

9.1

IBM MQ dans des conteneurs

IBM

Remarque

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section [«Remarques»](#), à la page 55.

Cette édition s'applique à la version 9 édition 1 d' IBM® MQ et à toutes les éditions et modifications ultérieures, sauf indication contraire dans les nouvelles éditions.

Lorsque vous envoyez des informations à IBM, vous accordez à IBM le droit non exclusif d'utiliser ou de distribuer les informations de la manière qu'il juge appropriée, sans aucune obligation de votre part.

© **Copyright International Business Machines Corporation 2007, 2024.**

Table des matières

IBM MQ dans des conteneurs.....	5
Planification d'IBM MQ dans des conteneurs.....	5
Comment utiliser IBM MQ dans des conteneurs.....	5
Prise en charge des conteneurs certifiés d'IBM MQ.....	6
Prise en charge de la génération de vos propres images de conteneur IBM MQ et de vos propres chartes.....	8
Remarques sur le stockage pour le IBM MQ Advanced certified container.....	9
Haute disponibilité pour le IBM MQ Advanced certified container.....	10
Authentification et autorisation des utilisateurs pour le IBM MQ Advanced certified container.....	12
Installation et désinstallation d'IBM MQ Operator sous OpenShift.....	12
Installation d'IBM MQ Operator à l'aide de la console Web OpenShift.....	13
Installation d'IBM MQ Operator à l'aide de l'interface de ligne de commande OpenShift.....	14
Déploiement de conteneurs certifiés IBM MQ.....	16
Préparation de votre projet OpenShift pour IBM MQ à l'aide de l'interface de ligne de commande OpenShift.....	16
Déploiement d'un gestionnaire de files d'attente à l'aide d'IBM Cloud Pak for Integration Platform Navigator.....	17
Déploiement d'un gestionnaire de files d'attente à l'aide de la console Web OpenShift.....	18
Déploiement d'un gestionnaire de files d'attente à l'aide de l'interface de ligne de commande OpenShift.....	19
Intégration au tableau de bord des opérations d'IBM Cloud Pak for Integration.....	21
Génération d'une image avec des fichiers MQSC et INI personnalisés à l'aide de l'interface de ligne de commande OpenShift.....	22
Déploiement de conteneurs certifiés IBM MQ à l'aide de Helm.....	24
Déploiement d'éditions CD précédentes d'IBM MQ dans un cluster IBM Cloud Private.....	28
Ajout d'éditions CD précédentes d'une image d'IBM MQ dans un cluster IBM Cloud Private.....	29
Ajout d'éditions CD précédentes d'une image d'IBM MQ dans un cluster IBM Cloud Kubernetes Service.....	30
Connexion à un gestionnaire de files d'attente déployé dans un cluster OpenShift.....	31
Connexion à IBM MQ Console dans un cluster OpenShift.....	33
Sauvegarde et restauration de la configuration de gestionnaire de files d'attente à l'aide de l'interface de ligne de commande OpenShift.....	34
Génération de votre propre conteneur IBM MQ.....	35
Planification de votre propre image de gestionnaire de files d'attente IBM MQ à l'aide d'un conteneur.....	35
Génération d'une image de gestionnaire de files d'attente IBM MQ avec Docker.....	36
Exécution d'applications de liaison locale dans des conteneurs distincts.....	39
Référence d'API pour IBM MQ Operator.....	41
Référence d'API pour mq.ibm.com/v1beta1.....	41
Remarques.....	55
Documentation sur l'interface de programmation.....	56
Marques.....	56

Les conteneurs permettent de conditionner un gestionnaire de files d'attente IBM MQ ou une application client IBM MQ avec toutes ses dépendances dans une unité normalisée pour le développement de logiciels.

Vous pouvez exécuter IBM MQ dans le conteneur préconditionné fourni dans IBM MQ Advanced et IBM MQ Advanced for Developers. Ce IBM MQ Advanced certified container met à disposition une image et une charte Helm prises en charge et peut être utilisé pour déployer une image d'IBM MQ prête pour la production sur Red Hat® OpenShift®, dans IBM Cloud Private ou dans IBM Cloud Kubernetes Service.

Vous pouvez aussi exécuter IBM MQ dans un conteneur IBM Cloud Pak for Integration ou dans un conteneur que vous générez.

  Pour plus d'informations sur le IBM MQ Advanced certified container, voir les liens suivants :

Planification d'IBM MQ dans des conteneurs

Lors de la planification d'IBM MQ dans des conteneurs, prenez en compte le support fourni par IBM MQ pour diverses options d'architecture, par exemple la façon dont la haute disponibilité est gérée et la manière de sécuriser vos gestionnaires de files d'attente.

Pourquoi et quand exécuter cette tâche

Avant de planifier l'utilisation d'IBM MQ dans une architecture de conteneurs, familiarisez-vous avec les concepts de base d'IBM MQ (voir [Présentation technique d'IBM MQ](#)) et les concepts de base de Kubernetes/OpenShift (voir [OpenShift Container Platform architecture](#)).

Procédure

- [«Comment utiliser IBM MQ dans des conteneurs»](#), à la page 5.
- [«Haute disponibilité pour le IBM MQ Advanced certified container»](#), à la page 10.
- [«Authentification et autorisation des utilisateurs pour le IBM MQ Advanced certified container»](#), à la page 12.

Comment utiliser IBM MQ dans des conteneurs

Il existe plusieurs façons d'utiliser IBM MQ dans des conteneurs : vous pouvez choisir d'utiliser des conteneurs certifiés préconditionnés ou générer vos propres images et votre propre code de déploiement.

Utilisation de conteneurs certifiés IBM MQ Advanced

Si vous prévoyez un déploiement sous Red Hat OpenShift Container Platform, vous utiliserez certainement des conteneurs certifiés. Il existe trois variétés de conteneur certifié :

- IBM MQ Advanced certified container pour IBM Cloud Pak for Integration. Il s'agit d'un produit distinct d'IBM qui inclut une version d'un certificat certifié.
- IBM MQ Advanced certified container
- Conteneur certifié IBM MQ Advanced for Developers (non garanti)

IBM MQ 9.1.4 et les éditions CD antérieures étaient également prises en charge sous IBM Cloud Private et IBM Cloud Kubernetes Service.

Notez que les conteneurs certifiés évoluent rapidement et que par conséquent, ils ne sont pris en charge que dans les éditions [Continuous Delivery](#).

Les conteneurs certifiés incluent des images de conteneur préconfigurées, ainsi que le code de déploiement pour l'exécution sur Red Hat OpenShift Container Platform. A partir de IBM MQ 9.1.5, les gestionnaires de files d'attente sont gérés à l'aide d'un opérateur IBM MQ. Les versions antérieures d'IBM MQ, jusqu'à la version 9.1.5 incluse, sont gérées à l'aide de chartes Helm.

Certaines fonctions d'IBM MQ ne sont pas prises en charge lors de l'utilisation de conteneurs certifiés. Vous devez générer vos propres images et vos propres chartes si vous voulez effectuer les opérations suivantes :

- Utiliser les API REST pour l'administration ou la messagerie
- Utiliser l'un des composants MQ suivants :
 - Des agents Managed File Transfer et leurs ressources. Toutefois, vous pouvez utiliser les conteneurs certifiés pour fournir un ou plusieurs gestionnaires de files d'attente de coordination, de commande ou d'agent.
 - AMQP
 - IBM MQ Bridge to Salesforce
 - IBM MQ Bridge to blockchain (non pris en charge dans les conteneurs)
- Utilisez le serveur Web lors du déploiement à l'aide de chartes Helm (sauf pour IBM Cloud Pak for Integration)
- Personnaliser les options utilisées avec `crtmqm`, `strmqm` et `endmqm`, par exemple pour la configuration des journaux de reprise

Génération de vos propres images et de vos propres chartes

Il s'agit de la solution de conteneur la plus souple, qui exige toutefois de solides compétences relatives à la configuration des conteneurs et qui requiert que vous "possédiez" le conteneur résultant. Si vous ne prévoyez pas d'utiliser Red Hat OpenShift Container Platform, vous devez générer vos propres images et votre propre code de déploiement.

Des exemples de génération d'images sont disponibles. Voir [«Génération de votre propre conteneur IBM MQ»](#), à la page 35. Les chartes Helm fournies avec les conteneurs certifiés sont publiées sur GitHub et peuvent être utilisées comme exemples lorsque vous générez vos propres images :

- [Charte Helm pour le IBM MQ Advanced certified container](#)
- [Charte Helm pour le conteneur certifié IBM MQ Advanced for Developers](#)

Concepts associés

[«Prise en charge des conteneurs certifiés d'IBM MQ»](#), à la page 6

Les conteneurs certifiés d'IBM MQ ne sont pris en charge que dans certains environnements Kubernetes.

[«Prise en charge de la génération de vos propres images de conteneur IBM MQ et de vos propres chartes»](#), à la page 8

Informations à prendre en compte si vous utilisez des conteneurs sur un système Linux.

Linux

Prise en charge des conteneurs certifiés d'IBM MQ

Les conteneurs certifiés d'IBM MQ ne sont pris en charge que dans certains environnements Kubernetes.

 Pour l'édition CD versions 9.1.4 et ultérieures, le IBM MQ Advanced certified container est pris en charge pour une utilisation avec Red Hat OpenShift. Voir [«Déploiement d'un gestionnaire de files d'attente à l'aide de l'interface de ligne de commande Helm»](#), à la page 25.

Les éditions CD antérieures à la version 9.1.4 sont prises en charge dans les environnements Kubernetes suivants :

- IBM Cloud Kubernetes Service
- IBM Cloud Private
- IBM Cloud Private avec Red Hat OpenShift

Pour les versions prises en charge spécifiques de Kubernetes, voir les fichiers `qualification.yaml` et `Chart.yaml` dans une charte Helm IBM MQ Advanced téléchargée. Ces versions varient en fonction de l'édition.

IBM MQ Advanced certified container est pris en charge uniquement lorsqu'il est déployé à l'aide de l'opérateur IBM MQ ou lorsqu'il utilise l'une des chartes Helm suivantes:

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

Remarque : L'utilisation des chartes Helm est obsolète, à la suite de l'édition de l'opérateur IBM MQ .

Etant donné que la technologie des conteneurs évolue rapidement, le IBM MQ Advanced certified container est pris en charge uniquement dans la version la plus récente des plateformes prises en charge par cette charte au moment de sa publication. Si vous voulez utiliser une version de plateforme plus ancienne, il peut être nécessaire d'utiliser une version plus ancienne du IBM MQ Advanced certified container.

L'image IBM MQ Advanced certified container est basée sur les éditions IBM MQ Continuous Delivery (CD). Les deux éditions CD les plus récentes sont prises en charge, ou toute édition CD de moins d'un an (en fonction de la durée la plus longue). Les éditions Long Term Support de IBM MQ ne sont pas disponibles en tant que conteneur certifié.

Depuis le IBM MQ Advanced certified container version 4.0, l'image fournit une installation d'IBM MQ sur Red Hat Universal Base Image (UBI), qui inclut les bibliothèques et les utilitaires Linux principaux utilisés par IBM MQ. L'image UBI est prise en charge par Red Hat en cas d'exécution sur un hôte Red Hat Enterprise Linux. Les versions précédentes du IBM MQ Advanced certified container utilisaient une image de base Ubuntu non prise en charge.

Concepts associés

«Prise en charge de la [génération de vos propres images de conteneur IBM MQ et de vos propres chartes](#)», à la page 8

Informations à prendre en compte si vous utilisez des conteneurs sur un système Linux.

Linux > MQ Adv. > CD Versions prises en charge pour IBM MQ Advanced certified container

Ensemble de tableaux présentant le mappage entre les versions prises en charge du IBM MQ Advanced certified container, d'IBM MQ, d'IBM Cloud Kubernetes Service, d'IBM Cloud Pak for Integration et d'IBM Cloud Private.

Opérateur IBM MQ

V 9.1.5

L'opérateur IBM MQ est pris en charge pour une utilisation avec IBM Cloud Pak for Integration version 2020.2, ou indépendamment, avec IBM MQ version 9.1.5 et versions ultérieures.

L'opérateur IBM MQ est pris en charge sur Red Hat OpenShift Container Platform version 4.4 ou ultérieure.

IBM MQ Advanced certified container V 9.1.5 (charteHelm)-obsolète

Inclut la charte Helm `ibm-mqadvanced-server-prod`.

V 9.1.5 Depuis la version 5.0.x du IBM MQ Advanced certified container, la charte Helm, l'image et les correctifs sont livrés via IBM Entitled Catalog and Registry. Les versions antérieures étaient livrées via Passport Advantage et les éditions de correctif sont disponibles depuis IBM Fix Central.

Tableau 1. Prise en charge pour le IBM MQ Advanced certified container

Version	Version de IBM MQ	Fin de prise en charge	Plateformes prises en charge
6.0.x	9.1.5, édition Continuous Delivery	Mars 2021	Detailed System Requirements
5.0.x	9.1.4, édition Continuous Delivery	Décembre 2020	Detailed System Requirements
4.1.x	9.1.3, édition Continuous Delivery	Juillet 2020	Detailed System Requirements

Logiciel IBM MQ Advanced certified container pour la IBM Cloud Pak for Integration

V 9.1.5 (charteHelm)-obsolète

Inclut la charte Helm `ibm-mqadvanced-server-integration-prod`.

Tableau 2. Versions prises en charge pour le logiciel du IBM MQ Advanced certified container pour IBM Cloud Pak for Integration

Version	Version de IBM MQ	Version de IBM Cloud Pak for Integration
6.0.x	9.1.4, édition Continuous Delivery	2020.1.1 (configuration système requise)
5.0.x	9.1.3, édition Continuous Delivery	2019.4.1 (configuration système requise)
4.1.x	9.1.3, édition Continuous Delivery	2019.3.2.2 (configuration système requise)
4.0.x	9.1.3, édition Continuous Delivery	2019.3.2 (configuration système requise)
3.0.x	9.1.3, édition Continuous Delivery	2019.3.1 (configuration système requise)

Voir les notes sur l'édition d'IBM Cloud Pak for Integration pour des informations sur les versions prises en charge.

Linux Prise en charge de la génération de vos propres images de conteneur IBM MQ et de vos propres chartes

Informations à prendre en compte si vous utilisez des conteneurs sur un système Linux.

- L'image de base utilisée par l'image de conteneur doit utiliser un système d'exploitation Linux qui est pris en charge.
- Vous devez utiliser les programmes d'installation d'IBM MQ pour installer le produit dans l'image de conteneur.
- Pour la liste des packages pris en charge, voir [Composants rpm d'IBM MQ pour systèmes Linux](#).
- **V 9.1.0** Les packages suivants ne sont pas pris en charge :
 - MQSeriesBCBridge
 - MQSeriesRDQM

- Le répertoire de données (/var/mqm par défaut) du gestionnaire de files d'attente doit être stocké sur un volume de conteneur qui conserve un état persistant.

Important : Vous ne pouvez pas utiliser le système de fichiers union.

Vous devez monter un répertoire hôte comme volume de données ou utiliser un conteneur de volume de données. Pour plus d'informations, voir [Manage data in containers](#).

- Vous devez pouvoir exécuter des commandes IBM MQ, telles que **endmqm**, dans le conteneur.
- Vous pouvez pouvoir obtenir des fichiers et des répertoires à partir du conteneur à des fins de diagnostic.
- **V 9.1.0** Vous pouvez créer un espace de nom pour partager les espaces de nom du conteneur pour le gestionnaire de files d'attente avec d'autres conteneurs afin de lier localement des applications à un gestionnaire de files d'attente s'exécutant dans des conteneurs distincts. Pour plus d'informations, voir [«Exécution d'applications de liaison locale dans des conteneurs distincts»](#), à la page 39.

Concepts associés

«Prise en charge des conteneurs certifiés d'IBM MQ», à la page 6

Les conteneurs certifiés d'IBM MQ ne sont pris en charge que dans certains environnements Kubernetes.

Linux

MQ Adv.

CD

V 9.1.5

Remarques sur le stockage pour le IBM

MQ Advanced certified container

Le IBM MQ Advanced certified container peut être exécuté dans deux modes de stockage :

- Le **stockage éphémère** est utilisé lorsque toutes les informations relatives à l'état du conteneur peuvent être effacées lorsque le conteneur redémarre. En général, il est utilisé lorsque des environnements sont créés à des fins de démonstration ou lors d'un développement avec des gestionnaires de files d'attente autonomes.
- Le **stockage persistant** constitue la configuration courante pour IBM MQ et garantit que si le conteneur est redémarré, la configuration, les journaux et les messages persistants existants seront disponibles dans le conteneur redémarré.

IBM MQ Operator permet de personnaliser les caractéristiques de stockage qui peuvent différer considérablement selon l'environnement, ainsi que le mode de stockage souhaité.

Stockage éphémère

IBM MQ est une application avec état et elle conserve cet état dans le stockage en vue d'une reprise en cas de redémarrage. Si vous utilisez le stockage éphémère, toutes les informations relatives à l'état du gestionnaire de files d'attente sont perdues en cas de redémarrage. Seront perdus :

- Tous les messages
- Toutes les informations relatives à l'état des communications entre les gestionnaires de files d'attente (numéros de séquence des messages de canal)
- L'identité du cluster MQ du gestionnaire de files d'attente
- Toutes les informations relatives à l'état des transactions
- L'intégralité de la configuration du gestionnaire de files d'attente
- Toutes les données de diagnostic locales

Ainsi, vous devez déterminer si le stockage éphémère est approprié pour un scénario de production, de test ou de développement, par exemple lorsque tous les messages sont non persistants et que le gestionnaire de files d'attente n'est pas membre d'un cluster MQ. Toutes les informations relatives à l'état de la messagerie sont effacées au redémarrage, et la configuration du gestionnaire de files d'attente est également supprimée. Pour obtenir un conteneur intégralement éphémère, vous devez ajouter la configuration d'IBM MQ à l'image de conteneur (pour plus d'informations, voir [«Génération d'une image avec des fichiers MQSC et INI personnalisés à l'aide de l'interface de ligne de commande OpenShift»](#), à la page 22). Sinon, IBM MQ devra être configuré à chaque fois que le conteneur redémarre.

Par exemple, pour configurer IBM MQ avec un stockage éphémère, le type de stockage de QueueManager doit inclure les éléments suivants :

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

Stockage de persistance

Normalement, IBM MQ s'exécute avec un stockage persistant afin de garantir que les messages persistants et la configuration du gestionnaire de files d'attente sont conservés après un redémarrage. Il s'agit donc du comportement par défaut. Etant donné les divers fournisseurs de stockage et les différentes fonctions que chaque fournisseur prend en charge, cela signifie souvent qu'il est nécessaire de personnaliser la configuration. Vous trouverez ci-dessous les zones communes permettant de personnaliser la configuration de stockage de MQ dans l'API v1beta1 :

- [spec.queueManager.availability](#) contrôle le mode de disponibilité. Si vous utilisez `SingleInstance`, le stockage `ReadWriteOnce` est suffisant, alors que `MultiInstance` requiert une classe d'archivage qui prend en charge `ReadWriteMany` avec les caractéristiques de verrouillage de fichier appropriées. IBM MQ fournit une [déclaration de prise en charge](#) et une [déclaration de test](#). Le mode de disponibilité a également un impact sur la présentation des volumes persistants. Pour plus d'informations, voir «[Haute disponibilité pour le IBM MQ Advanced certified container](#)», à la page 10.
- [spec.queueManager.storage](#) contrôle les paramètres de stockage individuels. Un gestionnaire de files d'attente peut être configuré pour utiliser entre un et quatre volumes persistants.

L'exemple suivant est un fragment de configuration simple qui utilise un gestionnaire de files d'attente mono-instance :

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

L'exemple suivant est un fragment de configuration de gestionnaire de files d'attente multi-instance, qui présente une classe d'archivage autre que la classe d'archivage par défaut, ainsi qu'un stockage de fichiers nécessitant des groupes supplémentaires :

```
spec:
  queueManager:
    availability:
      type: MultiInstance
    storage:
      queueManager:
        enabled: true
        class: ibmc-file-gold-gid
        persistedData:
          enabled: true
          class: ibmc-file-gold-gid
        recoveryLogs:
          enabled: true
          class: ibmc-file-gold-gid
      securityContext:
        supplementalGroups: [99]
```

Linux MQ Adv. CD Haute disponibilité pour le IBM MQ Advanced certified container

Vous disposez de deux options principales pour la haute disponibilité avec le IBM MQ Advanced certified container : un **gestionnaire de files d'attente multi-instance** (paire de pods actif-secours utilisant un système de fichiers partagé en réseau) et un **gestionnaire de files d'attente résilient simple** (qui offre une approche simple pour la haute disponibilité en utilisant un stockage réseau).

Vous devez envisager la disponibilité des **messages** et la disponibilité des **services** séparément. Avec IBM MQ for Multiplatforms, un message est stocké dans un gestionnaire de files d'attente et un seul. Ainsi, si ce gestionnaire de files d'attente n'est plus disponible, vous perdez temporairement l'accès aux messages qu'il contient. Pour que les messages soient hautement disponibles, vous devez être capable de récupérer un gestionnaire de files d'attente aussi vite que possible. Vous pouvez assurer la disponibilité des services en créant plusieurs instances des files d'attente que les applications client pourront utiliser, par exemple à l'aide d'un cluster uniforme IBM MQ.

Vous pouvez considérer qu'un gestionnaire de files d'attente est composé de deux parties : les données stockées sur disque et les processus en cours d'exécution qui permettent d'accéder aux données. Vous pouvez déplacer tout gestionnaire de files d'attente sur un noeud Kubernetes différent, tant que le noeud conserve les mêmes données (fournies par des volumes Kubernetes persistants) et qu'il peut être associé à une adresse sur le réseau par les applications client. Dans Kubernetes, un service est utilisé pour fournir une identité réseau cohérente.

IBM MQ s'appuie sur la disponibilité des données sur les volumes persistants. Par conséquent, la disponibilité du stockage fournissant les volumes persistants est critique pour la disponibilité du gestionnaire de files d'attente, car IBM MQ ne peut pas être plus disponible que le stockage qu'il utilise. Si vous décidez de tolérer l'indisponibilité d'une zone de disponibilité entière, vous devez utiliser un fournisseur de volumes qui réplique les écritures sur disque dans une autre zone.

Gestionnaire de files d'attente multi-instance

Les gestionnaires de files d'attente multi-instance impliquent un pod **actif** et un pod **en veille** Kubernetes, qui s'exécutent en tant que partie d'un ensemble de statistiques Kubernetes avec exactement deux répliques et un ensemble de volumes persistants Kubernetes. Les données et les journaux des transactions du gestionnaire de files d'attente sont conservés sur deux volumes persistants à l'aide d'un système de fichiers partagé.

Les gestionnaires de files d'attente multi-instances exigent que le pod **actif** et le pod **de secours** disposent d'un accès simultané au volume persistant. Pour configurer cet accès, vous utilisez des volumes Kubernetes persistants pour lesquels le mode d'accès (paramètre **access mode**) est `ReadWriteMany`. Les volumes doivent également répondre aux IBM MQ exigences pour les systèmes de fichiers partagés, car IBM MQ s'appuie sur la libération automatique des verrous de fichier pour déclencher une reprise en ligne du gestionnaire de files d'attente. IBM MQ fournit une liste de systèmes de fichiers testés.

Les temps de reprise pour un gestionnaire de files d'attente multi-instance dépendent des facteurs suivants :

1. Le temps nécessaire au système de fichiers partagé, après un échec, pour libérer les verrous initialement placés par l'instance active.
2. Le temps nécessaire à l'instance de secours pour acquérir les verrous, puis démarrer.
3. Le temps nécessaire à la sonde de vigilance du pod Kubernetes pour détecter si le conteneur est prêt. Vous pouvez configurer ce paramètre dans la charte Helm.
4. Le temps nécessaire aux clients IBM MQ pour se reconnecter.

Gestionnaire de files d'attente résilient unique

Un gestionnaire de files d'attente résilient unique est une instance unique d'un gestionnaire de files d'attente qui s'exécute dans un pod Kubernetes unique, où Kubernetes surveille le gestionnaire de files d'attente et remplace le pod si nécessaire.

Les IBM MQ exigences pour les systèmes de fichiers partagés s'appliquent également lors de l'utilisation d'un seul gestionnaire de files d'attente résilient (sauf pour le verrouillage basé sur un bail), mais vous n'avez pas besoin d'utiliser un système de fichiers partagé. Vous pouvez utiliser le stockage par blocs, avec un système de fichiers adapté, comme `xf`s ou `ext4`.

Les temps de reprise pour un gestionnaire de files d'attente résilient unique dépendent des facteurs suivants :

1. Le temps d'exécution de la sonde de non-défaillance et le nombre d'échecs qu'elle tolère. Vous pouvez configurer ce paramètre dans la charte Helm.
2. Le temps nécessaire au planificateur Kubernetes pour replanifier le pod défectueux sur un nouveau noeud.
3. Le temps nécessaire pour télécharger l'image de conteneur sur le nouveau noeud. Si vous avez associé le paramètre **imagePullPolicy** à la valeur `IfNotPresent`, il se peut que l'image soit déjà disponible sur ce noeud.
4. Le temps nécessaire à la nouvelle instance de gestionnaire de files d'attente pour démarrer.
5. Le temps nécessaire à la sonde de vigilance du pod Kubernetes pour détecter si le conteneur est prêt. Vous pouvez configurer ce paramètre dans la charte Helm.
6. Le temps nécessaire aux clients IBM MQ pour se reconnecter.

Important :

Bien que le modèle de gestionnaire de files d'attente résilient unique présente certains avantages, vous devez déterminer si vous pouvez atteindre vos objectifs de disponibilité avec les limitations liées aux échecs de noeud.

Dans Kubernetes, un pod défectueux est généralement récupéré rapidement, mais l'échec d'un noeud entier est traité différemment. Si le noeud Kubernetes principal perd le contact avec un noeud worker, il ne peut pas déterminer si le noeud a échoué ou s'il a simplement perdu la connectivité au réseau. Ainsi, Kubernetes n'effectue **aucune action** dans ce cas, sauf si l'un des événements suivants survient :

1. Le noeud est restauré dans un état permettant la communication avec le noeud principal Kubernetes.
2. Une action d'administration est effectuée pour supprimer explicitement le pod sur le noeud Kubernetes principal. Elle n'arrête pas nécessairement l'exécution du pod, mais le supprime du magasin Kubernetes. Par conséquent, l'action d'administration doit être utilisée avec précaution.

Concepts associés

[Configurations à haute disponibilité](#)

Linux > MQ Adv. > CD **Authentification et autorisation des utilisateurs pour le IBM MQ Advanced certified container**

IBM MQ peut être configuré en vue de l'utilisation d'utilisateurs et de groupes LDAP pour l'autorisation. Cette approche est recommandée pour le IBM MQ Advanced certified container.

Dans un environnement conteneurisé à service partagé tel que Red Hat OpenShift Container Platform, des contraintes de sécurité sont mises en place pour éviter tout problème de sécurité. Par exemple, dans Red Hat OpenShift Container Platform, l'objet `SecurityContextConstraints` par défaut (appelé `restricted`) utilise un ID utilisateur aléatoire pour décourager l'utilisation d'un ID utilisateur local au conteneur. En général, IBM MQ utilise l'escalade de privilèges pour vérifier les mots de passe des utilisateurs, ce qui n'est pas non plus recommandé dans les environnements conteneurisés à service partagé. Pour ces raisons, l'utilisation d'utilisateurs définis dans les bibliothèques du système d'exploitation dans un conteneur en cours d'exécution n'est pas prise en charge dans les conteneurs certifiés IBM MQ.

Vous devez configurer votre gestionnaire de files d'attente afin qu'il utilise LDAP pour l'authentification et l'autorisation des utilisateurs. Pour des informations sur la configuration d'IBM MQ dans cette optique, voir [Connection authentication: User repositories](#) et [LDAP authorization](#).

Linux > MQ Adv. > CD > V 9.1.5 **Installation et désinstallation d'IBM MQ Operator sous OpenShift**

IBM MQ Operator peut être installé sous OpenShift à l'aide d'Operator Hub.

Avant de commencer

Procédure

- «Installation d'IBM MQ Operator à l'aide de l'interface de ligne de commande OpenShift», à la page [14](#).
- «Installation d'IBM MQ Operator à l'aide de la console Web OpenShift», à la page [13](#).

Linux > MQ Adv. > CD > V 9.1.5 Installation d'IBM MQ Operator à l'aide de la console Web OpenShift

IBM MQ Operator peut être installé sous OpenShift à l'aide d'Operator Hub.

Avant de commencer

Connectez-vous à la console Web de votre cluster OpenShift.

Procédure

1. Ajoutez les opérateurs IBM Common Services à la liste des opérateurs pouvant être installés.
 - a) Cliquez sur le signe plus. La boîte de dialogue **Import YAML** s'ouvre.
 - b) Collez la ressource de définition suivante dans la boîte de dialogue :

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

- c) Cliquez sur **Créer**.
2. Ajoutez les opérateurs IBM à la liste des opérateurs pouvant être installés.
 - a) Cliquez sur le signe plus. La boîte de dialogue **Import YAML** s'ouvre.
 - b) Collez la ressource de définition suivante dans la boîte de dialogue :

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

- c) Cliquez sur **Créer**.
3. Créer un espace de nom à utiliser pour IBM MQ Operator.

IBM MQ Operator peut être installé dans un espace de nom unique ou dans tous les espaces de nom. Cette étape n'est nécessaire que si vous voulez procéder à l'installation dans un espace de nom particulier qui n'existe pas encore.

 - a) Depuis le panneau de navigation, cliquez sur **Home > Projects**.
La page Projects s'affiche.
 - b) Cliquez sur **Create Project**. Une zone Create Project s'affiche.

- c) Entrez les détails de l'espace de nom que vous créez. Par exemple, vous pouvez spécifier le nom "ibm-mq".
 - d) Cliquez sur **Créer**. L'espace de nom pour IBM MQ Operator est créé.
4. Installez l'opérateur IBM MQ.
- a) Depuis le panneau de navigation, cliquez sur **Operators > OperatorHub**.
La page OperatorHub s'affiche.
 - b) Dans la zone **All Items**, entrez "IBM MQ".
L'entrée de catalogue IBM MQ est affichée.
 - c) Sélectionnez **IBM MQ**.
La fenêtre IBM MQ s'ouvre.
 - d) Cliquez sur **Install**.
La page Create Operator Subscription s'affiche.
 - e) Choisissez le mode d'installation : vous pouvez procéder à l'installation dans l'espace de nom spécifique que vous avez créé ou dans le cluster.
 - f) Cliquez sur **Subscribe**.
IBM MQ apparaît dans la page Installed Operators.
 - g) Cliquez sur le statut de l'opérateur dans la page Installed Operators ; il devient Succeeded une fois l'installation terminée.

Que faire ensuite

[«Déploiement de conteneurs certifiés IBM MQ», à la page 16](#)

Linux > MQ Adv. > CD > V 9.1.5 Installation d'IBM MQ Operator à l'aide de l'interface de ligne de commande OpenShift

IBM MQ Operator peut être installé sous OpenShift à l'aide d'Operator Hub.

Avant de commencer

Connectez-vous à l'interface de ligne de commande OpenShift avec la commande **oc login**. Pour pouvoir effectuer ces étapes, vous devez être un administrateur de cluster.

Procédure

1. Créez une ressource OperatorSource pour les opérateurs IBM Common Services.

- a) Créez un fichier YAML définissant la ressource OperatorSource

Créez un fichier nommé "operator-source-cs.yaml" dont le contenu est le suivant :

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

- b) Appliquez le OperatorSource au serveur.

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

2. Créez une ressource OperatorSource pour les opérateurs IBM.

a) Créez un fichier YAML définissant la ressource OperatorSource

Créez un fichier nommé "operator-source-ibm.yaml" dont le contenu est le suivant :

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

b) Appliquez le OperatorSource au serveur.

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

3. Créer un espace de nom à utiliser pour IBM MQ Operator.

IBM MQ Operator peut être installé dans un espace de nom unique ou dans tous les espaces de nom. Cette étape n'est nécessaire que si vous voulez procéder à l'installation dans un espace de nom particulier qui n'existe pas encore.

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

4. Affichez la liste des opérateurs disponibles dans le cluster depuis OperatorHub.

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

5. Inspectez IBM MQ Operator pour vérifier les modes d'installation qu'il prend en charge et les canaux disponibles.

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

6. Créer un fichier YAML d'objet OperatorGroup

Une OperatorGroup est une ressource OLM qui sélectionne les espaces de nom cible dans lesquels générer l'accès RBAC requis pour tous les opérateurs du même espace de nom que OperatorGroup.

L'espace de nom auquel vous abonnez l'opérateur doit comporter une ressource OperatorGroup qui correspond à la valeur InstallMode de l'opérateur, c'est-à-dire au mode AllNamespaces ou SingleNamespace. Si l'opérateur que vous prévoyez d'installer utilise le mode AllNamespaces, une ressource OperatorGroup appropriée existe déjà dans l'espace de nom openshift-operators.

Toutefois, si l'opérateur utilise le mode SingleNamespace et que vous ne disposez pas d'une ressource OperatorGroup appropriée, vous devez en créer une.

a) Créez un fichier nommé "mq-operator-group.yaml" dont le contenu est le suivant :

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

b) Création de l'objet OperatorGroup

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

7. Créez un fichier YAML pour un objet Subscription afin d'abonner un espace de nom à MQ Operator.

a) Créez un fichier nommé "mq-sub.yaml" dont le contenu est le suivant :

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

Pour l'utilisation de AllNamespaces **InstallMode** , spécifiez l'espace de nom openshift-operators . Sinon, indiquez l'espace de nom unique correspondant à l'utilisation de Espace de nom unique **InstallMode**.

b) Création de l'objet Subscription

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

8. Vérification du statut de l'opérateur

Une fois que l'installation de l'opérateur a abouti, l'état du pod s'affiche sous la forme *Exécution*. Pour l'utilisation de AllNamespaces **InstallMode**, indiquez **openshift-operators** comme espace de nom. Sinon, indiquez l'espace de nom unique correspondant à l'utilisation de Espace de nom unique **InstallMode**.

Que faire ensuite

[«Déploiement de conteneurs certifiés IBM MQ», à la page 16](#)

Linux

MQ Adv.

CD

Déploiement de conteneurs certifiés IBM

MQ

IBM MQ version 9.1.5 et versions ultérieures peuvent être déployés sur Red Hat OpenShift à l'aide de l'opérateur IBM MQ . Les versions IBM MQ 9.1.5 et 9.1.4 peuvent être déployées sur Red Hat OpenShift à l'aide de Helm. Les versions de CD antérieures peuvent être déployées sur un cluster IBM Cloud Private ou IBM Cloud Kubernetes Service , à l'aide de Helm.

Pourquoi et quand exécuter cette tâche

Procédure

- [«Déploiement d'un gestionnaire de files d'attente à l'aide de l'interface de ligne de commande Helm», à la page 25.](#)
- [«Déploiement d'éditions CD précédentes d'IBM MQ dans un cluster IBM Cloud Private», à la page 28.](#)
- [«Ajout d'éditions CD précédentes d'une image d'IBM MQ dans un cluster IBM Cloud Private», à la page 29.](#)
- [«Ajout d'éditions CD précédentes d'une image d'IBM MQ dans un cluster IBM Cloud Kubernetes Service», à la page 30.](#)

Linux

MQ Adv.

CD

Préparation de votre projet OpenShift pour IBM MQ à l'aide de l'interface de ligne de commande OpenShift

Préparez votre cluster Red Hat OpenShift Container Platform pour qu'il puisse déployer un gestionnaire de files d'attente à l'aide d'IBM MQ Operator. Cette tâche doit être effectuée par un administrateur de projet.

Avant de commencer

Remarque : Si vous prévoyez d'utiliser IBM MQ dans un projet dans lequel d'autres composants IBM Cloud Pak for Integration sont déjà installés, il n'est pas nécessaire de suivre les instructions ci-dessous.

Connectez-vous à votre cluster avec **cloudctl login** (pour IBM Cloud Pak for Integration) ou **oc login**.

Pourquoi et quand exécuter cette tâche

Les images de IBM MQ Advanced certified container sont extraites depuis un registre de conteneurs qui effectue un contrôle des autorisations de licence. Ce contrôle requiert une clé d'autorisation qui est stockée dans un secret d'extraction `docker-registry`. Si vous ne possédez pas encore de clé d'autorisation, suivez les instructions ci-après pour en obtenir une et créer un secret d'extraction.

Procédure

1. Obtenez la clé d'autorisation qui est affectée à votre ID.
 - a) Connectez-vous à [Mon IBM - Bibliothèque des logiciels de conteneur](#) avec l'ID et le mot de passe IBM associés au logiciel autorisé.
 - b) Dans la section **Clé d'autorisation**, cliquez sur **Copier la clé** pour copier la clé d'autorisation dans le presse-papiers.
2. Créez un secret contenant votre clé d'autorisation dans le projet dans lequel vous voulez déployer votre gestionnaire de files d'attente.

Exécutez la commande suivante, où `<entitlement-key>` est la clé extraite à l'étape 1, et `<user-email>` est l'ID IBM associé au logiciel autorisé.

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

Que faire ensuite

«Déploiement d'un gestionnaire de files d'attente à l'aide de l'interface de ligne de commande OpenShift», à la page 19

Linux > MQ Adv. > CD > V 9.1.5 Déploiement d'un gestionnaire de files d'attente à l'aide d'IBM Cloud Pak for Integration Platform Navigator

Utilisez la ressource personnalisée QueueManager pour déployer un gestionnaire de files d'attente dans un cluster Red Hat OpenShift Container Platform à l'aide d'IBM Cloud Pak for Integration Platform Navigator. Cette tâche doit être effectuée par un administrateur de projet.

Avant de commencer

Dans un navigateur, lancez IBM Cloud Pak for Integration Platform Navigator.

Si c'est la première fois qu'un gestionnaire de files d'attente est déployé dans ce projet Red Hat OpenShift, suivez les étapes présentées dans la rubrique [«Préparation de votre projet OpenShift pour IBM MQ à l'aide de l'interface de ligne de commande OpenShift»](#), à la page 16.

Procédure

1. Déployez un gestionnaire de files d'attente.

L'exemple ci-après déploie un gestionnaire de files d'attente "à démarrage rapide" qui utilise un stockage éphémère (non persistant) et désactive la sécurité MQ. Les messages ne seront pas conservés suite aux redémarrages du gestionnaire de files d'attente. Vous pouvez ajuster la configuration afin de changer de nombreux paramètres du gestionnaire de files d'attente.

- a) Dans IBM Cloud Pak for Integration Platform Navigator, cliquez sur **Runtime and instances**.
- b) Cliquez sur **Create instance**.
- c) Sélectionnez **Queue Manager**, puis cliquez sur **Next**.

Le formulaire de création d'instance d'une ressource QueueManager s'affiche.

Remarque : Vous pouvez aussi cliquer sur **Code** pour afficher ou changer le fichier YAML de configuration de la ressource QueueManager.

- d) Dans la section **Details**, vérifiez ou mettez à jour la zone **Name** et spécifiez dans la zone **Namespace** l'espace de nom dans lequel créer l'instance de gestionnaire de files d'attente.
- e) Si vous acceptez le contrat de licence d'IBM Cloud Pak for Integration, associez **License acceptance** à **On**.
Vous devez accepter la licence pour pouvoir déployer un gestionnaire de files d'attente.
- f) Dans la section **Queue Manager Config**, vérifiez ou mettez à jour la zone **Name** du gestionnaire de files d'attente sous-jacent.
Par défaut, le nom du gestionnaire de files d'attente utilisé par les applications client IBM MQ est identique au nom de la ressource QueueManager, sans les éventuels caractères non valides (tels que les traits d'union), qui sont retirés. Si vous voulez forcer l'utilisation d'un nom particulier, vous pouvez éditer le nom ici.
- g) Cliquez sur **Créer**.

La liste des gestionnaires de files d'attente qui se trouvent dans le projet (espace de nom) en cours est affichée. La nouvelle ressource QueueManager doit avoir le statut Pending.

2. Vérifiez que le gestionnaire de files d'attente est en cours d'exécution.

La création est terminée lorsque le statut de la ressource QueueManager est Running.

Tâches associées

«Connexion à un gestionnaire de files d'attente déployé dans un cluster OpenShift», à la page 31
Ensemble d'exemples de configuration pour la connexion à un gestionnaire de files d'attente déployé dans un cluster Red Hat OpenShift.

«Connexion à IBM MQ Console dans un cluster OpenShift», à la page 33
Connexion à la console IBM MQ Console d'un gestionnaire de files d'attente qui a été déployé dans un cluster Red Hat OpenShift Container Platform.

Linux > MQ Adv. > CD > V 9.1.5 Déploiement d'un gestionnaire de files d'attente à l'aide de la console Web OpenShift

Utilisez la ressource personnalisée QueueManager pour déployer un gestionnaire de files d'attente dans un cluster Red Hat OpenShift Container Platform à l'aide de la console Web Red Hat OpenShift. Cette tâche doit être effectuée par un administrateur de projet.

Avant de commencer

Connectez-vous à la console Web de votre cluster OpenShift. Vous devez sélectionner un projet (espace de nom) existant ou en créez un.

Si c'est la première fois qu'un gestionnaire de files d'attente est déployé dans ce projet Red Hat OpenShift, suivez les étapes présentées dans la rubrique «[Préparation de votre projet OpenShift pour IBM MQ à l'aide de l'interface de ligne de commande OpenShift](#)», à la page 16.

Procédure

1. Déployez un gestionnaire de files d'attente.

L'exemple ci-après déploie un gestionnaire de files d'attente "à démarrage rapide" qui utilise un stockage éphémère (non persistant) et désactive la sécurité MQ. Les messages ne seront pas conservés suite aux redémarrages du gestionnaire de files d'attente. Vous pouvez ajuster la configuration afin de changer de nombreux paramètres du gestionnaire de files d'attente.

- a) Dans la console Web OpenShift, depuis le panneau de navigation, cliquez sur **Operators > Installed Operators**.
 - b) Cliquez sur **IBM MQ**.
 - c) Cliquez sur l'onglet **Queue Manager**.
 - d) Cliquez sur le bouton **Create QueueManager**.
Un éditeur YAML contenant un exemple de fichier YAML pour une ressource QueueManager s'ouvre.
Remarque : Vous pouvez aussi cliquer sur **Edit Form** pour afficher ou changer la configuration de QueueManager.
 - e) Si vous acceptez le contrat de licence, associez **License acceptance** à **On**.
IBM MQ est disponible avec plusieurs licences. Pour plus d'informations sur les licences valides, voir [«Référence relative à l'octroi de licence pour mq.ibm.com/v1beta1»](#), à la page 41. Vous devez accepter la licence pour pouvoir déployer un gestionnaire de files d'attente.
 - f) Cliquez sur **Créer**.
La liste des gestionnaires de files d'attente qui se trouvent dans le projet (espace de nom) en cours est affichée. La nouvelle ressource QueueManager doit être à l'état Pending.
2. Vérifiez que le gestionnaire de files d'attente est en cours d'exécution.
La création est terminée lorsque le statut de la ressource QueueManager est Running.

Tâches associées

«Connexion à un gestionnaire de files d'attente déployé dans un cluster OpenShift», à la page 31
Ensemble d'exemples de configuration pour la connexion à un gestionnaire de files d'attente déployé dans un cluster Red Hat OpenShift.

«Connexion à IBM MQ Console dans un cluster OpenShift», à la page 33
Connexion à la console IBM MQ Console d'un gestionnaire de files d'attente qui a été déployé dans un cluster Red Hat OpenShift Container Platform.

Linux > MQ Adv. > CD > V 9.1.5 **Déploiement d'un gestionnaire de files d'attente à l'aide de l'interface de ligne de commande OpenShift**

Utilisez la ressource personnalisée QueueManager pour déployer un gestionnaire de files d'attente dans un cluster Red Hat OpenShift Container Platform à l'aide de l'interface de ligne de commande (CLI). Cette tâche doit être effectuée par un administrateur de projet.

Avant de commencer

Vous devez installer l'[interface de ligne de commande Red Hat OpenShift Container Platform](#).

Connectez-vous à votre cluster avec **cloudctl login** (pour IBM Cloud Pak for Integration) ou **oc login**.

Si c'est la première fois qu'un gestionnaire de files d'attente est déployé dans ce projet Red Hat OpenShift, suivez les étapes présentées dans la rubrique [«Préparation de votre projet OpenShift pour IBM MQ à l'aide de l'interface de ligne de commande OpenShift»](#), à la page 16.

Procédure

1. Déployez un gestionnaire de files d'attente.

L'exemple ci-après déploie un gestionnaire de files d'attente "à démarrage rapide" qui utilise un stockage éphémère (non persistant) et désactive la sécurité MQ. Les messages ne seront pas conservés suite aux redémarrages du gestionnaire de files d'attente. Vous pouvez ajuster le contenu du fichier YAML afin de changer de nombreux paramètres du gestionnaire de files d'attente.

- a) Création d'un fichier QueueManager.yaml

Par exemple, pour installer un gestionnaire de files d'attente de base dans IBM Cloud Pak for Integration, créez le fichier "mq-quickstart.yaml" dont le contenu est le suivant :

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

Important : si vous acceptez le contrat de licence d'IBM Cloud Pak for Integration, remplacez `accept: false` par `accept: true`. Voir «Référence relative à l'octroi de licence pour mq.ibm.com/v1beta1», à la page 41 pour des détails sur la licence.

Cet exemple inclut également un serveur Web déployé avec le gestionnaire de files d'attente, où la console Web est activée pour la connexion unique avec Cloud Pak Identity and Access Manager.

Pour installer un gestionnaire de files d'attente de base indépendamment d'IBM Cloud Pak for Integration, créez le fichier "mq-quickstart.yaml" dont le contenu est le suivant :

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

Important : si vous acceptez le contrat de licence MQ, modifiez `accept: false` par `accept: true`. Voir «Référence relative à l'octroi de licence pour mq.ibm.com/v1beta1», à la page 41 pour des détails sur la licence.

b) Création de l'objet QueueManager

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

2. Vérifiez que le gestionnaire de files d'attente est en cours d'exécution.

Vous pouvez valider le déploiement en exécutant

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

, puis en vérifiant le statut.

Par exemple, exécutez

```
oc describe queuemanager quickstart
```

, et vérifiez que la zone `status.Phase` indique `Running`

Tâches associées

«[Connexion à un gestionnaire de files d'attente déployé dans un cluster OpenShift](#)», à la page 31
Ensemble d'exemples de configuration pour la connexion à un gestionnaire de files d'attente déployé dans un cluster Red Hat OpenShift.

«[Connexion à IBM MQ Console dans un cluster OpenShift](#)», à la page 33
Connexion à la console IBM MQ Console d'un gestionnaire de files d'attente qui a été déployé dans un cluster Red Hat OpenShift Container Platform.

V 9.1.4 Linux MQ Adv. CD Intégration au tableau de bord des opérations d'IBM Cloud Pak for Integration

La capacité de tracer les transactions via IBM Cloud Pak for Integration est fournie par le tableau de bord des opérations.

Pourquoi et quand exécuter cette tâche

L'activation de l'intégration au tableau de bord des opérations installe un exit API MQ sur votre gestionnaire de files d'attente. L'exit API enverra au magasin de données du tableau de bord des opérations des données de trace sur les messages qui transitent par le gestionnaire de files d'attente.

Notez que seuls les messages qui sont envoyés à l'aide de liaisons client MQ sont tracés.

Procédure

1. Déployez un gestionnaire de files d'attente avec la fonction de trace activée.

Par défaut, la fonction de trace est désactivée.

Si vous procédez au déploiement à l'aide d'IBM Cloud Pak for Integration Platform Navigator, vous pouvez activer la fonction de trace lors du déploiement en associant **Enable Tracing** à la valeur **On** et en définissant l'espace de nom dans lequel le tableau de bord des opérations est installé dans la zone **Tracing Namespace**. Pour plus d'informations sur le déploiement d'un gestionnaire de files d'attente, voir «[Déploiement d'un gestionnaire de files d'attente à l'aide d'IBM Cloud Pak for Integration Platform Navigator](#)», à la page 17.

Si vous procédez au déploiement à l'aide de l'interface de ligne de commande OpenShift ou de la console Web OpenShift, vous pouvez activer la fonction de trace avec le fragment YAML suivant :

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

Si vous déployez à l'aide de Helm, vous pouvez activer le traçage en définissant `odTracingConfig.enabled=true` et `odTracingConfig.odTracingNamespace=<Operations_Dashboard_Namespace>`. Si vous voulez activer l'intégration du tableau de bord des opérations sur un gestionnaire de files d'attente existant, vous pouvez appliquer ce paramètre au cours de la mise à niveau de l'édition de Helm.

Important : le gestionnaire de files d'attente ne démarre pas tant que MQ n'est pas enregistré dans le tableau de bord des opérations (voir l'étape suivante).

Notez que lorsque la fonction est activée, elle exécute deux conteneurs sidecar ("Agent" et "Collecteur") en plus du conteneur de gestionnaire de files d'attente. Les images de ces conteneurs sidecar sont disponibles dans le même registre que l'image principale de MQ et utilisent la même stratégie d'extraction et le même secret d'extraction. Des paramètres supplémentaires sont disponibles pour configurer les limites relatives à l'unité centrale et à la mémoire.

2. S'il s'agit de la première fois qu'un gestionnaire de files d'attente avec l'intégration du tableau de bord des opérations est déployé dans cet espace de nom, vous devez [Enregistrer](#) dans le tableau de bord des opérations.

L'enregistrement crée un objet Secret dont le pod du gestionnaire de files d'attente a besoin pour démarrer.

Linux

MQ Adv.

CD

V 9.1.5

Génération d'une image avec des fichiers MQSC et INI personnalisés à l'aide de l'interface de ligne de commande OpenShift

Utilisez un pipeline Red Hat OpenShift Container Platform pour créer une image de conteneur IBM MQ, avec les fichiers MQSC et INI que vous souhaitez appliquer aux gestionnaires de files d'attente à l'aide de cette image. Cette tâche doit être effectuée par un administrateur de projet.

Avant de commencer

Vous devez installer l'[interface de ligne de commande Red Hat OpenShift Container Platform](#).

Connectez-vous à votre cluster avec **cloudctl login** (pour IBM Cloud Pak for Integration) ou **oc login**.

Si vous ne disposez pas d'un secret OpenShift pour le registre autorisé IBM dans votre projet Red Hat OpenShift, suivez les étapes présentées dans la rubrique [«Préparation de votre projet OpenShift pour IBM MQ à l'aide de l'interface de ligne de commande OpenShift»](#), à la page 16.

Procédure

1. Créez un ImageStream

Un flux d'image et les étiquettes qui lui sont associées fournissent une abstraction pour les images de conteneur de référence depuis Red Hat OpenShift Container Platform. Ils vous permettent de savoir quelles images sont disponibles et de vous assurer que vous utilisez l'image spécifique dont vous avez besoin, même si l'image dans le référentiel change.

```
oc create imagestream mymq
```

2. Créer un BuildConfig pour votre nouvelle image

Un BuildConfig permet de créer pour votre nouvelle image, qui sera basée sur les images officielles IBM, mais qui ajoutera tous les fichiers MQSC ou INI que vous souhaitez exécuter sur le démarrage du conteneur.

- a) Créez un fichier YAML définissant la ressource BuildConfig

Par exemple, créez un fichier nommé "mq-build-config.yaml" dont le contenu est le suivant :

```
apiVersion: build.openshift.io/v1
kind: BuildConfig
metadata:
  name: mymq
spec:
  source:
    dockerfile: |-
      FROM cp.icr.io/cp/ibm-mqadvanced-server-integration:9.1.5.0-r2-amd64
      RUN printf "DEFINE QLOCAL(foo) REPLACE\n" > /etc/mqm/my.mqsc \
        && printf "Channels:\n\tMQIBindType=FASTPATH\n" > /etc/mqm/my.ini
      LABEL summary "My custom MQ image"
  strategy:
    type: Docker
    dockerStrategy:
```

```
from:
  kind: "DockerImage"
  name: "cp.icr.io/cp/ibm-mqadvanced-server-integration:9.1.5.0-r2-amd64"
pullSecret:
  name: ibm-entitlement-key
output:
  to:
    kind: ImageStreamTag
    name: 'mymq:latest-amd64'
```

Vous devrez remplacer les deux emplacements du produit IBM MQ de base afin de désigner l'image de base appropriée pour la version et le correctif que vous voulez utiliser. Au fur et à mesure que les correctifs sont appliqués, vous devez répéter ces étapes afin de régénérer votre image.

Cet exemple crée une nouvelle image basée sur l'image officielle IBM et ajoute les fichiers "my.mqsc" et "my.ini" dans le répertoire /etc/mqm. Tout fichier MQSC ou INI trouvé dans ce répertoire sera appliqué par le conteneur au démarrage. Les fichiers INI sont appliqués avec l'option **crtmqm -ii** et fusionnés avec les fichiers INI existants. Les fichiers MQSC sont appliqués par ordre alphabétique.

Il est important que vos commandes MQSC puissent être réexécutées, car elles seront exécutées à *chaque fois* que le gestionnaire de files d'attente démarre. Cela implique généralement d'ajouter le paramètre REPLACE à toutes les commandes DEFINE et d'ajouter le paramètre IGNSTATE (YES) à toutes les commandes START ou STOP.

- b) Appliquez le BuildConfig au serveur.

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

3. Exécutez une génération pour créer votre image.

- a) Démarrez la génération.

```
oc start-build mymq
```

Une sortie similaire à la suivante apparaît :

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

- b) Vérifiez le statut de la génération.

Par exemple, vous pouvez exécuter la commande suivante en utilisant l'identificateur de génération renvoyé à l'étape précédente :

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

4. Déployez un gestionnaire de files d'attente en utilisant votre nouvelle image.

Suivez les étapes décrites dans la rubrique [«Déploiement d'un gestionnaire de files d'attente à l'aide de l'interface de ligne de commande OpenShift»](#), à la page 19 pour ajouter votre nouvelle image personnalisée dans le fichier YAML.

Vous pouvez ajouter le fragment suivant dans votre fichier YAML standard pour QueueManager, où *my-namespace* est le projet/l'espace de nom OpenShift que vous utilisez et *my-image* est le nom de l'image que vous avez créée précédemment (par exemple "mymq:latest-amd64") :

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

Tâches associées

[«Déploiement d'un gestionnaire de files d'attente à l'aide de l'interface de ligne de commande OpenShift»](#), à la page 19

Utilisez la ressource personnalisée QueueManager pour déployer un gestionnaire de files d'attente dans un cluster Red Hat OpenShift Container Platform à l'aide de l'interface de ligne de commande (CLI). Cette tâche doit être effectuée par un administrateur de projet.

l'aide de Helm

A partir de IBM MQ 9.1.5.0, la méthode recommandée pour déployer un gestionnaire de files d'attente consiste à utiliser l'opérateur IBM MQ . IBM MQ 9.1.5.0 et les éditions CD précédentes peuvent être déployées à l'aide de Helm, en suivant les instructions ci-après.

Pourquoi et quand exécuter cette tâche

Procédure

- «Préparation de votre cluster OpenShift pour IBM MQ sur OpenShift à l'aide de Helm», à la page 24.
- «Déploiement d'un gestionnaire de files d'attente à l'aide de l'interface de ligne de commande Helm», à la page 25.

Préparation de votre cluster OpenShift pour IBM MQ sur OpenShift à l'aide de Helm

Préparez votre cluster Red Hat OpenShift Container Platform pour qu'il soit prêt à déployer un gestionnaire de files d'attente à l'aide de Helm. Cette tâche doit être effectuée par un administrateur de cluster.

Avant de commencer

Remarque : Si vous utilisez IBM Cloud Pak for Integration, le programme d'installation doit avoir préparé un projet OpenShift (espace de nom) à utiliser avec IBM MQ, de sorte que vous n'avez peut-être pas besoin de suivre ces instructions.

Connectez-vous à votre cluster avec **cloudctl login** (pour IBM Cloud Pak for Integration) ou **oc login**.

Procédure

1. Assurez-vous d'avoir ajouté le référentiel IBM Helm à votre copie locale de Helm.

Par exemple, vous pouvez exécuter la commande suivante :

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

2. Assurez-vous qu'un serveur Helm (appelé "Tiller") est installé dans votre cluster.
Suivez les instructions présentées dans la page [Getting started with Helm on OpenShift](#) pour installer Helm dans votre cluster.
3. Assurez-vous que les comptes de service dans votre projet OpenShift (espace de nom) sont autorisés à utiliser les contraintes de contexte de sécurité (SCC) appropriées.

V 9.1.5

IBM MQ fonctionne avec la contrainte de contexte de sécurité par défaut "restricted" ; par conséquent, vous pouvez normalement ignorer cette étape.

L'application de modifications aux contraintes de contexte de sécurité doit être effectuée par un administrateur de cluster OpenShift. Chaque version de charte Helm présente des exigences différentes relatives aux contraintes de contexte de sécurité, qui sont documentées dans les fichiers Readme individuels de chaque charte Helm :

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

Chaque fichier Readme contient des instructions permettant de configurer l'autorisation pour les contraintes de contexte de sécurité. Notez que les chartes Helm IBM MQ créent un compte de service pour leur propre utilisation, ce qui signifie que les droits relatifs aux contraintes de contexte de

sécurité doivent être appliqués au niveau du groupe (pour tous les comptes de service dans l'espace de nom).

4. Assurez-vous de disposer d'un "secret d'extraction d'image" valide pour extraire des images depuis le registre de conteneurs de votre choix.

Les images de IBM MQ Advanced certified container sont extraites depuis un registre de conteneurs qui effectue un contrôle des autorisations de licence. Ce contrôle requiert une clé d'autorisation qui est stockée dans un secret d'extraction `docker-registry`. Si vous ne possédez pas encore de clé d'autorisation, suivez les instructions ci-après pour en obtenir une et créer un secret d'extraction.

- a) Obtenez la clé d'autorisation qui est affectée à votre ID.
 - i) Connectez-vous à [Mon IBM - Bibliothèque des logiciels de conteneur](#) avec l'ID et le mot de passe IBM associés au logiciel autorisé.
 - ii) Dans la section *Clé d'autorisation*, cliquez sur **Copier la clé** pour copier la clé d'autorisation dans le presse-papiers.
- b) Créez le secret dans l'espace de nom dans lequel vous voulez déployer votre gestionnaire de files d'attente.
 - Exécutez la commande suivante, où `<entitlement-key>` est la clé extraite à l'étape 1, et `<user-email>` est l'ID IBM associé au logiciel autorisé.

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

Que faire ensuite

[«Déploiement d'un gestionnaire de files d'attente à l'aide de l'interface de ligne de commande Helm»](#), à la page 25

V 9.1.4 Linux MQ Adv CD Déploiement d'un gestionnaire de files d'attente à l'aide de l'interface de ligne de commande Helm

Utilisez Helm pour déployer un gestionnaire de files d'attente dans un cluster Red Hat OpenShift Container Platform. Cette tâche doit être effectuée par un administrateur de projet.

Avant de commencer

Vous devez installer Helm V2 et l'interface de ligne de commande de Red Hat OpenShift Container Platform. Si vous n'utilisez pas IBM Cloud Pak for Integration, suivez la procédure de [«Préparation de votre cluster OpenShift pour IBM MQ sur OpenShift à l'aide de Helm»](#), à la page 24.

Connectez-vous à votre cluster avec **cloudctl login** (pour IBM Cloud Pak for Integration) ou **oc login**.

Procédure

1. Assurez-vous d'avoir ajouté le référentiel IBM Helm à votre copie locale de Helm. Par exemple, vous pouvez exécuter la commande suivante :

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

2. Révisez les options de configuration pour votre gestionnaire de files d'attente.

L'étape de déploiement inclut des étapes d'installation et des étapes de configuration. Certains paramètres pour votre gestionnaire de files d'attente doivent être définis lors du déploiement et leur modification requiert un redéploiement.

Vous pouvez afficher le fichier Readme de la charte Helm pour des détails sur toutes les options de déploiement disponibles, en exécutant l'une des commandes suivantes :

- Pour un IBM MQ Advanced certified container dans IBM Cloud Pak for Integration :

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

- Pour un IBM MQ Advanced certified container :

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

En général, les paramètres suivants au moins sont nécessaires :

- a. Le nom de l'édition. Exemple : `my-release`
- b. Le référentiel Helm distant. Exemple : `ibm-entitled-charts`
- c. La charte Helm. Exemple : `ibm-mqadvanced-server-prod` ou `ibm-mqadvanced-server-integration-prod`
- d. Le nom du secret d'extraction de l'image. Exemple : `entitled-registry`. Notez qu'il n'est pas nécessaire si vous procédez au déploiement dans le projet prédéfini pour MQ dans IBM Cloud Pak for Integration.

3. Déployez un gestionnaire de files d'attente.

Notez que par défaut, la charte Helm suppose qu'une [classe de stockage](#) par défaut est définie dans votre cluster Red Hat OpenShift Container Platform.

Par exemple, pour installer un gestionnaire de files d'attente de base dans IBM Cloud Pak for Integration, exécutez la commande suivante :

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

Vous pouvez entrer n'importe quel nom d'hôte dans la zone `tls.hostname` (il s'agit d'une zone requise mais elle n'est pas utilisée dans cet exemple car nous générons ici un nouveau certificat autosigné).

Pour installer un gestionnaire de files d'attente de base indépendamment d'IBM Cloud Pak for Integration, vous pouvez exécuter la commande suivante :

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

Tâches associées

«Connexion à un gestionnaire de files d'attente déployé dans un cluster OpenShift», à la page 31
Ensemble d'exemples de configuration pour la connexion à un gestionnaire de files d'attente déployé dans un cluster Red Hat OpenShift.

«Connexion à IBM MQ Console dans un cluster OpenShift», à la page 33
Connexion à la console IBM MQ Console d'un gestionnaire de files d'attente qui a été déployé dans un cluster Red Hat OpenShift Container Platform.

Déploiement d'un gestionnaire de files d'attente avec IBM Cloud File Storage à l'aide de l'interface de ligne de commande Helm

Exemple de scénario d'utilisation de Helm pour déployer un gestionnaire de files d'attente sur un cluster Red Hat OpenShift on IBM Cloud, à l'aide de IBM Cloud File Storage. Cette tâche doit être effectuée par un administrateur de projet.

Avant de commencer

Vous devez installer [Helm V2](#) et [l'interface de ligne de commande de Red Hat OpenShift Container Platform](#). Si vous n'utilisez pas IBM Cloud Pak for Integration, suivez la procédure de [«Préparation de votre cluster OpenShift pour IBM MQ sur OpenShift à l'aide de Helm»](#), à la page 24.

Connectez-vous à votre cluster avec **cloudctl login** (pour IBM Cloud Pak for Integration) ou **oc login**.

Procédure

1. Vérifiez que vous avez ajouté le référentiel IBM Helm à votre copie locale de Helm.
Par exemple, vous pouvez exécuter la commande suivante :

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

2. Déployez un gestionnaire de files d'attente.

Lors de l'utilisation de IBM Cloud File Storage, vous verrez généralement les meilleurs résultats avec la classe de stockage `ibmc-file-gold-gid`. Cette classe de stockage active le stockage dans lequel les utilisateurs du groupe de systèmes de fichiers approprié peuvent écrire.

Par exemple, pour installer un gestionnaire de files d'attente de base dans IBM Cloud Pak for Integration, exécutez la commande suivante :

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

Vous pouvez entrer n'importe quel nom d'hôte dans la zone `tls.hostname` (il s'agit d'une zone obligatoire qui n'est pas utilisée ici, car dans cet exemple, nous générons un nouveau certificat autosigné).

Pour installer un gestionnaire de files d'attente de base indépendamment d'IBM Cloud Pak for Integration, vous pouvez exécuter la commande suivante :

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

Tâches associées

[«Connexion à un gestionnaire de files d'attente déployé dans un cluster OpenShift»](#), à la page 31
Ensemble d'exemples de configuration pour la connexion à un gestionnaire de files d'attente déployé dans un cluster Red Hat OpenShift.

[«Connexion à IBM MQ Console dans un cluster OpenShift»](#), à la page 33

Connexion à la console IBM MQ Console d'un gestionnaire de files d'attente qui a été déployé dans un cluster Red Hat OpenShift Container Platform.

Linux

MQ Adv.

CD

Déploiement d'éditions CD précédentes d'IBM MQ dans un cluster IBM Cloud Private

Pour les versions CD de IBM MQ antérieures à 9.1.4, utilisez la console de gestion IBM Cloud Private pour déployer un gestionnaire de files d'attente dans IBM Cloud Private.

Avant de commencer



Avertissement : **V 9.1.4** Ce déploiement n'est pas pris en charge dans IBM MQ 9.1.4 ou les versions ultérieures.

Cette tâche suppose que vous avez déjà [ajouté une image IBM MQ dans un cluster IBM Cloud Private](#).

Le fichier README .md de la charte Helm est disponible depuis l'entrée de catalogue IBM Cloud Private, qui s'affiche une fois que vous avez terminé cette sous-étape, ou depuis la ligne de commande si vous [ajoutez votre référentiel **local-charts** d'IBM Cloud Private en tant que référentiel Helm distant](#) en exécutant la commande suivante :

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

Vous devez disposer d'une règle [PodSecurityPolicy](#) ou d'une contrainte [SecurityContextConstraint](#) (pour IBM Cloud Private sous Red Hat OpenShift) qui prend en charge le contexte de sécurité nécessaire. Vous trouverez des détails, ainsi que des exemples, dans le fichier README .md de la charte Helm.

Vous trouverez également des détails sur la configuration de votre édition Helm dans le fichier README .md de la charte Helm.

Remarque :

- Si vous procédez au déploiement dans un environnement IBM Cloud Private qui ne prend pas en charge les paramètres de sécurité par défaut, activez votre déploiement en suivant les instructions présentées dans la rubrique [Déploiement de chartes Helm nécessitant des privilèges élevés dans un espace de nom autre que celui défini par défaut](#) dans la documentation du produit IBM Cloud Private.
- Si vous utilisez SELinux, vous devez satisfaire les IBM MQ exigences décrites dans [IBM MQ support for SELinux on Red Hat Enterprise Linux](#).

Pourquoi et quand exécuter cette tâche

IBM Cloud Private propose une plateforme de gestion des applications conteneurisées sur site. Une fois que vous avez ajouté une image IBM MQ dans un cluster IBM Cloud Private, vous pouvez utiliser la console de gestion d'IBM Cloud Private ou la ligne de commande pour déployer un gestionnaire de files d'attente.

Procédure

- Utilisation de la console de gestion d'IBM Cloud Private
 - a) Ouvrez la console de gestion d'IBM Cloud Private dans un navigateur Web et cliquez sur **Catalogue**.
Voir [Accès à votre cluster IBM Cloud Private à l'aide de la console de gestion](#) dans la documentation du produit IBM Cloud Private.
 - b) Sélectionnez le graphique `ibm-mqadvanced-server-prod` dans la liste.
 - c) Sélectionnez **Configurer**, puis effectuez les étapes de configuration ci-après :
 - a. Entrez un nom de version.
 - b. Lisez et acceptez le contrat de licence.

- c. Sous la section **dataPVC**, pour **storageclass**, définissez le nom de classe d'archivage de votre choix. N'indiquez rien pour sélectionner la classe d'archivage par défaut.
- d. Dans la section **image**, pour le référentiel, définissez le chemin d'accès complet à l'image.
Exemple :

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

- e. Dans la section **image**, spécifiez l'étiquette de l'image comme étiquette. Exemple :

```
9.1.3.0-r1
```

- f. Si vous avez besoin d'un secret d'extraction Kubernetes pour accéder au registre des images, ajoutez-le à l'aide du paramètre **pullSecret**.
- g. Dans la section **queueManager**, définissez le nom du gestionnaire de files d'attente.
- d) Cliquez sur **Installer** pour déployer votre gestionnaire de files d'attente en tant qu'*édition Helm*.

- Utilisation de la ligne de commande

- a) Configurez **cloudctl** pour accéder à votre cluster IBM Cloud Private.

Voir [Installing the IBM Cloud Private CLI](#) dans la documentation du produit IBM Cloud Private.

- b) Assurez-vous d'avoir ajouté votre référentiel **local-charts** d'IBM Cloud Private en tant que référentiel Helm distant.

- c) Installez la charte.

Exécutez la commande suivante en spécifiant les paramètres ci-dessous :

- a. Le nom de l'édition (par exemple `my-release`)
- b. Le nom du référentiel Helm distant qui contient la charte `ibm-mqadvanced-server-prod` (par exemple `my-repo`)
- c. Référentiel d'images (par exemple, `mycluster.icp:8500/namespace_name/ibm-mqadvanced-server-prod`)
- d. L'étiquette de l'image (par exemple `9.1.3.0-r1`)

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

Tâches associées

«[Déploiement d'un gestionnaire de files d'attente à l'aide de l'interface de ligne de commande Helm](#)», à la page 25

Utilisez Helm pour déployer un gestionnaire de files d'attente dans un cluster Red Hat OpenShift Container Platform. Cette tâche doit être effectuée par un administrateur de projet.

«[Ajout d'éditions CD précédentes d'une image d'IBM MQ dans un cluster IBM Cloud Private](#)», à la page 29

Pour les versions CD de IBM MQ antérieures à 9.1.4, préparez votre cluster IBM Cloud Private à déployer une image prête pour la production pour IBM MQ.

«[Ajout d'éditions CD précédentes d'une image d'IBM MQ dans un cluster IBM Cloud Kubernetes Service](#)», à la page 30

Pour les versions CD de IBM MQ antérieures à 9.1.4, importez une image prête pour la production pour IBM MQ dans IBM Cloud Kubernetes Service.

Linux

MQ Adv.

CD

Ajout d'éditions CD précédentes d'une image d'IBM MQ dans un cluster IBM Cloud Private

Pour les versions CD de IBM MQ antérieures à 9.1.4, préparez votre cluster IBM Cloud Private à déployer une image prête pour la production pour IBM MQ.

Pourquoi et quand exécuter cette tâche



Avertissement : **V 9.1.4** Cette importation n'est pas prise en charge dans IBM MQ 9.1.4 ou les versions ultérieures.

Vous pouvez télécharger une image IBM MQ depuis Passport Advantage et l'importer dans un conteneur IBM Cloud Private.

Procédure

1. Téléchargez l'image d'IBM MQ la plus récente depuis le [Site Web Passport Advantage et Passport Advantage Express](#).

Pour les détails des téléchargements disponibles, accédez à la page [Téléchargement de IBM MQ 9.1](#), puis cliquez sur l'onglet correspondant à l'édition que vous souhaitez télécharger. Le nom et la référence du composant à télécharger sont répertoriés dans un tableau.

2. Importez le fichier archive téléchargé dans IBM Cloud Private.

Voir [Ajout de logiciels IBM dans le catalogue IBM Cloud Private](#) dans la documentation du produit IBM Cloud Private.

Que faire ensuite

Vous êtes maintenant prêt à [déployer un gestionnaire de files d'attente dans IBM Cloud Private](#).

Tâches associées

«[Déploiement d'un gestionnaire de files d'attente à l'aide de l'interface de ligne de commande Helm](#)», à la page 25

Utilisez Helm pour déployer un gestionnaire de files d'attente dans un cluster Red Hat OpenShift Container Platform. Cette tâche doit être effectuée par un administrateur de projet.

«[Déploiement d'éditions CD précédentes d'IBM MQ dans un cluster IBM Cloud Private](#)», à la page 28

Pour les versions CD de IBM MQ antérieures à 9.1.4, utilisez la console de gestion IBM Cloud Private pour déployer un gestionnaire de files d'attente dans IBM Cloud Private.

«[Ajout d'éditions CD précédentes d'une image d'IBM MQ dans un cluster IBM Cloud Kubernetes Service](#)», à la page 30

Pour les versions CD de IBM MQ antérieures à 9.1.4, importez une image prête pour la production pour IBM MQ dans IBM Cloud Kubernetes Service.

Linux

MQ Adv.

CD

Ajout d'éditions CD précédentes d'une image d'IBM MQ dans un cluster IBM Cloud Kubernetes Service

Pour les versions CD de IBM MQ antérieures à 9.1.4, importez une image prête pour la production pour IBM MQ dans IBM Cloud Kubernetes Service.

Pourquoi et quand exécuter cette tâche



Avertissement : **V 9.1.4** Cette importation n'est pas prise en charge dans IBM MQ 9.1.4 ou les versions ultérieures.

Vous pouvez télécharger une image IBM MQ depuis Passport Advantage et l'importer dans un cluster IBM Cloud Kubernetes Service.

Procédure

1. Téléchargez l'image d'IBM MQ la plus récente depuis le [Site Web Passport Advantage et Passport Advantage Express](#).

Pour les détails des téléchargements disponibles, accédez à la page [Téléchargement de IBM MQ 9.1](#), puis cliquez sur l'onglet correspondant à l'édition que vous souhaitez télécharger. Le nom et la référence du composant à télécharger sont répertoriés dans un tableau.

2. Importez le fichier archive téléchargé dans IBM Cloud Kubernetes Service.

Voir [Exécution d'images IBM Cloud Private dans des conteneurs Kubernetes publics](#).

Tâches associées

«[Déploiement d'un gestionnaire de files d'attente à l'aide de l'interface de ligne de commande Helm](#)», à la page 25

Utilisez Helm pour déployer un gestionnaire de files d'attente dans un cluster Red Hat OpenShift Container Platform. Cette tâche doit être effectuée par un administrateur de projet.

«[Déploiement d'éditions CD précédentes d'IBM MQ dans un cluster IBM Cloud Private](#)», à la page 28

Pour les versions CD de IBM MQ antérieures à 9.1.4, utilisez la console de gestion IBM Cloud Private pour déployer un gestionnaire de files d'attente dans IBM Cloud Private.

«[Ajout d'éditions CD précédentes d'une image d'IBM MQ dans un cluster IBM Cloud Private](#)», à la page 29

Pour les versions CD de IBM MQ antérieures à 9.1.4, préparez votre cluster IBM Cloud Private à déployer une image prête pour la production pour IBM MQ.

V 9.1.4 Linux MQ Adv. CD Connexion à un gestionnaire de files d'attente déployé dans un cluster OpenShift

Ensemble d'exemples de configuration pour la connexion à un gestionnaire de files d'attente déployé dans un cluster Red Hat OpenShift.

Pourquoi et quand exécuter cette tâche

Une [route OpenShift](#) est nécessaire pour la connexion d'une application à un gestionnaire de files d'attente IBM MQ depuis un emplacement hors du cluster Red Hat OpenShift.

Vous devez activer TLS dans votre application client et votre gestionnaire de files d'attente IBM MQ car l'[indication de nom de serveur](#) (SNI) est disponible uniquement dans le protocole TLS. Le routeur Red Hat OpenShift Container Platform utilise l'indication de nom de serveur pour acheminer les demandes vers le gestionnaire de files d'attente IBM MQ.

La configuration requise pour la route OpenShift dépend du comportement de l'indication de nom de serveur de votre application client.

Pour définir l'en-tête SNI comme TLS 1.2 ou supérieur, un CipherSpec ou CipherSuite doit être utilisé pour votre communication TLS.

L'interface SNI est définie sur le canal MQ si les conditions suivantes sont remplies:

- Le client IBM MQ C est V8 ou version ultérieure.
- Le client Java/JMS est V9.1.1 ou ultérieure, et l'installation Java prend en charge la classe `javax.net.ssl.SNIHostName`.
- Le client .NET est en mode non géré.

La valeur SNI est définie sur le nom d'hôte si un nom d'hôte est fourni comme nom de connexion et que les conditions suivantes sont remplies:

- Le client .NET est en mode géré.
- Le client AMQP ou XR est utilisé.
- Les clients Java/JMS sont utilisés avec **AllowOutboundSNI** défini sur NO.

Le SNI n'est pas défini et est vide dans les conditions suivantes:

- Le client IBM MQ C est V7.5 ou une version antérieure.
- IBM MQ C Client est utilisé avec **AllowOutboundSNI** défini sur NO.

- Les clients Java/JMS sont utilisés avec une installation Java qui ne prend pas en charge la classe `javax.net.ssl.SNIHostName`.

Exemple

Routes OpenShift basées sur le nom d'hôte : pour les applications client qui définissent le nom d'hôte comme indicateur de nom de serveur

Les chartes Helm ci-dessous créent automatiquement une route OpenShift basée sur le nom d'hôte pour la connexion d'une application à un gestionnaire de files d'attente IBM MQ. Les applications client qui définissent le nom d'hôte comme indicateur de nom de serveur peuvent utiliser cette route OpenShift.

- `ibm-mqadvanced-server-dev`
- `ibm-mqadvanced-server-prod`
- `ibm-mqadvanced-server-integration-prod` dans IBM Cloud Pak for Integration.

Si vous n'utilisez pas ces chartes et devez créer votre propre route OpenShift basée sur le nom d'hôte, vous pouvez appliquer le code `yaml` suivant dans votre cluster :

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: <provide a unique name for the Route>
  namespace: <namespace of your MQ deployment>
spec:
  to:
    kind: Service
    name: <name of the Kubernetes Service for your MQ deployment (for example "<Helm Release>-ibm-mq")>
  port:
    targetPort: 1414
  tls:
    termination: passthrough
```

Routes OpenShift basées sur le canal MQ : pour les applications client qui définissent le canal MQ comme indicateur de nom de serveur

Les applications client qui définissent le canal MQ comme indication de nom de serveur requièrent la création d'une route OpenShift pour chaque canal auquel vous voulez vous connecter. Vous devez aussi utiliser des noms de canal uniques dans votre cluster Red Hat OpenShift pour permettre un routage vers le gestionnaire de files d'attente approprié.

Afin de déterminer le nom d'hôte requis pour chaque nouvelle route OpenShift, vous devez mapper chaque nom de canal à une adresse SNI comme expliqué ici : <https://www.ibm.com/support/pages/ibm-websphere-mq-how-does-mq-provide-multiple-certificates-certlabl-capability>

Ensuite, vous devez créer une route OpenShift (pour chaque canal) en appliquant le code `yaml` suivant dans votre cluster :

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: <provide a unique name for the Route>
  namespace: <the namespace of your MQ deployment>
spec:
  host: <SNI address mapping for the channel>
  to:
    kind: Service
    name: <the name of the Kubernetes Service for your MQ deployment (for example "<Helm Release>-ibm-mq")>
  port:
    targetPort: 1414
  tls:
    termination: passthrough
```

Configuration des détails de connexion de votre application client

Vous pouvez déterminer le nom d'hôte à utiliser pour votre connexion client en exécutant la commande suivante :

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

Le port pour votre connexion client doit être le port utilisé par le routeur OpenShift Container Platform (OCP) ; il s'agit normalement du port 443.

Tâches associées

«Déploiement d'un gestionnaire de files d'attente à l'aide de l'interface de ligne de commande Helm», à la page 25

Utilisez Helm pour déployer un gestionnaire de files d'attente dans un cluster Red Hat OpenShift Container Platform. Cette tâche doit être effectuée par un administrateur de projet.

«Connexion à IBM MQ Console dans un cluster OpenShift», à la page 33

Connexion à la console IBM MQ Console d'un gestionnaire de files d'attente qui a été déployé dans un cluster Red Hat OpenShift Container Platform.

V 9.1.4 Linux MQ Adv. CD Connexion à IBM MQ Console dans un cluster OpenShift

Connexion à la console IBM MQ Console d'un gestionnaire de files d'attente qui a été déployé dans un cluster Red Hat OpenShift Container Platform.

Pourquoi et quand exécuter cette tâche

Si vous utilisez l'opérateur IBM MQ , l'URL IBM MQ Console se trouve sur la page des détails de QueueManager dans la console Web OpenShift ou dans IBM Cloud Pak for Integration Platform Navigator. Vous pouvez également l'identifier à partir de l'interface de ligne de commande OpenShift en exécutant la commande suivante :

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

Exemple

Les chartes Helm suivantes créent automatiquement une route OpenShift pour l'accès à IBM MQ Console :

- ibm-mqadvanced-server-dev
- ibm-mqadvanced-server-integration-prod dans IBM Cloud Pak for Integration.

Vous pouvez obtenir le nom d'hôte de la route OpenShift en exécutant la commande suivante :

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

Vous pouvez accéder à IBM MQ Console avec l'URL suivante :

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

Tâches associées

«Déploiement d'un gestionnaire de files d'attente à l'aide de l'interface de ligne de commande Helm», à la page 25

Utilisez Helm pour déployer un gestionnaire de files d'attente dans un cluster Red Hat OpenShift Container Platform. Cette tâche doit être effectuée par un administrateur de projet.

«Connexion à un gestionnaire de files d'attente déployé dans un cluster OpenShift», à la page 31

Ensemble d'exemples de configuration pour la connexion à un gestionnaire de files d'attente déployé dans un cluster Red Hat OpenShift.

Sauvegarde et restauration de la configuration de gestionnaire de files d'attente à l'aide de l'interface de ligne de commande OpenShift

Effectuez une sauvegarde de la configuration de gestionnaire de files d'attente pour pouvoir régénérer un gestionnaire de files d'attente depuis ses définitions en cas de perte de la configuration de gestionnaire de files d'attente. Cette procédure n'effectue pas de sauvegarde des données de journal du gestionnaire de files d'attente. En raison de la nature transitoire des messages, les données de journal historiques ne sont généralement pas pertinentes au moment de la restauration.

Avant de commencer

Connectez-vous à votre cluster avec **cloudctl login** (pour IBM Cloud Pak for Integration) ou **oc login**.

Procédure

- Effectuez une sauvegarde de la configuration de gestionnaire de files d'attente.

Vous pouvez utiliser la commande **dmpmqcfg** pour vider la configuration d'un gestionnaire de files d'attente IBM MQ.

- Obtenez le nom du pod pour votre gestionnaire de files d'attente.

Par exemple, si vous utilisez l'opérateur, vous pouvez exécuter la commande suivante, où *queue_manager_name* est le nom de votre ressource QueueManager :

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

Par exemple, si vous utilisez Helm, vous pouvez exécuter la commande suivante, où *nom_édition* est le nom de votre édition Helm .

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

- Exécutez la commande **dmpmqcfg** sur le pod, en dirigeant la sortie dans un fichier sur votre machine locale.

dmpmqcfg génère la configuration MQSC du gestionnaire de files d'attente.

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

- Restaurez la configuration de gestionnaire de files d'attente.

Après avoir suivi la procédure de sauvegarde décrite à l'étape précédente, vous devez disposer d'un fichier `backup.mqsc` contenant la configuration du gestionnaire de files d'attente. Vous pouvez restaurer la configuration en appliquant ce fichier à un nouveau gestionnaire de files d'attente.

- Obtenez le nom du pod pour votre gestionnaire de files d'attente.

Par exemple, si vous utilisez l'opérateur, vous pouvez exécuter la commande suivante, où *queue_manager_name* est le nom de votre ressource QueueManager :

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

Par exemple, si vous utilisez Helm, vous pouvez exécuter la commande suivante, où *nom_édition* est le nom de votre édition Helm .

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

- Exécutez la commande **runmqsc** sur le pod, en dirigeant la sortie dans le fichier `backup.mqsc`.

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

Génération de votre propre conteneur IBM MQ

Développez un conteneur que vous avez généré vous-même, anciennement une "image de conteneur Docker". Il s'agit de la solution de conteneur la plus souple, qui exige toutefois de solides compétences relatives à la configuration des conteneurs et qui requiert que vous "possédiez" le conteneur résultant.

Avant de commencer

Avant de développer votre propre conteneur, déterminez si vous pouvez utiliser l'un des conteneurs préconditionnés fournis par IBM. Voir [IBM MQ dans des conteneurs](#).

Pourquoi et quand exécuter cette tâche

Lorsque vous conditionnez IBM MQ en tant qu'image de conteneur, les modifications apportées à votre application peuvent être déployées sur des systèmes de test et de transfert rapidement et facilement, ce qui peut constituer un avantage majeur pour la distribution continue dans votre entreprise.

Procédure

- Pour des informations sur la génération d'une image de conteneur IBM MQ avec Docker, voir les sous-rubriques suivantes :
 -  [«Prise en charge de la génération de vos propres images de conteneur IBM MQ et de vos propres chartes»](#), à la page 8
 - [«Planification de votre propre image de gestionnaire de files d'attente IBM MQ à l'aide d'un conteneur»](#), à la page 35
 - [«Génération d'une image de gestionnaire de files d'attente IBM MQ avec Docker»](#), à la page 36
 - [«Exécution d'applications de liaison locale dans des conteneurs distincts»](#), à la page 39

Concepts associés

[IBM MQ dans des conteneurs](#)

Planification de votre propre image de gestionnaire de files d'attente IBM MQ à l'aide d'un conteneur

Vous devez tenir compte de plusieurs exigences lorsque vous exécutez un gestionnaire de files d'attente IBM MQ dans un conteneur. L'exemple d'image de conteneur répond à ces exigences, mais si vous voulez utiliser votre propre image, vous devez examiner la façon dont ces exigences sont traitées.

Supervision du processus

Lorsque vous exécutez un conteneur, vous exécutez principalement un processus unique (PID 1 dans le conteneur), qui peut ensuite engendrer des processus enfant.

Si le processus principal s'arrête, l'exécution du conteneur arrête le conteneur. Un gestionnaire de files d'attente IBM MQ requiert l'exécution de plusieurs processus en arrière-plan.

Par conséquent, vous devez vous assurer que votre processus principal reste actif tant que le gestionnaire de files d'attente est en cours d'exécution. Il est recommandé de vérifier que le gestionnaire de files d'attente est actif depuis ce processus, par exemple en émettant des requêtes administratives.

Remplissage de `/var/mqm`

Les conteneurs doivent être configurés avec `/var/mqm` en tant que volume.

Dans ce cas, le répertoire du volume est vide lorsque le conteneur démarre pour la première fois. En général, ce répertoire est rempli à l'installation, mais l'installation et l'exécution sont des environnements distincts dans le cadre de l'utilisation d'un conteneur.

V 9.1.0 Pour résoudre ce problème, lorsque votre conteneur démarre, vous pouvez utiliser la commande `crtmqdir` pour remplir `/var/mqm` lorsqu'il s'exécute pour la première fois.

Génération d'une image de gestionnaire de files d'attente IBM MQ avec Docker

Utilisez ces informations pour générer un exemple d'image de conteneur afin d'exécuter un gestionnaire de files d'attente IBM MQ dans un conteneur.

Pourquoi et quand exécuter cette tâche

Tout d'abord, vous générez une image de base contenant un système de fichiers Red Hat Universal Base Image et une installation propre d'IBM MQ.

Ensuite, vous générez une autre couche d'image de conteneur sur la base, qui ajoute une configuration IBM MQ assurant une sécurité de base par ID utilisateur et mot de passe.

Enfin, vous exécutez un conteneur à l'aide de cette image comme système de fichiers, avec le contenu de `/var/mqm` fourni par un volume spécifique au conteneur sur le système de fichiers hôte.

Procédure

- Pour des informations sur la génération d'un exemple d'image de conteneur pour l'exécution d'un gestionnaire de files d'attente IBM MQ dans un conteneur, voir les sous-rubriques suivantes :
 - [«Génération d'un exemple d'image de gestionnaire de files d'attente IBM MQ de base»](#), à la page 36
 - [«Génération d'un exemple d'image de gestionnaire de files d'attente IBM MQ configurée»](#), à la page 36

Génération d'un exemple d'image de gestionnaire de files d'attente IBM MQ de base

Pour utiliser IBM MQ dans votre propre image de conteneur, vous devez d'abord générer une image de base avec une installation propre d'IBM MQ. Les étapes ci-dessous expliquent comment générer un exemple d'image de base à l'aide d'un exemple de code hébergé sur GitHub.

Procédure

- Utilisez les fichiers `make` fournis dans le [référentiel GitHub du conteneur mq](#) pour générer l'image de conteneur de production.

Suivez les instructions de la section [Génération d'une image de conteneur](#) sur GitHub.

Résultats

A présent, vous disposez d'une image de conteneur de base dans laquelle IBM MQ est installé.

Génération d'un exemple d'image de gestionnaire de files d'attente IBM MQ configurée

Une fois que vous avez généré l'image de conteneur IBM MQ de base générique, vous devez appliquer votre propre configuration pour autoriser l'accès sécurisé. Pour ce faire, créez votre propre couche d'image de conteneur en utilisant l'image générique comme parent.

Avant de commencer

Pour une image IBM MQ 9.1, vous ne pouvez pas configurer l'accès sécurisé à l'aide de la contrainte de contexte de sécurité (SCC) Red Hat OpenShift Container Platform "restreinte". La contrainte de contexte de sécurité "restreinte" utilise des ID utilisateur aléatoires et empêche l'escalade des privilèges en changeant d'utilisateur. Le programme d'installation basé sur RPM IBM MQ 9.1 s'appuie sur un utilisateur et un groupe `mqm` et utilise également des bits `setuid` sur les programmes exécutables.

Cette restriction est supprimée dans IBM MQ 9.2.

Procédure

1. Créez un répertoire et ajoutez un fichier nommé `config.mqsc` dont le contenu est le suivant :

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

Notez que l'exemple précédent utilise une authentification simple par ID utilisateur et mot de passe. Toutefois, vous pouvez appliquer toute configuration de sécurité requise par votre entreprise.

2. Créez un fichier nommé `Dockerfile` dont le contenu est le suivant :

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

où :

- `johndoe` est l'ID utilisateur à ajouter
- `passw0rd` est le mot de passe d'origine

3. Générez votre image de conteneur personnalisée avec la commande suivante :

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

Où "." est le répertoire contenant les deux fichiers que vous venez de créer.

Docker crée ensuite un conteneur temporaire à l'aide de cette image, et exécute les commandes restantes.

La commande **RUN** ajoute un utilisateur nommé `johndoe` dont le mot de passe est `passw0rd`, et la commande **COPY** ajoute le fichier `config.mqsc` dans un emplacement spécifique connu de l'image parent.

Remarque : Sous Red Hat Enterprise Linux (RHEL), vous utilisez la commande **docker** (RHEL V7) ou **podman** (RHEL V7 ou RHEL V8). Dans le cas de **podman**, il n'est pas nécessaire d'indiquer **sudo** au début de la commande.

4. Exécutez votre nouvelle image personnalisée afin de créer un nouveau conteneur avec l'image de disque que vous venez de créer.

Votre nouvelle couche d'image ne spécifie pas de commande particulière à exécuter ; par conséquent, la commande est héritée de l'image parent. Le point d'entrée du parent (code disponible sur GitHub) :

- Crée un gestionnaire de files d'attente
- Démarre le gestionnaire de files d'attente
- Crée un programme d'écoute par défaut
- Exécutez ensuite les commandes MQSC à partir de `/etc/mqm/config.mqsc`.

Emettez les commandes suivantes pour exécuter votre nouvelle image personnalisée :

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

où :

Le premier paramètre env

Transmet une variable d'environnement dans le conteneur, qui reconnaît votre acceptation de la licence pour IBM IBM WebSphere MQ. Vous pouvez aussi définir la variable LICENSE afin d'afficher la licence.

Voir [Informations sur les licences IBM MQ](#) pour plus de détails sur les licences d'IBM MQ.

Le deuxième paramètre env

Définit le nom du gestionnaire de files d'attente que vous utilisez.

Le paramètre volume

Indique que le conteneur que MQ écrit dans /var/mqm doit en fait être écrit sur /var/example sur l'hôte.

Cette option signifie qu'il est facile de supprimer le conteneur ultérieurement tout en conservant les données persistantes. Elle facilite également l'affichage des fichiers journaux.

Le paramètre publish

Mappe des ports du système hôte à des ports dans le conteneur. Le conteneur s'exécute par défaut avec sa propre adresse IP interne, ce qui signifie que vous devez mapper spécifiquement tout port que vous voulez exposer.

Dans cet exemple, cela signifie que vous devez mapper le port 1414 sur l'hôte au port 1414 dans le conteneur.

Le paramètre detach

Exécute le conteneur en arrière-plan.

Résultats

Vous avez généré une image de conteneur configurée et pouvez afficher les conteneurs en cours d'exécution avec la commande `docker ps`. Vous pouvez afficher les processus IBM MQ qui s'exécutent dans votre conteneur avec la commande `docker top`.



Avertissement :

Vous pouvez afficher les journaux d'un conteneur avec la commande `docker logs $ {CONTAINER_ID}`.

Que faire ensuite

- Si votre conteneur ne s'affiche pas lorsque vous utilisez la commande `docker ps`, il se peut que le conteneur soit défaillant. Vous pouvez voir les conteneurs ayant échoué à l'aide de la commande `docker ps -a`.
- Lorsque vous utilisez la commande `docker ps -a`, l'ID de conteneur est affiché. Il l'est également lorsque vous émettez la commande `docker run`.
- Vous pouvez afficher les journaux d'un conteneur avec la commande `docker logs ${CONTAINER_ID}`.
- Vous pouvez définir le nombre maximal de fichiers ouverts à l'aide de la commande `sysctl fs.file-max=524288`.

V 9.1.0 Exécution d'applications de liaison locale dans des conteneurs distincts

Avec le partage d'espace de nom de processus entre des conteneurs dans Docker, vous pouvez exécuter des applications qui requièrent une connexion de liaison locale à IBM MQ dans des conteneurs distincts depuis le gestionnaire de files d'attente IBM MQ.

Pourquoi et quand exécuter cette tâche

Cette fonctionnalité est prise en charge dans IBM MQ 9.0.3 et dans les gestionnaires de files d'attente ultérieurs.

Vous devez respecter les restrictions suivantes :

- Vous devez partager l'espace de nom PID des conteneurs avec l'argument `--pid`.
- Vous devez partager l'espace de nom IPC des conteneurs avec l'argument `--ipc`.
- Vous devez :
 1. Partager l'espace de nom UTS des conteneurs avec l'hôte avec l'argument `--uts` ou
 2. Vous assurer que les conteneurs possèdent le même nom d'hôte avec l'argument `-h` ou `--hostname`.
- Vous devez monter le répertoire de données IBM MQ dans un volume disponible pour tous les conteneurs sous le répertoire `/var/mqm`.

Vous pouvez essayer cette fonctionnalité en effectuant les étapes ci-après sur un système Linux sur lequel Docker est déjà installé.

L'exemple ci-dessous utilise l'exemple d'image de conteneur IBM MQ. Vous trouverez les détails de cette image sur [Github](#).

Procédure

1. Créez un répertoire temporaire qui servira de volume en émettant la commande suivante :

```
mkdir /tmp/dockerVolume
```

2. Créez un gestionnaire de files d'attente (QM1) dans un conteneur, avec le nom `sharedNamespace`, en émettant la commande suivante :

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

3. Démarrez un deuxième conteneur nommé `secondaryContainer`, qui repose sur `ibmcom/mq`, sans créer de gestionnaire de files d'attente, en émettant la commande suivante :

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

4. Exécutez la commande `dspmqr` dans le deuxième conteneur pour afficher le statut des deux gestionnaires de files d'attente en émettant la commande suivante :

```
docker exec secondaryContainer dspmqr
```

5. Exécutez la commande suivante afin de traiter les commandes MQSC pour le gestionnaire de files d'attente s'exécutant dans l'autre conteneur :

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

Résultats

Désormais, vous disposez d'applications locales qui s'exécutent dans des conteneurs distincts et vous pouvez exécuter des commandes telles que **dspmq**, **amqsput**, **amqsget** et **runmqsc** en tant que liaisons locales pour le gestionnaire de files d'attente QM1 depuis le deuxième conteneur.

Si les résultats ne sont pas ceux que vous attendiez, voir [«Traitement des incidents liés à vos applications d'espace de nom»](#), à la page 40 pour plus d'informations.

V 9.1.0 Traitement des incidents liés à vos applications d'espace de nom

Lorsque vous utilisez des espaces de nom partagés, vous devez vous assurer que vous partagez tous les espaces de nom (IPC, PID et UTS/nom d'hôte) et tous les volumes montés ; si tel n'est pas le cas, vos applications ne fonctionneront pas.

Voir [«Exécution d'applications de liaison locale dans des conteneurs distincts»](#), à la page 39 pour la liste des restrictions à respecter.

Si votre application ne répond pas à toutes les restrictions répertoriées, il se peut que vous rencontriez des problèmes. Par exemple, le conteneur pourra démarrer, mais la fonctionnalité que vous attendez ne fonctionnera pas.

La liste ci-après met en évidence certaines causes communes et le comportement qui peut découler du non-respect de l'une des restrictions.

- Si vous oubliez de partager l'espace de nom (UTS/PID/IPC) ou le nom d'hôte des conteneurs et que vous montez le volume, votre conteneur pourra voir le gestionnaire de files d'attente mais ne pourra pas interagir avec lui.
 - Pour les commandes **dspmq**, le code suivant s'affiche :

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

- Pour les commandes **runmqsc** ou d'autres commandes qui tentent d'établir la connexion au gestionnaire de files d'attente, vous êtes susceptible de recevoir le message d'erreur AMQ8146 :

```
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
```

- Si vous partagez tous les espaces de nom requis mais que vous ne montez pas de volume partagé dans le répertoire `/var/mqm` et que vous disposez d'un chemin de données IBM MQ valide, vos commandes reçoivent également des messages d'erreur AMQ8146.

Toutefois, **dspmq** ne peut pas voir votre gestionnaire de files d'attente et il renvoie une réponse vierge à la place :

```
docker exec container dspmq
```

- Si vous partagez tous les espaces de nom requis, mais que vous ne montez pas de volume partagé dans le répertoire `/var/mqm` et que vous ne disposez pas d'un chemin de données IBM MQ valide (ou d'un chemin de données IBM MQ), plusieurs erreurs se produisent, car le chemin de données est un composant clé d'une installation IBM MQ. Sans le chemin d'accès aux données, IBM MQ ne peut pas fonctionner.

Si vous exécutez l'une des commandes suivantes et que des réponses similaires aux exemples sont affichées, vérifiez que vous avez monté le répertoire ou créé un répertoire de données IBM MQ :

```
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 dlrmqm QM1  
AMQ7002: An error occurred manipulating a file.
```

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

Linux > MQ Adv. > CD > V 9.1.5 **Référence d'API pour IBM MQ Operator**

IBM MQ met à disposition un opérateur Kubernetes qui permet une intégration native à OpenShift Container Platform.

Linux > MQ Adv. > CD > V 9.1.5 **Référence d'API pour mq.ibm.com/v1beta1**

Vous pouvez utiliser l'API v1beta1 pour créer et gérer des ressources QueueManager.

Linux > MQ Adv. > CD > V 9.1.5 **Référence relative à l'octroi de licence pour mq.ibm.com/v1beta1**

La zone `spec.license.license` doit contenir l'identificateur de licence de la licence que vous acceptez. Les valeurs valides sont les suivantes :

Valeur de <code>spec.license.license</code>	Valeur de <code>spec.license.use</code>	Informations sur la licence
L-RJON-BN7PN3	Production ou NonProduction	IBM Cloud Pak for Integration 2020.2
L-RJON-BPHL2Y		IBM Cloud Pak for Integration Edition limitée 2020.2
L-APIG-BJAKBF	Production ou Development	IBM MQ Advanced 9.1.5
L-APIG-BM7GDH	Development	IBM MQ Advanced for Developers 9.1.5

Notez que la *version* de licence est spécifiée, et qu'elle n'est pas toujours identique à la version d'IBM MQ.

Référence d'API pour le gestionnaire de files d'attente (mq.ibm.com/v1beta1)

QueueManager

Un gestionnaire de files d'attente est un serveur IBM MQ qui fournit des services de mise en file d'attente et de publication/abonnement à des applications.

Zone	Description
Chaîne apiVersion	APIVersion définit le schéma versionné de cette représentation d'un objet. Les serveurs doivent convertir les schémas reconnus dans la valeur interne la plus récente et peuvent rejeter les valeurs non reconnues. Pour plus d'informations : https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources .
Chaîne kind	Kind est une valeur de chaîne qui représente la ressource REST représentée par cet objet. Les serveurs peuvent déduire cette valeur du noeud final auquel le client soumet les demandes. Elle ne peut pas être mise à jour. Casse mixte. Pour plus d'informations : https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds .
metadata	
spec « Spécification QueueManager », à la page 46	Etat souhaité pour le gestionnaire de files d'attente.
status « QueueManagerStatus », à la page 47	Etat observé pour le gestionnaire de files d'attente.

Disponibilité

Paramètres de disponibilité pour le gestionnaire de files d'attente, indiquant par exemple si une paire actif-secours doit être utilisée.

Apparaît dans :

- «[Configuration de QueueManager](#)», à la page [44](#)

Zone	Description
Chaîne type	Type de disponibilité à utiliser. Utilisez "SingleInstance" pour un seul pod, qui sera redémarré automatiquement (dans certains cas) par Kubernetes. Utilisez "MultiInstance" pour une paire de pods, dont l'un est le gestionnaire de files d'attente "actif" et l'autre est un gestionnaire de files d'attente de secours. Voir Haute disponibilité pour IBM MQ dans les conteneurs dans la dernière version d' IBM MQ.

Licence

Paramètres contrôlant votre acceptation de la licence et les métriques de licence à utiliser.

Apparaît dans :

- «[Spécification QueueManager](#)», à la page [46](#)

Zone	Description
Chaîne use	Paramètre qui contrôle la façon dont le logiciel sera utilisé, où la licence prend en charge plusieurs utilisations. Pour connaître les valeurs admises, voir https://ibm.biz/BdqvCF .
Valeur booléenne accept	Indique si vous acceptez ou non la licence associée à ce logiciel (requis).
Chaîne license	Identificateur de la licence que vous acceptez. Il doit s'agir de l'identificateur de licence correct pour la version de MQ que vous utilisez. Pour connaître les valeurs admises, voir https://ibm.biz/BdqvCF .
Chaîne metric	Paramètre qui spécifie la métrique de licence à utiliser. Par exemple, "ProcessorValueUnit", "VirtualProcessorCore" ou "ManagedVirtualServer".

Limites

QueueManagerResourceList définit les paramètres d'UC et de mémoire.

Apparaît dans :

- «Ressources», à la page 50

Zone	Description
cpu	
memory	

Référence LocalObject

LocalObjectReference contient suffisamment d'informations pour vous permettre de localiser l'objet référencé dans le même espace de nom.

Apparaît dans :

- «Spécification QueueManager», à la page 46

Zone	Description
Chaîne name	Nom du référent. Pour plus d'informations : https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names :NONE.

PKI

Paramètres de l'infrastructure à clés publiques (PKI) pour la définition de clés et de certificats à utiliser avec Transport Layer Security (TLS) ou MQ Advanced Message Security (AMS).

Apparaît dans :

- «Spécification QueueManager», à la page 46

Zone	Description
Tableau keys «PKISource», à la page 43	Clés privées à ajouter au référentiel de clés du gestionnaire de files d'attente.
Tableau trust «PKISource», à la page 43	Certificats à ajouter au référentiel de clés du gestionnaire de files d'attente.

PKISource

PKISource définit une source des informations de l'infrastructure à clés publiques (PKI), comme des clés ou des certificats.

Apparaît dans :

- «PKI», à la page 43

Zone	Description
Chaîne name	Name est utilisé comme libellé pour la clé ou le certificat. Il doit s'agir d'une chaîne alphanumérique en minuscules.
secret « Secrète », à la page 50	Fournissez une clé à l'aide d'un secret Kubernetes.

Configuration de QueueManager

QueueManagerConfig définit les paramètres du conteneur de gestionnaire de files d'attente et du gestionnaire de files d'attente sous-jacent.

Apparaît dans :

- «[Spécification QueueManager](#)», à la page 46

Zone	Description
Chaîne logFormat	Indique le format de journal à utiliser pour ce conteneur. Utilisez "JSON" pour les journaux au format JSON du conteneur. Utilisez "De base" pour les messages au format texte.
metrics « Métriques QueueManager », à la page 45	Paramètres pour les métriques de style Prometheus.
readinessProbe « QueueManagerReadinessProbe », à la page 46	Paramètres qui contrôlent la sonde de vigilance.
resources « Ressources », à la page 50	Paramètres qui contrôlent les exigences relatives aux ressources.
storage « QueueManagerStockage », à la page 49	Paramètres de stockage pour contrôler l'utilisation des volumes persistants et des classes de stockage par le gestionnaire de files d'attente.
availability « Disponibilité », à la page 42	Paramètres de disponibilité pour le gestionnaire de files d'attente, indiquant par exemple si une paire actif-secours doit être utilisée.
Chaîne imagePullPolicy	Paramètre qui contrôle à quel moment le kubelet tente d'extraire l'image spécifiée.
livenessProbe « QueueManagerLivenessProbe », à la page 45	Paramètres qui contrôlent la sonde de non-défaillance.
Valeur booléenne debug	Indique si les messages de débogage du code propre au conteneur doivent être consignés ou non dans le journal du conteneur.
Chaîne image	Image de conteneur à utiliser.
Chaîne name	Nom du gestionnaire de files d'attente MQ sous-jacent, s'il est différent de metadata.name. Servez-vous de cette zone si vous voulez utiliser un nom de gestionnaire de files d'attente qui n'est pas conforme aux règles Kubernetes relatives aux noms (par exemple, un nom incluant des majuscules).

QueueManagerLivenessProbe

Paramètres qui contrôlent la sonde de non-défaillance.

Apparaît dans :

- «Configuration de QueueManager», à la page 44

Zone	Description
Entier <code>failureThreshold</code>	Nombre minimal d'échecs consécutifs nécessaire pour que l'on considère que la sonde a échoué après une réussite.
Entier <code>initialDelaySeconds</code>	Nombre de secondes après le démarrage du conteneur avant que des sondes de non-défaillance ne soient initiées. Pour plus d'informations : https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
Entier <code>periodSeconds</code>	Fréquence d'exécution de la sonde (en secondes).
Entier <code>successThreshold</code>	Nombre minimal de réussites consécutives nécessaire pour que l'on considère que la sonde a abouti après un échec.
Entier <code>timeoutSeconds</code>	Délai d'expiration de la sonde, en secondes. Pour plus d'informations : https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

Métriques QueueManager

Paramètres pour les métriques de style Prometheus.

Apparaît dans :

- «Configuration de QueueManager», à la page 44

Zone	Description
Valeur booléenne <code>enabled</code>	Indique si un noeud final doit être activé pour permettre à Prometheus de collecter des métriques compatibles avec Prometheus.

QueueManagerOptionalVolume

Détails du volume persistant pour les journaux de reprise MQ. Requis en cas d'utilisation d'un gestionnaire de files d'attente multi-instance.

Apparaît dans :

- «QueueManagerStockage», à la page 49

Zone	Description
Chaîne <code>class</code>	Classe d'archivage à utiliser pour ce volume. Valide uniquement si "type" est "persistent-claim".
Valeur booléenne <code>enabled</code>	Indique si ce volume doit être activé en tant que volume distinct ou s'il doit être placé sur le volume "queueManager" par défaut.
Chaîne <code>size</code>	Taille de PersistentVolume à transmettre à Kubernetes. Valide uniquement si "type" est "persistent-claim".
Chaîne <code>sizeLimit</code>	Limite de taille lors de l'utilisation d'un volume "éphémère". Les fichiers continuent d'être écrits dans un répertoire temporaire ; ainsi, vous pouvez utiliser cette option pour limiter la taille. Valide uniquement si type est "éphémère".

Zone	Description
Chaîne type	Type de volume à utiliser. Choisissez ephemeral pour créer un volume "emptyDir" non persistant ou persistent-claim pour utiliser un volume persistant.

QueueManagerReadinessProbe

Paramètres qui contrôlent la sonde de vigilance.

Apparaît dans :

- «Configuration de QueueManager», à la page 44

Zone	Description
Entier failureThreshold	Nombre minimal d'échecs consécutifs nécessaire pour que l'on considère que la sonde a échoué après une réussite.
Entier initialDelaySeconds	Nombre de secondes après le démarrage du conteneur avant que des sondes de non-défaillance ne soient initiées. Pour plus d'informations : https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
Entier periodSeconds	Fréquence d'exécution de la sonde (en secondes).
Entier successThreshold	Nombre minimal de réussites consécutives nécessaire pour que l'on considère que la sonde a abouti après un échec.
Entier timeoutSeconds	Délai d'expiration de la sonde, en secondes. Pour plus d'informations : https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

Spécification QueueManager

Etat souhaité pour le gestionnaire de files d'attente.

Apparaît dans :

- «QueueManager», à la page 42

Zone	Description
license «Licence», à la page 42	Paramètres contrôlant votre acceptation de la licence et les métriques de licence à utiliser.
pki «PKI», à la page 43	Paramètres de l'infrastructure à clés publiques (PKI) pour la définition de clés et de certificats à utiliser avec Transport Layer Security (TLS) ou MQ Advanced Message Security (AMS).
queueManager «Configuration de QueueManager», à la page 44	QueueManagerConfig définit les paramètres du conteneur de gestionnaire de files d'attente et du gestionnaire de files d'attente sous-jacent.
securityContext «SecurityContext», à la page 50	Paramètres de sécurité à ajouter au pod securityContext du gestionnaire de files d'attente.
tracing «TracingConfig», à la page 52	Paramètres de traçage de l'intégration au tableau de bord des opérations de Cloud Pak for Integration.

Zone	Description
Chaîne version	Paramètre qui contrôle la version de MQ qui sera utilisée (requis). Par exemple: "9.1.5.0-r2" spécifierait MQ version 9.1.5.0, à l'aide de la deuxième révision de l'image de conteneur. Les correctifs propres au conteneur sont souvent appliqués dans des révisions, comme les correctifs de l'image de base.
affinity	Règles d'affinité Kubernetes standard. Pour plus d'informations, voir https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#affinity-v1-core .
Tableau imagePullSecrets «Référence LocalObject», à la page 43	Liste facultative de références à des secrets dans le même espace de nom, à utiliser pour extraire les images utilisées par ce gestionnaire de files d'attente. Si cette liste est spécifiée, les secrets sont transmis à des implémentations de programme d'extraction individuelles pour que celles-ci puissent les utiliser. Par exemple, dans le cas de docker, seuls les secrets de type DockerConfig sont honorés. Pour plus d'informations, voir https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod .
template «Modèle», à la page 51	Création avancée de modèle pour les ressources Kubernetes. Le modèle permet aux utilisateurs d'indiquer comment IBM MQ génère les ressources Kubernetes sous-jacentes, telles que les objets StatefulSet, Pods et Services. Ce paramètre est réservé aux utilisateurs avancés, car il peut interrompre le fonctionnement normal de MQ s'il n'est pas utilisé correctement. Toute valeur spécifiée ailleurs dans la ressource QueueManager sera remplacée par les paramètres figurant dans le modèle.
Entier terminationGracePeriod Seconds	Durée facultative en secondes au bout de laquelle le pod doit s'arrêter correctement. La valeur doit être un entier non négatif. La valeur zéro indique la suppression immédiate. Heure cible à laquelle l'arrêt du gestionnaire de files d'attente est tenté, avec l'escalade des phases de la déconnexion des applications. Les tâches de maintenance essentielles du gestionnaire de files d'attente sont interrompues si nécessaire.
web «Configuration de WebServer», à la page 53	Paramètres pour le serveur Web MQ.

QueueManagerStatus

Etat observé pour le gestionnaire de files d'attente.

Apparaît dans :

- «QueueManager», à la page 42

Zone	Description
Tableau endpoints «QueueManagerStatusEndpoint», à la page 48	Informations sur les noeuds finaux que ce gestionnaire de files d'attente expose, tels que les noeuds finaux de l'API ou de l'interface utilisateur.
Chaîne name	Nom du gestionnaire de files d'attente.
versions «QueueManagerStatusVersion», à la page 48	Version de MQ utilisée et autres versions disponibles depuis le registre autorisé IBM.
Chaîne adminUiUrl	URL de l'interface utilisateur d'administration.

Zone	Description
Tableau conditions « QueueManagerStatusCondition », à la page 48	Les conditions représentent les dernières observations disponibles de l'état du gestionnaire de files d'attente.

QueueManagerStatusCondition

QueueManagerStatusCondition définit les conditions du gestionnaire de files d'attente.

Apparaît dans :

- «[QueueManagerStatus](#)», à la page 47

Zone	Description
Chaîne message	Message lisible par l'utilisateur qui présente des détails sur la dernière transition.
Chaîne type	Type de condition.
Chaîne lastTransitionTime	Dernière transition de la condition d'un statut à un autre.

QueueManagerStatusEndpoint

QueueManagerStatusEndpoint définit les noeuds finaux pour le gestionnaire de files d'attente.

Apparaît dans :

- «[QueueManagerStatus](#)», à la page 47

Zone	Description
Chaîne name	Nom du noeud final.
Chaîne type	Type du noeud final, par exemple 'UI' pour un noeud final d'interface utilisateur, 'API' pour un noeud final d'API, 'OpenAPI' pour la documentation de l'API.
Chaîne uri	URI du noeud final.

QueueManagerStatusVersion

Version de MQ utilisée et autres versions disponibles depuis le registre autorisé IBM.

Apparaît dans :

- «[QueueManagerStatus](#)», à la page 47

Zone	Description
available « QueueManagerStatusVersionDisponible », à la page 48	Autres versions de MQ disponibles depuis le registre autorisé IBM.
Chaîne reconciled	Version spécifique d'IBM MQ qui est utilisée. Si une image personnalisée est spécifiée, cette valeur peut ne pas correspondre à la version de MQ réellement utilisée.

QueueManagerStatusVersionDisponible

Autres versions de MQ disponibles depuis le registre autorisé IBM.

Apparaît dans :

- «QueueManagerStatusVersion», à la page 48

Zone	Description
Tableau channels	Canaux disponibles pour la mise à jour automatique de la version de MQ.
Tableau versions «Versions», à la page 52	Versions spécifiques de MQ qui sont disponibles.

QueueManagerStockage

Paramètres de stockage pour contrôler l'utilisation des volumes persistants et des classes de stockage par le gestionnaire de files d'attente.

Apparaît dans :

- «Configuration de QueueManager», à la page 44

Zone	Description
persistedData «QueueManagerOptionalVolume», à la page 45	Détails du volume persistant pour les données conservées par MQ, notamment la configuration, les files d'attente et les messages. Requis en cas d'utilisation d'un gestionnaire de files d'attente multi-instance.
queueManager «Volume QueueManager», à la page 49	Volume persistant par défaut pour les données normalement sous /var/mqm. Il contient toutes les données conservées et tous les journaux de reprise, si aucun autre volume n'est spécifié.
recoveryLogs «QueueManagerOptionalVolume», à la page 45	Détails du volume persistant pour les journaux de reprise MQ. Requis en cas d'utilisation d'un gestionnaire de files d'attente multi-instance.

Volume QueueManager

Volume persistant par défaut pour les données normalement sous /var/mqm. Il contient toutes les données conservées et tous les journaux de reprise, si aucun autre volume n'est spécifié.

Apparaît dans :

- «QueueManagerStockage», à la page 49

Zone	Description
Chaîne class	Classe d'archivage à utiliser pour ce volume. Valide uniquement si "type" est "persistent-claim".
Chaîne size	Taille de PersistentVolume à transmettre à Kubernetes. Valide uniquement si "type" est "persistent-claim".
Chaîne sizeLimit	Limite de taille lors de l'utilisation d'un volume "éphémère". Les fichiers continuent d'être écrits dans un répertoire temporaire ; ainsi, vous pouvez utiliser cette option pour limiter la taille. Valide uniquement si type est "éphémère".
Chaîne type	Type de volume à utiliser. Choisissez ephemeral pour créer un volume "emptyDir" non persistant ou persistent-claim pour utiliser un volume persistant.

Demandes

QueueManagerResourceList définit les paramètres d'UC et de mémoire.

Apparaît dans :

- [«Ressources», à la page 50](#)

Zone	Description
memory	
cpu	

Ressources

Paramètres qui contrôlent les exigences relatives aux ressources.

Apparaît dans :

- [«Configuration de QueueManager», à la page 44](#)

Zone	Description
limits «Limites», à la page 43	QueueManagerResourceList définit les paramètres d'UC et de mémoire.
requests «Demandes», à la page 49	QueueManagerResourceList définit les paramètres d'UC et de mémoire.

Secrète

Fournissez une clé à l'aide d'un secret Kubernetes.

Apparaît dans :

- [«PKISource», à la page 43](#)

Zone	Description
Tableau items	Clés dans le secret Kubernetes qui doivent être ajoutées au conteneur du gestionnaire de files d'attente.
Chaîne secretName	Nom du secret Kubernetes.

SecurityContext

Paramètres de sécurité à ajouter au pod securityContext du gestionnaire de files d'attente.

Apparaît dans :

- [«Spécification QueueManager», à la page 46](#)

Zone	Description
Tableau supplementalGroups	Une liste des groupes appliqués au premier processus s'exécute dans chaque conteneur, en plus du GID principal du conteneur. Si cette liste n'est pas spécifiée, aucun groupe n'est ajouté à aucun conteneur.
Entier fsGroup	Groupe supplémentaire spécial qui s'applique à tous les conteneurs dans un pod. Certains types de volume permettent à Kubelet de changer la propriété de ce volume pour être la propriété de la nacelle : 1. Le GID propriétaire sera le FSGroup 2. Le bit setgid est défini (les nouveaux fichiers créés dans le volume seront la propriété de FSGroup) 3. Les bits d'autorisation sont sujets d'un OR avec rw-rw ---- Si la configuration est annulée, Kubelet ne modifiera pas la propriété et les autorisations de n'importe quel volume.

Zone	Description
Valeur booléenne initVolumeAsRoot	Elle a un impact sur le contexte de sécurité utilisé par le conteneur qui initialise le volume persistant. Définissez ce paramètre sur "true" si vous utilisez un fournisseur de stockage qui requiert que vous soyez l'utilisateur root pour accéder aux volumes nouvellement mis à disposition. La valeur "true" affecte l'objet SCC (Security Context Constraints) que vous pouvez utiliser et le démarrage du gestionnaire de files d'attente peut échouer si vous n'êtes pas autorisé à utiliser une contrainte SCC qui autorise l'utilisateur root. Pour plus d'informations, voir https://docs.openshift.com/container-platform/latest/authentication/managing-security-context-constraints.html .

Modèle

Création avancée de modèle pour les ressources Kubernetes. Le modèle permet aux utilisateurs d'indiquer comment IBM MQ génère les ressources Kubernetes sous-jacentes, telles que les objets StatefulSet, Pods et Services. Ce paramètre est réservé aux utilisateurs avancés, car il peut interrompre le fonctionnement normal de MQ s'il n'est pas utilisé correctement. Toute valeur spécifiée ailleurs dans la ressource QueueManager sera remplacée par les paramètres figurant dans le modèle.

Apparaît dans :

- «Spécification QueueManager», à la page 46

Zone	Description
pod	Substitutions pour le modèle utilisé pour le pod. Voir https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#podspec-v1-core .

TracingAgent

Dans Cloud Pak for Integration uniquement, vous pouvez configurer des paramètres pour l'agent de trace facultatif.

Apparaît dans :

- «TracingConfig», à la page 52

Zone	Description
Chaîne image	Image de conteneur à utiliser.
Chaîne imagePullPolicy	Paramètre qui contrôle à quel moment le kubelet tente d'extraire l'image spécifiée.
livenessProbe «TracingProbe», à la page 52	Paramètres qui contrôlent la sonde de non-défaillance.
readinessProbe «TracingProbe», à la page 52	Paramètres qui contrôlent la sonde de vigilance.

TracingCollector

Dans Cloud Pak for Integration uniquement, vous pouvez configurer des paramètres pour le collecteur de trace facultatif.

Apparaît dans :

- «TracingConfig», à la page 52

Zone	Description
Chaîne image	Image de conteneur à utiliser.

Zone	Description
Chaîne <code>imagePullPolicy</code>	Paramètre qui contrôle à quel moment le kubelet tente d'extraire l'image spécifiée.
<code>livenessProbe</code> « <code>TracingProbe</code> », à la page 52	Paramètres qui contrôlent la sonde de non-défaillance.
<code>readinessProbe</code> « <code>TracingProbe</code> », à la page 52	Paramètres qui contrôlent la sonde de vigilance.

TracingConfig

Paramètres de traçage de l'intégration au tableau de bord des opérations de Cloud Pak for Integration.

Apparaît dans :

- «[Spécification QueueManager](#)», à la page 46

Zone	Description
agent « <code>TracingAgent</code> », à la page 51	Dans Cloud Pak for Integration uniquement, vous pouvez configurer des paramètres pour l'agent de trace facultatif.
collector « <code>TracingCollector</code> », à la page 51	Dans Cloud Pak for Integration uniquement, vous pouvez configurer des paramètres pour le collecteur de trace facultatif.
Valeur booléenne <code>enabled</code>	Indique si l'intégration au tableau de bord des opérations de Cloud Pak for Integration doit être activée ou non, via la fonction de trace.
Chaîne <code>namespace</code>	Espace de nom dans lequel le tableau de bord des opérations de Cloud Pak for Integration est installé.

TracingProbe

Paramètres qui contrôlent la sonde de vigilance.

Apparaît dans :

- «[TracingCollector](#)», à la page 51

Zone	Description
Entier <code>failureThreshold</code>	Nombre minimal d'échecs consécutifs nécessaire pour que l'on considère que la sonde a échoué après une réussite.
Entier <code>initialDelaySeconds</code>	Nombre de secondes après le démarrage du conteneur avant que des sondes de non-défaillance ne soient initiées. Pour plus d'informations : https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
Entier <code>periodSeconds</code>	Fréquence d'exécution de la sonde (en secondes).
Entier <code>successThreshold</code>	Nombre minimal de réussites consécutives nécessaire pour que l'on considère que la sonde a abouti après un échec.
Entier <code>timeoutSeconds</code>	Délai d'expiration de la sonde, en secondes. Pour plus d'informations : https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

Versions

`QueueManagerStatusVersion` définit une version de MQ.

Apparaît dans :

- «[QueueManagerStatusVersionDisponible](#)», à la page 48

Zone	Description
Chaîne name	Version "name" pour cette version de QueueManager. Valeurs valides pour la zone <code>spec.version</code> .

Configuration de WebServer

Paramètres pour le serveur Web MQ.

Apparaît dans :

- «[Spécification QueueManager](#)», à la page 46

Zone	Description
Valeur booléenne enabled	Indique si le serveur Web doit être activé ou non.

Remarques

:NONE.

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non annoncés dans ce pays. Pour plus de détails, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial IBM. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service IBM puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

IBM EMEA Director of Licensing
IBM Corporation
Tour Descartes
Armonk, NY 10504-1785
U.S.A.

Pour toute demande d'informations relatives au jeu de caractères codé sur deux octets, contactez le service de propriété intellectuelle IBM ou envoyez vos questions par courrier à l'adresse suivante :

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japon

Le paragraphe suivant ne s'applique ni au Royaume-Uni, ni dans aucun pays dans lequel il serait contraire aux lois locales. LE PRESENT DOCUMENT EST LIVRE "EN L'ETAT" SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFACON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Le présent document peut contenir des inexactitudes ou des coquilles. Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut, à tout moment et sans préavis, modifier les produits et logiciels décrits dans ce document.

Les références à des sites Web non IBM sont fournies à titre d'information uniquement et n'impliquent en aucun cas une adhésion aux données qu'ils contiennent. Les éléments figurant sur ces sites Web ne font pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation de sa part, tout ou partie des informations qui lui seront fournies.

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

IBM Corporation
Coordinateur d'interopérabilité logicielle, département 49XA
3605 Autoroute 52 N

Rochester, MN 55901
U.S.A.

Ces informations peuvent être soumises à des conditions particulières, prévoyant notamment le paiement d'une redevance.

Le logiciel sous licence décrit dans le présent document et tous les éléments sous disponibles s'y rapportant sont fournis par IBM conformément aux dispositions du Contrat sur les produits et services IBM, aux Conditions Internationales d'Utilisation de Logiciels IBM ou de tout autre accord équivalent.

Les données de performance indiquées dans ce document ont été déterminées dans un environnement contrôlé. Par conséquent, les résultats peuvent varier de manière significative selon l'environnement d'exploitation utilisé. Certaines mesures évaluées sur des systèmes en cours de développement ne sont pas garanties sur tous les systèmes disponibles. En outre, elles peuvent résulter d'extrapolations. Les résultats peuvent donc varier. Il incombe aux utilisateurs de ce document de vérifier si ces données sont applicables à leur environnement d'exploitation.

Les informations concernant des produits non IBM ont été obtenues auprès des fournisseurs de ces produits, par l'intermédiaire d'annonces publiques ou via d'autres sources disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Elle ne peut recevoir aucune réclamation concernant des produits non IBM. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

Toute instruction relative aux intentions d'IBM pour ses opérations à venir est susceptible d'être modifiée ou annulée sans préavis, et doit être considérée uniquement comme un objectif.

Le présent document peut contenir des exemples de données et de rapports utilisés couramment dans l'environnement professionnel. Ces exemples mentionnent des noms fictifs de personnes, de sociétés, de marques ou de produits à des fins illustratives ou explicatives uniquement. Toute ressemblance avec des noms de personnes, de sociétés ou des données réelles serait purement fortuite.

Licence sur les droits d'auteur :

Le présent logiciel contient des exemples de programmes d'application en langage source destinés à illustrer les techniques de programmation sur différentes plateformes d'exploitation. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation des plateformes pour lesquels ils ont été écrits ou aux interfaces de programmation IBM. Ces exemples de programmes n'ont pas été rigoureusement testés dans toutes les conditions. Par conséquent, IBM ne peut garantir expressément ou implicitement la fiabilité, la maintenabilité ou le fonctionnement de ces programmes.

Si vous visualisez ces informations en ligne, il se peut que les photographies et illustrations en couleur n'apparaissent pas à l'écran.

Documentation sur l'interface de programmation

Les informations d'interface de programmation, si elles sont fournies, sont destinées à vous aider à créer un logiciel d'application à utiliser avec ce programme.

Ce manuel contient des informations sur les interfaces de programmation prévues qui permettent au client d'écrire des programmes pour obtenir les services de WebSphere MQ.

Toutefois, lesdites informations peuvent également contenir des données de diagnostic, de modification et d'optimisation. Ces données vous permettent de déboguer votre application.

Important : N'utilisez pas ces informations de diagnostic, de modification et d'optimisation en tant qu'interface de programmation car elles sont susceptibles d'être modifiées.

Marques

IBM, le logo IBM, ibm.com, sont des marques d'IBM Corporation dans de nombreux pays. La liste actualisée de toutes les marques d'IBM est disponible sur la page Web "Copyright and trademark".

information"www.ibm.com/legal/copytrade.shtml. Les autres noms de produits et de services peuvent être des marques d'IBM ou d'autres sociétés.

Microsoft et Windows sont des marques de Microsoft Corporation aux Etats-Unis et/ou dans d'autres pays.

UNIX est une marque de The Open Group aux Etats-Unis et dans certains autres pays.

Linux est une marque de Linus Torvalds aux Etats-Unis et/ou dans certains autres pays.

Ce produit inclut des logiciels développés par le projet Eclipse (<http://www.eclipse.org/>).

Java ainsi que tous les logos et toutes les marques incluant Java sont des marques d'Oracle et/ou de ses sociétés affiliées.



Référence :

(1P) P/N: