージャー・コンテナ・イメージ 9.3.1.1-r1 には、[IBM Instana MQ 出口のバージョン 2.4.3 \(2022.4.3\)](#)が含まれています。

9.3.1.0-r3

CD

必要なオペレーター・バージョン

[2.2.1](#) 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- `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`

新機能

- [IBM MQ 9.3.1 の新機能](#)

変更点

- [IBM MQ 9.3.1 の変更内容](#)
- [Red Hat Universal Base Image 8.7-923.1669829893](#) に基づいています。
- IBM MQ キュー・マネージャー・コンテナ・イメージ 9.3.1.0-r3 には、[IBM Instana MQ 出口のバージョン 2.4.3 \(2022.4.3\)](#)が含まれています。

9.3.1.0-r2

CD

必要なオペレーター・バージョン
2.2.0 以上

サポートされているアーキテクチャー
amd64, s390x, ppc64le

イメージ

- `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`

新機能

- [IBM MQ 9.3.1 の新機能](#)
- 9.3.1.0-r2 (CD) IBM MQ キュー・マネージャー・コンテナ・イメージから、IBM Instana トレースはネイティブでサポートされます。IBM MQ バージョン 9.3.1.0-r2 には、[IBM Instana MQ 出口のバージョン 2.4.0 \(2022.4.0\)](#)が含まれています。IBM Instana トレースを使用可能にするには、[158 ページの『IBM MQ と IBM Instana トレースの統合』](#)を参照してください。

変更点

- [IBM MQ 9.3.1 の変更内容](#)
- [Red Hat Universal Base Image 8.7-923](#)に基づいています。
- 鍵と証明書が指定されていない場合、キュー・マネージャー属性 **SSLKEYR** は、「`/run/runmqserver/tls/key`」に設定されるのではなく、空白に設定されるようになりました。

9.3.1.0-r1

CD

必要なオペレーター・バージョン
2.1.0 以上

サポートされているアーキテクチャー
amd64, s390x, ppc64le

イメージ

- `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`

新機能

- [IBM MQ 9.3.1 の新機能](#)

変更点

- [IBM MQ 9.3.1 の変更内容](#)
- [Red Hat Universal Base Image 8.6-941](#)に基づいています。

9.3.0.17-r3

CP4I-LTS

必要なオペレーター・バージョン
2.0.22 以上

サポートされているアーキテクチャー
amd64, s390x, ppc64le

イメージ

- 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

新機能

- [IBM MQ 9.3.0 の新機能](#)

変更点

- [IBM MQ 9.3.0 の変更内容](#)
- IBM MQ 9.3.0.0-r1 に構築されたセキュリティー専用の更新
- [Red Hat Universal Base Image 9.4-949.1716471857](#) に基づく

9.3.0.17-r2

CP4I-LTS

必要なオペレーター・バージョン

[2.0.22](#) 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- 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

新機能

- [IBM MQ 9.3.0 の新機能](#)

変更点

- [IBM MQ 9.3.0 の変更内容](#)
- IBM MQ 9.3.0.0-r1 に構築されたセキュリティー更新
- ベース [Red Hat Universal Base Image 8.9-1161.1715068733](#)
- golang.org/x/net ライブラリーは、報告された脆弱性を修復するためにアップグレードされました。

9.3.0.17-r1

CP4I-LTS

必要なオペレーター・バージョン

[2.0.21](#) 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- 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

新機能

- [IBM MQ 9.3.0 の新機能](#)

変更点

- [IBM MQ 9.3.0 の変更内容](#)
- IBM MQ 9.3.0.0-r1 に構築されたセキュリティー更新
- [Red Hat Universal Base Image 8.9-1161](#) に基づく
- 「dependabot」は、セキュリティーの脆弱性が対処されたと報告しています。

9.3.0.16-r2

CP4I-LTS

必要なオペレーター・バージョン

[2.0.20](#) 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- `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`

新機能

- [IBM MQ 9.3.0 の新機能](#)

変更点

- [IBM MQ 9.3.0 の変更内容](#)
- IBM MQ 9.3.0.0-r1 に構築されたセキュリティー更新
- [Red Hat Universal Base Image 8.9-1137](#) に基づく
- オペレーション・ダッシュボードが有効になっている場合にのみ、新しい 9.3.0.16-r2 イメージをピックアップする必要があります。

9.3.0.16-r1

CP4I-LTS

必要なオペレーター・バージョン

[2.0.19](#) 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- `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`

新機能

- [IBM MQ 9.3.0 の新機能](#)

変更点

- [IBM MQ 9.3.0 の変更内容](#)
- IBM MQ 9.3.0.0-r1 に構築されたセキュリティー更新
- [Red Hat Universal Base Image 8.9-1137](#) に基づく
- [golang.org/x/crypto](#) ライブラリーは、CVE-2023-48795 の脆弱性を修復するためにアップグレードされました。

- Web 鍵ストアに自己署名証明書を作成するために、SHA256 の代わりによりセキュアな SHA512 アルゴリズムが使用されます。
- IBM MQ Web サーバーで使用する PKCS#12 鍵ストアは、**Pkcs12.Modern.Encode** 機能を使用して生成されるようになりました。この機能は、SHA-2 暗号化を使用します (以前は従来の SHA-1 暗号化を使用して生成されていました)。
- **PathTraversal** メソッドの使用に関して報告される脆弱性が修正されました。

9.3.0.15-r1

CP4I-LTS

必要なオペレーター・バージョン

2.0.18 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- `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`

新機能

- [IBM MQ 9.3.0 の新機能](#)

変更点

- [IBM MQ 9.3.0 の変更内容](#)
- [IBM MQ 9.3.0.0-r1](#) で構築されたセキュリティーのみの更新
- [Red Hat Universal Base Image 8.9-1108](#) に基づく

9.3.0.11-r2

CP4I-LTS

必要なオペレーター・バージョン

2.0.17 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- `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`

新機能

- [IBM MQ 9.3.0 の新機能](#)

変更点

- [IBM MQ 9.3.0 の変更内容](#)
- [IBM MQ 9.3.0.0-r1](#) で構築されたセキュリティーのみの更新
- [Red Hat Universal Base Image 8.9-1029](#) に基づいています。

9.3.0.11-r1

CP4I-LTS

必要なオペレーター・バージョン

2.0.16 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- [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](#)

新機能

- [IBM MQ 9.3.0 の新機能](#)

変更点

- [IBM MQ 9.3.0 の変更内容](#)
- [IBM MQ 9.3.0.0-r1](#) で構築されたセキュリティーのみの更新
- [Red Hat Universal Base Image 8.8-1072.1697626218](#) に基づいています。
- libcurl のレベルを 8.4.0 に更新しました。

9.3.0.10-r2

CP4I-LTS

必要なオペレーター・バージョン

2.0.15 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- [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](#)

新機能

- [IBM MQ 9.3.0 の新機能](#)

変更点

- [IBM MQ 9.3.0 の変更内容](#)
- [IBM MQ 9.3.0.0-r1](#) で構築されたセキュリティーのみの更新
- [Red Hat Universal Base Image 8.8-1037](#) に基づいています。

9.3.0.10-r1

CP4I-LTS

必要なオペレーター・バージョン

2.0.14 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- [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](#)

新機能

- [IBM MQ 9.3.0 の新機能](#)

変更点

- [IBM MQ 9.3.0 の変更内容](#)
- [IBM MQ 9.3.0.0-r1](#) で構築されたセキュリティーのみの更新
- [Red Hat Universal Base Image 8.8-1037](#) に基づいています。

9.3.0.6-r1

CP4I-LTS

必要なオペレーター・バージョン

[2.0.13](#) 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- `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`

新機能

- [IBM MQ 9.3.0 の新機能](#)

変更点

- [IBM MQ 9.3.0 の変更内容](#)
- [IBM MQ 9.3.0.0-r1](#) で構築されたセキュリティーのみの更新
- [Red Hat Universal Base Image 8.8-1014](#) に基づいています。

9.3.0.5-r3

CP4I-LTS

必要なオペレーター・バージョン

[2.0.12](#) 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- `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`

新機能

- [IBM MQ 9.3.0 の新機能](#)

変更点

- [IBM MQ 9.3.0 の変更内容](#)
- [IBM MQ 9.3.0.0-r1](#) で構築されたセキュリティーのみの更新
- [Red Hat Universal Base Image 8.8-860](#) に基づいています。

9.3.0.5-r2

CP4I-LTS

必要なオペレーター・バージョン

[2.0.11](#) 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- [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](#)

新機能

- [IBM MQ 9.3.0 の新機能](#)

変更点

- [IBM MQ 9.3.0 の変更内容](#)
- [IBM MQ 9.3.0.0-r1](#) で構築されたセキュリティーのみの更新
- [Red Hat Universal Base Image 8.7-1107](#) に基づいています。

重要: IBM MQ LTS Queue Manager Container Image 9.3.0.5-r2 の Operations Dashboard のユーザー向け

Operations Dashboard が有効になると、IBM MQ LTS Queue Manager Container Image 9.3.0.5-r2 は、一般出荷版の時点で使用可能な最新のセキュリティー・フィックスが含まれていない Operations Dashboard Agent イメージおよび Collector イメージをデプロイします。

緩和: Operations Dashboard を有効にして、少なくとも 9.3.0.5-r3 すべての IBM MQ LTS Queue Manager Container 9.3.0.5-r2 イメージにアップグレードします。132 ページの『[Red Hat OpenShift を使用した IBM MQ キュー・マネージャーのアップグレード](#)』を参照してください。

9.3.0.5-r1

CP4I-LTS

必要なオペレーター・バージョン

[2.0.10](#) 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- [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](#)

新機能

- [IBM MQ 9.3.0 の新機能](#)

変更点

- [IBM MQ 9.3.0 の変更内容](#)
- [IBM MQ 9.3.0.0-r1](#) で構築されたセキュリティーのみの更新
- [Red Hat Universal Base Image 8.7-1107](#) に基づいています。

重要: IBM MQ LTS Queue Manager Container Image 9.3.0.5-r1 上の Operations Dashboard のユーザー向け

Operations Dashboard が有効になると、IBM MQ LTS Queue Manager Container Image 9.3.0.5-r1 は、一般出荷版の時点で使用可能な最新のセキュリティー・フィックスが含まれていない Operations Dashboard Agent イメージおよび Collector イメージをデプロイします。

緩和: Operations Dashboard が有効になっている状態で、少なくとも 9.3.0.5-r3 すべての IBM MQ LTS Queue Manager Container 9.3.0.5-r1 イメージにアップグレードします。[132 ページの『Red Hat OpenShift を使用した IBM MQ キュー・マネージャーのアップグレード』](#)を参照してください。

9.3.0.4-r2

CP4I-LTS

必要なオペレーター・バージョン

[2.0.9](#) 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- `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`

新機能

- [IBM MQ 9.3.0 の新機能](#)

変更点

- [IBM MQ 9.3.0 の変更内容](#)
- [IBM MQ 9.3.0.0-r1](#) で構築されたセキュリティーのみの更新
- [Red Hat Universal Base Image 8.7-1085](#) に基づいています。

9.3.0.4-r1

CP4I-LTS

必要なオペレーター・バージョン

[2.0.8](#) 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- `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`

新機能

- [IBM MQ 9.3.0 の新機能](#)

変更点

- [IBM MQ 9.3.0 の変更内容](#)
- [IBM MQ 9.3.0.0-r1](#) で構築されたセキュリティーのみの更新
- [Red Hat Universal Base Image 8.7-1049.1675784874](#) に基づいています。

9.3.0.3-r1

CP4I-LTS

必要なオペレーター・バージョン

2.0.7 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- [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](#)

新機能

- [IBM MQ 9.3.0 の新機能](#)

変更点

- [IBM MQ 9.3.0 の変更内容](#)
- [IBM MQ 9.3.0.0-r1](#) で構築されたセキュリティーのみの更新
- [Red Hat Universal Base Image 8.7-1031](#) に基づいています。

9.3.0.1-r4

CP4I-LTS

必要なオペレーター・バージョン

2.0.6 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- [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](#)

新機能

- [IBM MQ 9.3.0 の新機能](#)

変更点

- [IBM MQ 9.3.0 の変更内容](#)
- [IBM MQ 9.3.0.0-r1](#) で構築されたセキュリティーのみの更新
- [Red Hat Universal Base Image 8.7-923.1669829893](#) に基づいています。

9.3.0.1-r3

CP4I-LTS

必要なオペレーター・バージョン

2.0.5 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- [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](#)

新機能

- [IBM MQ 9.3.0 の新機能](#)

変更点

- [IBM MQ 9.3.0 の変更内容](#)
- [IBM MQ 9.3.0.0-r1](#) で構築されたセキュリティーのみの更新
- [Red Hat Universal Base Image 8.7-923](#) に基づいています。

9.3.0.1-r2

CP4I-LTS

必要なオペレーター・バージョン

2.0.4 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- `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`

新機能

- [IBM MQ 9.3.0 の新機能](#)

変更点

- [IBM MQ 9.3.0 の変更内容](#)
- [IBM MQ 9.3.0.0-r1](#) で構築されたセキュリティーのみの更新
- [Red Hat Universal Base Image 8.6-941](#) に基づいています。

9.3.0.1-r1

CD

CP4I-LTS

必要なオペレーター・バージョン

2.0.3 以上

サポートされているアーキテクチャー

amd64, s390x, ppc64le

イメージ

- `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`

新機能

- [IBM MQ 9.3.0 の新機能](#)

変更点

- [IBM MQ 9.3.0 の変更内容](#)
- [IBM MQ 9.3.0.0-r1](#) で構築されたセキュリティーのみの更新
- [Red Hat Universal Base Image 8.6-941](#) に基づいています。

9.3.0.0-r3

CD CP4I-LTS

必要なオペレーター・バージョン
2.0.2 以上

サポートされているアーキテクチャー
amd64, s390x, ppc64le

イメージ

- `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`

新機能

- [IBM MQ 9.3.0 の新機能](#)

変更点

- [IBM MQ 9.3.0 の変更内容](#)
- [IBM MQ 9.3.0.0-r1](#) で構築されたセキュリティーのみの更新
- [Red Hat Universal Base Image 8.6-902](#) に基づいています。

9.3.0.0-r2

CD CP4I-LTS

必要なオペレーター・バージョン
2.0.1 以上

サポートされているアーキテクチャー
amd64, s390x, ppc64le

イメージ

- `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`

新機能

- [IBM MQ 9.3.0 の新機能](#)

変更点

- [IBM MQ 9.3.0 の変更内容](#)
- [IBM MQ 9.3.0.0-r1](#) で構築されたセキュリティーのみの更新
- [Red Hat Universal Base Image 8.6-854](#) に基づいています。

9.3.0.0-r1

CD CP4I-LTS

必要なオペレーター・バージョン
2.0.0 以上

サポートされているアーキテクチャー
amd64, s390x, ppc64le

イメージ

- `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

新機能

- [IBM MQ 9.3.0 の新機能](#)

変更点

- [IBM MQ 9.3.0 の変更内容](#)
- MQ Advanced for Developers イメージのデフォルトの開発者構成で ANY_TLS12_OR_HIGHER が使用されるようになりました。
- Java 設定がないことが原因でログにエラーが発生する IBM MQ Web サーバーの問題が修正されました。
- [Red Hat Universal Base Image 8.6-751.1655117800](#) に基づいています。

OpenShift CP4I イメージ署名の検証

2023 年 3 月以降、IBM MQ Operator および IBM MQ キュー・マネージャーのコンテナ・イメージはデジタル署名されます。

署名する最初の IBM MQ 個の演算子:

- 2.3.1 (CD)
- 2.0.9 (LTS)

署名される最初の IBM MQ キュー・マネージャー・コンテナ・イメージ:

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

このタスクについて

デジタル署名により、コンテンツの利用者は、ダウンロードしたものが本物 (期待されるソースからのもの) であり、整合性があること (期待されるもの) を確認することができます。

手順

- IBM MQ Operator および IBM MQ キュー・マネージャー・コンテナ・イメージの署名を検査します。
 - [Operator 3.0.0](#) [V 9.3.4](#) 3.0.0 以降の IBM MQ Operator、または 9.3.4.0-r1 以降の IBM MQ キュー・マネージャー・コンテナ・イメージについては、IBM Cloud Pak for Integration (CP4I) 2023.4 資料の [イメージ署名の検証](#) を参照してください。
 - [Operator 2.4.0](#) 2.4.x の IBM MQ Operator、または 9.3.3.x の IBM MQ キュー・マネージャー・コンテナ・イメージについては、CP4I 2023.2 資料の [イメージ署名の検証](#) を参照してください。
 - 2.4.0 より前の IBM MQ Operator、または 9.3.3.0-r1 より前の IBM MQ キュー・マネージャー・コンテナ・イメージについては、CP4I 2022.4 資料の [イメージの署名の検証](#) を参照してください。

OpenShift CD CP4I-LTS IBM Cloud Pak for Integration への IBM MQ のマ イグレーション

この一連のトピックでは、IBM Cloud Pak for Integration の IBM MQ Operator を使用して、既存の IBM MQ キュー・マネージャーをコンテナ環境にマイグレーションするための主要なステップについて説明します。

このタスクについて

IBM MQ on Red Hat OpenShift で展開するクライアントは、以下のシナリオに分けることができる。

1. 新規アプリケーションのための新しい IBM MQ デプロイメントを Red Hat OpenShift に作成する。

2. Red Hat OpenShift 内の新規アプリケーションのために IBM MQ ネットワークを Red Hat OpenShift に拡張します。
3. 既存のアプリケーションを引き続きサポートするには、IBM MQ デプロイメントを Red Hat OpenShift に移動します。

ご使用の IBM MQ 構成をマイグレーションする必要があるのは、シナリオ 3 の場合のみです。その他のシナリオは、新規デプロイメントと見なされます。

この一連のトピックでは、シナリオ 3 に焦点を当て、IBM MQ Operator を使用して既存の IBM MQ キュー・マネージャーをコンテナ環境にマイグレーションするための重要なステップについて説明します。IBM MQ は柔軟性と拡張性に優れているため、オプションで行える手順がいくつかあります。これらの各手順に「これを行う必要がありますか?」セクションがあります。社内でどのようなニーズがあるかを確認しておくなら、マイグレーション時に時間を節約することができるはずです。

どのデータをマイグレーションするか検討することも必要です。

1. 同じ構成で IBM MQ をマイグレーションするが、キューに入っている既存のメッセージは含めない。
2. 同じ構成で IBM MQ をマイグレーションし、既存のメッセージも含める。

標準的なバージョン間マイグレーションでは、どちらのアプローチでも使用できます。IBM MQ キュー・マネージャーの標準的なマイグレーションでは、マイグレーション時にキューにメッセージが保管されていたとしてもわずかなので、多くの場合はオプション 1 が適切です。コンテナ・プラットフォームへのマイグレーションでは、それにも増してオプション 1 を使用することが一般的になっています。これは、マイグレーションの複雑な手順をなくして、ブルー・グリーン・デプロイメントを行えるようにするためです。そのため、このシナリオに焦点を当てて説明します。

このシナリオの目的は、既存のキュー・マネージャーの定義と一致するキュー・マネージャーをコンテナ環境内に作成することです。この方法では、ネットワークに接続された既存のアプリケーションは、新しいキュー・マネージャーを指すように再構成するだけで済みますので、他の構成やアプリケーション・ロジックを変更する必要はありません。

このマイグレーション全体を通して、新しいキュー・マネージャーに適用される複数の構成ファイルを生成します。これらのファイルの管理を簡素化するために、ディレクトリーを 1 つ作成し、生成したファイルはそのディレクトリーに入れるようにしてください。

手順

1. [80 ページの『必要な機能を利用できることの確認』](#)
2. [81 ページの『キュー・マネージャー構成の抽出』](#)
3. オプション: [82 ページの『オプション: キュー・マネージャーの鍵と証明書の抽出および取得』](#)
4. オプション: [84 ページの『オプション: LDAP の構成』](#)
5. オプション: [91 ページの『オプション: IBM MQ 構成内の IP アドレスとホスト名の変更』](#)
6. [93 ページの『コンテナ環境用のキュー・マネージャー構成の更新』](#)
7. [95 ページの『コンテナで実行されている IBM MQ のためのターゲット HA アーキテクチャーの選択』](#)
8. [96 ページの『キュー・マネージャー用のリソースの作成』](#)
9. [97 ページの『Red Hat OpenShift での新しいキュー・マネージャーの作成』](#)
10. [101 ページの『新規コンテナ・デプロイメントの検証』](#)

OpenShift CD CP4I-LTS **必要な機能を利用できることの確認**

IBM MQ Operator には、IBM MQ Advanced 内で使用可能な機能がすべて含まれているわけではないので、除外されている機能が必要ないことを確認する必要があります。その他の機能は部分的にサポートされており、コンテナ内で使用可能なものと一致するように再構成することもできます。

始める前に

これは、[79 ページの『IBM Cloud Pak for Integration への IBM MQ のマイグレーション』](#)の最初のステップです。

手順

1. 必要なすべての機能がターゲット・コンテナ・イメージに含まれていることを確認します。
最新情報については、5 ページの『[コンテナ内の IBM MQ の使用方法の選択](#)』を参照してください。
2. IBM MQ Operator には、リスナーと呼ばれる IBM MQ トラフィック・ポートが1つあります。複数のリスナーがある場合は、これを単純化して、コンテナで1つのリスナーを使用するようにします。これは一般的なシナリオではないため、この変更についての詳細な説明は行いません。
3. IBM MQ 出口が使用されている場合は、IBM MQ 出口バイナリー内で階層化することにより、それらをコンテナにマイグレーションします。これは上級のマイグレーション・シナリオであるため、ここには記載しません。手順の概要については、166 ページの『[Red Hat OpenShift CLI を使用した、カスタム MQSC および INI ファイルを使用したイメージの作成](#)』を参照してください。
4. IBM MQ システムに高可用性が設定されている場合は、使用可能なオプションを確認します。
25 ページの『[コンテナ内の IBM MQ の高可用性](#)』を参照してください。

次のタスク

これで、キュー・マネージャー構成を抽出する準備ができました。

OpenShift CD CP4I-LTS キュー・マネージャー構成の抽出

構成の大部分は、キュー・マネージャー間で移植可能です。例えば、アプリケーションが対話する内容 (キュー、トピック、およびチャネルの定義など) です。既存の IBM MQ キュー・マネージャーから構成を抽出するには、このタスクを使用します。

始める前に

このタスクでは、[必要な機能が使用可能であることを確認済みであることを前提](#)としています。

手順

1. 既存の IBM MQ インストール済み環境があるマシンにログインします。
2. 構成のバックアップをとります。

以下のコマンドを実行します。

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

このコマンドの使用上の注意:

- このコマンドは、バックアップを tmp ディレクトリーに保管します。別の場所にバックアップを保管することもできますが、このシナリオの以下のコマンドでは、tmp ディレクトリーを使用することを想定しています。
- QMGR_NAME は、ご使用の環境のキュー・マネージャー名に置き換えてください。値が分からない場合は、**dspm** コマンドを実行して、このマシンで使用可能なキュー・マネージャーを表示します。ここでは、qm1 という名前のキュー・マネージャーの **dspm** コマンド出力例を示します。

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

dspm コマンドを実行するには、IBM MQ キュー・マネージャーが開始している必要があります。開始していない場合は、次のエラーを受け取ります。

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

必要に応じて、次のコマンドを実行してキュー・マネージャーを開始します。

```
strmqm QMGR_NAME
```

次のタスク

これで、キュー・マネージャーの鍵と証明書を抽出して取得する準備ができました。

OpenShift CD CP4I-LTS オプション: キュー・マネージャーの鍵と証明書の抽出および取得

TLS を使用してキュー・マネージャーへのトラフィックを暗号化するように IBM MQ を構成することができます。このタスクを使用して、キュー・マネージャーが TLS を使用しているかどうかの検証、鍵と証明書の抽出、マイグレーション済みのキュー・マネージャーでの TLS の構成を行います。

始める前に

このタスクでは、キュー・マネージャー構成を抽出済みであることを前提としています。

このタスクについて

これを行う必要がありますか？

キュー・マネージャーへのトラフィックを暗号化するように IBM MQ を構成することができます。この暗号化には、キュー・マネージャー上で構成された鍵リポジトリの使用が欠かせません。IBM MQ チャネルはそれを使って TLS 通信を有効にします。鍵リポジトリがご使用の環境で構成済みであるか分からない場合は、次のコマンドを実行して確認してください。

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

結果が表示されない場合、TLS は使用されていません。しかし、このことはマイグレーション済みのキュー・マネージャーで TLS を構成できないことを意味するものではありません。以下の状況では、この状態を変更することが必要となる場合があります。

- Red Hat OpenShift 環境に対するセキュリティー・アプローチは、前の環境と比較して拡張する必要があります。
- Red Hat OpenShift 環境の外部からマイグレーション済みキュー・マネージャーにアクセスする必要がある場合は、Red Hat OpenShift ルートを通過するために TLS が必要です。

注: **V 9.3.2** 発行者 (CA) 証明書と同じサブジェクト識別名 (DN) を持つキュー・マネージャー証明書はサポートされません。証明書には固有のサブジェクト識別名が必要です。製品は、DN が同じではないことを検査するようになりました。

手順

1. 既存のストアから信頼できる証明書をすべて抽出します。

現在、キュー・マネージャーで TLS を使用している場合、キュー・マネージャーにいくつかのトラステッド証明書が保管されている可能性があります。これらを抽出して、新しいキュー・マネージャーにコピーする必要があります。以下のいずれかのオプションの手順を実行します。

- 証明書の抽出を簡素化するには、ローカル・システム上で以下のスクリプトを実行します。

```
#!/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
```

```
done
fi
```

このスクリプトを実行するときに、IBM MQ バックアップの場所を引数として指定すると、証明書が抽出されます。例えば、スクリプトが `extractCert.sh` という名前で、IBM MQ バックアップが `/tmp/backup.mqsc` にある場合は、以下のコマンドを実行します。

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

- または、以下のコマンドを上から順に実行します。

- a. TLS ストアの場所を特定します。

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

出力例は次のとおりです。

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

ここで、鍵ストアは `/run/runmqserver/tls/key.kdb` にあります

- b. このロケーション情報に基づいて鍵ストアを照会し、保管されている証明書を判別します。

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

出力例は次のとおりです。

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

- c. リストされた各証明書を抽出します。これを行うには、以下のコマンドを実行します。

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

直前に示した例は、次のコマンドに相当します。

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

2. キュー・マネージャーの新しい鍵と証明書を取得します。

マイグレーション済みのキュー・マネージャー上で TLS を構成するには、新しい鍵と証明書を生成します。これが後でデプロイメント時に使用されます。多くの組織では、このためにセキュリティー・チームに連絡して鍵と証明書を提供してもらうことが必要になります。組織によっては、このオプションが使えないため、自己署名証明書を使用します。

以下の例では、有効期限を 10 年に設定して自己署名証明書を生成します。

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

次の 2 つの新規ファイルが作成されます。

- `qmgr.key` は、キュー・マネージャーの秘密鍵です
- `qmgr.crt` はパブリック証明書です

次のタスク

これで、LDAP を構成する準備ができました。

OpenShift CD CP4I-LTS オプション: LDAP の構成

IBM MQ Operator は、複数の異なるセキュリティー・アプローチを使用するように構成できます。通常、LDAP はエンタープライズ・デプロイメントに最も効果的なので、このマイグレーション・シナリオでは LDAP を使用します。

始める前に

このタスクでは、キュー・マネージャーの鍵と証明書を抽出して取得済みであることを前提としています。

このタスクについて

これを行う必要がありますか？

認証と許可に LDAP を既に使用している場合は、変更する必要はありません。

LDAP を使用しているかどうか分からない場合は、次のコマンドを実行します。

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

出力例は次のとおりです。

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

出力の中に、特に注目できる 2 つの属性があります。

AUTHTYPE

ここに値 IDPWLDAP が設定されている場合は、認証に LDAP を使用しています。

この値が空白であるか、または別の値である場合は、LDAP が構成されていません。この場合、許可に LDAP ユーザーが使用されているかどうかを確認するために、AUTHORMD 属性を確認してください。

AUTHORMD

ここに値 OS が設定されている場合は、許可に LDAP を使用していません。

LDAP を使用するように許可と認証を変更するには、以下の作業を行います。

手順

1. LDAP サーバーの IBM MQ バックアップを更新します。
2. LDAP 許可情報の IBM MQ バックアップを更新します。

アップの更新

LDAP をセットアップする方法についての包括的な説明は、このシナリオでは扱われません。このトピックでは、プロセスの要約、サンプル、および詳細情報の参照先を示します。

始める前に

このタスクでは、[キュー・マネージャーの鍵と証明書を抽出して取得済みであることを前提として](#)います。

このタスクについて

これを行う必要がありますか？

認証と許可に LDAP を既に使用している場合は、変更する必要はありません。LDAP を使用しているかどうか分からない場合は、[84 ページの『オプション: LDAP の構成』](#)を参照してください。

LDAP サーバーのセットアップには 2 つの段階があります。

1. LDAP 構成を定義します。
2. LDAP 構成をキュー・マネージャー定義に関連付けます。

この構成について詳しくは、以下を参照してください。

- [ユーザー・リポジトリの概要](#)
- [AUTHINFO コマンドの参照ガイド](#)

手順

1. LDAP 構成を定義します。

backup.mqsc ファイルを編集して、LDAP システム用の新しい **AUTHINFO** オブジェクトを定義します。以下に例を示します。

```
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
```

説明:

- **CONNAME** は、LDAP サーバーに対応するホスト名とポートです。回復力を高めるために複数のアドレスが用意されている場合は、これらをコンマ区切りリストにして構成することができます。
- **LDAPUSER** は、IBM MQ が LDAP に接続してユーザー・レコードを照会するとき使用するユーザーに対応する識別名です。
- **LDAPPWD** は、**LDAPUSER** ユーザーに対応するパスワードです。
- **SECCOM** は、LDAP サーバーへの通信に TLS を使用する必要があるかどうかを指定します。指定可能な値:
 - YES: TLS を使用し、証明書は IBM MQ サーバーによって提供されます。
 - ANON: TLS を使用しますが、証明書は IBM MQ サーバーによって提供されません。
 - NO: 接続中に TLS を使用しません。

- **USRFIELD** は、提示されたユーザー名の突き合わせに使用する LDAP レコード内のフィールドを指定します。
- **SHORTUSR** は、LDAP レコード内の長さが 12 文字を超えないフィールドです。認証に成功すると、このフィールド内の値は表明された ID になります。
- **BASEDNU** は、LDAP の検索に使用する必要がある基本 DN です。
- **BASEDNG** は、LDAP 内のグループの基本 DN です。
- **AUTHORMD** は、ユーザーのグループ・メンバーシップを解決するために使用するメカニズムを定義します。次の 4 つのオプションがあります。
 - OS: 短い名前に関連付けられているグループのオペレーティング・システムを照会します。
 - SEARCHGRP: LDAP 内のグループ・エントリーから認証済みユーザーを検索します。
 - SEARCHUSR: 認証済みユーザー・レコードからグループ・メンバーシップ情報を検索します。
 - SRCHGRPSN: LDAP 内のグループ・エントリーから、認証済みユーザーの短いユーザー名 (SHORTUSR フィールドで定義する) を検索します。
- **GRPFIELD** は、単純名に対応する LDAP グループ・レコード内の属性です。これを指定すると、許可レコードの定義に使用できます。
- **CLASSUSR** は、ユーザーに対応する LDAP オブジェクト・クラスです。
- **CLASSGRP** は、グループに対応する LDAP オブジェクト・クラスです。
- **FINDGRP** は、グループ・メンバーシップに対応する LDAP レコード内の属性です。

新しい項目はファイル内の任意の場所に置くことができますが、新しい項目をファイルの先頭に置くと便利です。

```
Open ▾ [icon]
backup.mqsc
*****
* Script generated on 2020-10-21 at 11.48.32
* Script generated by user ' CallumJackso' on host 'LAPTOP-VLQ
* 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) +
```

2. LDAP 構成をキュー・マネージャー定義に関連付けます。

LDAP 構成をキュー・マネージャー定義に関連付ける必要があります。DEFINE AUTHINFO 項目のすぐ下に ALTER QMGR 項目があります。新しく作成された AUTHINFO 名に対応するように CONNAUTH 項目を変更します。例えば直前の例では、AUTHINFO(USE.LDAP) が定義されています。これは、名前が USE.LDAP であることを示しています。それで、CONNAUTH('SYSTEM.DEFAULT.AUTHINFO.IDPWOS') を CONNAUTH('USE.LDAP') に変更します。

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

LDAP への切り替えがすぐに行われるように、ALTER QMGR コマンドの直後に行を追加して、REFRESH SECURITY コマンドを呼び出します。


```

*backup.mqsc
*****
* Script generated on 2020-10-21 at 11.48.32
* Script generated by user ' CallumJackso' on host 'LAPTOP-VLQKJ5UH'
* Queue manager name: qm1
* Queue manager platform: Windows
* Queue manager command level: (920/920)
* Command issued: dmpmqcfc -m qm1
*****
DEFINE AUTHINFO(USE.LDAP) +
  AUTHTYPE(IDPWLDAP) +
  CONNAME('ldap-service.ldap(389)') +
  LDAPUSER('cn=admin,dc=ibm,dc=com') +
  LDAPPWD('admin') +
  SECCOMM(NO) +
  USRFIELD('uid') +
  SHORTUSR('uid') +
  BASEDNU('ou=people,dc=ibm,dc=com') +
  AUTHORMD(SEARCHGRP) +
  BASEDNG('ou=groups,dc=ibm,dc=com') +
  GRPFIELD('cn') +
  CLASSGRP('groupOfUniqueNames') +
  FINDGRP('uniqueMember') +
  REPLACE
ALTER QMGR +
* 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

```

次のタスク

これで、LDAP 許可情報の IBM MQ バックアップを更新する準備ができました。

OpenShift > CD > CP4I-LTS **LDAP パート 2: LDAP 許可情報の IBM MQ バックアップの更新**

IBM MQ には、IBM MQ オブジェクトへのアクセスを制御する、細分化された許可規則が用意されています。この認証および許可を LDAP に変更すると、許可規則が無効になって更新が必要になる場合があります。

始める前に

このタスクでは、LDAP サーバーのバックアップを更新済みであることを前提としています。

このタスクについて

これを行う必要がありますか？

認証と許可に LDAP を既に使用している場合は、変更する必要はありません。LDAP を使用しているかどうか分からない場合は、[84 ページの『オプション: LDAP の構成』](#)を参照してください。

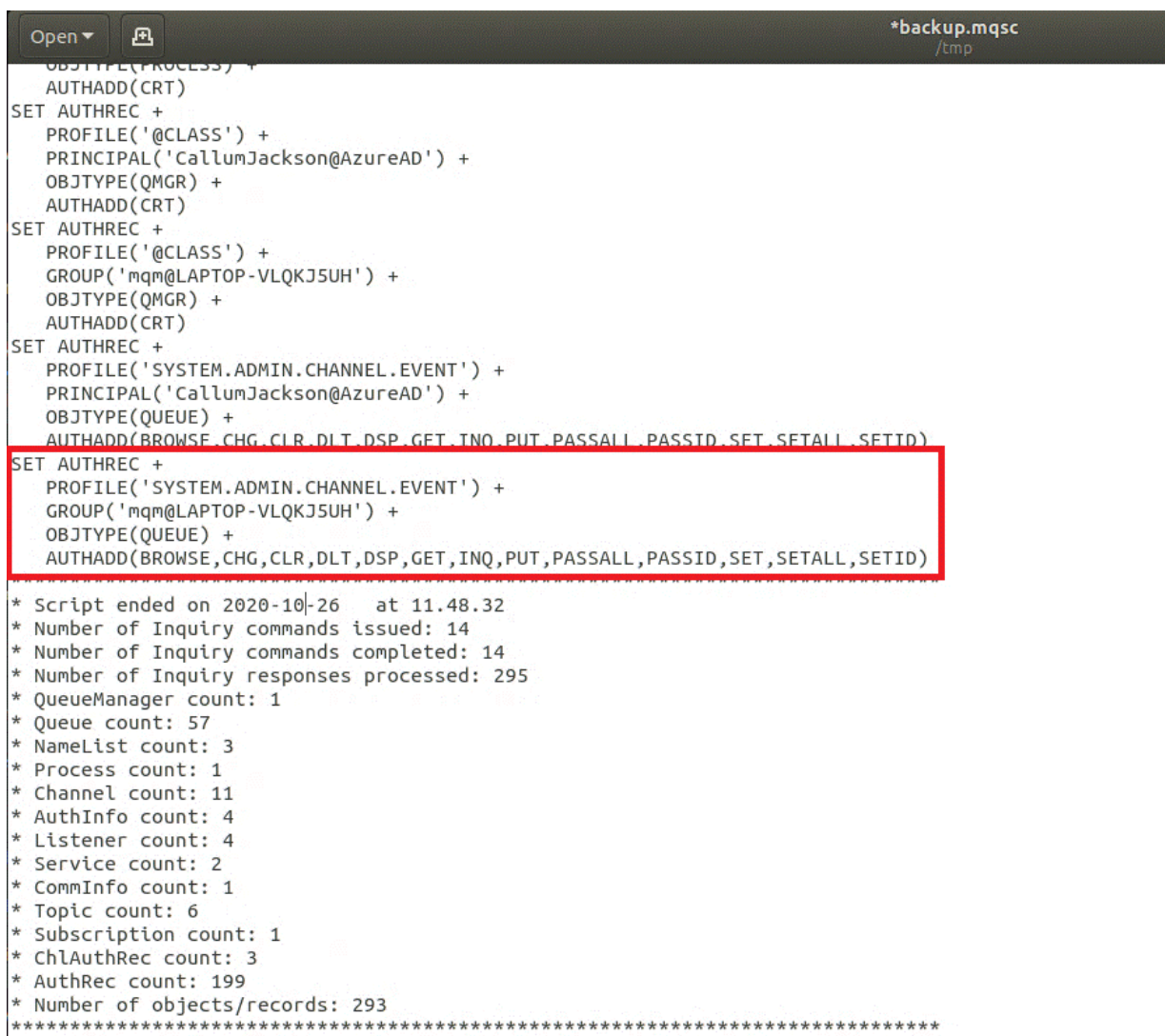
LDAP 許可情報の更新には 2 つの部分があります。

1. 既存のすべての許可をファイルから削除します。
2. LDAP 用の新しい許可情報を定義します。

手順

1. 既存のすべての許可をファイルから削除します。

バックアップ・ファイルのファイルの終わり近くに、SET AUTHREC で始まるいくつかの項目があるはずで



```
Open [icon] *backup.mqsc /tmp
OBJTYPE(PROCESS) +
AUTHADD(CRT)
SET AUTHREC +
PROFILE('@CLASS') +
PRINCIPAL('CallumJackson@AzureAD') +
OBJTYPE(QMGR) +
AUTHADD(CRT)
SET AUTHREC +
PROFILE('@CLASS') +
GROUP('mqm@LAPTOP-VLQKJ5UH') +
OBJTYPE(QMGR) +
AUTHADD(CRT)
SET AUTHREC +
PROFILE('SYSTEM.ADMIN.CHANNEL.EVENT') +
PRINCIPAL('CallumJackson@AzureAD') +
OBJTYPE(Queue) +
AUTHADD(BROWSE,CHG,CLR,DLT,DSP,GET,INQ,PUT,PASSALL,PASSID,SET,SETALL,SETID)
SET AUTHREC +
PROFILE('SYSTEM.ADMIN.CHANNEL.EVENT') +
GROUP('mqm@LAPTOP-VLQKJ5UH') +
OBJTYPE(Queue) +
AUTHADD(BROWSE,CHG,CLR,DLT,DSP,GET,INQ,PUT,PASSALL,PASSID,SET,SETALL,SETID)
* Script ended on 2020-10-26 at 11.48.32
* Number of Inquiry commands issued: 14
* Number of Inquiry commands completed: 14
* Number of Inquiry responses processed: 295
* QueueManager count: 1
* Queue count: 57
* NameList count: 3
* Process count: 1
* Channel count: 11
* AuthInfo count: 4
* Listener count: 4
* Service count: 2
* CommInfo count: 1
* Topic count: 6
* Subscription count: 1
* ChlAuthRec count: 3
* AuthRec count: 199
* Number of objects/records: 293
*****
```

既存の項目を見つけて削除します。一番手っ取り早い方法は、既存の SET AUTHREC ルールをすべて削除してから、LDAP 項目に基づいて新しい項目を作成する方法です。

2. LDAP 用の新しい許可情報を定義します。

キュー・マネージャーの構成、およびリソースとグループの数によっては、この作業に時間がかかる場合もありますし、簡単にできる場合もあります。以下の例では、キュー・マネージャーには Q1 という名前のキューが 1 つしかなく、LDAP グループ apps にアクセス権限を許可しようとしている状況を想定しています。

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

最初の AUTHREC コマンドは、キュー・マネージャーにアクセスするための権限を追加し、2 番目のコマンドはキューへのアクセス権限を設定します。2 番目のキューにアクセスする必要がある場合は、3 番目の AUTHREC コマンドが必要になります。あるいは、ワイルドカードを使用して一般化したアクセス権限を設定することもできます。

ここで別の例を見てみましょう。管理者グループ(名前は admins)がキュー・マネージャーに対する全アクセス権限を必要とする場合は、以下のコマンドを追加します。

```
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)
```

次のタスク

これで、[IBM MQ 構成の IP アドレスとホスト名を変更する準備](#)ができました。

オプション: IBM MQ 構成内の IP アドレスとホスト名の変更

IBM MQ 構成には、IP アドレスとホスト名が指定されている場合があります。状況によっては、これらをそのまま使用できる場合もありますが、更新する必要がある場合もあります。

始める前に

このタスクでは、[LDAP を構成済み](#)であることを前提としています。

このタスクについて

これを行う必要がありますか？

まず、直前のセクションで定義した LDAP 構成とは別に、IP アドレスまたはホスト名を指定したかどうかを判別します。そのためには、以下のコマンドを実行します。

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

出力例は次のとおりです。

```
*****
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(' ') +
```

この例では、検索によって3つの結果が返されています。1つの結果は、前に定義したLDAP構成に対応しています。LDAPサーバーのホスト名は同じままであるため、これは無視できます。他の2つの結果は空の接続項目であるため、これらも無視できます。さらに他の項目がなければ、このトピックの残りの部分をスキップできます。

手順

1. 返された項目の意味を考えます。

IBM MQ では、構成の多くの側面に IP アドレス、ホスト名、およびポートを含めることができます。これらは、次の2つのカテゴリーに分類できます。

- a. **このキュー・マネージャーのロケーション:** このキュー・マネージャーが使用またはパブリッシュするロケーション情報。IBM MQ ネットワーク内の他のキュー・マネージャーやアプリケーションはこの情報を使用して接続することができます。
- b. **キュー・マネージャーの依存関係のロケーション:** このキュー・マネージャーが認識している必要がある他のキュー・マネージャーまたはシステムのロケーション。

このシナリオは、このキュー・マネージャー構成への変更のみに焦点を当てているため、カテゴリー (a) の構成の更新のみについて説明します。ただし、このキュー・マネージャーのロケーションが他のキュー・マネージャーまたはアプリケーションによって参照されている場合は、このキュー・マネージャーの新しいロケーションと一致するように、それらの構成の更新が必要になることがあります。

更新する必要がある情報が含まれている可能性がある主なオブジェクトは、次の2つです。

- リスナー: これは、IBM MQ が listen するネットワーク・アドレスを表します。
 - CLUSTER RECEIVER チャンネル: キュー・マネージャーが IBM MQ クラスターの一部である場合、このオブジェクトが存在します。これは、他のキュー・マネージャーが接続できるネットワーク・アドレスを指定します。
2. `grep 'CONNAME\|LOCLADDR\|IPADDRV' -B 3 backup.mqsc` コマンドからの元の出力で、CLUSTER RECEIVER チャンネルが定義されているかどうかを確認します。定義されている場合は、その IP アドレスを更新します。

CLUSTER RECEIVER チャンネルが定義されているかどうかを確認するには、オリジナルの出力の中に、次のように `CHLTYPE(CLUSRCVR)` を持つ項目を見つけます。

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

項目が存在する場合は、IBM MQ Red Hat OpenShift 経路を使用して `CONNAME` を更新します。この値は、Red Hat OpenShift 環境に基づいており、予測可能な構文を使用します。

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

例えば、キュー・マネージャーのデプロイメントが `cp4i` 名前空間内で `qm1` にという名前で、`openshift_app_route_hostname` が `apps.callumj.icp4i.com` の場合、経路 URL は次のようになります。

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

このルートのポート番号は、通常 443 です。Red Hat OpenShift 管理者が異なる方法で指示しない限り、これは通常、正しい値になります。この情報を使用して、CONNAME フィールドを更新します。以下に例を示します。

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

`grep 'CONNAME\|LOCLADDR\|IPADDRV' -B 3 backup.mqsc` コマンドの元の出力で、LOCLADDR または IPADDRV の項目が存在するかどうかを確認します。それらが存在する場合は、削除します。それらはコンテナ環境では関係ありません。

次のタスク

これで、[コンテナ環境用にキュー・マネージャー構成を更新する準備](#)ができました。

コンテナ環境用のキュー・マネージャー構成の更新

コンテナで実行する場合、構成の特定の側面はコンテナによって定義され、エクスポートされた構成と対立する可能性があります。

始める前に

このタスクでは、[IBM MQ 構成の IP アドレスとホスト名を変更済み](#)であることを前提としています。

このタスクについて

以下の構成の側面は、コンテナによって定義されます。

- リスナー定義 (公開されたポートに対応)。
- TLS ストアの候補となる場所。

そのような理由で、エクスポートした構成を更新する必要があります。

1. [リスナー定義をすべて削除](#)します。
2. [TLS 鍵リポジトリの場所を定義](#)します。

手順

1. リスナー定義をすべて削除します。

バックアップ構成内で `DEFINE LISTENER` を検索します。これは、AUTHINFO 定義と SERVICE 定義の間にあるはずで、強調表示されている領域を削除します。

*backup.mqsc

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

2. TLS 鍵リポジトリの場所を定義します。

キュー・マネージャーのバックアップには、元の環境の TLS 構成が含まれています。これはコンテナ環境とは異なるため、いくつかの更新が必要です。

- **CERTLABL** 項目を default に変更します
- TLS キー・リポジトリの場所 (**SSLKEYR**) を /run/runmqserver/tls/key に変更します

ファイル内の **SSLKEYR** 属性の位置を検索するには、**SSLKEYR** を検索します。通常は 1 つの項目のみが検出されます。複数の項目が検出された場合は、以下の図に示すように、**QMGR** オブジェクトを編集集中であるか確認します。

```

*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 +
* ALTDATA(2020-10-26) +
* ALTTIME(11.43.11) +
  CCSTD(850) +
  CERTLABL('default') +
  CLWLUSEQ(LOCAL) +
* COMMANDQ(SYSTEM.ADMIN.COMMAND.QUEUE) +
  CONNAUTH('USE.LDAP') +
* CRDATE(2020-10-26) +
* CRTIME(11.43.11) +
* QMID(qm1_2020-10-26_11.43.11) +
  SSLCRYP(' ') +
  SSLKEYR('/run/runmqserver/tls/key') +
  SUITEB(NONE) +
* VERSION(09020000) +
  FORCE
REFRESH SECURITY

```

次のタスク

これで、コンテナで実行されている IBM MQ のためのターゲット・アーキテクチャーを選択する準備が整いました。

OpenShift CD CP4I-LTS コンテナで実行されている IBM MQ のためのターゲット HA アーキテクチャーの選択

高可用性の要件を満たすために、単一インスタンス (単一の Kubernetes ポッド) と複数インスタンス (2 つのポッド) のいずれかを選択します。

始める前に

このタスクでは、コンテナ環境用のキュー・マネージャー構成を更新済みであることを前提としています。

このタスクについて

IBM MQ Operator には、以下の 2 つの高可用性オプションが用意されています。

- **単一インスタンス:** 単一のコンテナ (ポッド) が開始され、障害が発生した場合に Red Hat OpenShift が再始動するのはその責任になります。Kubernetes 内のステートフル・セットの特性が原因で、このフェイルオーバーの際に、長い時間がかかったり管理アクションの実行が求められたりする状態がいくつかあります。
- **複数インスタンス:** 2 つのコンテナ (それぞれを別個のポッドに配置) が、1 つはアクティブ・モード、もう 1 つはスタンバイで開始します。このトポロジーでは、フェイルオーバーの大幅な高速化が可能です。これには、IBM MQ の要件を満たす Read Write Many ファイル・システムが必要です。

このタスクでは、ターゲット HA アーキテクチャーのみを選択します。選択したアーキテクチャーを構成する手順については、このシナリオの後続のタスク (97 ページの『Red Hat OpenShift での新しいキュー・マネージャーの作成』) で説明します。

手順

1. 2 つのオプションを検討します。

これらの 2 つのオプションの総合的な説明については、25 ページの『コンテナ内の IBM MQ の高可用性』を参照してください。

2. ターゲット HA アーキテクチャーを選択します。

どちらのオプションを選択すべきか分からない場合、まず単一インスタンスのオプションで開始し、これがご使用の環境の高可用性要件を満たしているかどうかを確認します。

次のタスク

これで、キュー・マネージャー・リソースを作成する準備ができました。

キュー・マネージャー用のリソースの作成

IBM MQ 構成、TLS 証明書、および鍵を Red Hat OpenShift 環境にインポートします。

始める前に

このタスクでは、コンテナで実行される IBM MQ 用のターゲット・アーキテクチャーを選択済みであることを前提としています。

このタスクについて

前の各セクションで、次の 2 つのリソースの抽出、更新、および定義を完了しました。

- IBM MQ 構成
- TLS 証明書および鍵

キュー・マネージャーがデプロイされる前に、これらのリソースを Red Hat OpenShift 環境にインポートする必要があります。

手順

1. IBM MQ 構成を Red Hat OpenShift にインポートします。

以下の説明では、現行ディレクトリーの backup.mqsc というファイルに IBM MQ 構成があることを前提としています。そうでない場合は、ご使用の環境に基づいてファイル名をカスタマイズする必要があります。

a) `oc login` を使用してクラスターにログインします。

b) IBM MQ 構成を `configmap` にロードします。

以下のコマンドを実行します。

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

c) ファイルが正常にロードされたことを確認します。

以下のコマンドを実行します。

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

2. IBM MQ TLS リソースをインポートします。

82 ページの『[オプション: キュー・マネージャーの鍵と証明書の抽出および取得](#)』で説明しているように、キュー・マネージャーのデプロイメントに TLS が必要な場合があります。その場合は、`.crt` および `.key` で終わるファイルの数が既に存在するはずですが、キュー・マネージャーがこれらをデプロイメント時に参照するために、これらのファイルを Kubernetes シークレットに追加する必要があります。

キュー・マネージャー用の鍵と証明書が存在する場合、それらは一例として次のような名前になります。

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

これらのファイルをインポートするには、以下のコマンドを実行します。

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

Kubernetes では、一致する公開鍵と秘密鍵のインポート時に、この役立つユーティリティーが提供されます。例えば、キュー・マネージャーのトラストストアに追加する証明書がさらにある場合、次のコマンドを実行します。

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

例えば、インポートするファイルが `trust1.crt`、`trust2.crt` と `trust3.crt` の場合、コマンドは次のようになります：

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

次のタスク

これで、[Red Hat OpenShift で新しいキュー・マネージャーを作成する準備ができました](#)。

Red Hat OpenShift での新しいキュー・マネージャーの作成

Red Hat OpenShift に単一インスタンス・キュー・マネージャーまたは複数インスタンス・キュー・マネージャーのいずれかをデプロイします。

始める前に

このタスクは、[キュー・マネージャー・リソースの作成および IBM MQ Operator を Red Hat OpenShift にインストールする](#)があることを前提としています。

このタスクについて

95 ページの『[コンテナで実行されている IBM MQ のためのターゲット HA アーキテクチャーの選択](#)』で概略を説明しているように、候補となるデプロイメント・トポロジは 2 つあります。このため、このトピックでは、次の 2 つの異なるテンプレートが用意されています。

- [単一インスタンス・キュー・マネージャーをデプロイする](#)。
- [複数インスタンス・キュー・マネージャーをデプロイする](#)。

重要: ご使用のトポロジに基づいて、2 つのテンプレートのうちの 1 つのみを実行してください。

手順

- [単一インスタンス・キュー・マネージャーをデプロイする](#)。

マイグレーションされたキュー・マネージャーは、YAML ファイルを使用して Red Hat OpenShift にデプロイされます。前の各トピックで使用された名前に基づいたサンプルを以下に示します。

```
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
```

実行した手順によっては、直前の YAML をカスタマイズする必要があります。これを行うための手引きとして、この YAML の説明を次に示します。

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

これは、Kubernetes のオブジェクト、タイプ、および名前を定義します。カスタマイズが必要となるフィールドは name フィールドのみです。

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

これは、デプロイメントのバージョン情報とライセンス情報に対応しています。これをカスタマイズする必要がある場合は、183 ページの『mq.ibm.com/v1beta1 のライセンスのリファレンス』に記載されている情報を使用してください。

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

TLS を使用するように構成されているキュー・マネージャーの場合は、関連する証明書と鍵を参照する必要があります。secretName フィールドは、Kubernetes シークレット ([IBM MQ TLS リソースのインポート](#) のセクション内で作成されたもの) を参照します。項目 (tls.key および tls.crt) のリストは、Kubernetes が oc create secret tls 構文の使用時に割り当てる標準名です。トラストストアに追加する証明書がさらにある場合は、それらは同様の方法で追加できますが、各項目はインポート時に使用された対応する各ファイル名です。例えば、トラストストア証明書を作成するために次のコードを使用できます。

```
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: TLS が必要ない場合は、YAML の TLS セクションを削除します。

```
web:
  enabled: true
```

これは、デプロイメントの Web コンソールを有効にします。

```
queueManager:
  name: QM1
```

これは、キュー・マネージャーの名前を QM1 と定義します。キュー・マネージャーは、お客様の要件 (例えば、元のキュー・マネージャー名) に基づいてカスタマイズします。

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

直前のコードは、[IBM MQ 構成をインポートする](#) セクションでインポートされたキュー・マネージャー構成をプルします。別の名前を使用した場合は、my-mqsc-migrated と backup.mqsc を変更する必要があります。

サンプルの YAML では、Red Hat OpenShift 環境のデフォルトのストレージ・クラスが RWX または RWO ストレージ・クラスのいずれかとして定義されていることが前提となっていることに注意してください。ご使用の環境内でデフォルトが定義されていない場合は、使用するストレージ・クラスを指定する必要があります。これは、YAML を次のように拡張することで行えます。

```
queueManager:
  name: QM1
  storage:
    defaultClass: my_storage_class
```

```
queueManager:
  type: persistent-claim
```

強調表示されたテキストを、ご使用の環境に合わせてカスタマイズしたクラス属性を設定して追加します。ご使用の環境内のストレージ・クラス名を検出するには、次のコマンドを実行します。

```
oc get storageclass
```

このコマンドによって返される出力例を次に示します。

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

以下のコードは、IBM MQ 構成 ([IBM MQ 構成のインポートのセクション](#)でインポートされたもの) を参照する方法を示しています。別の名前を使用した場合は、`my-mqsc-migrated` と `backup.mqsc` を変更する必要があります。

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

単一インスタンス・キュー・マネージャーのデプロイが完了しました。これにより、テンプレートが完了します。これで、[新しいコンテナ・デプロイメントを検証する準備](#)ができました。

- 複数インスタンス・キュー・マネージャーをデプロイする。

マイグレーションされたキュー・マネージャーは、YAML ファイルを使用して Red Hat OpenShift にデプロイされます。前の各セクションで使用された名前に基づいたサンプルを以下に示します。

```
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
```

ここでは、この YAML について説明します。構成の大部分は、[単一インスタンス・キュー・マネージャーをデプロイする方法](#)と同じであるため、ここでは、キュー・マネージャーの可用性とストレージの側面についてのみ説明します。

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

これは、キュー・マネージャー名を `QM1` として指定し、デプロイメントがデフォルトの単一インスタンスではなく `MultiInstance` になるように設定します。

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

IBM MQ 複数インスタンス・キュー・マネージャーは、RWX ストレージに依存します。デフォルトでは、キュー・マネージャーは単一インスタンス・モードでデプロイされるため、複数インスタンス・モードに変更する場合は追加のストレージ・オプションが必要になります。直前の YAML サンプルでは、3つのストレージ永続ボリュームと1つの永続ボリューム・クラスが定義されています。この永続ボリューム・クラスは、RWX ストレージ・クラスでなければなりません。ご使用の環境内の各ストレージ・クラス名が分からない場合は、次のコマンドを実行してそれらを検出することができます。

```
oc get storageclass
```

このコマンドによって返される出力例を次に示します。

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

以下のコードは、IBM MQ 構成 ([IBM MQ 構成のインポートのセクション](#)でインポートされたもの) を参照する方法を示しています。別の名前を使用した場合は、`my-mqsc-migrated` と `backup.mqsc` を変更する必要があります。

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

複数インスタンス・キュー・マネージャーのデプロイが完了しました。これにより、テンプレートが完了します。これで、[新しいコンテナ・デプロイメントを検証する準備](#)ができました。

OpenShift CD CP4I-LTS 新規コンテナ・デプロイメントの検証

IBM MQ が Red Hat OpenShift にデプロイされたので、IBM MQ サンプルを使用して環境を検査できます。

始める前に

このタスクでは、[Red Hat OpenShift に新しいキュー・マネージャーを作成済み](#)であることを前提としています。

重要: このタスクでは、キュー・マネージャーで TLS が有効になっていないことを前提としています。

このタスクについて

このタスクでは、マイグレーション済みキュー・マネージャーのコンテナの内部から IBM MQ サンプルを実行します。ただし、別の環境から実行される独自のアプリケーションを使用することもできます。

以下の情報が必要になります。

- LDAP ユーザー名
- LDAP パスワード
- IBM MQ チャンネル名
- キュー名

このサンプル・コードは、以下の設定を使用します。この設定はご使用の環境によって異なることに注意してください。

- LDAP ユーザー名: mqapp
- LDAP パスワード: mqapp
- IBM MQ チャンネル名: DEV.APP.SVRCONN
- キュー名: Q1

手順

1. 実行中の IBM MQ コンテナに `exec` を実行します。

以下のコマンドを使用します。

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

ここで `qm1-ibm-mq-0` は、[97 ページの『Red Hat OpenShift での新しいキュー・マネージャーの作成』](#)でデプロイしたポッドです。別のデプロイメントを呼び出した場合は、この値をカスタマイズします。

2. メッセージを送信します。

以下のコマンドを実行します。

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

パスワードの入力を求めるプロンプトが表示されたら、メッセージを送信できます。

3. メッセージが正常に受信されたことを確認します。

GET サンプルを実行します。

```
./amqsgetc Q1 QM1
```

タスクの結果

[79 ページの『IBM Cloud Pak for Integration への IBM MQ のマイグレーション』](#)が完了しました。

次のタスク

より複雑なマイグレーション・シナリオについては、以下の情報を参考にしてください。

キューに入ったメッセージのマイグレーション

キューに入っている既存のメッセージをマイグレーションするには、新しいキュー・マネージャーの配置後にメッセージをエクスポートおよびインポートするために、[2つのシステム間での dmpmqmsg ユーティリティの使用のトピックのガイド](#)に従ってください。

Red Hat OpenShift 環境の外部からの IBM MQ への接続

デプロイされたキュー・マネージャーは、Red Hat OpenShift 環境外の IBM MQ クライアントおよびキュー・マネージャーに公開することができます。このプロセスは、Red Hat OpenShift 環境に接続する IBM MQ のバージョンによって異なります。[155 ページの『Red Hat OpenShift クラスターの外部からキュー・マネージャーに接続するためのルートの構成』](#)を参照してください。

IBM MQ Operator は、OpenShift コンソールまたはコマンド・ライン・インターフェース (CLI) を使用して Red Hat OpenShift にインストールできます。

始める前に

インストールができるだけスムーズに行われるようにするには、インストールを開始する前に、すべての前提条件と要件を理解しておく必要があります。5 ページの『[コンテナ内の IBM MQ の計画](#)』を参照してください。

重要: [V9.3.4](#) IBM MQ Operator をインストールする前に、[デプロイメントの構造化](#)に関するガイドンスを確認してください。

このタスクについて

以下のステップは、IBM MQ Operator をインストールするための標準的な作業フローを示しています。

1. [Red Hat OpenShift Container Platform](#) をインストールします。
2. [ストレージを構成する](#)。
3. [ミラー・イメージ \(エア・ギャップのみ\)](#)。
4. [IBM オペレーター・カタログ](#)を追加し、[クラスターを準備](#)します。
5. [IBM MQ Operator](#) をインストールします。
6. [ライセンス・キー・シークレット](#)を作成します (オンライン・インストールのみ)。
7. [オプション: IBM Cloud Pak for Integration \(CP4I\) とその依存関係をインストール](#)します。
8. [License Service](#) をデプロイします。
9. [キュー・マネージャー](#)をデプロイします。

手順

1. Red Hat OpenShift Container Platform をインストールします。

OpenShift をインストールする詳細なステップについては、[Red Hat ソフトウェア 4.6 以降のインストール](#)を参照してください。

重要: サポートされているバージョンの OpenShift Container Platform がインストールされていることを確認してください。例えば、IBM MQ Operator 2.0 以降を使用するには、OpenShift Container Platform 4.12 以降をインストールする必要があります。また、OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされることにも注意してください。これらのリリースは、4.14 や 4.16 などの偶数番号のマイナー・リリースです。詳しくは、[IBM Cloud Pak および Red Hat OpenShift Container Platform の互換性](#)を参照してください。

Red Hat OpenShift Container Platform CLI を使用するすべてのステップで、oc login を使用して OpenShift クラスターにログインする必要があります。CLI をインストールするには、[OpenShift CLI の概要](#)を参照してください。

OpenShift をインストールした後、[ライセンス・キー・シークレットの作成](#)で作成した IBM ライセンス・キーを使用して、コンテナ・ソフトウェアを検証し、アクセス権限を取得できます。

2. ストレージを構成します。

Red Hat OpenShift Container Platform でストレージ・クラスを定義し、サイジング要件を満たすようにストレージ構成を設定する必要があります。

重要: IBM MQ 単一インスタンスおよびネイティブ HA キュー・マネージャーは RWO アクセス・モードを使用できますが、複数インスタンス・キュー・マネージャーは、21 ページの『[IBM MQ Operator のストレージに関する考慮事項](#)』で説明されているように RWX を必要とします。IBM MQ 複数インスタンス・キュー・マネージャーには、特定のファイル・システム特性が必要です。これは、[IBM MQ の共有ファイル・システムのテストの手順](#)を使用して検証できます。

既知の準拠ファイル・システムと非準拠ファイル・システムのリスト、およびその他の制限や制約事項については、[IBM MQ ファイル・システムのテスト・ステートメント](#)を参照してください。

推奨されるストレージ・プロバイダーについては、CP4I [「ストレージに関する考慮事項」](#) ページを参照してください。

3. **V9.3.4**

イメージをミラーリングします(エア・ギャップのみ)。

クラスターが制限付き(エアギャップ)ネットワーク環境にある場合は、IBM MQ イメージをミラーリングする必要があります。構成によっては、いくつかの追加コンポーネントのミラーリングが必要になる場合もあります。以下の情報を読み、必要に応じてイメージをミラーリングします。

- IBM MQ イメージをミラーリングする必要があります。以下の値を使用します。

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

- また、以下のすべての記述が当てはまるキュー・マネージャーを少なくとも1つデプロイする予定の場合は、いくつかの追加の必須コンポーネントもミラーリングする必要があります。

- IBM MQ 9.3.4 以降を使用している。
- CP4I ライセンスを使用している。
- IBM MQ Console が使用可能になります。
- IBM MQ Console シングル・サインオン (SSO) 認証および許可に IBM Cloud Pak for Integration Keycloak サービスを使用している (デフォルト)。

上記の記述が当てはまる場合、SSO は Keycloak によって提供され、以下の各コンポーネントをミラーリングする必要があります。

- IBM Cloud Pak foundational services
- Certificate Manager. バージョン 4.4 より前のバージョンの IBM Cloud Pak foundational services オペレーターをインストールした場合は、Certificate Manager をミラーリングする必要があります。⁶
- IBM Cloud Pak for Integration
- Keycloak (Red Hat OpenShift 演算子)

ミラー・イメージを作成するには、[エアギャップ・クラスターのミラーリング・イメージ](#)を参照してください。

4. IBM MQ Operator カタログ・ソースを追加します。

オペレーターをクラスターで使用できるようにするカタログ・ソースを追加します。[106 ページの『IBM MQ Operator カタログ・ソースの追加』](#)を参照してください。

5. IBM MQ Operator のインストール

以下の2つのオプションのいずれかを選択します(コンソールを使用するか、CLIを使用します)。

- オプション 1: [OpenShift コンソールを使用して IBM MQ Operator をインストールします。](#)
- オプション 2: [OpenShift CLI を使用して IBM MQ Operator をインストールします。](#)

6. 使用権キー・シークレットを作成します(オンライン・インストールのみ)。

IBM MQ Operator は、ライセンス資格検査を実行するコンテナ・レジストリーからプルされたキュー・マネージャー・イメージをデプロイします。この検査には、docker-registry プル・シークレットに保管されているライセンス・キーが必要です。キュー・マネージャーをインストールする名前空間にまだライセンス・キーがない場合は、以下の手順に従ってライセンス・キーを取得し、プル・シークレットを作成します。

注: IBM MQ Advanced for Developers (非警告) キュー・マネージャーのみをデプロイする場合は、ライセンス・キーは必要ありません。

⁶ IBM Cloud Pak foundational services のバージョン 4.4 以降、このミラーリングは不要になりました。

OpenShift コンソールまたは CLI を使用して、使用権キー・シークレットを作成できます。以下の例では、CLI を使用します。

- a. IBM ID に割り当てられているライセンス・キーを取得します。ライセンスが付与されているソフトウェアに関連付けられた IBM ID とパスワードを使用して、[MyIBM Container Software Library](#) にログインします。
- b. 「**使用権キー (Entitlement keys)**」セクションで「**キーのコピー (Copy key)**」を選択して、使用権キーをクリップボードにコピーします。
- c. OpenShift CLI から以下のコマンドを実行して、`ibm-entitlement-key` という名前のイメージ・プル・シークレットを作成します。

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

ここで、`<entitlement-key>` はステップ b でコピーした使用権キー、`<user-email>` は使用権付きソフトウェアに関連付けられた IBM ID、`<namespace>` は IBM MQ Operator をインストールした名前空間です。

7. オプション: CP4I とその依存関係をインストールします。

少なくとも 1 つのキュー・マネージャーをデプロイするときに、以下のすべての記述が当てはまる場合は、いくつかの追加の必須コンポーネントがあります。

- IBM MQ 9.3.4 以降を使用している。
- CP4I ライセンスを使用している。
- IBM MQ Console が使用可能になります。
- IBM MQ Console シングル・サインオン (SSO) 認証および許可に CP4I Keycloak サービスを使用している (デフォルト)。

上記の記述がすべて当てはまる場合、SSO は Keycloak によって提供されるため、以下の追加ステップを実行する必要があります。

- CP4I Operator と同じインストール・モードで IBM Cloud Pak foundational services Operator をインストールします。サポートされるバージョンについては、[このリリースのオペレーター・チャンネルのバージョン](#)を参照してください。
- バージョン 4.4 より前のバージョンの IBM Cloud Pak foundational services オペレーターをインストールした場合は、[Red Hat OpenShift Container Platform の cert-manager Operator](#) をインストールします。⁷
- [CP4I Operator](#) をインストールします。
- オプション: プラットフォーム UI をデプロイします。
 - a. `ibm-common-services` 名前空間を作成します。CLI を使用して OpenShift クラスターにログインしたら、以下のコマンドを実行します。

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

- b. [プラットフォーム UI](#) をデプロイします。

8. License Service をデプロイします。

これは、キュー・マネージャーのライセンス使用状況をモニターするために必要です。[License Service](#) の手順に従います。

9. キュー・マネージャーをデプロイします。

⁷ IBM Cloud Pak foundational services のバージョン 4.4 以降では、これは不要になりました。

サンプルの "クイック・スタート" ・キュー・マネージャーのデプロイ手順については、[116 ページの『Red Hat OpenShift Container Platform クラスターにキュー・マネージャーをデプロイする方法』](#)を参照してください。

関連タスク

[119 ページの『IBM MQ Operator のアンインストール』](#)

Red Hat OpenShift コンソールまたは CLI を使用して、Red Hat OpenShift から IBM MQ Operator をアンインストールできます。

関連資料

[110 ページの『エア・ギャップ環境での IBM MQ Operator 2.x のインストール』](#)

このチュートリアルでは、インターネットに接続されていない Red Hat OpenShift クラスターに IBM MQ Operator 2.x をインストールする方法について説明します。ポータブル・ストレージ・デバイスまたは要塞マシンを使用して、エア・ギャップ環境に IBM MQ Operator をインストールできます。

OpenShift CP4I IBM MQ Operator カタログ・ソースの追加

OpenShift クラスターにカタログ・ソースを追加すると、インストール可能なオペレーターのリストに IBM オペレーターが追加されます。

始める前に

このタスクでは、[103 ページの『IBM MQ Operator のインストール』](#)の最初の 3 つのステップが完了していることを前提としています。

このタスクは、クラスター管理者が実行する必要があります。

このタスクについて

IBM MQ Operator カタログは、Red Hat OpenShift Container Platform クラスターの API を拡張して IBM ソフトウェア製品を使用可能にするために使用できるオペレーターの索引です。

クラスターが制限付き (エアギャップ) ネットワーク環境にあるか、クラスターがインターネットにアクセスできるかに応じて、**オプション A: エア・ギャップ** または **オプション B: インターネット** のいずれかを実行します。

手順

9.3.4

オプション A: エア・ギャップ エアギャップ・ネットワーク環境でカタログ・ソースを追加します。

a) IBM MQ Operator カタログ・ソースを追加します。

[クラスターへのカタログ・ソースの追加の手順に従います。](#)

注: オペレーター・インストール・ステップ「[イメージのミラーリング \(エア・ギャップのみ\) \(Mirror images \(air-gap only\)\)](#)」を既に完了しているため、カタログ・ソースを適用するステップのみを実行する必要があります。以下に例を示します。

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

b) 追加の必須コンポーネントのカタログ・ソースを追加します。

少なくとも 1 つのキュー・マネージャーをデプロイするときに、以下の **すべて** の記述が当てはまる場合は、いくつかの追加の必須コンポーネントがあります。

- IBM MQ 9.3.4 以降を使用している。
- IBM Cloud Pak for Integration ライセンスを使用している。
- IBM MQ Console が使用可能になります。
- IBM MQ Console シングル・サインオン (SSO) 認証および許可に IBM Cloud Pak for Integration Keycloak サービスを使用している (デフォルト)。

上記の記述がすべて当てはまる場合、SSO は Keycloak によって提供されます。したがって、IBM MQ Operator カタログ・ソースの場合と同様に、以下の追加の必須コンポーネントのそれぞれについて、[クラスターへのカタログ・ソースの追加](#)のステップも実行する必要があります。

- IBM Cloud Pak foundational services
- Certificate Manager. バージョン 4.4 より前のバージョンの IBM Cloud Pak foundational services オペレーターをインストールした場合は、Certificate Manager をミラーリングする必要があります。⁸
- IBM Cloud Pak for Integration
- **オプション B: インターネット** インターネットにアクセスできる環境でカタログ・ソースを追加します。OpenShift CLI を使用して CatalogSource を作成します。

以下の YAML ファイルを Red Hat OpenShift Container Platform クラスターに適用して、カタログを追加します。

a) CatalogSource YAML を作成します。

以下のリソース定義を catalog_source.yaml という名前のファイルとして保存します。

```
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) CatalogSource YAML を適用します。

これを行うには、Red Hat OpenShift Container Platform Web コンソールで「+」ボタンをクリックするか、コマンド行を使用します。

例えば、以下のコマンドを実行してファイルを適用します。

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

c) CatalogSource が正常に作成されたことの確認

以下のコマンドを実行します。

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

成功した場合、以下の出力が表示されます。

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

タスクの結果

これで、IBM MQ Operator のインストールのステップ 5 を実行する準備ができました。

OpenShift コンソールを使用した IBM MQ Operator のインストール

IBM MQ Operator は、OperatorHub を使用して Red Hat OpenShift にインストールできます。

始める前に

このタスクは、103 ページの『IBM MQ Operator のインストール』のステップ 1 から 4 を完了していることを前提としています。

⁸ IBM Cloud Pak foundational services のバージョン 4.4 以降、このミラーリングは不要になりました。

手順

1. Red Hat OpenShift クラスター・コンソールにログインします。
2. ナビゲーション・ペインで、「オペレーター (Operators)」 > 「OperatorHub」をクリックします。
「OperatorHub」ページが表示されます。
3. 「All Items」フィールドで、「IBM MQ」と入力します。
IBM MQ カタログ・エントリーが表示されます。
4. 「IBM MQ」を選択します。
「IBM MQ」ウィンドウが表示されます。
5. 「インストール」をクリックします。
「Install Operator」ページが表示されます。
6. 以下の値を入力します。
 - a) 「チャンネル」を、選択したバージョンに設定します。
11 ページの『IBM MQ Operator のバージョン・サポート』を確認して、選択するオペレーター・チャンネルを判別します。
 - b) 「インストール・モード」を「クラスター上の特定の名前空間」(次のステップで作成できます)またはクラスター全体の有効範囲のいずれかに設定します。

異なるバージョンのオペレーターを異なる名前空間にインストールすると問題が発生する可能性があるため、クラスター全体の有効範囲を選択することをお勧めします。オペレーターは、コントロール・プレーンの拡張機能となるように設計されています。
 - c) オプション: 「クラスター上の特定の名前空間」を選択した場合は、「名前空間」を、オペレーターのインストール先のプロジェクト(名前空間)の値に設定します。

注: コンソールを使用してオペレーターをインストールする場合は、既存の名前空間を使用するか、オペレーターによって提供されるデフォルトの名前空間を使用するか、新しい名前空間を作成することができます。このフォームから新しい名前空間を作成する場合は、以下のようになります。ナビゲーション・ペインで **ホーム** > 「プロジェクト」をクリックし、「プロジェクトの作成」を選択し、作成するプロジェクトの **名前**(名前空間)を指定して、「作成」をクリックします。
 - d) 「承認戦略」を「自動」に設定します。
7. 「インストール」をクリックし、オペレーターがインストールするまで待ちます。

インストールが完了すると、確認が表示されます。

インストールを検証するには、**オペレーター** > 「インストール済みオペレーター (Installed Operators)」にナビゲートし、「プロジェクト (Projects)」ドロップダウン・リストからプロジェクトを選択します。インストールが完了すると、オペレーターの状況が「成功」に変わります。

次のタスク

これで、ライセンス・キー・シークレットを作成する準備ができました (103 ページの『IBM MQ Operator のインストール』のステップ 6)。

Red Hat OpenShift CLI を使用した IBM MQ Operator のインストール

IBM MQ Operator は、コマンド・ライン・インターフェース (CLI) を使用して Red Hat OpenShift にインストールできます。

始める前に

このタスクは、103 ページの『IBM MQ Operator のインストール』のステップ 1 から 4 を完了していることを前提としています。

手順

1. **oc login** を使用して、Red Hat OpenShift コマンド・ライン・インターフェース (CLI) にログインします。
2. オプション: IBM MQ Operator に使用する名前空間を作成します。

IBM MQ Operator は、単一の名前空間またはすべての名前空間を範囲としてインストールできます。このステップは、まだ存在しない特定の名前空間にインストールする場合にのみ必要です。

CLI で新しい名前空間を作成するには、以下のコマンドを実行します。

```
oc create namespace <namespace_name>
```

ここで、<namespace_name> は、作成する名前空間の名前です。

3. OperatorHub から、クラスターで使用可能なオペレーターのリストを表示します。

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

4. IBM MQ Operator を調べて、サポートされている **InstallModes** かつ使用可能な **Channels** ことを確認します。

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

5. オプション: **OperatorGroup** を作成します。

OperatorGroup は、**OperatorGroup** と同じ名前空間におけるすべてのオペレーターに必要となる RBAC アクセス権限を生成するターゲット名前空間を選択する OLM リソースです。

オペレーターのサブスクリプション先となる名前空間には **OperatorGroup** が必要です。これは、オペレーターの **InstallMode** (AllNamespaces モードまたは SingleNamespace モードのいずれか) に適合するものです。

インストールするオペレーターが AllNamespaces モードを使用する場合は、`openshift-operators` 名前空間に適切な **OperatorGroup** が既に設定されているため、このステップをスキップできます。

オペレーターが SingleNamespace モードを使用し、適切な **OperatorGroup** がまだ設定されていない場合は、以下のコマンドを実行して作成します。

```
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. [11 ページの『IBM MQ Operator のバージョン・サポート』](#)を確認して、選択するオペレーター・チャンネルを判別します。
7. オペレーターをインストールします。

以下のコマンドを使用します。<ibm-mq-演算子-channel> を、インストールする IBM MQ Operator のバージョンのチャンネルと一致するように変更し、<ネーム・スペース名> を **openshift-operators** (「AllNamespaces」モードを使用している場合) または IBM MQ Operator のデプロイ先の名前空間 (「SingleNamespace」モードを使用している場合) に変更します。

```
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
```

- 数分後に、オペレーターがインストールされます。以下のコマンドを実行して、すべてのコンポーネントが「成功」状態であることを確認します。

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

ここで、<namespace_name> は、「AllNamespaces」モードを使用している場合は **openshift-operators**、「SingleNamespace」モードを使用している場合はプロジェクト (名前空間) 名です。

次のタスク

これで、[ライセンス・キー・シークレット](#)を作成する準備ができました ([103 ページの『IBM MQ Operator のインストール』](#)のステップ 6)。

OpenShift CP4I Linux エア・ギャップ環境での IBM MQ Operator 2.x のインストール

このチュートリアルでは、インターネットに接続されていない Red Hat OpenShift クラスターに IBM MQ Operator 2.x をインストールする方法について説明します。ポータブル・ストレージ・デバイスまたは要塞マシンを使用して、エア・ギャップ環境に IBM MQ Operator をインストールできます。

始める前に

以下の手順は、エア・ギャップ環境に 2.x バージョンの IBM MQ Operator をインストールするためのものです。IBM MQ Operator 3.0.0 以降をインストールするには、エア・ギャップ固有のステップに特に注意して、[103 ページの『IBM MQ Operator のインストール』](#)を参照してください。

ポータブル・ストレージ・デバイスを使用したエア・ギャップ環境での IBM MQ Operator のインストール

インストールを完了する手順については、IBM Cloud Pak for Integration 資料の [ポータブル・ストレージ・デバイスを使用したミラーリング・イメージ](#)を参照してください。IBM MQ のみをインストールする場合は、以下の環境変数のすべてのオカレンスを、以下に示す値に置き換えます。

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

ここで、*version_number* は、エア・ギャップ・インストールを実行するために使用するケースのバージョンです。使用可能なケースのバージョンのリストについては、<https://github.com/IBM/cloud-pak/tree/master/repo/case/ibm-mq>を参照してください。[11 ページの『IBM MQ Operator のバージョン・サポート』](#)を確認して、選択するオペレーター・チャンネルを判別します。

要塞マシンを使用したエア・ギャップ環境での IBM MQ Operator のインストール

- [111 ページの『前提条件』](#)
- [111 ページの『Docker レジストリーの準備』](#)
- [111 ページの『踏み台ホストの準備』](#)
- [112 ページの『インストーラーおよびイメージ・インベントリー用の環境変数の作成』](#)
- [112 ページの『IBM MQ インストーラーおよびイメージ・インベントリーのダウンロード』](#)
- [113 ページの『クラスター管理者として OpenShift Container Platform クラスターにログインします。』](#)
- [113 ページの『IBM MQ Operator 用の Kubernetes 名前空間の作成』](#)
- [113 ページの『イメージのミラーリングおよびクラスターの構成』](#)

9. [115 ページの『IBM MQ Operator のインストール』](#)

10. [115 ページの『IBM MQ キュー・マネージャーのデプロイ』](#)

前提条件

1. OpenShift Container Platform クラスターをインストールする必要があります。サポートされている OpenShift Container Platform のバージョンについては、[11 ページの『IBM MQ Operator のバージョン・サポート』](#)を参照してください。
2. Docker レジストリーが使用可能である必要があります。詳しくは、[111 ページの『Docker レジストリーの準備』](#)を参照してください。
3. 踏み台サーバーを構成する必要があります。詳細については、[111 ページの『踏み台ホストの準備』](#)を参照してください。

Docker レジストリーの準備

ローカルの Docker レジストリーを使用して、ローカル環境内のすべてのイメージを保管します。そのようなレジストリーを作成し、それが以下の要件を満たすようにする必要があります。

- [Docker Manifest V2, Schema 2](#) をサポートしている。
- マルチ・アーキテクチャー・イメージをサポートしている。
- 踏み台サーバーと OpenShift Container Platform クラスター・ノードの両方からアクセス可能である。
- 踏み台ホストからターゲット・レジストリーに書き込むことができるユーザーのユーザー名とパスワードがある。
- Red Hat OpenShift クラスター・ノード上にあるターゲット・レジストリーから読み取ることができるユーザーのユーザー名とパスワードを持っています。
- イメージ名でパス分離文字が許可される。

Docker レジストリーを作成した後に、以下のようにレジストリーを構成する必要があります。

- 単純なレジストリーの例は、Red Hat OpenShift 資料の「[Creating a mirror registry for installation in a restricted network](#)」に記載されています。
- 各名前空間が以下の要件を満たしていることを確認します。
 - 自動リポジトリ作成をサポートしている。
 - リポジトリの書き込みと作成ができるユーザーの資格情報がある。踏み台ホストが該当する資格情報を使用します。
 - すべてのリポジトリを読み取ることができるユーザーの資格情報がある。OpenShift Container Platform クラスターがこれらの資格情報を使用している。

踏み台ホストの準備

OpenShift Container Platform クラスター、ローカル Docker レジストリー、およびインターネットにアクセスできる踏み台ホストを準備します。要塞ホストは、IBM Cloud Pak CLI および OpenShift Container Platform CLI がサポートするオペレーティング・システムを備えた Linux for x86-64 プラットフォーム上になければなりません。

踏み台ノードで以下のステップを実行します。

1. OpenSSL バージョン 1.11.1 以上をインストールします。
2. 踏み台ノードに Docker または Podman をインストールします。
 - Docker をインストールするには、以下のコマンドを実行します。

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

- Podman をインストールするには、[Podman Installation Instructions](#) を参照してください。

3. 踏み台ノードに `skopeo` バージョン `1.x.x` をインストールします。 `skopeo` をインストールするには、以下のコマンドを実行します。

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

4. IBM Cloud PakCLI をインストールします。 ご使用のプラットフォーム用のバイナリー・ファイルの最新バージョンをインストールします。 詳しくは、[cloud-pak-cli](#) を参照してください。

- a. バイナリー・ファイルをダウンロードします。

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

以下に例を示します。

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

- b. バイナリー・ファイルを解凍します。

```
tar -xf binary-file-name
```

- c. 以下のコマンドを実行して、ファイルを変更および移動します。

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

- d. `cloudctl` がインストールされていることを確認します。

```
cloudctl --help
```

5. oc OpenShift Container Platform CLI ツールをインストールします。

詳しくは、[OpenShift Container Platform CLI ツール](#) を参照してください。

6. オフライン・ストアとして機能するディレクトリーを作成します。

以下に、ディレクトリーの例を示します。 後続のステップでこの例を使用します。

```
mkdir $HOME/offline
```

注: データを複数回転送しなくても済むように、このオフライン・ストアはパーシスタントでなければなりません。 また、パーシスタンスは、複数回、またはスケジュールに従って、ミラーリング・プロセスを実行するのにも役立ちます。

インストーラーおよびイメージ・インベントリー用の環境変数の作成

インストーラー・イメージ名とイメージ・インベントリーを使用して、以下の環境変数を作成します。

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

ここで、`version_number` は、エアギャップ・インストールのために使用するケースのバージョンです。 使用可能なケースのバージョンのリストについては、<https://github.com/IBM/cloud-pak/tree/master/repo/case/ibm-mq> を参照してください。 選択するオペレーター・チャンネルを決定するには、[IBM MQ Operator のバージョン・サポート](#) を参照してください。

IBM MQ インストーラーおよびイメージ・インベントリーのダウンロード

`ibm-mq` インストーラーおよびイメージ・インベントリーを要塞ホストにダウンロードします。

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



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

クラスター管理者として OpenShift Container Platform クラスターにログインします。

以下に、OpenShift Container Platform クラスターにログインするためのコマンド例を示します。

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

IBM MQ Operator 用の Kubernetes 名前空間の作成

IBM MQ Operator をインストールするための名前空間を持つ環境変数を作成してから、名前空間を作成します。

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

イメージのミラーリングおよびクラスターの構成

以下のステップを実行して、イメージをミラーリングし、クラスターを構成します。

注: いずれのコマンドでも二重引用符の間にティルドは使用しないでください。例えば、args "--registry registry --user registry_userid --pass registry_password --inputDir ~/offline" は使用しないでください。波形記号が展開されず、コマンドが失敗する可能性があります。

1. すべてのソース Docker レジストリーの認証資格情報を保管します。

IBM Cloud Platform Common Services、IBM MQ Operator イメージ、IBM MQ Advanced Developer イメージはすべて、認証を必要としないパブリック・レジストリーに保管されています。ただし、IBM MQ Advanced Server (Developer 以外)、他の製品、サード・パーティーのコンポーネントには、認証ありのレジストリーが1つ以上必要です。以下のレジストリーには認証が必要です。

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

これらのレジストリーについて詳しくは、[レジストリー名前空間の作成](#)を参照してください。

以下のコマンドを実行して、認証が必要なすべてのレジストリーの資格情報を構成する必要があります。該当するレジストリーごとに別個に以下のコマンドを実行します。

```
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"
```

コマンドは、\$HOME/.airgap/secrets ロケーションにあるファイル・システム上のファイルにレジストリー資格情報を保管およびキャッシュします。

2. ローカル Docker レジストリー接続情報を指定した環境変数を作成します。

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

注: Docker レジストリーは、80 や 443 などの標準ポートを使用します。Docker レジストリーで標準以外のポートを使用する場合は、構文 `host:port` を使用してポートを指定します。以下に例を示します。

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

3. ローカル Docker レジストリーの認証秘密を構成します。

注: このステップは、1 回のみ実行する必要があります。

```
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}"
```

コマンドは、\$HOME/.airgap/secrets ロケーションにあるファイル・システム上のファイルにレジストリー資格情報を保管およびキャッシュします。

4. グローバル・イメージ・プル秘密および **ImageContentSourcePolicy** を構成します。

a. ノードの再始動が必要かどうかを確認してください。

- OpenShift Container Platform バージョン 4.4 以降、およびエアギャップを使用する IBM MQ Operator の新規インストールでは、このステップによりすべてのクラスター・ノードが再始動されます。新しいプル秘密が適用されるまで、クラスター・リソースは使用不可になる可能性があります。
- IBM MQ Operator 1.8 では、CASE が更新され、イメージの追加ミラーリング・ソースが組み込まれました。そのため、以前のバージョンの IBM MQ Operator からバージョン 1.8 以上にアップグレードすると、ノードの再始動がトリガーされます。
- このステップでノードの再始動が必要かどうかを確認するには、このステップのコードに `--dry-run` オプションを追加します。これにより、最新の **ImageContentSourcePolicy** が生成され、コンソール・ウィンドウ (**stdout**) に表示されます。この **ImageContentSourcePolicy** 構成されているクラスターと異なる **ImageContentSourcePolicy** 場合は、再始動が行われます。

```
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. グローバル・イメージ・プル・シークレットと **ImageContentSourcePolicy** を構成するには、`--dry-run` オプションを指定せずにこのステップのコードを実行します。

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

5. **ImageContentSourcePolicy** リソースが作成されていることを確認します。

```
oc get imageContentSourcePolicy
```

6. オプション: 非セキュアなレジストリーを使用している場合は、ローカル・レジストリーをクラスターの **insecureRegistries** リストに追加する必要があります。

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

7. クラスター・ノード状況を確認します。

```
oc get nodes
```

imageContentsourcePolicy およびグローバル・イメージ・プル秘密が適用されると、ノード状況が **Ready**、**Scheduling**、または **Disabled** として表示されることがあります。すべてのノードで **Ready** 状況が表示されるまで待機します。

8. イメージをローカル・レジストリーにミラーリングします。

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

IBM MQ Operator のインストール

1. Red Hat OpenShift クラスターのウェブコンソールにログインします。
2. カタログ・ソースを作成します。前の手順を実行したのと同じ端末を使用してください。

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

3. 共通サービス・インストーラー・オペレーターの **CatalogSource** が作成されていることを確認します。

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

4. OLM を使用して IBM MQ Operator をインストールします。

- a. ナビゲーション・ペインで、「**オペレーター (Operators)**」 > 「**OperatorHub**」をクリックします。
「**OperatorHub**」ページが表示されます。
- b. 「**すべてのアイテム (All Items)**」フィールドに、「IBM MQ」と入力します。
IBM MQ カタログ・エントリーが表示されます。
- c. 「**IBM MQ**」を選択します。
「**IBM MQ**」ウィンドウが表示されます。
- d. 「**インストール**」をクリックします。
「**オペレーター・サブスクリプションの作成 (Create Operator Subscription)**」ページが表示されます。
- e. [11 ページの『IBM MQ Operator のバージョン・サポート』](#)を確認して、選択するオペレーター・チャンネルを判別します。
- f. 「**インストール・モード (Installation Mode)**」を、作成した特定の名前空間、または、クラスター全体の範囲のどちらかに設定します。
- g. 「**サブスクライブ**」をクリックします。
「**インストール済みのオペレーター (Installed Operators)**」ページに **IBM MQ** が追加されます。
- h. 「**インストール済みのオペレーター (Installed Operators)**」ページで、オペレーターの状況を確認します。インストールが完了すると、状況が **Succeeded** に変わります。

IBM MQ キュー・マネージャーのデプロイ

インストールしたオペレーターを使用して新規キュー・マネージャーを作成するには、[116 ページの『Red Hat OpenShift Container Platform クラスターにキュー・マネージャーをデプロイする方法』](#)を参照してください。

関連タスク

127 ページの『[非推奨]エア・ギャップ環境での最新の IBM MQ 2.x オペレーターまたはキュー・マネージャーへのアップグレードの準備』

インターネットに接続されていない Red Hat OpenShift クラスタでは、IBM MQ 2.x オペレーターまたはキュー・マネージャーをアップグレードする前に実行する必要がある準備ステップがあります。

OpenShift CP4I Red Hat OpenShift Container Platform クラスタにキュー・マネージャーをデプロイする方法

この例では、一時 (非永続) ストレージを使用する「クイック・スタート」キュー・マネージャーをデプロイし、IBM MQ セキュリティーをオフにします。メッセージは、キュー・マネージャーの再始動後は保持されません。構成を調整することで、キュー・マネージャーのさまざまな設定を変更できます。

このタスクについて

このタスクには、キュー・マネージャーを OpenShift にデプロイするための 3 つのオプションが用意されています。

1. [OpenShift コンソールを使用してキュー・マネージャーをデプロイします。](#)
2. [OpenShift CLI を使用してキュー・マネージャーをデプロイします。](#)
3. [IBM Cloud Pak for Integration Platform UI を使用してキュー・マネージャーをデプロイします。](#)

手順

- **オプション 1: OpenShift コンソールを使用してキュー・マネージャーをデプロイする。**

a) キュー・マネージャーをデプロイします。

- a. Red Hat OpenShift Container Platform クラスタ管理者資格情報を使用して、OpenShift コンソールにログインします。
- b. 「プロジェクト」を、IBM MQ Operator をインストールした名前空間に変更します。「プロジェクト」ドロップダウン・リストから名前空間を選択します。
- c. ナビゲーション・ペインで、「オペレーター」 > 「インストール済みオペレーター」をクリックします。
- d. 「インストール済みオペレーター (Installed Operators)」パネルのリストで、**IBM MQ** を見つけてクリックします。
- e. 「キュー・マネージャー (Queue Manager)」タブをクリックします。
- f. 「QueueManager の作成 (Create QueueManager)」ボタンをクリックします。インスタンス作成パネルが表示され、リソースを構成するための 2 つの方法 (フォーム・ビューと YAML ビュー) が示されます。デフォルトでは、「フォーム・ビュー」が選択されています。

b) キュー・マネージャーを構成します。

ステップ 2 オプション 1: フォーム・ビューで構成します。

「フォーム・ビュー」は、リソース構成を表示または変更するために使用できるフォームを開きます。

- a. 「ライセンス」の横にある矢印をクリックして、ライセンス承諾セクションを展開します。
- b. ご使用条件に同意する場合は、「ライセンスの受諾」を **true** に設定します。
- c. 矢印をクリックしてドロップダウン・リストを開き、ライセンスを選択します。IBM MQ は、複数の異なるライセンスで提供されています。有効なライセンスについて詳しくは、[183 ページの『mq.ibm.com/v1beta1 のライセンスのリファレンス』](#)を参照してください。キュー・マネージャーをデプロイするには、ライセンスに同意する必要があります
- d. 「作成」をクリックします。現在のプロジェクト (名前空間) 内のキュー・マネージャーのリストが表示されます。新しい QueueManager が Pending の状態になっているはずですが。

ステップ 2 オプション 2: YAML ビューで構成します。

YAML ビュー は、QueueManager の YAML ファイルの例を含むエディターを開きます。以下の手順に従って、ファイル内の値を更新します。

- a. `metadata.namespace` をプロジェクト (名前空間) 名に変更します。
 - b. `spec.license.license` の値を、要件に一致するライセンス・ストリングに変更します。ライセンスの詳細については、[183 ページの『mq.ibm.com/v1beta1 のライセンスのリファレンス』](#)を参照してください。
 - c. ご使用条件に同意する場合は、`spec.license.accept` を `true` に変更します。
 - d. 「作成」をクリックします。現在のプロジェクト (名前空間) 内のキュー・マネージャーのリストが表示されます。新しい QueueManager が Pending の状態になっているはずですが、
- c) キュー・マネージャーの作成を確認してください。
キュー・マネージャーが作成されたことを確認するには、以下の手順を実行します。
- a. IBM MQ Operator を作成した名前空間内であることを確認します。
 - b. 「ホーム」画面で、オペレーター > 「インストール済みオペレーター」をクリックし、キュー・マネージャーを作成した対象のインストール済み IBM MQ Operator を選択します。
 - c. 「キュー・マネージャー (Queue Manager)」タブをクリックします。QueueManager の状況が Running になったら、作成は完了です。
- オプション 2: OpenShift CLI を使用してキュー・マネージャーをデプロイする。
 - a) QueueManager YAML ファイルの作成

例えば、IBM Cloud Pak for Integration に基本的なキュー・マネージャーをインストールするには、以下を内容とするファイル「mq-quickstart.yaml」を作成します。

```
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: ご使用条件に同意する場合は、`accept: false` を `accept: true` に変更します。ライセンスについて詳しくは、[183 ページの『mq.ibm.com/v1beta1 のライセンスのリファレンス』](#)を参照してください。

この例には、キュー・マネージャーとともにデプロイされ、IBM Cloud Pak for Integration 内でシングル・サインオンを使用して Web コンソールが使用可能になっている Web サーバーも含まれています。 **V 9.3.4** IBM Cloud Pak for Integration 2023.4.1 以降、シングル・サインオンを機能させるには、まず [他の IBM Cloud Pak for Integration コンポーネントをインストールする必要があります](#)。

IBM Cloud Pak for Integration とは別に、基本的なキュー・マネージャーをインストールするには、以下を内容とするファイル「mq-quickstart.yaml」を作成します。

```
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: MQ のご使用条件に同意する場合は、`accept: false` を `accept: true` に変更します。ライセンスについて詳しくは、[183 ページの『mq.ibm.com/v1beta1 のライセンスのリファレンス』](#)を参照してください。

- b) QueueManager オブジェクトを作成します。

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

- c) キュー・マネージャーの作成を確認してください。

以下の手順を実行して、キュー・マネージャーが作成されたことを確認します。

- a. デプロイメントを検証します。

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

- b. 状況を確認します。

```
oc describe queuemanager quickstart
```

- **オプション 3: IBM Cloud Pak for Integration Platform UI を使用してキュー・マネージャーをデプロイする。**

- a) ブラウザーで、IBM Cloud Pak for Integration Platform UI を起動します。

- b) IBM Cloud Pak for Integration Platform UI で、「**インスタンスの作成**」をクリックします。

- c) 「**メッセージング**」を選択し、「**次へ**」をクリックします。

インスタンス作成パネルが表示され、リソースを構成するための 2 つの方法 (**フォーム・ビュー** と **YAML ビュー**) が示されます。デフォルトでは、「**フォーム・ビュー**」が選択されています。

- d) 「**詳細**」セクションで、「**名前**」フィールドを確認または更新し、キュー・マネージャー・インスタンスを作成する**名前空間**を指定します。

- e) IBM Cloud Pak for Integration の使用条件に同意する場合は、「**ライセンスへの同意 (License acceptance)**」を「**オン (On)**」に変更します。

ライセンスについて詳しくは、[183 ページの『mq.ibm.com/v1beta1 のライセンスのリファレンス』](#)を参照してください。キュー・マネージャーをデプロイするには、ライセンスに同意する必要があります

- f) 「**キュー・マネージャー (Queue Manager)**」セクションで、基礎キュー・マネージャーの**名前**を確認または更新します。古いバージョンの IBM Cloud Pak for Integration Platform UI で、「**キュー・マネージャーの構成 (Queue Manager Config)**」セクションを使用します。

デフォルトでは、IBM MQ クライアント・アプリケーションによって使用されるキュー・マネージャーの名前は、QueueManager の名前と同じですが、無効文字 (ハイフンなど) は削除されます。

- g) **作成** をクリックします。

現在のプロジェクト (名前空間) 内のキュー・マネージャーのリストが表示されます。新しい QueueManager の状況は Pending になっているはずです。

- h) キュー・マネージャーの作成を確認してください。

QueueManager の状況が Running になったら、作成は完了です。

関連タスク

[155 ページの『Red Hat OpenShift クラスターの外部からキュー・マネージャーに接続するためのルートの構成』](#)

Red Hat OpenShift クラスターの外部から IBM MQ キュー・マネージャーにアプリケーションを接続するには、Red Hat OpenShift 経路が必要です。IBM MQ キュー・マネージャーおよびクライアント・アプリケーションで TLS を有効にする必要があります。SNI は、TLS 1.2 以上のプロトコルが使用されている場合にの

み TLS プロトコルで使用できるためです。Red Hat OpenShift Container Platform Router では、IBM MQ キュー・マネージャーへの要求のルーティングに SNI が使用されます。

[173 ページの『IBM MQ Console クラスターにデプロイされた Red Hat OpenShift への接続』](#)

Red Hat OpenShift Container Platform クラスターにデプロイされているキュー・マネージャーの IBM MQ Console に接続する方法について説明します。

[135 ページの『キュー・マネージャーの構成例』](#)

QueueManager カスタム・リソースの内容を調整することによってキュー・マネージャーを構成できます。

OpenShift CP4I IBM MQ Operator のアンインストール

Red Hat OpenShift コンソールまたは CLI を使用して、Red Hat OpenShift から IBM MQ Operator をアンインストールできます。

手順

- オプション 1: OpenShift コンソールを使用して IBM MQ Operator をアンインストールします。

注: クラスター上のすべてのプロジェクト/名前空間に IBM MQ Operator がインストールされている場合は、キュー・マネージャーを削除するプロジェクトごとに、以下の手順のステップ 2 から 6 を繰り返します。

- a) Red Hat OpenShift Container Platform クラスター管理者資格情報を使用して、Red Hat OpenShift Container Platform Web コンソールにログインします。
- b) 「プロジェクト」を、IBM MQ Operator をアンインストールする名前空間に変更します。「プロジェクト」ドロップダウン・リストから名前空間を選択します。
- c) ナビゲーション・ペインで、「オペレーター」>「インストール済みオペレーター」をクリックします。
- d) 「IBM MQ」オペレーターをクリックします。
- e) 「キュー・マネージャー」タブをクリックして、この IBM MQ Operator によって管理されているキュー・マネージャーを表示します。
- f) 1 つ以上のキュー・マネージャーを削除します。
これらのキュー・マネージャーは引き続き稼働しますが、IBM MQ Operator がいない状態では正常に機能しない場合があることに注意してください。
- g) オプション: 必要に応じて、キュー・マネージャーを削除するプロジェクトごとにステップ 2 から 6 を繰り返します。
- h) 「オペレーター (Operators)」>「インストール済みのオペレーター (Installed Operators)」に戻ります。
- i) 「IBM MQ」オペレーターの横にある 3 点メニューをクリックし、「オペレーターのアンインストール (Uninstall Operator)」を選択します。

- オプション 2: OpenShift CLI を使用して IBM MQ Operator をアンインストールする

- a) `oc login` を使用して Red Hat OpenShift クラスターにログインします。
- b) IBM MQ Operator が 1 つの名前空間にインストールされている場合は、以下のサブステップを実行します。
 - a. アンインストールする IBM MQ Operator が含まれているプロジェクトにいることを確認します。

```
oc project <project_name>
```

- b. プロジェクトにインストールされているキュー・マネージャーを表示します。

```
oc get qmgr
```

- c. 1 つ以上のキュー・マネージャーを削除します。

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

これらのキュー・マネージャーは引き続き稼働しますが、IBM MQ Operator がない状態では正常に機能しない場合があることに注意してください。

- d. **ClusterServiceVersion** インスタンスを表示します。

```
oc get csv
```

- e. IBM MQ **ClusterServiceVersion** を削除します。

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

- f. サブスクリプションを表示します。

```
oc get subscription
```

- g. すべてのサブスクリプションを削除します。

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

- h. 共通サービスを使用しているものがほかにない場合は、共通サービス・オペレーターをアンインストールし、オペレーター・グループを削除できます。

i) IBM Cloud Pak foundational services 製品資料の「[Uninstalling foundational services](#)」の説明に従って、共通サービス・オペレーターをアンインストールします。

- ii) オペレーター・グループを表示します。

```
oc get operatorgroup
```

- iii) オペレーター・グループを削除します。

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

- c) IBM MQ Operator がクラスター上のすべての名前空間にインストールされて使用可能な状態になっている場合は、以下のサブステップを実行します。

- a. インストールされているすべてのキュー・マネージャーを表示します。

```
oc get qmgr -A
```

- b. 1つ以上のキュー・マネージャーを削除します。

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

これらのキュー・マネージャーは引き続き稼働しますが、IBM MQ Operator がない状態では正常に機能しない場合があることに注意してください。

- c. **ClusterServiceVersion** インスタンスを表示します。

```
oc get csv -A
```

- d. クラスターから IBM MQ **ClusterServiceVersion** を削除します。

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

- e. サブスクリプションを表示します。

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

- f. サブスクリプションを削除します。

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

- g. オプション: 共通サービスを使用しているものが他にない場合は、共通サービス・オペレーターをアンインストールすることができます。これを行うには、IBM Cloud Pak foundational services 製品資料の「[基本サービスのアンインストール](#)」の説明に従ってください。

グレード

IBM MQ Operator の Continuous Delivery (CD) バージョンと Long Term Support (LTS) バージョンには、異なるアップグレード・プロセスがあります。ご使用のデプロイメント・タイプのアップグレード・ステップを実行します。

このタスクについて

IBM MQ Operator およびキュー・マネージャーをアップグレードするには、以下のいずれかのステップを実行します。

手順

- オプション 1: 現在のオペレーター・チャンネルでデプロイメントを最新バージョンにアップグレードします。

現在のオペレーター・チャンネルで IBM MQ Operator のデプロイメントを最新バージョンにアップグレードするには、[121 ページの『IBM MQ Operator チャンネルの最新セキュリティ・リリースへのアップグレード』](#)を参照してください。

- オプション 2: **CD デプロイメントをアップグレード**します。

IBM MQ Operator の以前の CD デプロイメントを最新の CD バージョンの IBM MQ Operator (バージョン 3.1.3) にアップグレードするには、[123 ページの『IBM MQ Operator の現行 CD チャンネルへのマイグレーション』](#)を参照してください。

注:

バージョン 2.0.x は CD と LTS の両方のリリースとしてリリースされたため、[123 ページの『IBM MQ Operator の現行 CD チャンネルへのマイグレーション』](#)の手順を使用して、任意のバージョン 2.0.x IBM MQ Operator から最新の CD バージョンの IBM MQ Operator にアップグレードできます。

IBM MQ Operator チャンネルの最新セキュリティ・リリースへのアップグレード

IBM MQ Operator をアップグレードすると、キュー・マネージャーをアップグレードできます。

始める前に

重要: このトピックでは、IBM MQ Operator のデプロイメントを、デプロイメントのチャンネル上の最新のセキュリティ・リリースにアップグレードします。これがご使用のデプロイメントに該当しない場合は、[121 ページの『IBM MQ Operator とキュー・マネージャーのアップグレード』](#)で説明されている代替アップグレード・パスを参照してください。

インターネットに接続されていない Red Hat OpenShift クラスター内の IBM MQ Operator のデプロイメントの場合は、[127 ページの『\[非推奨\]エア・ギャップ環境での最新の IBM MQ 2.x オペレーターまたはキュー・マネージャーへのアップグレードの準備』](#)の手順に従います。

手順

1. IBM MQ Operator を新しいバージョンにアップグレードします。

自動アップグレードが設定されている場合、新しいセキュリティ・リリースがリリースされると、IBM MQ Operator はアップグレードを完了します。

自動アップグレードが設定されていない場合は、IBM MQ Operator のアップグレードを手動で承認します。

- 使用可能なアップグレードがある場合、**Upgrade Status** は「Upgrade available」の可能性がります。

- この場合、IBM MQ Operator をアップグレードする **InstallPlan** を承認するために使用できる制御が存在する可能性があります。

2. IBM MQ キュー・マネージャーを新しいバージョンにアップグレードします。

以下の表に、アクティブなオペレーター・チャンネルごとの IBM MQ キュー・マネージャーの最新バージョンを示します。関連するバージョンを使用して、132 ページの『Red Hat OpenShift を使用した IBM MQ キュー・マネージャーのアップグレード』の手順に従います。

オペレーター・チャンネル	最新の IBM MQ キュー・マネージャー
 V2.0 (LTS)	9.3.0.17-r3
 V3.1 (CD)	9.3.5.1-r2

IBM MQ Operator の LTS チャンネルへのマイグレーション

IBM MQ Operator をアップグレードすると、キュー・マネージャーをアップグレードできます。

始める前に

重要: このトピックでは、1.3.x Long Term Support (LTS) IBM MQ Operator のデプロイメントを IBM MQ Operator 2.0.x のみの LTS ストリームにアップグレードします。これがご使用のデプロイメントに該当しない場合は、[IBM MQ Operator およびキュー・マネージャーのアップグレード](#)で説明されている代替アップグレード・パスを参照してください。

インターネット接続のない Red Hat OpenShift クラスターへの IBM MQ Operator のデプロイメントの場合は、[127 ページの『\[非推奨\]エア・ギャップ環境での最新の IBM MQ 2.x オペレーターまたはキュー・マネージャーへのアップグレードの準備』](#)の手順に従ってください。

重要: IBM MQ Operator 2.0.x には、以下が必要です。

- Red Hat OpenShift Container Platform 4.12.

アップグレードするには、「[Red Hat OpenShift のアップグレード](#)」の手順に従ってください。

- IBM Cloud Pak foundational services 3.19.x

IBM MQ Operator 1.3.x (2020.4) からアップグレードすると、複数インスタンス・キュー・マネージャーの両方のインスタンスが同時に再始動されます。これは、IBM MQ のバージョンを 9.2.0.5-r3-eus に変更すると発生します。IBM MQ Operator 1.3.x から 2.0.x にアップグレードすると、IBM MQ キュー・マネージャーのローリング更新が行われます。IBM Cloud Pak for Integration Platform UI がインストールされている場合、IBM Cloud Pak for Integration Platform UI のバージョンを 2020.4.1-8-eus に変更し、2022.2.1-0 に変更すると、IBM MQ の追加の再始動が行われます。

手順

1. ステップ 2 のリンクをたどる前に、アップグレードに必要な以下の情報をお読みください。

- インストールしていないコンポーネントについては、すべてのサブステップを省略する必要があります。これがインストールされていない場合は、IBM Cloud Pak for Integration Platform UI が含まれます。
- ステップ 2 では、IBM Cloud Pak for Integration の資料が表示されます。アップグレード・プロセス中に、IBM MQ のトピック「[IBM MQ キュー・マネージャーのアップグレード](#)」に戻り、IBM MQ オペランドをアップグレードします。
- すべての IBM MQ ユーザーは、ステップ 2 のリンクに記載されている手順を使用して、少なくとも以下のタスクを実行することをお勧めします。また、ご使用の環境に適用される他のすべてのタスクも実行することをお勧めします。
 - パッチ IBM MQ Operator およびオペランド (パッチ 2020.4):

- v1.3-eus オペレーター・チャンネルで、IBM MQ Operator を少なくともバージョン 1.3.5 にアップグレードします。
- IBM MQ オペラント (キュー・マネージャー・コンテナ・イメージ) を少なくともバージョン 9.2.0.5-r3-eus にアップグレードします。

注: IBM MQ オペラントを少なくともこのバージョンに更新することをお勧めしますが、これは必須ではありません。

- 依存関係のアップグレード:
 - IBM Cloud Pak foundational services をアップグレードします。
 - OpenShift Container Platform をアップグレードします。
 - オペレーターをアップグレードします。
 - IBM MQ Operator を 2.0.23 にアップグレードします。
 - 機能をアップグレードします。
 - IBM MQ オペラント (キュー・マネージャー・コンテナ・イメージ) を最新の 9.3.0 バージョン (9.3.0.17-r3) にアップグレードして、最新のセキュリティ・フィックスを受け取ります。
2. IBM MQ Operator 1.3-eus (IBM Cloud Pak for Integration 2020.4) からのアップグレードを実行して、IBM MQ Operator とキュー・マネージャーをアップグレードします。

関連タスク

[127 ページの『\[非推奨\]エア・ギャップ環境での最新の IBM MQ 2.x オペレーターまたはキュー・マネージャーへのアップグレードの準備』](#)

インターネットに接続されていない Red Hat OpenShift クラスターでは、IBM MQ 2.x オペレーターまたはキュー・マネージャーをアップグレードする前に実行する必要がある準備ステップがあります。

[130 ページの『Red Hat OpenShift を使用した IBM MQ Operator のアップグレード』](#)

Red Hat OpenShift Web コンソールまたは CLI を使用して、IBM MQ Operator をアップグレードできます。

[132 ページの『Red Hat OpenShift を使用した IBM MQ キュー・マネージャーのアップグレード』](#)

OpenShift CP4I CD IBM MQ Operator の現行 CD チャンネルへのマイグレーション

以前の IBM MQ Operator からバージョン 3.1.3 にアップグレードします。オペレーターをアップグレードすると、キュー・マネージャーをアップグレードできます。

始める前に

このトピックでは、3.1.0 より前の IBM MQ Operator の Continuous Delivery (CD) デプロイメントをバージョン 3.1.3 のみにアップグレードします。これがご使用のデプロイメントに該当しない場合は、[IBM MQ Operator およびキュー・マネージャーのアップグレード](#)で説明されている代替アップグレード・パスを参照してください。

IBM MQ Operator 3.1.3 にアップグレードするには、Red Hat OpenShift Container Platform 4.12 以降を実行している必要があります。プラットフォームをアップグレードするには、[Red Hat OpenShift のアップグレード](#)を参照してください。

注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これらのリリースは、4.14 や 4.16 などの偶数のマイナー・リリースです。

手順

1. オプション: **2.0.0 より前の CD バージョンである IBM MQ Operator をアップグレード**します。

IBM MQ Operator が現在 1.x CD バージョンである場合は、まず [125 ページの『1.x CD IBM MQ Operator のバージョン 2.0.x へのマイグレーション』](#)の手順に従ってから、ここに戻って最新の CD バージョンにアップグレードします。

2. オプション: 現在バージョン **2.2.x** または **2.3.x** の **IBM MQ Operator** を **2.4.x** にアップグレードします。

ご使用の IBM MQ Operator が現在 2.2.x または 2.3.x のバージョンである場合は、[126 ページの『IBM MQ Operator の v2.4 チャンネルへのマイグレーション』](#)の関連手順に従ってから、ここに戻って最新の CD バージョンにアップグレードしてください。これは、バージョン 3.1.3 にアップグレードする前の必須の前提条件ステップであることに注意してください。

3. コンポーネントをアップグレードします。

次のオプションのいずれかを選択してください。

- **オプション 1:** CP4I ユーザーである場合、または CP4I ライセンスを使用して少なくとも 1 つのキュー・マネージャーをデプロイした場合は、生成されたアップグレード計画を使用して、**すべてのコンポーネントのアップグレード**(IBM MQ Operator とキュー・マネージャーを含む)に関連する手順に従います。

- バージョン 2023.2 からアップグレードするには、[アップグレード計画の生成による 2023.2 からのアップグレード](#)を参照してください。

- バージョン 2022.2 からアップグレードするには、[アップグレード・プランの生成による 2022.2 からのアップグレード](#)を参照してください。

- **オプション 2:** 他のすべてのユーザーの場合:

a. ミラー・イメージ(エア・ギャップのみ)。

IBM MQ イメージをミラーリングする必要があります。これらの値のみを使用して、以下のリンクでステップを実行します。

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

以前のインストールまたはアップグレード時にイメージ・レジストリーへの接続がセットアップされている必要があるため、セクション 3.5 「クラスタの構成」を省略する必要があります。

リンク: [エアギャップ・クラスタ用のミラーリング・イメージ](#)。

b. **IBM MQ Operator** を **3.1.3** にアップグレードします。

[130 ページの『Red Hat OpenShift を使用した IBM MQ Operator のアップグレード』](#)を参照してください。

c. インスタンスをアップグレードします。

最新の機能とセキュリティ・フィックスを入手するには、IBM MQ Operand (キュー・マネージャー・コンテナ・イメージ)を最新の CD バージョン (9.3.5.1-r2) にアップグレードします。[132 ページの『Red Hat OpenShift を使用した IBM MQ キュー・マネージャーのアップグレード』](#)を参照してください。

4. オプション: **Red Hat OpenShift Container Platform 4.12** を **4.14** にアップグレードします。

IBM MQ Operator 3.0.0 以降では、Red Hat OpenShift Container Platform 4.12 が必要です。オプションで、さらに Red Hat OpenShift 4.14 にアップグレードすることを選択できることに注意してください。各 IBM MQ Operator チャンネルの互換性のあるバージョンを確認するには、[13 ページの『互換性のある Red Hat OpenShift Container Platform のバージョン』](#)を参照してください。アップグレードするには、[Red Hat OpenShift のアップグレード](#)を参照してください。

5. オプション: **IBM MQ Operator** の特定のカタログ・ソースをピン留めします。

アップグレードするインストール済み環境で IBM MQ Operator カタログが使用されている場合は、IBM MQ Operator の特定のカタログ・ソースをピン留めする必要があります。[各オペレーター特定のカタログ・ソースへの移動](#)を参照してください。

関連タスク

[127 ページの『\[非推奨\]エア・ギャップ環境での最新の IBM MQ 2.x オペレーターまたはキュー・マネージャーへのアップグレードの準備』](#)

インターネットに接続されていない Red Hat OpenShift クラスターでは、IBM MQ 2.x オペレーターまたはキュー・マネージャーをアップグレードする前に実行する必要がある準備ステップがあります。

OpenShift CP4I Operator 2.0 1.x CD IBM MQ Operator のバージョン 2.0.x へのマイグレーション

IBM MQ Operator をアップグレードすると、キュー・マネージャーをアップグレードできます。

始める前に

重要: このトピックは、IBM MQ Operator バージョン 2.0.x より前のバージョンの Continuous Delivery (CD) デプロイメントをバージョン 2.0.x のみにアップグレードするためのものです。これがデプロイメントに該当しない場合は、[IBM MQ Operator およびキュー・マネージャーのアップグレードで説明されている代替アップグレード・パスを参照してください](#)。

インターネット接続のない Red Hat OpenShift クラスターへの IBM MQ Operator のデプロイメントの場合は、[127 ページの『\[非推奨\]エア・ギャップ環境での最新の IBM MQ 2.x オペレーターまたはキュー・マネージャーへのアップグレードの準備』の手順に従ってください](#)。

このアップグレードを完了するには、IBM MQ Operator 2.0.0 の以下の要件を満たす必要があります。

- Red Hat OpenShift Container Platform 4.12.
アップグレードするには、「[Red Hat OpenShift のアップグレード](#)」の手順に従ってください。
- IBM Cloud Pak foundational services 3.19

手順

1. ステップ 2 のリンクをたどる前に、アップグレードに必要な以下の情報をお読みください。

- インストールしていないコンポーネントのすべてのサブステップを省略します。これがインストールされていない場合は、IBM Cloud Pak for Integration Platform UI が含まれます。
- ステップ 2 では、IBM Cloud Pak for Integration の資料が表示されます。アップグレード・プロセス中に、IBM MQ のトピック「[IBM MQ キュー・マネージャーのアップグレード](#)」に戻り、IBM MQ オペランドをアップグレードします。
- すべての IBM MQ ユーザーは、ステップ 2 のリンクに記載されている手順を使用して、少なくとも以下のタスクを実行することをお勧めします。また、ご使用の環境に適用される他のすべてのタスクも実行することをお勧めします。
 - パッチ IBM MQ Operator およびオペランド (パッチ 2021.4):
 - v1.8 オペレーター・チャンネルで、IBM MQ Operator を少なくともバージョン 1.8.0 にアップグレードします。
 - IBM MQ オペランド (キュー・マネージャー・コンテナ・イメージ) を少なくともバージョン 9.2.5.0-r3 にアップグレードします。
注: IBM MQ オペランドを現行バージョン (9.3.0.17-r3) に更新することをお勧めしますが、これは必須ではありません。
 - 依存関係のアップグレード:
 - IBM Cloud Pak foundational services をアップグレードします。
 - OpenShift Container Platform をアップグレードします。
 - オペレーターをアップグレードします。
 - IBM MQ Operator を 2.0.23 にアップグレードします。
 - 機能をアップグレードします。
 - IBM MQ オペランド (キュー・マネージャー・コンテナ・イメージ) を最新の 9.3.0 バージョン (9.3.0.17-r3) にアップグレードして、最新のセキュリティ・フィックスを受け取ります。

2. [IBM MQ Operator 1.8 \(IBM Cloud Pak for Integration 2021.4\)](#) 以前の CD IBM MQ Operator バージョンからのアップグレードを実行して、IBM MQ Operator とキュー・マネージャーをアップグレードします。

次のタスク

これで、IBM MQ Operator およびキュー・マネージャーを最新の CD バージョン (3.1.3) にアップグレードする準備ができました。 [123 ページの『IBM MQ Operator の現行 CD チャンネルへのマイグレーション』](#)を参照してください。

IBM MQ Operator の v2.4 チャンネルへのマイグレーション

IBM MQ Operator をアップグレードすると、キュー・マネージャーをアップグレードできます。

始める前に

重要: このトピックでは、バージョン 2.4.0 より前の IBM MQ Operator の Continuous Delivery (CD) デプロイメントをバージョン 2.4.8 のみにアップグレードします。これは、最新の CD バージョンの IBM MQ Operator にアップグレードするための中間ステップです。v2.4 チャンネルはセキュリティー更新を受け取りません。これがデプロイメントに該当しない場合は、[IBM MQ Operator およびキュー・マネージャーのアップグレード](#)で説明されている代替アップグレード・パスを参照してください。

インターネットに接続されていない Red Hat OpenShift クラスター内の IBM MQ Operator のデプロイメントの場合は、[127 ページの『\[非推奨\]エア・ギャップ環境での最新の IBM MQ 2.x オペレーターまたはキュー・マネージャーへのアップグレードの準備』](#)の手順に従います。

このアップグレードを完了するには、IBM MQ Operator 2.4.8 に関する以下の要件を満たす必要があります。

- Red Hat OpenShift Container Platform 4.12.

アップグレードするには、[Red Hat OpenShift のアップグレード](#)の手順に従ってください。

注: OpenShift Container Platform Extended Update Support (EUS) リリースのみがサポートされます。これは、偶数番号のマイナー・リリースです (例えば、4.14 や 4.16)。

- IBM Cloud Pak foundational services 3.19 から 3.24 (両端を含む)。

手順

1. オプション: 現在 CD バージョンが 2.0.0 より前のバージョンである IBM MQ Operator をアップグレードします。

ご使用の IBM MQ Operator が現在 1.x CD バージョンである場合は、まず [125 ページの『1.x CD IBM MQ Operator のバージョン 2.0.x へのマイグレーション』](#)の手順に従い、次にここに戻って最新の 2.4 バージョンにアップグレードします。

2. CD バージョン 2.x.x の IBM MQ Operator を最新の 2.4 バージョン (2.4.8) にアップグレードします。

[130 ページの『Red Hat OpenShift を使用した IBM MQ Operator のアップグレード』](#)に示されている手順に従う。

3. オプション: IBM Cloud Pak for Integration の他のコンポーネントをアップグレードします。

IBM Cloud Pak for Integration のユーザーの場合は、アップグレードする他のコンポーネントが存在する可能性があります。他のコンポーネントをアップグレードする手順については、ご使用のデプロイメントに応じて以下の関連ステップを参照してください。

- オプション 1: [IBM MQ Operator 2.0.x/2.1.x からのアップグレード](#) (IBM Cloud Pak for Integration 2022.2)。
- オプション 2: [IBM MQ Operator 2.2.x/2.3.x からのアップグレード](#) (IBM Cloud Pak for Integration 2022.4)。

4. オプション: IBM Cloud Pak foundational services をアップグレードします。

IBM Cloud Pak for Integration のユーザーの場合は、IBM Cloud Pak foundational services をバージョン 3.19.x からバージョン 3.24.x にアップグレードすることをお勧めします。このアップグレードを実行する手順については、[IBM Cloud Pak foundational services のアップグレード](#)を参照してください。

関連タスク

[127 ページの『\[非推奨\]エア・ギャップ環境での最新の IBM MQ 2.x オペレーターまたはキュー・マネージャーへのアップグレードの準備』](#)

インターネットに接続されていない Red Hat OpenShift クラスターでは、IBM MQ 2.x オペレーターまたはキュー・マネージャーをアップグレードする前に実行する必要がある準備ステップがあります。

[130 ページの『Red Hat OpenShift を使用した IBM MQ Operator のアップグレード』](#)

Red Hat OpenShift Web コンソールまたは CLI を使用して、IBM MQ Operator をアップグレードできます。

[132 ページの『Red Hat OpenShift を使用した IBM MQ キュー・マネージャーのアップグレード』](#)

エア・ギャップ環境での最新の IBM MQ 2.x オペレーターまたはキュー・マネージャーへのアップグレードの準備

インターネットに接続されていない Red Hat OpenShift クラスターでは、IBM MQ 2.x オペレーターまたはキュー・マネージャーをアップグレードする前に実行する必要がある準備ステップがあります。

始める前に

注：以下の手順は、エア・ギャップ環境で 2.x バージョンの IBM MQ Operator にアップグレードするためのものです。IBM MQ Operator 3.0.0 以降にアップグレードするには、エア・ギャップ固有のステップに特に注意して、[121 ページの『IBM MQ Operator とキュー・マネージャーのアップグレード』](#)を参照してください。

このトピックでは、以前にリリースされた IBM Cloud Pak for Integration イメージがミラーリングされるローカル・イメージ・レジストリーが既に構成されていることを前提としています。

このタスクについて

エアギャップ環境で IBM MQ Operator またはキュー・マネージャーをアップグレードする前に、最新の IBM Cloud Pak for Integration イメージをミラーリングする必要があります。

このタスクの最初の 4 つのステップは、[110 ページの『エア・ギャップ環境での IBM MQ Operator 2.x のインストール』](#)時に実行するステップと同じであることに注意してください。

手順

1. インストーラーおよびイメージ・インベントリー用の環境変数を作成します。

インストーラー・イメージ名とイメージ・インベントリーを使用して、以下の環境変数を作成します。

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

ここで、*version_number* は、エアギャップ・インストールのために使用するケースのバージョンです。使用可能なケースのバージョンのリストについては、<https://github.com/IBM/cloud-pak/tree/master/repo/case/ibm-mq> を参照してください。選択するオペレーター・チャンネルを決定するには、[IBM MQ Operator のバージョン・サポート](#)を参照してください。

2. IBM MQ インストーラーおよびイメージ・インベントリーをダウンロードします。

ibm-mq インストーラーおよびイメージ・インベントリーを要塞ホストにダウンロードします。

```
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. クラスター管理者として OpenShift Container Platform クラスターにログインします。

以下に、OpenShift Container Platform クラスターにログインするためのコマンド例を示します。

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

4. イメージをミラーリングして、クラスターを構成します。

以下のステップを実行して、イメージをミラーリングし、クラスターを構成します。

注: いずれのコマンドでも二重引用符の間にティルドは使用しないでください。例えば、args "--registry registry --user registry_userid --pass registry_password --inputDir ~/offline" は使用しないでください。波形記号が展開されず、コマンドが失敗する可能性があります。

a. すべてのソース Docker レジストリーの認証資格情報を保管します。

IBM Cloud Platform Common Services、IBM MQ Operator イメージ、IBM MQ Advanced Developer イメージはすべて、認証を必要としないパブリック・レジストリーに保管されています。ただし、IBM MQ Advanced Server (Developer 以外)、他の製品、サード・パーティーのコンポーネントには、認証ありのレジストリーが1つ以上必要です。以下のレジストリーには認証が必要です。

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

これらのレジストリーについては、[レジストリー名前空間の作成](#)を参照してください。

以下のコマンドを実行して、認証が必要なすべてのレジストリーの資格情報を構成する必要があります。該当するレジストリーごとに別個に以下のコマンドを実行します。

```
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"
```

コマンドは、\$HOME/.airgap/secrets ロケーションにあるファイル・システム上のファイルにレジストリー資格情報を保管およびキャッシュします。

b. ローカル Docker レジストリー接続情報を指定した環境変数を作成します。

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

注: Docker レジストリーは、80 や 443 などの標準ポートを使用します。Docker レジストリーで標準以外のポートを使用する場合は、構文 `host:port` を使用してポートを指定します。以下に例を示します。

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

c. ローカル Docker レジストリーの認証秘密を構成します。

注: このステップは、1 回のみ実行する必要があります。

```
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}"
```

コマンドは、\$HOME/.airgap/secrets ロケーションにあるファイル・システム上のファイルにレジストリー資格情報を保管およびキャッシュします。

d. グローバル・イメージ・プル秘密および **ImageContentSourcePolicy** を構成します。

i) ノードの再始動が必要かどうかを確認してください。

- OpenShift Container Platform バージョン 4.4 以降、およびエアギャップを使用する IBM MQ Operator の新規インストールでは、このステップによりすべてのクラスター・ノードが再始動されます。新しいプル秘密が適用されるまで、クラスター・リソースは使用不可になる可能性があります。
- IBM MQ Operator 1.8 では、CASE が更新され、イメージの追加ミラーリング・ソースが組み込まれました。そのため、以前のバージョンの IBM MQ Operator からバージョン 1.8 以上にアップグレードすると、ノードの再始動がトリガーされます。
- このステップでノードの再始動が必要かどうかを確認するには、このステップのコードに `--dry-run` オプションを追加します。これにより、最新の **ImageContentSourcePolicy** が生成され、コンソール・ウィンドウ (**stdout**) に表示されます。この **ImageContentSourcePolicy** 構成されているクラスターと異なる **ImageContentSourcePolicy** 場合は、再始動が行われます。

```
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) グローバル・イメージ・プル・シークレットと **ImageContentSourcePolicy** を構成するには、`--dry-run` オプションを指定せずにこのステップのコードを実行します。

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

e. **ImageContentSourcePolicy** リソースが作成されていることを確認します。

```
oc get imageContentSourcePolicy
```

f. オプション: 非セキュアなレジストリーを使用している場合は、ローカル・レジストリーをクラスターの **insecureRegistries** リストに追加する必要があります。

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

g. クラスター・ノード状況を確認します。

```
oc get nodes
```

imageContentsourcePolicy およびグローバル・イメージ・プル秘密が適用されると、ノード状況が **Ready**、**Scheduling**、または **Disabled** として表示されることがあります。すべてのノードで **Ready** 状況が表示されるまで待機します。

h. イメージをローカル・レジストリーにミラーリングします。

```
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. カタログ・ソースをアップグレードします。

前の手順を実行したのと同じ端末を使用してください。

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

次のタスク

IBM Cloud Pak for Integration のアップグレードを完了するには、IBM Cloud Pak for Integration の資料に戻る必要がある場合があります。

そうでない場合は、以下のいずれかのタスクを実行して、IBM MQ Operator とキュー・マネージャーをアップグレードする準備ができました。

- [130 ページの『Red Hat OpenShift を使用した IBM MQ Operator のアップグレード』](#)
- [132 ページの『Red Hat OpenShift を使用した IBM MQ キュー・マネージャーのアップグレード』](#)

Red Hat OpenShift を使用した IBM MQ Operator のアップグレード

Red Hat OpenShift Web コンソールまたは CLI を使用して、IBM MQ Operator をアップグレードできます。

手順

Red Hat OpenShift を使用して IBM MQ Operator をアップグレードするには、以下のいずれかのタスクを実行します。

- [130 ページの『Red Hat OpenShift ウェブコンソールを使用して IBM MQ Operator をアップグレードする』](#)
- [131 ページの『Red Hat OpenShift CLI を使用して IBM MQ Operator をアップグレードする』](#)

Red Hat OpenShift ウェブコンソールを使用して IBM MQ Operator をアップグレードする

Operator Hub を使用して、IBM MQ Operator をアップグレードできます。

始める前に

注：最新の CD バージョンの IBM MQ Operator は 3.1.3 です。最新の LTS バージョンの IBM MQ Operator は 2.0.23 です。最新の IBM MQ Operator リリース・ノートについては、[35 ページの『IBM MQ Operator のリリース履歴』](#)を参照してください。

Red Hat OpenShift クラスターのウェブコンソールにログインします。

手順

1. [11 ページの『IBM MQ Operator のバージョン・サポート』](#)を確認して、どのオペレーター・チャンネルにアップグレードすべきかを判別します。
2. 最新のカatalog・ソースを適用します。

ibm-operator-catalog ではなく、IBM MQ 固有のカatalog・ソース (すべてのエア・ギャップ・インストール) を使用する場合は、IBM MQ バージョンのカatalog・ソースを適用する必要があります。

[クラスターへのCatalog・ソースの追加の手順](#)に従います。

注: エア・ギャップ「[イメージのミラーリング \(エア・ギャップのみ\) \(Mirror images \(air-gap only\)\)](#)」のオペレーター・インストール・ステップを既に完了している場合は、[カタログ・ソースを適用するステップ](#)を実行するだけで済みます。以下に例を示します。

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

3. IBM MQ Operator をアップグレードします。IBM MQ Operator の新規のメジャー/マイナーのバージョンは、新しいサブスクリプション・チャンネルを介して配信されます。Operator を新規のメジャー/マイナーのバージョンにアップグレードするには、IBM MQ Operator の「サブスクリプション」で、選択したチャンネルを更新する必要があります。
 - a) ナビゲーション・ペインで、「オペレーター (Operators)」 > 「インストール済みのオペレーター (Installed Operators)」をクリックします。
指定したプロジェクトにインストールされているすべてのオペレーターが表示されます。
 - b) 「IBM MQ Operator」を選択します。
 - c) 「サブスクリプション」タブにナビゲートします。
 - d) 「チャンネル」をクリックします。
「サブスクリプション更新チャンネルの変更 (Change Subscription Update Channel)」ウィンドウが表示されます。
 - e) 必要なチャンネルを選択して、「保存」をクリックします。
新しいチャンネルで使用できる最新バージョンに Operator がアップグレードされます。 [11 ページ](#)の『[IBM MQ Operator のバージョン・サポート](#)』を参照してください。


Red Hat OpenShiftCLI を使用して IBM MQ Operator をアップグレードする

IBM MQ Operator はコマンド・ラインからアップグレードできます。

始める前に

注: 最新の CD バージョンの IBM MQ Operator は 3.1.3 です。最新の LTS バージョンの IBM MQ Operator は 2.0.23 です。最新の IBM MQ Operator リリース・ノートについては、[35 ページ](#)の『[IBM MQ Operator のリリース履歴](#)』を参照してください。

oc login を使用してクラスターにログインします。

エア・ギャップ環境で IBM MQ Operator をアップグレードする前に、最新の IBM Cloud Pak for Integration イメージをミラーリングする必要があります。IBM MQ Operator 3.0 以上にアップグレードする場合、[IBM MQ Operator の現行 CD チャンネルへのマイグレーション](#)には、エア・ギャップ固有のステップが含まれます。以前の IBM MQ オペレーター・バージョンへのアップグレードについては、 **Deprecated** エア・ギャップ環境での最新の IBM MQ 2.x オペレーターまたはキュー・マネージャーへのアップグレードの準備を参照してください。

手順

1. [11 ページ](#)の『[IBM MQ Operator のバージョン・サポート](#)』を確認して、どのオペレーター・チャンネルにアップグレードすべきかを判別します。
2. 最新のカタログ・ソースを適用します。

`ibm-operator-catalog`ではなく、IBM MQ 固有のカタログ・ソース (すべてのエア・ギャップ・インストール) を使用する場合は、IBM MQ バージョンのカタログ・ソースを適用する必要があります。
[クラスターへのカタログ・ソースの追加の手順](#)に従います。

注: エア・ギャップ「[イメージのミラーリング \(エア・ギャップのみ\) \(Mirror images \(air-gap only\)\)](#)」のオペレーター・インストール・ステップを既に完了している場合は、カタログ・ソースを適用するステップを実行するだけで済みます。以下に例を示します。

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

3. IBM MQ Operator をアップグレードします。IBM MQ Operator の新規のメジャー/マイナーのバージョンは、新しいサブスクリプション・チャンネルを介して配信されます。オペレーターを新規メジャー/マイナー・バージョンにアップグレードするには、IBM MQ Operator サブスクリプションで、選択したチャンネルを更新する必要があります。

a) 必要な IBM MQ Operator アップグレード・チャンネルが使用可能であることを確認します。

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

b) Subscription にパッチを適用して目的の更新チャンネルに移動します (vX.Y は前のステップで識別された目的の更新チャンネルです)。

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

Red Hat OpenShift を使用した IBM MQ キュー・マネージャーのアップグレード

始める前に

IBM MQ キュー・マネージャーをアップグレードするプロセスの一環として、IBM Cloud Pak for Integration の資料からこのトピックに移動した可能性があります。

手順

Red Hat OpenShift を使用して IBM MQ キュー・マネージャーをアップグレードするには、以下のいずれかのタスクを実行します。

- [132 ページの『Red Hat OpenShift ウェブコンソールを使用した IBM MQ キュー・マネージャーのアップグレード』](#)
- [133 ページの『Red Hat OpenShift CLI を使用した IBM MQ キュー・マネージャーのアップグレード』](#)
- [134 ページの『プラットフォーム UI を使用した Red Hat OpenShift での IBM MQ キュー・マネージャーのアップグレード』](#)

次のタスク

IBM Cloud Pak for Integration のアップグレードを完了するには、IBM Cloud Pak for Integration の資料に戻らなければならない場合があります。

Red Hat OpenShift ウェブコンソールを使用した IBM MQ キュー・マネージャーのアップグレード

IBM MQ Operator を使用してデプロイされた IBM MQ キュー・マネージャーは、オペレーター・ハブを使用して Red Hat OpenShift でアップグレードすることができます。

始める前に

注: 最新の CD バージョンの IBM MQ キュー・マネージャーは、9.3.5.1-r2 です。最新の LTS バージョンの IBM MQ キュー・マネージャーは、9.3.0.17-r3 です。最新の IBM MQ キュー・マネージャー・リリース・ノートについては、[58 ページの『IBM MQ Operator で使用するキュー・マネージャー・コンテナ・イメージのリリース履歴』](#)を参照してください。

- Red Hat OpenShift クラスターのウェブコンソールにログインします。

- IBM MQ Operator で対象の更新チャンネルを使用していることを確認してください。 [130 ページの『Red Hat OpenShift を使用した IBM MQ Operator のアップグレード』](#)を参照してください。

エア・ギャップ環境でキュー・マネージャーをアップグレードする前に、最新の IBM Cloud Pak for Integration イメージをミラーリングする必要があります。 IBM MQ Operator 3.0 以上にアップグレードする場合、 IBM MQ Operator の現行 CD チャンネルへの [マイグレーション](#) には、エア・ギャップ固有のステップが含まれます。以前の IBM MQ オペレーター・バージョンへのアップグレードについては、

Deprecated エア・ギャップ環境での最新の IBM MQ 2.x オペレーターまたはキュー・マネージャーへのアップグレードの準備を参照してください。

手順

1. ナビゲーション・ペインで、「オペレーター (Operators)」 > 「インストール済みのオペレーター (Installed Operators)」をクリックします。
指定したプロジェクトにインストールされているすべてのオペレーターが表示されます。
2. 「IBM MQ Operator」を選択します。
「IBM MQ Operator」ウィンドウが表示されます。
3. 「キュー・マネージャー」タブにナビゲートします。
「キュー・マネージャーの詳細」ウィンドウが表示されます。
4. アップグレードするキュー・マネージャーを選択します。
5. YAML タブに移動します。
6. 以下のフィールドを更新します。必要に応じて、対象の IBM MQ キュー・マネージャーのバージョン・アップグレードに合わせてください。

- spec.version
- spec.license.licence

IBM MQ Operator バージョンと IBM MQ キュー・マネージャー・コンテナ・イメージのマッピングについては、 [58 ページの『IBM MQ Operator で使用するキュー・マネージャー・コンテナ・イメージのリリース履歴』](#)を参照してください。

7. 更新したキュー・マネージャー YAML を保存します。

Red Hat OpenShift CLI を使用した IBM MQ キュー・マネージャーのアップグレード

IBM MQ Operator を使用してデプロイされた IBM MQ キュー・マネージャーは、コマンド行を使用して Red Hat OpenShift にアップグレードすることができます。

始める前に

注: 最新の CD バージョンの IBM MQ キュー・マネージャーは、9.3.5.1-r2 です。最新の LTS バージョンの IBM MQ キュー・マネージャーは、9.3.0.17-r3 です。最新の IBM MQ キュー・マネージャー・リリース・ノートについては、 [58 ページの『IBM MQ Operator で使用するキュー・マネージャー・コンテナ・イメージのリリース履歴』](#)を参照してください。

以下の手順は、クラスター管理者でないと実行できません。

- oc login を使用して、Red Hat OpenShift コマンド・ライン・インターフェース (CLI) にログインします。
- IBM MQ Operator で対象の更新チャンネルを使用していることを確認してください。 [121 ページの『IBM MQ Operator とキュー・マネージャーのアップグレード』](#)を参照してください。

エア・ギャップ環境でキュー・マネージャーをアップグレードする前に、最新の IBM Cloud Pak for Integration イメージをミラーリングする必要があります。 IBM MQ Operator 3.0 以上にアップグレードする場合、 IBM MQ Operator の現行 CD チャンネルへの [マイグレーション](#) には、エア・ギャップ固有のステップが含まれます。以前の IBM MQ オペレーター・バージョンへのアップグレードについては、

Deprecated エア・ギャップ環境での最新の IBM MQ 2.x オペレーターまたはキュー・マネージャーへのアップグレードの準備を参照してください。

手順

QueueManager リソースを編集して以下のフィールドを更新します。必要に応じて、対象の IBM MQ キュー・マネージャーのバージョン・アップグレードに合わせてください。

- spec.version
- spec.license.licence

チャンネルから IBM MQ Operator バージョンおよび IBM MQ キュー・マネージャー・バージョンへのマッピングについては、[11 ページの『IBM MQ Operator のバージョン・サポート』](#)を参照してください。

以下のコマンドを使用します。

```
oc edit queuemanager my_qmgr
```

ここで、*my_qmgr* は、アップグレードする QueueManager リソースの名前です。

CP4I プラットフォーム UI を使用した Red Hat OpenShift での IBM MQ キュー・マネージャーのアップグレード

IBM MQ Operator を使用してデプロイされた IBM MQ キュー・マネージャーは、IBM Cloud Pak for Integration Platform UI (previously the Platform Navigator) を使用して Red Hat OpenShift でアップグレードすることができます。

始める前に

注:最新の CD バージョンの IBM MQ キュー・マネージャーは、9.3.5.1-r2 です。最新の LTS バージョンの IBM MQ キュー・マネージャーは、9.3.0.17-r3 です。最新の IBM MQ キュー・マネージャー・リリース・ノートについては、[58 ページの『IBM MQ Operator で使用するキュー・マネージャー・コンテナ・イメージのリリース履歴』](#)を参照してください。

- アップグレードするキュー・マネージャーが含まれている名前空間内の IBM Cloud Pak for Integration Platform UI にログインします。
- IBM MQ Operator で対象の更新チャンネルを使用していることを確認してください。[121 ページの『IBM MQ Operator とキュー・マネージャーのアップグレード』](#)を参照してください。

エア・ギャップ環境でキュー・マネージャーをアップグレードする前に、最新の IBM Cloud Pak for Integration イメージをミラーリングする必要があります。IBM MQ Operator 3.0 以上にアップグレードする場合、[IBM MQ Operator の現行 CD チャンネルへのマイグレーション](#)には、エア・ギャップ固有のステップが含まれます。以前の IBM MQ オペレーター・バージョンへのアップグレードについては、

Deprecated エア・ギャップ環境での最新の IBM MQ 2.x オペレーターまたはキュー・マネージャーへのアップグレードの準備を参照してください。

手順

1. IBM Cloud Pak for Integration Platform UI (previously the Platform Navigator) ホーム・ページで「**ランタイム (Runtimes)**」タブをクリックします。
2. アップグレードが可能なキュー・マネージャーには、「バージョン」の隣に青色の **i** マークが付いています。**i** マークをクリックすると、「**利用できる新しいバージョン (New version available)**」が表示されます。
3. アップグレードするキュー・マネージャーの右端にある 3 点メニューをクリックして、「**バージョンの変更 (Change version)**」をクリックします。
4. 「**新しいチャンネルまたはバージョンの選択 (Select a new channel or version)**」の下で、必要なアップグレード・バージョンを選択します。
5. 「**バージョンの変更 (Change version)**」をクリックします。

タスクの結果

キュー・マネージャーがアップグレードされます。

OpenShift IBM MQ Operator を使用したキュー・マネージャーの構成

構成例: HA の構成、OpenShift クラスターの外部からの接続、CP4i ダッシュボードとの統合、Instana トレースとの統合、カスタム MQSC ファイルおよび INI ファイルを使用したイメージの作成、カスタム・アノテーションおよびラベルの追加。

このタスクについて

手順

- [135 ページの『キュー・マネージャーの構成例』](#)。
- [144 ページの『IBM MQ Operator を使用したキュー・マネージャーの高可用性の構成』](#)。
- [155 ページの『Red Hat OpenShift クラスターの外部からキュー・マネージャーに接続するためのルート構成』](#)。
- [157 ページの『IBM Cloud Pak for Integration Operations Dashboard との統合』](#)。
- [158 ページの『IBM MQ と IBM Instana トレースの統合』](#)。
- [166 ページの『Red Hat OpenShift CLI を使用した、カスタム MQSC および INI ファイルを使用したイメージの作成』](#)。
- [167 ページの『キュー・マネージャー・リソースへのカスタム・アノテーションとカスタム・ラベルの追加』](#)。
- [168 ページの『実行時 Webhook チェックの無効化』](#)。
- [169 ページの『キュー・マネージャー仕様に対するデフォルト値更新の無効化』](#)。

OpenShift CP4I キュー・マネージャーの構成例

QueueManager カスタム・リソースの内容を調整することによってキュー・マネージャーを構成できます。

このタスクについて

QueueManager YAML ファイルを使用してキュー・マネージャーを構成するのに役立つ例を以下に挙げます。

手順

- [135 ページの『例: MQSC ファイルと INI ファイルの提供』](#)
- [139 ページの『例: 相互 TLS 認証を使用したキュー・マネージャーの構成』](#)

OpenShift CP4I 例: MQSC ファイルと INI ファイルの提供

この例では、2つの MQSC ファイルと 1つの INI ファイルを組み込んだ Kubernetes ConfigMap を作成します。その後、それらの MQSC ファイルと INI ファイルを処理するキュー・マネージャーをデプロイします。

このタスクについて

MQSC ファイルと INI ファイルは、キュー・マネージャーのデプロイ時に提供できます。MQSC と INI のデータは、1つ以上の Kubernetes ConfigMap とシークレットで定義する必要があります。キュー・マネージャーをデプロイする名前空間(プロジェクト)内にそれらを作成してください。

注: MQSC ファイルや INI ファイルに機密データを組み込む場合は、Kubernetes シークレットを使用してください。

例

2つのMQSCファイルと1つのINIファイルを組み込んだKubernetes ConfigMapを作成する例を以下に示します。その後、それらのMQSCファイルとINIファイルを処理するキュー・マネージャーをデプロイします。

ConfigMapの例 - 以下のYAMLをクラスターに適用します。

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

QueueManagerの例-コマンド行またはRed Hat OpenShift Container Platform Web コンソールを使用して、以下の構成でキュー・マネージャーをデプロイします。

```
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: IBM MQ Advancedのご使用条件に同意する場合は、`accept: false`を`accept: true`に変更します。ライセンスの詳細については、mq.ibm.com/v1beta1のライセンス交付に関する参照資料を参照してください。

追加情報:

- キュー・マネージャーは、単一のKubernetes ConfigMapまたはシークレット(この例を参照)を使用するように構成することも、複数のConfigMapsおよびシークレットを使用するように構成することもできます。
- この例のように1つのKubernetes ConfigMapまたはシークレットからMQSCとINIのすべてのデータを使用することも、キュー・マネージャーごとに使用可能なファイルのサブセットだけを使用するように構成することも可能です。
- MQSCファイルとINIファイルは、それぞれのキーに基づいてアルファベット順に処理されます。したがって、`example1.mqsc`は、キュー・マネージャー構成での表示順序に関係なく、常に`example2.mqsc`の前に処理されます。

- 複数の Kubernetes ConfigMap またはシークレットにまたがる複数の MQSC ファイルや INI ファイルが同じキーを持っている場合、その一連のファイルはそれぞれがキュー・マネージャー構成で定義されている順序に基づいて処理されます。
- キュー・マネージャー・ポッドの実行中は、Kubernetes ConfigMap に対する変更は反映されません。これは、IBM MQ Operator が変更を認識しないためです。ConfigMap に変更を加えた場合 (例えば、MQSC コマンドまたは INI ファイルに変更を加えた場合)、それらの変更を有効にするには、キュー・マネージャーを手動で再始動する必要があります。単一インスタンス・キュー・マネージャーの場合は、ポッドを削除して、必要な再始動をトリガーします。ネイティブ HA デプロイメントの場合は、まずスタンバイ・ポッドを削除して再始動します。再度実行状態になったら、アクティブ・ポッドを削除して再始動します。この再始動の順序により、キュー・マネージャーのダウン時間を最小限に抑えることができます。

OpenShift CP4I OpenSSL を使用した自己署名 PKI の作成

IBM MQ では、認証に相互 TLS を使用できます。この場合、接続の両端で証明書が提供され、証明書の詳細を使用してキュー・マネージャーとの ID が確立されます。このトピックでは、OpenSSL コマンド行ツールを使用して Public Key Infrastructure (PKI) の例を作成し、他の例で使用できる 2 つの証明書を作成する方法について説明します。

始める前に

OpenSSL コマンド・ライン・ツールがインストールされていることを確認します。

IBM MQ client をインストールし、samp/bin と bin をパスに追加します。runmqicred コマンドが必要です。このコマンドは、以下のように IBM MQ client の一部としてインストールできます。

- **Windows** **Linux** Windows および Linux の場合: ご使用のオペレーティング・システム用の IBM MQ 再配布可能クライアントを <https://ibm.biz/mq93redistclients> からインストールします。
- **macOS** Mac の場合: IBM MQ MacOS Toolkit をダウンロードしてセットアップします。 <https://developer.ibm.com/tutorials/mq-macos-dev/>

このタスクについて

重要: ここで説明する例は、実稼働環境には適しておらず、すぐに使用できるようにするための例としてのみ意図されています。証明書管理は、上級者向けの複雑なサブジェクトです。実動環境では、ローテーション、取り消し、鍵の長さ、災害復旧などを考慮する必要があります。

これらのステップは、OpenSSL 3.1.4 を使用してテストされています。

手順

1. 内部認証局に使用する秘密鍵を作成します

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

内部認証局の秘密鍵は、ca.key というファイルに作成されます。このファイルは、安全に秘密にしておく必要があります。内部認証局の証明書に署名するために使用されます。

2. 内部認証局の自己署名証明書の発行

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

-days は、ルート CA 証明書が有効である日数を指定します。

ca.crt というファイルに証明書が作成されます。この証明書には、内部認証局に関する公開情報が含まれており、自由に共有できます。

3. キュー・マネージャーの秘密鍵と証明書の作成

- a) キュー・マネージャーの秘密鍵および証明書署名要求の作成

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

example-qm.key という名前のファイルに秘密鍵が作成され、*example-qm.csr* という名前のファイルに証明書署名要求が作成されます。

- b) 内部認証局を使用してキュー・マネージャー鍵に署名します。

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

`-days` は、証明書が有効になる日数を指定します。

example-qm.crt というファイルに署名証明書が作成されます。

- c) キュー・マネージャーの鍵と証明書を使用して Kubernetes 秘密を作成します。

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

example-qm-tls という Kubernetes シークレットが作成されます。この秘密には、キュー・マネージャーの秘密鍵、公開証明書、および CA 証明書が含まれています。

4. アプリケーションの秘密鍵と証明書の作成

- a) アプリケーションの秘密鍵および証明書署名要求の作成

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

example-app1.key というファイルに秘密鍵が作成され、*example-app1.csr* というファイルに証明書署名要求が作成されます。

- b) 内部認証局を使用してキュー・マネージャー鍵に署名します。

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

`-days` は、証明書が有効になる日数を指定します。

署名付き証明書が *example-app1.crt* というファイルに作成されます。

- c) アプリケーションの鍵と証明書を使用して PKCS#12 鍵ストアを作成します。

IBM MQ は、個々の鍵ファイルではなく、鍵データベースを使用します。コンテナ化されたキュー・マネージャーは、シークレットからキュー・マネージャーの鍵データベースを作成しますが、クライアント・アプリケーションの場合は、鍵データベースを手動で作成する必要があります。

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

`<PASSWORD>` は、任意のパスワードに置き換えてください。

example-app1.p12 という名前のファイルに鍵ストアが作成されます。アプリケーションの鍵と証明書は、CA 証明書だけでなく、「example-app1」の「label」または「friendly name」という名前で内部に保管されます。

- d) arm64 Apple Mac を使用している場合は、アプリケーション証明書と CA 証明書を組み合わせた追加ファイルを作成する必要があります。

以下に例を示します。

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

関連タスク

[139 ページの『例: 相互 TLS 認証を使用したキュー・マネージャーの構成』](#)

この例では、IBM MQ Operator を使用して、キュー・マネージャーを OpenShift Container Platform にデプロイします。相互 TLS は、認証に使用され、TLS 証明書からキュー・マネージャー内の ID にマップされます。

[147 ページの『例: IBM MQ Operator を使用したネイティブ HA の構成』](#)

この例では、IBM MQ Operator を使用して、ネイティブ高可用性フィーチャーを使用するキュー・マネージャーを OpenShift Container Platform にデプロイします。相互 TLS は、認証に使用され、TLS 証明書からキュー・マネージャー内の ID にマップされます。

153 ページの『IBM MQ Operator を使用した複数インスタンス・キュー・マネージャーの構成』

この例では、IBM MQ Operator を使用して、複数インスタンス・キュー・マネージャーを OpenShift Container Platform にデプロイします。相互 TLS は、認証に使用され、TLS 証明書からキュー・マネージャー内の ID にマップされます。

OpenShift CP4I Linux 例: 相互 TLS 認証を使用したキュー・マネージャーの構成

この例では、IBM MQ Operator を使用して、キュー・マネージャーを OpenShift Container Platform にデプロイします。相互 TLS は、認証に使用され、TLS 証明書からキュー・マネージャー内の ID にマップされます。

始める前に

この例を完了するには、まず以下の前提条件を満たしておく必要があります。

- この例のための OpenShift Container Platform (OCP) プロジェクト / 名前空間を作成します。
- コマンド・ラインで OCP クラスターにログインし、上記の名前空間に切り替えます。
- 上記の名前空間に IBM MQ Operator がインストールされ、使用可能な状態になっていることを確認します。

このタスクについて

この例では、OpenShift Container Platform にデプロイするキュー・マネージャーを定義したカスタム・リソース YAML を提供しています。また、TLS を有効にしてキュー・マネージャーをデプロイするために必要な追加のステップも詳しく説明しています。

手順

1. [137 ページの『OpenSSL を使用した自己署名 PKI の作成』](#)の説明に従って、証明書のペアを作成します。
2. MQSC コマンドと INI ファイルを含む構成マップを作成します。

MQSC コマンドを含む Kubernetes ConfigMap を作成して、新しいキューと SVRCONN チャネルを作成し、チャネルへのアクセスを許可するチャネル認証レコードを追加します。

前に作成した名前空間 ([始める前](#)に参照) にいることを確認してから、OCP Web コンソールで、またはコマンド・ラインを使用して、以下の YAML を入力します。

```
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 は、*MTLS.SVRCONN* というチャンネルと、*EXAMPLE.QUEUE*。チャンネルは、「共通名」が *example-app1* の証明書を提示するクライアントにのみアクセスを許可するように構成されています。これは、ステップ [139 ページの『1』](#) で作成した証明書のいずれかで使用される共通名です。この共通名を持つこのチャンネル上の接続は、*app1* というユーザー ID にマップされます。このユーザー ID は、キュー・マネージャーに接続し、サンプル・キューにアクセスすることを許可されています。INI ファイルはセキュリティ・ポリシーを使用可能にします。これは、*app1* ユーザー ID が外部ユーザー・レジストリーに存在する必要がないことを意味します。この ID は、この構成に名前としてのみ存在します。

3. キュー・マネージャーのデプロイ

以下のカスタム・リソース YAML を使用して、新しいキュー・マネージャーを作成します。このタスクを開始する前に作成した名前空間にいることを確認してから、OCP Web コンソールで、またはコマンド・ラインを使用して、以下の YAML を入力します。正しいライセンスが指定されていることを確認し、`false` を `true` に変更してライセンスを受け入れます。

```
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
```

シークレット *example-qm-tls* がステップ [139 ページの『1』](#) で作成され、ConfigMap *example-tls-configmap* がステップ [139 ページの『2』](#) で作成されたことに注意してください。

4. キュー・マネージャーが稼働していることの確認

キュー・マネージャーがデプロイされます。続行する前に、Running 状態であることを確認してください。以下に例を示します。

```
oc get qmgr exampleqm
```

5. キュー・マネージャーへの接続のテスト

キュー・マネージャーが相互 TLS 通信に構成されていることを確認するには、[141 ページの『ラップトップからキュー・マネージャーへの相互 TLS 接続のテスト』](#) の手順に従います。

タスクの結果

これで、TLS が有効になっているキュー・マネージャーが正常にデプロイされ、TLS 証明書で提供された詳細を使用してキュー・マネージャーで認証を行い、ID を指定することができました。

接続のテスト

IBM MQ Operator を使用してキュー・マネージャーを作成した後、そのキュー・マネージャーに接続し、メッセージの書き込みと読み取りを行うことによって、そのキュー・マネージャーが機能しているかどうかをテストできます。このタスクでは、ラップトップなどの Kubernetes クラスター外部のマシン上で IBM MQ サンプル・プログラムを実行することにより、それらのプログラムを使用して接続する方法について説明します。

始める前に

この例を完了するには、まず以下の前提条件を満たしておく必要があります。

- IBM MQ client のインストール `amqsputc` コマンドと `amqsgetc` コマンドが必要です。これらのコマンドは、以下のように IBM MQ client の一部としてインストールできます。
 - **Windows** **Linux** Windows および Linux の場合: ご使用のオペレーティング・システム用の IBM MQ 再配布可能クライアントを <https://ibm.biz/mq93redistclients> からインストールします。
 - **macOS** Mac の場合: IBM MQ MacOS Toolkit をダウンロードしてセットアップします。 <https://developer.ibm.com/tutorials/mq-macos-dev/>
- 必要な鍵ファイルと証明書ファイルがマシン上のディレクトリーにダウンロードされていること、および鍵ストアのパスワードがわかっていることを確認してください。例えば、以下のファイルが [137 ページ](#) の『[OpenSSL を使用した自己署名 PKI の作成](#)』に作成されます。
 - `example-app1.p12`
 - `example-app1-chain.crt` (arm64 Apple Mac を使用している場合のみ)
- TLS で構成されたキュー・マネージャーを OCP クラスターにデプロイします。例えば、[139 ページ](#) の『[例: 相互 TLS 認証を使用したキュー・マネージャーの構成](#)』の手順に従います。

このタスクについて

この例では、ラップトップなどの Kubernetes クラスター外部のマシン上で実行されている IBM MQ サンプル・プログラムを使用して、TLS で構成された QueueManager に接続し、メッセージの書き込みと取得を行います。

手順

1. キュー・マネージャーが稼働していることの確認

キュー・マネージャーがデプロイされます。続行する前に、Running 状態であることを確認してください。以下に例を示します。

```
oc get qmgr exampleqm
```

2. キュー・マネージャーのホスト名の検索

自動的に作成される経路を使用して、OCP クラスターの外部からキュー・マネージャーの完全修飾ホスト名を検索するには、次のコマンドを使用します。 `exampleqm-ibm-mq-qm`

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

3. IBM MQ クライアント・チャネル定義テーブル (CCDT) の作成

以下の内容を含む `ccdt.json` というファイルを作成します。

```
{
  "channel": [
    {
      "name": "MTLS.SVRCONN",
      "clientConnection": {
        "connection":
```

```

    [
      {
        "host": "<hostname from previous step>",
        "port": 443
      }
    ],
    "queueManager": "EXAMPLEQM"
  },
  "transmissionSecurity":
  {
    "cipherSpecification": "ANY_TLS13",
    "certificateLabel": "example-app1"
  },
  "type": "clientConnection"
}
]
}

```

この接続では、Red Hat OpenShift Container Platform ルーターが listen しているポートであるため、ポート 443 が使用されます。トラフィックは、ポート 1414 でキュー・マネージャーに転送されます。

別のチャンネル名を使用した場合は、それも調整する必要があります。相互 TLS の例では、`MTLS.SVRCONN` という名前のチャンネルを使用します。

詳しくは、[JSON 形式の CCDT の構成](#) を参照してください。

4. 接続の詳細を構成するためのクライアント INI ファイルの作成

現行ディレクトリーに `mqclient.ini` という名前のファイルを作成します。このファイルは、**amqsputc** および **amqsgetc** によって読み取られます。

```

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

```

必ず、`SSLKeyRepository` 「パスワード」を、PKCS#12 ファイルの作成時に選択したパスワードに更新してください。暗号化されたパスワードを使用するなど、鍵ストアのパスワードを設定する方法は他にもあります。詳しくは、[AIX, Linux, and Windows での IBM MQ MQI client の鍵リポジトリー・パスワードの提供](#) を参照してください。

Red Hat OpenShift Container Platform Router では、IBM MQ キュー・マネージャーへの要求のルーティングに SNI が使用されます。`OutboundSNI=HOSTNAME` 属性は、ルーターが IBM MQ Operator によって構成されたデフォルト経路を処理するために必要な情報が IBM MQ クライアントに確実に含まれるようにします。詳しくは、[155 ページの『Red Hat OpenShift クラスターの外部からキュー・マネージャーに接続するためのルートの構成』](#) を参照してください。

5. arm64 Apple Mac を使用している場合は、追加の環境変数を構成する必要があります。

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

このファイルには、アプリケーションおよび CA 証明書を含む完全な証明書チェーンが含まれています。

6. キューへのメッセージの書き込み

以下のコマンドを実行します。

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

キュー・マネージャーへの接続が成功すると、以下の応答が出力されます。

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

任意のテキストを入力してから **Enter** を押す操作を何回か繰り返すことで、キューに複数のメッセージを書き込みます。

書き込みを終了するには、**Enter** を 2 回押します。

7. キューからのメッセージの取得

以下のコマンドを実行します。

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

前のステップで追加したメッセージがコンシュームされ、出力されます。数秒後にコマンドが終了します。

タスクの結果

これで、TLS が有効になっているキュー・マネージャーの接続のテストが正常に完了し、クライアントからキュー・マネージャーにメッセージを安全に書き込んだり、取得したりできることが示されました。

OpenShift CP4I 例：ライセンス・サービスの注釈のカスタマイズ

IBM MQ Operator は、デプロイ済みリソースに IBM License Service アノテーションを自動的に追加します。これらは IBM License Service によってモニターされ、必要な資格に対応するレポートが生成されます。

このタスクについて

IBM MQ Operator によって追加される注釈は、標準シチュエーションで予期される注釈であり、キュー・マネージャーのデプロイメント中に選択されたライセンス値に基づいています。

例

License が L-RJON-BZFQU2 (IBM Cloud Pak for Integration 2021.2.1) に設定されていて、**Use** が NonProduction に設定されている場合は、次のアノテーションが適用されます：

- cloudpakId: c8b82d189e7545f0892db9ef2731b90d
- cloudpakName: IBM Cloud Pak for Integration
- productChargedContainers: qmgr
- productCloudpakRatio: '4:1'
- productID: 21dfe9a0f00f444f888756d835334909
- 実動用の製品名: IBM MQ Advanced 非実動用
- 製品メトリック：VIRTUAL_PROCESSOR_CORE
- productVersion: 9.2.3.0

IBM Cloud Pak for Integration 内で、IBM App Connect Enterprise のデプロイメントには、IBM MQ に対する制限付き使用権が含まれます。このような状況では、これらのアノテーションをオーバーライドして、IBM License Service が正しい使用法をキャプチャーすることを確認する必要があります。そのためには、[167 ページの『キュー・マネージャー・リソースへのカスタム・アノテーションとカスタム・ラベルの追加』](#)で説明されている方法を使用してください。

例えば、IBM MQ が IBM App Connect Enterprise ライセンスの下にデプロイされている場合は、以下のコード・フラグメントに示されている方法を使用してください。

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

ライセンスアノテーションの変更が必要となる理由は、他に 2 つあります。

1. IBM MQ Advanced は、別の IBM 製品の使用権に含まれています。
 - この状態では、IBM App Connect Enterprise について前に説明したアプローチを使用してください。
2. IBM MQ は IBM Cloud Pak for Integration ライセンスの下にデプロイされます。

- IBM Cloud Pak for Integration ライセンスがある場合は、IBM MQ または IBM MQ Advanced の比率の下にキュー・マネージャーをデプロイすることを決定できます。IBM MQ の比率の下にデプロイする場合は、ネイティブ HA や Advanced Message Security などの高度な機能を使用しないようにする必要があります。
- このような場合は、以下のアノテーションを使用して制作してください。

```
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
```

- 非実動使用には以下の注釈を使用します。

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: mq4ace
  namespace: cp4i
spec:
  annotations:
    productID: 151bec68564a4a47a14e6fa99266def
    productCloudpakRatio: '8:1'
    productName: IBM MQ for Non-Production
    productMetric: VIRTUAL_PROCESSOR_CORE
```

OpenShift MQ Adv. IBM MQ Operator を使用したキュー・マネージャーの高可用性の構成

このタスクについて

手順

- [144 ページの『ネイティブ HA』](#)。
- [147 ページの『例: IBM MQ Operator を使用したネイティブ HA の構成』](#)。
- [153 ページの『IBM MQ Operator を使用した複数インスタンス・キュー・マネージャーの構成』](#)。

OpenShift MQ Adv. ネイティブ HA

ネイティブ HA は、IBM MQ 用のネイティブ (組み込み) 高可用性ソリューションであり、クラウド・ブロック・ストレージで使用するときに適しています。

ネイティブ HA 構成によって、高可用性キュー・マネージャーを実行できます。このキュー・マネージャーは、リカバリー可能な MQ データ (メッセージなど) のレプリケーションをストレージの複数セットにまたがって行うことにより、ストレージの障害によるデータの損失を防ぎます。このキュー・マネージャーには実行中のインスタンスが複数あり、そのうちの 1 つがリーダーになります。他のキュー・マネージャーは、障害発生時にすぐにテークオーバーできるように準備を整えているので、キュー・マネージャーとメッセージへのアクセスが最大化されます。

ネイティブ HA 構成は、3 つの Kubernetes ポッドから成り、その個々にキュー・マネージャーのインスタンスがあります。1 つのインスタンスがアクティブ・キュー・マネージャーとして機能し、メッセージはそこで処理されてリカバリー・ログに書き込まれます。そのリカバリー・ログへの書き込みが行われると、アクティブ・キュー・マネージャーはレプリカと呼ばれる他の 2 つのインスタンスにデータを送信します。各レプリカは、個別に所有するリカバリー・ログへの書き込みを行い、データを認知し、複製されたりリカバリー・ログからキュー・データをそれぞれ更新します。アクティブ・キュー・マネージャーを実行しているポッドに障害が発生すると、キュー・マネージャーのレプリカ・インスタンスの 1 つがアクティブの役割を引き継ぎ、このインスタンスにある最新データで運用が行われます。

ログ・タイプは「複製ログ」と呼ばれます。複製されたログは基本的にリニア・ログであり、自動ログ管理と自動メディア・イメージが有効になっています。ログのタイプを参照してください。リニア・ログの管理に使用するものと同じ手法を使用して、複製されたログを管理します。

ネットワーク・トラフィックを扱う準備ができていて唯一のポッドであると識別された現行アクティブ・インスタンスに TCP/IP クライアント接続をルーティングするために、Kubernetes サービスが使用されます。この処理を行うために、クライアント・アプリケーションがさまざまなインスタンスを認識しておく必要はありません。

3つのポッドを使用すると、スプリット・ブレンという状態になる可能性が大幅に減少します。2ポッドの高可用性システムでは、2つのポッド間の接続が切断されるとスプリット・ブレンが発生する可能性があります。接続のない状態だと、両方のポッドでキュー・マネージャーが同時に実行され、別々のデータが累積される可能性があります。そうすると、接続が復元された時に2つの別々のバージョンのデータ（「スプリット・ブレン」）が存在することになるので、保持するデータ・セットと破棄するデータ・セットを決定するために手操作による介入が必要になります。

ネイティブ HA では、スプリット・ブレン状態を回避するために、クォーラムを設定した3ポッド・システムを使用します。1つのポッドが他の1つ以上のポッドと通信できる場合は、その通信可能なポッド同士がクォーラムを形成します。キュー・マネージャーは、クォーラムを形成するポッド上にある場合のみ、アクティブ・インスタンスになることができます。キュー・マネージャーは、他の1つ以上のポッドと接続していないポッド上ではアクティブになることはできないので、2つのアクティブ・インスタンスが同時に存在することは絶対にありません。

- 1つのポッドで障害が発生しても、他の2つのポッドのいずれかにあるキュー・マネージャーが引き継ぐことができます。2つのポッドで障害が発生すると、残りの1つのポッドにあるキュー・マネージャーはアクティブ・インスタンスになることができません。そのポッドはクォーラムを形成していないからです（その残りのポッドは、他の2つのポッドで障害が発生したのか、それとも自分の接続が失われていて他の2つのポッドはまだ実行されているのかの違いを判別できません）。
- 1つのポッドが接続を失うと、そのポッドにあるキュー・マネージャーはアクティブになることができません。そのポッドはクォーラムを形成していないからです。残りの2つのポッドでは、いずれか1つのキュー・マネージャーが引き継ぎを実行できます。クォーラムを形成しているからです。すべてのポッドが接続を失うと、どのポッドのキュー・マネージャーもアクティブになることができません。どのポッドもクォーラムを形成していないからです。

アクティブ・ポッドに障害が発生し、その後復旧すると、レプリカの役割でグループに再参加することができます。

パフォーマンスと信頼性のために、ネイティブ HA 構成では RWO (ReadWriteOnce) 永続ストレージを使用することをお勧めします。すべてのストレージ・プロバイダーからの RWO ポリ्यूームは、以下の条件を満たしている場合にサポートされます。

- ブロック・ストレージ・プロバイダーから取得されます。
- ext4 または XFS としてフォーマットされます（これにより、POSIX 準拠が保証されます）。
- 動的ポリ्यूーム・プロビジョニングと「volumeBinding モード: WaitForFirstConsumer」をサポートします。

以下のプロバイダーは、明示的に禁止されています。

- NFS
- GlusterFS
- その他の非ブロック・プロバイダー。

以下の図に、1つのキュー・マネージャーの3つのインスタンスが3つのコンテナ内にデプロイされている、標準的なデプロイメントを示します。

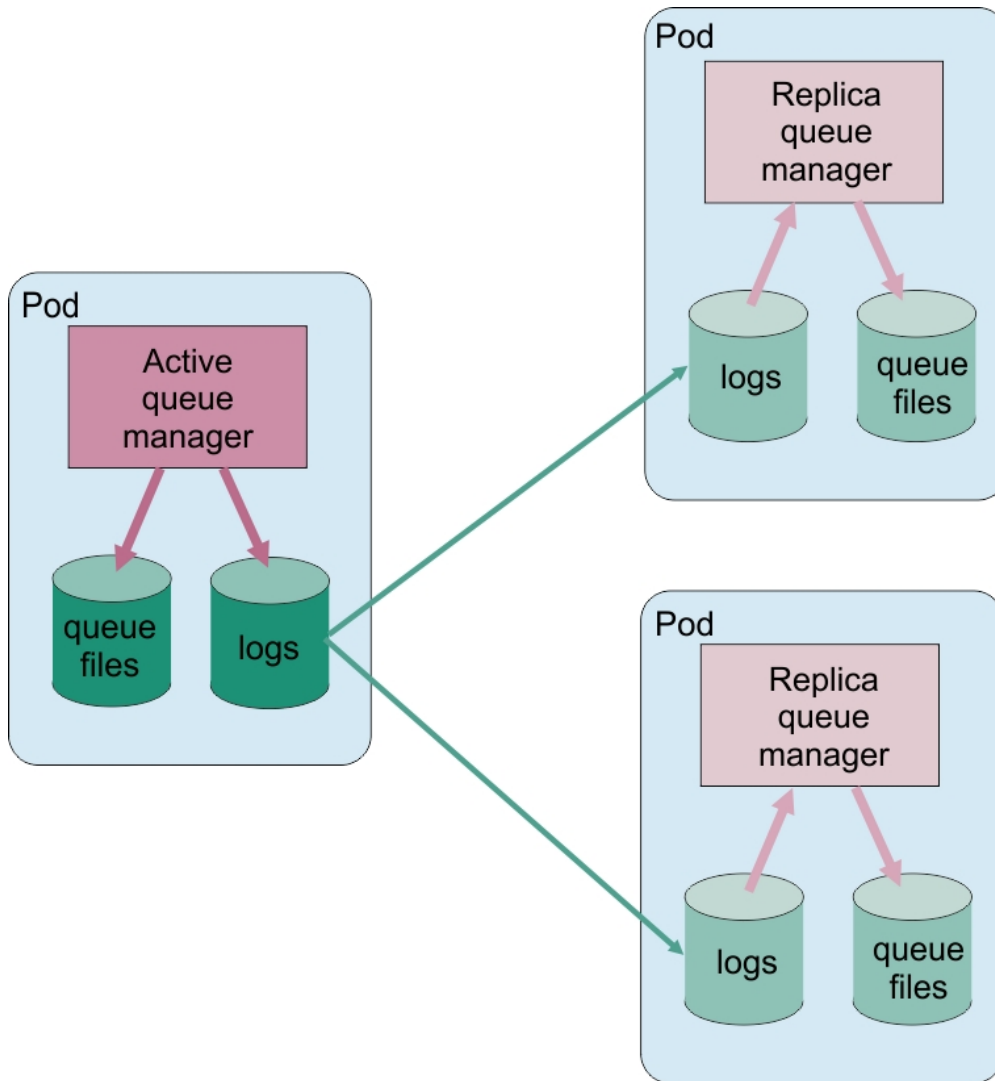


図 1. ネイティブ HA 構成の例

OpenShift MQ Adv. IBM MQ Operator を使用したネイティブ HA の構成
 ネイティブ HA は QueueManager API を使用して構成され、拡張オプションは INI ファイルを使用して利用できます。

ネイティブ HA は、`.spec.queueManager.availability` (QueueManager API) を使用して構成されます。以下に例を示します。

```

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` フィールドは NativeHA に設定する必要があります。

`.spec.queueManager.availability` では、複製時にキュー・マネージャー・インスタンス間で使用する TLS シークレットと TLS 暗号を構成することもできます。この構成を行うことを強くお勧めします。

147 ページの『例: IBM MQ Operator を使用したネイティブ HA の構成』には、ステップバイステップ形式のガイドが用意されています。

関連タスク

147 ページの『例: IBM MQ Operator を使用したネイティブ HA の構成』

この例では、IBM MQ Operator を使用して、ネイティブ高可用性フィーチャーを使用するキュー・マネージャーを OpenShift Container Platform にデプロイします。相互 TLS は、認証に使用され、TLS 証明書からキュー・マネージャー内の ID にマップされます。

 例: IBM MQ Operator を使用したネイティブ HA の構成

この例では、IBM MQ Operator を使用して、ネイティブ高可用性フィーチャーを使用するキュー・マネージャーを OpenShift Container Platform にデプロイします。相互 TLS は、認証に使用され、TLS 証明書からキュー・マネージャー内の ID にマップされます。

始める前に

この例を完了するには、まず以下の前提条件を満たしておく必要があります。

- この例のための OpenShift Container Platform (OCP) プロジェクト / 名前空間を作成します。
- コマンド・ラインで OCP クラスターにログインし、上記の名前空間に切り替えます。
- 上記の名前空間に IBM MQ Operator がインストールされ、使用可能な状態になっていることを確認します。

このタスクについて

この例では、OpenShift Container Platform にデプロイするキュー・マネージャーを定義したカスタム・リソース YAML を提供しています。また、TLS を有効にしてキュー・マネージャーをデプロイするために必要な追加のステップも詳しく説明しています。

手順

1. 137 ページの『OpenSSL を使用した自己署名 PKI の作成』の説明に従って、証明書のペアを作成します。
2. MQSC コマンドと INI ファイルを含む構成マップを作成します。

MQSC コマンドを含む Kubernetes ConfigMap を作成して、新しいキューと SVRCONN チャネルを作成し、チャネルへのアクセスを許可するチャネル認証レコードを追加します。

前に作成した名前空間 (始める前にを参照) にいることを確認してから、OCP Web コンソールで、またはコマンド・ラインを使用して、以下の YAML を入力します。

```
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 は、*MTLS.SVRCONN* というチャンネルと、*EXAMPLE.QUEUE*。チャンネルは、「共通名」が *example-app1* の証明書を提示するクライアントにのみアクセスを許可するように構成されています。これは、ステップ 147 ページの『1』で作成した証明書のいずれかで使用される共通名です。この共通名を持つこのチャンネル上の接続は、*app1* というユーザー ID にマップされます。このユーザー ID は、キュー・マネージャーに接続し、サンプル・キューにアクセスすることを許可されています。INI ファイルはセキュリティ・ポリシーを使用可能にします。これは、*app1* ユーザー ID が外部ユーザー・レジストリーに存在する必要がないことを意味します。この ID は、この構成に名前としてのみ存在します。

3. キュー・マネージャーのデプロイ

以下のカスタム・リソース YAML を使用して、新しいキュー・マネージャーを作成します。このタスクを開始する前に作成した名前空間にいることを確認してから、OCP Web コンソールで、またはコマンド・ラインを使用して、以下の YAML を入力します。正しいライセンスが指定されていることを確認し、*false* を *true* に変更してライセンスを受け入れます。

```
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
```

シークレット *example-qm-tls* がステップ 147 ページの『1』で作成され、ConfigMap *example-nativeha-configmap* がステップ 147 ページの『2』で作成されたことに注意してください。

可用性タイプが *NativeHA* に設定され、永続ストレージが選択されます。Kubernetes クラスターで構成されているデフォルトのストレージ・クラスが使用されます。ストレージ・クラスがデフォルトとして構成されていない場合、または別のストレージ・クラスを使用したい場合は、*defaultClass*: *<storage_class_name>* を *spec.queueManager.storage* の下に追加します。

ネイティブ HA キュー・マネージャー内の 3 つのポッドは、ネットワークを介してデータを複製します。このリンクはデフォルトでは暗号化されませんが、この例ではトラフィックの暗号化にキュー・マネージャーの証明書を使用します。セキュリティを強化するために、別の証明書を指定することができます。ネイティブ HA TLS シークレットは、特定の構造を持つ Kubernetes TLS シークレットでなければなりません (例えば、秘密鍵を *tls.key* という名前にする必要があります)。

4. キュー・マネージャーが稼働していることの確認

キュー・マネージャーがデプロイされます。続行する前に、Running 状態であることを確認してください。以下に例を示します。

```
oc get qmgr exampleqm
```

5. キュー・マネージャーへの接続のテスト

キュー・マネージャーが構成済みで使用可能であることを確認するには、[141 ページの『ラップトップからキュー・マネージャーへの相互 TLS 接続のテスト』](#)の手順に従います。

6. アクティブ・ポッドの障害の強制試行

キュー・マネージャーの自動復旧を検証するには、ポッドの障害をシミュレートします。

a) アクティブ・ポッドとスタンバイ・ポッドの表示

以下のコマンドを実行します。

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

READY フィールドでは、アクティブ・ポッドは値 1/1 を返し、レプリカ・ポッドは値 0/1 を返すことに注意してください。

b) アクティブ・ポッドの削除

アクティブ・ポッドの絶対パス名を指定して次コマンドを実行します。

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

c) ポッドの状況の再表示

以下のコマンドを実行します。

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

d) キュー・マネージャーの状況の表示

他のいずれかのポッドの絶対パス名を指定して次のコマンドを実行します。

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

アクティブ・インスタンスが変更されたことを示す状況が表示されるはずですが、以下に例を示します。

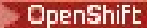

```
QMNAME(EXAMPLEQM) ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

e) キュー・マネージャーへの接続を再度テストします。

キュー・マネージャーが回復したことを確認するには、[141 ページの『ラップトップからキュー・マネージャーへの相互 TLS 接続のテスト』](#)の手順に従います。

タスクの結果

これで、ネイティブの高可用性と相互 TLS 認証を使用してキュー・マネージャーが正常にデプロイされ、アクティブなポッドで障害が発生したときに自動的に復旧することが確認されました。

  IBM MQ コンテナのネイティブ HA キュー・マネージャーの状況の表示
IBM MQ コンテナの場合、実行中のいずれかのポッド内で **dspmq** コマンドを実行することで、ネイティブ HA インスタンスの状況を表示できます。

このタスクについて

実行中のポッドのうち1つで **dspm** コマンドを使用すると、キュー・マネージャー・インスタンスの運用状況を表示できます。返される情報は、インスタンスがアクティブとレプリカのどちらであるかに応じて異なります。アクティブ・インスタンスで提供される情報が確定的なもので、レプリカ・ノードからの情報は古くなっている可能性があります。

以下のアクションを実行できます。

- 現行ノード上のキュー・マネージャー・インスタンスがアクティブかレプリカを表示します。
- 現行ノード上のインスタンスのネイティブ HA の運用状況を表示します。
- ネイティブ HA 構成に属する3つのインスタンスすべての運用状況を表示します。

以下の状況フィールドが、ネイティブ HA 構成状況の報告に使用されます。

ROLE

これは、現行インスタンス・ロールを指定します。これは、Active、Replica、またはUnknownのいずれかです。

INSTANCE

このキュー・マネージャー・インスタンスの作成時に **crtmqm** コマンドの **-lr** オプションを使用してこのキュー・マネージャー・インスタンスに対して指定された名前。

INSYNC

必要な場合にインスタンスがアクティブ・インスタンスとしてテークオーバーできるかどうかを示します。

QUORUM

クォラムの状況を *number_of_instances_in-sync/number_of_instances_configured* という形式でレポートします。

REPLADDR

キュー・マネージャー・インスタンスの複製アドレス。

CONNECTV

ノードがアクティブ・インスタンスに接続されているかどうかを示します。

BACKLOG

このインスタンスがどれだけ遅れているかを KB 数で示します。

CONNINST

指定されたインスタンスがこのインスタンスに接続されているかどうかを示します。

ALTDAT

この情報が最後に更新された日付を示します (更新されたことがない場合には空白)。

ALTTIME

この情報が最後に更新された時刻を示します (更新されたことがない場合には空白)。

手順

- キュー・マネージャーの一部であるポッドを見つけます。

```
oc get pod --selector app.kubernetes.io/instance=nativeha-qm
```

- いずれかのポッドで **dspm** を実行します。

```
oc exec -t Pod dspm
```

```
oc ish Pod
```

対話式シェルの場合は、**dspm** を直接実行できる場所です。

- キュー・マネージャー・インスタンスがアクティブ・インスタンスとして実行されているか、それともレプリカとして実行されているか判別するには、次のようにします

```
oc exec -t Pod dspmq -o status -m QMgrName
```

BOB という名前のキュー・マネージャーのアクティブ・インスタンスからは、次の状況が報告されます

```
QMNAME(BOB)          STATUS(Running)
```

BOB という名前のキュー・マネージャーのレプリカ・インスタンスからは、次の状況が報告されます

```
QMNAME(BOB)          STATUS(Replica)
```

非アクティブ・インスタンスからは、次の状況が報告されます

```
QMNAME(BOB)          STATUS(Ended Immediately)
```

- 指定されたポッド内のインスタンスのネイティブ HA 運用状況を判別するには、次のようにします

```
oc exec -t Pod dspmq -o nativeha -m QMgrName
```

BOB という名前のキュー・マネージャーのアクティブ・インスタンスからは、次のような状況が報告されます

```
QMNAME(BOB)          ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
```

BOB という名前のキュー・マネージャーのレプリカ・インスタンスからは、次のような状況が報告されます

```
QMNAME(BOB)          ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
```

BOB という名前のキュー・マネージャーの非アクティブ・インスタンスからは、次のような状況が報告されます

```
QMNAME(BOB)          ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
```

- ネイティブ HA 構成内のすべてのインスタンスのネイティブ HA 運用状況を判別するには、次のようにします

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

キュー・マネージャー BOB のアクティブ・インスタンスを実行しているノード上でこのコマンドを発行すると、以下のような状況が表示されます

```
QMNAME(BOB)          ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

キュー・マネージャー BOB のレプリカ・インスタンスを実行しているノード上でこのコマンドを発行すると、以下のような状況が表示されます。これは、レプリカの 1 つで処理が遅れていることを示しています

```
QMNAME(BOB)          ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(No) BACKLOG(435)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

キュー・マネージャー BOB の非アクティブ・インスタンスを実行しているノード上でこのコマンドを発行すると、以下のような状況が表示されます

```
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()
```

どのインスタンスがアクティブでどれがレプリカになるかをまだネゴシエーションしている間にコマンドを発行すると、次の状況が表示されます

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

関連タスク

147 ページの『例: IBM MQ Operator を使用したネイティブ HA の構成』

この例では、IBM MQ Operator を使用して、ネイティブ高可用性フィーチャーを使用するキュー・マネージャーを OpenShift Container Platform にデプロイします。相互 TLS は、認証に使用され、TLS 証明書からキュー・マネージャー内の ID にマップされます。

関連資料

[dspmq \(キュー・マネージャーの表示\) コマンド](#)

ネイティブ HA の詳細チューニング

タイミングと間隔のチューニングに関する詳細設定。デフォルトがシステムの要件に適合しないことが判明している場合を除いて、これらの設定を使用する必要はないはずです。

ネイティブ HA を構成するための基本的なオプションは、QueueManager API を使用して処理されます。IBM MQ Operator はこの API を使用して、基礎となるキュー・マネージャー INI ファイルを自動的に構成します。NativeHALocal インスタンス・スタンザの下には、INI ファイルを使用してのみ構成可能な拡張オプションがいくつかあります。INI ファイルの構成方法について詳しくは、135 ページの『例: MQSC ファイルと INI ファイルの提供』も参照してください。

HeartbeatInterval

ハートビート間隔は、ネイティブ HA キュー・マネージャーのアクティブ・インスタンスがネットワーク・ハートビートを送信する頻度をミリ秒単位で定義します。ハートビート間隔値の有効範囲は 500 (0.5 秒) から 60000 (1 分) で、この範囲外の値を使用するとキュー・マネージャーの開始が失敗します。この属性を省略すると、デフォルト値の 5000 (5 秒) が使用されます。各インスタンスで同じハートビート間隔を使用する必要があります。

HeartbeatTimeout

ハートビート・タイムアウトは、ネイティブ HA キュー・マネージャーのレプリカ・インスタンスが、アクティブ・インスタンスが応答しないと判断するまで待機する時間の長さを定義します。ハートビート間隔タイムアウト値の有効範囲は 500 (0.5 秒) から 120000 (2 分) です。ハートビート・タイムアウトの値は、ハートビート間隔以上でなければなりません。

無効な値を指定すると、キュー・マネージャーの開始が失敗します。この属性が省略されると、レプリカは、新しいアクティブ・インスタンスを選択するプロセスを開始する前に 2 x HeartbeatInterval 待機します。各インスタンスで同じハートビート・タイムアウトを使用する必要があります。

RetryInterval

再試行間隔は、障害が発生した複製リンクがネイティブ HA キュー・マネージャーで再試行される頻度をミリ秒単位で定義します。再試行間隔の有効範囲は 500 (0.5 秒) から 120000 (2 分) です。この属性が省略されると、レプリカは、障害が発生した複製リンクを再試行する前に 2 x HeartbeatInterval 待機します。

ネイティブ HA キュー・マネージャーの終了

endmqm コマンドを使用して、ネイティブ HA グループの一部であるアクティブ・キュー・マネージャーまたはレプリカ・キュー・マネージャーを終了できます。

手順

- キュー・マネージャーのアクティブ・インスタンスを終了するには、この資料の「構成」セクションの「[ネイティブ HA キュー・マネージャーの終了](#)」を参照してください。

ンス・キュー・マネージャーの構成

この例では、IBM MQ Operator を使用して、複数インスタンス・キュー・マネージャーを OpenShift Container Platform にデプロイします。相互 TLS は、認証に使用され、TLS 証明書からキュー・マネージャー内の ID にマップされます。

始める前に

この例を完了するには、まず以下の前提条件を満たしておく必要があります。

- この例のための OpenShift Container Platform (OCP) プロジェクト / 名前空間を作成します。
- コマンド・ラインで OCP クラスターにログインし、上記の名前空間に切り替えます。
- 上記の名前空間に IBM MQ Operator がインストールされ、使用可能な状態になっていることを確認します。

このタスクについて

この例では、OpenShift Container Platform にデプロイするキュー・マネージャーを定義したカスタム・リソース YAML を提供しています。また、TLS を有効にしてキュー・マネージャーをデプロイするために必要な追加のステップも詳しく説明しています。

手順

1. 適切なストレージ・クラスの判別

Kubernetes クラスター内のストレージには、複数の [永続ボリューム・アクセス・モード](#) を使用してアクセスできます。複数インスタンス・キュー・マネージャーは、複数の永続ボリューム (キュー・マネージャーごとに 1 つ、少なくとも 1 つの共有ボリューム) を作成します。複数インスタンス・キュー・マネージャーの共有ボリュームでは、ReadWriteMany ストレージ・クラスを使用する必要があります。Kubernetes クラスターのデフォルトのストレージ・クラスは、通常、ReadWriteOnce ストレージ・クラス (ブロック・ストレージ) 用です。例えば、Red Hat OpenShift Data Foundation を使用している場合、ストレージ・クラス `ocs-storagecluster-cephfs` は適切な共有ファイル・システムを提供します。すべての共有ファイル・システムが同じ方法でファイル・ロックを処理するわけではないため、ファイル・システムの選択は非常に重要です。[Multiplatforms](#) での [ファイル・システム・サポートの計画](#) および [IBM MQ 複数インスタンス・キュー・マネージャー・ファイル・システムのテスト・ステートメント](#) を参照してください。

2. [137 ページの『OpenSSL を使用した自己署名 PKI の作成』](#) の説明に従って、証明書のペアを作成します。

3. MQSC コマンドと INI ファイルを含む構成マップを作成します。

MQSC コマンドを含む Kubernetes ConfigMap を作成して、新しいキューと SVRCONN チャネルを作成し、チャネルへのアクセスを許可するチャネル認証レコードを追加します。

前に作成した名前空間 ([始める前に](#)を参照) にいることを確認してから、OCP Web コンソールで、またはコマンド・ラインを使用して、以下の YAML を入力します。

```
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 は、*MTLS.SVRCONN* というチャンネルと、*EXAMPLE.QUEUE*。チャンネルは、「共通名」が *example-app1* の証明書を提示するクライアントにのみアクセスを許可するように構成されています。これは、ステップ 153 ページの『2』で作成した証明書のいずれかで使用される共通名です。この共通名を持つこのチャンネル上の接続は、*app1* というユーザー ID にマップされます。このユーザー ID は、キュー・マネージャーに接続し、サンプル・キューにアクセスすることを許可されています。INI ファイルはセキュリティ・ポリシーを使用可能にします。これは、*app1* ユーザー ID が外部ユーザー・レジストリーに存在する必要がないことを意味します。この ID は、この構成に名前としてのみ存在します。

4. キュー・マネージャーのデプロイ

以下のカスタム・リソース YAML を使用して、新しいキュー・マネージャーを作成します。このタスクを開始する前に作成した名前空間にいることを確認してから、OCP Web コンソールで、またはコマンド・ラインを使用して、以下の YAML を入力します。正しいライセンスが指定されていることを確認し、`false` を `true` に変更してライセンスを受け入れます。

```
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
```

<STORAGE CLASS> を、ステップ 153 ページの『1』で指定したストレージ・クラスに変更します。

シークレット *example-qm-tls* がステップ 153 ページの『2』で作成され、ConfigMap *example-miqm-configmap* がステップ 153 ページの『3』で作成されたことに注意してください。

可用性タイプは *MultiInstance* に設定されます。これにより、永続ストレージが自動的に選択されます。

5. キュー・マネージャーが稼働していることの確認

キュー・マネージャーがデプロイされます。続行する前に、Running 状態であることを確認してください。以下に例を示します。

```
oc get qmgr exampleqm
```

6. キュー・マネージャーへの接続のテスト

キュー・マネージャーが構成済みで使用可能であることを確認するには、141 ページの『ラップトップからキュー・マネージャーへの相互 TLS 接続のテスト』の手順に従います。

7. アクティブ・ポッドの障害の強制試行

キュー・マネージャーの自動復旧を検証するには、ポッドの障害をシミュレートします。

a) アクティブ・ポッドとスタンバイ・ポッドの表示

以下のコマンドを実行します。

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

READY フィールドでは、アクティブ・ポッドが値 1/1 を返し、スタンバイ・ポッドが値 0/1 を返すことに注意してください。

b) アクティブ・ポッドの削除

アクティブ・ポッドの絶対パス名を指定して次コマンドを実行します。

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

c) ポッドの状況の再表示

以下のコマンドを実行します。

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

d) キュー・マネージャーの状況の表示

他のポッドのフルネームを指定して、以下のコマンドを実行します。

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

アクティブ・インスタンスが変更されたことを示す状況が表示されるはずですが、以下に例を示します。

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

e) キュー・マネージャーへの接続を再度テストします。

キュー・マネージャーが回復したことを確認するには、[141 ページの『ラップトップからキュー・マネージャーへの相互 TLS 接続のテスト』](#)の手順に従います。

タスクの結果

これで、相互 TLS 認証を使用して複数インスタンス・キュー・マネージャーが正常にデプロイされ、アクティブなポッドで障害が発生したときに自動的に復旧することが確認されました。

OpenShift CP4I V9.3.0 CD Red Hat OpenShift クラスターの外部からキュー・マネージャーに接続するためのルートの構成

Red Hat OpenShift クラスターの外部から IBM MQ キュー・マネージャーにアプリケーションを接続するには、Red Hat OpenShift 経路が必要です。IBM MQ キュー・マネージャーおよびクライアント・アプリケーションで TLS を有効にする必要があります。SNI は、TLS 1.2 以上のプロトコルが使用されている場合のみ TLS プロトコルで使用できるためです。Red Hat OpenShift Container Platform Router では、IBM MQ キュー・マネージャーへの要求のルーティングに SNI が使用されます。

このタスクについて

Red Hat OpenShift Route の必要な構成は、クライアント・アプリケーションの [Server Name Indication \(SNI\)](#) の動作によって異なります。IBM MQ では、構成とクライアントのタイプに応じて 2 種類の SNI ヘッダー設定がサポートされています。SNI ヘッダーは、クライアントの宛先のホスト名に設定されるか、または IBM MQ チャンネル名に設定されます。IBM MQ でチャンネル名がどのようにホスト名にマップされるかについては、[IBM MQ で複数の証明書の機能を提供する方法](#)を参照してください。

SNI ヘッダーを IBM MQ チャンネル名に設定するか、ホスト名に設定するかは、**OutboundSNI** 属性を使用して制御します。可能な値は、**OutboundSNI=CHANNEL** (デフォルト値) または **OutboundSNI=HOSTNAME** です。詳しくは、[クライアント構成ファイルの SSL スタンザ](#)を参照してください

い。CHANNEL および HOSTNAME は、使用する正確な値です。これらは、実際のチャンネル名またはホスト名に置き換える変数名ではありません。

OutboundSNI 設定が異なるクライアントの動作

OutboundSNI が HOSTNAME に設定されていて、接続名でホスト名が指定されていると、以下のクライアントではホスト名の SNI が設定されます。

- C クライアント
- 非管理対象モードの .NET クライアント
- Java/JMS クライアント

OutboundSNI が HOSTNAME に設定されていて、接続名で IP アドレスが使用されていると、以下のクライアントではブランクの SNI ヘッダーが送信されます。

- C クライアント
- 非管理対象モードの .NET クライアント
- Java/JMS クライアント (ホスト名の逆引き DNS ルックアップを実行できない)

OutboundSNI が CHANNEL に設定されている場合や、何も設定されていない場合は、ホスト名と IP アドレスのどちらの接続名が使用されていても、IBM MQ チャンネル名が代わりに使用されて常に送信されます。

以下のクライアント・タイプでは、SNI ヘッダーを IBM MQ チャンネル名に設定できないので、**OutboundSNI** の設定に関係なく常に SNI ヘッダーをホスト名に設定しようとします。

- AMQP クライアント
- XR クライアント
- 管理対象モードの .NET クライアント (IBM MQ 9.3.0 より前)

IBM MQ 9.3.0 以降、IBM MQ 管理対象 .NET クライアントが更新され、**OutboundSNI** プロパティが HOSTNAME に設定されている場合に SERVERNAME がそれぞれのホスト名に設定されるようになりました。これにより、IBM MQ 管理対象 .NET クライアントは Red Hat OpenShift 経路を使用してキュー・マネージャーに接続できます。

クライアント・アプリケーションが IBM MQ Internet Pass-Thru (MQIPT) を介して Red Hat OpenShift クラスターにデプロイされたキュー・マネージャーに接続する場合、MQIPT は、ルート定義内の [SSLClientOutboundSNI](#) プロパティを使用して、SNI をホスト名に設定するように構成することができます。

OutboundSNI、複数の証明書、および Red Hat OpenShift 経路

IBM MQ は、SNI ヘッダーを使用して複数の証明書機能を提供します。アプリケーションが、CERTLABL フィールドを介して別の証明書を使用するように構成されている IBM MQ チャンネルに接続する場合、アプリケーションは CHANNEL の **OutboundSNI** 設定を使用して接続する必要があります。

Red Hat OpenShift 経路構成に HOSTNAME SNI が必要な場合は、IBM MQ の複数の証明書機能を使用できず、IBM MQ チャンネル・オブジェクトに CERTLABL 設定を設定できません。

OutboundSNI に CHANNEL 以外の設定を持つアプリケーションが、証明書ラベルが構成されたチャンネルに接続すると、そのアプリケーションは MQRC_SSL_INITIALIZATION_ERROR で拒否され、キュー・マネージャーのエラー・ログに AMQ9673 メッセージが出力されます。

IBM MQ が複数の証明書機能を提供する方法については、[IBM MQ が複数の証明書機能を提供する方法](#) を参照してください。

例

SNI を MQ チャンネルに設定するクライアント・アプリケーションには、接続先のチャンネルごとに新しい Red Hat OpenShift ルートが作成されている必要があります。また、適切なキュー・マネージャーにルーティンクできるようにするには、Red Hat OpenShift Container Platform クラスターで一意的チャンネル名を使用する必要があります。

IBM MQ がチャンネル名を SNI ヘッダーにマップする方法により、MQ チャンネル名の末尾が小文字にならないことが重要です。

それぞれの新規 Red Hat OpenShift ルートに必要なホスト名を判別するには、各チャンネル名を SNI アドレスにマップする必要があります。詳しくは、[IBM MQ で複数の証明書の機能を提供する方法](#)を参照してください。

次に、クラスターに以下の yaml を適用して、チャンネルごとに新しい Red Hat OpenShift 経路を作成する必要があります。

```
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
```

クライアント・アプリケーション接続の詳細の構成

以下のコマンドを実行すると、クライアント接続で使用するホスト名を判別できます。

```
oc get route <Name of hostname based Route (for example "<Queue Manager Name>-ibm-mq-qn")>
-n <namespace of your MQ deployment> -o jsonpath="{.spec.host}"
```

クライアント接続用のポートは、Red Hat OpenShift Container Platform ルーターが使用するポート (通常は 443) に設定する必要があります。

関連タスク

173 ページの『[IBM MQ Console クラスターにデプロイされた Red Hat OpenShift への接続](#)』

Red Hat OpenShift Container Platform クラスターにデプロイされているキュー・マネージャーの IBM MQ Console に接続する方法について説明します。

CP4I > Deprecated > Removed IBM Cloud Pak for Integration Operations

Dashboard との統合

IBM Cloud Pak for Integration によるトランザクションのトレース機能は、Operations Dashboard によって提供されます。

始める前に



重要:

Deprecated > V 9.3.0 > V 9.3.0 IBM MQ Operator 2.0.0 以降、Operations Dashboard は推奨されなくなり、これ以上の更新はありません。Operations Dashboard の新しい使用法は作成しないでください。

V 9.3.3 > Removed IBM MQ Operator 2.4.0 以降、Operations Dashboard が削除されました。そのキュー・マネージャー・コンテナー・イメージをサポートする IBM MQ Operator 上では、9.3.3.0-r1 より古い既存のキュー・マネージャーに対して Operations Dashboard が引き続き使用される可能性があることに注意してください。IBM MQ Operator のバージョン・サポートについては、12 ページの『[使用可能な IBM MQ のバージョン](#)』を参照してください。

Operations Dashboard のサポートは 2024 年 6 月 30 日に終了します。詳しくは、[ソフトウェアの営業活動終了およびサポート終了](#)を参照してください。

このタスクについて

Operations Dashboard との統合を有効にすると、MQ API 出口がキュー・マネージャーにインストールされます。この API 出口が、キュー・マネージャーを流れるメッセージに関するトレース・データを Operations Dashboard のデータ・ストアに送信します。

MQ クライアント・バインディングを使用して送信されるメッセージのみがトレースされることに注意してください。

手順

1. トレースを有効にしてキュー・マネージャーをデプロイします。

デフォルトでは、トレース機能は無効になっています。

IBM Cloud Pak for Integration Platform UI (previously the Platform Navigator) を使用してデプロイする場合は、「**トレースの有効化 (Enable Tracing)**」を「**オン (On)**」に設定し、「**トレース名前空間 (Tracing Namespace)**」を Operations Dashboard がインストールされている名前空間に設定して、デプロイ時にトレースを有効にすることができます。キュー・マネージャーのデプロイについて詳しくは、[IBM Cloud Pak for Integration Platform UI を使用したキュー・マネージャーのデプロイ](#)を参照してください。

Red Hat OpenShift CLI または Red Hat OpenShift Web コンソールを使用してデプロイする場合は、以下の YAML スニペットを使用してトレースを使用可能にすることができます。

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

重要: MQ を Operations Dashboard に登録する (次の手順を参照) まで、キュー・マネージャーは開始されません。

この機能を有効にすると、キュー・マネージャー・コンテナに加えて、2つのサイドカー・コンテナ(「エージェント」と「コレクター」)が実行されるようになります。これらのサイドカー・コンテナのイメージは、メインの MQ イメージと同じレジストリーにあり、使用するプル・ポリシーとプル・シークレットも同じです。CPU とメモリーの制限を構成するための追加の設定を使用できます。

2. この名前空間にオペレーション・ダッシュボード統合を持つキュー・マネージャーが初めてデプロイされる場合は、オペレーション・ダッシュボードを使用して [レジスター](#) を実行する必要があります。登録すると、キュー・マネージャーのポッドの正常な始動に必要な Secret オブジェクトが作成されます。

OpenShift CP4I Operator 2.2.0 IBM MQ と IBM Instana トレースの統合

IBM Instana を使用して、IBM Cloud Pak for Integration 内のトランザクションをトレースできます。

始める前に

本書では、IBM Instana トレースについて説明します。これは、システムを介してメッセージをトレースするプロセスです。このトピックでは、IBM MQ キュー・マネージャーの状態に関する詳細を取得する IBM Instana モニターについては説明しません。IBM Instana による IBM MQ のモニターについては、[Monitoring IBM MQ](#) を参照してください。認証済みモニターについて詳しくは、[160 ページの『TLS を使用した認証済み IBM Instana モニターの構成』](#)を参照してください。

注:

- この機能は、IBM MQ Operator バージョン 2.2.0 以降でのみ使用できます。この機能は、IBM MQ バージョン 9.3.1.0-r2 以降のオペラントでのみサポートされます。
- IBM Instana トレースは、以前のバージョンの IBM MQ オペレーターおよびキュー・マネージャーで実行できますが、ネイティブでは実行できません。IBM Instana 資料の [IBM MQ トレースの構成](#) を参照してください。

IBM MQ Operator を使用して IBM Instana トレースを実行する前に、IBM Instana バックエンド・エージェントと IBM Instana エージェントの両方をデプロイする必要があります。デフォルトでは、IBM MQ キュー・マネージャーは、キュー・マネージャー・ポッドと同じノードにデプロイされた IBM Instana エージェントと通信します。

このタスクについて

IBM Instana との統合を有効にすると、IBM MQ API 出口がキュー・マネージャーにインストールされます。API 出口は、キュー・マネージャーを流れるメッセージに関するトレース・データを IBM Instana エージェントに送信します。

API 出口は、各メッセージに RFH2 ヘッダーを追加します。これらのヘッダーにはトレース情報が含まれます。

IBM Instana エージェントは、IBM Instana バックエンドへのトレース・データの送信を担当します。

IBM Instana バックエンド・エージェントおよび IBM Instana エージェントのデプロイについては、IBM Instana 資料の [CP4I Platform UI](#) での [IBM Instana モニターの有効化](#) を参照してください。

手順

標準デプロイメント

- IBM Instana トレースを有効にしてキュー・マネージャーをデプロイします。

デフォルトでは、IBM Instana トレースは無効になっています。

IBM Cloud Pak for Integration Platform UI (previously the Platform Navigator) または OpenShift Web コンソールを使用している場合:

- 「テレメトリー」 > 「トレース」 > 「インスタンス」をクリックします。
- 「**Instana** トレースを有効にする (Enable Instana tracing)」 トグルを true に設定します。

YAML を使用してデプロイする場合は、以下のスニペットを使用します。

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

拡張デプロイメント

- https 経由で IBM Instana エージェントと通信します。

デフォルトでは、IBM MQ の IBM Instana 出口は HTTP を介して IBM Instana エージェントと通信します。エージェントのホスト・アドレスは、キュー・マネージャーが実行されているノードの IP アドレスに設定されます。これは、IBM Instana 資料の「[IBM Instana モニターの有効化](#)」で説明されている構成と一致します。ここで、IBM Instana エージェントは、IBM Instana Agent Operator によって daemonset としてデプロイされます。

現在、IBM MQ の IBM Instana 出口と IBM Instana エージェントの間の通信では、http プロトコルまたは https プロトコルがサポートされています。https を使用するには、まず TLS 暗号化を使用するように IBM Instana エージェントを構成する必要があります。IBM Instana 資料の [エージェント・エンドポイントの TLS 暗号化のセットアップ](#) を参照してください。その後、以下のようにプロトコルを https に設定できます。

OpenShift Web コンソールを使用している場合:

- 「**Telemetry**」 > 「**Instana**」をクリックします。
- 「**拡張構成**」 ドロップダウン・リストを展開します。
- 「**Instana エージェント通信プロトコル**」を https に設定します。

YAML を使用してデプロイする場合は、以下のスニペットを使用します。

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

- **agentHost** を設定します。

IBM Instana エージェントが、キュー・マネージャーが実行されている OpenShift クラスターにデーモンセットとしてデプロイされていない場合は、**agentHost** 値を、IBM Instana エージェントが実行されているホスト名または IP アドレスに設定する必要があります。**agentHost** 値にプロトコルまたはポートを含めることはできません。

OpenShift Web コンソールを使用している場合:

1. 「**Telemetry**」 > 「**Instana**」をクリックします。
2. 「**拡張構成**」 ドロップダウン・リストを展開します。
3. 「**Instana エージェント・ホスト (Instana agent host)**」 テキスト・ボックスにホスト名を入力します。

YAML を使用してデプロイする場合は、以下のスニペットを使用します。

```
spec:
  telemetry:
    instana:
      enabled: true
      agentHost: 9.9.9.9
```

次のタスク

[116 ページの『Red Hat OpenShift Container Platform クラスターにキュー・マネージャーをデプロイする方法』](#)も参照してください。

TLS を使用した認証済み IBM Instana モニターの構成

IBM Instana エージェントを介してキュー・マネージャーをモニターできるようにするには、エージェントとキュー・マネージャーの両方を構成する必要があります。

始める前に

IBM Instana 資料の「[IBM MQ のモニター](#)」の「[構成](#)」セクションには、IBM Instana モニター構成に関する一般情報が記載されています。ただし、キュー・マネージャーの構成に関する詳細は含まれていません。

IBM MQ Operator を使用して IBM Instana トレースを実行する前に、IBM Instana バックエンド・エージェントと IBM Instana エージェントの両方をデプロイする必要があります。これを行うには、IBM Instana 資料の [CP4I Platform UI](#) での [IBM Instana モニターの有効化](#) を参照してください。

手順

1. [証明書](#)を生成します。
2. [IBM Instana エージェント](#)を構成します。
3. [キュー・マネージャー](#)を構成します。
4. [検証およびデバッグ](#)。

関連タスク

[158 ページの『IBM MQ と IBM Instana トレースの統合』](#)

IBM Instana を使用して、IBM Cloud Pak for Integration 内のトランザクションをトレースできます。

一の証明書と鍵を生成します。

IBM Instana エージェントとキュー・マネージャの間の TLS 通信の場合、両方に証明書とそれに対応する秘密鍵が必要です。

始める前に

これは、[TLS を使用して認証済み IBM Instana モニターを構成する 4 つのタスクのうちの最初のタスク](#)です。

注：これらの証明書の生成に使用される値は、デモンストレーション用です。実稼働環境にデプロイする場合は、証明書のサブジェクトと有効期限が適切であることを確認してください。

手順

IBM MQ キュー・マネージャ

TLS を介して IBM Instana エージェントと通信するには、キュー・マネージャに証明書とそれに対応する秘密鍵が必要です。これらが既にある場合は、このセクションをスキップしてください。

1. キュー・マネージャの証明書と秘密鍵を生成します。

以下のコマンドを実行します。

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

IBM Instana エージェント

エージェントが IBM MQ キュー・マネージャとの TLS 通信を実行するには、エージェントに証明書とそれに対応する秘密鍵が必要です。使用する JKS 鍵ストア内に秘密鍵と証明書が既にある場合は、このセクションをスキップしてください。

2. IBM Instana エージェントの証明書と秘密鍵を生成します。

以下のコマンドを実行します。

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

3. 証明書と秘密鍵を PKCS12 鍵ストアに保管します。

以下のコマンドを実行します。 *your_password* は、鍵ストアを保護するために使用するパスワードに置き換えてください。この置換は、後続のすべてのステップで実行します。

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

4. PKCS12 鍵ストアを JKS 鍵ストアに変換します。

以下のコマンドを実行します。

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

5. 証明書にラベルを付けます。

以下のコマンドを実行します。

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

- キュー・マネージャー証明書を鍵ストアにインポートします。

以下のコマンドを実行します。

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

次のタスク

これで、[IBM Instana モニター用にエージェントを構成する準備](#)ができました。

OpenShift CP4I Operator 2.2.0 Instana モニター: エージェントの構成

鍵ストアを IBM Instana エージェントにマウントしてから、特定のキュー・マネージャーのモニターを構成します。

始める前に

このタスクは、[IBM Instana エージェントおよびキュー・マネージャーの証明書と鍵を生成](#)していることを前提としています。

手順

IBM Instana エージェントへの鍵ストアのマウント

- IBM Instana エージェント名前空間内の JKS 鍵ストアから秘密を作成します。

以下のコマンドを実行します。 `keystore_secret_name` は、使用する名前に置き換えてください。この置換は、後続のすべてのステップで実行します。

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

- `instana-agent` 名前空間で、`oc edit daemonset instana-agent` コマンドを使用して `instana-agent` デモン・セットを編集し、以下の追加の `volumeMount` とボリュームを追加します。

```
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
```

特定のキュー・マネージャーのモニターの構成

- `instana-agent` 名前空間で、`oc edit configmap instana-agent` コマンドを使用して `instana-agent` 構成マップを編集します。
- `configuration.yaml`: |の下に以下のセクションを追加します。このセクションを既に定義している場合は、新しいキュー・マネージャーをリストに追加するだけです。

```
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'
```

説明:

- `your_password` は JKS 鍵ストアのパスワードです。

- `QUEUE_MANAGER_NAME` は、キュー・マネージャー・オペランドの名前ではなく、デプロイする基礎となる IBM MQ キュー・マネージャーの名前です。

注: `QUEUE_MANAGER_NAME` が基礎となるキュー・マネージャー名に設定されておらず、代わりにオペランドに設定されている場合、モニターは機能しません。基礎となる名前は、キュー・マネージャー・オペランドの `spec.queuemanager.name` で定義されます。

5. `instana-agent` 名前空間内の `instana-agent` ポッドを削除します。これにより、それらのユーザーは再始動し、新しい設定でモニターを開始します。

次のタスク

これで、IBM Instana モニター用にキュー・マネージャーを構成する準備ができました。

OpenShift CP4I Operator 2.2.0 Instana モニター: キュー・マネージャーの構成

TLS を使用して IBM Instana エージェントと通信するキュー・マネージャーをセットアップします。この接続の認証は、[SSLPEERMAP](#) を使用して行われます。

始める前に

このタスクは、[IBM Instana モニター用にエージェントを構成している](#)ことを前提としています。

手順

1. MQSC と INI の両方を使用してキュー・マネージャーを構成します。

MQSC は、新しい TLS 対応チャンネルをセットアップするために使用されます。その後、必須フィールドが指定された証明書がある場合に、接続先の IBM Instana エージェントを認証するようにそのチャンネルを構成します。この場合、`CN=instana-agent,OU=app team1` フィールドを含む証明書を持つ接続クライアントをユーザー `app1` にマップします。次に、MQSC は、IBM Instana モニターに必要な操作を実行する権限をユーザー `app1` に付与します。

INI ファイルは、外部ユーザー `app1` に許可を付与するために使用されます。

以下の構成マップには、必須の MQSC 設定と INI 設定が含まれています。キュー・マネージャーの名前空間にデプロイします。

```
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
```

ここで、`your-queue-manager-namespace` は、キュー・マネージャーがデプロイされる名前空間です。

注: ユーザー定義キューをモニターする場合は、構成マップ MQSC に追加の行を追加して、DSP、CHG、および GET 許可をそれらのキューに付与する必要があります。以下に例を示します。

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

この例では、MQSC データと INI データに configmap を使用しますが、追加した内容が機密情報である場合は、シークレットを使用できます。MQSC および INI を使用したデプロイに関する一般情報については、[135 ページの『例: MQSC ファイルと INI ファイルの提供』](#)を参照してください。

2. TLS 接続を行うには、キュー・マネージャーが IBM Instana エージェントの証明書を信頼する必要があります。これを行うには、IBM Instana エージェントの証明書のみを含むシークレットを作成します。

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

3. キュー・マネージャーは、TLS ハンドシェイク用に独自の証明書を提示する必要があり、関連付けられた秘密鍵へのアクセスを必要とします。以前に作成したか、既に所有している鍵と証明書を含む秘密をデプロイします。

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

構成マップとシークレットが作成されると、キュー・マネージャー自体を作成する準備が整います。

4. キュー・マネージャー YAML がキュー・マネージャー・コンテナで環境変数 **MQSNOAUT** を設定していないことを確認します。

そうしないと、有効にした後に認証メカニズムが機能しなくなります。デプロイメント後にこの変数を削除しても、メカニズムは再有効化されないため、キュー・マネージャーを再作成する必要があります。

5. キュー・マネージャー定義に以下のセクションを追加します。ここで、**MYQM** はキュー・マネージャーの名前です。

```
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
```

この仕様のフラグが立てられたセクションについて、以下で説明します。

- a. 基礎となるキュー・マネージャーに固有の名前を指定したことを確認してください。基礎となるキュー・マネージャーに固有の名前がない場合は、モニターが意図したとおりに機能しない可能性があります。この名前は、前に編集した IBM Instana エージェント構成マップ内の名前と一致している必要があります。
- b. 構成マップに書き込まれた INI 情報がキュー・マネージャーに追加されます。
- c. 構成マップに書き込まれた MQSC 情報がキュー・マネージャーに追加されます。

- d. キュー・マネージャーの証明書と秘密鍵がキュー・マネージャーの鍵ストアに追加されます。
- e. IBM Instana エージェント証明書がキュー・マネージャーのトラストストアに追加されます。
6. オプション: モニター対象キュー・マネージャーで IBM Instana トレースを有効にします。
これを行う場合は、[158 ページの『IBM MQ と IBM Instana トレースの統合』](#)を参照してください。
7. キュー・マネージャーをデプロイします。

次のタスク

これで、[IBM Instana モニターの検証とデバッグ](#)を行う準備ができました。

OpenShift CP4I Operator 2.2.0 Instana モニター: 検証およびデバッグ

IBM Instana エージェントを介してキュー・マネージャーをモニターできるようにするには、エージェントとキュー・マネージャーの両方を構成する必要があります。

始める前に

この作業は、[IBM Instana モニター用にキュー・マネージャーが構成されている](#)ことを前提としています。

手順

の検証

1. デプロイメントが正常に完了したことを確認するには、IBM Instana ダッシュボードでキュー・マネージャーを表示します。

キュー・マネージャーは、アプリケーション・ページのサービス・セクションと「インフラストラクチャー」ビューにも表示されます。

デバッグ

注: これらのデバッグ手順では、デーモンセットとして実行されている IBM Instana エージェントの OpenShift デプロイメントを想定しています。

IBM Instana ダッシュボードにキュー・マネージャーが表示されない場合は、キュー・マネージャーの構成が誤っている可能性があります。調査するには、以下の手順を使用します。

2. アクティブなキュー・マネージャー・ポッドが実行されているノードを識別します。

キュー・マネージャーの名前空間で以下のコマンドを実行します。

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

3. キュー・マネージャーと同じノードで実行されている IBM Instana エージェント・ポッドを判別するには、`instana-agent` 名前空間で同じコマンドを実行します。

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

4. IBM Instana エージェント側の問題を理解しやすくするために、IBM Instana エージェント・ポッドのログを取得し、「mq」またはキュー・マネージャーの名前に関連する項目を探します。

以下のコマンドを実行します。

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

5. キュー・マネージャーのログを確認してください。

エージェントがキュー・マネージャーへの接続を試行した場合、キュー・マネージャーのログに、接続が失敗した理由が示されているはずです。以下のコマンドを実行します。

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

タスクの結果

これで、[TLS を使用した認証済み IBM Instana モニターの構成のための 4 つのタスク](#)がすべて完了しました。

OpenShift CP4I Red Hat OpenShift CLI を使用した、カスタム MQSC および INI ファイルを使用したイメージの作成

Red Hat OpenShift Container Platform パイプラインを使用して、新規の IBM MQ コンテナ・イメージを作成できます。このイメージを使用するキュー・マネージャーに適用する MQSC ファイルと INI ファイルも指定できます。このタスクは、プロジェクト管理担当者が実行する必要があります。

始める前に

[Red Hat OpenShift Container Platform のコマンド・ライン・インターフェース](#)をインストールする必要があります。

cloudctl login (IBM Cloud Pak for Integration の場合) または **oc login** を使用してクラスターにログインします。

Red Hat OpenShift プロジェクトに IBM Entitled Registry の Red Hat OpenShift シークレットがない場合は、[ライセンス・キー・シークレットの作成の手順](#)に従います。

手順

1. ImageStream の作成

イメージ・ストリームおよびそのストリームに関連付けられたタグによって、Red Hat OpenShift Container Platform 内からコンテナ・イメージを参照するための抽象化が可能になります。イメージ・ストリームおよびそのストリームのタグによって、使用可能なイメージを確認できます。また、必要とする特定のイメージがリポジトリ内で変更されたとしても、確実にそのイメージを使用することができます。

```
oc create imagestream mymq
```

2. 新規イメージ用に BuildConfig を作成

BuildConfig は、新しいイメージのビルドを許可します。これは、IBM 公式イメージに基づきますが、コンテナの始動時に実行される MQSC ファイルまたは INI ファイルが追加されます。

a) BuildConfig リソースを定義する YAML ファイルを作成します

例えば、以下を内容とする「mq-build-config.yaml」というファイルを作成します。

```
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'
```

ベースの IBM MQ が指定されている 2 箇所を、使用するバージョンとフィックスを表す正しいベース・イメージを指すように置き換える必要があります (詳しくは、35 ページの『[IBM MQ Operator のリリース履歴](#)』を参照)。フィックスが適用されたときには、この手順を繰り返してイメージを再ビルドする必要があります。

この例では、IBM の公式イメージに基づいて新規イメージを作成し、"my.mqsc" および "my.ini" というファイルを /etc/mqm ディレクトリーに追加します。このディレクトリー内にある MQSC ファイルまたは INI ファイルは、始動時にコンテナーによって適用されます。INI ファイルは、**crtmqm -ii** オプションを使用して適用され、既存の INI ファイルとマージされます。MQSC ファイルは、アルファベット順に適用されます。

MQSC コマンドは、キュー・マネージャーが開始されるたびに実行されるので、反復可能であることが重要です。通常、これは、REPLACE パラメーターを DEFINE コマンドに追加すること、および IGNSTATE (YES) パラメーターを START コマンドまたは STOP コマンドに追加することを意味します。

b) BuildConfig をサーバーに適用します。

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

3. ビルドを実行してイメージを作成します。

a) ビルドを開始します。

```
oc start-build mymq
```

次のような出力が表示されます。

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

b) ビルドの状況を確認します。

例えば、前の手順で返されたビルド ID を使用して、次のコマンドを実行します。

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

4. 新規イメージを使用してキュー・マネージャーをデプロイします。

[116 ページの『Red Hat OpenShift Container Platform クラスターにキュー・マネージャーをデプロイする方法』](#)で説明している手順に従って、新規カスタム・イメージを YAML に追加します。

以下の YAML スニペットを通常の QueueManager YAML に追加できます。my-namespace は使用する Red Hat OpenShift プロジェクト/名前空間です。image は前に作成したイメージの名前 (例えば、「mymq:latest-amd64」) です。

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

関連タスク

[116 ページの『Red Hat OpenShift Container Platform クラスターにキュー・マネージャーをデプロイする方法』](#)

この例では、一時 (非永続) ストレージを使用する「クイック・スタート」キュー・マネージャーをデプロイし、IBM MQ セキュリティーをオフにします。メッセージは、キュー・マネージャーの再始動後は保持されません。構成を調整することで、キュー・マネージャーのさまざまな設定を変更できます。

キュー・マネージャー・リソースへのカスタム・アノテーションとカスタム・ラベルの追加

QueueManager メタデータにカスタム・アノテーションとカスタム・ラベルを追加します。

このタスクについて

PVCを除くすべてのリソースにカスタム・アノテーションとカスタム・ラベルを追加できます。カスタム・アノテーションまたはカスタム・ラベルが既存のキーと一致する場合は、IBM MQ Operator によって設定される値が使用されます。

手順

- カスタム・アノテーションを追加します。

ポッドなどのキュー・マネージャー・リソースにカスタム・アノテーションを追加するには、`metadata`の下にアノテーションを追加します。以下に例を示します。

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
  annotations:
    annotationKey: "value"
```

- カスタム・ラベルを追加します。

ポッドなどのキュー・マネージャー・リソースにカスタム・ラベルを追加するには、`metadata`の下にラベルを追加します。以下に例を示します。

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
  labels:
    labelKey: "value"
```

OpenShift CP4I 実行時 Webhook チェックの無効化

実行時 Webhook チェックによって、ストレージ・クラスがキュー・マネージャーで実行可能かどうかを確認します。パフォーマンスを向上させたい場合や、ご使用の環境で有効でない場合は、このチェックを無効にできます。

このタスクについて

実行時 Webhook チェックは、キュー・マネージャー構成に対して実行します。選択したキュー・マネージャー・タイプにストレージ・クラスが適しているかを確認します。

キュー・マネージャーの作成にかかる時間を短縮したい場合や、ご使用の特定の環境で有効でない場合は、チェックを無効にできます。

注: 実行時 Webhook チェックを無効にすると、すべてのストレージ・クラス値が有効になります。その結果、キュー・マネージャーの失敗につながることもあります。

手順

- 実行時 Webhook チェックを無効にします。

`metadata`の下に以下のアノテーションを追加します。以下に例を示します。

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
  annotations:
    "com.ibm.cp4i/disable-webhook-runtime-checks" : "true"
```


更新の無効化

IBM MQ Operator は、キュー・マネージャー仕様内の指定されていない値をデフォルト値で更新します。キュー・マネージャー仕様に変更を加えないようにする場合は、この動作を無効にすることができます。キュー・マネージャー状況フィールドは引き続き更新されます。

手順

- キュー・マネージャーのデフォルト値の更新を無効にします。

metadata の下に以下のアノテーションを追加します。以下に例を示します。

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
  annotations:
    "com.ibm.mq/write-defaults-spec" : "false"
```

注：この機能は、MQ Operator 2.1.0 以降のバージョンでのみ使用できます。IBM MQ Operator 2.1.0 以降、クイック・スタートの例では、このアノテーションがデフォルトで適用されます。

V 9.3.4 読み取り専用ルート・ファイル・システムを使用した IBM MQ コンテナの実行

IBM MQ Operator 3.0.0 および IBM MQ コンテナ 9.3.4.0 から、読み取り専用のルート・ファイル・システムで実行するように IBM MQ コンテナを構成できます。これにより、攻撃者はコンテナ内の悪意のあるコードをコピーして実行することができなくなります。

このタスクについて

読み取り専用ルート・ファイル・システムを有効にすると、コンテナ・ファイルが変更不可能になります。つまり、コンテナ・ファイル・システムでは、ファイルを表示することはできますが、変更することはできず、新規ファイルを作成することはできません。ファイルは、マウントされたファイル・システム上でのみ変更または作成することができます。

読み取り専用ルート・ファイル・システムが使用可能な場合、2つの一時ボリューム Scratch および Tmp が作成され、それぞれコンテナ内の /run ディレクトリーおよび /tmp ディレクトリーにマウントされます。

- 「スクラッチ」ボリュームには、キュー・マネージャーの構成に使用されるファイル、鍵ストア、およびその他のファイルが含まれています。
- Tmp ボリュームには、キュー・マネージャー RAS ファイルなどの診断ファイルが含まれています。

これらのボリュームは一時ボリュームであるため、これらのボリューム上のファイルはポッドの再始動時に失われます。

キュー・マネージャー・データ用に作成されるボリュームのタイプは、ストレージ・タイプによって異なります。デフォルトでは、永続ボリュームがマウントされます。あるいは、ストレージ・タイプが ephemeral の場合は、一時ボリュームがマウントされます。ボリューム内のデータのサイズが **sizeLimit** プロパティに指定された値を超える場合、Kubernetes はコンテナを排出して新しいコンテナを作成できます。IBM MQ Operator 3.0.0 より前では、キュー・マネージャー・データに一時ストレージを使用する場合、サイズ制限は適用されませんでした。

読み取り専用ルート・ファイル・システムは、デフォルトでは使用可能になっていません。有効にするには、以下の手順を実行します。

手順

1. `spec.securityContext` API を使用して、読み取り専用ルート・ファイル・システムを有効にします。
キュー・マネージャーの場合は、[197 ページの『.spec.securityContext』](#)の `readOnlyRootFilesystem` プロパティを `true` に設定します。
IBM MQ Operator は、`Scratch` および `Tmp` という 2 つの一時ボリュームを作成します。
2. オプション: キュー・マネージャーのデータ・ストレージ・タイプを設定または変更します。
デフォルトでは、永続ボリューム要求は `/mnt/mqm` にマウントされます。あるいは、[195 ページの『.spec.queueManager.storage.queueManager』](#)で `type` プロパティが `ephemeral` に設定されている場合、一時ボリュームが作成され、マウントされます。
3. 一時ボリュームごとに、データがどの程度増大する可能性があるかを慎重に検討してください。SI 単位を含め、`sizeLimit` プロパティの値を適宜設定します。
 - `Scratch` 一時ボリュームの場合、[196 ページの『.spec.queueManager.storage.scratch』](#)で `sizeLimit` プロパティを設定します。デフォルト値は「100M」です。
 - `Tmp` 一時ボリュームの場合、[197 ページの『.spec.queueManager.storage.tmp』](#)で `sizeLimit` プロパティを設定します。デフォルト値は「2Gi」です。
 - キュー・マネージャー・ボリュームの `type` が `ephemeral` に設定されている場合、[195 ページの『.spec.queueManager.storage.queueManager』](#)に `sizeLimit` プロパティを設定します。デフォルト値は「2Gi」です。

OpenShift V 9.3.4 IBM MQ Operator を使用した基本レジストリーでの IBM MQ Console の構成

IBM MQ Console にログインするために、キュー・マネージャーに独自の構成を提供することができます。

始める前に

IBM MQ Advanced for Developers ライセンスを使用してキュー・マネージャーをデプロイする場合は、単純な構成が組み込まれています。[25 ページの『\[MQ 9.3.4 2023 年 12 月\] admin および app ユーザーのパスワードを指定する方法を説明するキュー・マネージャー YAML の例』](#)を参照してください。

IBM Cloud Pak for Integration ライセンス・キュー・マネージャーをデプロイする場合は、シングル・サインオンを使用して IBM MQ Console にログインするために、IBM Cloud Pak for Integration Keycloak との統合を有効にすることができます。[173 ページの『IBM MQ Console クラスターにデプロイされた Red Hat OpenShift への接続』](#)を参照してください。

手順

1. パスワードを作成し、`securityUtility` を使用して暗号化します。
`ConfigMap` は、キュー・マネージャーへのアクセスに使用する資格情報を保管するために使用されます。セキュリティを強化するために、`securityUtility` コマンドを使用してこれらの資格情報をエンコードします。
あるいは、Kubernetes 層で資格情報を保護するシークレットを使用することもできます。ただし、モニター・ツールやトラブルシューティング・ツールを使用すると、基礎となるファイルがセキュアでなくなる可能性があります。
2. オプション: **Red Hat OpenShift コマンド・ライン・インターフェース (CLI)** にログインします。
OpenShift CLI を使用している場合は、`oc login` を使用してログインします。
あるいは、OpenShift コンソールを使用することもできます。
3. ご使用の構成で **ConfigMap** を作成します。
XML 構成の作成については、[IBM MQ Console および REST API セキュリティー](#)を参照してください。

以下の例では、グループ MQWebAdminGroup 内にユーザーを作成します。MQWebAdminGroup のメンバーには、MQWebAdmin 役割が割り当てられます。この例のそれぞれの指定の意味は次のとおりです。

- **USERNAME** および **PASSWORD** を独自の値に置き換える **必要があります**。この例では、**USERNAME** が 2 回使用されていることに注意してください。

NAMESPACE は、IBM MQ Operator がデプロイされ、キュー・マネージャーがデプロイされる場所、または既にデプロイされている場所として指定する **必要があります**。

a) OpenShift コンソールまたはコマンド行を使用して、以下の ConfigMap を作成します。

```
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) オプション: コマンド行を使用する場合は、ConfigMap を適用します。

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

残りのステップでは、以下のいずれかのオプションを選択します。

- IBM MQ Console にアクセスするための構成を使用して、新しいキュー・マネージャーをデプロイします。
 - 既存のキュー・マネージャーに対するアクセス権限を IBM MQ Console に付与する構成を適用します。
4. オプション: **IBM MQ Console** にアクセスするための構成を使用して、新規キュー・マネージャーをデプロイします。

a) キュー・マネージャーを作成します。

以下のいずれかのオプションを使用して、認証および許可プロバイダーを手動に設定し、新しく作成した ConfigMap mqwebuserconfigmap を指定します。

- オプション 1: キュー・マネージャー YAML を使用する

キュー・マネージャー YAML の web セクションの下に以下のコードを追加します。

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

- オプション 2: OpenShift コンソール・フォーム・ビューを使用して、以下の操作を行います。
 - i) OpenShift コンソールで、「オペレーター」 > 「インストール済みオペレーター」を選択します。
 - ii) IBM MQ Operator のデプロイメントを選択します。
 - iii) 「キュー・マネージャー」を選択し、「作成」 **QueueManager** をクリックします。
 - iv) キュー・マネージャーに関連するオプションを選択します。
 - v) 「**Web**」を選択し、「**Web サーバーを使用可能にする**」を true に設定します。
 - vi) 「**拡張構成**」リスト・ボックスを開きます。
 - vii) 「**コンソール**」リスト・ボックスで、「**認証**」と「**許可**」の両方の **プロバイダー** を manual に設定します。
 - viii) 「**構成**」リスト・ボックスを開きます。
 - ix) **ConfigMap** リスト・ボックスを開き、ステップ [170 ページの『3』](#) で作成した ConfigMap mqwebuserconfigmap を選択します。
 - x) 「**作成**」をクリックします。

これで、ステップ [170 ページの『3』](#) で作成した ConfigMap で指定した資格情報を使用して、新しいキュー・マネージャーの IBM MQ Console にアクセスできるようになりました。

5. オプション: 既存のキュー・マネージャーに対して **IBM MQ Console** を有効にする構成を適用します。

IBM MQ Console を有効にするキュー・マネージャーの YAML を編集します。

- a. OpenShift コンソールで、「オペレーター」 > 「インストール済みオペレーター」を選択します。
- b. IBM MQ Operator のデプロイメントを選択します。
- c. 「キュー・マネージャー」を選択し、キュー・マネージャーの名前を選択します。
- d. 「**YAML**」を選択します。
- e. キュー・マネージャー YAML の既存の web セクションを以下のコードに置き換えます。

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

- f. **保存** をクリックします。

これで、ステップ [170 ページの『3』](#) で作成した ConfigMap で指定した資格情報を使用して、既存のキュー・マネージャーの IBM MQ Console にアクセスできるようになりました。

OpenShift CP4I IBM MQ Operator を使用した IBM MQ

手順

- [173 ページの『IBM MQ Console クラスターにデプロイされた Red Hat OpenShift への接続』](#).
- [174 ページの『IBM MQ Operator 使用時のモニター』](#).
- [179 ページの『Red Hat OpenShift CLI を使用したキュー・マネージャー構成のバックアップおよびリストア』](#).

OpenShift への接続

Red Hat OpenShift Container Platform クラスターにデプロイされているキュー・マネージャーの IBM MQ Console に接続する方法について説明します。

このタスクについて

IBM MQ Console の URL は、Red Hat OpenShift Web コンソールまたは IBM Cloud Pak for Integration Platform UI (previously the Platform Navigator) の QueueManager の詳細ページで確認できます。あるいは、以下のコマンドを実行することにより、Red Hat OpenShiftCLI からそれを検出することもできます。

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

IBM Cloud Pak for Integration ライセンスを使用している場合:

- IBM MQ Operator 3.0.0 以降の場合、IBM MQ Console は、ID およびアクセス管理に Keycloak を使用します。IBM Cloud Pak for Integration 資料の [Identity and Access management](#) を参照してください。
- バージョン 3.0.0 より前の IBM MQ Operator デプロイメントの場合、IBM MQ Console は IBM Cloud Pak Identity and Access Manager (IAM) を使用します。IAM コンポーネントは、クラスター管理者によって既にセットアップされている可能性があります。ただし、Red Hat OpenShift クラスターで初めて IAM が使用された場合は、初期管理者パスワードを取得する必要があります。[初期管理パスワードの取得](#) を参照してください。

IBM MQ ライセンスを使用している場合、IBM MQ Console は事前構成されていないため、自分で構成する必要があります。詳しくは、[ユーザーおよび役割の構成](#)を参照してください。例については、[170 ページの『IBM MQ Operator を使用した基本レジストリーでの IBM MQ Console の構成』](#)を参照してください。

関連タスク

[155 ページの『Red Hat OpenShift クラスターの外部からキュー・マネージャーに接続するためのルートの構成』](#)

Red Hat OpenShift クラスターの外部から IBM MQ キュー・マネージャーにアプリケーションを接続するには、Red Hat OpenShift 経路が必要です。IBM MQ キュー・マネージャーおよびクライアント・アプリケーションで TLS を有効にする必要があります。SNI は、TLS 1.2 以上のプロトコルが使用されている場合にのみ TLS プロトコルで使用できるためです。Red Hat OpenShift Container Platform Router では、IBM MQ キュー・マネージャーへの要求のルーティングに SNI が使用されます。

IBM MQ Console の許可は、IBM Cloud Pak for Integration Platform UI (previously the Platform Navigator) ではなく、IBM Cloud Pak 管理ハブを介して管理されます。IBM MQ は、IBM Cloud Pak for Integration によって提供される「自動化」権限を使用せず、代わりに IBM Cloud Pak Identity and Access Manager (IAM) によって有効化される基本的な権限を使用します。

手順

1. IBM Cloud Pak 管理コンソールを開きます。

IBM Cloud Pak for Integration Platform UI で、ツールバーの右上隅にある Cloud Pak スイッチャー (9 ドット・アイコン) をクリックし、**IBM Cloud 「Pak 管理」** パネルをクリックします。

2. 左上隅のナビゲーション・メニューで、「**ID およびアクセス (Identity and access)**」を選択し、「**チームおよびサービス ID (Teams and services IDs)**」を選択します。
3. チームを作成し、それにユーザーを追加します。

- a) 「**チームの作成**」を選択します。
- b) チーム名を入力し、管理するユーザーのセキュリティー・ドメインを選択します。
- c) ユーザーを検索します。

これらのユーザーは、ID プロバイダーに既に存在している必要があります。

- d) 各ユーザーを見つけたら、それらのユーザーに役割を付与します。 IBM MQ Console を使用して IBM MQ を管理するには、「Administrator」または「Cluster Administrator」でなければなりません。
4. 各ユーザーを名前空間に追加します。
- a) 編集するチームを選択します。
 - b) 「リソース」 > 「リソースの管理」を選択します。
 - c) このチームに管理させる名前空間を選択します。 これらは、キュー・マネージャーを持つ任意の名前空間にすることができます。

OpenShift CP4I IBM MQ Operator 使用時のモニター

IBM MQ Operator が管理するキュー・マネージャーは、Prometheus と互換性のあるメトリックを生成できます。

これらのメトリックは、Red Hat OpenShift Container Platform (OCP) モニター・スタックを使用して表示できます。 OCP の「メトリック」タブを開き、「監視」 > 「メトリック」をクリックします。 キュー・マネージャーのメトリックはデフォルトで有効になっていますが、`.spec.metrics.enabled` を `false` に設定することで無効にすることができます。

Prometheus は、データベースと規則を評価してメトリックを時系列で取得するエンジンです。 Prometheus は、IBM MQ コンテナで公開されるメトリック・エンドポイントを利用して照会を行うことができます。 MQ システム・トピックから、モニタリングとアクティビティ・トレースを行うためのメトリックが生成されます。

OpenShift Container Platform には、Prometheus サーバーを使用する自己更新型のモニタリング・スタックが、事前インストールおよび事前構成されています。 ユーザー定義のプロジェクトをモニターするには、OpenShift Container Platform モニタリング・スタックを構成する必要があります。 詳しくは、[Enabling monitoring for user-defined projects](#) を参照してください。 IBM MQ Operator によって `ServiceMonitor` が作成されるのは、メトリックを有効にして `QueueManager` を作成するときです。 Prometheus オペレーターはこれをディスカバーできます。

旧バージョンの IBM Cloud Pak for Integration では、代わりに IBM Cloud Platform Monitoring サービスを使用して Prometheus サーバーを提供することもできます。

OpenShift CP4I IBM MQ Operator の使用時にパブリッシュされるメトリック

キューマネージャーコンテナは、Red Hat OpenShift モニタリングと互換性のあるメトリクスを公開することができます。

メトリック	タイプ	説明
<code>ibmmq_qmgr_commit_total</code>	counter	コミット・カウント
<code>ibmmq_qmgr_cpu_load_fifteen_minute_average_percentage</code>	gauge	CPU 負荷 - 15 分間の平均
<code>ibmmq_qmgr_cpu_load_five_minute_average_percentage</code>	gauge	CPU 負荷 - 5 分間の平均
<code>ibmmq_qmgr_cpu_load_one_minute_average_percentage</code>	gauge	CPU 負荷 - 1 分間の平均
<code>ibmmq_qmgr_destructive_get_bytes_total</code>	counter	破壊的 GET のインターバルにおける合計 - バイト数
<code>ibmmq_qmgr_destructive_get_total</code>	counter	破壊的 GET のインターバルにおける合計 - 数

メトリック	タイプ	説明
ibmmq_qmgr_durable_subscription_alter_total	counter	永続サブスクリプション変更回数
ibmmq_qmgr_durable_subscription_create_total	counter	永続サブスクリプション作成回数
ibmmq_qmgr_durable_subscription_delete_total	counter	永続サブスクリプション削除回数
ibmmq_qmgr_durable_subscription_resume_total	counter	永続サブスクリプション再開回数
ibmmq_qmgr_errors_file_system_free_space_percentage	gauge	MQ エラー・ファイル・システム - 空き領域
ibmmq_qmgr_errors_file_system_in_use_bytes	gauge	MQ エラー・ファイル・システム - 使用中バイト数
ibmmq_qmgr_expired_message_total	counter	期限切れメッセージ数
ibmmq_qmgr_failed_browse_total	counter	失敗したブラウズの数
ibmmq_qmgr_failed_mqcb_total	counter	失敗した MQCB の数
ibmmq_qmgr_failed_mqclose_total	counter	失敗した MQCLOSE の数
ibmmq_qmgr_failed_mqconn_mqconnx_total	counter	失敗した MQCONN/MQCONNX の数
ibmmq_qmgr_failed_mqget_total	counter	失敗した MQGET - 数
ibmmq_qmgr_failed_mqinq_total	counter	失敗した MQINQ の数
ibmmq_qmgr_failed_mqopen_total	counter	失敗した MQOPEN の数
ibmmq_qmgr_failed_mqput1_total	counter	失敗した MQPUT1 の数
ibmmq_qmgr_failed_mqput_total	counter	失敗した MQPUT の数
ibmmq_qmgr_failed_mqset_total	counter	失敗した MQSET の数
ibmmq_qmgr_failed_mqsubrq_total	counter	失敗した MQSUBRQ の数

メトリック	タイプ	説明
ibmmq_qmgr_failed_subscription_create_alter_resume_total	counter	サブスクリプションの作成/変更/再開に失敗した回数
ibmmq_qmgr_failed_subscription_delete_total	counter	サブスクリプション削除に失敗した回数
ibmmq_qmgr_failed_topic_mqput_mqput1_total	counter	失敗したトピック MQPUT/MQPUT1 の数
ibmmq_qmgr_fdc_files	gauge	MQ FDC ファイル数
ibmmq_qmgr_log_file_system_in_use_bytes	gauge	ログ・ファイル・システム - 使用中バイト数
ibmmq_qmgr_log_file_system_max_bytes	gauge	ログ・ファイル・システム - 最大バイト数
ibmmq_qmgr_log_in_use_bytes	gauge	ログ - 使用中バイト数
ibmmq_qmgr_log_logical_written_bytes_total	counter	ログ - 書き込まれた論理バイト数
ibmmq_qmgr_log_max_bytes	gauge	ログ - 最大バイト数
ibmmq_qmgr_log_occupied_by_reusable_extents_bytes	gauge	ログ - 再使用可能エクステン트가占有するバイト数
ibmmq_qmgr_log_physical_written_bytes_total	counter	ログ - 書き込まれた物理バイト数
ibmmq_qmgr_log_primary_space_in_use_percentage	gauge	ログ - 使用中の現行 1 次スペース
ibmmq_qmgr_log_required_for_media_recovery_bytes	gauge	ログ - メディア・リカバリーに必要なバイト数
ibmmq_qmgr_log_workload_primary_space_utilization_percentage	gauge	ログ - ワークロード 1 次スペースの使用状況
ibmmq_qmgr_log_write_latency_seconds	gauge	ログ - 書き込み待ち時間
ibmmq_qmgr_log_write_size_bytes	gauge	ログ - 書き込みサイズ
ibmmq_qmgr_mqcb_total	counter	MQCB の数

メトリック	タイプ	説明
ibmmq_qmgr_mqclose_total	counter	MQCLOSE の数
ibmmq_qmgr_mqconn_mqconnx_total	counter	MQCONN/MQCONNX の数
ibmmq_qmgr_mqctl_total	counter	MQCTL の数
ibmmq_qmgr_mqdisc_total	counter	MQDISC の数
ibmmq_qmgr_mqinq_total	counter	MQINQ の数
ibmmq_qmgr_mqopen_total	counter	MQOPEN の数
ibmmq_qmgr_mqput1_mqput1_bytes_total	counter	MQPUT/MQPUT1 のバイト数のインターバルにおける合計
ibmmq_qmgr_mqput1_mqput1_total	counter	MQPUT/MQPUT1 の数のインターバルにおける合計
ibmmq_qmgr_mqset_total	counter	MQSET の数
ibmmq_qmgr_mqstat_total	counter	MQSTAT の数
ibmmq_qmgr_mqsubrq_total	counter	MQSUBRQ の数
ibmmq_qmgr_non_durable_subscription_create_total	counter	非永続サブスクリプション作成回数
ibmmq_qmgr_non_durable_subscription_delete_total	counter	非永続サブスクリプション削除回数
ibmmq_qmgr_non_persistent_message_browse_bytes_total	counter	非持続メッセージのブラウズ - バイト数
ibmmq_qmgr_non_persistent_message_browse_total	counter	非持続メッセージのブラウズ - 数
ibmmq_qmgr_non_persistent_message_destructive_get_total	counter	非持続メッセージの破壊的 GET - 数
ibmmq_qmgr_non_persistent_message_get_bytes_total	counter	取得された非持続メッセージ - バイト数
ibmmq_qmgr_non_persistent_message_mqput1_total	counter	非持続メッセージ MQPUT1 の数

メトリック	タイプ	説明
ibmmq_qmgr_non_persistent_message_mqput_total	counter	非持続メッセージ MQPUT の数
ibmmq_qmgr_non_persistent_message_put_bytes_total	counter	書き込まれた非持続メッセージ - バイト数
ibmmq_qmgr_non_persistent_topic_mqput_mqput1_total	counter	非持続 - トピック MQPUT/MQPUT1 の数
ibmmq_qmgr_persistent_message_browse_bytes_total	counter	持続メッセージのブラウズ - バイト数
ibmmq_qmgr_persistent_message_browse_total	counter	持続メッセージのブラウズ - 数
ibmmq_qmgr_persistent_message_destructive_get_total	counter	持続メッセージの破壊的 GET - 数
ibmmq_qmgr_persistent_message_get_bytes_total	counter	取得された持続メッセージ - バイト数
ibmmq_qmgr_persistent_message_mqput1_total	counter	持続メッセージ MQPUT1 の数
ibmmq_qmgr_persistent_message_mqput_total	counter	持続メッセージ MQPUT の数
ibmmq_qmgr_persistent_message_put_bytes_total	counter	書き込まれた持続メッセージ - バイト数
ibmmq_qmgr_persistent_topic_mqput_mqput1_total	counter	持続 - トピック MQPUT/MQPUT1 の数
ibmmq_qmgr_published_to_subscribers_bytes_total	counter	サブスクライバーへのパブリッシュ - バイト数
ibmmq_qmgr_published_to_subscribers_message_total	counter	サブスクライバーへのパブリッシュ - メッセージ数
ibmmq_qmgr_purged_queue_total	counter	ページされたキュー数
ibmmq_qmgr_queue_manager_file_system_free_space_percentage	gauge	キュー・マネージャー・ファイル・システム - 空き領域

メトリック	タイプ	説明
ibmmq_qmgr_queue_manager_file_system_in_use_bytes	gauge	キュー・マネージャー・ファイル・システム - 使用中バイト数
ibmmq_qmgr_ram_free_percentage	gauge	RAM 空き領域パーセンテージ
ibmmq_qmgr_ram_usage_estimate_for_queue_manager_bytes	gauge	RAM 合計バイト数 - キュー・マネージャーの見積もり
ibmmq_qmgr_rollback_total	counter	ロールバック数
ibmmq_qmgr_system_cpu_time_estimate_for_queue_manager_percentage	gauge	システム CPU 時間 - キュー・マネージャーのパーセンテージ見積もり
ibmmq_qmgr_system_cpu_time_percentage	gauge	システム CPU 時間パーセンテージ
ibmmq_qmgr_topic_mqput_mqput1_total	counter	トピック MQPUT/MQPUT1 のインターバルにおける合計
ibmmq_qmgr_topic_put_bytes_total	counter	書き込まれたトピック・バイト数のインターバルにおける合計
ibmmq_qmgr_trace_file_system_free_space_percentage	gauge	MQ トレース・ファイル・システム - 空き領域
ibmmq_qmgr_trace_file_system_in_use_bytes	gauge	MQ トレース・ファイル・システム - 使用中バイト数
ibmmq_qmgr_user_cpu_time_estimate_for_queue_manager_percentage	gauge	ユーザー CPU 時間 - キュー・マネージャーのパーセンテージ見積もり
ibmmq_qmgr_user_cpu_time_percentage	gauge	ユーザー CPU 時間パーセンテージ

関連情報

[システム・トピックにパブリッシュされるメトリック](#)

Red Hat OpenShift CLI を使用したキュー・マネージャー構成のバックアップおよびリストア

キュー・マネージャー構成をバックアップすると、キュー・マネージャー構成が失われた場合に、キュー・マネージャーをその定義から再構築することができます。この手順を実行しても、キュー・マネージャーのログ・データはバックアップされません。メッセージは特定の状況で出される一時的なものなので、履歴ログ・データは復元時の対象となりません。

始める前に

cloudctl login (IBM Cloud Pak for Integration の場合) または **oc login** を使用してクラスターにログインします。

手順

- キュー・マネージャー構成をバックアップします。

dmpmqcfg コマンドを使用して、IBM MQ キュー・マネージャーの構成をダンプすることができます。

- キュー・マネージャーのポッドの名前を取得します。
例えば、次のコマンドを実行します。ここで、*queue_manager_name* は QueueManager リソースの名前です。

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

- 出力をローカル・マシン上のファイルに指定して、ポッド上で **dmpmqcfg** コマンドを実行します。

dmpmqcfg がキュー・マネージャーの MQSC 構成を出力します。

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

- キュー・マネージャー構成を復元します。

前のステップで概説したバックアップ手順に従っている場合は、キュー・マネージャー構成を含む **backup.mqsc** ファイルが必要になります。このファイルを新しいキュー・マネージャーに適用することで、構成を復元できます。

- キュー・マネージャーのポッドの名前を取得します。
例えば、次のコマンドを実行します。ここで、*queue_manager_name* は QueueManager リソースの名前です。

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

- ポッド上で **runmqsc** コマンドを実行し、**backup.mqsc** ファイルの内容を読み込みます。

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

OpenShift CP4I IBM MQ Operator に関する問題のトラブルシューティング

IBM MQ Operator で問題が発生する場合、説明されている手法を使用して問題の診断と解決を実施します。

手順

- [180 ページの『IBM MQ Operator を使用してデプロイされたキュー・マネージャーのトラブルシューティング情報の収集』](#)
- [182 ページの『トラブルシューティング: キュー・マネージャー・データへのアクセスの取得』](#)

OpenShift CP4I IBM MQ Operator を使用してデプロイされたキュー・マネージャーのトラブルシューティング情報の収集

新しいサポート Case を提出する際に IBM サポートに提供する必要があるトラブルシューティング情報を収集する。

手順

- クラウド・プロバイダー情報を収集します。

これは、Red Hat OpenShift クラスターをホストするクラウド・プロバイダーです (例えば、IBM Cloud)。

- アーキテクチャー情報を収集します。

Red Hat OpenShift クラスターのアーキテクチャーは、以下のいずれかです。

- Linux for x86-64
 - Linux on Power Systems (ppc64le)
 - Linux for IBM Z
3. IBM MQ デプロイメント情報を収集します。
- a) bash/zsh シェルを使用して、Red Hat OpenShift クラスターにログオンします。
 - b) 以下の環境変数を設定します。

```
export QM=QueueManager_name
export QM_NAMESPACE=QueueManager_namespace
export MQ_OPERATOR_NAMESPACE=mq_operator_namespace
```

ここで、`QueueManager_name` は QueueManager リソースの名前、`QueueManager_namespace` はそれがデプロイされている名前空間、`mq_operator_namespace` は IBM MQ Operator がデプロイされている名前空間です。これは、QueueManager 名前空間と同じ場合があります。

- c) 以下のコマンドを実行し、結果の出力ファイルをすべて IBM サポートに提供します。

```
# 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
```

注:

これらのコマンドの大部分は、キュー・マネージャーがデプロイされている名前空間へのアクセス権限を必要とします。ただし、IBM MQ Operator が **cluster-scoped** にインストールされている場合は、IBM MQ Operator ログを収集するために、**クラスター管理者** アクセス権限が追加が必要になることがあります。

関連タスク

IBM サポートのトラブルシューティング情報の収集

OpenShift CP4I トラブルシューティング: キュー・マネージャー・データへのアクセスの取得

PVC インспекター・ツールを使用して、キュー・マネージャー・ポッドに対してリモート・シェルを確立できないキュー・マネージャー PVC 上のファイルにアクセスできるようにします。これは、ポッドが **Error** 状態または **CrashLoopBackOff** 状態であることが原因である可能性があります。このツールは、IBM MQ Operator によってデプロイされたキュー・マネージャーで使用するために設計されています。

始める前に

PVC Inspector ツールを使用する場合。キュー・マネージャーの名前空間へのアクセス権限が必要です。

このタスクについて

トラブルシューティングを支援するために、特定のキュー・マネージャーに関連付けられた永続ボリューム要求 (PVC) に保管されているデータにアクセスできます。これを行うには、ツールを使用して、一連のインспекター・ポッドに PVC をマウントします。その後、リモート・シェルをインспекター・ポッドのいずれかに取得して、ファイルを読み取ることができます。

デプロイメントのタイプに応じて、1つから3つのインспекター・ポッドが作成されます。Native-HA キュー・マネージャーまたは複数インスタンス・キュー・マネージャーの特定のポッドに固有のボリュームは、関連付けられた PVC インспекター・ポッドで使用可能です。共有ボリュームはすべてのインспекターで使用可能です。インспекター・ポッドの名前には、関連付けられたキュー・マネージャー・ポッドの名前が含まれます。

手順

1. MQ PVC インспекター・ツールをダウンロードします。

このツールは、<https://github.com/ibm-messaging/mq-pvc-tool> から入手できます。

2. クラスターにログインしていることを確認します。
3. キュー・マネージャーの名前と、キュー・マネージャーが実行されている名前空間を調べます。
4. キュー・マネージャーに対してインспекター・ツールを実行します。
 - a) キュー・マネージャー名とその名前空間名を指定して、以下のコマンドを実行します。

```
./pvc-tool.sh queue_manager_name queue_manager_namespace_name
```

- b) ツールが完了したら、以下のコマンドを実行して、作成されるインспекター・ポッドを表示します。

```
oc get pods
```

5. インспекター・ポッドにマウントされたファイルを表示します。

- a) 各 PVC インспекター・ポッドはキュー・マネージャー・ポッドに関連付けられるため、複数のインспекター・ポッドが存在する可能性があります。以下のコマンドを実行して、これらのポッドのいずれかにアクセスします。

```
oc rsh pvc-inspector-pod-name
```

マウントされた PVC ディレクトリーが入っているディレクトリーに置かれます。

- b) 以下のコマンドを実行して、PVC ディレクトリーをリストします。

```
ls
```

- c) リモート・シェル・セッションの外部で以下のコマンドを実行して、PVC のリストを表示します。

```
oc get pvc
```

- d) 以下のコマンドを実行して、ツールによって作成されたポッドをクリーンアップします。

```
oc delete pods -l tool=mq-pvc-inspector
```

OpenShift CP4I IBM MQ Operator の API リファレンス

IBM MQ は、Red Hat OpenShift コンテナ・プラットフォームとのネイティブ統合を提供する、Kubernetes オペレーターを提供します。

OpenShift CP4I mq.ibm.com/v1beta1 の API リファレンス

v1beta1 API を使用して、QueueManager リソースを作成および管理できます。

OpenShift CP4I CD CP4I-LTS mq.ibm.com/v1beta1 のライセンスのリファレンス

現行バージョンのライセンス

spec.license.license フィールドには、同意しようとしているライセンスのライセンス ID が含まれていなければなりません。有効な値は以下のとおりです。

spec.license.license の値	spec.license.use の値	ライセンス情報	利用可能な IBM MQ のバージョン
L-VTPK-22YZPK	Production または NonProduction	IBM Cloud Pak for Integration 2023.4.1	9.3.4 または 9.3.5
L-QYQF-8UFZBN	Production または NonProduction	IBM Cloud Pak for Integration 限定版 2023.4.1	9.3.4 または 9.3.5
L-AMRD-XH6P3Q	Production	IBM MQ Advanced および IBM MQ Advanced for Non-Production Environment 9.3 -05/2023	9.3.3、9.3.4 または 9.3.5
L-AXAF-JLZ53A	Development	IBM MQ Advanced for Developers (保証適用外) 9.3 -05/2023	9.3.3、9.3.4 または 9.3.5
L-YBXJ-ADJNSM	Production または NonProduction	IBM Cloud Pak for Integration 2023.2.1	9.3.3

spec.license.1 licenseの値	spec.license.uses の値	ライセンス情報	利用可能な IBM MQ のバージョン
L-PYRA-849GYQ	Production または NonProduction	IBM Cloud Pak for Integration 限定版 2023.2.1	9.3.3
L-RJON-CJR2RX	Production または NonProduction	IBM Cloud Pak for Integration 2022.4.1	9.3.1 または 9.3.2
L-RJON-CJR2TC	Production または NonProduction	IBM Cloud Pak for Integration 限定版 2022.4.1	9.3.1 または 9.3.2
L-UPFX-8MW49T	Production	IBM MQ Advanced および IBM MQ Advanced for Non-Production Environment 9.3 -02/2023	9.3.2
L-APIG-CAUEQC	Development	IBM MQ Advanced for Developers (保証適用外) 9.3	9.3.0、9.3.1 または 9.3.2
L-RJON-CD3JKX	Production または NonProduction	IBM Cloud Pak for Integration 2022.2.1	9.3.0 または 9.3.1
L-RJON-CD3JJU	Production または NonProduction	IBM Cloud Pak for Integration 限定版 2022.2.1	9.3.0 または 9.3.1
L-APIG-CAUEBE	Production	IBM MQ Advanced および IBM MQ Advanced for Non-Production Environment 9.3	9.3.0 または 9.3.1

ライセンスのバージョンを指定しますが、これは必ずしも IBM MQ のバージョンとは同じでないことに注意してください。

古いバージョンのライセンス

IBM MQ 9.2 資料の [旧ライセンス・バージョン](#) を参照してください。

  **QueueManager (mq.ibm.com/v1beta1) の API リファレンス**

QueueManager

QueueManager は、アプリケーションにキューイングとパブリッシュ/サブスクライブのサービスを提供する IBM MQ サーバーです。IBM MQ 資料: <https://ibm.biz/BdPZqj>。ライセンス参照: <https://ibm.biz/BdPZfq>。

フィールド	説明
apiVersion 文字列	APIVersion は、このオブジェクトの表記のスキーマのバージョンを定義します。サーバーは、認識されたスキーマを最新の内部値に変換する必要があります。認識されない値は拒否されることがあります。詳細情報: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources 。

フィールド	説明
kind 文字列	kind は、このオブジェクトが表している REST リソースを表す文字列の値です。サーバーは、クライアントが要求を送信するエンドポイントからこれを推測することがあります。更新することはできません。キャメル・ケースの値です。詳細情報: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds 。
metadata	
specQueueManagerSpec	QueueManager について必要とする状態。
statusQueueManagerStatus	QueueManager について観測された状態。

.spec

QueueManager について必要とする状態。

以下の中に含まれます:

- 184 ページの『QueueManager』

フィールド	説明
affinity	標準的な Kubernetes アフィニティー・ルール。詳しくは、 https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#affinity-v1-core を参照してください。
annotations Annotations	annotations フィールドは、ポッド・アノテーションのパススルーの役目を果たします。ユーザーはこのフィールドにアノテーションを追加することによって、ポッドにアノテーションを適用できます。ここでアノテーションを指定すると、デフォルトのアノテーションが上書きされます。MQ Operator 1.3.0 以上が必要です。
imagePullSecrets LocalObjectReference 配列	(オプション) この QueueManager で使用するイメージをプルするために使用する、同じ名前空間にあるシークレットへの参照のリスト。指定すると、それらのシークレットがプル実行プログラムに渡されて使用されます。例えば、Docker の場合は、DockerConfig タイプのシークレットだけが適用されます。詳しくは、 https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod を参照してください。
labels Labels	labels フィールドは、ポッド・ラベルのパススルーの役目を果たします。ユーザーはこのフィールドにラベルを追加することによって、ポッドにラベルを適用できます。ここでラベルを指定すると、デフォルトのラベルが上書きされません。MQ Operator 1.3.0 以上が必要です。
license License	ライセンスの受け入れを制御する設定、および使用するライセンス・メトリック。
pki PKI	Transport Layer Security (TLS) または MQ Advanced Message Security (AMS) で使用する鍵と証明書を定義するための Public Key Infrastructure の設定。
queueManager QueueManagerConfig	キュー・マネージャーのコンテナおよび基礎キュー・マネージャーの設定。
securityContext SecurityContext	キュー・マネージャー・ポッドの securityContext に追加するセキュリティー設定です。
telemetry テレメトリー	Open Telemetry 構成の設定。MQ Operator 2.2.0 以上が必要です。

フィールド	説明
template テンプレート	Kubernetes リソースの拡張テンプレート。このテンプレートは、基礎の Kubernetes リソース (StatefulSet、Pod、Service など) を IBM MQ で生成する方法をユーザーがオーバーライドすることを可能にします。これは上級者専用です。誤って使用すると MQ の正常な動作が阻害される可能性があります。QueueManager リソースの他の場所で指定された値は、このテンプレートの設定によってオーバーライドされます。
terminationGracePeriod Seconds 整数	(オプション) ポットの正常な終了にかかる時間 (秒単位)。値は負以外の整数でなければなりません。ゼロの値は、ただちに削除することを意味します。このキュー・マネージャーの終了の目標時間を達成するために、アプリケーションの切断のフェーズが押し進められます。必要な場合は、キュー・マネージャーの重要なメンテナンス・タスクが中断されます。デフォルトは 30 秒です。
tracing TracingConfig	Cloud Pak for Integration Operations Dashboard とのトレースの統合のための設定。
version 文字列	使用する MQ のバージョンを制御する設定 (必須)。例えば、9.1.5.0-r2 は、コンテナ・イメージの 2 番目のリビジョンを使用する MQ バージョン 9.1.5.0 を指します。ベース・イメージのフィックスなどのように、コンテナ固有のフィックスが、リビジョンの中で適用されることがよくあります。
web WebServerConfig	MQ Web サーバーの設定。

.spec.annotations

annotations フィールドは、ポッド・アノテーションのパススルーの役目を果たします。ユーザーはこのフィールドにアノテーションを追加することによって、ポッドにアノテーションを適用できます。ここでアノテーションを指定すると、デフォルトのアノテーションが上書きされます。MQ Operator 1.3.0 以上が必要です。

以下の中に含まれます:

- [185 ページの『.spec』](#)

.spec.imagePullSecrets

LocalObjectReference に、同じ名前空間の内部で参照されているオブジェクトを見つけることができる十分な情報が含まれています。

以下の中に含まれます:

- [185 ページの『.spec』](#)

フィールド	説明
name 文字列	参照の名前。詳細情報: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names TODO: その他の便利なフィールドを追加してください。apiVersion、kind、uid など。

.spec.labels

labels フィールドは、ポッド・ラベルのパススルーの役目を果たします。ユーザーはこのフィールドにラベルを追加することによって、ポッドにラベルを適用できます。ここでラベルを指定すると、デフォルトのラベルが上書きされます。MQ Operator 1.3.0 以上が必要です。

以下の中に含まれます:

- [185 ページの『.spec』](#)

.spec.license

ライセンスの受け入れを制御する設定、および使用するライセンス・メトリック。

以下の中に含まれます:

- [185 ページの『.spec』](#)

フィールド	説明
accept ブール値	このソフトウェアに関連付けられているライセンスを受け入れるかどうか (必須)。
license 文字列	受け入れるライセンスの ID。これは、使用する MQ のバージョンの正確なライセンス ID である必要があります。有効な値については、「 https://ibm.biz/BdPZfq 」を参照してください。
metric 文字列	使用するライセンス・メトリックを指定する設定。例えば、ProcessorValueUnit、VirtualProcessorCore、ManagedVirtualServer などです。MQ ライセンスを使用する場合のデフォルトは ProcessorValueUnit、Cloud Pak for Integration ライセンスを使用する場合のデフォルトは VirtualProcessorCore です。
use 文字列	複数の使用方法をサポートするライセンスの場合に、ソフトウェアの使用方法を制御する設定。有効な値については、「 https://ibm.biz/BdPZfq 」を参照してください。

.spec.pki

Transport Layer Security (TLS) または MQ Advanced Message Security (AMS) で使用する鍵と証明書を定義するための Public Key Infrastructure の設定。

以下の中に含まれます:

- [185 ページの『.spec』](#)

フィールド	説明
keys PKISource 配列	キュー・マネージャーの鍵リポジトリに追加する秘密鍵です。
trust PKISource 配列	キュー・マネージャーの鍵リポジトリに追加する証明書です。

.spec.pki.keys

PKISource は、鍵や証明書などの Public Key Infrastructure 情報のソースを定義します。

以下の中に含まれます:

- [187 ページの『.spec.pki』](#)

フィールド	説明
name 文字列	name は鍵や証明書のラベルとして使用されます。小文字の英数字の文字列である必要があります。
secret Secret	Kubernetes シークレットを使用して鍵を渡します。

.spec.pki.keys.secret

Kubernetes シークレットを使用して鍵を渡します。

以下の中に含まれます:

- [187 ページの『.spec.pki.keys』](#)

フィールド	説明
items 配列	キュー・マネージャーのコンテナに追加する必要がある Kubernetes シークレットの内部にある鍵。
secretName 文字列	Kubernetes シークレットの名前。

.spec.pki.trust

PKISource は、鍵や証明書などの Public Key Infrastructure 情報のソースを定義します。

以下の中に含まれます:

- [187 ページの『.spec.pki』](#)

フィールド	説明
name 文字列	name は鍵や証明書のラベルとして使用されます。小文字の英数字の文字列である必要があります。
secret Secret	Kubernetes シークレットを使用して鍵を渡します。

.spec.pki.trust.secret

Kubernetes シークレットを使用して鍵を渡します。

以下の中に含まれます:

- [188 ページの『.spec.pki.trust』](#)

フィールド	説明
items 配列	キュー・マネージャーのコンテナに追加する必要がある Kubernetes シークレットの内部にある鍵。
secretName 文字列	Kubernetes シークレットの名前。

.spec.queueManager

キュー・マネージャーのコンテナおよび基礎キュー・マネージャーの設定。

以下の中に含まれます:

- [185 ページの『.spec』](#)

フィールド	説明
availability Availability	キュー・マネージャーの可用性の設定 (アクティブとスタンバイのペアやネイティブの高可用性を使用するかどうかなど)。
debug ブール値	コンテナ固有のコードからのデバッグ・メッセージをコンテナ・ログに記録するかどうか。デフォルトは false です。
image 文字列	使用するコンテナ・イメージ。
imagePullPolicy 文字列	指定されたイメージのプルを Kubelet が試行するタイミングを制御する設定。デフォルトは IfNotPresent です。
ini INISource 配列	キュー・マネージャーに INI を提供するための設定。MQ Operator 1.1.0 以上が必要です。
livenessProbe QueueManagerLivenessProbe	Liveness プロブを制御する設定。

フィールド	説明
logFormat 文字列	このコンテナで使用するログの形式。JSON 形式のコンテナ・ログには、JSON を使用します。テキスト形式のメッセージには、Basic を使用します。デフォルトは Basic です。
metrics QueueManagerMetrics	Prometheus スタイルのメトリックの設定。
mjsc MQSCSource 配列	キュー・マネージャーに MQSC を提供するための設定。MQ Operator 1.1.0 以上が必要です。
name 文字列	基礎 MQ キュー・マネージャーの名前 (metadata.name と異なる場合)。名前の Kubernetes 規則に準拠しないキュー・マネージャー名 (例えば、大文字を含む名前) が必要な場合は、このフィールドを使用します。
readinessProbe QueueManagerReadinessProbe	Readiness プロブを制御する設定。
recoveryLogs RecoveryLogs	MQ リカバリー・ログの設定。MQ Operator 2.4.0 以上が必要です。
resources Resources	リソース要件を制御する設定。
route Route	キュー・マネージャー・ルートの設定。MQ Operator 1.4.0 以上が必要です。
startupProbe StartupProbe	始動プロブを制御する設定。MultiInstance デプロイメントと NativeHA デプロイメントにのみ適用されます。MQ Operator 1.5.0 以上が必要です。
storage QueueManagerStorage	キュー・マネージャーが永続ボリュームおよびストレージ・クラスを使用することを制御するストレージ設定です。

.spec.queueManager.availability

キュー・マネージャーの可用性の設定 (アクティブとスタンバイのペアやネイティブの高可用性を使用するかどうかなど)。

以下の中に含まれます:

- 188 ページの『.spec.queueManager』

フィールド	説明
tls Tls	NativeHA レプリカ間のセキュア通信を構成するためのオプションの TLS 設定。MQ Operator 1.5.0 以上が必要です。
type 文字列	使用する可用性のタイプ。Kubernetes が (一部の状況で) 自動的に再始動する単一ポッドには、SingleInstance を使用します。MultiInstance は、ポッドのペア (1 つは active キュー・マネージャー、もう 1 つはスタンバイ) に使用します。ネイティブの高可用性の複製には NativeHA を使用します (MQ Operator 1.5.0 以上が必要です)。デフォルトは SingleInstance です。詳しくは、 http://ibm.biz/BdqAQa を参照してください。
updateStrategy 文字列	MultiInstance および NativeHA キュー・マネージャーに使用する更新方針。RollingUpdate を使用して、キュー・マネージャーの構成が変更されるたびに自動ローリング更新を有効にします。自動ローリング更新を無効にするには、OnDelete を使用します。キュー・マネージャーの変更は、ポッドが削除された場合のみ適用されます (外部要因によってトリガーされたポッド削除も含む)。デフォルトは RollingUpdate です。MQ Operator 1.6.0 以上が必要です。

.spec.queueManager.availability.tls

NativeHA レプリカ間のセキュア通信を構成するためのオプションの TLS 設定。MQ Operator 1.5.0 以上が必要です。

以下の中に含まれます:

- [189 ページの『.spec.queueManager.availability』](#)

フィールド	説明
cipherSpec 文字列	NativeHA TLS 用の CipherSpec の名前。
secretName 文字列	Kubernetes シークレットの名前。

.spec.queueManager.ini

INI 構成ファイルのソース。

以下の中に含まれます:

- [188 ページの『.spec.queueManager』](#)

フィールド	説明
configMap ConfigMapINISource	configMap は、INI 情報が入っている Kubernetes ConfigMap を表します。
secret SecretINISource	secret は、INI 情報が入っている Kubernetes シークレットを表します。

.spec.queueManager.ini.configMap

configMap は、INI 情報が入っている Kubernetes ConfigMap を表します。

以下の中に含まれます:

- [190 ページの『.spec.queueManager.ini』](#)

フィールド	説明
items 配列	適用する必要がある、Kubernetes ソース内部の鍵。
name 文字列	Kubernetes ソースの名前。

.spec.queueManager.ini.secret

secret は、INI 情報が入っている Kubernetes シークレットを表します。

以下の中に含まれます:

- [190 ページの『.spec.queueManager.ini』](#)

フィールド	説明
items 配列	適用する必要がある、Kubernetes ソース内部の鍵。
name 文字列	Kubernetes ソースの名前。

.spec.queueManager.livenessProbe

Liveness プロブを制御する設定。

以下の中に含まれます:

- [188 ページの『.spec.queueManager』](#)

フィールド	説明
failureThreshold 整数	プローブが成功した後に、この回数以上連続して失敗すると、失敗したプローブと見なされます。デフォルトは1です。
initialDelaySeconds 整数	コンテナが始動してからプローブが開始されるまでの秒数。SingleInstance の場合、デフォルトは90秒です。MultiInstance デプロイメントと NativeHA デプロイメントの場合、デフォルトは0秒です。詳細情報: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes 。
periodSeconds 整数	プローブを実行する間隔(秒単位)。デフォルトは10秒です。
successThreshold 整数	プローブが成功した後に、この回数以上連続して成功すると、成功したプローブと見なされます。デフォルトは1です。
timeoutSeconds 整数	プローブがタイムアウトになるまでの秒数。デフォルトは5秒です。詳細情報: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes 。

.spec.queueManager.metrics

Prometheus スタイルのメトリックの設定。

以下の中に含まれます:

- [188 ページ](#)の『.spec.queueManager』

フィールド	説明
enabled ブール値	Prometheus 互換のメトリックのエンドポイントを有効にするかどうか。デフォルトは true です。

.spec.queueManager.mqsc

MQSC 構成ファイルのソース。

以下の中に含まれます:

- [188 ページ](#)の『.spec.queueManager』

フィールド	説明
configMap ConfigMapMQSCSource	configMap は、MQSC 情報が入っている Kubernetes ConfigMap を表します。
secret SecretMQSCSource	secret は、MQSC 情報が入っている Kubernetes シークレットを表します。

.spec.queueManager.mqsc.configMap

configMap は、MQSC 情報が入っている Kubernetes ConfigMap を表します。

以下の中に含まれます:

- [191 ページ](#)の『.spec.queueManager.mqsc』

フィールド	説明
items 配列	適用する必要がある、Kubernetes ソース内部の鍵。
name 文字列	Kubernetes ソースの名前。

.spec.queueManager.mqsc.secret

secret は、MQSC 情報が入っている Kubernetes シークレットを表します。

以下の中に含まれます:

- [191 ページの『.spec.queueManager.mqsc』](#)

フィールド	説明
items 配列	適用する必要がある、Kubernetes ソース内部の鍵。
name 文字列	Kubernetes ソースの名前。

.spec.queueManager.readinessProbe

Readiness プロブを制御する設定。

以下の中に含まれます:

- [188 ページの『.spec.queueManager』](#)

フィールド	説明
failureThreshold 整数	プローブが成功した後に、この回数以上連続して失敗すると、失敗したプローブと見なされます。デフォルトは 1 です。
initialDelaySeconds 整数	コンテナが起動してからプローブが開始されるまでの秒数。SingleInstance の場合、デフォルトは 10 秒です。MultiInstance デプロイメントと NativeHA デプロイメントの場合、デフォルトは 0 です。詳細情報: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes 。
periodSeconds 整数	プローブを実行する間隔 (秒単位)。デフォルトは 5 秒です。
successThreshold 整数	プローブが成功した後に、この回数以上連続して成功すると、成功したプローブと見なされます。デフォルトは 1 です。
timeoutSeconds 整数	プローブがタイムアウトになるまでの秒数。デフォルトは 3 秒です。詳細情報: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes 。

.spec.queueManager.recoveryLogs

MQ リカバリー・ログの設定。MQ Operator 2.4.0 以上が必要です。

以下の中に含まれます:

- [188 ページの『.spec.queueManager』](#)

フィールド	説明
logFilePages 整数	回復ログ・データは一連のファイルに保持されます。ログ・ファイル・サイズは、4 KB ページ単位で指定します。

.spec.queueManager.resources

リソース要件を制御する設定。

以下の中に含まれます:

- [188 ページの『.spec.queueManager』](#)

フィールド	説明
limits Limits	CPU とメモリーの設定。

フィールド	説明
requests Requests	CPU とメモリーの設定。

.spec.queueManager.resources.limits

CPU とメモリーの設定。

以下の中に含まれます:

- [192 ページ](#)の『.spec.queueManager.resources』

フィールド	説明
cpu	
memory	

.spec.queueManager.resources.requests

CPU とメモリーの設定。

以下の中に含まれます:

- [192 ページ](#)の『.spec.queueManager.resources』

フィールド	説明
cpu	
memory	

.spec.queueManager.route

キュー・マネージャー・ルートの設定。MQ Operator 1.4.0 以上が必要です。

以下の中に含まれます:

- [188 ページ](#)の『.spec.queueManager』

フィールド	説明
enabled ブール値	ルートを有効にするかどうか。デフォルトは true です。

.spec.queueManager.startupProbe

始動プローブを制御する設定。MultiInstance デプロイメントと NativeHA デプロイメントにのみ適用されます。MQ Operator 1.5.0 以上が必要です。

以下の中に含まれます:

- [188 ページ](#)の『.spec.queueManager』

フィールド	説明
failureThreshold 整数	この回数以上連続して失敗すると、失敗したプローブと見なされます。デフォルトは 24 です。
initialDelaySeconds 整数	コンテナが始動してからプローブが開始されるまでの秒数。デフォルトは 0 秒です。詳細情報: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes 。
periodSeconds 整数	プローブを実行する間隔 (秒単位)。デフォルトは 5 秒です。

フィールド	説明
successThreshold 整数	この回数以上連続して成功すると、成功したプローブと見なされます。デフォルトは 1 です。
timeoutSeconds 整数	プローブがタイムアウトになるまでの秒数。デフォルトは 5 秒です。詳細情報: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes 。

.spec.queueManager.storage

キュー・マネージャーが永続ボリュームおよびストレージ・クラスを使用することを制御するストレージ設定です。

以下の中に含まれます:

- 188 ページの『.spec.queueManager』

フィールド	説明
defaultClass 文字列	このキュー・マネージャーのすべての永続ボリュームにデフォルトで適用するストレージ・クラス。個々の永続ボリュームで独自のストレージ・クラスを定義すると、このデフォルトのストレージ・クラス設定がオーバーライドされます。type of availability が SingleInstance または NativeHA の場合は、ストレージ・クラスのタイプを ReadWriteOnce か ReadWriteMany にできます。type of availability が MultiInstance の場合は、ストレージ・クラスのタイプを ReadWriteMany にしなければなりません。
defaultDeleteClaim ブール値	キュー・マネージャーの削除時にすべてのボリュームを削除するかどうか。個々の永続ボリュームで独自の deleteClaim 値を定義すると、この defaultDeleteClaim 設定がオーバーライドされます。デフォルトは false です。
persistedData QueueManagerOptionalVolume	構成、キュー、メッセージなどの MQ の永続データに使用する永続ボリュームの詳細。マルチインスタンスのキュー・マネージャーを使用する場合は必須です。
queueManager QueueManagerVolume	あらゆるデータに使用するデフォルトの永続ボリューム (通常は、/var/mqm 下に置かれます)。他のボリュームを指定しない場合は、すべての永続データとリカバリー・ログがここに格納されます。
recoveryLogs QueueManagerOptionalVolume	MQ リカバリー・ログ用の永続ボリュームの詳細。マルチインスタンスのキュー・マネージャーを使用する場合は必須です。
scratch スクラッチ	キュー・マネージャーの Scratch 一時ボリュームの設定。このボリュームは、「/run」フォルダーとしてコンテナにマウントされます。ルート・ファイル・システムが読み取り専用設定されている場合にのみ適用されます。MQ Operator 3.0.0 以上が必要です。
tmp 一時	キュー・マネージャーの一時一時ボリュームの設定。このボリュームは、「/tmp」フォルダーとしてコンテナにマウントされます。runmqras コマンドによって作成された zip ファイルなどの診断データ・ファイルが、このボリュームに作成されます。ルート・ファイル・システムが読み取り専用設定されている場合にのみ適用されます。MQ Operator 3.0.0 以上が必要です。

.spec.queueManager.storage.persistedData

構成、キュー、メッセージなどの MQ の永続データに使用する永続ボリュームの詳細。マルチインスタンスのキュー・マネージャーを使用する場合は必須です。

以下の中に含まれます:

• 194 ページの『.spec.queueManager.storage』

フィールド	説明
class 文字列	このボリュームに使用するストレージ・クラス。type が persistent-claim である場合にのみ有効です。type of availability が SingleInstance または NativeHA の場合は、ストレージ・クラスのタイプを ReadWriteOnce か ReadWriteMany にできます。type of availability が MultiInstance の場合は、ストレージ・クラスのタイプを ReadWriteMany にしなければなりません。
deleteClaim ブール値	キュー・マネージャーの削除時にこのボリュームを削除するかどうか。
enabled ブール値	このボリュームを別個のボリュームとして使用可能にするかどうか、あるいは、デフォルトの queueManager ボリュームに置くかどうか。デフォルトは false です。
size 文字列	Kubernetes に渡す PersistentVolume のサイズ (SI 単位を含む)。type が persistent-claim である場合にのみ有効です。例: 2Gi。デフォルトは 2Gi です。
sizeLimit 文字列	ephemeral ボリュームを使用する場合のサイズの制限。ファイルは引き続き一時ディレクトリーに書き込まれるので、このオプションを使用してサイズを制限できます。type が ephemeral で、ルート・ファイル・システムが読み取り専用設定されている場合にのみ有効です。MQ Operator 3.0.0 以上が必要です。
type 文字列	使用するボリュームのタイプ。非永続ストレージを使用する場合は ephemeral を選択し、永続ボリュームを使用する場合は persistent-claim を選択します。デフォルトは persistent-claim です。

.spec.queueManager.storage.queueManager

あらゆるデータに使用するデフォルトの永続ボリューム (通常は、/var/mqm 下に置かれます)。他のボリュームを指定しない場合は、すべての永続データとリカバリー・ログがここに格納されます。

以下の中に含まれます:

• 194 ページの『.spec.queueManager.storage』

フィールド	説明
class 文字列	このボリュームに使用するストレージ・クラス。type が persistent-claim である場合にのみ有効です。type of availability が SingleInstance または NativeHA の場合は、ストレージ・クラスのタイプを ReadWriteOnce か ReadWriteMany にできます。type of availability が MultiInstance の場合は、ストレージ・クラスのタイプを ReadWriteMany にしなければなりません。
deleteClaim ブール値	キュー・マネージャーの削除時にこのボリュームを削除するかどうか。
size 文字列	Kubernetes に渡す PersistentVolume のサイズ (SI 単位を含む)。type が persistent-claim である場合にのみ有効です。例: 2Gi。デフォルトは 2Gi です。
sizeLimit 文字列	ephemeral ボリュームを使用する場合のサイズの制限。ファイルは引き続き一時ディレクトリーに書き込まれるので、このオプションを使用してサイズを制限できます。type が ephemeral で、ルート・ファイル・システムが読み取り専用設定されている場合にのみ有効です。MQ Operator 3.0.0 以上が必要です。

フィールド	説明
type 文字列	使用するボリュームのタイプ。非永続ストレージを使用する場合は ephemeral を選択し、永続ボリュームを使用する場合は persistent-claim を選択します。デフォルトは persistent-claim です。

.spec.queueManager.storage.recoveryLogs

MQ リカバリー・ログ用の永続ボリュームの詳細。マルチインスタンスのキュー・マネージャーを使用する場合は必須です。

以下の中に含まれます:

- [194 ページの『.spec.queueManager.storage』](#)

フィールド	説明
class 文字列	このボリュームに使用するストレージ・クラス。type が persistent-claim である場合にのみ有効です。type of availability が SingleInstance または NativeHA の場合は、ストレージ・クラスのタイプを ReadWriteOnce か ReadWriteMany にできます。type of availability が MultiInstance の場合は、ストレージ・クラスのタイプを ReadWriteMany にしなければなりません。
deleteClaim ブール値	キュー・マネージャーの削除時にこのボリュームを削除するかどうか。
enabled ブール値	このボリュームを別個のボリュームとして使用可能にするかどうか、あるいは、デフォルトの queueManager ボリュームに置くかどうか。デフォルトは false です。
size 文字列	Kubernetes に渡す PersistentVolume のサイズ (SI 単位を含む)。type が persistent-claim である場合にのみ有効です。例: 2Gi。デフォルトは 2Gi です。
sizeLimit 文字列	ephemeral ボリュームを使用する場合のサイズの制限。ファイルは引き続き一時ディレクトリーに書き込まれるので、このオプションを使用してサイズを制限できます。type が ephemeral で、ルート・ファイル・システムが読み取り専用で設定されている場合にのみ有効です。MQ Operator 3.0.0 以上が必要です。
type 文字列	使用するボリュームのタイプ。非永続ストレージを使用する場合は ephemeral を選択し、永続ボリュームを使用する場合は persistent-claim を選択します。デフォルトは persistent-claim です。

.spec.queueManager.storage.scratch

キュー・マネージャーの Scratch 一時ボリュームの設定。このボリュームは、「/run」フォルダーとしてコンテナにマウントされます。ルート・ファイル・システムが読み取り専用で設定されている場合にのみ適用されます。MQ Operator 3.0.0 以上が必要です。

以下の中に含まれます:

- [194 ページの『.spec.queueManager.storage』](#)

フィールド	説明
sizeLimit 文字列	一時ボリュームのサイズ制限 (SI 単位を含む)。例えば、2Gi などです。ルート・ファイル・システムが読み取り専用で設定されている場合にのみ有効です。MQ Operator 3.0.0 以上が必要です。

.spec.queueManager.storage.tmp

キュー・マネージャーの一時一時ボリュームの設定。このボリュームは、「/tmp」フォルダーとしてコンテナにマウントされます。runmqras コマンドによって作成された zip ファイルなどの診断データ・ファイルが、このボリュームに作成されます。ルート・ファイル・システムが読み取り専用で設定されている場合にのみ適用されます。MQ Operator 3.0.0 以上が必要です。

以下の中に含まれます:

- [194 ページの『.spec.queueManager.storage』](#)

フィールド	説明
sizeLimit 文字列	一時ボリュームのサイズ制限 (SI 単位を含む)。例えば、2Gi などです。ルート・ファイル・システムが読み取り専用で設定されている場合にのみ有効です。MQ Operator 3.0.0 以上が必要です。

.spec.securityContext

キュー・マネージャー・ポッドの securityContext に追加するセキュリティ設定です。

以下の中に含まれます:

- [185 ページの『.spec』](#)

フィールド	説明
fsGroup 整数	ポッド内のすべてのコンテナに適用される特殊な補助グループ。いくつかのボリューム・タイプでは、Kubelet がそのボリュームの所有権をポッドによって所有されるように変更することができます: 1. 所有 GID は、FSGroup 2 になります。setgid ビットが設定されます (ボリューム内に作成された新規ファイルは、FSGroup が所有します) 3 です。許可ビットは OR で rw-rw---- 設定されていない場合、Kubelet はボリュームの所有権と許可を変更しません。
initVolumeAsRoot ブール値	これは、PersistentVolume を初期化するコンテナで使用されるセキュリティ・コンテキストに影響を与えます。新しくプロビジョンしたボリュームには root ユーザーを使用してアクセスしなければならないストレージ・プロバイダーを使用する場合は、true に設定します。これを true に設定すると、使用できるセキュリティ・コンテキスト制約 (SCC) オブジェクトが影響を受けます。root ユーザーを許可する SCC を使用する権限を持っていないと、キュー・マネージャーの始動に失敗する可能性があります。デフォルトは false です。詳しくは、 https://docs.openshift.com/container-platform/latest/authentication/managing-security-context-constraints.html を参照してください。
readOnlyRootFilesystem ブール値	キュー・マネージャーの読み取り専用ルート・ファイル・システム設定を有効にするかどうか。デフォルトは false です。MQ Operator 3.0.0 以上が必要です。
supplementalGroups 配列	コンテナの 1 次 GID に加えて、最初のプロセスに適用されるグループのリストは、各コンテナで実行されます。指定しない場合は、コンテナにグループは追加されません。

.spec.telemetry

Open Telemetry 構成の設定。MQ Operator 2.2.0 以上が必要です。

以下の中に含まれます:

- [185 ページの『.spec』](#)

フィールド	説明
tracing トレース	Open Telemetry トレースの設定。

.spec.telemetry.tracing

Open Telemetry トレースの設定。

以下の中に含まれます:

- [197 ページの『.spec.telemetry』](#)

フィールド	説明
instana インスタナ	Instana トレースの設定。

.spec.telemetry.tracing.instana

Instana トレースの設定。

以下の中に含まれます:

- [198 ページの『.spec.telemetry.tracing』](#)

フィールド	説明
agentHost 文字列	トレース・データの送信先の Instana エージェントのホスト名。これにプロトコルを含めることはできません。
enabled ブール値	Instana トレースを有効にするかどうか。デフォルトは false です。
protocol 文字列	Instana エージェントとの通信で使用されるプロトコル。http および https がサポートされています。

.spec.template

Kubernetes リソースの拡張テンプレート。このテンプレートは、基礎の Kubernetes リソース (StatefulSet、Pod、Service など) を IBM MQ で生成する方法をユーザーがオーバーライドすることを可能にします。これは上級者専用です。誤って使用すると MQ の正常な動作が阻害される可能性があります。QueueManager リソースの他の場所で指定された値は、このテンプレートの設定によってオーバーライドされます。

以下の中に含まれます:

- [185 ページの『.spec』](#)

フィールド	説明
pod	ポッドに使用するテンプレートのオーバーライド。 https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#podspec-v1-core を参照してください。

.spec.tracing

Cloud Pak for Integration Operations Dashboard とのトレースの統合のための設定。

以下の中に含まれます:

- [185 ページの『.spec』](#)

フィールド	説明
agent TracingAgent	Cloud Pak for Integration でのみ、オプションの Tracing Agent の設定を構成できます。
collector TracingCollector	Cloud Pak for Integration でのみ、オプションの Tracing Collector の設定を構成できます。

フィールド	説明
enabled ブール値	Cloud Pak for Integration Operations Dashboard とのトレースの統合を有効にするかどうか。デフォルトは false です。
namespace 文字列	Cloud Pak for Integration Operations Dashboard がインストールされている名前空間。

.spec.tracing.agent

Cloud Pak for Integration でのみ、オプションの Tracing Agent の設定を構成できます。

以下の中に含まれます:

- 198 ページの『[.spec.tracing](#)』

フィールド	説明
image 文字列	使用するコンテナ・イメージ。
imagePullPolicy 文字列	指定されたイメージのプルを Kubelet が試行するタイミングを制御する設定。デフォルトは IfNotPresent です。
livenessProbe TracingProbe	Liveness プロブを制御する設定。
readinessProbe TracingProbe	Readiness プロブを制御する設定。

.spec.tracing.agent.livenessProbe

Liveness プロブを制御する設定。

以下の中に含まれます:

- 199 ページの『[.spec.tracing.agent](#)』

フィールド	説明
failureThreshold 整数	プロブが成功した後に、この回数以上連続して失敗すると、失敗したプロブと見なされます。デフォルトは 1 です。
initialDelaySeconds 整数	コンテナが始動してから Liveness プロブが開始されるまでの秒数。デフォルトは 10 秒です。詳細情報: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes 。
periodSeconds 整数	プロブを実行する間隔 (秒単位)。デフォルトは 10 秒です。
successThreshold 整数	プロブが成功した後に、この回数以上連続して成功すると、成功したプロブと見なされます。デフォルトは 1 です。
timeoutSeconds 整数	プロブがタイムアウトになるまでの秒数。デフォルトは 2 秒です。詳細情報: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes 。

.spec.tracing.agent.readinessProbe

Readiness プロブを制御する設定。

以下の中に含まれます:

- 199 ページの『[.spec.tracing.agent](#)』

フィールド	説明
failureThreshold 整数	プローブが成功した後に、この回数以上連続して失敗すると、失敗したプローブと見なされます。デフォルトは1です。
initialDelaySeconds 整数	コンテナが起動してから Liveness プローブが開始されるまでの秒数。デフォルトは10秒です。詳細情報: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes 。
periodSeconds 整数	プローブを実行する間隔(秒単位)。デフォルトは10秒です。
successThreshold 整数	プローブが成功した後に、この回数以上連続して成功すると、成功したプローブと見なされます。デフォルトは1です。
timeoutSeconds 整数	プローブがタイムアウトになるまでの秒数。デフォルトは2秒です。詳細情報: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes 。

.spec.tracing.collector

Cloud Pak for Integration でのみ、オプションの Tracing Collector の設定を構成できます。

以下の中に含まれます:

- [198 ページの『.spec.tracing』](#)

フィールド	説明
image 文字列	使用するコンテナ・イメージ。
imagePullPolicy 文字列	指定されたイメージのプルを Kubelet が試行するタイミングを制御する設定。デフォルトは IfNotPresent です。
livenessProbe TracingProbe	Liveness プローブを制御する設定。
readinessProbe TracingProbe	Readiness プローブを制御する設定。

.spec.tracing.collector.livenessProbe

Liveness プローブを制御する設定。

以下の中に含まれます:

- [200 ページの『.spec.tracing.collector』](#)

フィールド	説明
failureThreshold 整数	プローブが成功した後に、この回数以上連続して失敗すると、失敗したプローブと見なされます。デフォルトは1です。
initialDelaySeconds 整数	コンテナが起動してから Liveness プローブが開始されるまでの秒数。デフォルトは10秒です。詳細情報: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes 。
periodSeconds 整数	プローブを実行する間隔(秒単位)。デフォルトは10秒です。
successThreshold 整数	プローブが成功した後に、この回数以上連続して成功すると、成功したプローブと見なされます。デフォルトは1です。
timeoutSeconds 整数	プローブがタイムアウトになるまでの秒数。デフォルトは2秒です。詳細情報: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes 。

.spec.tracing.collector.readinessProbe

Readiness プローブを制御する設定。

以下の中に含まれます:

- [200 ページの『.spec.tracing.collector』](#)

フィールド	説明
failureThreshold 整数	プローブが成功した後に、この回数以上連続して失敗すると、失敗したプローブと見なされます。デフォルトは 1 です。
initialDelaySeconds 整数	コンテナが始動してから Liveness プローブが開始されるまでの秒数。デフォルトは 10 秒です。詳細情報: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes 。
periodSeconds 整数	プローブを実行する間隔 (秒単位)。デフォルトは 10 秒です。
successThreshold 整数	プローブが成功した後に、この回数以上連続して成功すると、成功したプローブと見なされます。デフォルトは 1 です。
timeoutSeconds 整数	プローブがタイムアウトになるまでの秒数。デフォルトは 2 秒です。詳細情報: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes 。

.spec.web

MQ Web サーバーの設定。

以下の中に含まれます:

- [185 ページの『.spec』](#)

フィールド	説明
console コンソール	MQ Web コンソールの設定。MQ Operator 3.0.0 以上が必要です。
enabled ブール値	Web サーバーを有効にするかどうか。デフォルトは false です。
manualConfig ManualConfig	Web サーバー XML 構成を指定するための設定。MQ Operator 3.0.0 以上が必要です。

.spec.web.console

MQ Web コンソールの設定。MQ Operator 3.0.0 以上が必要です。

以下の中に含まれます:

- [201 ページの『.spec.web』](#)

フィールド	説明
authentication 認証	MQ Web コンソールの認証設定。MQ Operator 3.0.0 以上が必要です。
authorization 許可	MQ Web コンソールの許可設定。MQ Operator 3.0.0 以上が必要です。

.spec.web.console.authentication

MQ Web コンソールの認証設定。MQ Operator 3.0.0 以上が必要です。

以下の中に含まれます:

- [201 ページの『.spec.web.console』](#)

フィールド	説明
provider 文字列	MQ Web コンソール用に使用する認証プロバイダー。integration-keycloak を使用して、Cloud Pak for Integration Platform UI (Keycloak) でシングル・サインオンを使用します。デフォルトは、Cloud Pak for Integration ライセンスを使用する場合は integration-keycloak、MQ ライセンスを使用する場合は manual です。独自の構成を指定する場合は、manual を使用します。

.spec.web.console.authorization

MQ Web コンソールの許可設定。MQ Operator 3.0.0 以上が必要です。

以下の中に含まれます:

- [201 ページの『.spec.web.console』](#)

フィールド	説明
provider 文字列	MQ Web コンソール用に使用する許可プロバイダー。integration-keycloak を使用して、Cloud Pak for Integration Keycloak によって提供される役割を使用します。独自の構成を指定する場合は、manual を使用します。デフォルトは、Cloud Pak for Integration ライセンスを使用する場合は integration-keycloak、MQ ライセンスを使用する場合は manual です。

.spec.web.manualConfig

Web サーバー XML 構成を指定するための設定。MQ Operator 3.0.0 以上が必要です。

以下の中に含まれます:

- [201 ページの『.spec.web』](#)

フィールド	説明
configMap ConfigMap	ConfigMap は、Web サーバー XML 構成を含む Kubernetes ConfigMap を表します。
secret Secret	シークレットは、Web サーバー XML 構成を含む Kubernetes シークレットを表します。シークレットを使用すると、Kubernetes レイヤー内のすべての資格情報が保護されますが、モニター・ツールやトラブルシューティング・ツールによって、基礎となるファイルが非セキュアに公開される可能性があります。セキュリティを向上させるには、「securityUtility」を使用して資格情報をエンコードします。

.spec.web.manualConfig.configMap

ConfigMap は、Web サーバー XML 構成を含む Kubernetes ConfigMap を表します。

以下の中に含まれます:

- [202 ページの『.spec.web.manualConfig』](#)

フィールド	説明
name 文字列	Kubernetes ソースの名前。

.spec.web.manualConfig.secret

シークレットは、Web サーバー XML 構成を含む Kubernetes シークレットを表します。シークレットを使用すると、Kubernetes レイヤー内のすべての資格情報が保護されますが、モニター・ツールやトラブルシ

ユーティリティ・ツールによって、基礎となるファイルが非セキュアに公開される可能性があります。セキュリティを向上させるには、「securityUtility」を使用して資格情報をエンコードします。

以下の中に含まれます:

- [202 ページの『.spec.web.manualConfig』](#)

フィールド	説明
name 文字列	Kubernetes ソースの名前。

. 状況

QueueManager について観測された状態。

以下の中に含まれます:

- [184 ページの『QueueManager』](#)

フィールド	説明
adminUiUrl 文字列	管理 UI の URL。
availability Availability	キュー・マネージャーの可用性状況。
conditions QueueManagerStatusCondition 配列	条件は、キュー・マネージャーの状態に関する最新の使用可能な監視を表します。
endpoints QueueManagerStatusEndpoint 配列	このキュー・マネージャーが公開しているエンドポイントに関する情報 (API エンドポイントや UI エンドポイントなど)。
metadata メタデータ	メタデータは、統合-Keycloak 状況など、キュー・マネージャーの追加情報を表します。
name 文字列	キュー・マネージャーの名前。
phase 文字列	キュー・マネージャーの状態のフェーズです。
versions QueueManagerStatusVersion	使用されている MQ のバージョン、および IBM Entitled Registry から取得できるその他のバージョン。

.status.availability

キュー・マネージャーの可用性状況。

以下の中に含まれます:

- [203 ページの『. 状況』](#)

フィールド	説明
initialQuorumEstablished ブール値	NativeHA に初期クォーラムが確立されているかどうか。

.status.conditions

QueueManagerStatusCondition は、キュー・マネージャーの状況を示します。

以下の中に含まれます:

- [203 ページの『. 状況』](#)

フィールド	説明
lastTransitionTime 文字列	状況のステータスが最後に遷移した時刻。
message 文字列	最後の遷移についての詳細を示す、人間が読める形式のメッセージ。
reason 文字列	最後にこのステータスに遷移した理由。
status 文字列	状況のステータス。
type 文字列	状況のタイプ。

.status.endpoints

QueueManagerStatusEndpoint は、QueueManager のエンドポイントを示します。

以下の中に含まれます:

- [203 ページの『. 状況』](#)

フィールド	説明
name 文字列	エンドポイントの名前。
type 文字列	エンドポイントのタイプ。UI エンドポイントの場合は「UI」、API エンドポイントの場合は「API」、API 資料の場合は「OpenAPI」です。
uri 文字列	エンドポイントの URI。

.status.metadata

メタデータは、統合-Keycloak 状況など、キュー・マネージャーの追加情報を表します。

以下の中に含まれます:

- [203 ページの『. 状況』](#)

フィールド	説明
integrationKeycloak IntegrationKeycloak	QueueManagerStatusIntegrationKeycloak は、QueueManager の統合 Keycloak 状況を定義します。

.status.metadata.integrationKeycloak

QueueManagerStatusIntegrationKeycloak は、QueueManager の統合 Keycloak 状況を定義します。

以下の中に含まれます:

- [204 ページの『.status.metadata』](#)

フィールド	説明
clientName 文字列	

.status.versions

使用されている MQ のバージョン、および IBM Entitled Registry から取得できるその他のバージョン。

以下の中に含まれます:

- [203 ページの『. 状況』](#)

フィールド	説明
available QueueManagerStatusVersionAvailable	IBM Entitled Registry から取得できるその他のバージョンの MQ。
reconciled 文字列	使用されている IBM MQ の具体的なバージョン。カスタム・イメージが示されている場合は、実際に使用されている MQ のバージョンと一致していない可能性があります。

.status.versions.available

IBM Entitled Registry から取得できるその他のバージョンの MQ。

以下の中に含まれます:

- [204 ページの『.status.versions』](#)

フィールド	説明
channels 配列	MQ のバージョンを自動的に更新するために使用できるチャンネル。
versions Versions 配列	使用可能な MQ の具体的なバージョン。

.status.versions.available.versions

QueueManagerStatusVersion は MQ のバージョンを示します。

以下の中に含まれます:

- [205 ページの『.status.versions.available』](#)

フィールド	説明
licenses Licenses 配列	このバージョンの QueueManager に適用可能なライセンス。
name 文字列	このバージョンの QueueManager のバージョン name 。これは、spec.version フィールドで有効な値です。

.status.versions.available.versions.licenses

QueueManagerStatusLicense はライセンスを定義します。

以下の中に含まれます:

- [205 ページの『.status.versions.available.versions』](#)

フィールド	説明
displayName 文字列	ライセンスの表示名。
link 文字列	ライセンス・コンテンツへのリンク。
matchesCurrentType ブール値	ライセンスが現在使用されているライセンスのタイプと一致するかどうか。
name 文字列	ライセンスの名前。

QueueManager (mq.ibm.com/v1beta1) の状況状態

status.conditions フィールドは、QueueManager リソースの状態に合わせて更新されます。状態は通常、異常な状況を説明するものです。正常な作動可能状態のキュー・マネージャーは、**Error** や **Pending** の状態になりません。通知のための **Warning** 状態になることはあります。

状態のサポートは IBM MQ Operator 1.2 で導入されました。
QueueManager リソースでは以下の状態が定義されています。

表 1. キュー・マネージャー状況状態

コンポーネント	状態タイプ	理由コード	メッセージ警告
QueueManager ⁹	ブロック	OperatorDependency	このインスタンスをインストールするには、[IBM Cloud Pak for Integration]で Keycloak を構成する必要があります。このインスタンスは、この QueueManager の Cp4iServicesBinding リソースで Keycloak が [KeycloakReady] として報告されるまで、[Pending] 状況のままになります。 このインスタンスをインストールするには、オペレーター [IBM IAM] が必要です。このインスタンスは、オペレーターが [IBM Cloud Pak 基本サービス] によってインストールされるまで「ブロック」状況のままになります。
	保留中	Creating	MQ キュー・マネージャーのデプロイ中です
	保留中	OidcPending	MQ キュー・マネージャーが OIDC クライアント登録を待っています。
	エラー	失敗	MQ キュー・マネージャーのデプロイが失敗しました
	警告	UnsupportedVersion	¹⁰ オペランドが、OCP バージョン <ocp_version> でサポートされていないオペレーターによってインストールされました。このオペランドはサポートされません。
	警告	CP4I-LTS サポート	¹¹ CP4I-LTS オペランド <mq_version> がインストールされていますが、拡張サポート期間に適格でないオペレーターによって管理されています。このオペランドは拡張サポート期間に対応していません。
	警告	CP4I-LTS サポート	¹² CP4I-LTS オペランド <mq_version> はインストールされていますが、OCP バージョン 4 <ocp_version> は拡張サポート期間に適格ではありません。このオペランドは拡張サポート期間に対応していません。
	警告	CP4I-LTS サポート	¹³ CP4I-LTS オペランド <mq_version> がインストールされていますが、OCP バージョン <ocp_version> は拡張サポート期間に適格ではありません。このオペランドは通常の進行状況を Web 控制台が有効になっている

⁹ 条件 Creating および Failed は、キュー・マネージャーのデプロイを完了させます。IBM Cloud Pak for Integration ライセンスを使用している場合、OidcPending 条件は、OIDC クライアント登録が IAM で完了するのを待機している間のキュー・マネージャーの状況をログに記録します。

表 1. キュー・マネージャー状況状態 (続き)

コンポーネント	状態タイプ	理由コード	メッセージ警告
ポッド ¹⁴	保留中	PodPending	MQ キュー・マネージャーのポッドのデプロイ中です
	エラー	PodFailed	MQ キュー・マネージャーのポッドのデプロイ中です
記憶域 ¹⁵	保留中	StoragePending	MQ キュー・マネージャーのストレージのプロビジョン中です
	警告	StorageEphemeral	実動 MQ キュー・マネージャーで一時的ストレージを使用しています
	エラー	StorageFailed	MQ キュー・マネージャーのストレージのプロビジョンが失敗しました

Multi 独自の IBM MQ コンテナおよびデプロイメント・コードのビルド

自作コンテナを開発します。これは最も柔軟なコンテナソリューションですが、コンテナの設定に強いスキルが必要であり、結果としてのコンテナを"所有する"必要があります。

始める前に

独自のコンテナを開発する前に、代わりに IBM MQ Operator を使用できるかどうかを検討してください。5 ページの『コンテナ内の IBM MQ の使用方法の選択』を参照してください。

このタスクについて

手順

- [208 ページの『コンテナを使用する独自の IBM MQ キュー・マネージャー・イメージの計画』](#)
- [209 ページの『サンプルの IBM MQ キュー・マネージャーのコンテナ・イメージのビルド』](#)
- [212 ページの『別々のコンテナでのローカル・バインディング・アプリケーションの実行』](#)
- [IBM MQ サンプルの Helm チャートを確認します。](#)

Multi コンテナを使用する独自の IBM MQ キュー・マネージャー・イメージの計画

コンテナで IBM MQ キュー・マネージャーを実行するには、考慮すべき要件がいくつかあります。サンプルのコンテナ・イメージは、これらの要件に対応できるようになっていますが、独自のイメージを使用する場合は、これらの要件に対応する方法を検討する必要があります。

¹⁰ オペレーター 1.4.0 以降

¹¹ オペレーター 1.4.0 以降

¹² オペレーター 1.4.0 以降

¹³ オペレーター 1.3.0 のみ

¹⁴ POD 条件は、キュー・マネージャーのデプロイメント中にポッドの状況をモニターします。PodFailed 条件が表示された場合は、キュー・マネージャー全体の条件も Failed に設定されます。

¹⁵ ストレージ条件は、永続ストレージのボリュームを作成する要求の進行状況 (StoragePending 条件) をモニターし、バインディング・エラーおよびその他の障害を報告します。ストレージのプロビジョニング中にエラーが発生した場合、StorageFailed 条件が条件リストに追加され、キュー・マネージャー全体の条件も Failed に設定されます。

プロセス監視

コンテナを実行することは、基本的に単一のプロセス (コンテナ内部の PID 1) を実行することになります。この単一のプロセスは、後で子プロセスを spawn することができます。

メインプロセスが終了すると、コンテナ・ランタイムがコンテナを停止します。IBM MQ キュー・マネージャーでは、複数のプロセスをバックグラウンドで実行する必要があります。

このため、キュー・マネージャーの実行中は、メインプロセスがアクティブな状態であることを確認する必要があります。グッド・プラクティスとして、例えば管理照会などを実行して、キュー・マネージャーがアクティブであることをこのプロセスから確認してください。

/var/mqm への移植

コンテナは、/var/mqm をボリュームとして構成する必要があります。

これを行うと、コンテナが最初に始動したときに、このボリュームのディレクトリーは空の状態です。通常、このディレクトリーにはインストール時に内容が取り込まれますが、コンテナを使用する場合は、インストールとランタイムが別々の環境になります。

これを解決するために、コンテナが始動して初めて実行されるときに、**crtmqdir** コマンドを使用して、/var/mqm にデータを取り込みます。

コンテナのセキュリティ

ランタイムのセキュリティ要件を最小限にするために、サンプルのコンテナ・イメージは、unzip 可能な IBM MQ インストールを使用してインストールされます。これにより、setuid ビットが設定されなくなり、コンテナは特権エスカレーションを使用する必要がなくなります。コンテナ・システムの中には、使用できるユーザー ID を定義したものもありますが、この unzip 可能なインストールでは、使用可能なオペレーティング・システム・ユーザーについて一切想定されていません。

Multi サンプルの IBM MQ キュー・マネージャーのコンテナ・イメージのビルド

コンテナで IBM MQ キュー・マネージャーを実行するためにサンプルのコンテナ・イメージをビルドするには、この情報を活用してください。

このタスクについて

まず、Red Hat Universal Base Image ファイル・システムと IBM MQ のクリーン・インストールが含まれているベース・イメージをビルドします。

次に、そのベースの上に、ユーザー ID とパスワードによる基本的なセキュリティを可能にする IBM MQ 構成を追加するための別のコンテナ・イメージ層をビルドします。

最後に、このイメージをファイル・システムとして使用し、ホスト・ファイル・システム上のコンテナ固有のボリュームによって提供される /var/mqm の内容を使用してコンテナを実行します。

手順

- コンテナで IBM MQ キュー・マネージャーを実行するためにサンプルのコンテナ・イメージをビルドする方法については、以下のサブピックを参照してください。
 - [209 ページの『サンプルの IBM MQ キュー・マネージャーのベース・イメージのビルド』](#)
 - [210 ページの『サンプルの構成済みの IBM MQ キュー・マネージャーのイメージのビルド』](#)

Multi サンプルの IBM MQ キュー・マネージャーのベース・イメージのビルド

独自のコンテナ・イメージの IBM MQ を使用するためには、まず、IBM MQ クリーン・インストールを含むベース・イメージをビルドする必要があります。以下の手順は、GitHub でホストされているサンプル・コードを使用してサンプルのベース・イメージをビルドする方法を示しています。

手順

- [mq-container GitHub リポジトリ](#)で提供されている Make ファイルを使用して実稼働用のコンテナ・イメージをビルドします。
GitHub の [コンテナ・イメージの作成の指示](#)に従います。
- オプション: Red Hat OpenShift Container Platform "restricted" Security Context Constraint (SCC) を使用してセキュア・アクセスを構成する予定の場合は、IBM MQ 非インストール・イメージの1つを使用してください。
これらのイメージをダウンロードするためのリンクは、[IBM MQ ダウンロード](#)の「コンテナ」セクションにあります。

タスクの結果

IBM MQ がインストールされたベース・コンテナ・イメージができました。

これで、[構成済みのサンプル IBM MQ キュー・マネージャー・イメージを作成する](#)準備ができました。

Multi サンプルの構成済みの IBM MQ キュー・マネージャーのイメージのビルド

汎用的なベースの IBM MQ コンテナ・イメージをビルドしたら、セキュアなアクセスを可能にするために独自の構成を適用する必要があります。これを行うには、汎用イメージを親として使用して、独自のコンテナ・イメージ層を作成します。

始める前に

このタスクは、[サンプルの基本 IBM MQ キュー・マネージャー・イメージの作成](#)時に "インストールなし" の IBM MQ パッケージを使用したことが前提になっています。そうでない場合は、Red Hat OpenShift Container Platform の "制限付き" のセキュリティー・コンテキスト制約 (SCC) を使用してセキュア・アクセスを構成することができなくなります。デフォルトで使用される "制限付き" の SCC では、ランダムユーザー ID が使用され、別のユーザーへの変更による特権のエスカレーションが不可になります。IBM MQ の従来の RPM ベースのインストーラーは、mqm のユーザーとグループに依存していて、実行可能プログラムで `setuid` ビットを使用します。現行バージョンの IBM MQ では、"No-Install" IBM MQ パッケージを使用すると、mqm ユーザーも mqm グループも存在しなくなります。

手順

1. 新しいディレクトリーを作成し、以下を内容とする `config.mqsc` というファイルを追加します。

```
DEFINE QLOCAL(EXAMPLE.QUEUE.1) REPLACE
```

上記の例では、ユーザー ID とパスワードによる単純な認証が使用されることに注意してください。ただし、企業が必要とする任意のセキュリティー構成を適用できます。

2. 以下を内容とする `Dockerfile` というファイルを作成します。

```
FROM mq  
COPY config.mqsc /etc/mqm/
```

3. 次のコマンドを使用して、カスタム・コンテナ・イメージをビルドします。

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

「.」は、先ほど作成した2つのファイルが含まれているディレクトリーです。

Docker はそのイメージを使用して一時コンテナを作成し、残りのコマンドを実行します。

注: Red Hat Enterprise Linux (RHEL) では、コマンド `docker` (RHEL V7) または `podman` (RHEL V7 または RHEL V8) を使用します。Linux では、コマンドの先頭に `sudo` を指定して `docker` コマンドを実行し、追加の特権を取得する必要があります。

4. カスタマイズした新しいイメージを実行して、先ほど作成したディスク・イメージを含む新しいコンテナを作成します。

新しいイメージ層では、実行する特定のコマンドを指定しなかったため、親イメージから継承されています。親のエントリー・ポイント (このコードは GitHub で提供されています):

- キュー・マネージャーを作成する
- キュー・マネージャーを開始する
- デフォルトのリッスナーを作成する
- 次に、 `/etc/mqm/config.mqsc` から MQSC コマンドを実行します

以下のコマンドを発行して、カスタマイズした新しいイメージを実行します。

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

説明:

最初の env パラメーター

IBM IBM WebSphere® MQ のライセンスに同意したことを知らせる環境変数をコンテナに渡しています。LICENSE 変数を view に設定してライセンスを表示することもできます。

IBM MQ ライセンスについて詳しくは、[IBM MQ ライセンス情報](#) を参照してください。

2 番目の env パラメーター

使用するキュー・マネージャー名を設定しています。

volume パラメーター

MQ が `/var/mqm` 上で実際に `/var/example` に書き込む必要があることをコンテナに指示します。

このオプションは、後で永続データを保持したままコンテナを簡単に削除できることを意味します。このオプションにより、ログ・ファイルの表示も簡単になります。

publish パラメーター

ホスト・システムのポートをコンテナのポートにマップしています。デフォルトでは、コンテナはコンテナ自身の内部 IP アドレスを使用して実行されます。つまり、ポートを公開するには、具体的にポートをマップする必要があります。

この例では、ホストのポート 1414 をコンテナのポート 1414 にマップしています。

detach パラメーター

バックグラウンドでコンテナを実行します。

タスクの結果

構成したコンテナ・イメージをビルドしたので、`docker ps` コマンドを使用して、実行中のコンテナを表示できます。`docker top` コマンドを使用して、コンテナで実行されている IBM MQ プロセスを表示できます。



重要:

`docker logs ${CONTAINER_ID}` コマンドを使用して、コンテナのログを表示できます。

次のタスク

- `docker ps` コマンドを使用してもコンテナが表示されない場合は、コンテナが失敗している可能性があります。`docker ps -a` コマンドを使用して、失敗したコンテナを表示できます。
- `docker ps -a` コマンドを使用すると、コンテナ ID が表示されます。この ID は、`docker run` コマンドを発行したときにも表示されたものです。

- `docker logs ${CONTAINER_ID}` コマンドを使用して、コンテナのログを表示できます。

Multi 別々のコンテナでのローカル・バインディング・アプリケーションの実行

コンテナ間でプロセス名前空間を共有することにより、IBM MQ キュー・マネージャーとは別のコンテナにある IBM MQ へのローカル・バインディング接続を必要とするアプリケーションを実行できます。

このタスクについて

以下の制約事項に従う必要があります。

- `--pid` 引数を使用して、各コンテナの PID 名前空間を共有する必要があります。
- `--ipc` 引数を使用して、各コンテナの IPC 名前空間を共有する必要があります。
- 以下のいずれかが必要です。
 1. `--uts` 引数を使用して、各コンテナの UTS 名前空間をホストと共有する。
 2. `-h` 引数または `--hostname` 引数を使用して、各コンテナを同じホスト名にする。
- IBM MQ データディレクトリは、`/var/mqm` ディレクトリの下にあるすべてのコンテナが利用可能なボリュームにマウントする必要があります。

以下の例では、サンプルの IBM MQ コンテナ・イメージを使用しています。このイメージの詳細については、[Github](#) を参照してください。

手順

1. 次のコマンドを発行して、ボリュームとして機能する一時ディレクトリを作成します。

```
mkdir /tmp/dockerVolume
```

2. 次のコマンドを発行して、1つのコンテナ内にキュー・マネージャー (QM1) を作成し、名前 `sharedNamespace` を指定します。

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

3. 次のコマンドを発行して、`ibmcom/mq` をベースとする `secondaryContainer` という2つ目のコンテナを始動します。ただし、キュー・マネージャーは作成しません。

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

4. 次のコマンドを発行して、2つ目のコンテナに対して `dspsmq` コマンドを実行し、両方のキュー・マネージャーの状況を調べます。

```
docker exec secondaryContainer dspsmq
```

5. 次のコマンドを実行して、もう一方のコンテナで実行されているキュー・マネージャーに対する `MQSC` コマンドを処理します。

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

タスクの結果

これで、別々のコンテナでローカル・アプリケーションが実行されるようになりました。`dspsmq`、`amqsput`、`amqsget`、`runmqsc` などのコマンドを、QM1 キュー・マネージャーへのローカル・バインディングとして、2つ目のコンテナから正常に実行することができます。

予期した結果にならない場合は、[213 ページの『名前空間アプリケーションのトラブルシューティング』](#)で詳細を参照してください。

Multi 名前空間アプリケーションのトラブルシューティング

共有の名前空間を使用する場合は、すべての名前空間 (IPC、PID、UTS/ホスト名) およびマウント・ボリュームを共有する必要があります。そうしないと、アプリケーションは機能しません。

従う必要がある制約事項のリストについては、[212 ページの『別々のコンテナでのローカル・バインディング・アプリケーションの実行』](#)を参照してください。

リストされているすべての制約事項をアプリケーションが満たしていないと、コンテナが始動しても、予期したように機能しないという問題が発生する可能性があります。

以下のリストに、一般的な原因と、制約事項のいずれかを満たすことを忘れた場合に見られる動作をまとめています。

- コンテナの名前空間 (UTS/PID/IPC)、またはホスト名のいずれかを共有することを忘れた状態でボリュームをマウントした場合、コンテナは、キュー・マネージャーを認識できますが、キュー・マネージャーと対話することができません。
 - **dspmq** コマンドの場合は、以下のようになります。

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

- **runmqsc** コマンドなど、キュー・マネージャーに接続しようとするコマンドの場合は、AMQ8146 エラー・メッセージを受け取るはずですが。

```
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
```

- すべての必要な名前空間を共有しているが、共有ボリュームを `/var/mqm` ディレクトリにマウントせず、有効な IBM MQ データ・パスを持っている場合は、コマンドも アンキ 8146 のエラー・メッセージを受け取ります。

しかし、**dspmq** は、キュー・マネージャーをまったく認識できないので、代わりに空の応答を返します。

```
docker exec container dspmq
```

- 必要なすべての名前空間を共有しているが、共有ボリュームを `/var/mqm` ディレクトリにマウントせず、有効な IBM MQ データ・パス (または IBM MQ データ・パスなし) を持っていない場合は、データ・パスが IBM MQ インストールのキー・コンポーネントであるため、さまざまなエラーが表示されます。データ・パスがなければ、IBM MQ は動作できません。

以下のコマンドのいずれかを実行したときに、これらの例に示すような応答が表示される場合は、ディレクトリをマウントしたこと、または IBM MQ データ・ディレクトリを作成したことを確認する必要があります。

```
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 dlmqm QM1
AMQ7002: An error occurred manipulating a file.

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

MQ Adv. 独自のコンテナを作成する場合のネイティブ HA グループの作成

ネイティブ HA グループを作成するには、3つのキュー・マネージャーを作成、構成、および開始する必要があります。

このタスクについて

ネイティブ HA ソリューションを作成するために推奨される方法は、IBM MQ 演算子を使用することです (ネイティブ HA を参照)。あるいは、独自のコンテナを作成する場合は、以下の手順に従うことができます。

ネイティブ HA グループを作成するには、ログ・タイプを `log replication` に設定して、3つのノードに3つのキュー・マネージャーを作成します。次に、各キュー・マネージャーの `qm.ini` ファイルを編集して、3つのノードのそれぞれの接続の詳細を追加し、ログ・データを相互に複製できるようにします。

その後、3つすべてのキュー・マネージャーを開始して、3つすべてのインスタンスが相互に通信できることを確認し、どのインスタンスがアクティブ・インスタンスになり、どのインスタンスがレプリカになるかを判別できるようにする必要があります。

注: この方法で独自のコンテナにネイティブ HA グループを作成できるのは、Kubernetes または Red Hat OpenShift を実行している場合のみです。

手順

- 3つのノードのそれぞれで、ログ・レプリカのログ・タイプを指定し、各ログ・インスタンスに固有の名前を指定して、キュー・マネージャーを作成します。各キュー・マネージャーの名前は同じです。

```
crtmqm -lr instance_name qmname
```

以下に例を示します。

```
node 1> crtmqm -lr qm1_inst1 qm1
node 2> crtmqm -lr qm1_inst2 qm1
node 3> crtmqm -lr qm1_inst3 qm1
```

- 各キュー・マネージャーが正常に作成されると、`NativeHALocalInstance` という名前のスタンザがキュー・マネージャー構成ファイル `qm.ini` に追加されます。指定されたインスタンス名を指定する `Name` 属性がスタンザに追加されます。

オプションで、以下の属性を `qm.ini` ファイルの `NativeHALocalInstance` スタンザに追加できます。

KeyRepository

ログ複製トラフィックの保護のために、使用するデジタル証明書を保持する鍵リポジトリのローケーション。位置は語幹形式で指定されます。つまり、拡張子なしの絶対パスとファイル名が含まれます。KeyRepository スタンザ属性を省略すると、インスタンス間でログ複製データがプレーン・テキストで交換されます。

CertificateLabel

ログ複製トラフィックの保護に使用するデジタル証明書を識別する証明書ラベル。
KeyRepository が指定されているが、CertificateLabel が省略されている場合は、デフォルト値 `ibmwebsphermqueue_manager` が使用されます。

CipherSpec

ログ複製トラフィックを保護するために使用する MQCipherSpec。このスタンザ属性を指定する場合は、KeyRepository も指定する必要があります。KeyRepository が指定されているが、CipherSpec が省略されている場合は、デフォルト値 ANY が使用されます。

LocalAddress

ログ複製トラフィックを受け入れるローカル・ネットワーク・インターフェース・アドレス。このスタンザ属性が指定されている場合は、「[addr] [(port)]」の形式を使用してローカル・ネットワーク・インターフェースまたはポート (あるいはその両方) を識別します。ネットワーク・アドレスは、ホスト名、IPv4 小数点付き 10 進数、または IPv6 16 進形式で指定できます。この属性を省略すると、キュー・マネージャーはすべてのネットワーク・インターフェースへのバインドを試行し、ローカル・インスタンス名と一致する NativeHAInstances スタンザの ReplicationAddress に指定されているポートを使用します。

HeartbeatInterval

ハートビート間隔は、ネイティブ HA キュー・マネージャーのアクティブ・インスタンスがネットワーク・ハートビートを送信する頻度をミリ秒単位で定義します。ハートビート間隔値の有効範囲は 500 (0.5 秒) から 60000 (1 分) で、この範囲外の値を使用するとキュー・マネージャーの開始が失敗します。この属性を省略すると、デフォルト値の 5000 (5 秒) が使用されます。各インスタンスで同じハートビート間隔を使用する必要があります。

HeartbeatTimeout

ハートビート・タイムアウトは、ネイティブ HA キュー・マネージャーのレプリカ・インスタンスが、アクティブ・インスタンスが応答しないと判断するまで待機する時間の長さを定義します。ハートビート間隔タイムアウト値の有効範囲は 500 (0.5 秒) から 120000 (2 分) です。ハートビート・タイムアウトの値は、ハートビート間隔以上でなければなりません。

無効な値を指定すると、キュー・マネージャーの開始が失敗します。この属性が省略されると、レプリカは、新しいアクティブ・インスタンスを選択するプロセスを開始する前に 2 x HeartbeatInterval 待機します。各インスタンスで同じハートビート・タイムアウトを使用する必要があります。

RetryInterval

再試行間隔は、障害が発生した複製リンクがネイティブ HA キュー・マネージャーで再試行される頻度をミリ秒単位で定義します。再試行間隔の有効範囲は 500 (0.5 秒) から 120000 (2 分) です。この属性が省略されると、レプリカは、障害が発生した複製リンクを再試行する前に 2 x HeartbeatInterval 待機します。

- 各キュー・マネージャーの `qm.ini` ファイルを編集し、接続の詳細を追加します。ネイティブ HA グループ内のキュー・マネージャー・インスタンス (ローカル・インスタンスを含む) ごとに 1 つずつ、合計 3 つの NativeHAInstance スタンザを追加します。以下の属性を追加します。

名前

キュー・マネージャー・インスタンスの作成時に使用したインスタンス名を指定します。

ReplicationAddress

インスタンスのホスト名、IPv4 ドット 10 進または IPv6 16 進形式のアドレスを指定します。アドレスは、ホスト名、IPv4 小数点付き 10 進数、または IPv6 16 進形式のアドレスとして指定できます。複製アドレスは、グループ内の各インスタンスから解決可能かつルーティング可能でなければなりません。ログ複製に使用するポート番号は、大括弧で囲んで指定する必要があります。以下に例を示します。

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

注: NativeHAInstance スタンザはすべてのインスタンスで同じであり、自動構成 (`crtmqm -ii`) を使用して提供できます。

- 以下の 3 つのインスタンスのそれぞれを開始します。

```
strmqm QMgrName
```

インスタンスが開始されると、3つのインスタンスがすべて実行中であることを確認するために通信が行われます。次に、3つのインスタンスのうちどちらがアクティブ・インスタンスであるかを決定し、残りの2つのインスタンスは引き続きレプリカとして実行されます。

例

以下の例は、3つのインスタンスのいずれかに必要なネイティブ HA の詳細を指定する `qm.ini` ファイルのセクションを示しています。

```
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. ネイティブ HA キュー・マネージャーの独自ローリング更新を実行する場合の考慮事項

ネイティブ HA キュー・マネージャーの IBM MQ バージョンまたはポッド仕様を更新する場合は、キュー・マネージャー・インスタンスのローリング更新を実行する必要があります。これは IBM MQ Operator によって自動的に処理されますが、独自のデプロイメント・コードを作成する場合は、いくつかの重要な考慮事項があります。

注: サンプル Helm チャートには、ローリング更新を実行するためのシェル・スクリプトが含まれていますが、このスクリプトは、このトピックにおける考慮事項には対応していないため、実動での使用には適していません。

Kubernetes Kubernetes では、StatefulSet リソースは、順序付けられた始動およびローリング更新を管理するために使用されます。始動手順の一環として、各ポッドを個別に開始し、そのポッドが作動可能になるまで待ってから、次のポッドに移動します。すべてのポッドを開始してリーダー選出を実行できるようにするため、これはネイティブ HA では機能しません。したがって、`.spec.podManagementPolicy` フィールド (StatefulSet 上) を `Parallel` に設定する必要があります。これはすべてのポッドが並行して更新されることも意味しますが、これは特に望ましくありません。このため、StatefulSet は `OnDelete` 更新方針も使用する必要があります。

StatefulSet ローリング更新コードを使用できないため、カスタム・ローリング更新コードが必要になります。以下の点を考慮してください。

- 一般的なローリング更新手順
- 最適な順序でポッドを更新することによるダウン時間の最小化
- クラスター状態の変更を処理
- エラーの処理
- タイミングの問題を処理

一般的なローリング更新手順

ローリング更新コードは、各インスタンスが REPLICIA の状況を `dspmqr` から示すまで待ちます。つまり、インスタンスは何らかのレベルの始動を行った (例えば、コンテナが開始され、MQ プロセスが実行されている) が、必ずしもその他のインスタンスと対話するようにはなっていません。例えば、ポッド A が再始動され、REPLICIA 状態になるとすぐにポッド B が再始動されます。ポッド B は新しい構成で開始され

ると、ポッド A と対話できるようになり、クォーラムを形成できます。A または B のいずれかが新しいアクティブ・インスタンスになります。

その一環として、各ポッドが REPLICIA 状態になった後に遅延を設定して、そのポッドがピアに接続してクォーラムを確立できるようにすると便利です。

最適な順序でポッドを更新することによるダウン時間の最小化

ローリング更新コードは、既知のエラー状態にあるポッドから、正常に開始されなかったポッドまで、一度に 1 つずつポッドを削除する必要があります。通常、アクティブなキュー・マネージャー・ポッドは最後に更新する必要があります。

また、前回の更新でポッドが既知のエラー状態になった場合は、ポッドの削除を一時停止することも重要です。これにより、すべてのポッドにわたって、中断された更新がロールアウトされなくなります。例えば、これが行われる可能性があるのは、アクセスできない(または誤植が含まれる)新しいコンテナ・イメージを使用するようにポッドが更新される場合です。

クラスター状態の変更を処理

ローリング更新コードは、クラスター状態のリアルタイム変更に適切に対応する必要があります。例えば、キュー・マネージャーのポッドの 1 つが、ノード・リブートまたはノード圧力が原因で排除されることがあります。排除されたポッドは、クラスターがビジーだと即時に再スケジュールされない可能性があります。この場合、ローリング更新コードは、他のすべてのポッドを再始動する前に適切に待機する必要があります。

エラーの処理

ローリング更新コードは、Kubernetes API の呼び出し時や他の予期しないクラスター動作時の障害に対して堅牢でなければなりません。

さらに、ローリング更新コード自体は、再始動を許容できなければなりません。ローリング更新は長時間実行される可能性があり、コードの再始動が必要になる可能性があります。

タイミングの問題を処理

ローリング更新コードは、ポッドが再始動したことを確認できるように、ポッドの更新改訂を検査する必要があります。これにより、ポッドが「開始済み」であることを示していても実際はまだ終了していないというタイミングの問題が回避されます。

関連概念

5 ページの『[コンテナ内の IBM MQ の使用方法の選択](#)』

コンテナ内の IBM MQ の使用方法としては、いくつかの選択肢があります。プリパッケージされているコンテナ・イメージを使用する IBM MQ Operator を使用することも、独自のイメージとデプロイメント・コードをビルドすることもできます。

MQ Adv. カスタムビルト・コンテナのネイティブ HA キュー・マネージャーの状況の表示

カスタムビルト・コンテナの場合、**dspmq** コマンドを使用してネイティブ HA インスタンスの状況を表示できます。

このタスクについて

dspmq コマンドを使用して、ノード上のキュー・マネージャー・インスタンスの操作状況を表示できます。返される情報は、インスタンスがアクティブとレプリカのどちらであるかに応じて異なります。アクティブ・インスタンスで提供される情報が確定的なもので、レプリカ・ノードからの情報は古くなっている可能性があります。

以下のアクションを実行できます。

- 現行ノード上のキュー・マネージャー・インスタンスがアクティブかレプリカを表示します。
- 現行ノード上のインスタンスのネイティブ HA の運用状況を表示します。
- ネイティブ HA 構成に属する 3 つのインスタンスすべての運用状況を表示します。

以下の状況フィールドが、ネイティブ HA 構成状況の報告に使用されます。

ROLE

これは、現行インスタンス・ロールを指定します。これは、Active、Replica、または Unknown のいずれかです。

INSTANCE

このキュー・マネージャー・インスタンスの作成時に **crtmqm** コマンドの **-lr** オプションを使用してこのキュー・マネージャー・インスタンスに対して指定された名前。

INSYNC

必要な場合にインスタンスがアクティブ・インスタンスとしてテークオーバーできるかどうかを示します。

QUORUM

クォラムの状況を *number_of_instances_in-sync/number_of_instances_configured* という形式でレポートします。

REPLADDR

キュー・マネージャー・インスタンスの複製アドレス。

CONNECTV

ノードがアクティブ・インスタンスに接続されているかどうかを示します。

BACKLOG

このインスタンスがどれだけ遅れているかを KB 数で示します。

CONNINST

指定されたインスタンスがこのインスタンスに接続されているかどうかを示します。

ALTDATE

この情報が最後に更新された日付を示します (更新されたことがない場合には空白)。

ALTTIME

この情報が最後に更新された時刻を示します (更新されたことがない場合には空白)。

手順

- キュー・マネージャー・インスタンスがアクティブ・インスタンスとして実行されているか、それともレプリカとして実行されているか判別するには、次のようにします

```
dspmqr -o status -m QMgrName
```

BOB という名前のキュー・マネージャーのアクティブ・インスタンスからは、次の状況が報告されます

```
QMNAME(BOB)           STATUS(Running)
```

BOB という名前のキュー・マネージャーのレプリカ・インスタンスからは、次の状況が報告されます

```
QMNAME(BOB)           STATUS(Replica)
```

非アクティブ・インスタンスからは、次の状況が報告されます

```
QMNAME(BOB)           STATUS(Ended Immediately)
```

- 現行ノード上のインスタンスのネイティブ HA 運用状況を判別するには、以下のようになります。

```
dspmqr -o nativeha -m QMgrName
```

BOB という名前のキュー・マネージャーのアクティブ・インスタンスからは、次のような状況が報告されます

```
QMNAME(BOB)          ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
```

BOB という名前のキュー・マネージャーのレプリカ・インスタンスからは、次のような状況が報告されます

```
QMNAME(BOB)          ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
```

BOB という名前のキュー・マネージャーの非アクティブ・インスタンスからは、次のような状況が報告されます

```
QMNAME(BOB)          ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
```

- ネイティブ HA 構成内のすべてのインスタンスのネイティブ HA 運用状況を判別するには、次のようにします

```
dspmqr -o nativeha -x -m QMgrName
```

キュー・マネージャー BOB のアクティブ・インスタンスを実行しているノード上でこのコマンドを発行すると、以下のような状況が表示されます

```
QMNAME(BOB)          ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

キュー・マネージャー BOB のレプリカ・インスタンスを実行しているノード上でこのコマンドを発行すると、以下のような状況が表示されます。これは、レプリカの 1 つで処理が遅れていることを示しています

```
QMNAME(BOB)          ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(No) BACKLOG(435)
CONNINST(Yes) ALTDATA(2022-01-12) ALTTIME(12.03.44)
```

キュー・マネージャー BOB の非アクティブ・インスタンスを実行しているノード上でこのコマンドを発行すると、以下のような状況が表示されます

```
QMNAME(BOB)          ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
INSTANCE(inst1) ROLE(Unknown) REPLADDR(9.20.123.45) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
INSTANCE(inst2) ROLE(Unknown) REPLADDR(9.20.123.46) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
INSTANCE(inst3) ROLE(Unknown) REPLADDR(9.20.123.47) CONNACTV(No) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATA() ALTTIME()
```

どのインスタンスがアクティブでどれがレプリカになるかをまだネゴシエーションしている間にコマンドを発行すると、次の状況が表示されます

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

関連資料

[dspmqr \(キュー・マネージャーの表示\) コマンド](#)

MQ Adv. ネイティブ HA キュー・マネージャーの終了

endmqm コマンドを使用して、ネイティブ HA グループの一部であるアクティブ・キュー・マネージャーまたはレプリカ・キュー・マネージャーを終了できます。

手順

- キュー・マネージャーのアクティブ・インスタンスを終了するには、この資料の「構成」セクションの「[ネイティブ HA キュー・マネージャーの終了](#)」を参照してください。

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

IBM 本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒 103-8510

東京都中央区日本橋箱崎町 19 番 21 号

日本アイ・ビー・エム株式会社

日本アイ・ビー・エム株式会社

法務・知的財産

U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing

Legal and Intellectual Property Law

〒 103-8510

19-21, Nihonbashi-Hakozakicho, Chuo-ku

Tokyo 103-8510, Japan

以下の保証は、国または地域の法律に沿わない場合は、適用されません。 INTERNATIONAL BUSINESS MACHINES CORPORATION は、法律上の瑕疵担保責任、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。"" 国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

東京都中央区日本橋箱崎町 19 番 21 号

日本アイ・ビー・エム株式会社

Software Interoperability Coordinator, Department 49XA

3605 Highway 52 N

Rochester, MN 55901

U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っていません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

プログラミング・インターフェース情報

プログラミング・インターフェース情報 (提供されている場合) は、このプログラムで使用するアプリケーション・ソフトウェアの作成を支援することを目的としています。

本書には、プログラムを作成するユーザーが WebSphere MQ のサービスを使用するためのプログラミング・インターフェースに関する情報が記載されています。

ただし、この情報には、診断、修正、および調整情報が含まれている場合があります。診断、修正、調整情報は、お客様のアプリケーション・ソフトウェアのデバッグ支援のために提供されています。

重要: この診断、修正、およびチューニング情報は、変更される可能性があるため、プログラミング・インターフェースとして使用しないでください。

商標

IBM、IBM ロゴ、ibm.com® は、世界の多くの国で登録された IBM Corporation の商標です。現時点での IBM の商標リストについては、"Copyright and trademark information" www.ibm.com/legal/copytrade.shtml をご覧ください。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。

Microsoft および Windows は、Microsoft Corporation の米国およびその他の国における商標です。

UNIX は The Open Group の米国およびその他の国における登録商標です。

Linux は、Linus Torvalds 氏の米国およびその他の国における登録商標です。

この製品には、Eclipse Project (<https://www.eclipse.org/>) により開発されたソフトウェアが含まれています。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。



部品番号:

(1P) P/N: