9.4

IBM MQ dans des conteneurs

Remarque

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section «Remarques», à la page 183.

Cette édition s'applique à la version 9 édition 4 d' IBM® MQ et à toutes les éditions et modifications ultérieures, sauf indication contraire dans les nouvelles éditions.

Table des matières

IBM MQ en conteneurs et IBM Cloud Pak for Integration

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 en utilisant le IBM MQ Operator sur Red Hat® OpenShift®. Pour ce faire, utilisez IBM Cloud Pak for Integration, IBM MQ Advanced ou IBM MQ Advanced for Developers.

Vous pouvez aussi exécuter IBM MQ dans un conteneur que vous générez.

Pour plus d'informations sur le IBM MQ Operator, voir les liens suivants :

A propos de IBM MQ dans les conteneurs

Informations de présentation pour vous aider à démarrer avec IBM MQ dans des conteneurs.

Les conteneurs sont une technologie permettant le conditionnement et l'isolement du code avec son environnement d'exécution, qui peut être exécuté d'une manière qui est isolée d'autres logiciels sur la même infrastructure. Cela facilite le déplacement d'un gestionnaire de files d'attente ou d'une application entre des environnements (tels que le développement, le test et la production). Les orchestrateurs de conteneurs modernes, tels que Red Hat OpenShift Container Platform et Kubernetes , peuvent exécuter de nombreux types de conteneurs sur la même machine, chacun étant isolé les uns des autres en termes de ressources, de sécurité et d'échecs.

Vous pouvez exécuter des gestionnaires de files d'attente IBM MQ ou vos applications IBM MQ dans des conteneurs.

Information associée

Que sont les conteneurs ?

Historique des éditions de IBM MQ Operator

Remarques :

- Pour plus d'informations sur les opérateurs IBM MQ précédents, voir Release history for IBM MQ Operator dans la documentation IBM MQ 9.3 .
- Pour plus d'informations sur les mises à jour futures de IBM MQ , voir la page IBM MQ Recommended Fixes and Planifié Maintenance release dates .

IBM MQ Operator 3.2.1

Version de IBM Cloud Pak for Integration
IBM Cloud Pak for Integration 16.1.0

Canal opérateur
`v3.2-sc2`

Valeurs autorisées pour `.spec.version`
9.4.0.0-r1

Valeurs autorisées pour `.spec.version` lors de la migration
9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3,9.3.3.2-r3 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1, 9.3.0.16-r2, 9.3.0.17-r1, 9.3.0.17-r2, 9.3.0.17-r3, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3, 9.3.3.3-r1, 9.3.3.3-r2, 9.3.4.0-r1, 9.3.4.1-r1, 9.3.5.0-r1, 9.3.5.0-r2, 9.3.5.1-r1, 9.3.5.1-r2

Versions Red Hat OpenShift Container Platform
OpenShift Container Platform 4.12 et versions ultérieures.

Versions IBM Cloud Pak foundational services
IBM Cloud Pak foundational services version 4.6 uniquement.

Modifications

- Résout un problème dans OpenShift Container Platform 4.12 où la mise à niveau vers le canal v3.2-sc2 peut entraîner un comportement inattendu pour les utilisateurs IBM Cloud Pak for Integration . Pour plus d'informations, voir Mise à niveau à partir de 2023.4 dans la documentation IBM Cloud Pak for Integration .

IBM MQ Operator 3.2.0

Version de IBM Cloud Pak for Integration
IBM Cloud Pak for Integration 16.1.0

Canal opérateur
`v3.2-sc2`

Valeurs autorisées pour `.spec.version`
9.4.0.0-r1

Valeurs autorisées pour `.spec.version` lors de la migration
9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3,9.3.3.2-r3 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1, 9.3.0.16-r2, 9.3.0.17-r1, 9.3.0.17-r2, 9.3.0.17-r3, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3, 9.3.3.3-r1, 9.3.3.3-r2, 9.3.4.0-r1, 9.3.4.1-r1, 9.3.5.0-r1, 9.3.5.0-r2, 9.3.5.1-r1, 9.3.5.1-r2

Versions Red Hat OpenShift Container Platform
OpenShift Container Platform 4.12 et versions ultérieures.

Versions IBM Cloud Pak foundational services
IBM Cloud Pak foundational services version 4.6 uniquement.

Nouveautés

- Désormais, «Extension des volumes persistants», à la page 104 est pris en charge.
- Les gestionnaires de files d'attente peuvent désormais être arrêtés en ajoutant l'annotation `mq.ibm.com/stop` et en la définissant sur `true`. Voir «Arrêt d'un gestionnaire de files d'attente (mq.ibm.com/stop)», à la page 108.

  Remarques :

  - Un gestionnaire de files d'attente arrêté comporte la zone *.replicas* dans son `StatefulSet` définie sur 0.
  - Etant donné que IBM MQ Operator gère désormais activement la zone *.replicas* dans `StatefulSet`, si vous modifiez cette zone, elle est immédiatement annulée par l'opérateur.
  - Les anciennes versions de IBM MQ passent à l'état'Echec'si vous modifiez la zone *.replicas* tout en conservant la valeur modifiée. Si vos procédures d'exploitation existantes reposent sur ce comportement, à partir de IBM MQ 9.4 , vous devez utiliser l'annotation `mq.ibm.com/stop` .

Modifications

- Les éditions impaires de Red Hat OpenShift Container Platform sont désormais prises en charge.
- IBM MQ L'image de catalogue a été déplacée vers le format de catalogue basé sur un fichier à partir du format de base de données SQLite .
- Basé sur Red Hat Universal Base Image 9.4-949.1716471857. **Remarque:** UBI 9 possède une certification FIPS 140-3 en attente. UBI 9 n'est pas pris en charge sur l'architecture Power 8.
- Les vulnérabilités qui sont prises en compte sont détaillées dans ce bulletin de sécurité.

OpenShift CP4I Historique des éditions des images de conteneur de gestionnaire de files d'attente à utiliser avec IBM MQ Operator

Remarque : Pour plus d'informations sur les images de conteneur de gestionnaire de files d'attente antérieures, voir Release history for IBM MQ Operator dans la documentation IBM MQ 9.3 .

9.4.0.0-r1

CD CP4I-SC2

Version de l'opérateur requise
3.2.0 ou version ultérieure

Architectures prises en charge
amd64, s390x, ppc64le

Images

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.4.0.0-r1`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.4.0.0-r1`
- `icr.io/ibm-messaging/mq:9.4.0.0-r1`

Nouveautés

- Nouveautés d' IBM MQ 9.4.0 for Multiplatforms-autorisation d'utilisation de base et Advanced

Modifications

- Ce qui a changé dans IBM MQ 9.4.0
- Deprecated Lors de l'utilisation de IBM MQ Advanced for Developers, la définition des mots de passe pour les utilisateurs `admin` et `app` via une variable d'environnement est obsolète. Utilisez des secrets à la place.
- Une nouvelle valeur facultative `mqsc` est ajoutée pour la variable d'environnement *MQ_LOGGING_CONSOLE_SOURCE*. Cette option peut être utilisée pour refléter le contenu de `autocfgmqsc.LOG` dans le journal du conteneur.
- Basé sur Red Hat Universal Base Image 9.4-949.1716471857. **Remarque:** UBI 9 possède une certification FIPS 140-3 en attente. UBI 9 n'est pas pris en charge sur l'architecture Power 8.

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 votre IBM MQ dans l'architecture des conteneurs, vous devez vous familiariser avec les concepts de base de IBM MQ (voir la Présentation technique IBM MQ) ainsi que les concepts Kubernetes/Red Hat OpenShift de base (voir Architecture OpenShift Container Platform).

Procédure

- «Comment utiliser IBM MQ dans des conteneurs», à la page 8.
- «Prise en charge de IBM MQ dans les conteneurs», à la page 8.
- «Planification du stockage pour le IBM MQ Operator», à la page 17.
- «Planification de la haute disponibilité pour IBM MQ dans des conteneurs», à la page 19.
- «Reprise après incident d'IBM MQ dans des conteneurs», à la page 25.
- «Authentification et autorisation des utilisateurs pour IBM MQ dans les conteneurs», à la page 25.

Comment utiliser IBM MQ dans des conteneurs

Il existe plusieurs options d'utilisation de IBM MQ dans des conteneurs : vous pouvez choisir d'utiliser le IBM MQ Operator, qui utilise des images de conteneur pré-conditionnées, ou créer vos propres images et code de déploiement.

Utilisation de IBM MQ Operator

Si vous prévoyez un déploiement sur Red Hat OpenShift Container Platform, vous souhaitez probablement utiliser le IBM MQ Operator.

IBM MQ Operator étend l'API Red Hat OpenShift Container Platform pour ajouter une nouvelle ressource personnalisée `QueueManager` . L'opérateur surveille les nouvelles définitions de gestionnaire de files d'attente, puis les transforme en ressources de niveau inférieur nécessaires, telles que les ressources `StatefulSet` et `Service`. Dans le cas de Native HA, l'opérateur peut également effectuer la mise à jour évolutive complexe des instances de gestionnaire de files d'attente. Voir «Remarques sur l'exécution de votre propre mise à jour en continu d'un gestionnaire de files d'attente natif de haute disponibilité», à la page 23

Certaines fonctions de IBM MQ ne sont pas prises en charge lors de l'utilisation de IBM MQ Operator. Pour plus d'informations sur les éléments pris en charge lors de l'utilisation de IBM MQ Operator, voir «Prise en charge de IBM MQ dans les conteneurs», à la page 8 .

Génération de vos propres images et code de déploiement

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 «Préparation pour IBM MQ en créant votre propre image de conteneur», à la page 57.

Pour plus d'informations sur les éléments pris en charge lors de la génération de votre propre image et code de déploiement, voir «Prise en charge de IBM MQ dans les conteneurs», à la page 8 .

Référence associée

«Prise en charge de IBM MQ dans les conteneurs», à la page 8
Toutes les fonctions IBM MQ ne sont pas disponibles et prises en charge de la même manière dans les conteneurs.


Prise en charge de IBM MQ dans les conteneurs

Toutes les fonctions IBM MQ ne sont pas disponibles et prises en charge de la même manière dans les conteneurs.

Vous trouverez ci-dessous un tableau qui montre en détail comment les fonctions IBM MQ sont prises en charge avec IBM MQ Operator, ou lorsque vous générez vos propres conteneurs et code de déploiement.

Remarques :

- Les images de conteneur IBM MQ préconfigurées sur IBM Container Registry (icr.io et cp.icr.io) ne sont prises en charge et éligibles aux correctifs qu'en cas d'utilisation avec IBM MQ Operator.
- A partir du IBM MQ Operator canal v3.2, Long Term Support (LTS) est renommé en Support Cycle 2 (SC2). En effet, le seul chemin LTS disponible pour IBM MQ dans les conteneurs est une prise en charge de deux ans sous IBM Cloud Pak for Integration , et IBM Cloud Pak for Integration a adopté le terme SC2. Voici le tableau complet des droits:

- Avec l'autorisation d'utilisation IBM MQ , IBM MQ Operator ne peut déployer que les images IBM MQ Continuous Delivery (CD).
- Avec l'autorisation d'utilisation IBM Cloud Pak for Integration , IBM MQ Operator peut déployer des images CD ou SC2 (formerly LTS) .

Il n'est pas possible de "mettre à niveau" la licence de l'image IBM MQ Advanced for Developers prégénérée vers une autre licence. Le IBM MQ Operator déploie différentes images, en fonction de la licence sélectionnée.

Dans ce tableau, les termes suivants s'appliquent:

"Code d'activation de conteneur"
Les exécutables **runmqserver**, **runmqintegrationserver**, **chkmqhealthy**, **chkmqready** et **chkmqstarted**. Ce code est fourni à titre d'exemple et n'est pris en charge que dans le cadre des conteneurs préconfigurés lorsqu'il est utilisé avec IBM MQ Operator.

	Utilisation d'IBM MQ Operator et d'une licence IBM Cloud Pak for Integration	Utilisation d'IBM MQ Operator et d'une licence IBM MQ Advanced	Utilisation d'IBM MQ Operator et d'une licence IBM MQ Advanced for Developers	Image IBM MQ Advanced for Developers prégénérée	Build-your-own container (Génération de votre propre conteneur)
Plateformes prises en charge	Pris en charge sur Red Hat OpenShift Container Platform uniquement. Les éditions de Red Hat OpenShift Container Platform ne sont plus prises en charge par IBM MQ une fois que Red Hat a arrêté la prise en charge. Voir «Versions prises en charge pour IBM MQ Operator», à la page 15 pour plus de détails.		Disponible sous Red Hat OpenShift Container Platform uniquement, mais non pris en charge.	Fonctionne sur toute plateforme Docker, containerd ou cri-o, mais n'est pas prise en charge. Pour plus de détails, voir Configuration système requise pour IBM MQ.	Toute plateforme Docker, containerd ou cri-o. Pour plus de détails, voir Configuration système requise pour IBM MQ. La haute disponibilité native est prise en charge uniquement sous Kubernetes ou Red Hat OpenShift Container Platform. L'exemple d'image de conteneur utilise un Red Hat Universal Base Image (UBI), qui inclut les bibliothèques et les utilitaires Linux® utilisés par IBM MQ. UBI est pris en charge par Red Hat lorsqu'il est exécuté sous Red Hat OpenShift. Le *code d'activation de conteneur* n'est pas pris en charge.

	Utilisation d' IBM MQ Operator et d'une licence IBM Cloud Pak for Integration	Utilisation d' IBM MQ Operator et d'une licence IBM MQ Advanced	Utilisation d' IBM MQ Operator et d'une licence IBM MQ Advanced for Developers	Image IBM MQ Advanced for Developers prégénérée	Build-your-own container (Génération de votre propre conteneur)
Architectures d'UC	Pris en charge sous `amd64` et `s390x` z /Linux. Egalement pris en charge sur les systèmes `ppc64le` Power Systems version 9 et ultérieure. Notez que tous les noeuds du cluster Red Hat OpenShift Container Platform doivent utiliser la même architecture d'UC.		Disponible sous `amd64` et `s390x` z /Linux, mais non pris en charge. Également disponible sur les systèmes `ppc64le` Power Systems version 9 et versions ultérieures, mais non pris en charge. Notez que tous les noeuds du cluster Red Hat OpenShift Container Platform doivent utiliser la même architecture d'UC.		Conformément au logiciel IBM MQ .
Durée du support	IBM Cloud Pak for Integration - Support Cycle 2 (formerly Long Term Support) ou Continuous Delivery.[1] L'opérateur CD et les gestionnaires de files d'attente sont pris en charge jusqu'à la prochaine édition de IBM Cloud Pak for Integration CD ou CP4I-SC2 . L'opérateur CP4I-SC2 et les gestionnaires de files d'attente sont pris en charge jusqu'à la prochaine édition de IBM Cloud Pak for Integration CP4I-SC2 , plus un délai de grâce pour permettre la mise à niveau.	Flux Continuous Delivery uniquement, pour le IBM MQ Operatoret les gestionnaires de files d'attente. Chaque version de IBM MQ Operator et de gestionnaire de files d'attente est uniquement prise en charge jusqu'à la prochaine édition de CD .	Non pris en charge		Conformément au logiciel IBM MQ . Voir IBM MQ -Foire aux questions pour les éditions Long Term Support et Continuous Delivery. Le *code d'activation de conteneur* n'est pas pris en charge.

	Utilisation d' IBM MQ Operator et d'une licence IBM Cloud Pak for Integration	Utilisation d' IBM MQ Operator et d'une licence IBM MQ Advanced	Utilisation d' IBM MQ Operator et d'une licence IBM MQ Advanced for Developers	Image IBM MQ Advanced for Developers prégénérée	Build-your-own container (Génération de votre propre conteneur)
Disponibilité des correctifs de sécurité	Correctifs périodiques disponibles en tant qu'images de conteneur sur IBM Container Registry				Les correctifs du logiciel IBM MQ sont disponibles en tant que logiciels sous Fix Central. Le *code d'activation de conteneur* n'est pas pris en charge.
Disponibilité des correctifs temporaires	Des correctifs de gestionnaire de files d'attente sont disponibles en tant que logiciels et une génération d'image personnalisée est nécessaire. Les correctifs IBM MQ Operator ne sont pas disponibles en tant que correctifs temporaires.		Aucun correctif temporaire disponible.		Les correctifs des logiciels IBM MQ sont disponibles sous forme de logiciels sur Fix Central ou via le support IBM . Le *code d'activation de conteneur* n'est pas pris en charge.
Fonction: Advanced Message Security	Pris en charge. Notez qu'il n'est pas facile d'utiliser le chiffrement côté serveur, car IBM MQ Operator ne vous permet pas directement de spécifier votre propre magasin de clés pour Advanced Message Security.		Disponible mais non pris en charge.		Pris en charge par le logiciel IBM MQ , mais aucun exemple n'est disponible.

[1] Le IBM MQ Operator est pris en charge en tant qu'édition IBM MQ CD ou en tant qu'édition IBM Cloud Pak for Integration - Support Cycle 2 (formerly Long Term Support) :

- Les images de conteneur IBM MQ 9.4.0.x déployées avec IBM MQ Operator 3.2.x, lorsqu'elles sont utilisées dans le cadre de IBM Cloud Pak for Integration 16.1.0, sont éligibles pour le support CP4I-LTS . L'édition la plus récente de Support Cycle 2 (SC2) du IBM MQ Operator est 3.2.1, et l'image de conteneur SC2 la plus récente est 9.4.0.0-r1.
- Les images de conteneur IBM MQ 9.4.0.x déployées avec IBM MQ Operator 3.2.x, lorsqu'elles sont utilisées dans le cadre de IBM Cloud Pak for Integration 16.1.0, sont éligibles pour le support CD . L'édition la plus récente de Continuous Delivery (CD) du IBM MQ Operator est 3.2.1, et l'image de conteneur CD la plus récente est 9.4.0.0-r1.

	Utilisation d' IBM MQ Operator et d'une licence IBM Cloud Pak for Integration	Utilisation d' IBM MQ Operator et d'une licence IBM MQ Advanced	Utilisation d' IBM MQ Operator et d'une licence IBM MQ Advanced for Developers	Image IBM MQ Advanced for Developers prégénérée	Build-your-own container (Génération de votre propre conteneur)
Fonction: Managed File Transfer	Non disponible et non pris en charge. Toutefois, vous pouvez utiliser IBM MQ Operator pour fournir un ou plusieurs gestionnaires de files d'attente de coordination, de commande ou d'agent.			Non disponible et non pris en charge.	Pris en charge par le logiciel IBM MQ , avec sample pour l'agent.
Fonction: MQTT	Non disponible et non pris en charge.				Pris en charge par le logiciel IBM MQ , mais aucun exemple n'est disponible.
Fonction: AMQP	Non disponible et non pris en charge.				Pris en charge par le logiciel IBM MQ , mais aucun exemple n'est disponible.
Fonction: REST API	Disponible et pris en charge.				Disponible et prise en charge selon les logiciels IBM MQ .
Fonction: gestionnaires de files d'attente de données répliquées	Non disponible et non pris en charge. Les gestionnaires de files d'attente de données répliquées (RDQM) sont étroitement liés au noyau Linux et ne sont pas pris en charge dans les conteneurs.				
Fonction: Native HA	Disponible et pris en charge.		Disponible, mais non pris en charge.		Disponible uniquement sous Kubernetes et Red Hat OpenShift Container Platform. Pris en charge conformément aux logiciels IBM MQ .
Fonction: gestionnaires de files d'attente multi-instance	Disponible et pris en charge.		Disponible, mais non pris en charge.		Disponible et prise en charge selon les logiciels IBM MQ .

	Utilisation d' IBM MQ Operator et d'une licence IBM Cloud Pak for Integration	**Utilisation d' IBM MQ Operator et d'une licence IBM MQ Advanced**	**Utilisation d' IBM MQ Operator et d'une licence IBM MQ Advanced for Developers**	**Image IBM MQ Advanced for Developers prégénérée**	**Build-your-own container (Génération de votre propre conteneur)**
Fonction: Types de journaux de reprise	Journalisation avec réutilisation automatique des journaux ou journaux répliqués uniquement. La journalisation linéaire n'est pas prise en charge.				Disponible et prise en charge selon les logiciels IBM MQ . Vous devez configurer les options **crtmqm** .
Fonction: spécification d'options de ligne de commande personnalisées pour crtmqdir, crtmqm, strmqm et endmqm	Non disponible et non pris en charge. La plupart des options peuvent être configurées à l'aide d'un fichier INI, mais certaines ne peuvent pas être configurées, comme l'utilisation de la journalisation linéaire.				Facultatif, selon la façon dont vous implémentez votre *code d'activation de conteneur*.
Fonction: utilisateurs du système d'exploitation	Non disponible et non pris en charge.				Possible et pris en charge selon les logiciels IBM MQ , si vous installez IBM MQ à l'aide de RPM, mais qu'aucun exemple n'est disponible. Non recommandé en raison du risque de sécurité.

Remarque : L'expression "pris en charge selon le logiciel IBM MQ " signifie que le support technique IBM est limité au logiciel IBM MQ de base qui s'exécute dans le conteneur.

Concepts associés

FAQ d'IBM MQ pour les éditions de prise en charge à long terme (Long Term Support Release, LTSR) et de distribution continue (Continuous Delivery Release, CDR)

Référence associée

IBM Cloud Pak for Integration Software Support Lifecycle Addendum

Versions prises en charge pour IBM MQ Operator

Mappage entre les versions prises en charge d'IBM MQ, OpenShift Container Platform et IBM Cloud Pak for Integration.

Versions IBM MQ disponibles

Canal opérateur	Version de l'opérateur	Versions IBM MQ						
		9.4.0	9.3.5	9.3.4	9.3.3	9.3.2	9.3.1	9.3.0
v32-sc2	3.2	CD et SC2	DEP	DEP	DEP	DEP	DEP	MIG

Clé :

- CD: la prise en charge de Continuous Delivery est disponible.
- SC2: IBM Cloud Pak for Integration - Support Cycle 2 (formerly Long Term Support) est disponible.
- MIG: disponible uniquement lors de la migration d'un opérande IBM Cloud Pak for Integration - Support Cycle 2 (formerly Long Term Support) vers un opérande Continuous Delivery .
- DEP: Deprecated Obsolète. A mesure que les éditions IBM MQ ne sont plus prises en charge, elles peuvent toujours être configurables dans l'opérateur, mais elles ne sont plus éligibles au support et peuvent être supprimées dans les éditions ultérieures.

Pour plus de détails sur chaque version, notamment les fonctionnalités détaillées, les modifications et les correctifs de chaque version, reportez-vous à la rubrique «Historique des éditions de IBM MQ Operator», à la page 5.

Versions Red Hat OpenShift Container Platform compatibles

Canal opérateur	Version de l'opérateur	Versions OpenShift Container Platform[2]		
		4.15	4.14	4.12
v3.2-sc2	3.2.0 et versions ultérieures	SC2	SC2	SC2

Clé :

- CD: la prise en charge de Continuous Delivery est disponible.
- SC2: IBM Cloud Pak for Integration - Support Cycle 2 (formerly Long Term Support) est disponible.
- EOS: n'est plus pris en charge. Migrez vers une version ultérieure de OpenShift Container Platform .

[2] Les versions OpenShift Container Platform sont soumises à leurs propres dates pour la prise en charge. Pour plus d'informations, voir OpenShift Container Platform Règle de cycle de vie.

Versions IBM Cloud Pak for Integration

Prise en charge pour une utilisation dans le cadre de IBM Cloud Pak for Integration version 16.1.0, ou indépendamment:

- IBM MQ Operator 3.2.x

Versions IBM MQ disponibles dans les anciens opérateurs

Voir Versions IBM MQ disponibles dans la documentation IBM MQ 9.3 .

Versions de OpenShift Container Platform compatibles avec les opérateurs plus anciens

Voir Versions OpenShift Container Platform compatibles dans la documentation IBM MQ 9.3 .

Edition des ressources créées par IBM MQ Operator

IBM MQ Operator synchronise une ressource personnalisée QueueManager en créant et en gérant des ressources Kubernetes natives. Ces ressources gérées **ne doivent pas** être éditées directement.

Vous pouvez généralement déterminer si une ressource appartient à une autre ressource de niveau supérieur en consultant le `ownerReferences`. Par exemple, les métadonnées suivantes extraites d'un `StatefulSet` indiquent qu'elles appartiennent à la ressource `QueueManager` "qm1":

```
metadata:
  ownerReferences:
    - apiVersion: mq.ibm.com/v1beta1
      kind: QueueManager
      name: qm1
      uid: 60fda34c-9f7c-42d2-a293-78fec4315c62
      controller: true
      blockOwnerDeletion: true
```

Notez que toutes les ressources n'ont pas ces métadonnées.

Il incombe au IBM MQ Operator de gérer les ressources sous-jacentes, telles que `StatefulSet`, `Service` et `Route`. Si vous modifiez l'une de ces ressources sous-jacentes, le IBM MQ Operator les remodifie et vous risquez de connaître un temps d'indisponibilité si cette modification nécessite une mise à jour en continu.

La plupart des paramètres importants des gestionnaires de files d'attente sont disponibles sur la ressource `QueueManager` . Toutefois, si vous constatez que vous avez besoin d'un contrôle total des ressources sous-jacentes, vous disposez de plusieurs options:

- Si vous devez remplacer les paramètres sur le pod créé par le IBM MQ Operator, vous pouvez ajouter un modèle de remplacement de pod dans la section `.spec.template` du fichier YAML `QueueManager` .
- Si vous devez remplacer les paramètres du gestionnaire de files d'attente `Route` créé par IBM MQ Operator, vous devez désactiver la route en définissant entièrement le paramètre `.spec.route.enabled` sur "false", puis créer votre propre route.
- Les paramètres tels que les libellés et les annotations, ainsi que les paramètres Pod tels que `securityContext`, peuvent tous être définis sur la ressource `QueueManager` .
- Dans d'autres cas, IBM MQ Operator peut ne pas être adapté à votre cas d'utilisation si vous avez besoin d'un contrôle total.

Planification de l'octroi de licence à IBM MQ dans des conteneurs

L'octroi de licence de conteneur vous permet d'octroyer des licences uniquement pour la capacité disponible de vos conteneurs IBM MQ individuels, au lieu de vous obliger à octroyer des licences pour l'ensemble du serveur sur lequel vos conteneurs s'exécutent. Pour tirer parti de l'octroi de licence de conteneur, le IBM License Service doit être utilisé pour suivre l'utilisation des licences et déterminer les droits requis.

Référence associée

«Annotations de licence lors de la génération de votre propre image de conteneur IBM MQ», à la page 169
Les annotations de licence vous permettent de suivre l'utilisation en fonction des limites définies sur le conteneur, plutôt que sur la machine sous-jacente. Vous configurez vos clients pour déployer le conteneur avec des annotations spécifiques que IBM License Service utilise ensuite pour suivre l'utilisation.

Information associée

Licences de conteneur IBM

Foire aux questions sur l'octroi de

Installation du service de licence

Affichage et suivi de l'utilisation des licences

OpenShift CP4I Kubernetes Planification du stockage pour le IBM MQ Operator

Le IBM MQ Operator peut être exécuté dans deux modes de stockage :

- L'option **Stockage éphémère** est utilisée lorsque toutes les informations d'état du conteneur peuvent être supprimées lors du redémarrage du conteneur. 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 d'état du gestionnaire de files d'attente sont perdues au 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)
- 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. En plus de disposer de tous les états de messagerie au redémarrage, 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 Red Hat OpenShift», à la page 97). Sinon, IBM MQ devra être configuré à chaque fois que le conteneur redémarre.

OpenShift CP4I 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 persistant

IBM MQ s'exécute normalement avec un stockage de persistance pour garantir que le gestionnaire de files d'attente conserve ses messages persistants et sa configuration après un redémarrage. Il s'agit du comportement par défaut. Etant donné qu'il existe différents fournisseurs de stockage, chacun prenant en charge des fonctionnalités différentes, cela signifie souvent que la personnalisation de la configuration est requise. L'exemple ci-dessous décrit les zones communes qui personnalisent la configuration de stockage IBM MQ dans l'API v1beta1 :

- **`spec.queueManager.availability`** contrôle le mode de disponibilité. Si vous utilisez `SingleInstance` ou `NativeHA`, vous n'avez besoin que de stockage `ReadWriteOnce` . Pour `multiInstance` , vous avez besoin d'une classe de stockage prenant en charge `ReadWriteMany` avec les caractéristiques de verrouillage de fichier correctes. 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 la section «Planification de la haute disponibilité pour IBM MQ dans des conteneurs», à la page 19.
- **`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:
         class: ibmc-file-gold-gid
      persistedData:
        enabled: true
        class: ibmc-file-gold-gid
      recoveryLogs:
        enabled: true
        class: ibmc-file-gold-gid
  securityContext:
    supplementalGroups: [65534] # Change to 99 for clusters with RHEL7 or earlier worker nodes
```

Pour plus d'informations sur les remarques relatives au stockage pour les gestionnaires de files d'attente Native HA, voir «Native HA», à la page 21.

Remarque : Vous pouvez également configurer des groupes supplémentaires avec des gestionnaires de files d'attente à instance unique.

Capacité de stockage

Lorsque vous utilisez le IBM MQ Operator , vous devez essayer de vous assurer que vous demandez des volumes suffisamment grands pour vos besoins permanents. Toutefois, si vous devez augmenter la capacité de stockage d'un ou de plusieurs volumes, ces volumes peuvent être étendus si votre

classe de stockage prend en charge l'extension de volume. Les volumes peuvent être étendus à l'aide d'une procédure en ligne ou hors ligne. Une procédure hors ligne requiert le redémarrage des pods `QueueManager` , contrairement à une procédure en ligne. Pour déterminer si votre classe de stockage prend en charge l'extension de volume et la procédure à suivre pour l'extension de volume, reportez-vous à la documentation de votre fournisseur de stockage. Vous devez prendre en compte ces informations lors de la sélection d'une classe de stockage. Pour plus d'informations sur l'extension de volume, voir «Extension des volumes persistants», à la page 104.

Chiffrement

IBM MQ ne chiffre pas activement les données au repos. Par conséquent, vous devez utiliser un stockage chiffré passivement, ou IBM MQ Advanced Message Security, ou les deux, pour chiffrer vos messages. Sous IBM Cloud , le stockage par blocs et le stockage de fichiers sont disponibles avec le chiffrement passif au repos.

Planification de la haute disponibilité pour IBM MQ dans des conteneurs

Il existe trois options pour la haute disponibilité avec IBM MQ Operator: **Gestionnaire de files d'attente Native HA** (qui possède une réplique active et deux répliques de secours), **Gestionnaire de files d'attente multi-instance** (qui est une paire active-de secours, utilisant un système de fichiers partagé en réseau) ou **Gestionnaire de files d'attente Single resilient** (qui offre une approche simple pour la haute disponibilité utilisant le stockage en réseau). Les deux derniers s'appuient sur le système de fichiers pour assurer la disponibilité des données récupérables, contrairement à Native HA. Par conséquent, lorsque vous n'utilisez pas Native HA, la disponibilité du système de fichiers est essentielle à la disponibilité du gestionnaire de files d'attente. Si la récupération des données est importante, le système de fichiers doit assurer la redondance via la réplication.

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 Native HA

Les gestionnaires de files d'attente Native HA impliquent un **actif** et deux pods **réplique** Kubernetes , qui s'exécutent dans le cadre d'un Kubernetes StatefulSet avec exactement trois répliques chacune avec leur propre ensemble de volumes persistants Kubernetes . Les exigences IBM MQ pour les systèmes de fichiers partagés s'appliquent également lors de l'utilisation d'un gestionnaire de files d'attente Native HA (à l'exception du verrouillage basé sur 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 *xfs* ou *ext4*. Les temps de reprise d'un gestionnaire de files d'attente Native HA sont contrôlés par les facteurs suivants :

1. Le temps nécessaire aux répliques d'instances pour détecter que l'instance active a échoué. Ce facteur peut être configuré.
2. Le temps nécessaire à la sonde de vigilance du pod Kubernetes pour détecter si le conteneur prêt a changé et rediriger le trafic réseau. Ce facteur peut être configuré.
3. Le temps nécessaire aux clients IBM MQ pour se reconnecter.

Pour plus d'informations, voir «Native HA», à la page 21.

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 prêt a changé et rediriger le trafic réseau. Ce facteur peut être configuré.
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 *xfs* 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. Ce facteur peut être configuré.
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. Ce facteur peut être configuré.
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. Lorsque vous utilisez une charge de travail avec état telle que IBM MQ avec un objet Kubernetes StatefulSet, si un noeud maître Kubernetes 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 du 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.

Remarque : La modification des détails de l' StatefulSet d'un gestionnaire de files d'attente IBM MQ , y compris le nombre de répliques, n'est pas prise en charge lorsque le gestionnaire de files d'attente est créé via IBM MQ Operator.

Concepts associés

Configurations à haute disponibilité

Tâches associées

«Configuration de la haute disponibilité pour les gestionnaires de files d'attente à l'aide de IBM MQ Operator», à la page 78

CP4I MQ Adv. Native HA

Native HA est une solution de haute disponibilité native (intégrée) pour IBM MQ qui peut être utilisée avec le stockage par blocs sur cloud.

Une configuration Native HA fournit un gestionnaire de files d'attente hautement disponible dans lequel les données MQ récupérables (par exemple, les messages) sont répliquées sur plusieurs ensembles de stockage, ce qui empêche toute perte de données en cas d'incidents de stockage. Le gestionnaire de files d'attente est constitué de plusieurs instances actives, l'une étant l'instance principale et les autres étant prêtes à prendre rapidement le relais en cas d'échec, afin de maximiser l'accès au gestionnaire de files d'attente et à ses messages.

Une configuration Native HA configuration est constituée de trois pods Kubernetes, chacun contenant une instance du gestionnaire de files d'attente. Une instance correspond au gestionnaire de files d'attente actif, qui traite les messages et écrit dans son journal de reprise. A chaque écriture dans le journal de reprise, le gestionnaire de files d'attente actif envoie les données aux deux autres instances, appelées répliques. Chaque réplique écrit dans son propre journal de reprise, reconnaît les données, puis met à jour ses propres données de file d'attente à partir du journal de reprise répliqué. Si le pod qui exécute le gestionnaire de files d'attente actif échoue, l'une des répliques d'instance du gestionnaire de files d'attente devient actif et dispose des données à jour qu'il peut utiliser.

Le type de journal est appelé "journal répliqué". Un journal répliqué est essentiellement un journal linéaire, avec une gestion automatique des journaux et des images de support automatiques activées. Voir Types de journalisation. Vous utilisez les mêmes techniques de gestion du journal répliqué que celles utilisées pour la gestion d'un journal linéaire.

Un service Kubernetes est utilisé pour acheminer les connexions client TCP/IP à l'instance active en cours, identifiée comme étant le seul pod prêt pour le trafic réseau. Cela se produit sans qu'il soit nécessaire que l'application client ait conscience des différentes instances.

Trois pods sont utilisés pour réduire considérablement la possibilité d'une situation de split-brain. Dans un système haute disponibilité à deux pods, split-brain peut se produire lorsque la connectivité entre les deux pods est rompue. En l'absence de connectivité, les deux pods peuvent exécuter le gestionnaire de files d'attente en même temps et accumuler des données différentes. Lorsque la connexion est restaurée, il existe alors deux versions différentes des données (un "split-brain') et une intervention manuelle est requise pour déterminer les données à conserver et celles à supprimer.

Native HA utilise un système à trois pod avec quorum pour éviter une situation de split-brain. Les pods qui peuvent communiquer avec au moins l'un des autres pods constituent le quorum. Un gestionnaire de files d'attente ne peut devenir l'instance active que sur un pod qui dispose du quorum. Le gestionnaire de files d'attente ne pouvant pas devenir actif sur un pod non connecté à au moins un autre pod, il ne peut donc jamais exister deux instances actives en même temps :

- En cas de défaillance d'un seul pod, le gestionnaire de files d'attente sur l'un des deux autres pods peut prendre le relais. En cas de défaillance de deux pods, le gestionnaire de files d'attente ne peut pas devenir l'instance active sur le pod restant car ce pod ne dispose pas du quorum (le pod restant ne peut pas savoir si les deux autres pods ont échoué ou s'ils sont toujours actifs et qu'il a perdu la connectivité).
- Si un seul pod perd la connectivité, le gestionnaire de files d'attente ne peut pas devenir actif sur ce pod car le pod ne dispose pas du quorum. Le gestionnaire de files d'attente sur l'un des deux pods restants, qui dispose du quorum, peut prendre le relais. Si tous les pods perdent la connectivité, le gestionnaire de files d'attente ne peut pas devenir actif sur l'un des pods car aucun ne dispose du quorum.

Si un pod actif échoue, puis est restauré, il peut rejoindre le groupe avec le rôle de réplique.

Pour des raisons de performances et de fiabilité, il est recommandé d'utiliser le stockage persistant RWO (ReadWriteOnce) avec une configuration Native HA. Les volumes RWO de n'importe quel fournisseur de stockage sont pris en charge s'ils remplissent les conditions suivantes:

- Obtenu auprès d'un fournisseur de stockage par blocs.
- Formaté en tant que ext4 ou XFS (ce qui garantit la conformité POSIX).
- Prend en charge la mise à disposition de volume dynamique et le mode "volumeBinding: WaitForFirstConsumer".

Les fournisseurs suivants sont explicitement interdits:

- NFS
- GlusterFS
- Autres fournisseurs non bloqués.

La figure ci-après illustre un déploiement type avec trois instances d'un gestionnaire de files d'attente déployé dans trois conteneurs.

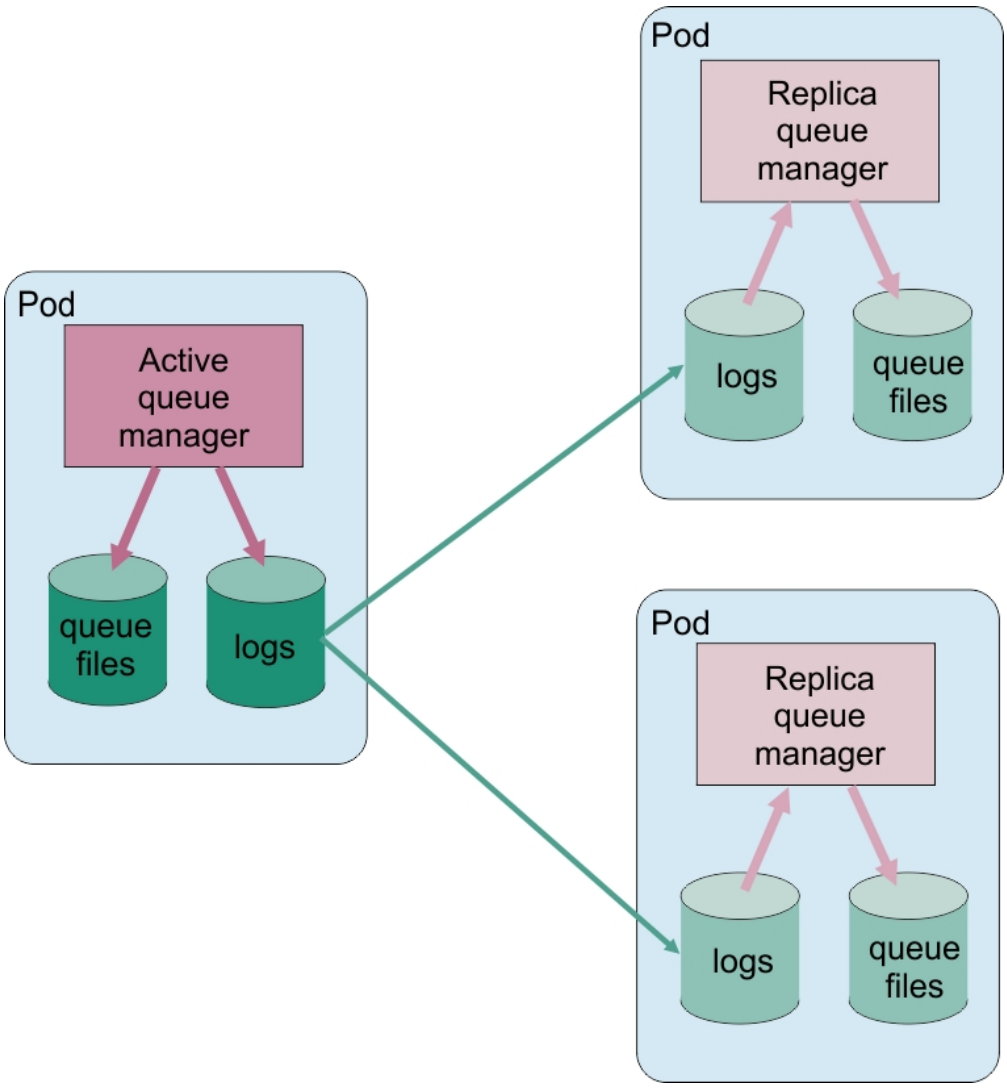


Figure 1. Exemple de configuration Native HA

MQ Adv. Remarques sur l'exécution de votre propre mise à jour en continu d'un gestionnaire de files d'attente natif de haute disponibilité

Toute mise à jour de la version IBM MQ ou de la spécification Pod pour un gestionnaire de files d'attente natif de haute disponibilité, vous demandera d'effectuer une mise à jour en continu des instances du gestionnaire de files d'attente. IBM MQ Operator gère cela automatiquement, mais si vous construisez votre propre code de déploiement, il y a des considérations importantes à prendre en compte.

Remarque : Le fichier Exemple de graphique Helm inclut un script de shell pour effectuer une mise à jour en continu, mais le script n'est **pas** adapté à l'utilisation de la production, car il n'aborde pas les considérations de cette rubrique.

Kubernetes Dans Kubernetes, les ressources `StatefulSet` sont utilisées pour gérer les mises à jour de démarrage et en continu commandées. Une partie de la procédure de démarrage consiste à démarrer chaque Pod individuellement, à attendre qu'il devienne prêt, puis à passer à la prochaine Pod. Cela ne fonctionnera pas pour Native HA, car tous les pods doivent être démarrés pour qu'ils puissent effectuer une élection de leader. Par conséquent, la zone `.spec.podManagementPolicy` sur le `StatefulSet` doit être définie sur `Parallel`. Cela signifie également que tous les Pods seront également mis à jour en parallèle, ce qui est particulièrement indésirable. Pour cette raison, le `StatefulSet` doit également utiliser la stratégie de mise à jour `OnDelete`.

L'inaptitude à utiliser le code de mise à jour en continu `StatefulSet` entraîne un besoin de code de mise à jour en continu personnalisé, qui doit prendre en compte les éléments suivants :

- Procédure générale de mise à jour en continu
- Réduire le temps d'indisponibilité en mettant à jour les Pods dans le meilleur ordre
- Traitement des modifications dans l'état du cluster
- Traitement des erreurs
- Traitement des problèmes de temps

Procédure générale de mise à jour en continu

Le code de mise à jour en continu doit attendre que chaque instance affiche un statut de `REPLICA` à partir de `dspmq`. Cela signifie que l'instance a exécuté un certain niveau de démarrage (par exemple, le conteneur est démarré et les processus MQ sont en cours d'exécution), mais qu'elle n'a pas encore réussi à parler aux autres instances. Par exemple, le pod A est redémarré et dès qu'il est à l'état `REPLICA` , le pod B est redémarré. Une fois que Pod B commence par la nouvelle configuration, il devrait être capable de parler à Pod A, et peut former le quorum, et soit A ou B deviendra la nouvelle instance active.

Dans ce cas, il est utile d'avoir un délai après que chaque Pod a atteint l'état `REPLICA`, afin de lui permettre de se connecter à ses homologues et d'établir le quorum.

Réduire le temps d'indisponibilité en mettant à jour les Pods dans le meilleur ordre

Le code de mise à jour en continu doit supprimer les Pods un à la fois, en commençant par les Pods qui se trouvent dans un état d'erreur connu, suivis des Pods qui n'ont pas démarré avec succès. Le gestionnaire Pod de files d'attente actif doit généralement être mis à jour en dernier.

Il est également important de mettre en pause la suppression des Pods si la dernière mise à jour a donné lieu à un Pod dans un état d'erreur connu. Cela empêche le déploiement d'une mise à jour interrompue sur tous les Pods. Par exemple, cela peut se produire si le Pod est mis à jour pour utiliser une nouvelle image de conteneur qui n'est pas accessible (ou contient une typo).

Traitement des modifications dans l'état du cluster

Le code de mise à jour en continu doit réagir de manière appropriée aux changements en temps réel dans l'état du cluster. Par exemple, l'un des Pods du gestionnaire de files d'attente peut être expulsé en raison d'un réamorçage du noeud ou de la pression du noeud. Il est possible qu'un Pod expulsé ne soit pas immédiatement reprogrammée si le cluster est occupé. Dans ce cas, le code de mise à jour en continu doit attendre correctement avant de redémarrer les autres Pods.

Traitement des erreurs

Le code de mise à jour en continu doit être robuste pour les échecs lors de l'appel de l'API de Kubernetes et d'autres comportements de cluster inattendus.

En outre, le code de mise à jour en continu lui-même doit être tolérant pour être redémarré. Une mise à jour en continu peut être longue et le code doit être redémarré.

Traitement des problèmes de temps

Le code de mise à jour en continu doit vérifier les révisions de mise à jour du Pod, de sorte qu'il puisse s'assurer que le Pod ait redémarré. Cela permet d'éviter les problèmes de temps où un Pod peut indiquer qu'il est « Démarré », mais n'est pas encore terminé en fait.

Concepts associés

«Comment utiliser IBM MQ dans des conteneurs», à la page 8
Il existe plusieurs options d'utilisation de IBM MQ dans des conteneurs : vous pouvez choisir d'utiliser le IBM MQ Operator, qui utilise des images de conteneur pré-conditionnées, ou créer vos propres images et code de déploiement.

OpenShift CP4I Kubernetes Reprise après incident d'IBM MQ dans des conteneurs

Vous devez prendre en compte le type de sinistre pour lequel vous vous préparez. Dans les environnements cloud, l'utilisation de zones de disponibilité offre un certain niveau de tolérance aux sinistres et est plus conviviale. Si vous disposez d'un nombre impair de centres de données (pour le quorum) et d'une liaison réseau à faible latence, vous pouvez éventuellement exécuter un cluster Red Hat OpenShift Container Platform ou Kubernetes unique avec plusieurs zones de disponibilité, chacune dans un emplacement physique distinct. Cette rubrique aborde les points à prendre en compte pour la reprise après incident lorsque ces critères ne peuvent pas être respectés, à savoir, si le nombre de centres de données est pair ou que la liaison réseau est à latence élevée.

Pour la reprise après incident, vous devez prendre en compte les points suivants :

- Réplication de données IBM MQ (conservées dans une ou plusieurs ressources `PersistentVolume`) dans l'emplacement de reprise après incident
- Nouvelle création du gestionnaire de files d'attente à l'aide des données répliquées
- ID réseau du gestionnaire de files d'attente visible par les applications clientes IBM MQ et les autres gestionnaires de files d'attente. Cet ID peut être une entrée du serveur de noms de domaine, par exemple.

Les données persistantes doivent être répliquées, de manière synchrone ou asynchrone, sur le site de reprise après incident. Ceci est généralement spécifique au fournisseur de stockage, mais peut également être effectué à l'aide d'un `VolumeSnapshot`. Pour plus d'informations sur les instantanés de volume, reportez-vous à la rubrique CSI volume snapshots.

Lors d'une reprise après incident, vous devez recréer l'instance de gestionnaire de files d'attente sur le nouveau cluster Kubernetes, à l'aide des données répliquées. Si vous utilisez IBM MQ Operator, vous aurez besoin de `QueueManager` YAML, ainsi que de YAML pour d'autres ressources de prise en charge telles que `ConfigMap` ou `Secret`.

Information associée

ha_for_ctr.dita

OpenShift CP4I Planification de la sécurité pour IBM MQ dans des conteneurs

Considérations de sécurité lors de la planification de votre IBM MQ dans la configuration des conteneurs.

Procédure

- «Authentification et autorisation des utilisateurs pour IBM MQ dans les conteneurs», à la page 25
  - «Contraintes de sécurité liées à l'utilisation des utilisateurs du système d'exploitation dans les conteneurs», à la page 26
- «Remarques relatives à la restriction du trafic réseau à IBM MQ dans les conteneurs», à la page 27

Authentification et autorisation des utilisateurs pour IBM MQ dans les conteneurs

IBM MQ dans des conteneurs peut être configuré pour authentifier les utilisateurs via LDAP, TLS mutuel ou un plug-in MQ personnalisé.

Notez que l'opérateur IBM MQ n'autorise pas l'utilisation des utilisateurs et des groupes du système d'exploitation dans l'image de conteneur. Pour plus d'informations, voir «Contraintes de sécurité liées à l'utilisation des utilisateurs du système d'exploitation dans les conteneurs», à la page 26.

LDAP

Pour plus d'informations sur la configuration d' IBM MQ pour utiliser un référentiel d'utilisateurs LDAP, voir Connection authentication: User repositories et LDAP authorization.

TLS mutuel

Si vous configurez des connexions entrantes vers un gestionnaire de files d'attente pour exiger un certificat TLS (TLS mutuel), vous pouvez mapper le nom distinctif du certificat à un nom d'utilisateur. Vous devez faire deux choses:

- Configurez un enregistrement d'authentification de canal pour créer le mappage à un nom d'utilisateur à l'aide de SSLPEER. Pour plus d'informations, voir Mappage d'un nom distinctif SSL ou TLS à un ID utilisateur MCAUSER.
- Configurez le gestionnaire de files d'attente pour vous permettre de définir des enregistrements de droits d'accès pour un nom d'utilisateur inconnu du système. Pour plus d'informations, voir Service stanza of qm.ini file.

Jetons Web JSON

Pour plus d'informations sur la configuration d' IBM MQ pour l'utilisation de jetons Web JSON (JWT), voir Utilisation de jetons d'authentification.

Plug-in MQ personnalisé

Il s'agit d'une technique avancée qui nécessite beaucoup plus de travail. Pour plus d'informations, voir Utilisation d'un service d'autorisation personnalisé.

Tâches associées

«Exemple: configuration d'un gestionnaire de files d'attente avec l'authentification TLS mutuelle», à la page 73

Cet exemple déploie un gestionnaire de files d'attente dans OpenShift Container Platform à l'aide de IBM MQ Operator. Le protocole TLS mutuel est utilisé pour l'authentification afin de mapper un certificat TLS à une identité dans le gestionnaire de files d'attente.

Contraintes de sécurité liées à l'utilisation des utilisateurs du système d'exploitation dans les conteneurs

L'utilisation d'utilisateurs du système d'exploitation dans des conteneurs n'est pas recommandée et est interdite avec l'opérateur IBM MQ .

Dans un environnement conteneurisé à service partagé, des contraintes de sécurité sont généralement mises en place pour éviter tout problème de sécurité. Par exemple :

- **Empêcher l'utilisation de l'utilisateur "root" dans un conteneur**
- **Forcer l'utilisation d'un ID utilisateur aléatoire**. Par exemple, dans Red Hat OpenShift Container Platform, l'objet `SecurityContextConstraints` par défaut (appelé `restricted`) utilise un ID utilisateur par conteneur.
- **Empêcher l'utilisation de l'escalade des privilèges**. IBM MQ on Linux utilise l'escalade des privilèges pour vérifier les mots de passe des utilisateurs-il utilise un programme "setuid" pour devenir l'utilisateur "root".

OpenShift CP4I Pour garantir la conformité avec ces mesures de sécurité, IBM MQ Operator n'autorise pas l'utilisation des ID définis sur les bibliothèques du système d'exploitation dans un conteneur. Aucun ID utilisateur ou groupe `mqm` n'est défini dans le conteneur.

Remarques relatives à la restriction du trafic réseau à IBM MQ dans les conteneurs

Vous pouvez définir des règles réseau pour limiter le trafic aux pods de votre cluster dans OpenShift Container Platform et Kubernetes. Cette rubrique décrit certaines considérations relatives à la manière dont les règles réseau peuvent s'appliquer à IBM MQ.

Pour l'entrée réseau vers un gestionnaire de files d'attente, il existe plusieurs ports à prendre en compte:

- Port 1414 pour le trafic du gestionnaire de files d'attente
- Port 9414 pour la haute disponibilité native
- Port 9157 pour les métriques
- Port 9443 pour la console Web et les API REST

La sortie réseau est plus complexe. Exemples de sorties réseau que vous souhaiterez peut-être prendre en compte:

- DNS-si vous avez des canaux ou d'autres configurations qui utilisent des noms DNS
- Autres gestionnaires de files d'attente
- Le protocole OCSP (Online Certificate Status Protocol) et les listes de révocation de certificat (CRL)-déterminées par votre fournisseur de certificat.
- Fournisseurs d'authentification:
  - LDAP
  - Ouvrez ID Connect ou un autre fournisseur de connexion configuré pour le serveur Web IBM MQ . Cela inclut IBM Cloud Pak Keycloak.
- Fournisseurs de traçage:
  - IBM Instana

  Remarque : Pour les versions IBM MQ antérieures, le tableau de bord des opérations IBM Cloud Pak for Integration était également disponible en tant que fournisseur de traçage. Toutefois, le tableau de bord des opérations a été supprimé dans IBM MQ 9.3.3 CD et IBM MQ 9.4.0 LTS.

Exemple d'entrée NetworkPolicy

Voici un exemple de règle réseau permettant de contrôler l'entrée pour un gestionnaire de files d'attente appelé "myqm", à utiliser sur Red Hat OpenShift Container Platform.

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

```
            network.openshift.io/policy-group: monitoring
    ports:
      - protocol: TCP
        port: 9157
  # Allow access to web server via Route
  - from:
      - namespaceSelector:
          matchLabels:
            network.openshift.io/policy-group: ingress
    ports:
      - protocol: TCP
        port: 9443
```

Conformité à la norme FIPS pour IBM MQ dans les conteneurs

Au démarrage, IBM MQ dans les conteneurs détecte si le système d'exploitation sur lequel le conteneur est démarré est conforme à la norme FIPS et (si tel est le cas) configure automatiquement la prise en charge de la norme FIPS. Les exigences et les limitations sont indiquées ici.

La norme FIPS (Federal Information Processing Standards)

Le gouvernement américain fournit des conseils techniques sur des systèmes informatiques et sur des systèmes de sécurité, y compris le chiffrement de données. Le National Institute for Standards and Technology (NIST) est un organisme gouvernemental chargé des systèmes informatiques et de la sécurité. Il émet des recommandations et établit des normes, notamment la norme FIPS (Federal Information Processing Standards).

Une norme FIPS importante est FIPS 140-2, qui nécessite l'utilisation d'algorithmes de cryptographie puissants. Elle spécifie également des exigences pour les algorithmes de hachage à utiliser afin de protéger les paquets contre toute modification en transit.

IBM MQ prend en charge FIPS 140-2 s'il a été configuré pour le faire.

Remarque : Sous AIX, Linux, and Windows, IBM MQ fournit la conformité à la norme FIPS 140-2 via le module cryptographique IBM Crypto for C (ICC) . Le certificat de ce module a été déplacé vers le statut Historique. Les clients doivent afficher le certificatIBM Crypto for C (ICC) et prendre connaissance des conseils fournis par le NIST. Un module FIPS 140-3 de remplacement est actuellement en cours et son statut peut être affiché en le recherchant dans la liste des modules NIST CMVP en cours de traitement.

IBM MQ Operator 3.2.0 et l'image de conteneur du gestionnaire de files d'attente à partir de la version 9.4.0.0 sont basés sur UBI 9. La conformité à la norme FIPS 140-3 est actuellement en attente et son statut peut être affiché en recherchant "Red Hat Enterprise Linux 9- OpenSSL FIPS Provider" dans la liste de processus des modules CMVP NIST.

Exigences

Pour les exigences liées à la configuration du cluster et d'autres considérations, voir FIPS Wall: Current IBM approche to FIPS compliance.

IBM MQ dans les conteneurs peut s'exécuter en mode de conformité FIPS 140-2. Lors du démarrage, IBM MQ dans les conteneurs détecte si le système d'exploitation hôte sur lequel le conteneur démarre est conforme à la norme FIPS. Si le système d'exploitation de l'hôte est conforme à la norme FIPS et que des clés privées et des certificats ont été fournis, le conteneur IBM MQ configure le gestionnaire de files d'attente, le serveur Web IBM MQ et le transfert de données entre les noeuds dans un déploiement Native High Availability, pour une exécution en mode de conformité FIPS.

Lors de l'utilisation de IBM MQ Operator pour déployer des gestionnaires de files d'attente, l'opérateur crée une route avec le type d'arrêt **Passthrough**. Cela signifie que le trafic est envoyé directement à la destination sans que le routeur ne fournisse de terminaison TLS. Le gestionnaire de files d'attente IBM MQ et le serveur Web IBM MQ sont les destinations dans ce cas, et ils fournissent déjà une communication sécurisée conforme à la norme FIPS.

Exigences clés:

1. Clé privée et certificats, fournis dans un secret au gestionnaire de files d'attente et au serveur Web, qui permettent à des clients externes de se connecter de manière sécurisée au gestionnaire de files d'attente et au serveur Web.
2. Clé privée et certificats pour le transfert de données entre différents noeuds dans une configuration à haute disponibilité native.

Limitations

Pour un déploiement conforme à la norme FIPS de IBM MQ dans des conteneurs, tenez compte des éléments suivants:

- IBM MQ dans des conteneurs fournit un noeud final pour la collecte de métriques. Actuellement, ce noeud final est uniquement HTTP. Vous pouvez désactiver le noeud final metrics pour rendre le reste d' IBM MQ conforme à la norme FIPS.
- IBM MQ dans les conteneurs autorise les substitutions d'image personnalisées. Autrement dit, vous pouvez générer des images personnalisées en utilisant l'image de conteneur IBM MQ comme image de base. La conformité à la norme FIPS peut ne pas s'appliquer à ces images personnalisées.
- Pour le suivi des messages à l'aide de IBM Instana, la communication entre IBM MQ et IBM Instana est HTTP ou HTTPS, sans conformité avec la norme FIPS.
- L'accès IBM MQ Operator aux services IBM IAM (Identity and Access Management) /Zen n'est pas conforme à la norme FIPS.

Détection de la conformité à la norme FIPS et configuration automatique de la prise en charge de la norme FIPS

Si le système d'exploitation sur lequel le conteneur démarre est conforme à la norme FIPS, la prise en charge de la norme FIPS est configurée automatiquement.

Remarque : Sous AIX, Linux, and Windows, IBM MQ fournit la conformité à la norme FIPS 140-2 via le module cryptographique IBM Crypto for C (ICC) . Le certificat de ce module a été déplacé vers le statut Historique. Les clients doivent afficher le certificatIBM Crypto for C (ICC) et prendre connaissance des conseils fournis par le NIST. Un module FIPS 140-3 de remplacement est actuellement en cours et son statut peut être affiché en le recherchant dans la liste des modules NIST CMVP en cours de traitement.

IBM MQ Operator 3.2.0 et l'image de conteneur du gestionnaire de files d'attente à partir de la version 9.4.0.0 sont basés sur UBI 9. La conformité à la norme FIPS 140-3 est actuellement en attente et son statut peut être affiché en recherchant "Red Hat Enterprise Linux 9- OpenSSL FIPS Provider" dans la liste de processus des modules CMVP NIST.

Lors du démarrage, IBM MQ dans les conteneurs détecte si le système d'exploitation sur lequel le conteneur est démarré est conforme à la norme FIPS. Si tel est le cas, les actions suivantes sont effectuées automatiquement:

Gestionnaire de files d'attente
Si le système d'exploitation hôte est conforme à la norme FIPS et que la clé privée et les certificats sont fournis, l'attribut de gestionnaire de files d'attente **SSLFIPS** est défini sur YES. Sinon, l'attribut **SSLFIPS** est défini sur NO.

IBM MQ serveur Web
Le serveur Web IBM MQ fournit une interface HTTP/HTTPS pour l'administration de IBM MQ. Si le système d'exploitation hôte est conforme à la norme FIPS, les options JVM sont mises à jour pour que le serveur Web utilise la cryptographie conforme à la norme FIPS. Pour pouvoir utiliser FIPS, la clé privée et les certificats doivent être fournis lors du démarrage du conteneur.

Native HA
La sécurité des données répliquées entre les noeuds est contrôlée par la section **NativeHALocalInstance** du fichier `qm.ini` . Exemple :

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

Si FIPS est activé, l'attribut **SSLFipsRequired** est ajouté à la strophe, avec la valeur définie sur Yes:

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

Si le conteneur s'exécute dans un cluster OpenShift sans prise en charge FIPS, le gestionnaire de files d'attente, le serveur Web IBM MQ et les composants Native HA n'ont pas leur prise en charge FIPS activée automatiquement. Seule l'architecture x86-64 est actuellement prise en charge par la plateforme OpenShift pour FIPS. Pour les architectures Power et Linux for IBM Z , OpenShift n'offre pas la prise en charge FIPS. Pour activer explicitement la prise en charge de FIPS dans les composants IBM MQ pour ces architectures, définissez la variable d'environnement *MQ_ENABLE_FIPS* sur `true` dans le fichier YAML du gestionnaire de files d'attente. Le fragment YAML suivant décrit l'utilisation de la variable d'environnement *MQ_ENABLE_FIPS* :

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

Remplacement du mode FIPS automatique pour IBM MQ dans les conteneurs

Utilisez la variable d'environnement *MQ_ENABLE_FIPS* pour activer ou désactiver explicitement le mode FIPS pour les composants IBM MQ dans le conteneur.

Avant de commencer

Remarque : Sous AIX, Linux, and Windows, IBM MQ fournit la conformité à la norme FIPS 140-2 via le module cryptographique IBM Crypto for C (ICC) . Le certificat de ce module a été déplacé vers le statut Historique. Les clients doivent afficher le certificatIBM Crypto for C (ICC) et prendre connaissance des conseils fournis par le NIST. Un module FIPS 140-3 de remplacement est actuellement en cours et son statut peut être affiché en le recherchant dans la liste des modules NIST CMVP en cours de traitement.

IBM MQ Operator 3.2.0 et l'image de conteneur du gestionnaire de files d'attente à partir de la version 9.4.0.0 sont basés sur UBI 9. La conformité à la norme FIPS 140-3 est actuellement en attente et son statut peut être affiché en recherchant "Red Hat Enterprise Linux 9- OpenSSL FIPS Provider" dans la liste de processus des modules CMVP NIST.

Pourquoi et quand exécuter cette tâche

MQ_ENABLE_FIPS prend en charge trois valeurs:

Automatique
Il s'agit de la valeur par défaut.

Si la norme FIPS est activée sur le système d'exploitation hôte, tous les composants (gestionnaire de files d'attente, serveur Web IBM MQ et Native HA) s'exécutent en mode FIPS.

Si le système d'exploitation hôte n'est pas activé pour FIPS, tous les composants ne s'exécutent pas en mode FIPS.

Oui
Cette valeur active la norme FIPS pour les composants sélectionnés dans le conteneur.

L'attribut de gestionnaire de files d'attente **SSLFIPS** est défini sur YES même si IBM MQ dans les conteneurs s'exécute sur un système d'exploitation hôte qui n'est pas conforme à la norme FIPS. Autrement dit, si le gestionnaire de files d'attente IBM MQ , le serveur Web et Native HA sont conformes à la norme FIPS, mais que le système d'exploitation du conteneur ne l'est pas.

false
Cette valeur désactive la conformité à la norme FIPS.

L'attribut de gestionnaire de files d'attente **SSLFIPS** est défini sur NO, même si IBM MQ dans des conteneurs s'exécute sur une machine hôte compatible FIPS. Toutefois, IBM MQ sécurise toujours les connexions si la clé privée et les certificats sont fournis.

Les options JVM ne sont pas mises à jour pour le serveur Web IBM MQ . Toutefois, le serveur Web IBM MQ exécute toujours un noeud final HTTPS si la clé privée et les certificats sont fournis.

La réplication de données dans Native HA n'utilise pas la cryptographie FIPS.

Exemple

Voici un exemple de fichier YAML de gestionnaire de files d'attente qui décrit l'activation de TLS et de FIPS pour le composant de gestionnaire de files d'attente:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  namespace: ibm-mq-fips
  name: ibm-mq-qm-ppcle
spec:
  license:
    accept: true
    license: L-EHXT-MQCRN9
    use: Production
  queueManager:
    name: PPCLEQM
    storage:
      queueManager:
        type: ephemeral
  template:
    pod:
      containers:
        - env:
            - name: MQ_ENABLE_FIPS
              value: "true"
          name: qmgr
  version: 9.4.0.0-r1
  web:
    enabled: false
  pki:
    keys:
      - name: ibm-mq-tls-certs
        secret:
          secretName: ibm-mq-tls-secret
          items:
          - tls.key
          - tls.crt
```

Planification de l'évolutivité et des performances pour IBM MQ dans les conteneurs

Dans la plupart des cas, la mise à l'échelle et les performances d' IBM MQ dans les conteneurs sont identiques à celles d' IBM MQ for Multiplatforms. Toutefois, quelques limites supplémentaires peuvent être imposées par la plateforme de conteneurs.

Pourquoi et quand exécuter cette tâche

Lorsque vous planifiez l'évolutivité et les performances d' IBM MQ dans des conteneurs, tenez compte des options suivantes:

Procédure

- **Limiter le nombre d'unités d'exécution et de processus.**

  IBM MQ utilise des unités d'exécution pour gérer les accès concurrents. Dans Linux, les unités d'exécution sont implémentées en tant que processus, de sorte que vous pouvez rencontrer des limites imposées par la plateforme de conteneur ou le système d'exploitation, sur le nombre maximal de processus. A partir de Red Hat OpenShift Container Platform 4.11, il existe une limite par défaut

de 4096 processus par conteneur. Bien que cela soit approprié pour la grande majorité des scénarios, cela peut avoir un impact sur le nombre de connexions client pour un gestionnaire de files d'attente.

La limite de processus dans Kubernetes peut être configurée par un administrateur de cluster à l'aide du paramètre de configuration kubelet **podPidsLimit**. Voir Process ID limits and reservation dans la documentation Kubernetes . Dans Red Hat OpenShift Container Platform, vous pouvez également créer une ressource personnalisée **ContainerRuntimeConfig** pour éditer les paramètres CRI-O.

Dans votre configuration IBM MQ , vous pouvez également définir le nombre maximal de connexions client pour un gestionnaire de files d'attente. Voir Limites de canal de connexion serveur pour l'application de limites à un canal de connexion serveur individuel et l' attribut MAXCHANNELS INI pour l'application de limites à l'ensemble du gestionnaire de files d'attente.

- **Limiter le nombre de volumes.**

  Dans les systèmes de cloud et de conteneur, les volumes de stockage connectés au réseau sont couramment utilisés. Le nombre de volumes pouvant être connectés à des noeuds Linux est limité. Par exemple, AWS EC2 ne limite pas plus de 30 volumes par machine virtuelle. Red Hat OpenShift Container Platform a une limite similaire, tout comme Microsoft Azure et Google Cloud Platform.

  Un gestionnaire de files d'attente Native HA requiert un volume pour chacune des trois instances et applique les instances à répartir entre les noeuds. Toutefois, vous pouvez configurer le gestionnaire de files d'attente pour qu'il utilise trois volumes par instance (données du gestionnaire de files d'attente, journaux de reprise et données persistantes).

- **Utiliser les techniques de mise à l'échelle IBM MQ .**

  Au lieu d'utiliser un petit nombre de gestionnaires de files d'attente volumineux, il peut être utile d'utiliser des techniques de mise à l'échelle IBM MQ telles que des clusters uniformes IBM MQ pour exécuter plusieurs gestionnaires de files d'attente avec la même configuration. Ceci a pour avantage supplémentaire de réduire l'impact d'un redémarrage d'un conteneur unique (par exemple, dans le cadre de la maintenance de la plateforme de conteneur).

Préparation, installation et mise à niveau de votre environnement pour IBM MQ dans des conteneurs

Vous effectuez une série de tâches pour préparer votre environnement pour IBM MQ

Pourquoi et quand exécuter cette tâche

Si vous utilisez IBM MQ Operator, vous préparez votre cluster Red Hat OpenShift Container Platformen installant l'opérateur. Voir «Installation et mise à niveau du IBM MQ Operator», à la page 32

Sinon, vous préparez votre environnement de conteneur en créant vos propres images de conteneur. Voir «Préparation pour IBM MQ en créant votre propre image de conteneur», à la page 57

Installation et mise à niveau du IBM MQ Operator

Vous effectuez une série de tâches pour installer, désinstaller et mettre à niveau IBM MQ Operator.

Pourquoi et quand exécuter cette tâche

Pour vous initier à l'installation et à la mise à niveau de IBM MQ Operator sous Red Hat OpenShift Container Platform, voir les rubriques suivantes.

Procédure

- «Dépendances pour IBM MQ Operator», à la page 33
- «Droits d'accès au cluster requis par IBM MQ Operator», à la page 33
- «Vérification des signatures d'image», à la page 33

- «Installation d'IBM MQ Operator», à la page 34
- «Mise à niveau de IBM MQ Operator et des gestionnaires de files d'attente», à la page 45
- «Désinstallation de IBM MQ Operator», à la page 55

OpenShift CP4I Dépendances pour IBM MQ Operator

Aucun autre opérateur n'est installé automatiquement lorsque vous installez IBM MQ Operator.

L'opérateur de licence IBM doit être installé séparément pour suivre l'utilisation des licences. Voir Déploiement de License Service dans la documentation IBM Cloud Pak for Integration .

Lorsque vous créez un `QueueManager` à l'aide d'une licence IBM Cloud Pak for Integration , vous pouvez choisir d'utiliser ou non la connexion unique avec l'instance IBM Cloud Pak for Integration de Keycloak. L'utilisation de Keycloak est activée par défaut avec une licence IBM Cloud Pak for Integration , mais s'il n'est pas installé, `QueueManager` passe à l'état "Bloqué" jusqu'à ce que les dépendances correctes soient installées. Voir «Installation d'IBM MQ Operator», à la page 34 pour plus de détails sur les dépendances.

OpenShift CP4I Droits d'accès au cluster requis par IBM MQ Operator

IBM MQ Operator requiert des droits d'accès au cluster pour gérer les webhooks d'admission et les exemples, ainsi que pour lire les informations de la classe de stockage et de la version du cluster.

IBM MQ Operator requiert les droits d'accès au cluster suivants :

- Droit de gérer les webhooks d'admission. Permet de créer, d'extraire et de mettre à jour des crochets spécifiques utilisés dans le processus de création et de gestion des conteneurs fournis par l'opérateur.
  - Groupes d'API : **`admissionregistration.k8s.io`**
  - Ressources : **`validatingwebhookconfigurations`**
  - verbs: **`get, delete`**
- Permet de créer et de gérer des ressources utilisées dans la console Red Hat OpenShift pour fournir des exemples et des fragments lors de la création de ressources personnalisées.
  - Groupes d'API : **`console.openshift.io`**
  - Ressources : **`consoleyamlsamples`**
  - verbs: **`create, get, update, delete`**
- Droit de lecture de la version du cluster. Permet à l'opérateur de faire part de tout problème concernant l'environnement du cluster.
  - Groupes d'API : **`config.openshift.io`**
  - Ressources : **`clusterversions`**
  - verbs: **`get, list, watch`**
- Droit de lecture des classes de stockage sur le cluster. Permet à l'opérateur de faire part de tout problème concernant certaines classes de stockage dans les conteneurs.
  - Groupes d'API : **`storage.k8s.io`**
  - Ressources : **`storageclasses`**
  - verbs: **`get, list`**

Remarque : IBM MQ Operator requiert également des droits d'accès au niveau de l'espace de nom. Si IBM MQ Operator est installé au niveau d'un cluster, les droits d'accès au niveau de l'espace de nom sont présents dans tous les espaces de nom.

OpenShift CP4I Vérification des signatures d'image

Les images de conteneur de gestionnaire de files d'attente IBM MQ Operator et IBM MQ sont signées numériquement.

Pourquoi et quand exécuter cette tâche

Les signatures numériques permettent aux consommateurs de contenu de s'assurer que ce qu'ils téléchargent est à la fois authentique (il provient de la source attendue) et intègre (c'est ce que nous attendons).

Procédure

- Vérifiez les signatures des images de conteneur de gestionnaire de files d'attente IBM MQ Operator et IBM MQ :
  - Voir Vérification des signatures d'image dans la documentation IBM Cloud Pak for Integration (CP4I) 16.1.0 .

OpenShift CP4I Installation d'IBM MQ Operator

IBM MQ Operator peut être installé sur Red Hat OpenShift à l'aide de la console OpenShift ou de l'interface de ligne de commande (CLI).

Avant de commencer

Important :

- Cette rubrique concerne l'installation de IBM MQ Operator pour une utilisation autonome **uniquement**. Si vous prévoyez d'utiliser la connexion unique IBM Cloud Pak for Integrationou Keycloak pour un ou plusieurs gestionnaires de files d'attente, voir «Installation du IBM MQ Operator pour une utilisation avec CP4I», à la page 41.
- Consultez les conseils relatifs à la structuration de votre déploiement avant d'installer le IBM MQ Operator.

Pour vous assurer que votre installation se déroule de la manière la plus fluide possible, assurez-vous que vous comprenez tous les prérequis et exigences avant de commencer l'installation. Voir «Planification d'IBM MQ dans des conteneurs», à la page 7.

Pourquoi et quand exécuter cette tâche

Les étapes suivantes représentent le flux de tâches standard pour l'installation de votre IBM MQ Operator:

1. Installez Red Hat OpenShift Container Platform.
2. Configurer le stockage.
3. Images miroir (air-gap uniquement).
4. Ajoutez le catalogue IBM MQ Operator.
5. Installez IBM MQ Operator.
6. Créez le secret de clé d'autorisation (installations en ligne uniquement).
7. Déployez le License Service.
8. Déployez un gestionnaire de files d'attente.

Procédure

1. Installez Red Hat OpenShift Container Platform.

   Pour connaître les étapes détaillées d'installation de OpenShift, voir Installation du logiciel Red Hat 4.6 ou version ultérieure.

   Important : Veillez à installer une version prise en charge de OpenShift Container Platform. Par exemple, pour utiliser IBM MQ Operator 3.2 ou version ultérieure, vous devez installer OpenShift

Container Platform 4.12 ou version ultérieure. Pour plus d'informations, voir IBM Cloud Pak and Red Hat OpenShift Container Platform compatibility.

Pour toute étape utilisant l'interface de ligne de commande Red Hat OpenShift Container Platform , vous devez être connecté à votre cluster OpenShift avec `oc login`. Pour installer l'interface de ligne de commande, voir Initiation à l'interface de ligne de commande OpenShift.

Après avoir installé OpenShift, vous pouvez vérifier et accéder à votre logiciel de conteneur à l'aide de la clé d'autorisation IBM que vous créez dans Création du secret de clé d'autorisation.

2. Configurez le stockage.

   Vous devez définir des classes de stockage dans Red Hat OpenShift Container Platform et définir votre configuration de stockage pour répondre à vos exigences de dimensionnement.

   Important : Les gestionnaires de files d'attente IBM MQ mono-instance et Native HA peuvent utiliser le mode d'accès RWO, tandis que les gestionnaires de files d'attente multi-instance requièrent RWX comme décrit dans «Planification du stockage pour le IBM MQ Operator», à la page 17. Les gestionnaires de files d'attente multi-instance IBM MQ requièrent des caractéristiques de système de fichiers particulières, qui peuvent être vérifiées à l'aide des instructions de la rubrique Test d'un système de fichiers partagé pour IBM MQ.

   Vous trouverez la liste des systèmes de fichiers compatibles et non conformes connus, ainsi que des remarques sur les autres limites ou restrictions, dans l' instruction de test pour les systèmes de fichiers IBM MQ.

   Les fournisseurs de stockage recommandés sont disponibles sur la page CP4I Remarques sur le stockage .

3. Images miroir (air-gap uniquement).

   Si votre cluster se trouve dans un environnement réseau restreint (avec isolation physique), vous devez mettre en miroir les images IBM MQ à l'aide des valeurs suivantes:

   ```
   export OPERATOR_PACKAGE_NAME=ibm-mq
   export OPERATOR_VERSION=3.2.1
   ```

   Pour créer des images miroir, voir Mirroring images for an air-gap cluster.

4. Ajoutez la source de catalogue IBM MQ Operator .

   Ajoutez la source de catalogue qui met les opérateurs à la disposition de votre cluster. Voir «Ajout de la source de catalogue IBM MQ Operator», à la page 36.

5. Installez IBM MQ Operator.

   Choisissez l'une des deux options suivantes (utilisez la console ou l'interface de ligne de commande):

   - Option 1: Installez IBM MQ Operator à l'aide de la console OpenShift.
   - Option 2: Installez IBM MQ Operator à l'aide de l'interface de ligne de commande OpenShift.

6. Créez le secret de la clé d'autorisation (installations en ligne uniquement).

   IBM MQ Operator déploie des images de gestionnaire de files d'attente extraites d'un registre de conteneur qui effectue une vérification des autorisations de licence. Ce contrôle requiert une clé d'autorisation qui est stockée dans un secret d'extraction `docker-registry`. Si vous ne disposez pas encore d'une clé d'autorisation dans l'espace de nom dans lequel vous allez installer les gestionnaires de files d'attente, suivez ces instructions pour obtenir une clé d'autorisation et créer un secret d'extraction.

   Remarque : La clé d'autorisation n'est pas requise si seuls les gestionnaires de files d'attente IBM MQ Advanced for Developers (non garantis) vont être déployés.

   Vous pouvez créer le secret de la clé d'autorisation à l'aide de la console OpenShift ou de l'interface de ligne de commande. L'exemple suivant utilise l'interface de ligne de commande:

   a. Obtenez la clé d'autorisation affectée à votre ID IBM . 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.

c. A partir de l'interface de ligne de commande OpenShift , exécutez la commande suivante pour créer un secret d'extraction d'image appelé `ibm-entitlement-key`.

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

Où *entitlement_key* est la clé d'autorisation que vous avez copiée à l'étape b, *user_email* est l'ID IBM associé au logiciel autorisé et *namespace* est l'espace de nom dans lequel vous avez installé votre IBM MQ Operator .

7. Déployez le License Service.

   Cette opération est requise pour la surveillance de l'utilisation des licences des gestionnaires de files d'attente. Suivez les instructions de la rubrique Déploiement du service de licence License Service.

8. Déployez un gestionnaire de files d'attente.

   Pour des instructions sur le déploiement d'un exemple de gestionnaire de files d'attente de "démarrage rapide" , voir «Déploiement d'un gestionnaire de files d'attente simple à l'aide de IBM MQ Operator», à la page 66.

Tâches associées

«Désinstallation de IBM MQ Operator», à la page 55
Vous pouvez utiliser la console ou l'interface de ligne de commande Red Hat OpenShift pour désinstaller IBM MQ Operator de Red Hat OpenShift.

OpenShift *Ajout de la source de catalogue IBM MQ Operator*

Ajoutez la source de catalogue IBM MQ Operator à votre cluster OpenShift pour rendre le IBM MQ Operator disponible pour l'installation. Cette tâche est également requise si vous appliquez des groupes de correctifs de source de catalogue avant d'effectuer une mise à niveau.

Pourquoi et quand exécuter cette tâche

Un catalogue d'opérateurs est un index des opérateurs disponibles pour étendre l'API d'un cluster Red Hat OpenShift Container Platform afin d'activer les produits logiciels IBM .

Les sources de catalogue suivantes sont disponibles:

Option 1: source de catalogue spécifique pour IBM MQ Operator.

En utilisant une source de catalogue IBM MQ Operator spécifique, vous obtenez un contrôle total de la gestion des versions de logiciels sur un cluster et lorsque des mises à niveau sont effectuées. Une nouvelle version de IBM MQ Operator devient disponible dans un OpenShift cluster **uniquement** après la mise à jour de la source de catalogue. Ce processus vous permet de contrôler manuellement les mises à niveau. Vous n'avez donc pas besoin d'utiliser l'option `Manuel` pour le paramètre **`Update approval`** des opérateurs. L'option **Manuel** force toutes les mises à niveau possibles à être effectuées en même temps et peut bloquer les mises à niveau. Par conséquent, utilisez l'option **Automatique** uniquement. Pour plus d'informations, voir la section "Restriction des mises à jour automatiques avec une stratégie d'approbation" de la rubrique Installation des opérateurs à l'aide de la console Red Hat OpenShift.

Choisissez cette option si vous effectuez une mise à niveau et que vous devez ajouter la source de catalogue IBM MQ Operator d'une version plus récente.

Pour utiliser cette option, passez à l' Option 1: Ajouter des sources de catalogue spécifiques pour le IBM MQ Operator.

Option 2: IBM Operator Catalog.

Avec cette option, de nouvelles versions d'opérateur deviennent disponibles et sont appliquées **sans** intervention de votre part. Par conséquent, utilisez cette option **uniquement** pour les installations en ligne pour lesquelles vous souhaitez des mises à niveau **automatiques** du IBM MQ Operatoret pour lesquelles les installations déterministes ne sont pas nécessaires.

Remarque : Cette option peut être utile pour les environnements de démonstration de faisabilité, mais elle n'est **pas adaptée aux environnements de production**.

Pour utiliser cette option, passez à l' Option 2: Ajouter le IBM catalogue des opérateurs.

Procédure

- **Option 1: Ajout de sources de catalogue spécifiques pour le IBM MQ Operator.**

  Cette tâche suppose que vous avez effectué les trois premières étapes de «Installation d'IBM MQ Operator», à la page 34.

  Cette tâche doit être effectuée par un administrateur de cluster et doit être effectuée à l'aide de l'interface de ligne de commande.

  a) Mise à niveau uniquement: si vous appliquez des groupes de correctifs de source de catalogue avant une mise à niveau, procédez comme suit:

  – Vérifiez que vos opérateurs s'exécutent correctement.

  – Si des mises à jour IBM MQ Operator en attente nécessitent une approbation manuelle, validez ces mises à jour avant de commencer cette procédure. Pour plus d'informations, voir "Restriction des mises à jour automatiques avec une stratégie d'approbation" dans Installation des opérateurs à l'aide de la console Red Hat OpenShift.

  b) Si vous ne l'avez pas déjà installé ou s'il doit être mis à jour, téléchargez le plug-in IBM Catalog Management (version 1.6.0 ou ultérieure) à partir de GitHub.

  Ce plug-in vous permet d'exécuter des commandes **oc ibm-pak** sur le cluster.

  c) Connectez-vous à votre cluster à l'aide de la commande **oc login** et de vos données d'identification utilisateur:

```
oc login openshift_url -u username -p password -n namespace
```

  d) Exportez les variables d'environnement suivantes pour IBM MQ Operator:

```
export OPERATOR_PACKAGE_NAME=ibm-mq
export OPERATOR_VERSION=3.2.1
export ARCH=ARCHITECTURE
```

  où *ARCHITECTURE* correspond à l'architecture du système sur lequel vous déployez le IBM MQ Operatoret a la valeur `amd64`, `ppc64leou` `s390x`.

  Important : Si vous passez du catalogue de l'opérateur IBM à la source de catalogue spécifique pour le IBM MQ Operator, définissez `OPERATOR_VERSION` sur la version de votre déploiement du IBM MQ Operator.

  e) Téléchargez les fichiers de l'opérateur IBM MQ .

  Remarque : Si vous effectuez une installation **avec isolation physique** , vous devez déjà disposer des fichiers dont vous avez besoin après avoir effectué l'étape "Images miroir" de la rubrique "Installation du IBM MQ Operator", auquel cas vous pouvez passer à l'étape «8», à la page 38 "Application de la source de catalogue IBM MQ Operator au cluster".

```
oc ibm-pak get ${OPERATOR_PACKAGE_NAME} --version ${OPERATOR_VERSION}
```

  f) Générez la source de catalogue requise pour IBM MQ Operator:

```
oc ibm-pak generate mirror-manifests ${OPERATOR_PACKAGE_NAME} icr.io --version $
{OPERATOR_VERSION}
```

g) Facultatif : Générez les sources de catalogue et enregistrez-les dans un autre répertoire.

a. Obtenez la source du catalogue:

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

b. (Facultatif) Accédez au répertoire dans votre navigateur de fichiers pour copier ces artefacts dans des fichiers que vous pouvez conserver en vue de leur réutilisation ou pour les pipelines.

h) Appliquez la source de catalogue IBM MQ Operator au cluster.

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

i) Vérifiez que la source de catalogue IBM MQ Operator a été créée dans l'espace de nom `openshift-marketplace` :

```
oc get catalogsource -n openshift-marketplace
```

Exemple de sortie :

```
oc get catalogsource -n openshift-marketplace

NAME                           DISPLAY               TYPE   PUBLISHER   AGE
ibmmq-operator-catalogsource   ibm-mq-3.1.3          grpc   IBM         23h
```

Vous êtes maintenant prêt à effectuer l' étape 5 de l'installation du IBM MQ Operator.

- **Option 2: Ajoutez le catalogue opérateur IBM .**

  Important : Utilisez le IBM catalogue des opérateurs **uniquement** pour les installations en ligne où vous souhaitez des mises à niveau **automatiques** du IBM MQ Operatoret où les installations déterministes ne sont pas nécessaires. Cette option peut être utile pour les environnements de démonstration de faisabilité, mais elle n'est **pas adaptée aux environnements de production**.

  Le catalogue des opérateurs IBM est un index des opérateurs disponibles pour étendre l'API d'un cluster Red Hat OpenShift Container Platform afin d'activer les produits logiciels IBM . L'ajout des sources de catalogue à votre cluster OpenShift ajoute les opérateurs IBM à la liste des opérateurs que vous pouvez installer.

  Cette tâche suppose que vous avez effectué les trois premières étapes de «Installation d'IBM MQ Operator», à la page 34.

  Cette tâche peut être effectuée à l'aide de l'interface de ligne de commande ou de la console Web OpenShift .

  Utilisation de l'interface de ligne de commande

  1. Copiez la définition de ressource suivante pour les opérateurs IBM dans un fichier local de votre ordinateur:

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

  2. Exécutez la commande suivante. Remplacez *filename*.yaml par le nom du fichier que vous avez créé à l'étape précédente:

```
oc apply -f filename.yaml
```

Utilisation de la console Web OpenShift

1. Connectez-vous à la console Web OpenShift avec vos données d'identification d'administrateur de cluster OpenShift .
2. Dans la bannière, cliquez sur le signe plus ("+") pour ouvrir la boîte de dialogue **Importer un fichier YAML** .

   Remarque : Il n'est pas nécessaire de sélectionner une valeur pour **Projet**. Le code YAML de l'étape suivante inclut déjà la valeur correcte pour `metadata:namespace`, qui garantit que la source de catalogue est installée dans le projet (espace de nom) approprié.
3. Collez la définition de ressource 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
  image: 'icr.io/cpopen/ibm-operator-catalog:latest'
  publisher: IBM
  sourceType: grpc
  updateStrategy:
    registryPoll:
      interval: 45m
```

4. Cliquez sur **Créer**.

Vous êtes maintenant prêt à effectuer l' étape 5 de l'installation du IBM MQ Operator.

OpenShift CP4I *Installation du IBM MQ Operator à l'aide de la console OpenShift*

IBM MQ Operator peut être installé sur Red Hat OpenShift à l'aide de OperatorHub.

Avant de commencer

Cette tâche suppose que vous avez effectué les étapes 1 à 4 de «Installation d'IBM MQ Operator», à la page 34.

Procédure

1. Connectez-vous à votre console de cluster Red Hat OpenShift.
2. Depuis le panneau de navigation, cliquez sur **Operators** > **OperatorHub**.

   La page OperatorHub s'affiche.
3. Dans la zone **All Items**, entrez "IBM MQ".

   L'entrée de catalogue IBM MQ est affichée.
4. Sélectionnez **IBM MQ**.

   La fenêtre IBM MQ s'ouvre.
5. Cliquez sur **Install**.

   La page Install Operator s'affiche.
6. Entrez les valeurs suivantes :

   a) Définissez **Channel** sur la version que vous avez choisie.

   Consultez la rubrique «Versions prises en charge pour IBM MQ Operator», à la page 15 pour déterminer le canal d'opérateur à sélectionner.

   b) Définissez le **Mode d'installation** sur "un espace de nom spécifique sur le cluster" (que vous pouvez créer à l'étape suivante) ou sur la portée à l'échelle du cluster.

   Il est recommandé de choisir la portée à l'échelle du cluster, car l'installation de différentes versions d'un opérateur dans différents espaces de nom peut entraîner des problèmes. Les opérateurs sont conçus pour être des extensions du plan de contrôle.

c) Facultatif : Si vous avez choisi "un espace de nom spécifique sur le cluster", définissez **Espace de nom** sur la valeur de projet (espace de nom) dans laquelle vous souhaitez installer l'opérateur.

Remarque : Lorsque vous utilisez la console pour installer l'opérateur, vous pouvez utiliser un espace de nom existant, l'espace de nom par défaut fourni par l'opérateur, ou créer un nouvel espace de nom. Si vous souhaitez créer un espace de nom, vous pouvez le créer à partir de ce formulaire, comme suit: dans le panneau de navigation, cliquez sur **Accueil** > **Projets**, sélectionnez **Créer un projet**, indiquez le **nom** du projet (l'espace de nom) à créer, puis cliquez sur **Créer**.

d) Définissez **Stratégie d'approbation** sur Automatique.

7. Cliquez sur **Installer** et attendez l'installation de votre opérateur.

   Une confirmation vous est fournie lorsque l'installation est terminée.

   Pour vérifier l'installation, accédez à **Operators** > **Installed Operators**et sélectionnez votre projet dans la liste déroulante **Projects** . Le statut de l'opérateur passe à Réussi lorsque l'installation est terminée.

Que faire ensuite

Vous êtes maintenant prêt à créer le secret de clé d'autorisation (étape 6 de la section «Installation d'IBM MQ Operator», à la page 34).

OpenShift CP4I *Installation de IBM MQ Operator à l'aide de l'interface de ligne de commande Red Hat OpenShift*

IBM MQ Operator peut être installé sur Red Hat OpenShift à l'aide de l'interface de ligne de commande (CLI).

Avant de commencer

Cette tâche suppose que vous avez effectué les étapes 1 à 4 de «Installation d'IBM MQ Operator», à la page 34.

Procédure

1. Connectez-vous à l'interface de ligne de commande Red Hat OpenShift à l'aide de **`oc login`**.
2. Facultatif : Créez un espace de nom à utiliser pour le IBM MQ Operator.

   IBM MQ Operator peut être installé dans un espace de nom unique ou dans tous les espaces de nom. Cette étape est nécessaire uniquement si vous souhaitez effectuer l'installation dans un espace de nom particulier qui n'existe pas déjà.

   Pour créer un nouvel espace de nom dans l'interface de ligne de commande, exécutez la commande suivante:

   ```
   oc create namespace namespace_name
   ```

   Où *nom_espace_nom* est le nom de l'espace de nom que vous souhaitez créer.
3. Affichez la liste des opérateurs disponibles pour le cluster à partir de OperatorHub:

   ```
   oc get packagemanifests -n openshift-marketplace
   ```
4. Inspectez le IBM MQ Operator pour vérifier s'il est pris en charge **`InstallModes`** et disponible **`Channels`**.

   ```
   oc describe packagemanifests ibm-mq -n openshift-marketplace
   ```
5. Facultatif : Créez un groupe **`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 souscrirez l'opérateur doit avoir un **`OperatorGroup`** qui correspond au **`InstallMode`**de l'opérateur, soit le mode `AllNamespaces` ou `SingleNamespace`.

Si l'opérateur que vous souhaitez installer utilise le mode `AllNamespaces` , l'espace de nom `openshift-operators` possède déjà un **OperatorGroup** approprié et vous pouvez ignorer cette étape.

Si l'opérateur utilise le mode `SingleNamespace` et que vous ne disposez pas déjà d'un **OperatorGroup** approprié, créez-en un en exécutant la commande suivante:

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: operatorgroup_name
  namespace: namespace_name
spec:
  targetNamespaces:
  - namespace_name
EOF
```

6. Consultez la rubrique «Versions prises en charge pour IBM MQ Operator», à la page 15 pour déterminer le canal d'opérateur à sélectionner.
7. Installez l'opérateur.

   Utilisez la commande suivante, en remplaçant *ibm-mq-operator-channel* par le canal correspondant à la version de l'opérateur IBM MQ que vous souhaitez installer, et en remplaçant *nom_espace de nom* par **openshift-operators** si vous utilisez le mode "AllNamespaces", ou par l'espace de nom dans lequel vous souhaitez déployer l'opérateur IBM MQ si vous utilisez le mode "SingleNamespace".

   ```
   cat << EOF | oc apply -f -
   apiVersion: operators.coreos.com/v1alpha1
   kind: Subscription
   metadata:
     name: ibm-mq
     namespace: namespace_name
   spec:
     channel: ibm-mq-operator-channel
     installPlanApproval: Automatic
     name: ibm-mq
     source: ibm-operator-catalog
     sourceNamespace: openshift-marketplace
   EOF
   ```

8. Après quelques minutes, l'opérateur est installé. Exécutez la commande suivante pour vérifier que tous les composants sont à l'état Réussite:

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

   Où *nom_espace_nom* est **openshift-operators** si vous utilisez le mode "AllNamespaces" ou le nom du projet (espace de nom) si vous utilisez le mode "SingleNamespace".

Que faire ensuite

Vous êtes maintenant prêt à créer le secret de clé d'autorisation (étape 6 de la section «Installation d'IBM MQ Operator», à la page 34).

OpenShift CP4I Installation du IBM MQ Operator pour une utilisation avec CP4I

Pour une utilisation avec le IBM Cloud Pak for Integration (CP4I), le IBM MQ Operator peut être installé sur Red Hat OpenShift via la console OpenShift ou l'interface de ligne de commande (CLI).

Avant de commencer

Important :

- Cette rubrique concerne l'installation de IBM MQ Operator en vue de son utilisation avec CP4Iou si vous prévoyez de déployer au moins l'un de vos gestionnaires de files d'attente à l'aide d'une CP4I licence

uniquement. Pour obtenir des instructions sur l'installation de IBM MQ Operator pour une utilisation autonome, voir «Installation d'IBM MQ Operator», à la page 34.

- Consultez les conseils relatifs à la structuration de votre déploiement avant d'installer le IBM MQ Operator.

Pour vous assurer que votre installation se déroule aussi facilement que possible, assurez-vous que vous comprenez tous les prérequis et exigences avant de commencer l'installation. Voir «Planification d'IBM MQ dans des conteneurs», à la page 7.

Pourquoi et quand exécuter cette tâche

Les étapes suivantes représentent le flux de tâches standard pour l'installation de votre IBM MQ Operator:

1. Installez Red Hat OpenShift Container Platform.
2. Configurer le stockage.
3. Images miroir (air-gap uniquement).
4. Ajoutez le catalogue IBM MQ Operator et préparez votre cluster.
5. Installez IBM MQ Operator.
6. Créez le secret de clé d'autorisation (installations en ligne uniquement).
7. Facultatif: Installez IBM Cloud Pak for Integration (CP4I) et ses dépendances.
8. Déployez le License Service.
9. Déployez un gestionnaire de files d'attente.

Procédure

1. Installez Red Hat OpenShift Container Platform.

   Pour connaître les étapes détaillées d'installation de OpenShift, voir Installation du logiciel Red Hat 4.6 ou version ultérieure.

   Important : Veillez à installer une version prise en charge de OpenShift Container Platform. Par exemple, pour utiliser IBM MQ Operator 3.2 ou version ultérieure, vous devez installer OpenShift Container Platform 4.12 ou version ultérieure. Pour plus d'informations, voir IBM Cloud Pak and Red Hat OpenShift Container Platform compatibility.

   Pour toute étape utilisant l'interface de ligne de commande Red Hat OpenShift Container Platform , vous devez être connecté à votre cluster OpenShift avec `oc login`. Pour installer l'interface de ligne de commande, voir Initiation à l'interface de ligne de commande OpenShift.

   Après avoir installé OpenShift, vous pouvez vérifier et accéder à votre logiciel de conteneur à l'aide de la clé d'autorisation IBM que vous créez dans Créer le secret de clé d'autorisation.
2. Configurez le stockage.

   Vous devez définir des classes de stockage dans Red Hat OpenShift Container Platform et définir votre configuration de stockage pour répondre à vos exigences de dimensionnement.

   Important : Les gestionnaires de files d'attente IBM MQ mono-instance et Native HA peuvent utiliser le mode d'accès RWO, tandis que les gestionnaires de files d'attente multi-instance requièrent RWX comme décrit dans «Planification du stockage pour le IBM MQ Operator», à la page 17. Les gestionnaires de files d'attente multi-instance IBM MQ requièrent des caractéristiques de système de fichiers particulières, qui peuvent être vérifiées à l'aide des instructions de la rubrique Test d'un système de fichiers partagé pour IBM MQ.

   Vous trouverez la liste des systèmes de fichiers compatibles et non conformes connus, ainsi que des remarques sur les autres limites ou restrictions, dans l' instruction de test pour les systèmes de fichiers IBM MQ.

Les fournisseurs de stockage recommandés sont disponibles sur la page CP4I Remarques sur le stockage .

3. Images miroir (air-gap uniquement).

   Si votre cluster se trouve dans un environnement réseau restreint (avec isolation physique), vous devez mettre en miroir les images IBM MQ . En fonction de votre configuration, vous devrez peut-être également mettre en miroir certains composants supplémentaires. Lisez les informations suivantes, puis reproduisez les images selon vos besoins.

   - Vous devez mettre en miroir les images IBM MQ . Utilisez les valeurs suivantes :

     ```
     export OPERATOR_PACKAGE_NAME=ibm-mq
     export OPERATOR_VERSION=3.2.1
     ```

   - Vous devez également mettre en miroir des composants supplémentaires requis si vous prévoyez de déployer au moins un gestionnaire de files d'attente où **toutes** les instructions suivantes sont vraies:

     – Vous utilisez une licence CP4I .

     – Le IBM MQ Console est activé.

     – Vous utilisez le service IBM Cloud Pak for Integration Keycloak pour l'authentification et l'autorisation de la connexion unique (SSO) IBM MQ Console (valeur par défaut).

     Si toutes les instructions précédentes sont vraies, la connexion unique est fournie par Keycloak. Par conséquent, de même que pour la source de catalogue IBM MQ Operator , vous devez également répéter les étapes pour chacun de ces composants supplémentaires requis:

     – IBM Cloud Pak foundational services

     – IBM Cloud Pak for Integration

     – Keycloak (opérateurRed Hat OpenShift)

   Pour créer des images miroir, voir Mirroring images for an air-gap cluster.

4. Ajoutez la source de catalogue IBM MQ Operator .

   Ajoutez la source de catalogue qui rend le IBM MQ Operator disponible pour votre cluster à l'aide des valeurs suivantes:

   ```
   export OPERATOR_PACKAGE_NAME=ibm-mq
   export OPERATOR_VERSION=3.2.1
   export ARCH=ARCHITECTURE
   ```

   où *ARCHITECTURE* fait référence à votre architecture système et a la valeur `amd64`, `ppc64le`ou `s390x`.

   Des composants supplémentaires sont requis lorsque vous déployez au moins un gestionnaire de files d'attente où **toutes** les instructions suivantes sont vraies:

   - Vous utilisez une licence CP4I .
   - Le IBM MQ Console est activé.
   - Vous utilisez le service IBM Cloud Pak for Integration Keycloak pour l'authentification et l'autorisation de la connexion unique (SSO) IBM MQ Console (valeur par défaut).

   Si toutes les instructions précédentes sont vraies, la connexion unique est fournie par Keycloak. Par conséquent, de même que pour la source de catalogue IBM MQ Operator , vous devez également répéter les étapes pour chacun de ces composants supplémentaires requis:

   - IBM Cloud Pak foundational services
   - IBM Cloud Pak for Integration
   - Keycloak (opérateurRed Hat OpenShift)

   Suivez les étapes correspondant à vos sources de catalogue requises dans Ajout de sources de catalogue à un cluster.

5. Installez IBM MQ Operator.

Choisissez l'une des deux options suivantes (utilisez la console ou l'interface de ligne de commande):

- Option 1: Installez IBM MQ Operator à l'aide de la console OpenShift.
- Option 2: Installez IBM MQ Operator à l'aide de l'interface de ligne de commande OpenShift.

6. Créez le secret de la clé d'autorisation (installations en ligne uniquement).

   IBM MQ Operator déploie des images de gestionnaire de files d'attente extraites d'un registre de conteneur qui effectue une vérification des autorisations de licence. Ce contrôle requiert une clé d'autorisation qui est stockée dans un secret d'extraction `docker-registry`. Si vous ne disposez pas encore d'une clé d'autorisation dans l'espace de nom dans lequel vous allez installer les gestionnaires de files d'attente, suivez ces instructions pour obtenir une clé d'autorisation et créer un secret d'extraction.

   Remarque : La clé d'autorisation n'est pas requise si seuls les gestionnaires de files d'attente IBM MQ Advanced for Developers (non garantis) vont être déployés.

   Vous pouvez créer le secret de la clé d'autorisation à l'aide de la console OpenShift ou de l'interface de ligne de commande. L'exemple suivant utilise l'interface de ligne de commande:

   a. Obtenez la clé d'autorisation affectée à votre ID IBM . 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.

   c. A partir de l'interface de ligne de commande OpenShift , exécutez la commande suivante pour créer un secret d'extraction d'image appelé `ibm-entitlement-key`.

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

   Où *entitlement_key* est la clé d'autorisation que vous avez copiée à l'étape b, *user_email>* est l'ID IBM associé au logiciel autorisé et *namespace* est l'espace de nom dans lequel vous avez installé votre IBM MQ Operator .

7. Facultatif : Installez CP4I et ses dépendances.

   Des composants supplémentaires sont requis lorsque vous déployez au moins un gestionnaire de files d'attente où **toutes** les instructions suivantes sont vraies:

   - Vous utilisez une licence CP4I .
   - Le IBM MQ Console est activé.
   - Vous utilisez le service CP4I Keycloak pour l'authentification et l'autorisation de la connexion unique (SSO) IBM MQ Console (valeur par défaut).

   Si toutes les instructions précédentes sont vraies, la connexion unique est fournie par Keycloak et vous devez effectuer les étapes supplémentaires suivantes:

   - Installez l'opérateur IBM Cloud Pak foundational services dans le même mode d'installation que l'opérateur CP4I . Pour connaître les versions prises en charge, voir Operator channel versions for this release.
   - Installez l'opérateur CP4I.
   - Facultatif: déployez l'interface utilisateur de la plateforme.

     a. Créez l'espace de nom `ibm-common-services` . Une fois connecté à votre cluster OpenShift via l'interface de ligne de commande, exécutez la commande suivante:

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

     b. Déployez l'interface utilisateur de la plateforme.

8. Déployez le License Service.

   Cette opération est requise pour la surveillance de l'utilisation des licences des gestionnaires de files d'attente. Suivez les instructions de la rubrique Déploiement du service de licence License Service.
9. Déployez un gestionnaire de files d'attente.

   Pour des instructions sur le déploiement d'un exemple de gestionnaire de files d'attente de "démarrage rapide" , voir «Déploiement d'un gestionnaire de files d'attente simple à l'aide de IBM MQ Operator», à la page 66.

Tâches associées

«Désinstallation de IBM MQ Operator», à la page 55
Vous pouvez utiliser la console ou l'interface de ligne de commande Red Hat OpenShift pour désinstaller IBM MQ Operator de Red Hat OpenShift.

OpenShift Operator2.0.0 CP4I Mise à niveau de IBM MQ Operator et des gestionnaires de files d'attente

Il existe différents processus de mise à niveau pour les utilisateurs du IBM MQ Operator, selon que vous utilisez des licences IBM MQ ou des licences IBM Cloud Pak for Integration (CP4I). Effectuez l'étape de mise à niveau pour votre type de déploiement.

Pourquoi et quand exécuter cette tâche

Pour mettre à niveau votre IBM MQ Operator et vos gestionnaires de files d'attente, effectuez l'une des opérations suivantes:

Procédure

- Option 1: **Mettez à niveau les déploiements vers la version la plus récente sur votre canal d'opérateur en cours.**

  Pour mettre à niveau les déploiements du IBM MQ Operator vers la version la plus récente sur votre canal d'opérateur en cours, voir «Mise à niveau vers une édition de sécurité la plus récente du canal IBM MQ Operator», à la page 46.
- Option 2: **Mettez à niveau les licences IBM MQ Operator for IBM MQ .**

  Pour mettre à niveau des déploiements du IBM MQ Operator où **seules** licences IBM MQ sont utilisées, voir «Mise à niveau du IBM MQ Operator», à la page 45.
- Option 3: **Mettez à niveau les utilisateurs IBM MQ Operator for CP4I .**

  Mise à niveau des déploiements du IBM MQ Operator pour les utilisateurs du IBM Cloud Pak for Integration. Cela inclut si vous avez déployé au moins un de vos gestionnaires de files d'attente sous une licence CP4I . Voir «Mise à niveau du IBM MQ Operator pour les utilisateurs CP4I», à la page 51.

OpenShift *Mise à niveau du IBM MQ Operator*

Mettez à niveau les déploiements du IBM MQ Operator où **seules les licences** IBM MQ sont utilisées.

Avant de commencer

Important : Cette tâche est destinée aux utilisateurs des licences IBM MQ Operator et **uniquement** IBM MQ . Si vous êtes un utilisateur IBM Cloud Pak for Integration (CP4I) ou si vous avez déployé au moins un de vos gestionnaires de files d'attente à l'aide d'une licence CP4I , voir «Mise à niveau du IBM MQ Operator pour les utilisateurs CP4I», à la page 51.

Pourquoi et quand exécuter cette tâche

Effectuez l'une des étapes suivantes qui correspond à la mise à niveau dont vous avez besoin.

Remarque : La version 3.2.x de IBM MQ Operator a été publiée en tant qu'édition CD et SC2 .

Procédure

- Option 1 :«Mise à niveau vers une édition de sécurité la plus récente du canal IBM MQ Operator», à la page 46
- Option 2 : «Mise à niveau d'une LTS IBM MQ Operator 2.0.x vers le canal 3.2.x SC2/CD», à la page 47
- Option 3 :«Mise à niveau d'un CD IBM MQ Operator vers le canal 3.2.x SC2/CD», à la page 48

Mise à niveau vers une édition de sécurité la plus récente du canal IBM MQ Operator

La mise à niveau de IBM MQ Operator vous permet de mettre à niveau vos gestionnaires de files d'attente.

Avant de commencer

Important : Cette rubrique concerne la mise à niveau des déploiements de IBM MQ Operator vers la dernière édition de sécurité sur le canal du déploiement. Si cela ne s'applique pas à votre déploiement, reportez-vous aux autres chemins de mise à niveau décrits dans «Mise à niveau de IBM MQ Operator et des gestionnaires de files d'attente», à la page 45.

Pourquoi et quand exécuter cette tâche

Vous devez d'abord mettre à niveau la source de catalogue, puis mettre à niveau les gestionnaires de files d'attente. Il existe deux options, en fonction de la source de catalogue utilisée pour déployer le IBM MQ Operator en cours de mise à niveau.

Option 1: Source de catalogue spécifique pour le IBM MQ Operator

Une nouvelle version de IBM MQ Operator devient disponible dans un OpenShift **uniquement** après la mise à jour de la source de catalogue. Ce processus vous permet de contrôler manuellement les mises à niveau. Vous n'avez donc pas besoin d'utiliser l'option **Manuel** pour le paramètre **`Update approval`** des opérateurs. L'option **Manuel** force toutes les mises à niveau possibles à être effectuées en même temps et peut bloquer les mises à niveau. Par conséquent, utilisez l'option **Automatique** uniquement. Pour plus d'informations, voir la section "Restriction des mises à jour automatiques avec une stratégie d'approbation" de la rubrique Installation des opérateurs à l'aide de la console Red Hat OpenShift.

Pour utiliser cette option, passez à la section Mise à niveau avec la source de catalogue spécifique pour IBM MQ Operator.

Option 2: IBM Operator Catalog

Avec cette option, de nouvelles versions d'opérateur deviennent disponibles et sont appliquées **sans** intervention de votre part. Par conséquent, utilisez cette option **uniquement** pour les installations en ligne où vous souhaitez des mises à niveau **automatiques** du IBM MQ Operatoret où les installations déterministes ne sont pas nécessaires. Cette option peut être utile pour les environnements de démonstration de faisabilité, mais elle n'est **pas adaptée aux environnements de production**.

Pour utiliser cette option, passez à la section Mise à niveau avec le catalogue d'opérateurs IBM.

Pour passer de l'utilisation du catalogue d'opérateurs IBM à l'utilisation de la source de catalogue spécifique pour IBM MQ Operator, qui vous permet de mieux contrôler les mises à niveau, voir «Passage à la source de catalogue spécifique pour le IBM MQ Operator», à la page 49.

Procédure

- **Mise à niveau avec la source de catalogue spécifique pour IBM MQ Operator**

  a) Appliquez la dernière source de catalogue.

  Suivez les instructions sous "Ajout de sources de catalogue spécifiques pour le IBM MQ Operator" dans la rubrique Ajout de la source de catalogue IBM MQ Operator.

b) Si vous avez le statut **Mettre à jour l'approbation** pour le IBM MQ Operator défini sur **Automatique**, votre opérateur est mis à niveau. Si vous avez défini l'option **Mettre à jour l'approbation** sur **Manuel**, procédez comme suit pour mettre à niveau le IBM MQ Operator:

a. Depuis le panneau de navigation, cliquez sur **Operators** > **Installed Operators**.

Tous les opérateurs installés dans le projet spécifié sont affichés.

b. Sélectionnez **IBM MQ Operator**.

c. Accédez à l'onglet **Subscription**.

d. Cliquez sur **Mise à niveau disponible**

e. Cliquez sur **Aperçu InstallPlan**

f. Cliquez sur **Approuver** pour terminer la mise à niveau.

L'opérateur est mis à niveau vers la nouvelle version.

c) Mettez à niveau tous les gestionnaires de files d'attente IBM MQ .

Suivez les instructions de la rubrique Mise à niveau des gestionnaires de files d'attente IBM MQ.

- **Mise à niveau avec le catalogue des opérateurs IBM**

a) Mettez à niveau IBM MQ Operator vers une version plus récente.

Si des mises à niveau automatiques sont définies, lors de l'édition d'une nouvelle édition de sécurité, votre IBM MQ Operator effectue une mise à niveau. Si vous ne disposez pas de mises à niveau automatiques, approuvez manuellement votre mise à niveau IBM MQ Operator :

– Si une mise à niveau est disponible, le **`Upgrade Status`** peut être "`Mise à niveau disponible`".

– Dans ce cas, il peut y avoir un contrôle disponible que vous pouvez utiliser pour approuver le **InstallPlan** qui met à niveau le IBM MQ Operator.

b) Mise à niveau des gestionnaires de files d'attente IBM MQ

Suivez les instructions de la rubrique Mise à niveau des gestionnaires de files d'attente IBM MQ.

- **Mettez à niveau les gestionnaires de files d'attente IBM MQ**.

Vous devez mettre à niveau les gestionnaires de files d'attente IBM MQ vers une version plus récente après la mise à niveau d' IBM MQ Operator.

Le tableau suivant décrit la version la plus récente du gestionnaire de files d'attente IBM MQ pour chaque canal opérateur actif. A l'aide de la version appropriée, suivez la procédure décrite dans «Mise à niveau d'un gestionnaire de files d'attente IBM MQ avec Red Hat OpenShift», à la page 53.

Canal opérateur	Dernier gestionnaire de files d'attente IBM MQ
3.2 (SC2/CD)	9.4.0.0-r1

OpenShift Operator2.0.0 V 9.4.0 CP4I *Mise à niveau d'une LTS IBM MQ Operator 2.0.x vers le canal 3.2.x SC2/CD*

La mise à niveau d' IBM MQ Operator vous permet de mettre à niveau vos gestionnaires de files d'attente.

Avant de commencer

Important :

- Cette tâche est destinée aux utilisateurs des licences IBM MQ Operator et **uniquement** IBM MQ . Si vous êtes un utilisateur IBM Cloud Pak for Integration (CP4I) ou si vous avez déployé au moins un de vos gestionnaires de files d'attente à l'aide d'une licence CP4I , voir «Mise à niveau du IBM MQ Operator pour les utilisateurs CP4I», à la page 51.
- Cette rubrique concerne la mise à niveau des déploiements de 2.0.x Long Term Support (LTS) IBM MQ Operator vers le canal Support Cycle 2 (SC2) de IBM MQ Operator 3.2.x **uniquement**. Si cela ne

s'applique pas à votre déploiement, consultez les autres chemins de mise à niveau décrits dans «Mise à niveau de IBM MQ Operator et des gestionnaires de files d'attente», à la page 45.

Pour effectuer une mise à niveau vers IBM MQ Operator 3.2.1 , vous devez exécuter Red Hat OpenShift Container Platform 4.12 ou une version ultérieure. Pour vérifier les versions compatibles de chaque canal IBM MQ Operator , voir «Versions Red Hat OpenShift Container Platform compatibles», à la page 15. Pour mettre à niveau la plateforme, voir Mise à niveau d' Red Hat OpenShift.

Procédure

1. **Images miroir (air-gap uniquement).**

   Vous devez mettre en miroir les images IBM MQ . Effectuez les étapes à l'aide du lien suivant, en utilisant uniquement ces valeurs:

   ```
   export OPERATOR_PACKAGE_NAME=ibm-mq
   export OPERATOR_VERSION=3.2.1
   ```

   Vous devez omettre la section 3.5 "Configurer le cluster", car la connexion au registre d'images doit avoir été configurée lors des installations ou des mises à niveau précédentes.

   Lien: Images de mise en miroir pour un cluster avec isolation physique.
2. **Mettez à niveau votre IBM MQ Operator vers 3.2.1.**

   Voir «Mise à niveau du IBM MQ Operator à l'aide de Red Hat OpenShift», à la page 51.
3. **Mettez à niveau les instances.**

   Pour recevoir les fonctions et les correctifs de sécurité les plus récents, mettez à niveau l'opérateur IBM MQ (image du conteneur de gestionnaire de files d'attente) vers la version la plus récente d' CD (9.4.0.0-r1). Voir «Mise à niveau d'un gestionnaire de files d'attente IBM MQ avec Red Hat OpenShift», à la page 53.

OpenShift CP4I V 9.4.0 *Mise à niveau d'un CD IBM MQ Operator vers le canal 3.2.x SC2/CD*

La mise à niveau d' IBM MQ Operator vous permet de mettre à niveau vos gestionnaires de files d'attente.

Avant de commencer

Important :

- Cette tâche est destinée aux utilisateurs des licences IBM MQ Operator et **uniquement** IBM MQ . Si vous êtes un utilisateur IBM Cloud Pak for Integration (CP4I) ou si vous avez déployé au moins un de vos gestionnaires de files d'attente à l'aide d'une licence CP4I , voir «Mise à niveau du IBM MQ Operator pour les utilisateurs CP4I», à la page 51.
- Cette rubrique concerne la mise à niveau des déploiements Continuous Delivery (CD) du IBM MQ Operator antérieur à la version 3.2.0, vers la version 3.2.1 **uniquement**. Si cela ne s'applique pas à votre déploiement, consultez les autres chemins de mise à niveau décrits dans «Mise à niveau de IBM MQ Operator et des gestionnaires de files d'attente», à la page 45.

Pour effectuer une mise à niveau vers IBM MQ Operator 3.2.1 , vous devez exécuter Red Hat OpenShift Container Platform 4.12 ou une version ultérieure. Pour vérifier les versions compatibles de chaque canal IBM MQ Operator , voir «Versions Red Hat OpenShift Container Platform compatibles», à la page 15. Pour mettre à niveau la plateforme, voir Mise à niveau d' Red Hat OpenShift.

Procédure

1. Facultatif : **Mettez à niveau un IBM MQ Operator dont la version est actuellement CD antérieure à 3.0.0.**

   Si votre IBM MQ Operator est actuellement à une version de CD antérieure à 3.0.0, suivez les étapes appropriées de la rubrique Migration vers le canal CD en cours de l'opérateur IBM MQ

(documentationIBM MQ 9.3), puis revenez ici pour effectuer la mise à niveau vers la version la plus récente de CD . Notez qu'il s'agit d'une étape prérequise obligatoire avant la mise à niveau vers la version 3.2.1.

2. **Images miroir (air-gap uniquement).**

   Vous devez mettre en miroir les images IBM MQ . Effectuez les étapes à l'aide du lien suivant, en utilisant uniquement ces valeurs:

   ```
   export OPERATOR_PACKAGE_NAME=ibm-mq
   export OPERATOR_VERSION=3.2.1
   ```

   Vous devez omettre la section 3.5 "Configurer le cluster", car la connexion au registre d'images doit avoir été configurée lors des installations ou des mises à niveau précédentes.

   Lien: Images de mise en miroir pour un cluster avec isolation physique.

3. **Mettez à niveau votre IBM MQ Operator vers 3.2.1.**

   Voir «Mise à niveau du IBM MQ Operator à l'aide de Red Hat OpenShift», à la page 51.

4. **Mettez à niveau les instances.**

   Pour recevoir les fonctions et les correctifs de sécurité les plus récents, mettez à niveau l'opérateur IBM MQ (image du conteneur de gestionnaire de files d'attente) vers la version la plus récente d' CD (9.4.0.0-r1). Voir «Mise à niveau d'un gestionnaire de files d'attente IBM MQ avec Red Hat OpenShift», à la page 53.

OpenShift V 9.4.0 CP4I V 9.4.0 *Passage à la source de catalogue spécifique pour le IBM MQ Operator*

Si vous disposez d'une installation de IBM MQ Operator à partir d'une édition précédente et que vous utilisez le catalogue de l'opérateur IBM , l'application de la source de catalogue spécifique est le moyen le plus efficace de contrôler complètement la gestion des versions de logiciels sur un cluster.

Avant de commencer

Important : Cette tâche doit être effectuée par un administrateur de cluster. Voir OpenShift Roles and permissions.

Les étapes suivantes sont effectuées à l'aide de l'interface de ligne de commande.

Pourquoi et quand exécuter cette tâche

Le catalogue des opérateurs IBM est un index des opérateurs disponibles pour étendre l'API d'un cluster Red Hat OpenShift Container Platform afin d'activer les produits logiciels IBM .

Cette procédure déplace une installation du IBM MQ Operator à partir du catalogue de l'opérateur IBM afin que vous puissiez utiliser la source de catalogue spécifique pour le IBM MQ Operator.

Procédure

1. Ajoutez le catalogue IBM MQ Operator .

   Suivez les instructions sous "Ajout de sources de catalogue spécifiques pour le IBM MQ Operator" dans la rubrique Ajout de la source de catalogue IBM MQ Operator.

2. Vérifiez que la source de catalogue IBM MQ Operator a été créée dans l'espace de nom `openshift-marketplace` .

   Exécutez ensuite la commande suivante :

   ```
   oc get catalogsource -n openshift-marketplace
   ```

   Exemple de sortie :

```
oc get catalogsource -n openshift-marketplace

NAME                           DISPLAY                TYPE   PUBLISHER   AGE
ibm-operator-catalog           IBM Operator Catalog   grpc   IBM         23h
ibmmq-operator-catalogsource   ibm-mq-3.1.3           grpc   IBM         23h
```

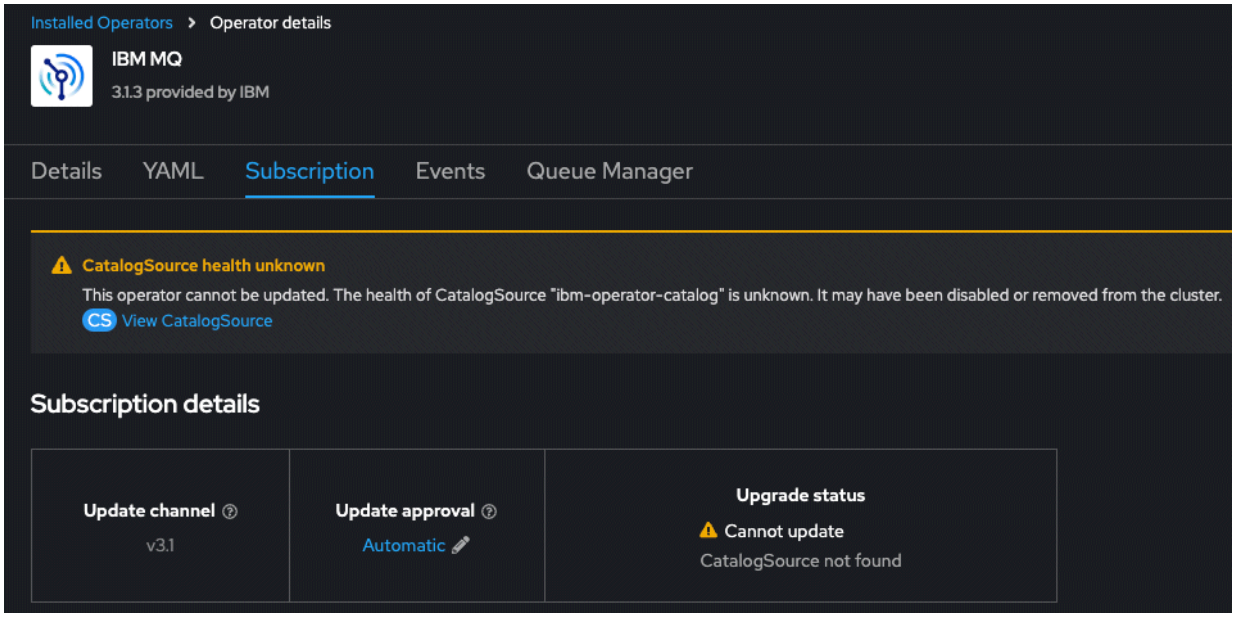
3. Facultatif : Supprimez la source de catalogue de l'opérateur IBM .

   Avertissement : Vous ne devez effectuer cette étape que si vous êtes certain qu'aucun autre opérateur n'utilise le catalogue d'opérateurs IBM .

   Exécutez ensuite la commande suivante :

   ```
   oc delete catalogsource ibm-operator-catalog -n openshift-marketplace
   ```

   Le statut IBM MQ Operator passe à `CatalogSource not found`. Ceci arrive fréquemment.



4. Modifiez l'abonnement de IBM MQ Operator pour qu'il pointe vers la nouvelle source de catalogue IBM MQ Operator spécifique.

   a) Editez l'abonnement.

      Exécutez la commande suivante en remplaçant *OPERATOR-NAMESPACE* par `openshift-operators` pour les installations du IBM MQ Operatorà l'échelle du cluster ou par l'espace de nom spécifique dans lequel IBM MQ Operator est déployé:

      ```
      oc edit subscription ibm-mq -n OPERATOR-NAMESPACE
      ```

   b) Remplacez la valeur `spec.source ibm-operator-catalog` par le nom de la source de catalogue créée à l'étape «1», à la page 49.

      Exemple :

      ```
      spec:
        channel: v3.1
        installPlanApproval: Automatic
        name: ibm-mq
        source: ibm-operator-catalog # CHANGE --> ibmmq-operator-catalogsource
        sourceNamespace: openshift-marketplace
      ```

   c) Sauvegardez les modifications.

      L'installation IBM MQ Operator pointe désormais vers la source de catalogue IBM MQ Operator . Si vous avez supprimé le catalogue de l'opérateur IBM , le statut "CatalogSource not found" devient "Succeeded".

Résultats

Votre installation de IBM MQ Operator pointe désormais vers la source de catalogue spécifique pour IBM MQ Operator. Vous disposez ainsi d'un contrôle total sur les mises à niveau de l'opérateur.

OpenShift CP4I *Mise à niveau du IBM MQ Operator pour les utilisateurs CP4I*

Mettez à niveau les déploiements du IBM MQ Operator où une licence IBM Cloud Pak for Integration (CP4I) est utilisée.

Avant de commencer

Important : Cette tâche est destinée aux utilisateurs CP4I . Cela inclut si vous avez déployé au moins un de vos gestionnaires de files d'attente sous une licence CP4I . Si cela ne s'applique pas à vous, voir «Mise à niveau du IBM MQ Operator», à la page 45.

Pourquoi et quand exécuter cette tâche

Choisissez l'une des options suivantes :

Procédure

- **Option 1:** Mise à niveau des déploiements de 2.0.x Long Term Support (LTS) IBM MQ Operator

  Suivez les étapes de la rubrique Mise à niveau depuis 2022.2 en générant un plan de mise à niveau.
- **Option 2:** Mise à niveau d'un déploiement 3.0.x ou 3.1.x du IBM MQ Operator

  Suivez les étapes de la rubrique Mise à niveau à partir de 2023.4 en générant un plan de mise à niveau.
- **Option 3:** Mise à niveau d'autres déploiements de IBM MQ Operator

  Suivez les étapes appropriées dans Migration vers le canal CD en cours du IBM MQ Operator (documentationIBM MQ 9.3), puis revenez ici et passez à l' **Option 2**. Notez qu'il s'agit d'une étape prérequise obligatoire.

OpenShift CP4I *Mise à niveau du IBM MQ Operator à l'aide de Red Hat OpenShift*

Vous pouvez mettre à niveau le IBM MQ Operator à l'aide de la console Web ou de l'interface de ligne de commande Red Hat OpenShift .

Procédure

Pour mettre à niveau le IBM MQ Operator à l'aide de Red Hat OpenShift, effectuez l'une des tâches suivantes:

- «Mise à niveau du IBM MQ Operator à l'aide de la console Red Hat OpenShift», à la page 51
- «Mise à niveau de IBM MQ Operator à l'aide de l'interface CLI Red Hat OpenShift», à la page 52

OpenShift CP4I *Mise à niveau du IBM MQ Operator à l'aide de la console Red Hat OpenShift*

IBM MQ Operator peut être mis à niveau à l'aide d'Operator Hub.

Avant de commencer

Remarque : La version CD la plus récente de IBM MQ Operator est 3.2.1. Il s'agit à la fois d'une version SC2 et d'une version CD . Pour les notes sur l'édition les plus récentes d' IBM MQ Operator , voir Release history for IBM MQ Operator.

Connectez-vous à votre console de cluster Red Hat OpenShift.

Procédure

1. Consultez la rubrique «Versions prises en charge pour IBM MQ Operator», à la page 15 pour déterminer le canal d'opérateur vers lequel vous souhaitez effectuer la mise à niveau.
2. Appliquez la source de catalogue la plus récente.

   Si vous utilisez la source de catalogue spécifique pour le IBM MQ Operator, plutôt que le `ibm-operator-catalog`, vous devez appliquer la source de catalogue pour la nouvelle version de IBM MQ .

   Pour passer de l'utilisation du catalogue d'opérateurs IBM à l'utilisation de la source de catalogue spécifique pour le IBM MQ Operatoret pour mieux contrôler les mises à niveau, reportez-vous aux étapes de la rubrique «Passage à la source de catalogue spécifique pour le IBM MQ Operator», à la page 49 avant de revenir à l'étape «3», à la page 52.

   Si vous utilisez le catalogue de l'opérateur IBM (certaines installations en ligne uniquement), passez à l'étape «3», à la page 52.

   Suivez les instructions présentées dans «Ajout de la source de catalogue IBM MQ Operator», à la page 36.
3. Mettez à niveau IBM MQ Operator. De nouvelles versions IBM MQ Operator majeures/mineures sont distribuées via les nouveaux canaux d'abonnement. Pour mettre à niveau votre opérateur vers une nouvelle version majeure/mineure, vous devez mettre à jour le canal sélectionné dans votre abonnement IBM MQ Operator.
   a) Depuis le panneau de navigation, cliquez sur **Operators** > **Installed Operators**.

      Tous les opérateurs installés dans le projet spécifié sont affichés.
   b) Sélectionnez **IBM MQ Operator**.
   c) Accédez à l'onglet **Subscription**.
   d) Cliquez sur **Channel**.

      La fenêtre **Change Subscription Update Channel** s'ouvre.
   e) Sélectionnez le canal de votre choix et cliquez sur **Save**.

      L'opérateur est mis à niveau avec la version la plus récente disponible sur le nouveau canal. Voir «Versions prises en charge pour IBM MQ Operator», à la page 15.

OpenShift CP4I *Mise à niveau de IBM MQ Operator à l'aide de l'interface CLI Red Hat OpenShift*

IBM MQ Operator peut être mis à niveau à partir de la ligne de commande.

Avant de commencer

Remarque : La version CD la plus récente de IBM MQ Operator est 3.2.1. Il s'agit à la fois d'une version SC2 et d'une version CD . Pour les notes sur l'édition les plus récentes d' IBM MQ Operator , voir Release history for IBM MQ Operator.

Connectez-vous à votre cluster à l'aide de la commande **`oc login`**.

Procédure

1. Consultez la rubrique «Versions prises en charge pour IBM MQ Operator», à la page 15 pour déterminer le canal d'opérateur vers lequel vous souhaitez effectuer la mise à niveau.
2. Appliquez la source de catalogue la plus récente.

   Si vous utilisez la source de catalogue spécifique pour le IBM MQ Operator, plutôt que le `ibm-operator-catalog`, vous devez appliquer la source de catalogue pour la nouvelle version de IBM MQ .

   Pour passer de l'utilisation du catalogue d'opérateurs IBM à l'utilisation de la source de catalogue spécifique pour le IBM MQ Operatoret pour mieux contrôler les mises à niveau, reportez-vous aux

étapes de la rubrique «Passage à la source de catalogue spécifique pour le IBM MQ Operator», à la page 49 avant de revenir à l'étape «3», à la page 53.

Si vous utilisez le catalogue de l'opérateur IBM (certaines installations en ligne uniquement), passez à l'étape «3», à la page 53.

Suivez les instructions présentées dans «Ajout de la source de catalogue IBM MQ Operator», à la page 36.

3. Mettez à niveau IBM MQ Operator. De nouvelles versions IBM MQ Operator majeures/mineures sont distribuées via les nouveaux canaux d'abonnement. Pour mettre votre opérateur à niveau vers une nouvelle version principale ou mineure, vous devez mettre à jour le canal sélectionné dans votre abonnement IBM MQ Operator.
   a) Vérifiez que le canal de mise à niveau IBM MQ Operator requis est disponible.

```
oc get packagemanifest ibm-mq -o=jsonpath='{.status.channels[*].name}'
```

   b) Utilisez le correctif de `Subscription` pour passer au canal de mise à jour souhaité (où *VX.O* est le canal de mise à jour souhaité identifié à l'étape précédente.

```
oc patch subscription ibm-mq --patch '{"spec":{"channel":"vX.Y"}}' --type=merge
```

OpenShift CP4I *Mise à niveau d'un gestionnaire de files d'attente IBM MQ avec Red Hat OpenShift*

Avant de commencer

Dans le cadre du processus de mise à niveau des gestionnaires de files d'attente IBM MQ , vous avez peut-être été envoyé à cette rubrique à partir de la documentation IBM Cloud Pak for Integration .

Procédure

Pour mettre à niveau le gestionnaire de files d'attente IBM MQ à l'aide de Red Hat OpenShift, effectuez l'une des tâches suivantes:

- «Mise à niveau d'un gestionnaire de files d'attente IBM MQ à l'aide de la console Red Hat OpenShift», à la page 53
- «Mise à niveau d'un gestionnaire de files d'attente IBM MQ à l'aide de Red Hat OpenShift CLI», à la page 54
- «Mise à niveau d'un gestionnaire de files d'attente IBM MQ dans Red Hat OpenShift à l'aide de l'interface utilisateur de la plateforme», à la page 55

Que faire ensuite

Pour effectuer une mise à niveau d' IBM Cloud Pak for Integration , vous devrez peut-être revenir à la documentation IBM Cloud Pak for Integration .

OpenShift CP4I *Mise à niveau d'un gestionnaire de files d'attente IBM MQ à l'aide de la console Red Hat OpenShift*

Un gestionnaire de files d'attente IBM MQ, déployé à l'aide de IBM MQ Operator, peut être mis à niveau dans Red Hat OpenShift à l'aide de l'opérateur Hub.

Avant de commencer

Remarque : La version la plus récente du gestionnaire de files d'attente IBM MQ est 9.4.0.0-r1. Il s'agit à la fois d'une version SC2 et d'une version CD . Pour les dernières notes sur l'édition du gestionnaire de files d'attente IBM MQ , voir Historique d'édition des images de conteneur de gestionnaire de files d'attente à utiliser avec IBM MQ Operator.

- Connectez-vous à votre console Web de cluster Red Hat OpenShift.

- Vérifiez que IBM MQ Operator utilise le canal de mise à jour souhaité. Voir «Mise à niveau du IBM MQ Operator à l'aide de Red Hat OpenShift», à la page 51.

Pour pouvoir mettre à niveau le gestionnaire de files d'attente dans un environnement avec isolation physique, vous devez mettre en miroir les images IBM Cloud Pak for Integration les plus récentes via les étapes spécifiques à l'isolation physique décrites dans la rubrique Mise à niveau d'un CD IBM MQ Operator vers le canal 3.2.x SC2/CD.

Procédure

1. Depuis le panneau de navigation, cliquez sur **Operators** > **Installed Operators**.
   Tous les opérateurs installés dans le projet spécifié sont affichés.
2. Sélectionnez **IBM MQ Operator**.
   La fenêtre **IBM MQ Operator** s'affiche.
3. Accédez à l'onglet **Queue Manager** .
   La fenêtre **Queue Manager Details** s'affiche.
4. Sélectionnez le gestionnaire de files d'attente à mettre à niveau.
5. Accédez à l'onglet YAML.
6. Mettez à jour les zones suivantes, le cas échéant, pour qu'elles correspondent à la mise à niveau de la version du gestionnaire de files d'attente IBM MQ souhaitée.
   - spec.version
   - spec.license.licence

   Voir «Historique des éditions des images de conteneur de gestionnaire de files d'attente à utiliser avec IBM MQ Operator», à la page 7 pour un mappage des versions IBM MQ Operator et des images de conteneur de gestionnaire de files d'attente IBM MQ .
7. Sauvegardez le gestionnaire de files d'attente mis à jour YAML.

OpenShift CP4I *Mise à niveau d'un gestionnaire de files d'attente IBM MQ à l'aide de Red Hat OpenShift CLI*

Un gestionnaire de files d'attente IBM MQ, déployé à l'aide de IBM MQ Operator, peut être mis à niveau dans Red Hat OpenShift à l'aide de la ligne de commande.

Avant de commencer

Remarque : La version la plus récente du gestionnaire de files d'attente IBM MQ est 9.4.0.0-r1. Il s'agit à la fois d'une version SC2 et d'une version CD . Pour les dernières notes sur l'édition du gestionnaire de files d'attente IBM MQ , voir Historique d'édition des images de conteneur de gestionnaire de files d'attente à utiliser avec IBM MQ Operator.

Vous devez être administrateur de cluster pour effectuer les étapes suivantes.

- Connectez-vous à l'interface de ligne de commande Red Hat OpenShift à l'aide de `oc login`.
- Vérifiez que IBM MQ Operator utilise le canal de mise à jour souhaité. Voir «Mise à niveau de IBM MQ Operator et des gestionnaires de files d'attente», à la page 45.

Pour pouvoir mettre à niveau le gestionnaire de files d'attente dans un environnement avec isolation physique, vous devez mettre en miroir les images IBM Cloud Pak for Integration les plus récentes via les étapes spécifiques à l'isolation physique décrites dans la rubrique Mise à niveau d'un CD IBM MQ Operator vers le canal 3.2.x SC2/CD.

Procédure

Editez la ressource **`QueueManager`** pour mettre à jour les zones suivantes, le cas échéant, pour qu'elles correspondent à la mise à niveau de la version du gestionnaire de files d'attente IBM MQ souhaitée.

- spec.version

- spec.license.licence

Voir «Versions prises en charge pour IBM MQ Operator», à la page 15 pour un mappage des canaux vers les versions IBM MQ Operator et les versions du gestionnaire de files d'attente IBM MQ.

Utilisez la commande suivante :

```
oc edit queuemanager my_qmgr
```

où *my_qmgr* est le nom de la ressource QueueManager que vous souhaitez mettre à niveau.

CP4I *Mise à niveau d'un gestionnaire de files d'attente IBM MQ dans Red Hat OpenShift à l'aide de l'interface utilisateur de la plateforme*
Un gestionnaire de files d'attente IBM MQ, déployé à l'aide de IBM MQ Operator, peut être mis à niveau dans Red Hat OpenShift à l'aide de IBM Cloud Pak for Integration Platform UI.

Avant de commencer

Remarque : La version la plus récente du gestionnaire de files d'attente IBM MQ est 9.4.0.0-r1. Il s'agit à la fois d'une version SC2 et d'une version CD . Pour les dernières notes sur l'édition du gestionnaire de files d'attente IBM MQ , voir Historique d'édition des images de conteneur de gestionnaire de files d'attente à utiliser avec IBM MQ Operator.

- Connectez-vous à IBM Cloud Pak for Integration Platform UI dans l'espace de nom contenant le gestionnaire de files d'attente que vous souhaitez mettre à niveau.
- Vérifiez que IBM MQ Operator utilise le canal de mise à jour souhaité. Voir «Mise à niveau de IBM MQ Operator et des gestionnaires de files d'attente», à la page 45.

Pour pouvoir mettre à niveau le gestionnaire de files d'attente dans un environnement avec isolation physique, vous devez mettre en miroir les images IBM Cloud Pak for Integration les plus récentes via les étapes spécifiques à l'isolation physique décrites dans la rubrique Mise à niveau d'un CD IBM MQ Operator vers le canal 3.2.x SC2/CD.

Procédure

1. A partir de la page d'accueil IBM Cloud Pak for Integration Platform UI, cliquez sur l'onglet **Runtimes**.
2. Les gestionnaires de files d'attente dont les mises à niveau sont disponibles ont un **i** bleu en regard de la **version**. Cliquez sur la lettre **i** pour afficher la **nouvelle version disponible**.
3. Cliquez sur les trois points à l'extrême droite du gestionnaire de files d'attente que vous souhaitez mettre à niveau, puis cliquez sur **Change version**.
4. Sous **Select a new channel or version**, sélectionnez la version de mise à niveau requise.
5. Cliquez sur **Change version**.

Résultats

Le gestionnaire de files d'attente est mis à niveau.

OpenShift CP4I Désinstallation de IBM MQ Operator

Vous pouvez utiliser la console ou l'interface de ligne de commande Red Hat OpenShift pour désinstaller IBM MQ Operator de Red Hat OpenShift.

Procédure

- Option 1: Désinstallez IBM MQ Operator à l'aide de la console OpenShift .

  Remarque : Si IBM MQ Operator est installé dans tous les projets / espaces de nom du cluster, répétez les étapes 2 à 6 de la procédure suivante pour chaque projet dans lequel vous souhaitez supprimer des gestionnaires de files d'attente.

a) Connectez-vous à la console Web Red Hat OpenShift Container Platform avec vos données d'identification d'administrateur de cluster Red Hat OpenShift Container Platform .

b) Remplacez **Project** par l'espace de nom à partir duquel vous souhaitez désinstaller IBM MQ Operator. Sélectionnez l'espace de nom dans la liste déroulante **Projet** .

c) Dans le panneau de navigation, cliquez sur **Opérateurs** > **Opérateurs installés**.

d) Cliquez sur l'opérateur **IBM MQ**.

e) Cliquez sur l'onglet **Queue Managers** pour afficher les gestionnaires de files d'attente gérés par IBM MQ Operator.

f) Supprimez un ou plusieurs gestionnaires de files d'attente.

Notez que, bien que ces gestionnaires de files d'attente continuent à s'exécuter, ils risquent de ne pas fonctionner comme prévu sans IBM MQ Operator.

g) Facultatif : Le cas échéant, répétez les étapes 2 à 6 pour chaque projet dans lequel vous souhaitez supprimer des gestionnaires de files d'attente.

h) Revenez à **Operators** > **Installed Operators**.

i) En regard de l'opérateur **IBM MQ**, cliquez sur le menu à trois points et sélectionnez **Uninstall Operator**.

- Option 2: Désinstallez le IBM MQ Operator à l'aide de l'interface de ligne de commande OpenShift

a) Connectez-vous à votre cluster Red Hat OpenShift à l'aide de `oc login`.

b) Si IBM MQ Operator est installé dans un espace de nom unique, procédez comme suit :

a. Vérifiez que vous êtes dans le projet contenant le IBM MQ Operator à désinstaller:

```
oc project project_name
```

b. Affichez les gestionnaires de files d'attente installés dans le projet :

```
oc get qmgr
```

c. Supprimez un ou plusieurs gestionnaires de files d'attente :

```
oc delete qmgr qmgr_name
```

Notez que, bien que ces gestionnaires de files d'attente continuent à s'exécuter, ils risquent de ne pas fonctionner comme prévu sans IBM MQ Operator.

d. Affichez les instances **`ClusterServiceVersion`** :

```
oc get csv
```

e. Supprimez le IBM MQ **`ClusterServiceVersion`**:

```
oc delete csv ibm_mq_csv_name
```

f. Affichez les abonnements :

```
oc get subscription
```

g. Supprimez tous les abonnements :

```
oc delete subscription ibm_mq_subscription_name
```

h. Si rien d'autre n'utilise les services communs, vous pouvez souhaiter désinstaller l'opérateur de services communs et supprimer le groupe d'opérateurs :

i) Désinstallez l'opérateur des services communs en suivant les instructions de la rubrique Désinstallation des services de base dans la documentation du produit IBM Cloud Pak foundational services .

ii) Affichez le groupe d'opérateurs :

```
oc get operatorgroup
```

iii) Supprimez le groupe d'opérateurs :

```
oc delete OperatorGroup operator_group_name
```

c) Si IBM MQ Operator est installé et disponible pour tous les espaces de nom du cluster, procédez comme suit :

a. Affichez tous les gestionnaires de files d'attente installés :

```
oc get qmgr -A
```

b. Supprimez un ou plusieurs gestionnaires de files d'attente :

```
oc delete qmgr qmgr_name -n namespace_name
```

Notez que, bien que ces gestionnaires de files d'attente continuent à s'exécuter, ils risquent de ne pas fonctionner comme prévu sans IBM MQ Operator.

c. Affichez les instances **ClusterServiceVersion** :

```
oc get csv -A
```

d. Supprimez le IBM MQ **ClusterServiceVersion** du cluster:

```
oc delete csv ibm_mq_csv_name -n openshift-operators
```

e. Affichez les abonnements :

```
oc get subscription -n openshift-operators
```

f. Supprimez les abonnements :

```
oc delete subscription ibm_mq_subscription_name -n openshift-operators
```

g. Facultatif: Si rien d'autre n'utilise les services communs, vous pouvez désinstaller l'opérateur des services communs. Pour ce faire, suivez les instructions de la rubrique Désinstallation des services de base dans la documentation du produit IBM Cloud Pak foundational services .

Préparation pour IBM MQ en créant votre propre image de conteneur

Développez un conteneur que vous avez généré vous-même. 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 le IBM MQ Operatorà la place. Voir «Comment utiliser IBM MQ dans des conteneurs», à la page 8

Pourquoi et quand exécuter cette tâche

Procédure

- «Remarques générales relatives à la génération de votre propre image de gestionnaire de files d'attente», à la page 58
- «Génération d'un exemple d'image de conteneur de gestionnaire de files d'attente IBM MQ», à la page 58
- «Exécution d'applications de liaison locale dans des conteneurs distincts», à la page 61
- Passez en revue l' IBM MQ exemple de charte Helm.

Remarques générales relatives à la génération de votre propre image de gestionnaire de files d'attente

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.

Pour résoudre ce problème, lorsque votre conteneur démarre, vous pouvez utiliser la commande **crtmqdir** pour remplir le répertoire `/var/mqm` si le conteneur s'exécute pour la première fois.

Sécurité de conteneur

Pour réduire les exigences de sécurité de l'environnement d'exécution, les exemples d'image de conteneur sont installés à l'aide de l'installation décompressable d'IBM MQ. Ainsi, aucun bit `setuid` n'est défini et le conteneur n'a pas besoin d'utiliser l'escalade de privilèges. Certains systèmes de conteneur définissent les ID utilisateur que vous pouvez utiliser et l'installation décompressable ne fait aucune supposition concernant les utilisateurs du système d'exploitation disponibles.

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

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 59
- «Génération d'un exemple d'image de gestionnaire de files d'attente IBM MQ configurée», à la page 59

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.
- Facultatif : Si vous prévoyez de configurer un accès sécurisé à l'aide de la contrainte de contexte de sécurité (SCC) Red Hat OpenShift Container Platform "restreinte" , utilisez l'une des images de non-installation IBM MQ .

  Des liens permettant de télécharger ces images sont disponibles dans la section Conteneurs des téléchargementsIBM MQ.

Résultats

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

Vous êtes maintenant prêt à générer un exemple d'image de gestionnaire de files d'attente IBM MQ configurée.

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

Cette tâche suppose que, lorsque vous avez créé votre exemple d'image de gestionnaire de files d'attente IBM MQ de base, vous avez utilisé le package "No-Install" IBM MQ. Sinon, 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 SCC "restricted", qui est utilisée par défaut, utilise des ID utilisateur aléatoires et empêche l'escalade des privilèges en passant à un autre utilisateur. Le programme d'installation traditionnel basé sur RPM IBM MQ repose sur un utilisateur et un groupe `mqm` et utilise également des bits `setuid` sur des programmes exécutables. Dans la version actuelle de IBM MQ, lorsque vous utilisez le package "No-Install" IBM MQ , il n'y a plus d'utilisateur `mqm` , ni de groupe `mqm` .

Procédure

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

```
DEFINE QLOCAL(EXAMPLE.QUEUE.1) REPLACE
```

   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
COPY config.mqsc /etc/mqm/
```

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

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

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

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

Remarque : Sous Red Hat Enterprise Linux (RHEL), vous utilisez la commande **docker** (RHEL V7) ou **podman** (RHEL V7 ou RHEL V8). Sous Linux, vous devez exécuter des commandes **docker** en indiquant **sudo** au début de la commande afin d'obtenir des privilèges supplémentaires.

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 :

```
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 **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}**.

Exécution d'applications de liaison locale dans des conteneurs distincts

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

Pourquoi et quand exécuter cette tâche

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`.

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 **dspmq** dans le deuxième conteneur pour afficher le statut des deux gestionnaires de files d'attente en émettant la commande suivante :

```
docker exec secondaryContainer dspmq
```

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 62 pour plus d'informations.

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 61 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 dltmqm QM1
AMQ7002: An error occurred manipulating a file.

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

MQ Adv. Création du groupe Native HA si vous créez vos propres conteneurs

Vous devez créer, configurer et démarrer trois gestionnaires de files d'attente pour créer le groupe Native HA.

Pourquoi et quand exécuter cette tâche

La méthode recommandée pour créer une solution Native HA consiste à utiliser l'opérateur IBM MQ (voir Native HA). Sinon, si vous créez vos propres conteneurs, vous pouvez suivre ces instructions.

Pour créer un groupe Native HA, vous créez trois gestionnaires de files d'attente sur trois noeuds dont le type de journal est défini sur `log replication`. Vous éditez ensuite le fichier `qm.ini` pour chaque gestionnaire de files d'attente afin d'ajouter les détails de connexion pour chacun des trois noeuds afin qu'ils puissent répliquer les données de journal les uns sur les autres.

Vous devez ensuite démarrer les trois gestionnaires de files d'attente afin qu'ils puissent vérifier que les trois instances peuvent communiquer entre elles et déterminer laquelle d'entre elles sera l'instance active et laquelle sera les répliques.

Remarque : Vous ne pouvez créer un groupe Native HA dans vos propres conteneurs de cette manière que si vous exécutez Kubernetes ou Red Hat OpenShift.

Procédure

1. Sur chacun des trois noeuds, créez un gestionnaire de files d'attente, en spécifiant un type de journal de réplique de journal et en fournissant un nom unique pour chaque instance de journal. Chaque gestionnaire de files d'attente porte le même nom:

   ```
   crtmqm -lr instance_name qmname
   ```

   Exemple :

   ```
   node 1> crtmqm -lr qm1_inst1 qm1

   node 2> crtmqm -lr qm1_inst2 qm1
   ```

```
node 3> crtmqm -lr qm1_inst3 qm1
```

2. Lorsque la création de chaque gestionnaire de files d'attente aboutit, une section supplémentaire nommée `NativeHALocalInstance` est ajoutée au fichier de configuration du gestionnaire de files d'attente, `qm.ini`. Un attribut `Name` est ajouté à la section spécifiant le nom d'instance fourni.

   Vous pouvez éventuellement ajouter les attributs suivants à la strophe `NativeHALocalInstance` dans le fichier `qm.ini` :

   KeyRepository
   Emplacement du référentiel de clés qui contient le certificat numérique à utiliser pour la protection du trafic de réplication des journaux. L'emplacement est donné au format de radical, c'est-à-dire qu'il inclut le chemin d'accès complet et le nom de fichier sans extension. Si l'attribut de section `KeyRepository` est omis, les données de réplication de journal sont échangées entre les instances en texte en clair.

   CertificateLabel
   Libellé de certificat identifiant le certificat numérique à utiliser pour la protection du trafic de réplication des journaux. Si `KeyRepository` est fourni mais que `CertificateLabel` est omis, la valeur par défaut `ibmwebspheremq`*`queue_manager`* est utilisée.

   CipherSpec
   Le MQ CipherSpec à utiliser pour protéger le trafic de réplication des journaux. Si cet attribut de section est fourni, `KeyRepository` doit également être fourni. Si `KeyRepository` est fourni mais que `CipherSpec` est omis, la valeur par défaut ANY est utilisée.

   LocalAddress
   Adresse de l'interface réseau locale qui accepte le trafic de réplication de journal. Si cet attribut de section est fourni, il identifie l'interface réseau locale et / ou le port en utilisant le format "[addr] [(port)]". L'adresse réseau peut être spécifiée sous la forme d'un nom d'hôte, IPv4 à notation décimale à point ou IPv6 au format hexadécimal. Si cet attribut est omis, le gestionnaire de files d'attente tente de se connecter à toutes les interfaces réseau et utilise le port spécifié dans `ReplicationAddress` dans la section `NativeHAInstances` correspondant au nom de l'instance locale.

   HeartbeatInterval
   L'intervalle des pulsations définit la fréquence en millisecondes à laquelle une instance active d'un gestionnaire de files d'attente Native HA envoie une pulsation réseau. Il est compris entre 500 (0,5 secondes) et 60000 (1 minute). Une valeur hors de cette plage empêche le démarrage du gestionnaire de files d'attente. Si cet attribut est omis, une valeur par défaut de 5000 (5 secondes) est utilisée. Chaque instance doit utiliser le même intervalle de pulsations.

   HeartbeatTimeout
   Le dépassement du délai d'attente du signal de présence définit le temps pendant lequel une réplique d'instance d'un gestionnaire de files d'attente Native HA attend avant de considérer que l'instance active ne répondra pas. Cette valeur doit être comprise entre 500 (0,5 secondes) et 120000 (2 minutes). La valeur du dépassement du délai d'attente du signal de présence doit être supérieure ou égale à celle de l'intervalle des pulsations.

   Une valeur non valide empêche le démarrage du gestionnaire de files d'attente. Si cet attribut est omis, une réplique attend 2 x `HeartbeatInterval` avant de lancer le processus pour sélectionner une nouvelle instance active. Chaque instance doit utiliser la même valeur de dépassement du délai d'attente du signal de présence.

   RetryInterval
   L'intervalle entre les nouvelles tentatives définit la fréquence en millisecondes à laquelle un gestionnaire de files d'attente Native HA doit retenter un lien de réplication défectueux. Cet intervalle doit être compris entre 500 (0,5 secondes) et 120000 (2 minutes). Si cet attribut est omis, une réplique attend 2 x `HeartbeatInterval` avant de réessayer un lien de réplication ayant échoué.

3. Editez le fichier `qm.ini` pour chaque gestionnaire de files d'attente et ajoutez les détails de connexion. Vous ajoutez trois sections `NativeHAInstance` , une pour chaque instance de

gestionnaire de files d'attente dans le groupe Native HA (y compris l'instance locale). Ajoutez les attributs suivants:

Nom
Indiquez le nom d'instance que vous avez utilisé lors de la création de l'instance de gestionnaire de files d'attente.

ReplicationAddress
Indiquez le nom d'hôte, IPv4 décimale à point ou IPv6 adresse au format hexadécimal de l'instance. Vous pouvez spécifier l'adresse en tant que nom d'hôte, IPv4 en notation décimale à point ou IPv6 en format hexadécimal. L'adresse de réplication doit pouvoir être résolue et routable à partir de chaque instance du groupe. Le numéro de port à utiliser pour la réplication de journal doit être indiqué entre crochets, par exemple:

```
ReplicationAddress=host1.example.com(4444)
```

Remarque : Les sections `NativeHAInstance` sont identiques sur chaque instance et peuvent être fournies à l'aide de la configuration automatique (**`crtmqm -ii`**).

4. Démarrez chacune des trois instances:

```
strmqm QMgrName
```

Lorsque les instances sont démarrées, elles communiquent pour vérifier que les trois instances sont en cours d'exécution, puis décident laquelle des trois instances est l'instance active, tandis que les deux autres instances continuent de s'exécuter en tant que répliques.

Exemple

L'exemple suivant illustre la section d'un fichier `qm.ini` spécifiant les détails Native HA requis pour l'une des trois instances:

```
NativeHALocalInstance:
  LocalName=node-1

NativeHAInstance:
  Name=node-1
  ReplicationAddress=host1.example.com(4444)
NativeHAInstance:
  Name=node-2
  ReplicationAddress=host2.example.com(4444)
NativeHAInstance:
  Name=node-3
  ReplicationAddress=host3.example.com(4444)
```

Déploiement et configuration de gestionnaires de files d'attente dans des conteneurs

Vous effectuez une série de tâches pour déployer et configurer des gestionnaires de files d'attente IBM MQ .

Pourquoi et quand exécuter cette tâche

Pour vous initier au déploiement et à la configuration des gestionnaires de files d'attente, voir les rubriques suivantes.

Procédure

- «Déploiement et configuration de gestionnaires de files d'attente à l'aide de IBM MQ Operator», à la page 66
- «Déploiement et configuration de gestionnaires de files d'attente à l'aide de Helm», à la page 109

OpenShift Déploiement et configuration de gestionnaires de files d'attente à l'aide de IBM MQ Operator

Exemples de configuration ; configuration de la haute disponibilité ; connexion depuis l'extérieur d'un cluster OpenShift ; intégration avec le tableau de bord CP4i ; intégration avec le traçage Instana ; génération d'une image avec des fichiers MQSC et INI personnalisés ; ajout d'annotations et de libellés personnalisés.

Pourquoi et quand exécuter cette tâche

Procédure

- «Exemples de configuration d'un gestionnaire de files d'attente», à la page 69.
- «Configuration de la haute disponibilité pour les gestionnaires de files d'attente à l'aide de IBM MQ Operator», à la page 78.
- «Configuration d'une route pour la connexion à un gestionnaire de files d'attente depuis l'extérieur d'un cluster Red Hat OpenShift», à la page 88.
- «Intégration d' IBM MQ à la fonction de trace IBM Instana», à la page 90.
- «Génération d'une image avec des fichiers MQSC et INI personnalisés, à l'aide de l'interface de ligne de commande Red Hat OpenShift», à la page 97.
- «Ajout d'annotations et d'étiquettes personnalisées aux ressources du gestionnaire de files d'attente», à la page 99.
- «Désactivation des vérifications des webhooks d'exécution», à la page 100.
- «Désactivation des mises à jour des valeurs par défaut de la spécification du gestionnaire de files d'attente», à la page 100.

OpenShift CP4I Déploiement d'un gestionnaire de files d'attente simple à l'aide de IBM MQ Operator

Cet exemple déploie un gestionnaire de files d'attente de démarrage rapide qui utilise un stockage éphémère (non persistant) et désactive la sécurité IBM MQ . Les messages ne sont pas conservés lors des redémarrages du gestionnaire de files d'attente. Vous pouvez ajuster la configuration afin de changer de nombreux paramètres du gestionnaire de files d'attente.

Pourquoi et quand exécuter cette tâche

Cette tâche offre 3 options pour le déploiement d'un gestionnaire de files d'attente sur OpenShift:

1. Déployez un gestionnaire de files d'attente avec la console OpenShift.
2. Déployez un gestionnaire de files d'attente avec l'interface de ligne de commande OpenShift.
3. Déployez un gestionnaire de files d'attente avec IBM Cloud Pak for Integration Platform UI.

Procédure

- **Option 1: Déploiement d'un gestionnaire de files d'attente à l'aide de la console OpenShift .**

  a) Déployez un gestionnaire de files d'attente.

  a. Connectez-vous à la console OpenShift avec vos données d'identification d'administrateur de cluster Red Hat OpenShift Container Platform .

  b. Remplacez **Project** par l'espace de nom dans lequel vous avez installé le IBM MQ Operator. Sélectionnez l'espace de nom dans la liste déroulante **Projet** .

  c. Dans le panneau de navigation, cliquez sur **Operators** > **Installed Operators**.

  d. Dans la liste du panneau Installed Operators, recherchez et cliquez sur **IBM MQ**.

e. cliquez sur l'onglet **Queue Manager**.

f. Cliquez sur le bouton **Create QueueManager**. Le panneau de création d'instance s'affiche et propose deux méthodes de configuration de la ressource: la **vue de fiche** et la **vue YAML**. La **vue de fiche** est sélectionnée par défaut.

b) Configurez le gestionnaire de files d'attente.

Etape 2 Option 1: Configuration dans la **vue de fiche**.

La **vue de formulaire** ouvre un formulaire que vous pouvez utiliser pour afficher ou modifier la configuration des ressources.

a. En regard de **Licence**, cliquez sur la flèche pour développer la section d'acceptation de la licence.

b. Définissez **License accept** sur **true** si vous acceptez le contrat de licence.

c. Cliquez sur la flèche pour ouvrir la liste déroulante et sélectionnez une licence. 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 144. Vous devez accepter la licence pour pouvoir déployer un gestionnaire de files d'attente.

d. 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`.

Etape 2 Option 2: Configuration dans la **vue YAML**.

La **vue YAML** ouvre un éditeur contenant un exemple de fichier YAML pour un `QueueManager`. Mettez à jour les valeurs dans le fichier en suivant les étapes ci-dessous.

a. Remplacez `metadata.namespace` par le nom de votre projet (espace de nom).

b. Remplacez la valeur de `spec.license.license` par la chaîne de licence qui correspond à vos besoins. Pour plus d'informations sur la licence, voir «Référence relative à l'octroi de licence pour mq.ibm.com/v1beta1», à la page 144 .

c. Remplacez `spec.license.accept` par `true` si vous acceptez le contrat de licence.

d. 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`.

c) Vérifiez la création du gestionnaire de files d'attente.

Vous pouvez vérifier que vous avez créé un gestionnaire de files d'attente en procédant comme suit:

a. Vérifiez que vous êtes dans l'espace de nom dans lequel vous avez créé votre IBM MQ Operator .

b. Dans l'écran **Accueil** , cliquez sur **Opérateurs** > **Opérateurs installés**, puis sélectionnez le IBM MQ Operator installé pour lequel vous avez créé le gestionnaire de files d'attente.

c. cliquez sur l'onglet **Queue Manager**. La création est terminée lorsque le statut de la ressource `QueueManager` est `Running`.

- **Option 2: Déploiement d'un gestionnaire de files d'attente à l'aide de l'interface de ligne de commande OpenShift .**

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: 9.4.0.0-r1
  license:
    accept: false
```

```
    license: L-BMSF-5YDSLR
    use: NonProduction
  web:
    enabled: true
  queueManager:
    name: "QUICKSTART"
    storage:
      queueManager:
        type: ephemeral
```

Important : Si vous acceptez le contrat de licence, remplacez `accept: false` par `accept: true`. Voir «Référence relative à l'octroi de licence pour mq.ibm.com/v1beta1», à la page 144 pour des détails sur la licence.

Cet exemple inclut également un serveur Web déployé avec le gestionnaire de files d'attente, avec la console Web activée avec la connexion unique dans IBM Cloud Pak for Integration. Pour que la connexion unique fonctionne, vous devez d'abord installer d'autres composants IBM Cloud Pak for Integration . Voir «Installation du IBM MQ Operator pour une utilisation avec CP4I», à la page 41.

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: 9.4.0.0-r1
  license:
    accept: false
    license: L-EHXT-MQCRN9
  web:
    enabled: true
  queueManager:
    name: "QUICKSTART"
    storage:
      queueManager:
        type: ephemeral
```

Important: si vous acceptez le contrat de licence MQ , remplacez `accept: false` par `accept: true`. Voir «Référence relative à l'octroi de licence pour mq.ibm.com/v1beta1», à la page 144 pour des détails sur la licence.

b) Créez l'objet `QueueManager` .

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

c) Vérifiez la création du gestionnaire de files d'attente.

Vérifiez que vous avez créé un gestionnaire de files d'attente en procédant comme suit:

a. Vérifiez le déploiement :

```
oc describe queuemanager Queue_Manager_Resource_Name
```

b. Vérifiez le statut :

```
oc describe queuemanager quickstart
```

- **Option 3: Déploiement d'un gestionnaire de files d'attente avec IBM Cloud Pak for Integration Platform UI.**

  Suivez les instructions de la rubrique Déploiement d'une instance à l'aide de l'interface utilisateur de la plateforme.

Tâches associées

«Configuration d'une route pour la connexion à un gestionnaire de files d'attente depuis l'extérieur d'un cluster Red Hat OpenShift», à la page 88
Vous avez besoin d'une route Red Hat OpenShift pour connecter une application à un gestionnaire de files d'attente IBM MQ depuis l'extérieur d'un cluster Red Hat OpenShift . Vous devez activer TLS sur votre

gestionnaire de files d'attente et votre application client IBM MQ , car SNI est disponible uniquement dans le protocole TLS lorsqu'un protocole TLS 1.2 ou supérieur est utilisé. Red Hat OpenShift Container Platform Router utilise l'indication de nom de serveur pour acheminer les demandes vers le gestionnaire de files d'attente IBM MQ.

«Connexion à IBM MQ Console déployée dans un cluster Red Hat OpenShift», à la page 134
Comment se connecter au IBM MQ Console d'un gestionnaire de files d'attente qui a été déployé sur un cluster Red Hat OpenShift Container Platform .

«Exemples de configuration d'un gestionnaire de files d'attente», à la page 69
Un gestionnaire de files d'attente peut être configuré en ajustant le contenu de la ressource personnalisée QueueManager.

OpenShift CP4I Exemples de configuration d'un gestionnaire de files d'attente

Un gestionnaire de files d'attente peut être configuré en ajustant le contenu de la ressource personnalisée QueueManager.

Pourquoi et quand exécuter cette tâche

Utilisez les exemples suivants pour vous aider à configurer un gestionnaire de files d'attente à l'aide du fichier YAML QueueManager.

Procédure

- «Exemple : fourniture de fichiers MQSC et INI», à la page 69
- «Exemple: configuration d'un gestionnaire de files d'attente avec l'authentification TLS mutuelle», à la page 73

OpenShift CP4I *Exemple : fourniture de fichiers MQSC et INI*

Cet exemple crée une mappe de configuration Kubernetes contenant deux fichiers MQSC et un fichier INI. Un gestionnaire de files d'attente qui traite ces fichiers MQSC et INI est alors déployé.

Pourquoi et quand exécuter cette tâche

Les fichiers MQSC et INI peuvent être fournis lorsqu'un gestionnaire de files d'attente est déployé. Les données MQSC et INI sont définies dans un ou plusieurs Kubernetes ConfigMaps et Secrets. Ces éléments doivent être créés dans l'espace de nom (projet) où vous déployez le gestionnaire de files d'attente.

Remarque : Un secret Kubernetes doit être utilisé si le fichier MQSC ou INI contient des données sensibles.

Exemple

L'exemple suivant crée une mappe de configuration Kubernetes contenant deux fichiers MQSC et un fichier INI. Un gestionnaire de files d'attente qui traite ces fichiers MQSC et INI est alors déployé.

Exemple de mappe de configuration (ConfigMap) - appliquez le code YAML suivant dans votre cluster :

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: mqsc-ini-example
data:
  example1.mqsc: |
    DEFINE QLOCAL('DEV.QUEUE.1') REPLACE
    DEFINE QLOCAL('DEV.QUEUE.2') REPLACE
  example2.mqsc: |
    DEFINE QLOCAL('DEV.DEAD.LETTER.QUEUE') REPLACE
  example.ini: |
```

```
    Channels:
       MQIBindType=FASTPATH
```

Exemple QueueManager -déployez votre gestionnaire de files d'attente avec la configuration suivante, à l'aide de la ligne de commande ou de la console Web Red Hat OpenShift Container Platform :

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: mqsc-ini-qm
spec:
  version: 9.4.0.0-r1
  license:
    accept: false
    license: L-EHXT-MQCRN9
    use: Production
  web:
    enabled: true
  queueManager:
    name: "MQSCINI"
    mqsc:
    - configMap:
        name: mqsc-ini-example
        items:
        - example1.mqsc
        - example2.mqsc
    ini:
    - configMap:
        name: mqsc-ini-example
        items:
        - example.ini
    storage:
      queueManager:
        type: ephemeral
```

Important : Si vous acceptez le contrat de licence IBM MQ Advanced, modifiez `accept: false` par `accept: true`. Pour plus d'informations sur la licence, voir Référence relative à l'octroi de licence pour mq.ibm.com/v1beta1.

Informations supplémentaires :

- Un gestionnaire de files d'attente peut être configuré pour utiliser un seul objet Kubernetes ConfigMap ou secret (comme illustré dans cet exemple) ou plusieurs objets ConfigMaps et secrets.
- Vous pouvez choisir d'utiliser toutes les données MQSC et INI à partir d'un élément ConfigMap ou Secret Kubernetes (comme indiqué dans cet exemple) ou de configurer chaque gestionnaire de files d'attente pour qu'il n'utilise qu'un sous-ensemble des fichiers disponibles.
- Les fichiers MQSC et INI sont traités par ordre alphabétique en fonction de leur clé. Ainsi, `example1.mqsc` sera toujours traité avant `example2.mqsc`, quel que soit l'ordre dans lequel ils apparaissent dans la configuration du gestionnaire de files d'attente.
- Si plusieurs fichiers MQSC ou INI possèdent la même clé, entre plusieurs éléments ConfigMap ou Secret Kubernetes, cet ensemble de fichiers est traité en fonction de l'ordre dans lequel les fichiers sont définis dans la configuration du gestionnaire de files d'attente.
- Lorsqu'un pod de gestionnaire de files d'attente est en cours d'exécution, les modifications apportées à Kubernetes ConfigMap ne sont pas prises en compte car le IBM MQ Operator n'a pas connaissance de la modification. Si vous apportez des modifications à ConfigMap, par exemple aux commandes MQSC ou aux fichiers INI, vous devez redémarrer manuellement les gestionnaires de files d'attente pour prendre en compte ces modifications. Pour les gestionnaires de files d'attente à instance unique, supprimez le pod pour déclencher le redémarrage requis. Pour les déploiements Native HA, redémarrez d'abord les pods de secours en les supprimant. Lorsqu'ils sont à nouveau dans un état en cours d'exécution, supprimez le pod actif pour le redémarrer. Cet ordre de redémarrages garantit un temps d'indisponibilité minimal pour le gestionnaire de files d'attente.

OpenShift CP4I *Création d'une infrastructure PKI autosignée à l'aide de OpenSSL*

IBM MQ vous permet d'utiliser le protocole TLS mutuel pour l'authentification, où les deux extrémités d'une connexion fournissent un certificat et les détails du certificat sont utilisés pour établir une identité avec le gestionnaire de files d'attente. Cette rubrique explique comment créer un exemple d'infrastructure PKI (Public Key Infrastructure) à l'aide de l'outil de ligne de commande OpenSSL , en créant deux certificats qui peuvent être utilisés dans d'autres exemples.

Avant de commencer

Vérifiez que l'outil de ligne de commande OpenSSL est installé.

Installez IBM MQ clientet ajoutez `samp/bin` et `bin` à votre *CHEMIN*. Vous avez besoin de la commande **`runmqicred`**, qui peut être installée dans le cadre de IBM MQ client comme suit:

- Windows Linux Pour Windows et Linux: installez le client redistribuable IBM MQ pour votre système d'exploitation à partir de https://ibm.biz/mq94redistclients
- macOS Pour Mac: téléchargez et configurez IBM MQ MacOS Toolkit: https://developer.ibm.com/tutorials/mq-macos-dev/

Pourquoi et quand exécuter cette tâche

Important : Les exemples décrits ici ne sont pas adaptés à un environnement de production, et sont uniquement destinés à servir d'exemples pour une mise en oeuvre rapide. La gestion des certificats est un sujet complexe pour les utilisateurs avancés. Pour la production, vous devez prendre en compte des éléments tels que la rotation, la révocation, la longueur de clé, la reprise après incident et bien plus encore.

Ces étapes ont été testées avec OpenSSL 3.1.4.

Procédure

1. Créez une clé privée à utiliser pour votre autorité de certification interne

   ```
   openssl genpkey -algorithm rsa -pkeyopt rsa_keygen_bits:4096 -out ca.key
   ```

   Une clé privée pour l'autorité de certification interne est créée dans un fichier appelé *ca.key*. Ce fichier doit être protégé et secret-il sera utilisé pour signer des certificats pour votre autorité de certification interne.
2. Emettez un certificat autosigné pour votre autorité de certification interne

   ```
   openssl req -x509 -new -nodes -key ca.key -sha512 -days 30 -subj "/CN=example-selfsigned-ca"
   -out ca.crt
   ```

   `-days` indique le nombre de jours pendant lesquels le certificat de l'autorité de certification racine sera valide.

   Un certificat est créé dans un fichier appelé *ca.crt*. Ce certificat contient les informations publiques sur l'autorité de certification interne et est librement partageable.
3. Création d'une clé privée et d'un certificat pour un gestionnaire de files d'attente
   a) Création d'une clé privée et d'une demande de signature de certificat pour un gestionnaire de files d'attente

      ```
      openssl req -new -nodes -out example-qm.csr -newkey rsa:4096 -keyout example-qm.key -subj
      '/CN=example-qm'
      ```

      Une clé privée est créée dans un fichier appelé *example-qm.key*, et une demande de signature de certificat est créée dans un fichier appelé *example-qm.csr*
   b) Signez la clé du gestionnaire de files d'attente avec votre autorité de certification interne

```
openssl x509 -req -in example-qm.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out
example-qm.crt -days 7 -sha512
```

`-days` indique le nombre de jours pendant lesquels le certificat sera valide.

Un certificat signé est créé dans un fichier appelé *example-qm.crt*

c) Création d'un secret Kubernetes avec la clé et le certificat du gestionnaire de files d'attente

```
oc create secret generic example-qm-tls --type="kubernetes.io/tls" --from-
file=tls.key=example-qm.key --from-file=tls.crt=example-qm.crt --from-file=ca.crt
```

Un secret Kubernetes appelé *example-qm-tls* est créé. Ce secret contient la clé privée du gestionnaire de files d'attente, le certificat public et le certificat de l'autorité de certification.

4. Créer une clé privée et un certificat pour une application

a) Créer une clé privée et une demande de signature de certificat pour une application

```
openssl req -new -nodes -out example-app1.csr -newkey rsa:4096 -keyout example-app1.key
-subj '/CN=example-app1'
```

Une clé privée est créée dans un fichier appelé *example-app1.key*et une demande de signature de certificat est créée dans un fichier appelé *example-app1.csr*

b) Signez la clé du gestionnaire de files d'attente avec votre autorité de certification interne

```
openssl x509 -req -in example-app1.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out
example-app1.crt -days 7 -sha512
```

`-days` indique le nombre de jours pendant lesquels le certificat sera valide.

Un certificat signé est créé dans un fichier appelé *example-app1.crt*

c) Création d'un magasin de clés PKCS#12 avec la clé et le certificat de l'application

IBM MQ utilise une base de données de clés et non des fichiers de clés individuels. Le gestionnaire de files d'attente conteneurisé crée la base de données de clés pour le gestionnaire de files d'attente à partir d'un secret, mais pour les applications client, vous devez créer manuellement la base de données de clés.

```
openssl pkcs12 -export -in "example-app1.crt" -name "example-app1" -certfile "ca.crt"
-inkey "example-app1.key" -out "example-app1.p12" -passout pass:PASSWORD
```

Où *PASSWORD* est un mot de passe de votre choix.

Un magasin de clés est créé dans un fichier appelé *example-app1.p12*. La clé et le certificat de l'application sont stockés à l'intérieur, avec un "label" ou un "nom usuel" de "example-app1", ainsi que le certificat de l'autorité de certification.

d) Si vous utilisez un Apple Mac arm64 , vous devez configurer un fichier supplémentaire combinant les certificats de l'application et de l'autorité de certification.
Exemple :

```
cat example-app1.crt ca.crt > example-app1-chain.crt
```

Tâches associées

«Exemple: configuration d'un gestionnaire de files d'attente avec l'authentification TLS mutuelle», à la page 73
Cet exemple déploie un gestionnaire de files d'attente dans OpenShift Container Platform à l'aide de IBM MQ Operator. Le protocole TLS mutuel est utilisé pour l'authentification afin de mapper un certificat TLS à une identité dans le gestionnaire de files d'attente.

«Exemple: Configuration de Native HA à l'aide de IBM MQ Operator», à la page 79
Cet exemple déploie un gestionnaire de files d'attente à l'aide de la fonction de haute disponibilité native dans OpenShift Container Platform à l'aide de IBM MQ Operator. Le protocole TLS mutuel est utilisé pour l'authentification afin de mapper un certificat TLS à une identité dans le gestionnaire de files d'attente.

«Configuration d'un gestionnaire de files d'attente multi-instance à l'aide de IBM MQ Operator», à la page 85
Cet exemple déploie un gestionnaire de files d'attente multi-instance à l'aide de OpenShift Container Platform à l'aide de IBM MQ Operator. Le protocole TLS mutuel est utilisé pour l'authentification afin de mapper un certificat TLS à une identité dans le gestionnaire de files d'attente.

Exemple: configuration d'un gestionnaire de files d'attente avec l'authentification TLS mutuelle

Cet exemple déploie un gestionnaire de files d'attente dans OpenShift Container Platform à l'aide de IBM MQ Operator. Le protocole TLS mutuel est utilisé pour l'authentification afin de mapper un certificat TLS à une identité dans le gestionnaire de files d'attente.

Avant de commencer

Pour mettre en oeuvre cet exemple, vous devez d'abord avoir rempli les conditions suivantes :

- Créez un projet / espace de nom OpenShift Container Platform (OCP) pour cet exemple.
- Sur la ligne de commande, connectez-vous au cluster OCP et passez à l'espace de nom ci-dessus.
- Vérifiez que IBM MQ Operator est installé et disponible dans l'espace de nom ci-dessus.

Pourquoi et quand exécuter cette tâche

Cet exemple fournit une ressource personnalisée YAML définissant un gestionnaire de files d'attente à déployer dans OpenShift Container Platform. Il détaille également les étapes supplémentaires requises pour déployer le gestionnaire de files d'attente avec TLS activé.

Procédure

1. Créez une paire de certificats comme décrit dans «Création d'une infrastructure PKI autosignée à l'aide de OpenSSL», à la page 71.
2. Créer une mappe de configuration contenant des commandes MQSC et un fichier INI

   Créez un objet Kubernetes ConfigMap contenant les commandes MQSC pour créer une file d'attente et un canal SVRCONN, et pour ajouter un enregistrement d'authentification de canal qui autorise l'accès au canal.

   Vérifiez que vous vous trouvez dans l'espace de nom que vous avez créé précédemment (voir Avant de commencer), puis entrez le fichier YAML suivant dans la console Web OCP ou à l'aide de la ligne de commande.

   ```
   apiVersion: v1
   kind: ConfigMap
   metadata:
     name: example-tls-configmap
   data:
     example-tls.mqsc: |
       DEFINE CHANNEL('MTLS.SVRCONN') CHLTYPE(SVRCONN) SSLCAUTH(REQUIRED)
   SSLCIPH('ANY_TLS13_OR_HIGHER') REPLACE
       SET CHLAUTH('MTLS.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=*') USERSRC(NOACCESS)
   ACTION(REPLACE)
       SET CHLAUTH('MTLS.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=example-app1') USERSRC(MAP)
   MCAUSER('app1') ACTION(REPLACE)
       SET AUTHREC PRINCIPAL('app1') OBJTYPE(QMGR) AUTHADD(CONNECT,INQ)
       DEFINE QLOCAL('EXAMPLE.QUEUE') REPLACE
       SET AUTHREC PROFILE('EXAMPLE.QUEUE') PRINCIPAL('app1') OBJTYPE(QUEUE)
   AUTHADD(BROWSE,PUT,GET,INQ)
     example-tls.ini: |
       Service:
           Name=AuthorizationService
           EntryPoints=14
           SecurityPolicy=UserExternal
   ```

   Le MQSC définit un canal appelé *MTLS.SVRCONN* et une file d'attente appelée *EXAMPLE.QUEUE*. Le canal est configuré pour autoriser l'accès uniquement aux clients qui présentent un certificat avec un "nom usuel" de *example-app1*. Il s'agit du nom usuel utilisé dans l'un des certificats créés à l'étape

«1», à la page 73. Les connexions sur ce canal avec ce nom usuel sont mappées à un ID utilisateur *app1*, qui est autorisé à se connecter au gestionnaire de files d'attente et à accéder à l'exemple de file d'attente. Le fichier INI active une règle de sécurité qui signifie que l'ID utilisateur *app1* n'a pas besoin d'exister dans un registre d'utilisateurs externe-il existe uniquement sous forme de nom dans cette configuration.

3. Déployez le gestionnaire de files d'attente

   Créez un gestionnaire de files d'attente à l'aide de la ressource personnalisée YAML suivante. Vérifiez que vous vous trouvez dans l'espace de nom que vous avez créé avant de commencer cette tâche, puis entrez le fichier YAML suivant dans la console Web OCP ou à l'aide de la ligne de commande. Vérifiez que la licence est correcte et acceptez-la en remplaçant `false` par `true`.

   ```
   apiVersion: mq.ibm.com/v1beta1
   kind: QueueManager
   metadata:
     name: exampleqm
   spec:
     license:
       accept: false
       license: L-EHXT-MQCRN9
       use: Production
     queueManager:
       name: EXAMPLEQM
       mqsc:
       - configMap:
           name: example-tls-configmap
           items:
           - example-tls.mqsc
       ini:
       - configMap:
           name: example-tls-configmap
           items:
           - example-tls.ini
       storage:
         queueManager:
           type: ephemeral
     version: 9.4.0.0-r1
     pki:
       keys:
         - name: default
           secret:
             secretName: example-qm-tls
             items:
               - tls.key
               - tls.crt
               - ca.crt
   ```

   Notez que le secret *example-qm-tls* a été créé à l'étape «1», à la page 73et que l' *exemple-tls-configmap* ConfigMap a été créé à l'étape «2», à la page 73

4. Confirmez que le gestionnaire de files d'attente est en cours d'exécution

   Le gestionnaire de files d'attente est en cours de déploiement. Confirmez qu'il se trouve dans l'état `Running` avant de continuer. Exemple :

   ```
   oc get qmgr exampleqm
   ```

5. Testez la connexion au gestionnaire de files d'attente

   Pour vérifier que le gestionnaire de files d'attente est configuré pour la communication TLS mutuelle, suivez les étapes de la rubrique «Test d'une connexion TLS mutuelle à un gestionnaire de files d'attente à partir de votre ordinateur portable», à la page 75.

Résultats

Félicitations, vous avez déployé avec succès un gestionnaire de files d'attente avec TLS activé et qui utilise les détails fournis dans le certificat TLS pour s'authentifier auprès du gestionnaire de files d'attente et fournir une identité.

OpenShift CP4I Linux *Test d'une connexion TLS mutuelle à un gestionnaire de files d'attente à partir de votre ordinateur portable*

Une fois que vous avez créé un gestionnaire de files d'attente à l'aide de IBM MQ Operator, vous pouvez vérifier qu'il fonctionne en vous y connectant, en insérant et en obtenant un message. Cette tâche vous explique comment vous connecter à l'aide des exemples de programme IBM MQ en les exécutant sur une machine en dehors du cluster Kubernetes , telle que votre ordinateur portable.

Avant de commencer

Pour mettre en oeuvre cet exemple, vous devez d'abord avoir rempli les conditions suivantes :

- Installez IBM MQ client. Vous avez besoin des commandes **amqsputc** et **amqsgetc** , qui peuvent être installées dans le cadre de IBM MQ client comme suit:
  - Windows Linux Pour Windows et Linux: installez le client redistribuable IBM MQ pour votre système d'exploitation à partir de https://ibm.biz/mq94redistclients
  - macOS Pour Mac: téléchargez et configurez IBM MQ MacOS Toolkit: https://developer.ibm.com/tutorials/mq-macos-dev/
- Vérifiez que vous disposez des fichiers de clés et de certificats nécessaires téléchargés dans un répertoire de votre machine et que vous connaissez le mot de passe du magasin de clés. Par exemple, ces fichiers sont créés dans «Création d'une infrastructure PKI autosignée à l'aide de OpenSSL», à la page 71:
  - `example-app1.p12`
  - `example-app1-chain.crt` (uniquement si vous utilisez une arm64 Apple Mac)
- Déployez un gestionnaire de files d'attente configuré avec TLS sur le cluster OCP, par exemple en suivant les étapes de la rubrique «Exemple: configuration d'un gestionnaire de files d'attente avec l'authentification TLS mutuelle», à la page 73

Pourquoi et quand exécuter cette tâche

Cet exemple utilise les exemples de programmes IBM MQ exécutés sur une machine hors du cluster Kubernetes , telle que votre ordinateur portable, pour se connecter à un QueueManager configuré avec TLS et pour insérer et extraire des messages.

Procédure

1. Confirmez que le gestionnaire de files d'attente est en cours d'exécution

   Le gestionnaire de files d'attente est en cours de déploiement. Confirmez qu'il se trouve dans l'état `Running` avant de continuer. Exemple :

   ```
   oc get qmgr exampleqm
   ```

2. Localisez le nom d'hôte du gestionnaire de files d'attente

   Utilisez la commande suivante pour rechercher le nom d'hôte qualifié complet du gestionnaire de files d'attente en dehors du cluster OCP, à l'aide de la route qui est créée automatiquement: `exampleqm-ibm-mq-qm`:

   ```
   oc get route exampleqm-ibm-mq-qm --template="{{.spec.host}}"
   ```

3. Création d'une table de définition de canal du client IBM MQ (CCDT)

   Créez un fichier appelé `ccdt.json` avec le contenu suivant:

   ```
   {
       "channel":
       [
           {
               "name": "MTLS.SVRCONN",
   ```

```
            "clientConnection":
            {
                "connection":
                [
                    {
                        "host": "hostname from previous step",
                        "port": 443
                    }
                ],
                "queueManager": "EXAMPLEQM"
            },
            "transmissionSecurity":
            {
              "cipherSpecification": "ANY_TLS13",
              "certificateLabel": "example-app1"
            },
            "type": "clientConnection"
        }
   ]
}
```

La connexion utilise le port 443, car il s'agit du port sur lequel le routeur Red Hat OpenShift Container Platform écoute. Le trafic sera réacheminé vers le gestionnaire de files d'attente sur le port 1414.

Si vous avez utilisé un autre nom de canal, vous devrez également l'ajuster. Les exemples TLS mutuel utilisent un canal nommé *MTLS.SVRCONN*

Pour plus de détails, voir Configuration d'une table de définition de canal du client au format JSON

4. Créez un fichier INI client pour configurer les détails de connexion

   Créez un fichier appelé `mqclient.ini` dans le répertoire de travail. Ce fichier sera lu par **amqsputc** et **amqsgetc**.

   ```
   Channels:
     ChannelDefinitionDirectory=.
     ChannelDefinitionFile=ccdt.json
   SSL:
     OutboundSNI=HOSTNAME
     SSLKeyRepository=example-app1.p12
     SSLKeyRepositoryPassword=password you used when creating the p12 file
   ```

   Veillez à mettre à jour le mot de passe *SSLKeyRepository* avec le mot de passe que vous avez choisi lors de la création du fichier PKCS#12 . Il existe d'autres façons de définir le mot de passe du fichier de clés, notamment à l'aide d'un mot de passe chiffré. Pour plus d'informations, voir Définition du mot de passe du référentiel de clés pour un IBM MQ MQI client sur AIX, Linux, and Windows

   Notez que le Red Hat OpenShift Container Platform Router utilise SNI pour acheminer les demandes vers le gestionnaire de files d'attente IBM MQ. L'attribut *OutboundSNI= HOSTNAME* garantit que le client IBM MQ inclut les informations nécessaires pour que le routeur utilise la route par défaut configurée par le IBM MQ Operator. Pour plus d'informations, voir «Configuration d'une route pour la connexion à un gestionnaire de files d'attente depuis l'extérieur d'un cluster Red Hat OpenShift», à la page 88.

5. Si vous utilisez un Apple Mac arm64 , vous devez configurer une variable d'environnement supplémentaire.

   ```
   export MQSSLTRUSTSTORE=example-app1-chain.crt
   ```

   Ce fichier contient la chaîne de certificats complète, y compris les certificats de l'application et de l'autorité de certification.

6. Insérez des messages dans la file d'attente

   Exécutez ensuite la commande suivante :

   ```
   /opt/mqm/samp/bin/amqsputc EXAMPLE.QUEUE EXAMPLEQM
   ```

   Si la connexion au gestionnaire de files d'attente aboutit, la réponse suivante apparaît :

   ```
   target queue is EXAMPLE.QUEUE
   ```

Placez plusieurs messages dans la file d'attente en entrant un texte, puis en appuyant sur **Entrée** à chaque fois.

Pour terminer, appuyez deux fois sur **Entrée**.

7. Extrayez les messages de la file d'attente

   Exécutez ensuite la commande suivante :

   ```
   /opt/mqm/samp/bin/amqsgetc EXAMPLE.QUEUE EXAMPLEQM
   ```

   Les messages que vous avez ajoutés à l'étape précédente ont été utilisés et sont renvoyés. Après quelques secondes, la commande prend fin.

Résultats

Félicitations, vous avez testé avec succès la connexion à un gestionnaire de files d'attente avec TLS activé et vous avez montré que vous pouvez placer et extraire des messages dans le gestionnaire de files d'attente à partir d'un client de manière sécurisée.

OpenShift CP4I *Exemple : Personnalisation des annotations de service de licence*

Le IBM MQ Operator ajoute automatiquement des annotations IBM License Service aux ressources déployées. Ils sont surveillés par IBM License Service, et des rapports correspondant à l'autorisation requise sont générés.

Pourquoi et quand exécuter cette tâche

Les annotations ajoutées par IBM MQ Operator sont celles attendues dans des situations standard et sont basées sur les valeurs de licence sélectionnées lors du déploiement d'un gestionnaire de files d'attente.

Exemple

Si **License** est défini sur `L-RJON-BZFQU2 (IBM Cloud Pak for Integration 2021.2.1)`et que **Use** est défini sur `Nonproduction`, les annotations suivantes sont appliquées :

- cloudpakId: c8b82d189e7545f0892db9ef2731b90d
- cloudpakName: IBM Cloud Pak for Integration
- productChargedContainers : qmgr
- productCloudpakRatio : '4:1'
- productID: 21dfe9a0f00f444f888756d835334909
- productName : IBM MQ Advanced for Non-Production
- ProductMetric : VIRTUAL_PROCESSOR_CORE
- productVersion: 9.2.3.0

Dans IBM Cloud Pak for Integration, les déploiements de IBM App Connect Enterprise incluent une autorisation restreinte pour IBM MQ. Dans ces situations, ces annotations doivent être remplacées pour garantir que IBM License Service capture l'utilisation correcte. Pour ce faire, utilisez l'approche décrite dans «Ajout d'annotations et d'étiquettes personnalisées aux ressources du gestionnaire de files d'attente», à la page 99.

Par exemple, si IBM MQ est déployé sous IBM App Connect Enterprise, utilisez l'approche illustrée dans le fragment de code suivant :

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: mq4ace
  namespace: cp4i
spec:
```

```
  annotations:
    productMetric: FREE
```

Il existe deux autres raisons pour lesquelles des annotations de licence peuvent être modifiées :

1. IBM MQ Advanced est inclus dans l'autorisation d'un autre produit IBM.
   - Dans ce cas, utilisez l'approche précédemment décrite pour IBM App Connect Enterprise.
2. IBM MQ est déployé sous une licence IBM Cloud Pak for Integration.
   - Si vous disposez d'une licence IBM Cloud Pak for Integration, vous pouvez décider de déployer un gestionnaire de files d'attente sous le rapport IBM MQ ou IBM MQ Advanced. Si vous effectuez un déploiement avec un rapport IBM MQ, vous devez vous assurer que vous n'utilisez pas de fonctions avancées telles que la haute disponibilité native ou Advanced Message Security.
   - Dans ce cas, utilisez les annotations suivantes pour l'utilisation en production :

     ```
     apiVersion: mq.ibm.com/v1beta1
     kind: QueueManager
     metadata:
       name: mq4ace
       namespace: cp4i
     spec:
       annotations:
         productID: c661609261d5471fb4ff8970a36bccea
         productCloudpakRatio: '4:1'
         productName: IBM MQ for Production
         productMetric: VIRTUAL_PROCESSOR_CORE
     ```

   - Utilisez les annotations suivantes pour une utilisation autre que la production :

     ```
     apiVersion: mq.ibm.com/v1beta1
     kind: QueueManager
     metadata:
       name: mq4ace
       namespace: cp4i
     spec:
       annotations:
         productID: 151bec68564a4a47a14e6fa99266deff
         productCloudpakRatio: '8:1'
         productName: IBM MQ for Non-Production
         productMetric: VIRTUAL_PROCESSOR_CORE
     ```

OpenShift MQ Adv. Configuration de la haute disponibilité pour les gestionnaires de files d'attente à l'aide de IBM MQ Operator

Pourquoi et quand exécuter cette tâche

Procédure

- «Native HA», à la page 21.
- «Exemple: Configuration de Native HA à l'aide de IBM MQ Operator», à la page 79.
- «Configuration d'un gestionnaire de files d'attente multi-instance à l'aide de IBM MQ Operator», à la page 85.

OpenShift MQ Adv. *Configuration de Native HA à l'aide de IBM MQ Operator*

Native HA est configurée à l'aide de l'API `QueueManager`, et des options avancées sont disponibles à l'aide d'un fichier INI.

Native HA est configurée à l'aide de `.spec.queueManager.availability` de l'API `QueueManager`, par exemple :

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

```
name: nativeha-example
spec:
  license:
    accept: false
    license: L-EHXT-MQCRN9
    use: Production
  queueManager:
    availability:
      type: NativeHA
  version: 9.4.0.0-r1
```

La zone `.spec.queueManager.availability.type` doit être définie sur `NativeHA`.

Sous `.spec.queueManager.availability`, vous pouvez également configurer un secret TLS et des chiffrements à utiliser entre les instances de gestionnaire de files d'attente lors de la réplication. Cela est fortement recommandé et un guide pas à pas est disponible dans la rubrique «Exemple: Configuration de Native HA à l'aide de IBM MQ Operator», à la page 79.

Tâches associées

«Exemple: Configuration de Native HA à l'aide de IBM MQ Operator», à la page 79
Cet exemple déploie un gestionnaire de files d'attente à l'aide de la fonction de haute disponibilité native dans OpenShift Container Platform à l'aide de IBM MQ Operator. Le protocole TLS mutuel est utilisé pour l'authentification afin de mapper un certificat TLS à une identité dans le gestionnaire de files d'attente.

Exemple: Configuration de Native HA à l'aide de IBM MQ Operator

Cet exemple déploie un gestionnaire de files d'attente à l'aide de la fonction de haute disponibilité native dans OpenShift Container Platform à l'aide de IBM MQ Operator. Le protocole TLS mutuel est utilisé pour l'authentification afin de mapper un certificat TLS à une identité dans le gestionnaire de files d'attente.

Avant de commencer

Pour mettre en oeuvre cet exemple, vous devez d'abord avoir rempli les conditions suivantes :

- Créez un projet / espace de nom OpenShift Container Platform (OCP) pour cet exemple.
- Sur la ligne de commande, connectez-vous au cluster OCP et passez à l'espace de nom ci-dessus.
- Vérifiez que IBM MQ Operator est installé et disponible dans l'espace de nom ci-dessus.

Pourquoi et quand exécuter cette tâche

Cet exemple fournit une ressource personnalisée YAML définissant un gestionnaire de files d'attente à déployer dans OpenShift Container Platform. Il détaille également les étapes supplémentaires requises pour déployer le gestionnaire de files d'attente avec TLS activé.

Procédure

1. Créez une paire de certificats comme décrit dans «Création d'une infrastructure PKI autosignée à l'aide de OpenSSL», à la page 71.
2. Créer une mappe de configuration contenant des commandes MQSC et un fichier INI

   Créez un objet Kubernetes ConfigMap contenant les commandes MQSC pour créer une file d'attente et un canal SVRCONN, et pour ajouter un enregistrement d'authentification de canal qui autorise l'accès au canal.

   Vérifiez que vous vous trouvez dans l'espace de nom que vous avez créé précédemment (voir Avant de commencer), puis entrez le fichier YAML suivant dans la console Web OCP ou à l'aide de la ligne de commande.

   ```
   apiVersion: v1
   kind: ConfigMap
   metadata:
     name: example-nativeha-configmap
   data:
     example-tls.mqsc: |
   ```

```
    DEFINE CHANNEL('MTLS.SVRCONN') CHLTYPE(SVRCONN) SSLCAUTH(REQUIRED)
SSLCIPH('ANY_TLS13_OR_HIGHER') REPLACE
    SET CHLAUTH('MTLS.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=*') USERSRC(NOACCESS)
ACTION(REPLACE)
    SET CHLAUTH('MTLS.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=example-app1') USERSRC(MAP)
MCAUSER('app1') ACTION(REPLACE)
    SET AUTHREC PRINCIPAL('app1') OBJTYPE(QMGR) AUTHADD(CONNECT,INQ)
    DEFINE QLOCAL('EXAMPLE.QUEUE') REPLACE
    SET AUTHREC PROFILE('EXAMPLE.QUEUE') PRINCIPAL('app1') OBJTYPE(QUEUE)
AUTHADD(BROWSE,PUT,GET,INQ)
  example-tls.ini: |
    Service:
        Name=AuthorizationService
        EntryPoints=14
        SecurityPolicy=UserExternal
```

Le MQSC définit un canal appelé *MTLS.SVRCONN* et une file d'attente appelée *EXAMPLE.QUEUE*. Le canal est configuré pour autoriser l'accès uniquement aux clients qui présentent un certificat avec un "nom usuel" de *example-app1*. Il s'agit du nom usuel utilisé dans l'un des certificats créés à l'étape «1», à la page 79. Les connexions sur ce canal avec ce nom usuel sont mappées à un ID utilisateur *app1*, qui est autorisé à se connecter au gestionnaire de files d'attente et à accéder à l'exemple de file d'attente. Le fichier INI active une règle de sécurité qui signifie que l'ID utilisateur *app1* n'a pas besoin d'exister dans un registre d'utilisateurs externe-il existe uniquement sous forme de nom dans cette configuration.

3. Déployez le gestionnaire de files d'attente

   Créez un gestionnaire de files d'attente à l'aide de la ressource personnalisée YAML suivante. Vérifiez que vous vous trouvez dans l'espace de nom que vous avez créé avant de commencer cette tâche, puis entrez le fichier YAML suivant dans la console Web OCP ou à l'aide de la ligne de commande. Vérifiez que la licence est correcte et acceptez-la en remplaçant `false` par `true`.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: exampleqm
spec:
  license:
    accept: false
    license: L-EHXT-MQCRN9
    use: Production
  queueManager:
    name: EXAMPLEQM
    availability:
      type: NativeHA
      tls:
        secretName: example-qm-tls
    mqsc:
    - configMap:
        name: example-nativeha-configmap
        items:
        - example-tls.mqsc
    ini:
    - configMap:
        name: example-nativeha-configmap
        items:
        - example-tls.ini
    storage:
      queueManager:
        type: persistent-claim
  version: 9.4.0.0-r1
  pki:
    keys:
      - name: default
        secret:
          secretName: example-qm-tls
          items:
            - tls.key
            - tls.crt
            - ca.crt
```

Notez que le secret *example-qm-tls* a été créé à l'étape «1», à la page 79et que l' *exemple-nativeha-configmap* ConfigMap a été créé à l'étape «2», à la page 79

Le type de disponibilité est défini sur *NativeHA*et le stockage persistant est sélectionné. La classe de stockage par défaut configurée dans votre cluster Kubernetes sera utilisée. Si aucune classe de stockage n'est configurée par défaut ou si vous souhaitez utiliser une autre classe de stockage, ajoutez `defaultClass: storage_class_name` sous `spec.queueManager.storage`.

Les trois pods d'un gestionnaire de files d'attente Native HA répliquent les données sur le réseau. Ce lien n'est pas chiffré par défaut, mais cet exemple utilise le certificat du gestionnaire de files d'attente pour le chiffrement du trafic. Vous pouvez spécifier un autre certificat pour une sécurité supplémentaire. Le secret TLS Native HA doit être un secret TLS Kubernetes , qui possède une structure particulière (par exemple, la clé privée doit être appelée *tls.key*).

4. Confirmez que le gestionnaire de files d'attente est en cours d'exécution

   Le gestionnaire de files d'attente est en cours de déploiement. Confirmez qu'il se trouve dans l'état `Running` avant de continuer. Exemple :

   ```
   oc get qmgr exampleqm
   ```

5. Testez la connexion au gestionnaire de files d'attente

   Pour vérifier que le gestionnaire de files d'attente est configuré et disponible, suivez les étapes de la rubrique «Test d'une connexion TLS mutuelle à un gestionnaire de files d'attente à partir de votre ordinateur portable», à la page 75.

6. Forcer l'échec du pod actif

   Pour valider la reprise automatique du gestionnaire de files d'attente, simulez un échec du pod :

   a) Afficher les pods actif et de secours

      Exécutez ensuite la commande suivante :

      ```
      oc get pods --selector app.kubernetes.io/instance=exampleqm
      ```

      Notez que dans la zone **READY** , le pod actif renvoie la valeur `1/1`, alors que les pods de réplique renvoient la valeur `0/1`.

   b) Supprimer le pod actif

      Exécutez la commande suivante en spécifiant le nom complet du pod actif :

      ```
      oc delete pod exampleqm-ibm-mq-value
      ```

   c) Afficher de nouveau le statut du pod

      Exécutez ensuite la commande suivante :

      ```
      oc get pods --selector app.kubernetes.io/instance=exampleqm
      ```

   d) Afficher le statut du gestionnaire de files d'attente

      Exécutez la commande suivante en spécifiant le nom complet de l'un ou l'autre des pods :

      ```
      oc exec -t Pod -- dspmq -o nativeha -x -m EXAMPLEQM
      ```

      Le statut doit indiquer que l'instance active a été modifiée. Par exemple :

      ```
      QMNAME(EXAMPLEQM) ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
      INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
      CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
      INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
      CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
      INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
      CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
      ```

   e) Testez à nouveau la connexion au gestionnaire de files d'attente

      Pour confirmer la reprise du gestionnaire de files d'attente, suivez les étapes de la rubrique «Test d'une connexion TLS mutuelle à un gestionnaire de files d'attente à partir de votre ordinateur portable», à la page 75.

Résultats

Félicitations, vous avez déployé avec succès un gestionnaire de files d'attente avec une haute disponibilité native et une authentification TLS mutuelle, et vous avez vérifié qu'il est automatiquement restauré lorsque le pod actif échoue.

OpenShift MQ Adv. *Affichage du statut des gestionnaires de files d'attente Native HA pour les conteneurs IBM MQ*

Pour les conteneurs IBM MQ , vous pouvez afficher le statut des instances Native HA en exécutant la commande **dspmq** dans l'un des pods en cours d'exécution.

Pourquoi et quand exécuter cette tâche

Vous pouvez utiliser la commande **dspmq** dans l'un des pods en cours d'exécution pour afficher le statut opérationnel d'une instance de gestionnaire de files d'attente. Les informations renvoyées varient selon que l'instance est active ou qu'il s'agit d'une réplique. Les informations fournies par l'instance active sont définitives, tandis que celles des noeuds de réplique peuvent être obsolètes.

Vous pouvez effectuer les actions suivantes :

- Déterminer si l'instance de gestionnaire de files d'attente sur le noeud actuel est active ou s'il s'agit d'une réplique.
- Afficher le statut Native HA opérationnel de l'instance sur le noeud actuel.
- Afficher le statut opérationnel des trois instances dans une configuration Native HA.

Les zones de statut suivantes sont utilisées pour signaler le statut de la configuration Native HA :

ROLE
Indique le rôle en cours de l'instance et est l'un des rôles `Active`, `Replica` ou `Unknown`.

INSTANCE
Nom fourni pour cette instance du gestionnaire de files d'attente lorsque ce dernier a été créé à l'aide de l'option **-lr** de la commande **crtmqm**.

INSYNC
Indique si l'instance peut prendre la relève en tant qu'instance active, si nécessaire.

QUORUM
Indique le statut de quorum au format *nombre_instances_synchronisées*/*nombre_instances_configurées*.

REPLADDR
Adresse de réplication de l'instance de gestionnaire de files d'attente.

CONNACTV
Indique si le noeud est connecté à l'instance active.

BACKLOG
Indique le nombre de kilooctets de retard de l'instance.

CONNINST
Indique si l'instance désignée est connectée à cette instance.

ALTDATE
Indique la date à laquelle ces informations ont été mises à jour pour la dernière fois (vide si elles n'ont jamais été mises à jour).

ALTTIME
Indique l'heure à laquelle ces informations ont été mises à jour pour la dernière fois (vide si elles n'ont jamais été mises à jour).

Procédure

- Recherchez les pods qui font partie de votre gestionnaire de files d'attente.

```
oc get pod --selector app.kubernetes.io/instance=nativeha-qm
```

- Exécutez le dspmq dans l'un des pods

```
oc exec -t Pod dspmq
```

```
oc rsh Pod
```

pour un shell interactif, où vous pouvez exécuter dspmq directement.

- Pour déterminer si une instance de gestionnaire de files d'attente est exécutée comme instance active ou comme réplique :

```
oc exec -t Pod dspmq -o status -m QMgrName
```

Une instance active d'un gestionnaire de files d'attente nommé BOB signale le statut suivant :

```
QMNAME(BOB)                STATUS(Running)
```

Une réplique d'instance d'un gestionnaire de files d'attente nommé BOB signale le statut suivant :

```
QMNAME(BOB)                STATUS(Replica)
```

Une instance inactive signale le statut suivant :

```
QMNAME(BOB)                STATUS(Ended Immediately)
```

- Pour déterminer le statut Native HA opérationnel de l'instance dans le pod spécifié :

```
oc exec -t Pod dspmq -o nativeha -m QMgrName
```

L'instance active d'un gestionnaire de files d'attente nommé BOB peut signaler le statut suivant :

```
QMNAME(BOB)              ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
```

Une réplique d'instance d'un gestionnaire de files d'attente nommé BOB peut signaler le statut suivant :

```
QMNAME(BOB)              ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
```

Une instance inactive d'un gestionnaire de files d'attente nommé BOB peut signaler le statut suivant :

```
QMNAME(BOB)              ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
```

- Pour déterminer le statut Native HA opérationnel de toutes les instances de la configuration Native HA :

```
oc exec -t Pod dspmq -o nativeha -x -m QMgrName
```

Si vous exécutez cette commande sur le noeud qui exécute l'instance active du gestionnaire de files d'attente BOB, vous risquez de recevoir le statut suivant :

```
QMNAME(BOB)            ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
 INSTANCE(inst1) ROLE(Active)  REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
 INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
 INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
```

Si vous exécutez cette commande sur un noeud qui exécute une réplique d'instance du gestionnaire de files d'attente BOB, vous risquez de recevoir le statut suivant, qui indique que l'une des répliques est en retard :

```
QMNAME(BOB)            ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
 INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
```

```
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
 INSTANCE(inst1) ROLE(Active)  REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
 INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(No)  BACKLOG(435)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
```

Si vous exécutez cette commande sur un noeud qui exécute une instance inactive du gestionnaire de files d'attente BOB, vous risquez de recevoir le statut suivant :

```
QMNAME(BOB)            ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
 INSTANCE(inst1) ROLE(Unknown) REPLADDR(9.20.123.45) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATE() ALTTIME()
 INSTANCE(inst2) ROLE(Unknown) REPLADDR(9.20.123.46) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATE() ALTTIME()
 INSTANCE(inst3) ROLE(Unknown) REPLADDR(9.20.123.47) CONNACTV(No) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATE() ALTTIME()
```

Si vous exécutez la commande alors que les instances sont encore en cours de négociation pour déterminer l'instance active et les répliques, vous recevez le statut suivant :

```
QMNAME(BOB)              STATUS(Negotiating)
```

Tâches associées

«Exemple: Configuration de Native HA à l'aide de IBM MQ Operator», à la page 79
Cet exemple déploie un gestionnaire de files d'attente à l'aide de la fonction de haute disponibilité native dans OpenShift Container Platform à l'aide de IBM MQ Operator. Le protocole TLS mutuel est utilisé pour l'authentification afin de mapper un certificat TLS à une identité dans le gestionnaire de files d'attente.

Référence associée

Commande dspmq (display queue managers)

OpenShift MQ Adv. *Réglage avancé pour Native HA*

Paramètres avancés pour l'optimisation des délais et des intervalles. Il n'est pas nécessaire d'utiliser ces paramètres sauf si les valeurs par défaut ne respectent pas la configuration requise par votre système.

Les options de base de configuration des AP natives sont gérées à l'aide de l'API `QueueManager`, que IBM MQ Operator utilise pour configurer les fichiers INI du gestionnaire de files d'attente sous-jacent pour vous. Il existe des options plus avancées qui ne sont configurables qu'à l'aide d'un fichier INI, dans la section NativeHALocalInstance. Voir aussi «Exemple : fourniture de fichiers MQSC et INI», à la page 69 pour plus d'informations sur la configuration d'un fichier INI.

`HeartbeatInterval`
: L'intervalle des pulsations définit la fréquence en millisecondes à laquelle une instance active d'un gestionnaire de files d'attente Native HA envoie une pulsation réseau. Il est compris entre 500 (0,5 secondes) et 60000 (1 minute). Une valeur hors de cette plage empêche le démarrage du gestionnaire de files d'attente. Si cet attribut est omis, une valeur par défaut de 5000 (5 secondes) est utilisée. Chaque instance doit utiliser le même intervalle de pulsations.

`HeartbeatTimeout`
: Le dépassement du délai d'attente du signal de présence définit le temps pendant lequel une réplique d'instance d'un gestionnaire de files d'attente Native HA attend avant de considérer que l'instance active ne répondra pas. Cette valeur doit être comprise entre 500 (0,5 secondes) et 120000 (2 minutes). La valeur du dépassement du délai d'attente du signal de présence doit être supérieure ou égale à celle de l'intervalle des pulsations.

  Une valeur non valide empêche le démarrage du gestionnaire de files d'attente. Si cet attribut est omis, une réplique attend 2 x `HeartbeatInterval` avant de lancer le processus pour sélectionner une nouvelle instance active. Chaque instance doit utiliser la même valeur de dépassement du délai d'attente du signal de présence.

`RetryInterval`
: L'intervalle entre les nouvelles tentatives définit la fréquence en millisecondes à laquelle un gestionnaire de files d'attente Native HA doit retenter un lien de réplication défectueux. Cet intervalle

doit être compris entre 500 (0,5 secondes) et 120000 (2 minutes). Si cet attribut est omis, une réplique attend 2 x `HeartbeatInterval` avant de réessayer un lien de réplication ayant échoué.

OpenShift MQ Adv. *Arrêt des gestionnaires de files d'attente Native HA*

Vous pouvez utiliser la commande **endmqm** pour arrêter un gestionnaire de files d'attente actif ou de réplique faisant partie d'un groupe Native HA.

Procédure

- Pour arrêter l'instance active d'un gestionnaire de files d'attente, voir Arrêt des gestionnaires de files d'attente Native HA dans la section Configuration de cette documentation.

OpenShift CP4I MQ Adv. Kubernetes *Configuration d'un gestionnaire de files d'attente multi-instance à l'aide de IBM MQ Operator*

Cet exemple déploie un gestionnaire de files d'attente multi-instance à l'aide de OpenShift Container Platform à l'aide de IBM MQ Operator. Le protocole TLS mutuel est utilisé pour l'authentification afin de mapper un certificat TLS à une identité dans le gestionnaire de files d'attente.

Avant de commencer

Pour mettre en oeuvre cet exemple, vous devez d'abord avoir rempli les conditions suivantes :

- Créez un projet / espace de nom OpenShift Container Platform (OCP) pour cet exemple.
- Sur la ligne de commande, connectez-vous au cluster OCP et passez à l'espace de nom ci-dessus.
- Vérifiez que IBM MQ Operator est installé et disponible dans l'espace de nom ci-dessus.

Pourquoi et quand exécuter cette tâche

Cet exemple fournit une ressource personnalisée YAML définissant un gestionnaire de files d'attente à déployer dans OpenShift Container Platform. Il détaille également les étapes supplémentaires requises pour déployer le gestionnaire de files d'attente avec TLS activé.

Procédure

1. Détermination d'une classe de stockage appropriée

   Le stockage dans un cluster Kubernetes est accessible à l'aide de plusieurs modes d'accès aux volumes persistants. Un gestionnaire de files d'attente multi-instance crée plusieurs volumes persistants: un pour chaque gestionnaire de files d'attente et au moins un volume partagé. Le volume partagé d'un gestionnaire de files d'attente multi-instance doit utiliser une classe de stockage `ReadWriteMany` . La classe de stockage par défaut dans un cluster Kubernetes est généralement destinée à une classe de stockage `ReadWriteOnce` (stockage par blocs). Par exemple, si vous utilisez Red Hat OpenShift Data Foundation, la classe de stockage *ocs-storagecluster-cephfs* fournit un système de fichiers partagé adapté. Le choix du système de fichiers est très important, car tous les systèmes de fichiers partagés ne gèrent pas le verrouillage de fichiers de la même manière. Voir Planification de la prise en charge du système de fichiers sur Multiplatforms et Testing statement for IBM MQ multi-instance queue manager file systems.
2. Créez une paire de certificats comme décrit dans «Création d'une infrastructure PKI autosignée à l'aide de OpenSSL», à la page 71.
3. Créer une mappe de configuration contenant des commandes MQSC et un fichier INI

   Créez un objet Kubernetes ConfigMap contenant les commandes MQSC pour créer une file d'attente et un canal SVRCONN, et pour ajouter un enregistrement d'authentification de canal qui autorise l'accès au canal.

Vérifiez que vous vous trouvez dans l'espace de nom que vous avez créé précédemment (voir Avant de commencer), puis entrez le fichier YAML suivant dans la console Web OCP ou à l'aide de la ligne de commande.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: example-miqm-configmap
data:
  example-tls.mqsc: |
    DEFINE CHANNEL('MTLS.SVRCONN') CHLTYPE(SVRCONN) SSLCAUTH(REQUIRED)
SSLCIPH('ANY_TLS13_OR_HIGHER') REPLACE
    SET CHLAUTH('MTLS.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=*') USERSRC(NOACCESS)
ACTION(REPLACE)
    SET CHLAUTH('MTLS.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=example-app1') USERSRC(MAP)
MCAUSER('app1') ACTION(REPLACE)
    SET AUTHREC PRINCIPAL('app1') OBJTYPE(QMGR) AUTHADD(CONNECT,INQ)
    DEFINE QLOCAL('EXAMPLE.QUEUE') REPLACE
    SET AUTHREC PROFILE('EXAMPLE.QUEUE') PRINCIPAL('app1') OBJTYPE(QUEUE)
AUTHADD(BROWSE,PUT,GET,INQ)
  example-tls.ini: |
    Service:
        Name=AuthorizationService
        EntryPoints=14
        SecurityPolicy=UserExternal
```

Le MQSC définit un canal appelé *MTLS.SVRCONN* et une file d'attente appelée *EXAMPLE.QUEUE*. Le canal est configuré pour autoriser l'accès uniquement aux clients qui présentent un certificat avec un "nom usuel" de *example-app1*. Il s'agit du nom usuel utilisé dans l'un des certificats créés à l'étape «2», à la page 85. Les connexions sur ce canal avec ce nom usuel sont mappées à un ID utilisateur *app1*, qui est autorisé à se connecter au gestionnaire de files d'attente et à accéder à l'exemple de file d'attente. Le fichier INI active une règle de sécurité qui signifie que l'ID utilisateur *app1* n'a pas besoin d'exister dans un registre d'utilisateurs externe-il existe uniquement sous forme de nom dans cette configuration.

4. Déployez le gestionnaire de files d'attente

   Créez un gestionnaire de files d'attente à l'aide de la ressource personnalisée YAML suivante. Vérifiez que vous vous trouvez dans l'espace de nom que vous avez créé avant de commencer cette tâche, puis entrez le fichier YAML suivant dans la console Web OCP ou à l'aide de la ligne de commande. Vérifiez que la licence est correcte et acceptez-la en remplaçant `false` par `true`.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: exampleqm
spec:
  license:
    accept: false
    license: L-EHXT-MQCRN9
    use: Production
  queueManager:
    name: EXAMPLEQM
    availability:
      type: MultiInstance
    mqsc:
    - configMap:
        name: example-miqm-configmap
        items:
        - example-tls.mqsc
    ini:
    - configMap:
        name: example-miqm-configmap
        items:
        - example-tls.ini
    storage:
      defaultClass: STORAGE_CLASS
  version: 9.4.0.0-r1
  pki:
    keys:
      - name: default
        secret:
          secretName: example-qm-tls
          items:
            - tls.key
```

```
            - tls.crt
            - ca.crt
```

Remplacez *STORAGE_CLASS* par la classe de stockage que vous avez identifiée à l'étape «1», à la page 85.

Notez que la valeur confidentielle *example-qm-tls* a été créée à l'étape «2», à la page 85et que la valeur confidentielle ConfigMap *example-miqm-configmap* a été créée à l'étape «3», à la page 85

Le type de disponibilité est défini sur *MultiInstance*, ce qui entraîne la sélection automatique du stockage persistant.

5. Confirmez que le gestionnaire de files d'attente est en cours d'exécution

   Le gestionnaire de files d'attente est en cours de déploiement. Confirmez qu'il se trouve dans l'état `Running` avant de continuer. Exemple :

   ```
   oc get qmgr exampleqm
   ```

6. Testez la connexion au gestionnaire de files d'attente

   Pour vérifier que le gestionnaire de files d'attente est configuré et disponible, suivez les étapes de la rubrique «Test d'une connexion TLS mutuelle à un gestionnaire de files d'attente à partir de votre ordinateur portable», à la page 75.

7. Forcer l'échec du pod actif

   Pour valider la reprise automatique du gestionnaire de files d'attente, simulez un échec du pod :

   a) Afficher les pods actif et de secours

      Exécutez ensuite la commande suivante :

      ```
      oc get pods --selector app.kubernetes.io/instance=exampleqm
      ```

      Notez que dans la zone **READY**, le pod actif renvoie la valeur `1/1`, alors que la pod de secours renvoie la valeur `0/1`.

   b) Supprimer le pod actif

      Exécutez la commande suivante en spécifiant le nom complet du pod actif :

      ```
      oc delete pod exampleqm-ibm-mq-value
      ```

   c) Afficher de nouveau le statut du pod

      Exécutez ensuite la commande suivante :

      ```
      oc get pods --selector app.kubernetes.io/instance=exampleqm
      ```

   d) Afficher le statut du gestionnaire de files d'attente

      Exécutez la commande suivante en spécifiant le nom complet de l'autre pod:

      ```
      oc exec -t Pod -- dspmq -x
      ```

      Le statut doit indiquer que l'instance active a été modifiée. Par exemple :

      ```
      QMNAME(EXAMPLEQM)                                         STATUS(Running as standby)
          INSTANCE(exampleqm-ibm-mq-1) MODE(Active)
          INSTANCE(exampleqm-ibm-mq-0) MODE(Standby)
      ```

   e) Testez à nouveau la connexion au gestionnaire de files d'attente

      Pour confirmer la reprise du gestionnaire de files d'attente, suivez les étapes de la rubrique «Test d'une connexion TLS mutuelle à un gestionnaire de files d'attente à partir de votre ordinateur portable», à la page 75.

Résultats

Félicitations, vous avez déployé avec succès un gestionnaire de files d'attente multi-instance avec authentification TLS mutuelle et vérifié qu'il est automatiquement restauré lorsque le pod actif échoue.

OpenShift CP4I CD Configuration d'une route pour la connexion à un gestionnaire de files d'attente depuis l'extérieur d'un cluster Red Hat OpenShift

Vous avez besoin d'une route Red Hat OpenShift pour connecter une application à un gestionnaire de files d'attente IBM MQ depuis l'extérieur d'un cluster Red Hat OpenShift . Vous devez activer TLS sur votre gestionnaire de files d'attente et votre application client IBM MQ , car SNI est disponible uniquement dans le protocole TLS lorsqu'un protocole TLS 1.2 ou supérieur est utilisé. Red Hat OpenShift Container Platform Router utilise l'indication de nom de serveur pour acheminer les demandes vers le gestionnaire de files d'attente IBM MQ.

Pourquoi et quand exécuter cette tâche

La configuration requise de l' Red Hat OpenShift Route dépend du comportement de l' indication de nom de serveur (SNI) de votre application client. IBM MQ prend en charge deux paramètres d'en-tête SNI différents selon le type de configuration et de client. Un en-tête SNI est défini sur le nom d'hôte de la destination du client ou sur le nom de canal IBM MQ . Pour plus d'informations sur la façon dont IBM MQ mappe un nom de canal à un nom d'hôte, voir How IBM MQ provides multiple certificates capability.

Si un en-tête SNI est défini sur un nom de canal IBM MQ ou qu'un nom d'hôte est contrôlé à l'aide de l'attribut **OutboundSNI** . Les valeurs possibles sont `OutboundSNI=CHANNEL` (valeur par défaut) ou `OutboundSNI=HOSTNAME`. Pour plus d'informations, voir Strophe SSL du fichier de configuration du client. Notez que `CHANNEL` et `HOSTNAME` sont les valeurs exactes que vous utilisez ; il ne s'agit pas de noms de variable que vous remplacez par un nom de canal ou un nom d'hôte réel.

Comportements des clients avec différents paramètres `OutboundSNI`

Si **`OutboundSNI`** est défini sur `HOSTNAME`, les clients suivants définissent une SNI de nom d'hôte tant qu'un nom d'hôte est fourni dans le nom de la connexion :

- Clients C
- Clients .NET en mode non géré
- Clients Java/JMS

Si **`OutboundSNI`** est défini sur `HOSTNAME` et qu'une adresse IP est utilisée dans le nom de connexion, les clients suivants envoient un en-tête SNI vide :

- Clients C
- Clients .NET en mode non géré
- Clients Java/JMS (qui ne peuvent pas effectuer une recherche DNS inverse du nom d'hôte)

Si **`OutboundSNI`** est défini sur `CHANNEL`ou n'est pas défini, un nom de canal IBM MQ est utilisé à la place et est toujours envoyé, qu'un nom d'hôte ou un nom de connexion d'adresse IP soit utilisé ou non.

Les types de client suivants ne prennent pas en charge la définition d'un en-tête SNI dans un nom de canal IBM MQ et tentent ainsi de définir l'en-tête SNI sur un nom d'hôte quel que soit le paramètre **`OutboundSNI`** :

- Clients AMQP
- Clients XR

Le client IBM MQ géré .NET définit SERVERNAME sur le nom d'hôte respectif si la propriété **`OutboundSNI`** est définie sur `HOSTNAME`, ce qui permet à un client IBM MQ géré .NET de se connecter à un gestionnaire de files d'attente à l'aide de routes Red Hat OpenShift .

Si une application client se connecte à un gestionnaire de files d'attente déployé dans un cluster Red Hat OpenShift via IBM MQ Internet Pass-Thru (MQIPT), MQIPT peut être configuré pour définir le SNI sur le nom d'hôte à l'aide de la propriété SSLClientOutboundSNI dans la définition de route.

OutboundSNI, plusieurs certificats et routes Red Hat OpenShift

IBM MQ utilise l'en-tête SNI pour fournir plusieurs fonctionnalités de certificats. Si une application se connecte à un canal IBM MQ configuré pour utiliser un certificat différent via la zone CERTLABL, elle doit se connecter avec le paramètre **OutboundSNI** de CHANNEL.

Si votre configuration de route Red Hat OpenShift requiert un HOSTNAME SNI, vous ne pouvez pas utiliser la fonctionnalité de certificats multiples de IBM MQ et vous ne pouvez pas définir de paramètre CERTLABL sur un objet canal IBM MQ .

Si une application avec un paramètre **OutboundSNI** autre que CHANNEL se connecte à un canal avec un libellé de certificat configuré, l'application est rejetée avec une erreur MQRC_SSL_INITIALIZATION_ERROR et un message AMQ9673 est imprimé dans les journaux d'erreurs du gestionnaire de files d'attente.

Pour plus d'informations sur la façon dont IBM MQ fournit plusieurs fonctionnalités de certificat, voir How IBM MQ fournit plusieurs fonctionnalités de certificat .

Exemple

Les applications client qui permettent de définir le SNI sur le canal MQ requièrent la création d'une nouvelle route Red Hat OpenShift pour chaque canal auquel vous souhaitez vous connecter. Vous devez aussi utiliser des noms de canal uniques dans votre cluster Red Hat OpenShift Container Platform pour permettre un routage vers le gestionnaire de files d'attente approprié.

Il est important que les noms de canal MQ ne se terminent pas par une lettre minuscule en raison de la manière dont IBM MQ mappe les noms de canal aux en-têtes SNI.

Pour déterminer le nom d'hôte requis pour chacune de vos nouvelles routes Red Hat OpenShift, vous devez associer chaque nom de canal à une adresse SNI. Pour plus d'informations, voir How IBM MQ provides multiple certificates capability.

Vous devez ensuite créer une nouvelle route Red Hat OpenShift pour chaque canal, en appliquant le `yaml` suivant dans votre cluster :

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: unique_name_for_the_route
  namespace: namespace_of_your_MQ_deployment
spec:
  host: SNI_address_mapping_for_the_channel
  to:
    kind: Service
    name: name_of_Kubernetes_Service_for_your_MQ_deployment (for example "queue_manager_name-
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 "queue_manager_name-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 Red Hat OpenShift Container Platform ; il s'agit normalement du port 443.

Tâches associées

«Connexion à IBM MQ Console déployée dans un cluster Red Hat OpenShift», à la page 134
Comment se connecter au IBM MQ Console d'un gestionnaire de files d'attente qui a été déployé sur un cluster Red Hat OpenShift Container Platform .

OpenShift CP4I Intégration d' IBM MQ à la fonction de trace IBM Instana

IBM Instana peut être utilisé pour tracer des transactions dans IBM Cloud Pak for Integration.

Avant de commencer

Ce document traite du traçage IBM Instana , qui est le processus de traçage des messages via un système. Il ne couvre pas la surveillance IBM Instana , dans laquelle des détails sont extraits sur l'état d'un gestionnaire de files d'attente IBM MQ . Pour plus d'informations sur la surveillance d' IBM MQ par IBM Instana , voir Surveillance d' IBM MQ. Pour des instructions détaillées sur la surveillance authentifiée, voir «Configuration de la surveillance IBM Instana authentifiée avec TLS», à la page 91.

Remarque :

- Cette fonction est prise en charge uniquement sur les opérandes d' IBM MQ version 9.3.1.0-r2 ou ultérieure.
- Vous pouvez exécuter la fonction de trace IBM Instana sur les versions précédentes d' IBM MQ Operator et du gestionnaire de files d'attente, mais pas en mode natif. Voir Configuration de la fonction de trace d' IBM MQ dans la documentation IBM Instana .

Avant de pouvoir exécuter le traçage IBM Instana avec l'opérateur IBM MQ , vous devez déployer à la fois un système de back end IBM Instana et des agents IBM Instana . Par défaut, un gestionnaire de files d'attente IBM MQ communique avec un agent IBM Instana déployé sur le même noeud que le pod du gestionnaire de files d'attente.

Pourquoi et quand exécuter cette tâche

L'activation de l'intégration à IBM Instana entraîne l'installation d'un exit API IBM MQ dans votre gestionnaire de files d'attente. L'exit API envoie des données de trace aux agents IBM Instana sur les messages qui transitent par le gestionnaire de files d'attente.

L'exit API ajoute des en-têtes RFH2 à chaque message. Ces en-têtes contiennent des informations de trace.

Les agents IBM Instana sont chargés d'envoyer les données de trace au système dorsal IBM Instana .

Pour plus d'informations sur le déploiement d'un système de back end IBM Instana et d'agents IBM Instana , voir Activation des liens de surveillance Instana dans l'interface utilisateur de la plateforme dans la documentation IBM Instana .

Procédure

Déploiement standard

- Déployez un gestionnaire de files d'attente avec la fonction de trace IBM Instana activée.

  Par défaut, la fonction de trace IBM Instana est désactivée.

  Si vous utilisez IBM Cloud Pak for Integration Platform UI ou la console Web OpenShift :

  1. Cliquez sur **Télémétrie** > **Fonction de trace** > **Instana**.
  2. Définissez le bouton à bascule **Activer la fonction de trace d'Instana** sur `true`.

  Si vous déployez via YAML, utilisez le fragment suivant:

```
spec:
  telemetry:
    tracing:
      instana:
        enabled: true
```

Déploiement avancé

- Communiquez avec l'agent IBM Instana via https.

Par défaut, l'exit IBM Instana pour IBM MQ communique avec l'agent IBM Instana via http. L'adresse hôte de l'agent est définie sur l'adresse IP du noeud sur lequel s'exécute le gestionnaire de files d'attente. Cela correspond à la configuration décrite dans Activation de la surveillance IBM Instana dans la documentation IBM Instana , où les agents IBM Instana sont déployés par l'opérateur d'agent IBM Instana en tant que daemonset.

Actuellement, la communication entre l'exit IBM Instana pour IBM MQ et l'agent IBM Instana prend en charge les protocoles http ou https. Pour utiliser https, l'agent IBM Instana doit d'abord être configuré pour utiliser le chiffrement TLS. Voir Configuration du chiffrement TLS pour le noeud final de l'agent dans la documentation IBM Instana . Le protocole peut ensuite être défini sur https comme suit:

Si vous utilisez la console Web OpenShift :

1. Cliquez sur **Telemetry** > **Instana**.
2. Développez la liste déroulante **Configuration avancée** .
3. Définissez le **protocole de communication de l'agent Instana** sur `https`.

Si vous déployez via YAML, utilisez le fragment suivant:

```
spec:
  telemetry:
    instana:
      enabled: true
      protocol: https
```

- Définissez le **agentHost**

  Si les agents IBM Instana n'ont pas été déployés comme daemonset sur le cluster Openshift sur lequel le gestionnaire de files d'attente s'exécute, vous devez définir la valeur **agentHost** sur le nom d'hôte ou l'adresse IP sur lequel l'agent IBM Instana s'exécute. La valeur **agentHost** ne doit pas inclure de protocole ou de port.

  Si vous utilisez la console Web OpenShift :

  1. Cliquez sur **Telemetry** > **Instana**.
  2. Développez la liste déroulante **Configuration avancée** .
  3. Entrez votre nom d'hôte dans la zone de texte **Hôte de l'agent Instana** .

  Si vous déployez via YAML, utilisez le fragment suivant:

```
spec:
  telemetry:
    instana:
      enabled: true
      agentHost: 9.9.9.9
```

Que faire ensuite

Voir aussi «Déploiement d'un gestionnaire de files d'attente simple à l'aide de IBM MQ Operator», à la page 66.

OpenShift CP4I Operator2.2.0 Configuration de la surveillance IBM Instana authentifiée avec TLS

Pour pouvoir surveiller un gestionnaire de files d'attente via un agent IBM Instana , vous devez configurer à la fois l'agent et le gestionnaire de files d'attente.

Avant de commencer

La section "Configuration" de la rubrique "Surveillance IBM MQ" de la documentation IBM Instana fournit des informations générales sur la configuration de la surveillance IBM Instana . Toutefois, il n'inclut pas de détails sur la configuration du gestionnaire de files d'attente.

Avant de pouvoir exécuter le traçage IBM Instana avec l'opérateur IBM MQ , vous devez déployer à la fois un système de back end IBM Instana et des agents IBM Instana . Pour ce faire, voir Enabling IBM Instana monitoring in CP4I Platform UI dans la documentation IBM Instana .

Procédure

1. Générer des certificats.
2. Configurez les agents IBM Instana.
3. Configurez le gestionnaire de files d'attente.
4. Vérification et débogage.

Tâches associées

«Intégration d' IBM MQ à la fonction de trace IBM Instana», à la page 90
IBM Instana peut être utilisé pour tracer des transactions dans IBM Cloud Pak for Integration.

Génération d'un certificat et d'une clé pour l'agent IBM Instana et le gestionnaire de files d'attente

Pour la communication TLS entre l'agent IBM Instana et le gestionnaire de files d'attente, les deux doivent disposer d'un certificat et d'une clé privée correspondante.

Avant de commencer

Il s'agit de la première des quatre tâches permettant de configurer la surveillance IBM Instana authentifiée avec TLS.

Remarque : Les valeurs utilisées lors de la génération de ces certificats sont utilisées à des fins de démonstration. Lors du déploiement dans un environnement de production, assurez-vous que le sujet et l'expiration du certificat sont appropriés.

Procédure

IBM MQ Gestionnaire de files d'attente

Pour communiquer avec l'agent IBM Instana via TLS, le gestionnaire de files d'attente doit disposer d'un certificat et d'une clé privée correspondante. Si vous en avez déjà, ignorez cette section.

1. Générez un certificat et une clé privée pour le gestionnaire de files d'attente.

   Exécutez ensuite la commande suivante :

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

agentIBM Instana

Pour que l'agent puisse établir une communication TLS avec le gestionnaire de files d'attente IBM MQ , il doit disposer d'un certificat et de la clé privée correspondante. Si vous disposez déjà d'une clé privée et d'un certificat dans un magasin de clés JKS que vous souhaitez utiliser, ignorez cette section.

2. Générez un certificat et une clé privée pour l'agent IBM Instana .

   Exécutez ensuite la commande suivante :

```
openssl req \
       -newkey rsa:2048 -nodes -keyout application.key \
       -subj "/CN=instana-agent/OU=app team1" \
       -x509 -days 3650 -out application.crt
```

3. Stockez le certificat et la clé privée dans un magasin de clés PKCS12 .

Exécutez la commande suivante, en remplaçant *votre_mot_de_passe* par le mot de passe que vous souhaitez utiliser pour sécuriser le magasin de clés. Effectuez ce remplacement dans toutes les étapes suivantes.

```
openssl pkcs12 -export -out application.p12 -inkey application.key -in application.crt
-passout pass:your_password
```

4. Convertissez le magasin de clés PKCS12 en magasin de clés JKS.

   Exécutez ensuite la commande suivante :

```
keytool -importkeystore \
        -srckeystore application.p12 \
        -srcstoretype pkcs12 \
        -destkeystore application.jks \
        -deststoretype JKS \
        -srcstorepass your_password \
        -deststorepass your_password \
        -noprompt
```

5. Etiquetez le certificat.

   Exécutez ensuite la commande suivante :

```
keytool -changealias -alias "1" -destalias "instana" -keypass your_password -keystore
application.jks -storepass your_password -noprompt
```

6. Importez le certificat du gestionnaire de files d'attente dans le magasin de clés.

   Exécutez ensuite la commande suivante :

```
keytool -importcert -file server.crt -keystore application.jks -storepass your_password
-alias myca -noprompt
```

Que faire ensuite

Vous êtes maintenant prêt à configurer les agents pour la IBM Instana surveillance.

OpenShift CP4I Operator 2.2.0 ***Surveillance d'Instana: configuration des agents***

Montez le magasin de clés sur les agents IBM Instana , puis configurez la surveillance pour un gestionnaire de files d'attente spécifique.

Avant de commencer

Cette tâche suppose que vous avez généré un certificat et une clé pour les agents IBM Instana et le gestionnaire de files d'attente.

Procédure

Montage du magasin de clés sur les agents IBM Instana

1. Créez un secret à partir de votre magasin de clés JKS dans l'espace de nom de l'agent IBM Instana .

   Exécutez la commande suivante en remplaçant *keystore_secret_name* par le nom que vous souhaitez utiliser. Effectuez ce remplacement dans toutes les étapes suivantes.

```
oc create secret generic keystore_secret_name --from-file=./application.jks -n instana-agent
```

2. Dans l'espace de nom instana-agent, utilisez la commande `oc edit daemonset instana-agent` pour éditer le daemonset instana-agent afin d'inclure le volumeMount et le volume supplémentaires suivants:

```
volumeMounts:
- name: mq-key-jks-name
   subPath: application.jks
   mountPath: /opt/instana/agent/etc/application.jks
volumes:
- name: mq-key-jks-name
```

```
    secret:
      secretName: keystore_secret_name
```

Configuration de la surveillance pour un gestionnaire de files d'attente spécifique

3. Dans l'espace de nom instana-agent, utilisez la commande `oc edit configmap instana-agent` pour éditer la mappe de configuration instana-agent.
4. Ajoutez la section suivante sous `configuration.yaml: |`. Si vous avez déjà défini cette section, ajoutez simplement le nouveau gestionnaire de files d'attente à la liste.

```
    com.instana.plugin.ibmmq:
      enabled: true
      poll_rate: 60
      queueManagers:
         QUEUE_MANAGER_NAME:
            channel: 'INSTANA.A.SVRCONN'
            keystorePassword: 'your_password'
            keystore: '/opt/instana/agent/etc/application.jks'
            cipherSuite: 'TLS_RSA_WITH_AES_256_CBC_SHA256'
```

Où

- *votre_mot de passe* est le mot de passe de votre magasin de clés JKS
- *QUEUE_MANAGER_NAME* est le nom du gestionnaire de files d'attente IBM MQ sous-jacent à déployer, et non le nom de l'opérateur du gestionnaire de files d'attente.

Remarque : Si *QUEUE_MANAGER_NAME* n'est pas défini sur le nom du gestionnaire de files d'attente sous-jacent et qu'il est défini sur Operand, la surveillance ne fonctionnera pas. Le nom sous-jacent est défini dans `spec.queuemanager.name` pour l'opérande du gestionnaire de files d'attente.

5. Supprimez les pods instana-agent dans l'espace de nom instana-agent. Cela les entraîne à redémarrer et à commencer la surveillance avec les nouveaux paramètres.

Que faire ensuite

Vous êtes maintenant prêt à configurer le gestionnaire de files d'attente pour la IBM Instana surveillance.

OpenShift CP4I Operator2.2.0 *Surveillance d'Instana: configuration du gestionnaire de files d'attente*

Configurez un gestionnaire de files d'attente qui utilise TLS pour communiquer avec l'agent IBM Instana .L'authentification pour cette connexion est effectuée à l'aide de SSLPEERMAP.

Avant de commencer

Cette tâche suppose que vous avez configuré les agents pour la IBM Instana surveillance.

Procédure

1. Configurez le gestionnaire de files d'attente via MQSC et INI.

   MQSC est utilisé pour configurer un nouveau canal activé pour TLS, puis pour configurer ce canal afin d'authentifier l'agent IBM Instana de connexion s'il possède un certificat avec les zones requises. Dans ce cas, nous mappons tout client de connexion avec un certificat contenant les zones `CN=instana-agent,OU=app team1` à l'utilisateur `app1`. MQSC accorde ensuite à l'utilisateur `app1` le droit d'effectuer les opérations requises pour la surveillance de IBM Instana .

   Le fichier INI est utilisé pour accorder des droits à notre utilisateur externe `app1`.

   La mappe de configuration suivante contient les paramètres MQSC et INI requis. Déployez-le dans l'espace de nom de votre gestionnaire de files d'attente.

```
apiVersion: v1
data:
  channel.mqsc: |-
    DEFINE CHANNEL('INSTANA.A.SVRCONN') CHLTYPE(SVRCONN) SSLCAUTH(REQUIRED)
SSLCIPH('ANY_TLS12_OR_HIGHER')
```

```
    ALTER QMGR CONNAUTH(' ')
    REFRESH SECURITY
    SET CHLAUTH('INSTANA.A.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=*') USERSRC(NOACCESS)
ACTION(REPLACE)
    SET CHLAUTH('*') TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS) ACTION(REPLACE)
    SET CHLAUTH('INSTANA.A.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=instana-agent,OU=app
team1') USERSRC(MAP) MCAUSER('app1')
    SET AUTHREC PRINCIPAL('app1') OBJTYPE(QMGR) AUTHADD(ALL)
    SET AUTHREC PROFILE('SYSTEM.ADMIN.COMMAND.QUEUE') PRINCIPAL('app1') OBJTYPE(QUEUE)
AUTHADD(PUT,INQ,DSP,CHG)
    SET AUTHREC PROFILE('SYSTEM.**') PRINCIPAL('app1') OBJTYPE(TOPIC) AUTHADD(DSP)
    SET AUTHREC PROFILE('*') PRINCIPAL('app1') OBJTYPE(TOPIC) AUTHADD(DSP)
    SET AUTHREC PROFILE('SYSTEM.**') PRINCIPAL('app1') OBJTYPE(QUEUE) AUTHADD(DSP, CHG, GET)
    SET AUTHREC PROFILE('SYSTEM.**') PRINCIPAL('app1') OBJTYPE(LISTENER) AUTHADD(DSP)
    SET AUTHREC PROFILE('AMQ.*') PRINCIPAL('app1') OBJTYPE(QUEUE) AUTHADD(DSP, CHG)
    REFRESH SECURITY TYPE(CONNAUTH)
  auth.ini: |-
    Service:
      Name=AuthorizationService
      EntryPoints=14
      SecurityPolicy=UserExternal
kind: ConfigMap
metadata:
  namespace: your-queue-manager-namespace
  name: qmgr-monitoring-config
```

où *your-queue-manager-namespace* est l'espace de nom dans lequel votre gestionnaire de files d'attente sera déployé.

Remarque : Si vous surveillez des files d'attente définies par l'utilisateur, vous devez ajouter des lignes supplémentaires à la mappe de configuration MQSC, en accordant des droits DSP, CHG et GET à ces files d'attente. Exemple :

```
SET AUTHREC PROFILE('MYQUEUE') PRINCIPAL('app1') OBJTYPE(QUEUE) AUTHADD(DSP, CHG, GET).
```

Cet exemple utilise une mappe de configuration pour les données MQSC et INI, mais vous pouvez utiliser un secret si les ajouts que vous effectuez sont confidentiels. Pour des informations générales sur le déploiement avec MQSC et INI, voir «Exemple : fourniture de fichiers MQSC et INI», à la page 69.

2. Pour qu'une connexion TLS soit établie, le gestionnaire de files d'attente doit faire confiance au certificat de l'agent IBM Instana . Pour ce faire, créez un secret contenant uniquement le certificat de l'agent IBM Instana :

```
oc create secret generic instana-certificate-secret --from-file=./application.crt -n your-
queue-manager-namespace
```

3. Le gestionnaire de files d'attente doit présenter son propre certificat pour l'établissement de liaison TLS et requiert l'accès à la clé privée associée. Déployez un secret contenant la clé et le certificat que vous avez créés précédemment ou que vous possédez déjà:

```
oc create secret tls qm-tls-secret --cert server.crt --key server.key -n your-queue-manager-
namespace
```

Une fois la mappe de configuration et le secret créés, vous êtes prêt à créer le gestionnaire de files d'attente lui-même.

4. Vérifiez que le fichier YAML de votre gestionnaire de files d'attente ne définit pas la variable d'environnement **MQSNOAUT** dans le conteneur du gestionnaire de files d'attente.

   Sinon, une fois activé, le mécanisme d'authentification ne fonctionnera pas. La suppression de la variable après le déploiement n'entraîne pas la réactivation du mécanisme et le gestionnaire de files d'attente doit être recréé.

5. Ajoutez les sections suivantes à votre définition de gestionnaire de files d'attente, où *MYQM* est le nom de votre gestionnaire de files d'attente:

```
spec:
  queueManager:
    name: MYQM   #(a)
    ini:   #(b)
```

```
      - configMap:
          items:
            - auth.ini
          name: qmgr-monitoring-config
    mqsc:   #(c)
      - configMap:
          items:
            - channel.mqsc
          name: qmgr-monitoring-config
pki:
  keys:   #(d)
      - name: default
        secret:
          items:
          - tls.key
          - tls.crt
          secretName: qm-tls-secret
  trust:   #(e)
    - name: app
      secret:
        items:
          - application.crt
        secretName: instana-certificate-secret
```

Les sections marquées de la spécification sont décrites comme suit:

a. Vérifiez que vous avez donné un nom unique à votre gestionnaire de files d'attente sous-jacent. Si le gestionnaire de files d'attente sous-jacent n'a pas de nom unique, la surveillance risque de ne pas fonctionner comme prévu. Ce nom doit correspondre à celui de la mappe de configuration de l'agent IBM Instana qui a été éditée précédemment.

b. Les informations INI qui ont été écrites dans la mappe de configuration sont ajoutées au gestionnaire de files d'attente.

c. Les informations MQSC qui ont été écrites dans la mappe de configuration sont ajoutées au gestionnaire de files d'attente.

d. Le certificat du gestionnaire de files d'attente et la clé privée sont ajoutés au magasin de clés du gestionnaire de files d'attente.

e. Le certificat de l'agent IBM Instana est ajouté au magasin de clés de confiance du gestionnaire de files d'attente.

6. Facultatif : Activez la fonction de trace IBM Instana sur votre gestionnaire de files d'attente surveillées.

   Pour ce faire, voir «Intégration d' IBM MQ à la fonction de trace IBM Instana», à la page 90.

7. Déployez le gestionnaire de files d'attente.

Que faire ensuite

Vous êtes maintenant prêt à vérifier et déboguer la surveillance IBM Instana.

Surveillance Instana: Vérification et débogage

Pour pouvoir surveiller un gestionnaire de files d'attente via un agent IBM Instana , vous devez configurer à la fois l'agent et le gestionnaire de files d'attente.

Avant de commencer

Cette tâche suppose que vous avez configuré le gestionnaire de files d'attente pour la IBM Instana surveillance.

Procédure

Vérification

1. Pour vérifier que votre déploiement a abouti, affichez votre gestionnaire de files d'attente dans le tableau de bord IBM Instana .

   Le gestionnaire de files d'attente doit être visible dans la section Services de la page d'application, ainsi que dans la vue Infrastructure.

Débogage

Remarque : Ces étapes de débogage supposent un déploiement Openshift de l'agent IBM Instana exécuté en tant qu'objet daemonset.

Si vous ne voyez pas votre gestionnaire de files d'attente dans le tableau de bord IBM Instana , il se peut que vous ayez mal configuré votre gestionnaire de files d'attente. Procédez comme suit pour effectuer des recherches.

2. Identifiez le noeud sur lequel votre pod de gestionnaire de files d'attente actif s'exécute.

   Exécutez la commande suivante dans l'espace de nom de votre gestionnaire de files d'attente:

   ```
   oc get pods -o wide -n your-queue-manager-namespace
   ```

3. Pour déterminer quel pod d'agent IBM Instana est en cours d'exécution sur le même noeud que votre gestionnaire de files d'attente, exécutez la même commande dans l'espace de nom instana-agent:

   ```
   oc get pods -o wide -n instana-agent-namespace
   ```

4. Pour vous aider à comprendre les problèmes du côté de l'agent IBM Instana , obtenez les journaux du pod de l'agent IBM Instana et recherchez les entrées relatives à'mq'ou au nom de votre gestionnaire de files d'attente.

   Exécutez ensuite la commande suivante :

   ```
   oc logs instana-agent-pod -c instana-agent -n instana-agent
   ```

5. Consultez les journaux du gestionnaire de files d'attente.

   Si l'agent a tenté de se connecter au gestionnaire de files d'attente, les journaux du gestionnaire de files d'attente doivent indiquer la raison pour laquelle la connexion a échoué. Exécutez ensuite la commande suivante :

   ```
   oc logs your-queue-manager-name -n your-queue-manager-namespace
   ```

Résultats

Vous avez effectué les quatre tâches de configuration de la surveillance IBM Instana authentifiée avec TLS.

OpenShift CP4I Génération d'une image avec des fichiers MQSC et INI personnalisés, à l'aide de l'interface de ligne de commande Red Hat 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 n'avez pas de secret Red Hat OpenShift pour IBM Entitled Registry dans votre projet Red Hat OpenShift , suivez les étapes de la rubrique Création du secret de clé d'autorisation.

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:9.4.0.0-r1
      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:9.4.0.0-r1"
      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 (voir «Historique des éditions de IBM MQ Operator», à la page 5 pour plus de détails). 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 simple à l'aide de IBM MQ Operator», à la page 66 pour ajouter votre nouvelle image personnalisée dans le fichier YAML.

   Vous pouvez ajouter le fragment suivant de YAML dans votre YAML `QueueManager` normal, où *SingleNamespace* correspond au projet / espace de nom Red Hat OpenShift que vous utilisez, et *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 simple à l'aide de IBM MQ Operator», à la page 66
Cet exemple déploie un gestionnaire de files d'attente de démarrage rapide qui utilise un stockage éphémère (non persistant) et désactive la sécurité IBM MQ . Les messages ne sont pas conservés lors des redémarrages du gestionnaire de files d'attente. Vous pouvez ajuster la configuration afin de changer de nombreux paramètres du gestionnaire de files d'attente.

OpenShift CP4I Ajout d'annotations et d'étiquettes personnalisées aux ressources du gestionnaire de files d'attente

Vous ajoutez des annotations et des étiquettes personnalisées aux métadonnées QueueManager.

Pourquoi et quand exécuter cette tâche

Les annotations et les étiquettes personnalisées sont ajoutées à toutes les ressources, à l'exception des PVC. Si une annotation ou une étiquette personnalisée correspond à une clé existante, la valeur définie par IBM MQ Operator est utilisée.

Procédure

- Ajoutez des annotations personnalisées.

  Pour ajouter des annotations personnalisées aux ressources du gestionnaire de files d'attente, y compris le pod, ajoutez les annotations sous `metadata`. Exemple :

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
  annotations:
    annotationKey: "value"
```

- Ajoutez des étiquettes personnalisées.

  Pour ajouter des étiquettes personnalisées aux ressources du gestionnaire de files d'attente, y compris le pod, ajoutez les étiquettes sous `metadata`. Exemple :

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
  labels:
    labelKey: "value"
```

OpenShift CP4I Désactivation des vérifications des webhooks d'exécution

Les vérifications des webhooks d'exécution garantissent que les classes de stockage sont viables pour votre gestionnaire de files d'attente. Vous les désactivez pour améliorer les performances, ou parce qu'elles ne sont pas valides pour votre environnement.

Pourquoi et quand exécuter cette tâche

Les vérifications des webhooks d'exécution sont effectuées sur la configuration du gestionnaire de files d'attente. Elles garantissent que les classes de stockage conviennent au type du gestionnaire de files d'attente sélectionné.

Vous pouvez choisir de désactiver ces vérifications pour réduire le temps de création du gestionnaire de files d'attente ou parce que les vérifications ne sont pas valides pour votre environnement spécifique.

Remarque : Lorsque vous désactivez les vérifications des webhooks d'exécution, toutes les valeurs de classe de stockage sont autorisées. Cela peut entraîner une rupture de gestionnaire de files d'attente.

Procédure

- Désactivez les vérifications des webhooks d'exécution.

  Ajoutez l'annotation suivante sous `metadata`. Exemple :

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
  annotations:
    "com.ibm.cp4i/disable-webhook-runtime-checks" : "true"
```

OpenShift CP4I Operator2.1.0 Désactivation des mises à jour des valeurs par défaut de la spécification du gestionnaire de files d'attente

IBM MQ Operator met à jour les valeurs non spécifiées dans la spécification du gestionnaire de files d'attente avec leurs valeurs par défaut. Vous pouvez désactiver ce comportement si vous souhaitez éviter toute modification de la spécification du gestionnaire de files d'attente. Les zones de statut du gestionnaire de files d'attente sont toujours mises à jour.

Procédure

- Désactiver les mises à jour des valeurs par défaut du gestionnaire de files d'attente

  Ajoutez l'annotation suivante sous `metadata`. Exemple :

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
  annotations:
    "com.ibm.mq/write-defaults-spec" : "false"
```

  Remarque : Dans les exemples de démarrage rapide, cette annotation est appliquée par défaut.

Exécution du conteneur IBM MQ avec un système de fichiers racine en lecture seule

Vous pouvez configurer le conteneur IBM MQ pour qu'il s'exécute avec un système de fichiers racine en lecture seule. Cela empêche les agresseurs de copier et d'exécuter du code malveillant dans le conteneur.

Pourquoi et quand exécuter cette tâche

L'activation du système de fichiers racine en lecture seule rend les fichiers de conteneur non modifiables. Autrement dit, sur le système de fichiers de conteneur, les fichiers peuvent être affichés mais pas modifiés et aucun nouveau fichier ne peut être créé. Les fichiers ne peuvent être modifiés ou créés que sur un système de fichiers monté.

Lorsqu'un système de fichiers racine en lecture seule est activé, deux volumes temporaires `Scratch` et `Tmp` sont créés et montés respectivement dans les répertoires `/run` et `/tmp` du conteneur.

- Le volume `Scratch` contient les fichiers, les magasins de clés et les autres fichiers utilisés pour configurer le gestionnaire de files d'attente.
- Le volume `Tmp` contient des fichiers de diagnostic, par exemple les fichiers RAS du gestionnaire de files d'attente.

Etant donné que ces volumes sont temporaires, les fichiers de ces volumes sont perdus lors du redémarrage du pod.

Le type du volume créé pour les données du gestionnaire de files d'attente dépend du type de stockage. Par défaut, un volume persistant est monté. Ou bien, si le type de stockage est `éphémère`, un volume éphémère est monté. Si la taille des données du volume dépasse la valeur spécifiée pour la propriété **`sizeLimit`** , Kubernetes peut éjecter le conteneur et en créer un nouveau.

Un système de fichiers racine en lecture seule n'est pas activé par défaut. Pour l'activer, procédez comme suit:

Procédure

1. Utilisez l'API `spec.securityContext` pour activer le système de fichiers racine en lecture seule.

   Pour votre gestionnaire de files d'attente, définissez la propriété **`readOnlyRootFilesystem`** dans «.spec.securityContext», à la page 158 sur `true`.

   IBM MQ Operator crée deux volumes éphémères, `Scratch` et `Tmp`.
2. Facultatif : Définissez ou modifiez le type de stockage des données du gestionnaire de files d'attente.

   Par défaut, une réservation de volume persistant est montée dans `/mnt/mqm`. Ou bien, si la propriété **`type`** est définie sur `ephemeral` dans «.spec.queueManager.storage.queueManager», à la page 156, un volume éphémère est créé et monté.
3. Pour chaque volume éphémère, déterminez avec attention la croissance des données. Définissez la valeur de la propriété **`sizeLimit`** en conséquence, y compris les unités SI.
   - Pour le volume éphémère `Scratch` , définissez la propriété **`sizeLimit`** dans «.spec.queueManager.storage.scratch», à la page 157. La valeur par défaut est "100M".
   - Pour le volume éphémère `Tmp` , définissez la propriété **`sizeLimit`** dans «.spec.queueManager.storage.tmp», à la page 158. La valeur par défaut est "2Gi".
   - Si le **`type`** du volume de gestionnaire de files d'attente est défini sur `ephemeral`, définissez la propriété **`sizeLimit`** dans «.spec.queueManager.storage.queueManager», à la page 156. La valeur par défaut est "2Gi".

OpenShift V 9.4.0 Configuration de IBM MQ Console avec un registre de base à l'aide de IBM MQ Operator

Pour vous connecter à IBM MQ Console, vous pouvez fournir votre propre configuration au gestionnaire de files d'attente.

Avant de commencer

Si vous déployez un gestionnaire de files d'attente avec une licence IBM MQ Advanced for Developers , une configuration simple est intégrée. Voir «Exemple de fichier YAML de gestionnaire de files d'attente qui décrit comment spécifier des mots de passe pour les utilisateurs admin et app», à la page 176. Si vous

déployez un gestionnaire de files d'attente de licence IBM Cloud Pak for Integration , vous pouvez activer l'intégration à IBM Cloud Pak for Integration Keycloak pour vous connecter à IBM MQ Console à l'aide de la connexion unique. Voir «Connexion à IBM MQ Console déployée dans un cluster Red Hat OpenShift», à la page 134.

Procédure

1. **Créez un mot de passe et chiffrez-le à l'aide de `securityUtility`.**

   Un `ConfigMap` est utilisé pour stocker les données d'identification que vous utilisez pour accéder à votre gestionnaire de files d'attente. Pour améliorer la sécurité, vous codez ces données d'identification à l'aide de la commande securityUtility.

   Vous pouvez également utiliser un secret, qui protège les données d'identification dans la couche Kubernetes . Toutefois, les outils de surveillance ou de traitement des incidents peuvent exposer le fichier sous-jacent de manière non sécurisée.
2. Facultatif : **Connectez-vous à l'interface de ligne de commande Red Hat OpenShift .**

   Si vous utilisez l'interface de ligne de commande OpenShift , connectez-vous à l'aide de `oc login`.

   Vous pouvez également utiliser la console OpenShift .
3. **Créez un `ConfigMap` avec votre configuration.**

   Pour obtenir de l'aide sur la création de la configuration XML, voir SécuritéIBM MQ Console et REST API.

   L'exemple suivant crée un utilisateur dans le groupe `MQWebAdminGroup`. Le rôle `MQWebAdmin` est affecté aux membres du `MQWebAdminGroup` . Dans cet exemple :

   - Vous **devez** remplacer *USERNAME* et *PASSWORD* par vos propres valeurs. Notez que *USERNAME* est utilisé deux fois dans l'exemple.

     Vous **devez** spécifier l'espace de nom *NAMESPACE* comme celui dans lequel votre IBM MQ Operator est déployé et où votre gestionnaire de files d'attente sera ou est déjà déployé.

   a) Utilisez la console OpenShift ou la ligne de commande pour créer les `ConfigMap`suivantes:

   ```
   kind: ConfigMap
   apiVersion: v1
   metadata:
     name: mqwebuserconfigmap
     namespace: NAMESPACE
   data:
     mqwebuser.xml: |
       <?xml version="1.0" encoding="UTF-8"?>
       <server>
         <featureManager>
           <feature>appSecurity-2.0</feature>
           <feature>basicAuthenticationMQ-1.0</feature>
         </featureManager>
         <enterpriseApplication id="com.ibm.mq.console">
           <application-bnd>
             <security-role name="MQWebAdmin">
               <group name="MQWebAdminGroup" realm="defaultRealm"/>
             </security-role>
           </application-bnd>
         </enterpriseApplication>
         <basicRegistry id="basic" realm="defaultRealm">
           <user name="USERNAME" password="PASSWORD"/>
           <group name="MQWebAdminGroup">
             <member name="USERNAME"/>
           </group>
         </basicRegistry>
         <sslDefault sslRef="mqDefaultSSLConfig"/>
       </server>
   ```

   b) Facultatif : Si vous utilisez la ligne de commande, appliquez le `ConfigMap`:

   ```
   oc apply -f mqwebuserconfigmap.yaml
   ```

   Pour les étapes restantes, choisissez l'une des options suivantes:

- Déployez un nouveau gestionnaire de files d'attente avec la configuration pour accéder à IBM MQ Console.
- Appliquez la configuration qui permet à IBM MQ Console d'accéder à un gestionnaire de files d'attente existant.

4. Facultatif : **Déployez un nouveau gestionnaire de files d'attente avec la configuration pour accéder au IBM MQ Console.**
   a) Créez votre gestionnaire de files d'attente.

      Définissez les fournisseurs d'authentification et d'autorisation sur `manual` et indiquez le `ConfigMap` `mqwebuserconfigmap` nouvellement créé à l'aide de l'une des options suivantes:

      - Option 1: via le fichier YAML du gestionnaire de files d'attente

        Ajoutez le code suivant dans la section web du fichier YAML du gestionnaire de files d'attente:

        ```
        ...
        web:
            enabled: true
            console:
              authentication:
                provider: manual
              authorization:
                provider: manual
            manualConfig:
              configMap:
                name: mqwebuserconfigmap
        ```

      - Option 2: via la vue de fiche de la console OpenShift :

        i) Sur la console OpenShift , sélectionnez **Operators** > **Installed Operators**.

        ii) Sélectionnez votre déploiement du IBM MQ Operator.

        iii) Sélectionnez **Gestionnaire de files d'attente** et cliquez sur **Créer QueueManager**.

        iv) Sélectionnez les options appropriées pour votre gestionnaire de files d'attente.

        v) Sélectionnez **Web** et définissez **Activer le serveur Web** sur `true`.

        vi) Ouvrez la zone de liste **Configuration avancée** .

        vii) Sous la zone de liste **Console** , définissez **fournisseur** pour **Authentification** et **Autorisation** sur `manuel`.

        viii) Ouvrez la zone de liste **Configuration** .

        ix) Ouvrez la zone de liste **ConfigMap** et sélectionnez l'objet ConfigMap `mqwebuserconfigmap` créé à l'étape «3», à la page 102.

        x) Cliquez sur **Créer**.

      Vous pouvez maintenant accéder à la IBM MQ Console de votre nouveau gestionnaire de files d'attente via les données d'identification spécifiées dans le `ConfigMap` créé à l'étape «3», à la page 102.

5. Facultatif : **Application de la configuration qui active IBM MQ Console pour un gestionnaire de files d'attente existant.**

   Editez le fichier YAML du gestionnaire de files d'attente pour lequel vous activez IBM MQ Console:

   a. Sur la console OpenShift , sélectionnez **Operators** > **Installed Operators**.

   b. Sélectionnez votre déploiement du IBM MQ Operator.

   c. Sélectionnez **Queue Mananger** et sélectionnez le nom de votre gestionnaire de files d'attente.

   d. Sélectionnez **YAML**.

   e. Remplacez la section web existante du fichier YAML du gestionnaire de files d'attente par le code suivant:

   ```
   ...
   web:
       enabled: true
   ```

```
    console:
      authentication:
        provider: manual
      authorization:
        provider: manual
    manualConfig:
      configMap:
        name: mqwebuserconfigmap
```

f. Cliquez sur **Sauvegarder**.

Vous pouvez maintenant accéder à la IBM MQ Console de votre gestionnaire de files d'attente existant via les données d'identification spécifiées dans le `ConfigMap` créé à l'étape «3», à la page 102.

OpenShift V 9.4.0 V 9.4.0 Extension des volumes persistants

Si votre fournisseur de stockage prend en charge l'extension de volume, utilisez cette tâche pour étendre un volume persistant. Selon le fournisseur de stockage, l'extension peut se produire en ligne ou hors ligne.

Avant de commencer

La réussite de l'extension de volume dépend de votre fournisseur de stockage pour répondre à la demande d'extension. Consultez la documentation de vos fournisseurs de stockage pour déterminer si le redimensionnement en ligne est pris en charge et pour plus d'informations sur les procédures de redimensionnement hors ligne.

Si votre fournisseur de stockage ne peut pas répondre à la demande d'extension, votre demande de volume persistant peut entrer dans un état avec des avertissements ou des erreurs. En cas d'échec de l'extension, un administrateur OpenShift peut récupérer manuellement l'état de la réservation de volume persistant et annuler l'extension. Voir Recovering from failure when expansion volumes dans la documentation Red Hat OpenShift .

Pourquoi et quand exécuter cette tâche

Pour vous aider à gérer le stockage persistant, Kubernetes définit deux ressources d'API:

- Un PersistentVolume (PV), qui est un élément de stockage dans le cluster qui a été mis à disposition par un administrateur ou dynamiquement mis à disposition à l'aide de classes de stockage. Il peut être mis à disposition de manière statique ou dynamique.
- Une réservation de volume persistant (PVC) PersistentVolume, qui est une demande de stockage par un utilisateur. Il agit également comme un contrôle de réclamation de la ressource.

Pour plus d'informations, voir Persistent Volumes dans la documentation Kubernetes .

Avertissement :

- Si la classe de stockage utilisée pour créer des PVC de gestionnaire de files d'attente ne prend pas en charge le redimensionnement en ligne, le redimensionnement hors ligne est effectué. Lors du redimensionnement hors ligne, une intervention de l'utilisateur est requise pour terminer l'extension de volume, de sorte que les gestionnaires de files d'attente sont confrontés à un temps d'indisponibilité.
- Pour le redimensionnement hors ligne des volumes partagés pour les gestionnaires de files d'attente multi-instance, les pods actifs et de secours doivent être arrêtés en même temps lors de l'intervention de l'utilisateur.
- OpenShift ne prend pas en charge la réduction de la taille des réservations de volume persistant. Si vous tentez de réduire la taille des volumes persistants, le gestionnaire de files d'attente passe à l'état'Echec'.
- Cette procédure ne s'applique pas aux volumes temporaires.

Pour développer un volume persistant utilisé par le conteneur IBM MQ , procédez comme suit.

Procédure

1. **Préparation de l'extension des volumes**
   a) Choisissez les volumes à développer.
   b) Déterminez la ou les classes de stockage utilisées par vos volumes.
      Exemple :

```
spec:
  queueManager:
    storage:
      persistedData:
        enabled: true
        type: persistent-claim
        class: ocs-storagecluster-cephfs (1)
      queueManager:
        type: persistent-claim
      recoveryLogs:
        enabled: true
        type: persistent-claim
      defaultClass: ocs-storagecluster-ceph-rbd (2)
```

      Remarques :

      - (1) Si le volume définit une classe de stockage spécifique, elle est utilisée par les PVC de ce type.
      - (2) Si **defaultClass** est défini, cette classe de stockage est utilisée pour tous les volumes sans classe de stockage spécifique. Si **defaultClass** n'est pas défini et qu'un type de volume n'a pas spécifié de classe, la classe de stockage par défaut du cluster est utilisée.

      Vous pouvez également confirmer la classe de stockage utilisée en décrivant les réservations de volume persistant sous-jacentes. Exemple :

```
oc describe pvc pvc-name
```

   c) Vérifiez que votre classe de stockage prend en charge l'extension de volume.

      Une classe de stockage peut avoir la propriété **.allowVolumeExpansion** définie:

      - Si cette propriété est définie sur `true`, l'extension de volume est prise en charge.
      - Si cette propriété est définie sur `false`ou si elle n'est pas définie, la classe de stockage n'autorise pas l'extension de volume. Dans ce cas, consultez la documentation de votre fournisseur de stockage pour voir si cette fonction peut être activée.

      Vous pouvez également décrire une classe de stockage pour déterminer si elle prend en charge l'extension de volume. Exemple :

```
oc describe sc storage-class-name
```

   d) Consultez la documentation de votre fournisseur de stockage pour savoir si une procédure en ligne ou hors ligne est utilisée pour l'extension de volume.

      Une procédure hors ligne requiert le redémarrage manuel des pods de gestionnaire de files d'attente, contrairement à une procédure en ligne. Consultez la documentation de votre fournisseur de stockage pour connaître les procédures de redimensionnement hors ligne.

   e) Vérifiez si votre gestionnaire de files d'attente a une condition de statut avec le motif'StorageMismatch'.

      Si votre gestionnaire de files d'attente possède cette condition de statut, les volumes répertoriés dans la condition sont étendus si vous activez l'extension de volume. Si vous ne souhaitez pas que cela se produise, modifiez les zones de taille associées à chaque type de volume dans votre définition de gestionnaire de files d'attente pour qu'elles correspondent aux réservations de volume persistant mises à disposition. La condition de statut est supprimée lorsque cette opération est effectuée pour tous les volumes non concordants.

2. **Développement de volumes**

Avertissement :

- Si vous avez précédemment modifié l'une des zones de taille de volume dans la définition de votre gestionnaire de files d'attente, les volumes commencent à se développer lorsque **`.allowVolumeExpansion`** est défini sur `true` dans la définition de votre gestionnaire de files d'attente.
- Votre fournisseur de stockage peut avoir des restrictions sur la taille maximale d'un volume en raison des limitations du système de fichiers ou de la disponibilité du matériel local. Pour éviter les échecs, validez ces limitations dans la documentation de votre fournisseur de stockage avant d'étendre les volumes.
- Les réductions de la taille de la réservation de volume persistant ne sont pas prises en charge par OpenShift. Si vous augmentez la taille d'un volume, vous ne pouvez pas le réduire. Si votre tentative échoue, le IBM MQ Operator ne peut pas renvoyer la réservation de volume persistant à son état d'origine.

Exemple de définition de gestionnaire de files d'attente illustrant l'extension de volume:

```
spec:
  queueManager:
    storage:
      allowVolumeExpansion: true (A)
      persistedData:
        enabled: true
        type: persistent-claim
        size: 3Gi (B)
      queueManager:
        type: persistent-claim
        size: 4Gi (B)
      recoveryLogs:
        enabled: true
        type: persistent-claim
        size: 3Gi (B)
```

a) Pour autoriser l'extension de volume pour le gestionnaire de files d'attente, définissez la zone **`.spec.queueManager.storage.allowVolumeExpansion`** (A) sur votre gestionnaire de files d'attente sur `true`.

b) Vous pouvez désormais augmenter les zones de taille (B) pour n'importe quel type de volume activé. L'application de ces modifications démarrera l'extension de volume.

3. **Vérifiez que vos réservations de volume persistant ont été redimensionnées.**

Remarques :

- L'extension du volume peut prendre un certain temps. Si la validation n'aboutit pas, la première fois, pensez à attendre quelques minutes et à effectuer une nouvelle validation.
- L'extension de volume se termine uniquement sans action de l'utilisateur lorsqu'un redimensionnement en ligne est effectué.
- Certains fournisseurs de stockage arrondissent la taille de stockage que vous avez demandée. La taille du volume étendu doit être identique ou supérieure à celle de votre demande.

a) Vérifiez les conditions de statut de votre gestionnaire de files d'attente. Consultez le tableau suivant pour connaître les conditions, les explications et les actions suggérées.

Tableau 1. Conditions de statut de stockage

Condition	message	Explication
`StorageMismatch`	`Storage sizes defined in the QueueManager resource do not match the capacity of one or more provisioned PVCs [pvc-list]. AllowVolumeExpansion is set to false in the QueueManager resource so the MQ Operator will not attempt to reconcile these differences.`	L'extension de volume n'a pas lieu car **`.allowVolumeExpansion`** n'a pas été défini sur `true` dans la définition de gestionnaire de files d'attente.
`StorageExpansionPending`	`Volume expansion is pending for the following PVCs [pvc-list]`	L'expansion du volume est toujours en cours. Si cette condition d'état persiste pendant une période prolongée, suivez les étapes ci-dessous pour collecter des informations supplémentaires car un redimensionnement hors ligne ou un échec de redimensionnement peut se produire.
`Failed`	Il existe de nombreux messages liés au stockage qui peuvent créer une condition de statut `'Failed'`. Par exemple : `'MQ Queue Manager failed to deploy: persistentvolumeclaims "<pvc>" is forbidden: only dynamically provisioned pvc can be resized and the storageclass the provisions the pvc must support resize.'`	Si le gestionnaire de files d'attente a des conditions de statut `'Failed'` avec du texte qui fait référence au stockage, reportez-vous au message dans la condition de statut. L'exemple de message présenté ici est généré par l'utilisation d'une classe de stockage qui ne prend pas en charge l'extension.

b) Pour chaque réservation de volume persistant que vous avez étendue, vérifiez que la capacité a augmenté pour correspondre ou être supérieure à la valeur spécifiée dans la définition du gestionnaire de files d'attente.

Les gestionnaires de files d'attente à haute disponibilité peuvent avoir plusieurs PVC de chaque type. Pour obtenir la capacité d'une réservation de volume persistant, exécutez la commande suivante:

```
oc get pvc pvc-name -o template --template '{{.status.capacity.storage}}'
```

c) Vérifiez que la réservation de volume persistant ne comporte pas de conditions de statut ou d'événements suggérant un échec de redimensionnement:

```
oc describe pvc pvc-name
```

- Votre réservation de volume persistant peut avoir la condition de statut `FileSystemResizePending` avec le message'Waiting for user to (re-) start a pod to finish file system resize of volume on node'. Cette condition de statut est déclenchée pour les redimensionnements en ligne et hors ligne. Pour un redimensionnement en ligne, cette condition de statut disparaît sans intervention de l'utilisateur une fois le redimensionnement en ligne terminé.
- Si votre PVC a un événement ou une condition de statut qui indique un échec de redimensionnement, voir Recovering from failure when expansion volumes dans la documentation Red Hat OpenShift .

d) Vérifiez que les pods de gestionnaire de files d'attente ne comportent pas de conditions de statut ou d'événements suggérant un échec de redimensionnement. Pour les déploiements à haute disponibilité, vérifiez chaque réplique.

```
oc describe pod queue-manager-pod-name
```

- Si votre pod comporte un événement ou une condition de statut indiquant un échec de redimensionnement, voir Recovering from failure when expansion volumes dans la documentation Red Hat OpenShift . Le texte de l'erreur peut vous aider à résoudre le problème ou à éviter que le même problème ne se produise si vous essayez de le redimensionner à nouveau après la reprise.

4. **Redémarrer les pods lors du redimensionnement hors ligne**

   Si votre fournisseur de stockage utilise une procédure de redimensionnement hors ligne lors de l'extension de volumes, pour que l'extension de volume soit terminée, vous devez redémarrer les pods de gestionnaire de files d'attente qui montent les volumes en cours de redimensionnement.

   Pour les gestionnaires de files d'attente multi-instance, les journaux de reprise et les volumes de données conservés sont partagés entre les pods actifs et de secours. Pour que le redimensionnement de ces volumes soit terminé, vous pouvez arrêter les deux pods en même temps.

   Consultez la documentation de votre fournisseur de stockage pour connaître la procédure de redimensionnement hors ligne.

OpenShift V 9.4.0 V 9.4.0 Arrêt d'un gestionnaire de files d'attente (`mq.ibm.com/stop`)

Arrêtez un gestionnaire de files d'attente en ajoutant une annotation à la définition de gestionnaire de files d'attente.

Pourquoi et quand exécuter cette tâche

Les gestionnaires de files d'attente créés par l'opérateur IBM MQ sont associés à un `StatefulSet`. Ce `StatefulSet` déclare le nombre de `Pods` à déployer pour un type de disponibilité de gestionnaire de files d'attente donné via la zone'`.replicas`'. Prend la valeur 1 (instance unique), 2 (instance multiple) ou 3 (NativeHA).

Remarque : La modification manuelle de la valeur dans la zone'`.replicas`'empêche le gestionnaire de files d'attente de fonctionner correctement.

Dans certains cas, vous souhaiterez peut-être arrêter votre gestionnaire de files d'attente de sorte que le `StatefulSet` ait un nombre de répliques égal à 0 et qu'aucun `Pods` ne soit déployé. Vous pouvez, par exemple, effectuer cette opération lors d'une procédure de maintenance ou de sauvegarde.

Remarque : Etant donné qu'aucun gestionnaire de files d'attente `Pods` n'est déployé lorsque le gestionnaire de files d'attente est arrêté, vous et vos applications ne pourrez pas accéder au gestionnaire de files d'attente tant qu'il ne sera pas redémarré.

Procédure

- Pour arrêter votre gestionnaire de files d'attente, ajoutez l'annotation suivante à la définition de gestionnaire de files d'attente sous la section'`.metadata.annotations`'.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: my-qm
  annotations:
    "mq.ibm.com/stop" : "true"
```

- Pour redémarrer le gestionnaire de files d'attente et le renvoyer à son nombre correct de répliques, supprimez l'annotation du gestionnaire de files d'attente ou définissez sa valeur sur'`false`'.

Déploiement et configuration de gestionnaires de files d'attente à l'aide de Helm

Vous pouvez déployer et configurer un gestionnaire de files d'attente sur Kubernetes à l'aide de l'exemple de charte Helm .

Pourquoi et quand exécuter cette tâche

Si vous n'utilisez pas Red Hat OpenShift Container Platform, IBM MQ Operator n'est pas pris en charge. Vous pouvez utiliser l'exemple de charte Helm pour effectuer un déploiement sur d'autres types de clusters Kubernetes .

Procédure

- Pour plus d'informations sur l'utilisation de Helm pour déployer votre propre image de conteneur IBM MQ , voir Exemple de charte IBM MQ Helm

Référence associée

«Prise en charge de IBM MQ dans les conteneurs», à la page 8

Toutes les fonctions IBM MQ ne sont pas disponibles et prises en charge de la même manière dans les conteneurs.

OpenShift CD CP4I-SC2 Migration vers IBM MQ Operator

Cet ensemble de rubriques décrit les principales étapes de la migration d'un gestionnaire de files d'attente IBM MQ existant vers un environnement de conteneur à l'aide de IBM MQ Operator dans Red Hat OpenShift Container Platform.

Pourquoi et quand exécuter cette tâche

Les clients qui déploient IBM MQ sous Red Hat OpenShift peuvent être séparés dans les scénarios suivants :

1. Création d'un déploiement IBM MQ dans Red Hat OpenShift pour les nouvelles applications.
2. Extension d'un réseau IBM MQ dans Red Hat OpenShift pour les nouvelles applications dans Red Hat OpenShift.
3. Transfert d'un déploiement IBM MQ dans Red Hat OpenShift pour continuer à prendre en charge les applications existantes.

Seul le scénario 3 requiert que vous migriez votre configuration IBM MQ. Les autres scénarios sont considérés comme de nouveaux déploiements.

Cet ensemble de rubriques se concentre sur le scénario 3 et décrit les principales étapes de la migration d'un gestionnaire de files d'attente IBM MQ existant dans un environnement de conteneur à l'aide de IBM MQ Operator. En raison de la flexibilité et de l'utilisation intensive d'IBM MQ, il existe plusieurs étapes

facultatives. Chacune d'elles inclut une section "Cette tâche est-elle obligatoire ?". Identifiez vos besoins pour gagner du temps lors de la migration.

Vous devez également déterminer les données à migrer :

1. Migrer IBM MQ avec la même configuration, mais sans les messages de file d'attente existants.
2. Migrer IBM MQ avec la même configuration et les messages existants.

Une migration type de version à version peut utiliser l'une ou l'autre de ces approches. Dans un gestionnaire de files d'attente IBM MQ type au point de migration, seuls quelques messages sont éventuellement stockés dans des files d'attente, ce qui rend l'option 1 appropriée dans de nombreux cas. Dans le cas d'une migration vers une plateforme de conteneur, il est encore plus courant d'utiliser l'option 1 pour réduire la complexité de la migration et permettre un déploiement Blue Green. Par conséquent, les instructions portent sur ce scénario.

Ce scénario a pour objectif de créer un gestionnaire de files d'attente dans l'environnement de conteneur qui correspond à la définition du gestionnaire de files d'attente existant. Ainsi, les applications existantes connectées au réseau peuvent simplement être reconfigurées pour pointer vers le nouveau gestionnaire de files d'attente, sans qu'il ne soit nécessaire de modifier le reste de la configuration ou la logique d'application.

Tout au long de cette migration, vous générez plusieurs fichiers de configuration à appliquer au nouveau gestionnaire de files d'attente. Pour simplifier la gestion de ces fichiers, vous devez créer un répertoire et les générer dans ce répertoire.

Procédure

1. «Vérification de la disponibilité des fonctions requises», à la page 110
2. «Extraction de la configuration du gestionnaire de files d'attente», à la page 111
3. Facultatif : «Facultatif : extraction et acquisition des clés et certificats du gestionnaire de files d'attente», à la page 112
4. Facultatif : «Facultatif : configuration de LDAP», à la page 114
5. Facultatif : «Facultatif : modification des adresses IP et noms d'hôte dans la configuration IBM MQ», à la page 122
6. «Mise à jour de la configuration du gestionnaire de files d'attente pour un environnement de conteneur», à la page 123
7. «Sélection de l'architecture haute disponibilité cible pour IBM MQ exécuté en conteneurs», à la page 127
8. «Création des ressources du gestionnaire de files d'attente», à la page 127
9. «Création du gestionnaire de files d'attente dans Red Hat OpenShift», à la page 129
10. «Vérification du nouveau déploiement de conteneur», à la page 133

OpenShift CD CP4I-SC2 Vérification de la disponibilité des fonctions requises

Le IBM MQ Operator n'inclut pas toutes les fonctionnalités disponibles dans IBM MQ Advanced et vous devez vérifier que ces dernières ne sont pas requises. D'autres fonctionnalités sont partiellement prises en charge et peuvent être reconfigurées conformément à celles disponibles dans le conteneur.

Avant de commencer

Il s'agit de la première étape de la rubrique «Migration vers IBM MQ Operator», à la page 109.

Procédure

1. Vérifiez que l'image du conteneur cible inclut toutes les fonctions requises.

Pour les informations les plus récentes, reportez-vous à la rubrique «Comment utiliser IBM MQ dans des conteneurs», à la page 8.

2. Le IBM MQ Operator possède un unique port de trafic IBM MQ, appelé programme d'écoute. Si vous disposez de plusieurs programmes d'écoute, procéder à une simplification pour n'utiliser qu'un seul programme d'écoute dans le conteneur. Comme il ne s'agit pas d'un scénario courant, cette modification n'est pas décrite en détail.
3. Si des exits IBM MQ sont utilisés, migrez-les dans le conteneur en les superposant dans les fichiers binaires d'exit de IBM MQ. Il s'agit d'un scénario de migration avancé, qui n'est donc pas inclus ici. Pour un aperçu des étapes, reportez-vous à la rubrique «Génération d'une image avec des fichiers MQSC et INI personnalisés, à l'aide de l'interface de ligne de commande Red Hat OpenShift», à la page 97.
4. Si votre système IBM MQ inclut la haute disponibilité, vérifiez les options disponibles.

   Voir «Planification de la haute disponibilité pour IBM MQ dans des conteneurs», à la page 19.

Que faire ensuite

Vous êtes maintenant prêt à extraire la configuration du gestionnaire de files d'attente.

OpenShift CD CP4I-SC2 Extraction de la configuration du gestionnaire de files d'attente

La majorité de la configuration est transférable entre les gestionnaires de files d'attente, notamment les éléments avec lesquels les applications interagissent, comme les définitions des files d'attente, des rubriques et des canaux. Utilisez cette tâche pour extraire la configuration du gestionnaire de files d'attente IBM MQ existant.

Avant de commencer

Cette tâche suppose que vous avez vérifié que les fonctions requises sont disponibles.

Procédure

1. Connectez-vous à la machine sur laquelle IBM MQ est installé.
2. Sauvegardez la configuration.

   Exécutez ensuite la commande suivante :

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

   Remarques sur l'utilisation de cette commande :

   - Cette commande stocke la sauvegarde dans le répertoire `tmp`. Vous pouvez la stocker dans un autre emplacement, mais le présent scénario utilise le répertoire `tmp` pour les commandes suivantes.
   - Remplacez *QMGR_NAME* par le nom du gestionnaire de files d'attente de votre environnement. Si vous n'êtes pas sûr de la valeur, exécutez la commande **dspmq** pour afficher les gestionnaires de files d'attente disponibles sur la machine. Vous trouverez ci-dessous un exemple de sortie de commande **dspmq** pour un gestionnaire de files d'attente nommé `qm1` :

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

     La commande **dspmq** requiert que le gestionnaire de files d'attente IBM MQ soit démarré, faute de quoi l'erreur suivante est générée :

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

     Si nécessaire, démarrez le gestionnaire de files d'attente à l'aide de la commande suivante :

```
strmqm QMGR_NAME
```

Que faire ensuite

Vous êtes maintenant prêt à extraire et acquérir les clés et certificats du gestionnaire de files d'attente.

OpenShift CD CP4I-SC2 Facultatif : extraction et acquisition des clés et certificats du gestionnaire de files d'attente

IBM MQ peut être configuré pour chiffrer le trafic réseau dans le gestionnaire de files d'attente avec TLS. Cette tâche permet de vérifier que votre gestionnaire de files d'attente utilise TLS, d'extraire des clés et des certificats et de configurer TLS sur le gestionnaire de files d'attente migré.

Avant de commencer

Cette tâche suppose que vous avez extrait la configuration du gestionnaire de files d'attente.

Pourquoi et quand exécuter cette tâche

Cette tâche est-elle obligatoire ?

IBM MQ peut être configuré pour chiffrer le trafic dans le gestionnaire de files d'attente. Ce chiffrement est effectué à l'aide d'un référentiel de clés configuré sur le gestionnaire de files d'attente. Les canaux IBM MQ permettent ensuite les communications TLS. Si vous ne savez pas si la communication TLS est configurée dans votre environnement, exécutez la commande suivante pour vérifier:

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

Si aucun résultat n'est trouvé, TLS n'est pas utilisé. Toutefois, cela ne signifie pas que TLS ne doit pas être configuré dans le gestionnaire de files d'attente migré. Il existe plusieurs raisons pour lesquelles vous pouvez modifier ce comportement :

- L'approche de sécurité de l'environnement Red Hat OpenShift doit être améliorée par rapport à l'environnement précédent.
- Si vous devez accéder au gestionnaire de files d'attente migré à partir de l'extérieur de l'environnement Red Hat OpenShift, TLS doit passer par la route Red Hat OpenShift.

Remarque : Les certificats du gestionnaire de files d'attente ayant le même nom distinctif (DN) de sujet que le certificat de l'émetteur (CA) ne sont pas pris en charge. Un certificat doit avoir un nom distinctif de sujet unique. Le produit vérifie que les noms distinctifs ne sont pas identiques.

Procédure

1. Extrayez les certificats sécurisés du magasin existant.

   Si TLS est actuellement utilisé dans le gestionnaire de files d'attente, un certain nombre de certificats sécurisés sont sans doute stockés pour ce dernier. Ils doivent être extraits et copiés dans le nouveau gestionnaire de files d'attente. Effectuez l'une des étapes facultatives suivantes :

   - Pour rationaliser l'extraction des certificats, exécutez le script suivant sur le système local :

```
#!/bin/bash

keyr=$(grep SSLKEYR $1)
if [ -n "${keyr}" ]; then
   keyrlocation=$(sed -n "s/^.*'\(.*\)'.*$/\1/ p" <<< ${keyr})
   mapfile -t runmqakmResult < <(runmqakm -cert -list -db ${keyrlocation}.kdb -stashed)
   cert=1
   for i in "${runmqakmResult[@]:2}"
   do
      certlabel=$(echo ${i:2} | xargs)
      echo Extracting certificate $certlabel to $cert.cert
```

```
      runmqakm -cert -extract -db ${keyrlocation}.kdb -label "$certlabel" -target $
{cert}.cert -stashed
      cert=$[$cert+1]
   done
fi
```

Lors de l'exécution de ce script, spécifiez l'emplacement de la sauvegarde IBM MQ comme argument et les certificats sont extraits. Par exemple, si le script s'intitule `extractCert.sh` et que la sauvegarde IBM MQ se trouve dans `/tmp/backup.mqsc`, exécutez la commande suivante :

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

- Vous pouvez également exécuter les commandes suivantes suivant l'ordre indiqué :

  a. Identifiez l'emplacement du référentiel de clés TLS du gestionnaire de files d'attente:

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

  Exemple de sortie :

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

  Où le magasin de clés se trouve dans `/run/runmqserver/tls/key.kdb`

  b. En fonction de ces informations d'emplacement, interrogez le magasin de clés pour déterminer les certificats stockés :

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

  Exemple de sortie :

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

  c. Extrayez chacun des certificats répertoriés, à l'aide de la commande suivante :

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

  Dans les exemples précédents, cela correspond aux commandes suivantes:

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

2. Procurez-vous une nouvelle clé et un nouveau certificat pour le gestionnaire de files d'attente.

   Pour configurer TLS sur le gestionnaire de files d'attente migré, générez une nouvelle clé et un nouveau certificat. Ces derniers seront utilisés lors du déploiement. Dans de nombreuses organisations, cela implique de contacter votre équipe de sécurité pour demander une clé et un certificat. Dans certaines organisations, cette option n'est pas disponible et des certificats autosignés sont utilisés.

   L'exemple suivant génère un certificat autosigné dont le délai d'expiration est défini sur 10 ans :

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

   Deux fichiers sont créés :

   - `qmgr.key` est la clé privée du gestionnaire de files d'attente

- `qmgr.crt` est le certificat public

Que faire ensuite

Vous êtes maintenant prêt à configurer LDAP.

OpenShift CD CP4I-SC2 Facultatif : configuration de LDAP

Le IBM MQ Operator peut être configuré de sorte à utiliser plusieurs approches de sécurité différentes. En général, LDAP est le plus efficace pour un déploiement d'entreprise ; il est utilisé pour ce scénario de migration.

Avant de commencer

Cette tâche suppose que vous avez extrait et acquis les clés et certificats du gestionnaire de files d'attente.

Pourquoi et quand exécuter cette tâche

Cette tâche est-elle obligatoire ?

Si vous utilisez déjà LDAP pour l'authentification et l'autorisation, aucune modification n'est requise.

Si vous n'êtes pas certain que LDAP est utilisé, exécutez la commande suivante :

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

Exemple de sortie :

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

Deux attributs de la sortie présentent un intérêt particulier :

AUTHTYPE
Si sa valeur est `IDPWLDAP`, vous utilisez LDAP pour l'authentification.

Si sa valeur est vide ou autre, LDAP n'est pas configuré. Dans ce cas, vérifiez l'attribut `AUTHORMD` pour déterminer si des utilisateurs LDAP sont utilisés pour l'autorisation.

AUTHORMD
Si sa valeur est `OS`, vous utilisez LDAP pour l'autorisation.

Pour modifier l'autorisation et l'authentification afin qu'elles utilisent LDAP, procédez comme suit :

Procédure

1. Mettez à jour la sauvegarde IBM MQ pour le serveur LDAP.
2. Mettez à jour la sauvegarde IBM MQ pour les informations d'autorisation LDAP.

OpenShift CD CP4I-SC2 LDAP - Partie 1 : mise à jour de la sauvegarde IBM MQ pour le serveur LDAP.

Ce scénario n'inclut pas de description complète de la procédure de configuration de LDAP. Cette rubrique résume la procédure et fournit un exemple et des références à des informations supplémentaires.

Avant de commencer

Cette tâche suppose que vous avez extrait et acquis les clés et certificats du gestionnaire de files d'attente.

Pourquoi et quand exécuter cette tâche

Cette tâche est-elle obligatoire ?

Si vous utilisez déjà LDAP pour l'authentification et l'autorisation, aucune modification n'est requise. Si vous n'êtes pas certain que LDAP est utilisé, reportez-vous à la rubrique «Facultatif : configuration de LDAP», à la page 114.

La configuration du serveur LDAP se déroule en deux parties :

1. Définissez une configuration LDAP.
2. Associez la configuration LDAP à la définition de gestionnaire de files d'attente.

Informations supplémentaires pour vous aider dans cette configuration :

- Présentation du référentiel d'utilisateurs
- Guide de référence de la commande AUTHINFO

Procédure

1. Définissez une configuration LDAP.

   Editez le fichier `backup.mqsc` afin de définir un nouvel objet **AUTHINFO** pour le système LDAP. Exemple :

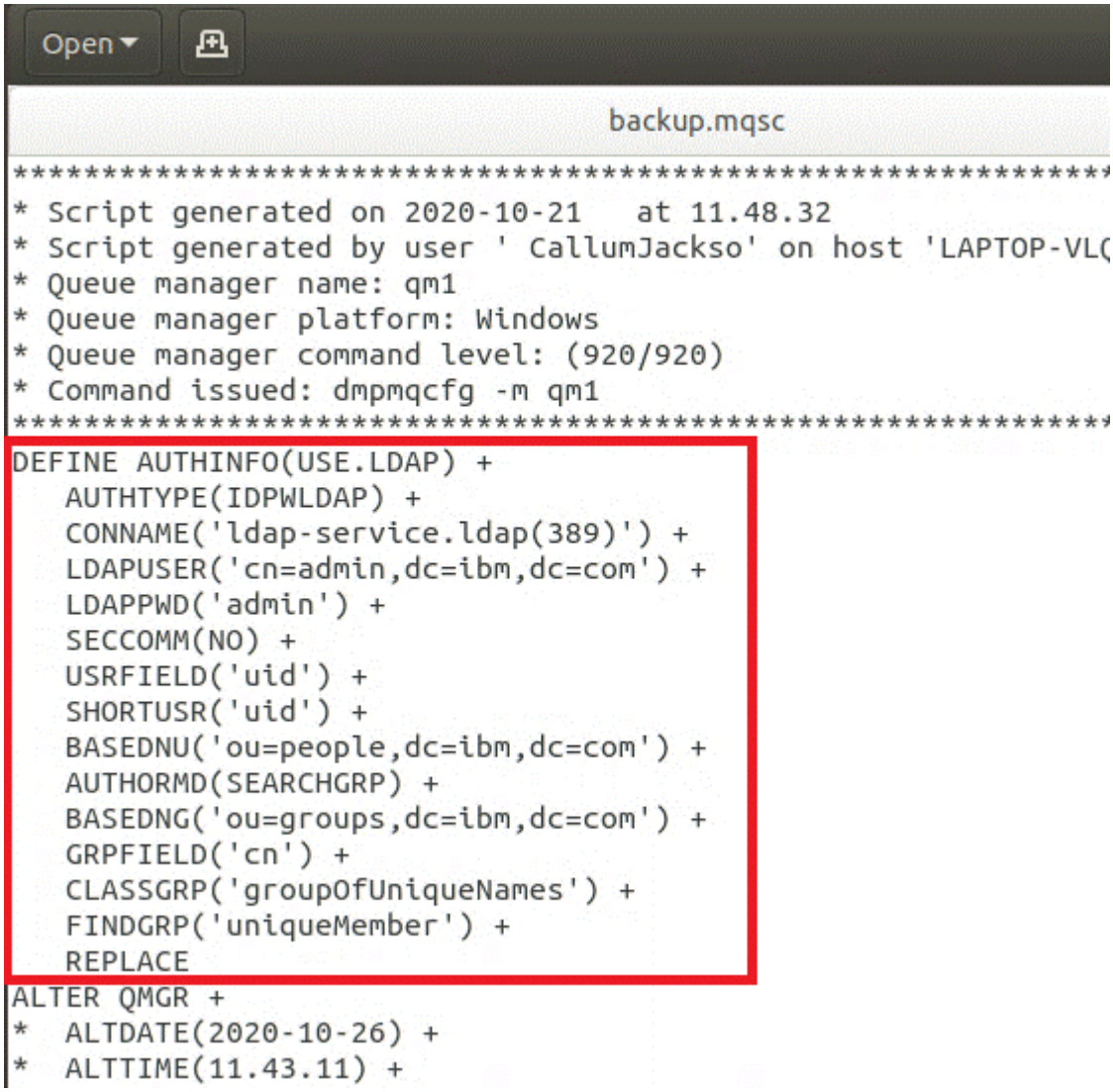
```
DEFINE AUTHINFO(USE.LDAP) +
   AUTHTYPE(IDPWLDAP) +
   CONNAME('ldap-service.ldap(389)') +
   LDAPUSER('cn=admin,dc=ibm,dc=com') +
   LDAPPWD('admin') +
   SECCOMM(NO) +
   USRFIELD('uid') +
   SHORTUSR('uid') +
   BASEDNU('ou=people,dc=ibm,dc=com') +
   AUTHORMD(SEARCHGRP) +
   BASEDNG('ou=groups,dc=ibm,dc=com')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

   Où

   - **CONNAME** représente le nom d'hôte et le port correspondant au serveur LDAP. S'il existe plusieurs adresses à des fins de résilience, elles peuvent être configurées à l'aide d'une liste de valeurs séparées par des virgules.
   - **LDAPUSER** représente le nom distinctif correspondant à l'utilisateur utilisé par IBM MQ lors de la connexion à LDAP pour interroger les enregistrements utilisateur.
   - **LDAPPWD** représente le mot de passe qui correspond à l'utilisateur **LDAPUSER**.
   - **SECCOM** indique si les communications avec le serveur LDAP doivent utiliser TLS. Valeurs possibles :
     - YES : TLS est utilisé et un certificat est présenté par le serveur IBM MQ.
     - ANON : TLS est utilisé sans certificat présenté par le serveur IBM MQ.

  - NO : TLS n'est pas utilisé lors de la connexion.
- **USRFIELD** spécifie la zone de l'enregistrement LDAP à laquelle le nom d'utilisateur présenté doit correspondre.
- **SHORTUSR** est une zone de l'enregistrement LDAP qui ne dépasse pas 12 caractères. La valeur de cette zone correspond à l'identité déclarée si l'authentification aboutit.
- **BASEDNU** représente le nom distinctif de base qui doit être utilisé pour les recherches dans LDAP.
- **BASEDNG** représente le nom distinctif de base des groupes dans LDAP.
- **AUTHORMD** définit le mécanisme utilisé pour résoudre l'appartenance à un groupe de l'utilisateur. Quatre options sont disponibles :
  - OS : recherchez les groupes associés au nom abrégé sur le système d'exploitation.
  - SEARCHGRP : recherchez l'utilisateur authentifié dans les entrées de groupe de LDAP.
  - SEARCHUSR : recherchez les informations sur l'appartenance à un groupe dans l'enregistrement de l'utilisateur authentifié.
  - SRCHGRPSN : recherchez le nom abrégé de l'utilisateur authentifié dans les entrées de groupe de LDAP (défini par la zone SHORTUSR).
- **GRPFIELD** représente l'attribut de l'enregistrement de groupe LDAP qui correspond à un nom simple. S'il est spécifié, il peut être utilisé pour définir des enregistrements d'autorisation.
- **CLASSUSR** représente la classe d'objet LDAP qui correspond à un utilisateur.
- **CLASSGRP** représente la classe d'objet LDAP qui correspond à un groupe.
- **FINDGRP** représente l'attribut de l'enregistrement LDAP qui correspond à l'appartenance à un groupe.

La nouvelle entrée peut être placée n'importe où dans le fichier, mais il peut s'avérer utile de placer les nouvelles entrées au début du fichier :



2. Associez la configuration LDAP à la définition de gestionnaire de files d'attente.

   Vous devez associer la configuration LDAP à la définition de gestionnaire de files d'attente. Immédiatement sous l'entrée `DEFINE AUTHINFO` figure une entrée `ALTER QMGR`. Modifiez l'entrée CONNAUTH pour qu'elle corresponde au nom AUTHINFO nouvellement créé. Par exemple, dans l'exemple précédent, `AUTHINFO(USE.LDAP)` étant défini, le nom est `USE.LDAP`. Vous devez donc remplacer `CONNAUTH('SYSTEM.DEFAULT.AUTHINFO.IDPWOS')` par `CONNAUTH('USE.LDAP')` :

Pour que le basculement vers LDAP soit immédiat, appelez une commande REFRESH SECURITY en ajoutant une ligne immédiatement après la commande ALTER QMGR :

*backup.mqsc

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

Que faire ensuite

Vous êtes maintenant prêt à mettre à jour la sauvegarde IBM MQ pour les informations d'autorisation LDAP.

OpenShift CD CP4I-SC2 LDAP - Partie 2 : mise à jour de la sauvegarde IBM MQ pour les informations d'autorisation LDAP

IBM MQ fournit des règles d'autorisation précises qui contrôlent l'accès aux objets IBM MQ. Si vous avez choisi LDAP pour l'authentification et l'autorisation, les règles d'autorisation peuvent être non valides et nécessiter une mise à jour.

Avant de commencer

Cette tâche suppose que vous avez mis à jour la sauvegarde du serveur LDAP.

Pourquoi et quand exécuter cette tâche

Cette tâche est-elle obligatoire ?

Si vous utilisez déjà LDAP pour l'authentification et l'autorisation, aucune modification n'est requise. Si vous n'êtes pas certain que LDAP est utilisé, reportez-vous à la rubrique «Facultatif : configuration de LDAP», à la page 114.

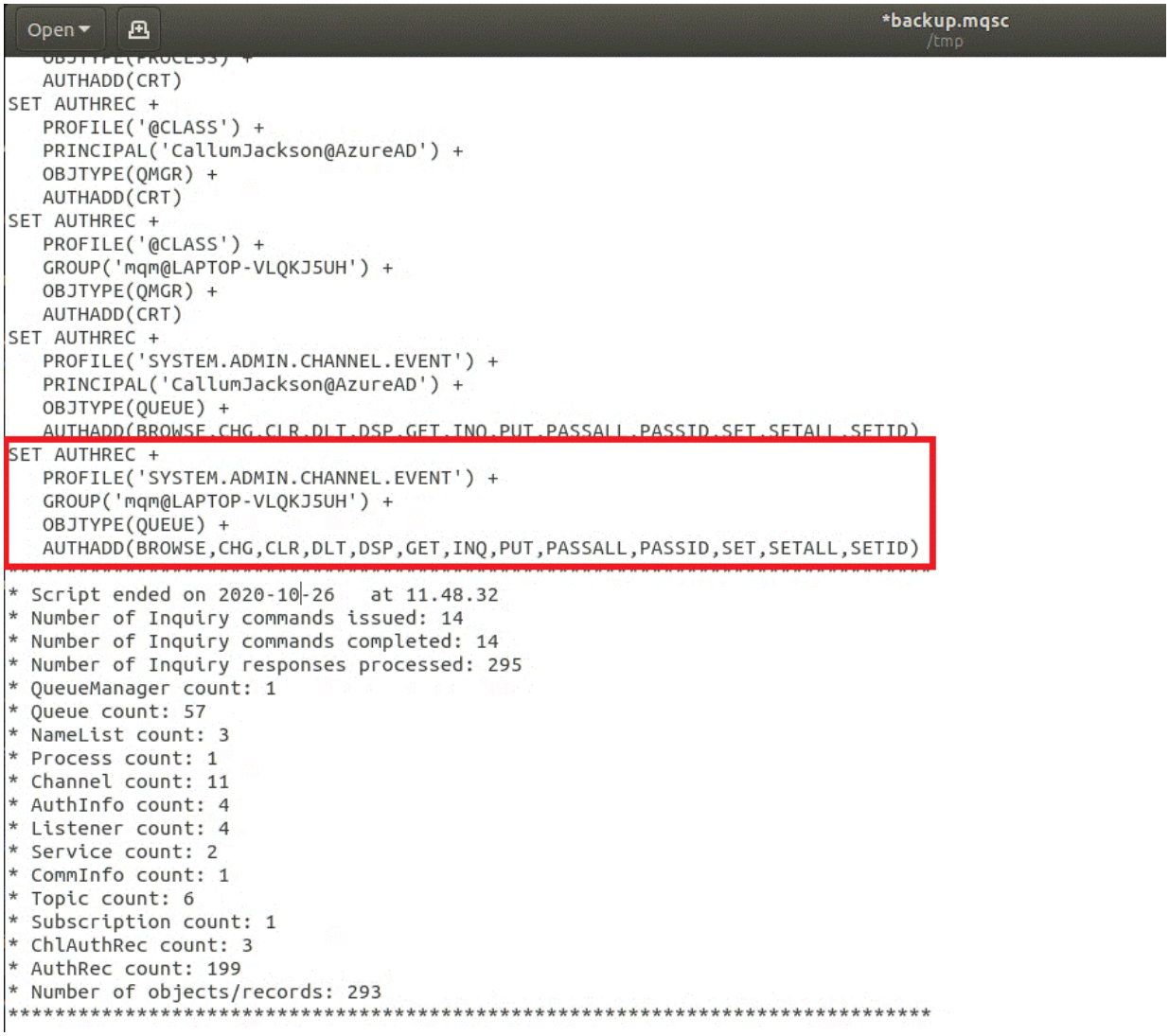
La mise à jour des informations d'autorisation LDAP se déroule en deux temps :

1. Suppression de toutes les autorisations existantes du fichier.
2. Définition des nouvelles informations d'autorisation pour LDAP.

Procédure

1. Supprimez toutes les autorisations existantes du fichier.

   Dans le fichier de sauvegarde, vers la fin du fichier, vous devez voir plusieurs entrées commençant par SET AUTHREC :



Recherchez les entrées existantes et supprimez-les. L'approche la plus directe consiste à supprimer toutes les règles SET AUTHREC existantes, puis à créer des entrées en fonction des entrées LDAP.

2. Définissez les nouvelles informations d'autorisation pour LDAP.

   En fonction de la configuration de votre gestionnaire de files d'attente et du nombre de ressources et de groupes, cette activité peut s'avérer fastidieuse ou simple. L'exemple suivant suppose que votre gestionnaire de files d'attente ne possède qu'une seule file d'attente appelée Q1 et que vous souhaitez que le groupe LDAP apps y ait accès.

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

   La première commande AUTHREC ajoute le droit d'accès au gestionnaire de files d'attente et la deuxième permet d'accéder à la file d'attente. Si l'accès à une deuxième file d'attente est requis, une troisième commande AUTHREC est nécessaire, à moins que vous n'ayez décidé d'utiliser des caractères génériques pour fournir un accès plus générique.

   Voici un autre exemple. Si un groupe d'administrateurs (appelé admins) a besoin d'un accès complet au gestionnaire de files d'attente, ajoutez les commandes suivantes :

```
SET AUTHREC PROFILE('*') OBJTYPE(QUEUE) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(TOPIC) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(CHANNEL) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(CLNTCONN) GROUP('admins') AUTHADD(ALL)
```

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

Que faire ensuite

Vous êtes maintenant prêt à modifier les adresses IP et noms d'hôte dans la configuration IBM MQ.

OpenShift CD CP4I-SC2 Facultatif : modification des adresses IP et noms d'hôte dans la configuration IBM MQ

Des adresses IP et noms d'hôte ont peut-être été spécifiés pour la configuration IBM MQ. Dans certains cas, ils peuvent être conservés, alors que dans d'autres, ils doivent être mis à jour.

Avant de commencer

Cette tâche suppose que vous avez configuré LDAP.

Pourquoi et quand exécuter cette tâche

Cette tâche est-elle obligatoire ?

Déterminez au préalable si des adresses IP ou des noms d'hôte ont été spécifiés, en dehors de la configuration LDAP définies dans la section précédente. Pour cela, exécutez la commande suivante :

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

Exemple de sortie :

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

Dans cet exemple, la recherche renvoie trois résultats. Un résultat correspond à la configuration LDAP définie précédemment. Ce résultat peut être ignoré car le nom d'hôte du serveur LDAP reste le même. Les deux autres résultats correspondent à des entrées de connexion vides et peuvent donc être également ignorés. Si vous ne disposez d'aucune autre entrée, vous pouvez ignorer le reste de cette rubrique.

Procédure

1. Vous devez comprendre les entrées renvoyées.

   IBM MQ peut inclure des adresses IP, des noms d'hôte et des ports dans de nombreux aspects de la configuration. Nous pouvons les classer en deux catégories :

   a. **Emplacement de ce gestionnaire de files d'attente** : informations d'emplacement que ce gestionnaire de files d'attente utilise ou publie, que d'autres gestionnaires de files d'attente ou applications au sein d'un réseau IBM MQ peuvent utiliser pour la connectivité.

b. **Emplacement des dépendances du gestionnaire de files d'attente** : emplacements des autres gestionnaires de files d'attente ou systèmes dont ce gestionnaire de files d'attente doit être conscient.

Ce scénario ne s'intéressant qu'aux modifications apportées à cette configuration de gestionnaire de files d'attente, nous ne traitons que les mises à jour de configuration de la catégorie (a). Toutefois, si l'emplacement de ce gestionnaire de files d'attente est référencé par d'autres gestionnaires de files d'attente ou applications, leur configuration peut avoir besoin d'être mise à jour conformément au nouvel emplacement de ce gestionnaire de files d'attente.

Deux objets clés peuvent contenir des informations à mettre à jour :

- Programmes d'écoute : ils représentent l'adresse réseau sur laquelle IBM MQ écoute.
- Canal CLUSTER RECEIVER : si le gestionnaire de files d'attente fait partie d'un cluster IBM MQ, cet objet existe. Il spécifie l'adresse réseau à laquelle les autres gestionnaires de files d'attente peuvent se connecter.

2. Dans la sortie d'origine de la commande `grep 'CONNAME\|LOCLADDR\|IPADDRV' -B 3 backup.mqsc` , identifiez si des canaux CLUSTER RECEIVER sont définis. Si tel est le cas, mettez à jour les adresses IP.

   Pour déterminer si des canaux CLUSTER RECEIVER sont définis, recherchez les entrées contenant CHLTYPE(CLUSRCVR) dans la sortie d'origine :

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

   Si des entrées existent, mettez à jour CONNAME avec la route IBM MQ Red Hat OpenShift. Cette valeur est basée sur l'environnement Red Hat OpenShift et utilise une syntaxe prévisible :

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

   Par exemple, si le déploiement du gestionnaire de files d'attente est nommé qm1 dans l'espace de nom cp4i et que *openshift_app_route_hostname* est `apps.callumj.icp4i.com`, l'URL de la route est la suivante :

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

   Le numéro de port du chemin est généralement 443. À moins que votre administrateur Red Hat OpenShift ne vous indique différemment, il s'agit normalement de la valeur correcte. A l'aide de ces informations, mettez à jour les zones CONNAME. Exemple :

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

   Dans la sortie d'origine de la commande `grep 'CONNAME\|LOCLADDR\|IPADDRV' -B 3 backup.mqsc` , vérifiez s'il existe des entrées pour LOCLADDR ou IPADDRV. Si c'est le cas, supprimez-les. Elles ne sont pas pertinentes dans un environnement de conteneur.

Que faire ensuite

Vous êtes maintenant prêt à mettre à jour la configuration du gestionnaire de files d'attente pour un environnement de conteneur.

OpenShift CD CP4I-SC2 Mise à jour de la configuration du gestionnaire de files d'attente pour un environnement de conteneur

Lors de l'exécution dans un conteneur, certains aspects de la configuration sont définis par le conteneur et peuvent être en conflit avec la configuration exportée.

Avant de commencer

Cette tâche suppose que vous avez modifié la configuration des adresses IP et noms d'hôte d'IBM MQ.

Pourquoi et quand exécuter cette tâche

Les aspects suivants de la configuration sont définis par le conteneur :

- Définitions du programme d'écoute (qui correspondent aux ports exposés).
- Emplacement de tout magasin TLS potentiel.

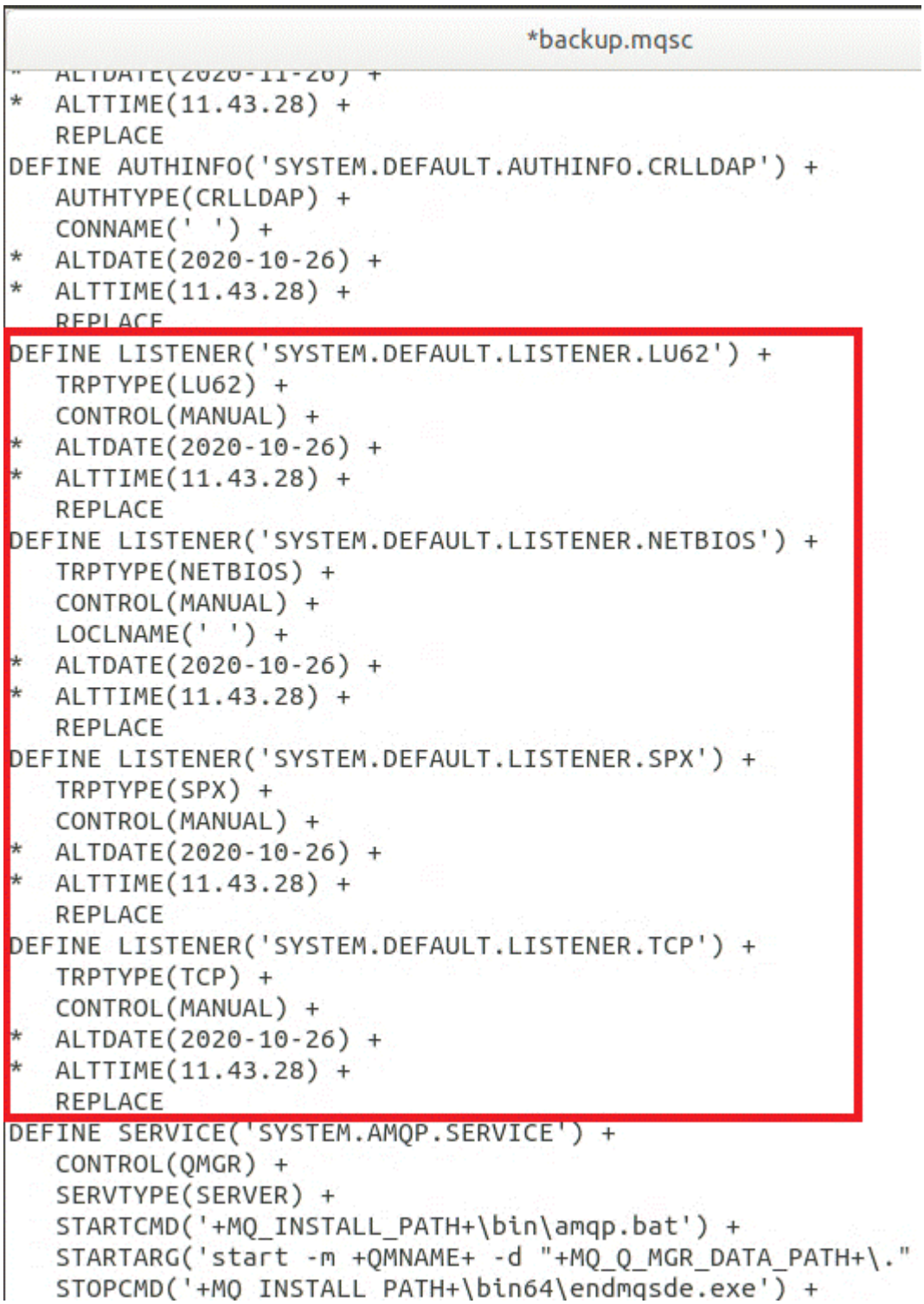
Par conséquent, vous devez mettre à jour la configuration exportée :

1. Supprimez les définitions du programme d'écoute.
2. Définissez l'emplacement du référentiel de clés TLS.

Procédure

1. Supprimez les définitions du programme d'écoute.

   Dans la configuration de sauvegarde, recherchez `DEFINE LISTENER`. Cette section doit se trouver entre les définitions `AUTHINFO` et `SERVICE`. Mettez en évidence la zone, puis supprimez-la.



2. Définissez l'emplacement du référentiel de clés TLS.

   La sauvegarde du gestionnaire de files d'attente contient la configuration TLS de l'environnement d'origine. Cela est différent de l'environnement de conteneur, et quelques mises à jour sont donc nécessaires :

   - Remplacez l'entrée **CERTLABL** par `default`
   - Remplacez l'emplacement du référentiel de clés TLS (**SSLKEYR**) par `/run/runmqserver/tls/key`

Pour rechercher l'emplacement de l'attribut **SSLKEYR** dans le fichier, recherchez **SSLKEYR**. En général, il n'existe qu'une seule entrée. Si plusieurs entrées sont détectées, veillez à bien éditer l'objet **QMGR** comme indiqué dans l'illustration suivante :

*backup.mqsc

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

Que faire ensuite

Vous êtes maintenant prêt à sélectionner l'architecture cible d'IBM MQ exécutée en conteneurs.

OpenShift CD CP4I-SC2 Sélection de l'architecture haute disponibilité cible pour IBM MQ exécuté en conteneurs

Choisissez entre une instance unique (un seul pod Kubernetes), plusieurs instances (deux pods) et Native HA (un pod de réplique active et deux pods de réplique de secours) pour répondre à vos exigences de haute disponibilité.

Avant de commencer

Cette tâche suppose que vous avez mis à jour la configuration du gestionnaire de files d'attente pour un environnement de conteneur.

Pourquoi et quand exécuter cette tâche

IBM MQ Operator fournit trois options de haute disponibilité:

- **Instance unique** : Un conteneur unique (Pod) est démarré et il est de la responsabilité de Red Hat OpenShift de redémarrer en cas d'échec. En raison des caractéristiques d'un ensemble avec état dans Kubernetes, il arrive parfois que cette reprise en ligne soit longue ou qu'elle requière une action administrative.
- **Multi-instance** : deux conteneurs (chacun dans un pod distinct) sont démarrés ; l'un en mode actif et l'autre en mode de secours. Cette topologie permet une reprise en ligne bien plus rapide. Elle requiert un système de fichiers RWM (Read Write Many) qui répond aux exigences d'IBM MQ.
- **Native HA**: Trois conteneurs (chacun dans un pod distinct), chacun avec une instance du gestionnaire de files d'attente. Une instance correspond au gestionnaire de files d'attente actif, qui traite les messages et écrit dans son journal de reprise. A chaque écriture dans le journal de reprise, le gestionnaire de files d'attente actif envoie les données aux deux autres instances, appelées répliques. Si le pod qui exécute le gestionnaire de files d'attente actif échoue, l'une des répliques d'instance du gestionnaire de files d'attente devient actif et dispose des données à jour qu'il peut utiliser.

Dans cette tâche, vous vous contentez de choisir l'architecture haute disponibilité cible. Les étapes de configuration de l'architecture choisie sont décrites dans une tâche ultérieure de ce scénario («Création du gestionnaire de files d'attente dans Red Hat OpenShift», à la page 129).

Procédure

1. Passez en revue les trois options.

   Pour une description complète de ces options, voir «Planification de la haute disponibilité pour IBM MQ dans des conteneurs», à la page 19.
2. Sélectionnez l'architecture haute disponibilité cible.

   Si vous ne savez pas quelle option choisir, commencez par l'option **Instance unique** et vérifiez si elle répond à vos besoins en matière de haute disponibilité.

Que faire ensuite

Vous êtes maintenant prêt à créer les ressources du gestionnaire de files d'attente.

OpenShift CD CP4I-SC2 Création des ressources du gestionnaire de files d'attente

Importez la configuration IBM MQ, ainsi que les certificats et les clés TLS, dans l'environnement Red Hat OpenShift.

Avant de commencer

Cette tâche suppose que vous avez sélectionné l'architecture cible pour IBM MQ exécuté en conteneurs.

Pourquoi et quand exécuter cette tâche

Dans les sections précédentes, vous avez extrait, mis à jour et défini deux ressources :

- Configuration de IBM MQ
- Certificats et clés TLS

Vous devez importer ces ressources dans l'environnement Red Hat OpenShift avant le déploiement du gestionnaire de files d'attente.

Procédure

1. Importez la configuration IBM MQ dans Red Hat OpenShift.

   Les instructions ci-après supposent que la configuration IBM MQ se trouve dans le répertoire de travail, dans un fichier appelé `backup.mqsc`. Sinon, vous devez personnaliser le nom de ce fichier en fonction de votre environnement.

   a) Connectez-vous à votre cluster à l'aide de la commande `oc login`.

   b) Chargez la configuration IBM MQ dans une `configmap`.

      Exécutez ensuite la commande suivante :

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

   c) Vérifiez que le fichier a bien été chargé.

      Exécutez ensuite la commande suivante :

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

2. Importez les ressources TLS d'IBM MQ

   Comme indiqué dans la rubrique «Facultatif : extraction et acquisition des clés et certificats du gestionnaire de files d'attente», à la page 112, TLS peut être requis pour le déploiement du gestionnaire de files d'attente. Si tel est le cas, vous devez déjà avoir un certain nombre de fichiers se terminant par `.crt` et `.key`. Vous devez les ajouter dans les secrets Kubernetes pour que le gestionnaire de files d'attente y fasse référence lors de la phase de déploiement.

   Par exemple, si vous disposez d'une clé et d'un certificat pour le gestionnaire de files d'attente, ils peuvent s'appeler :

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

   Pour importer ces fichiers, exécutez la commande suivante :

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

   Kubernetes fournit cet utilitaire utile lorsque vous importez une clé publique et privée correspondante. Pour ajouter d'autres certificats, par exemple, dans le magasin de clés de confiance du gestionnaire de files d'attente, exécutez la commande suivante :

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

   Par exemple, si les fichiers à importer sont `trust1.crt`, `trust2.crt` et `trust3.crt`, la commande est la suivante :

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

Que faire ensuite

Vous êtes maintenant prêt à créer le gestionnaire de files d'attente sur Red Hat OpenShift.

OpenShift CD CP4I-SC2 Création du gestionnaire de files d'attente dans Red Hat OpenShift

Déployez un gestionnaire de files d'attente à instance unique ou multi-instance dans Red Hat OpenShift.

Avant de commencer

Cette tâche suppose que vous avez créé les ressources du gestionnaire de files d'attente et installé le IBM MQ Operator dans Red Hat OpenShift.

Pourquoi et quand exécuter cette tâche

Comme indiqué dans «Sélection de l'architecture haute disponibilité cible pour IBM MQ exécuté en conteneurs», à la page 127, il existe trois topologies de déploiement possibles. Par conséquent, cette rubrique fournit trois modèles différents:

- Modèle 1: Déploiement d'un gestionnaire de files d'attente à instance unique.
- Modèle 2: Déploiement d'un gestionnaire de files d'attente multi-instance.
- Modèle 3: Déploiement d'un gestionnaire de files d'attente Native HA.

Important : Ne complétez qu'un seul des trois modèles, en fonction de votre topologie préférée.

Procédure

- **Modèle 1: Déploiement d'un gestionnaire de files d'attente à instance unique.**

  Le gestionnaire de files d'attente migré est déployé sur Red Hat OpenShift à l'aide d'un fichier YAML. En voici un exemple, basé sur les noms utilisés dans les rubriques précédentes :

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: qm1
spec:
  version: 9.4.0.0-r1
  license:
    accept: true
    license: L-BMSF-5YDSLR
    use: "Production"
  pki:
    keys:
    - name: default
      secret:
        secretName: my-tls-migration
        items:
          - tls.key
          - tls.crt
  web:
    enabled: true
  queueManager:
    name: QM1
    mqsc:
      - configMap:
          name: my-mqsc-migrated
          items:
            - backup.mqsc
```

  Selon les étapes que vous avez effectuées, il peut être nécessaire de personnaliser le fichier YAML précédent. Pour vous y aider, vous trouverez ci-dessous une explication de ce fichier YAML :

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
```

```
metadata:
  name: qm1
```

Définit l'objet Kubernetes, son type et son nom. La seule zone nécessitant une personnalisation est la zone `name`.

```
 spec:
  version: 9.4.0.0-r1
  license:
    accept: true
    license: L-BMSF-5YDSLR
    use: "Production"
```

Correspond aux informations de version et de licence du déploiement. Pour les personnaliser, utilisez les informations fournies dans la rubrique «Référence relative à l'octroi de licence pour mq.ibm.com/v1beta1», à la page 144.

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

Pour que le gestionnaire de files d'attente soit configuré afin d'utiliser TLS, il doit faire référence aux certificats et aux clés appropriés. La zone `secretName` fait référence au secret Kubernetes créé dans la section Importation des ressources IBM MQ du TLS, et la liste des éléments (`tls.key` et `tls.crt`) est le nom standard que Kubernetes attribue lors de l'utilisation de la syntaxe `oc create secret tls`. Si vous devez ajouter des certificats supplémentaires dans le magasin de clés de confiance, vous pouvez procéder de la même manière, mais les éléments sont les noms de fichier correspondants utilisés lors de l'importation. Par exemple, le code suivant peut être utilisé pour créer les certificats du magasin de clés de confiance :

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

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

Important : Si TLS n'est pas requis, supprimez la section TLS du fichier YAML.

```
  web:
    enabled: true
```

Active la console Web pour le déploiement

```
  queueManager:
    name: QM1
```

Spécifie QM1 comme nom de gestionnaire de files d'attente. Le gestionnaire de files d'attente est personnalisé en fonction de vos exigences (par exemple, le nom d'origine du gestionnaire de files d'attente).

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

Le code précédent extrait la configuration de gestionnaire de files d'attente importée dans la section Importez la configuration IBM MQ. Si vous avez utilisé des noms différents, vous devez modifier `my-mqsc-migrated` et `backup.mqsc`.

Notez que l'exemple YAML suppose que la classe de stockage par défaut pour l'environnement Red Hat OpenShift soit définie comme une classe de stockage RWX ou RWO. Si aucune valeur par défaut n'est définie dans votre environnement, vous devez spécifier la classe de stockage à utiliser. Pour cela, vous pouvez étendre le fichier YAML comme suit :

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

Ajoutez le texte mis en évidence, avec l'attribut de classe personnalisé en fonction de votre environnement. Pour découvrir les noms de classe de stockage dans votre environnement, exécutez la commande suivante :

```
oc get storageclass
```

Voici un exemple de sortie renvoyée par cette commande :

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

Le code suivant montre comment référencer la configuration IBM MQ importée dans la section Importez la configuration IBM MQ. Si vous avez utilisé des noms différents, vous devez modifier `my-mqsc-migrated` et `backup.mqsc`.

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

Vous avez déployé votre gestionnaire de files d'attente à instance unique. Le modèle est terminé. Vous êtes maintenant prêt à vérifier le nouveau déploiement de conteneur.

- **Modèle 2: Déploiement d'un gestionnaire de files d'attente multi-instance.**

  Le gestionnaire de files d'attente migré est déployé sur Red Hat OpenShift à l'aide d'un fichier YAML. L'exemple suivant reprend les noms utilisés dans les sections précédentes.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: qm1mi
spec:
  version: 9.4.0.0-r1
  license:
    accept: true
    license: L-BMSF-5YDSLR
    use: "Production"
  pki:
    keys:
    - name: default
      secret:
        secretName: my-tls-migration
        items:
          - tls.key
          - tls.crt
  web:
    enabled: true
  queueManager:
```

```
    name: QM1
    availability: MultiInstance
    storage:
      defaultClass: aws-efs
      persistedData:
        enabled: true
      queueManager:
        enabled: true
      recoveryLogs:
        enabled: true
    mqsc:
      - configMap:
          name: my-mqsc-migrated
          items:
            - backup.mqsc
```

Voici une explication de ce fichier YAML. La majorité de la configuration suit la même approche que le déploiement d'un gestionnaire de files d'attente à instance unique. Par conséquent, seuls les aspects de disponibilité et de stockage du gestionnaire de files d'attente sont expliqués ici.

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

Ceci indique le nom du gestionnaire de files d'attente sous la forme `QM1` et définit le déploiement comme étant `MultiInstance` au lieu de l'instance unique par défaut.

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

Un gestionnaire de files d'attente multi-instance IBM MQ dépend du stockage RWX. Par défaut, un gestionnaire de files d'attente est déployé en mode instance unique et des options de stockage supplémentaires sont donc requises lors d'un passage au mode multi-instance. Dans l'exemple de fichier YAML précédent, trois volumes persistants de stockage et une classe de volume persistant sont définis. Cette classe de volume persistant doit être une classe de stockage RWX. Si vous n'êtes pas certain des noms de classe de stockage dans votre environnement, vous pouvez exécuter la commande suivante pour les découvrir :

```
oc get storageclass
```

Voici un exemple de sortie renvoyée par cette commande :

```

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

Le code suivant montre comment référencer la configuration IBM MQ importée dans la section Importez la configuration IBM MQ. Si vous avez utilisé des noms différents, vous devez modifier `my-mqsc-migrated` et `backup.mqsc`.

```

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

Vous avez déployé votre gestionnaire de files d'attente multi-instance. Le modèle est terminé. Vous êtes maintenant prêt à vérifier le nouveau déploiement de conteneur.

- **Modèle 3: Déploiement d'un gestionnaire de files d'attente Native HA.**

Pour un exemple de création d'un gestionnaire de files d'attente Native HA, voir «Exemple: Configuration de Native HA à l'aide de IBM MQ Operator», à la page 79.

OpenShift CD CP4I-SC2 Vérification du nouveau déploiement de conteneur

Maintenant que IBM MQ est déployé sur Red Hat OpenShift, vous pouvez vérifier l'environnement à l'aide des exemples IBM MQ.

Avant de commencer

Cette tâche suppose que vous avez créé le gestionnaire de files d'attente sous Red Hat OpenShift.

Important : Cette tâche suppose que TLS n'est pas activé dans le gestionnaire de files d'attente.

Pourquoi et quand exécuter cette tâche

Dans cette tâche, vous exécutez les exemples IBM MQ depuis le conteneur du gestionnaire de files d'attente migré. Toutefois, vous pouvez choisir d'utiliser vos propres applications exécutées depuis un autre environnement.

Vous devez disposer des informations suivantes :

- Nom d'utilisateur LDAP
- Mot de passe LDAP
- Nom du canal IBM MQ
- Nom de la file d'attente

Cet exemple de code utilise les paramètres ci-après. Notez que vos paramètres seront différents.

- Nom d'utilisateur LDAP : mqapp
- Mot de passe LDAP : mqapp
- Nom du canal IBM MQ : DEV.APP.SVRCONN
- Nom de la file d'attente : Q1

Procédure

1. Exécutez la commande Exec dans le conteneur IBM MQ en cours d'exécution.

   Utilisez la commande suivante :

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

   où `qm1-ibm-mq-0` représente le pod que nous avons déployé dans la rubrique «Création du gestionnaire de files d'attente dans Red Hat OpenShift», à la page 129. Si votre déploiement porte un autre nom, personnalisez cette valeur.
2. Envoyez un message.

   Exécutez les commandes suivantes :

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

   Vous êtes invité à entrer un mot de passe, avant de pouvoir envoyer un message.
3. Vérifiez que le message a bien été reçu.

Exécutez l'exemple GET :

```
./amqsgetc Q1 QM1
```

Résultats

Vous avez terminé le «Migration vers IBM MQ Operator», à la page 109.

Que faire ensuite

Utilisez les informations suivantes pour vous aider dans des scénarios de migration plus complexes :

Migration des messages en file d'attente
Pour migrer des messages en file d'attente existants, suivez les conseils de la rubrique suivante pour l'exportation et l'importation de messages une fois que le nouveau gestionnaire de files d'attente est en place : Utilisation de l'utilitaire dmpmqmsg entre deux systèmes.

Connexion à IBM MQ à partir de l'environnement Red Hat OpenShift
Le gestionnaire de files d'attente déployé peut être exposé aux clients IBM MQ et aux gestionnaires de files d'attente en dehors de l'environnement Red Hat OpenShift. Le processus dépend de la version de IBM MQ qui se connecte à l'environnement Red Hat OpenShift. Voir «Configuration d'une route pour la connexion à un gestionnaire de files d'attente depuis l'extérieur d'un cluster Red Hat OpenShift», à la page 88.

Exploitation de IBM MQ dans des conteneurs

Si vous avez besoin d'utiliser ou d'interagir avec des gestionnaires de files d'attente IBM MQ s'exécutant dans des conteneurs, consultez les rubriques suivantes pour plus d'informations.

Procédure

- «Utilisation de IBM MQ à l'aide de IBM MQ Operator», à la page 134.
- «Affichage du statut des gestionnaires de files d'attente Native HA», à la page 142.
- «Arrêt manuel des instances de gestionnaire de files d'attente Native HA», à la page 144.

OpenShift CP4I Utilisation de IBM MQ à l'aide de IBM MQ Operator

Procédure

- «Connexion à IBM MQ Console déployée dans un cluster Red Hat OpenShift», à la page 134.
- «Surveillance lors de l'utilisation de IBM MQ Operator», à la page 135.
- «Sauvegarde et restauration de la configuration du gestionnaire de files d'attente à l'aide de l'interface de ligne de commande Red Hat OpenShift», à la page 141.

OpenShift CP4I Connexion à IBM MQ Console déployée dans un cluster Red Hat OpenShift

Comment se connecter au IBM MQ Console d'un gestionnaire de files d'attente qui a été déployé sur un cluster Red Hat OpenShift Container Platform .

Pourquoi et quand exécuter cette tâche

L'URL IBM MQ Console est disponible dans la page des détails `QueueManager` de la console Web Red Hat OpenShift ou dans le IBM Cloud Pak for Integration Platform UI. Vous pouvez également la trouver à partir de l'interface de ligne de commande Red Hat OpenShift en exécutant la commande suivante :

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

Si vous utilisez une licence IBM Cloud Pak for Integration , IBM MQ Console utilise Keycloak pour la gestion des identités et des accès. Voir Gestion des identités et des accès dans la documentation IBM Cloud Pak for Integration .

Si vous utilisez une licence IBM MQ , IBM MQ Console n'est pas préconfiguré et vous devez le configurer vous-même. Pour plus d'informations, voir Configuration des utilisateurs et des rôles. Pour un exemple, voir «Configuration de IBM MQ Console avec un registre de base à l'aide de IBM MQ Operator», à la page 101.

Tâches associées

«Configuration d'une route pour la connexion à un gestionnaire de files d'attente depuis l'extérieur d'un cluster Red Hat OpenShift», à la page 88
Vous avez besoin d'une route Red Hat OpenShift pour connecter une application à un gestionnaire de files d'attente IBM MQ depuis l'extérieur d'un cluster Red Hat OpenShift . Vous devez activer TLS sur votre gestionnaire de files d'attente et votre application client IBM MQ , car SNI est disponible uniquement dans le protocole TLS lorsqu'un protocole TLS 1.2 ou supérieur est utilisé. Red Hat OpenShift Container Platform Router utilise l'indication de nom de serveur pour acheminer les demandes vers le gestionnaire de files d'attente IBM MQ.

OpenShift CP4I *Octroi de droits pour le IBM MQ Console*

Les droits du IBM MQ Console sont gérés différemment en fonction de l'utilisation de votre licence.

Pourquoi et quand exécuter cette tâche

- Si vous utilisez une licence IBM Cloud Pak for Integration , IBM MQ Console utilise Keycloak pour la gestion des identités et des accès.
  - Voir Gestion des identités et des accès dans la documentation IBM Cloud Pak for Integration .
  - Si vous avez précédemment configuré des utilisateurs avec IAM sur des versions plus anciennes de IBM MQ Operator, voir Migration d'utilisateurs depuis IAM vers Keycloak.
- Si vous utilisez une licence IBM MQ , IBM MQ Console n'est pas préconfiguré et vous devez le configurer vous-même.
  - Pour plus d'informations sur les utilisateurs et les rôles, voir Configuration des utilisateurs et des rôles.
  - Pour un exemple simple, voir «Configuration de IBM MQ Console avec un registre de base à l'aide de IBM MQ Operator», à la page 101.
  - Vous pouvez également installer l'opérateur IBM Cloud Pak for Integration pour configurer Keycloak comme décrit précédemment.

OpenShift CP4I Surveillance lors de l'utilisation de IBM MQ Operator

Les gestionnaires de files d'attente gérés par le IBM MQ Operator peuvent produire des métriques compatibles avec Prometheus.

Vous pouvez afficher ces métriques à l'aide de la pile de surveillanceRed Hat OpenShift Container Platform (OCP). Ouvrez l'onglet **Métriques** dans OCP, puis cliquez sur **Observe** > **Métriques**. Les métriques de gestionnaire de files d'attente sont activées par défaut, mais peuvent être désactivées en définissant **`.spec.metrics.enabled`** sur `false`.

Prometheus est une base de données de série temporelle et un moteur d'évaluation de règle pour les métriques. Les conteneurs IBM MQ exposent un nœud final de métriques qui peut être interrogé par Prometheus. Les métriques sont générées à partir des rubriques du système MQ pour la surveillance et la trace d'activité.

OpenShift Container Platform inclut une pile de surveillance préconfigurée, préinstallée et mise à jour automatiquement qui utilise un serveur Prometheus. La pile de surveillance OpenShift Container Platform doit être configurée pour surveiller les projets définis par l'utilisateur. Pour plus d'informations, voir Enabling monitoring for user-defined projects. IBM MQ Operator crée un `ServiceMonitor` lorsque vous créez un `QueueManager` avec les mesures activées, que l'opérateur Prometheus peut ensuite reconnaître.

OpenShift CP4I *Métriques publiées lors de l'utilisation de IBM MQ Operator*

Les conteneurs du gestionnaire de files d'attente peuvent publier des mesures compatibles avec le monitoring Red Hat OpenShift.

Métrique	Tapez	Description
`ibmmq_qmgr_commit_total`	`counter`	Nombre de validations
`ibmmq_qmgr_cpu_load_fifteen_minute_average_percentage`	`gauge`	Charge UC - moyenne sur quinze minutes
`ibmmq_qmgr_cpu_load_five_minute_average_percentage`	`gauge`	Charge UC - moyenne sur cinq minutes
`ibmmq_qmgr_cpu_load_one_minute_average_percentage`	`gauge`	Charge UC - moyenne sur une minute
`ibmmq_qmgr_destructive_get_bytes_total`	`counter`	Commande get destructive d'intervalle total - nombre d'octets
`ibmmq_qmgr_destructive_get_total`	`counter`	Commande get destructive d'intervalle total - nombre
`ibmmq_qmgr_durable_subscription_alter_total`	`counter`	Modification du nombre d'abonnements durables
`ibmmq_qmgr_durable_subscription_create_total`	`counter`	Création du nombre d'abonnements durables
`ibmmq_qmgr_durable_subscription_delete_total`	`counter`	Suppression du nombre d'abonnements durables
`ibmmq_qmgr_durable_subscription_resume_total`	`counter`	Reprise du nombre d'abonnements durables
`ibmmq_qmgr_errors_file_system_free_space_percentage`	`gauge`	Système de fichiers d'erreurs MQ - espace disponible
`ibmmq_qmgr_errors_file_system_in_use_bytes`	`gauge`	Système de fichiers d'erreurs MQ - octets utilisés

Métrique	Tapez	Description
ibmmq_qmgr_expired_message_total	counter	Nombre de messages arrivés à expiration
ibmmq_qmgr_failed_browse_total	counter	Echec du calcul du nombre de visualisation
ibmmq_qmgr_failed_mqcb_total	counter	Echec du calcul du nombre de MQCB
ibmmq_qmgr_failed_mqclose_total	counter	Echec du calcul du nombre de MQCLOSE
ibmmq_qmgr_failed_mqconn_mqconnx_total	counter	Echec du calcul du nombre de MQCONN/MQCONNX
ibmmq_qmgr_failed_mqget_total	counter	Echec de MQGET - nombre
ibmmq_qmgr_failed_mqinq_total	counter	Echec du calcul du nombre de MQINQ
ibmmq_qmgr_failed_mqopen_total	counter	Echec du calcul du nombre de MQOPEN
ibmmq_qmgr_failed_mqput1_total	counter	Echec du calcul du nombre de MQPUT1
ibmmq_qmgr_failed_mqput_total	counter	Echec du calcul du nombre de MQPUT
ibmmq_qmgr_failed_mqset_total	counter	Echec du calcul du nombre de MQSET
ibmmq_qmgr_failed_mqsubrq_total	counter	Echec du calcul du nombre de MQSUBRQ
ibmmq_qmgr_failed_subscription_create_alter_resume_total	counter	Echec de la création/modification/reprise du calcul du nombre d'abonnements
ibmmq_qmgr_failed_subscription_delete_total	counter	Nombre d'échec de suppression d'abonnement
ibmmq_qmgr_failed_topic_mqput_mqput1_total	counter	Echec du calcul du nombre de MQPUT/MQPUT1 de rubrique
ibmmq_qmgr_fdc_files	gauge	Nombre de fichiers FDC MQ
ibmmq_qmgr_log_file_system_in_use_bytes	gauge	Système de fichier journal - octets utilisés
ibmmq_qmgr_log_file_system_max_bytes	gauge	Système de fichier journal - nombre d'octets maximal
ibmmq_qmgr_log_in_use_bytes	gauge	Journal - octets utilisés

Métrique	Tapez	Description
ibmmq_qmgr_log_logical_written_bytes_total	counter	Journal - octets logiques écrits
ibmmq_qmgr_log_max_bytes	gauge	Journal - nombre d'octets maximal
ibmmq_qmgr_log_occupied_by_reusable_extents_bytes	gauge	Journal - octets occupés par les domaines réutilisables
ibmmq_qmgr_log_physical_written_bytes_total	counter	Journal - octets physiques écrits
ibmmq_qmgr_log_primary_space_in_use_percentage	gauge	Journal - espace principal en cours utilisé
ibmmq_qmgr_log_required_for_media_recovery_bytes	gauge	Journal - octets requis pour la reprise sur incident
ibmmq_qmgr_log_workload_primary_space_utilization_percentage	gauge	Journal - utilisation de l'espace principal de charge de travail
ibmmq_qmgr_log_write_latency_seconds	gauge	Journal - temps d'attente d'écriture
ibmmq_qmgr_log_write_size_bytes	gauge	Journal - taille d'écriture
ibmmq_qmgr_mqcb_total	counter	Nombre de MQCB
ibmmq_qmgr_mqclose_total	counter	Nombre de MQCLOSE
ibmmq_qmgr_mqconn_mqconnx_total	counter	Nombre de MQCONN/MQCONNX
ibmmq_qmgr_mqctl_total	counter	Nombre de MQCTL
ibmmq_qmgr_mqdisc_total	counter	Nombre de MQDISC
ibmmq_qmgr_mqinq_total	counter	Nombre de MQINQ
ibmmq_qmgr_mqopen_total	counter	Nombre de MQOPEN
ibmmq_qmgr_mqput_mqput1_bytes_total	counter	Nombre d'octets MQPUT/MQPUT1 d'intervalle total
ibmmq_qmgr_mqput_mqput1_total	counter	Nombre de MQPUT/MQPUT1 d'intervalle total
ibmmq_qmgr_mqset_total	counter	Nombre de MQSET

Métrique	Tapez	Description
ibmmq_qmgr_mqstat_total	counter	Nombre de MQSTAT
ibmmq_qmgr_mqsubrq_total	counter	Nombre de MQSUBRQ
ibmmq_qmgr_non_durable_subscription_create_total	counter	Création du nombre d'abonnements non durables
ibmmq_qmgr_non_durable_subscription_delete_total	counter	Suppression du nombre d'abonnements non durables
ibmmq_qmgr_non_persistent_message_browse_bytes_total	counter	Visualisation de messages non persistants - nombre d'octets
ibmmq_qmgr_non_persistent_message_browse_total	counter	Visualisation de messages non persistants - nombre
ibmmq_qmgr_non_persistent_message_destructive_get_total	counter	Commande get destructive de message non persistant - nombre
ibmmq_qmgr_non_persistent_message_get_bytes_total	counter	Messages non persistants obtenus - nombre d'octets
ibmmq_qmgr_non_persistent_message_mqput1_total	counter	Nombre de MQPUT1 de message non persistant
ibmmq_qmgr_non_persistent_message_mqput_total	counter	Nombre de MQPUT de message non persistant
ibmmq_qmgr_non_persistent_message_put_bytes_total	counter	Messages non persistants insérés - nombre d'octets
ibmmq_qmgr_non_persistent_topic_mqput_mqput1_total	counter	Non persistant - nombre de MQPUT/MQPUT1 de rubrique
ibmmq_qmgr_persistent_message_browse_bytes_total	counter	Visualisation de messages persistants - nombre d'octets
ibmmq_qmgr_persistent_message_browse_total	counter	Visualisation de messages persistants - nombre
ibmmq_qmgr_persistent_message_destructive_get_total	counter	Commande get destructive de message persistant - nombre
ibmmq_qmgr_persistent_message_get_bytes_total	counter	Messages persistants obtenus - nombre d'octets

Métrique	Tapez	Description
ibmmq_qmgr_persistent_message_mqput1_total	counter	Nombre de MQPUT1 de message persistant
ibmmq_qmgr_persistent_message_mqput_total	counter	Nombre de MQPUT de message persistant
ibmmq_qmgr_persistent_message_put_bytes_total	counter	Messages persistants insérés - nombre d'octets
ibmmq_qmgr_persistent_topic_mqput_mqput1_total	counter	Persistant - nombre de MQPUT/MQPUT1 de rubrique
ibmmq_qmgr_published_to_subscribers_bytes_total	counter	Publication aux abonnés - nombre d'octets
ibmmq_qmgr_published_to_subscribers_message_total	counter	Publication aux abonnés - nombre de messages
ibmmq_qmgr_purged_queue_total	counter	Nombre de files d'attente purgées
ibmmq_qmgr_queue_manager_file_system_free_space_percentage	gauge	Système de fichiers du gestionnaire de files d'attente - espace disponible
ibmmq_qmgr_queue_manager_file_system_in_use_bytes	gauge	Système de fichiers du gestionnaire de files d'attente - octets utilisés
ibmmq_qmgr_ram_free_percentage	gauge	Pourcentage de mémoire vive disponible
ibmmq_qmgr_ram_usage_estimate_for_queue_manager_bytes	gauge	Nombre total d'octets de mémoire vive - estimation pour le gestionnaire de files d'attente
ibmmq_qmgr_rollback_total	counter	Nombre d'annulations
ibmmq_qmgr_system_cpu_time_estimate_for_queue_manager_percentage	gauge	Temps UC système - estimation du pourcentage pour le gestionnaire de files d'attente
ibmmq_qmgr_system_cpu_time_percentage	gauge	Pourcentage de temps UC système
ibmmq_qmgr_topic_mqput_mqput1_total	counter	Intervalle total MQPUT/MQPUT1 de rubrique
ibmmq_qmgr_topic_put_bytes_total	counter	Octets de rubrique d'intervalle total insérés

Métrique	Tapez	Description
ibmmq_qmgr_trace_file_system_free_space_percentage	gauge	Système de fichiers de trace MQ - espace disponible
ibmmq_qmgr_trace_file_system_in_use_bytes	gauge	Système de fichiers de trace MQ - octets utilisés
ibmmq_qmgr_user_cpu_time_estimate_for_queue_manager_percentage	gauge	Temps UC utilisateur - estimation du pourcentage pour le gestionnaire de files d'attente
ibmmq_qmgr_user_cpu_time_percentage	gauge	Pourcentage de temps UC utilisateur

Information associée

Métriques publiées sur les rubriques système

OpenShift CP4I Sauvegarde et restauration de la configuration du gestionnaire de files d'attente à l'aide de l'interface de ligne de commande Red Hat 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 à l'aide de la commande **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.

  a) Obtenez le nom du pod pour votre gestionnaire de files d'attente.
  Par exemple, vous pouvez exécuter la commande suivante, où *nom_gestionnaire_files_attente* est le nom de votre ressource `QueueManager` :

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

  b) 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.

a) Obtenez le nom du pod pour votre gestionnaire de files d'attente.
Par exemple, vous pouvez exécuter la commande suivante, où *nom_gestionnaire_files_attente* est le nom de votre ressource `QueueManager` :

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

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

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

MQ Adv. Affichage du statut des gestionnaires de files d'attente Native HA

Pour les conteneurs personnalisés, vous pouvez afficher le statut des instances Native HA à l'aide de la commande **dspmq** .

Pourquoi et quand exécuter cette tâche

Vous pouvez utiliser la commande **dspmq** pour afficher le statut opérationnel d'une instance de gestionnaire de files d'attente sur un noeud. Les informations renvoyées varient selon que l'instance est active ou qu'il s'agit d'une réplique. Les informations fournies par l'instance active sont définitives, tandis que celles des noeuds de réplique peuvent être obsolètes.

Vous pouvez effectuer les actions suivantes :

- Déterminer si l'instance de gestionnaire de files d'attente sur le noeud actuel est active ou s'il s'agit d'une réplique.
- Afficher le statut Native HA opérationnel de l'instance sur le noeud actuel.
- Afficher le statut opérationnel des trois instances dans une configuration Native HA.

Les zones de statut suivantes sont utilisées pour signaler le statut de la configuration Native HA :

ROLE
Indique le rôle en cours de l'instance et est l'un des rôles `Active`, `Replica` ou `Unknown`.

INSTANCE
Nom fourni pour cette instance du gestionnaire de files d'attente lorsque ce dernier a été créé à l'aide de l'option **-lr** de la commande **crtmqm**.

INSYNC
Indique si l'instance peut prendre la relève en tant qu'instance active, si nécessaire.

QUORUM
Indique le statut de quorum au format *nombre_instances_synchronisées*/*nombre_instances_configurées*.

REPLADDR
Adresse de réplication de l'instance de gestionnaire de files d'attente.

CONNACTV
Indique si le noeud est connecté à l'instance active.

BACKLOG
Indique le nombre de kilooctets de retard de l'instance.

CONNINST
Indique si l'instance désignée est connectée à cette instance.

ALTDATE
Indique la date à laquelle ces informations ont été mises à jour pour la dernière fois (vide si elles n'ont jamais été mises à jour).

ALTTIME
Indique l'heure à laquelle ces informations ont été mises à jour pour la dernière fois (vide si elles n'ont jamais été mises à jour).

Procédure

- Pour déterminer si une instance de gestionnaire de files d'attente est exécutée comme instance active ou comme réplique :

  ```
  dspmq -o status -m QMgrName
  ```

  Une instance active d'un gestionnaire de files d'attente nommé BOB signale le statut suivant :

  ```
  QMNAME(BOB)                STATUS(Running)
  ```

  Une réplique d'instance d'un gestionnaire de files d'attente nommé BOB signale le statut suivant :

  ```
  QMNAME(BOB)                STATUS(Replica)
  ```

  Une instance inactive signale le statut suivant :

  ```
  QMNAME(BOB)                STATUS(Ended Immediately)
  ```

- Pour déterminer le statut opérationnel Native HA de l'instance sur le noeud en cours:

  ```
  dspmq -o nativeha -m QMgrName
  ```

  L'instance active d'un gestionnaire de files d'attente nommé BOB peut signaler le statut suivant :

  ```
  QMNAME(BOB)               ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
  ```

  Une réplique d'instance d'un gestionnaire de files d'attente nommé BOB peut signaler le statut suivant :

  ```
  QMNAME(BOB)               ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
  ```

  Une instance inactive d'un gestionnaire de files d'attente nommé BOB peut signaler le statut suivant :

  ```
  QMNAME(BOB)               ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
  ```

- Pour déterminer le statut Native HA opérationnel de toutes les instances de la configuration Native HA :

  ```
  dspmq -o nativeha -x -m QMgrName
  ```

  Si vous exécutez cette commande sur le noeud qui exécute l'instance active du gestionnaire de files d'attente BOB, vous risquez de recevoir le statut suivant :

  ```
  QMNAME(BOB)            ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
   INSTANCE(inst1) ROLE(Active)  REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
  CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
   INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
  CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
   INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
  CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
  ```

  Si vous exécutez cette commande sur un noeud qui exécute une réplique d'instance du gestionnaire de files d'attente BOB, vous risquez de recevoir le statut suivant, qui indique que l'une des répliques est en retard :

  ```
  QMNAME(BOB)            ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
   INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
  CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
   INSTANCE(inst1) ROLE(Active)  REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
  CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
   INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(No)  BACKLOG(435)
  CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
  ```

Si vous exécutez cette commande sur un noeud qui exécute une instance inactive du gestionnaire de files d'attente BOB, vous risquez de recevoir le statut suivant :

```
QMNAME(BOB)            ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
 INSTANCE(inst1) ROLE(Unknown) REPLADDR(9.20.123.45) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATE() ALTTIME()
 INSTANCE(inst2) ROLE(Unknown) REPLADDR(9.20.123.46) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATE() ALTTIME()
 INSTANCE(inst3) ROLE(Unknown) REPLADDR(9.20.123.47) CONNACTV(No) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATE() ALTTIME()
```

Si vous exécutez la commande alors que les instances sont encore en cours de négociation pour déterminer l'instance active et les répliques, vous recevez le statut suivant :

```
QMNAME(BOB)              STATUS(Negotiating)
```

Référence associée

Commande dspmq (display queue managers)

MQ Adv. Arrêt manuel des instances de gestionnaire de files d'attente Native HA

Vous pouvez utiliser la commande **endmqm** pour arrêter un gestionnaire de files d'attente actif ou de réplique faisant partie d'un groupe Native HA.

Procédure

- Pour arrêter l'instance active d'un gestionnaire de files d'attente, voir Arrêt des gestionnaires de files d'attente Native HA dans la section Configuration de cette documentation.

OpenShift CP4I Informations de référence pour IBM MQ dans les conteneurs

IBM MQ fournit un opérateur Kubernetes, qui fournit une intégration native avec la plateforme de conteneurs Red Hat OpenShift.

OpenShift CP4I Référence d'API pour IBM MQ Operator

IBM MQ fournit un opérateur Kubernetes, qui fournit une intégration native avec la plateforme de conteneurs Red Hat OpenShift.

OpenShift CP4I Référence d'API pour mq.ibm.com/v1beta1

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

OpenShift CP4I CD CP4I-SC2 *Référence relative à l'octroi de licence pour mq.ibm.com/v1beta1*

Versions de licence actuelles

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 `spec.license.license`	Valeur de `spec.license.use`	Informations sur la licence	Versions IBM MQ applicables
`L-JTPV-KYG8TF`	`Production` ou `NonProduction`	IBM Cloud Pak for Integration 16.1.0	9.4.0
`L-BMSF-5YDSLR`	`Production` ou `NonProduction`	IBM Cloud Pak for Integration Edition limitée 16.1.0	9.4.0
`L-EHXT-MQCRN9`	`Production`	IBM MQ Advanced 9.4	9.4.0
`L-CLXQ-ADXTK3`	`Development`	IBM MQ Advanced for Developers (non garantie) 9.4	9.4.0

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

Versions de licence plus anciennes

Voir Versions de licence plus anciennes dans la documentation IBM MQ 9.3 .

OpenShift CP4I *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. IBM MQ : https://ibm.biz/BdPZqj. Référence de licence: https://ibm.biz/BdPZfq..

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` QueueManagerSpec	Etat souhaité pour le gestionnaire de files d'attente.
`status` QueueManagerStatus	Etat observé pour le gestionnaire de files d'attente.

.spec

Etat souhaité pour le gestionnaire de files d'attente.

Apparaît dans :

- «QueueManager», à la page 145

Zone	Description
`affinity`	Règles d'affinité Kubernetes standard. Pour plus d'informations, voir https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#affinity-v1-core.
`annotations` Annotations	La zone annotations sert de passe-système pour les annotations de pod. Les utilisateurs peuvent ajouter n'importe quelle annotation dans cette zone et l'appliquer au pod. Les annotations ici écrasent les annotations par défaut si elles sont fournies. Requiert MQ Operator 1.3.0 ou version ultérieure.
Tableau `imagePullSecrets` LocalObjectReference	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.
`labels` Libellés	La zone labels sert de passe-système pour les libellés de pod. Les utilisateurs peuvent ajouter n'importe quel libellé dans cette zone et l'appliquer au pod. Les libellés ici remplacent les libellés par défaut s'ils sont fournis. Requiert MQ Operator 1.3.0 ou version ultérieure.
`license` License	Paramètres contrôlant votre acceptation de la licence et les métriques de licence à utiliser.
`pki` 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).
`queueManager` QueueManagerConfig	Paramètres pour le conteneur du gestionnaire de files d'attente et le gestionnaire de files d'attente sous-jacent.
`securityContext` SecurityContext	Paramètres de sécurité à ajouter au pod securityContext du gestionnaire de files d'attente.
`telemetry` Télémétrie	Paramètres de configuration d'Open Telemetry. Requiert MQ Operator 2.2.0 ou version ultérieure.
`template` Template	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 `terminationGracePeriodSeconds`	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. La valeur par défaut est 30 secondes.
`tracing` TracingConfig	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). Exemple : `9.1.5.0-r2` spécifie la version 9.1.5.0 de MQ qui utilise 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.
`web` WebServerConfig	Paramètres pour le serveur Web MQ.

.spec.annotations

La zone annotations sert de passe-système pour les annotations de pod. Les utilisateurs peuvent ajouter n'importe quelle annotation dans cette zone et l'appliquer au pod. Les annotations ici écrasent les annotations par défaut si elles sont fournies. Requiert MQ Operator 1.3.0 ou version ultérieure.

Apparaît dans :

- «.spec», à la page 145

.spec.imagePullSecrets

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

Apparaît dans :

- «.spec», à la page 145

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

.spec.labels

La zone labels sert de passe-système pour les libellés de pod. Les utilisateurs peuvent ajouter n'importe quel libellé dans cette zone et l'appliquer au pod. Les libellés ici remplacent les libellés par défaut s'ils sont fournis. Requiert MQ Operator 1.3.0 ou version ultérieure.

Apparaît dans :

- «.spec», à la page 145

.spec.license

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

Apparaît dans :

- «.spec», à la page 145

Zone	Description
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/BdPZfq .

Zone	Description
Chaîne `metric`	Paramètre qui spécifie la métrique de licence à utiliser. Par exemple, `ProcessorValueUnit`, `VirtualProcessorCore` ou `ManagedVirtualServer`. Prend par défaut la valeur `ProcessorValueUnit` lors de l'utilisation d'une licence MQ et de `VirtualProcessorCore` lors de l'utilisation d'une licence Cloud Pak for Integration.
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/BdPZfq .

.spec.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 :

- «.spec», à la page 145

Zone	Description
Tableau `keys` PKISource	Clés privées à ajouter au référentiel de clés du gestionnaire de files d'attente.
Tableau `trust` PKISource	Certificats à ajouter au référentiel de clés du gestionnaire de files d'attente.

.spec.pki.keys

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

Apparaît dans :

- «.spec.pki», à la page 148

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

.spec.pki.keys.secret

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

Apparaît dans :

- «.spec.pki.keys», à la page 148

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.

.spec.pki.trust

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

Apparaît dans :

- «.spec.pki», à la page 148

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

.spec.pki.trust.secret

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

Apparaît dans :

- «.spec.pki.trust», à la page 148

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.

.spec.queueManager

Paramètres pour le conteneur du gestionnaire de files d'attente et le gestionnaire de files d'attente sous-jacent.

Apparaît dans :

- «.spec», à la page 145

Zone	Description
`availability` Availability	Paramètres de disponibilité pour le gestionnaire de files d'attente, indiquant par exemple si une paire actif-secours ou Native HA doit être utilisée.
Valeur booléenne `debug`	Indique si les messages de débogage du code propre au conteneur doivent être consignés ou non dans le journal du conteneur. La valeur par défaut est false.
Chaîne `image`	Image de conteneur à utiliser.
Chaîne `imagePullPolicy`	Paramètre qui contrôle à quel moment le kubelet tente d'extraire l'image spécifiée. La valeur par défaut est `IfNotPresent`.
Tableau `ini` INISource	Paramètres permettant de fournir les informations INI pour le gestionnaire de files d'attente. Requiert MQ Operator 1.1.0 ou version ultérieure.
`livenessProbe` QueueManagerLivenessProbe	Paramètres qui contrôlent la sonde de non-défaillance.
Chaîne `logFormat`	Indique le format de journal à utiliser pour ce conteneur. Utilisez `JSON` pour les journaux au format JSON du conteneur. Utilisez `Basic` pour les messages au format texte. La valeur par défaut est `Basic`.
`metrics` QueueManagerMetrics	Paramètres pour les métriques de style Prometheus.
Tableau `mqsc` MQSCSource	Paramètres permettant de fournir des informations MQSC pour le gestionnaire de files d'attente. Requiert MQ Operator 1.1.0 ou version ultérieure.

Zone	Description
Chaîne `name`	Nom du gestionnaire de files d'attente MQ sous-jacent, s'il est différent de metadata.name. Utilisez cette zone si vous souhaitez un nom de gestionnaire de files d'attente qui n'est pas conforme aux règles Kubernetes pour les noms (par exemple, un nom qui inclut des majuscules).
`readinessProbe` QueueManagerReadinessProbe	Paramètres qui contrôlent la sonde de vigilance.
`recoveryLogs` RecoveryLogs	Paramètres des journaux de reprise MQ . Requiert MQ Operator 2.4.0 ou version ultérieure.
`resources` Resources	Paramètres qui contrôlent les exigences relatives aux ressources.
`route` Route	Paramètres de la route du gestionnaire de files d'attente. Requiert MQ Operator 1.4.0 ou version ultérieure.
`startupProbe` StartupProbe	Paramètres qui contrôlent la sonde de démarrage. Ne s'applique qu'aux déploiements MultiInstance et NativeHA. Requiert MQ Operator 1.5.0 ou une version ultérieure.
`storage` QueueManagerStorage	Paramètres de stockage pour contrôler l'utilisation des volumes persistants et des classes de stockage par le gestionnaire de files d'attente.

.spec.queueManager.availability

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

Apparaît dans :

- «.spec.queueManager», à la page 149

Zone	Description
`tls` Tls	Paramètres TLS facultatifs permettant de configurer les communications sécurisées entre les répliques NativeHA. Requiert MQ Operator 1.5.0 ou une version ultérieure.
Chaîne `type`	Type de disponibilité à utiliser. Utilisez `SingleInstance` pour un pod unique, 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 `active` et l'autre est un gestionnaire de files d'attente de secours. Utilisez `NativeHA` pour la réplication Native HA (requiert MQ Operator 1.5.0 ou une version ultérieure). La valeur par défaut est `SingleInstance`. Pour plus d'informations, voir http://ibm.biz/BdqAQa.
Chaîne `updateStrategy`	Stratégie de mise à jour à utiliser pour les gestionnaires de files d'attente MultiInstance et NativeHA. Utilisez `RollingUpdate` pour activer les mises à jour automatiques chaque fois que la configuration du gestionnaire de files d'attente change. Utilisez `OnDelete` pour désactiver les mises à jour automatiques. Les modifications du gestionnaire de files d'attente ne seront appliquées que lorsque les pods sont supprimés (y compris les suppressions de pods déclenchées par des facteurs externes). La valeur par défaut est `RollingUpdate`. Nécessite MQ Operator 1.6.0 ou version supérieure.

.spec.queueManager.availability.tls

Paramètres TLS facultatifs permettant de configurer les communications sécurisées entre les répliques NativeHA. Requiert MQ Operator 1.5.0 ou une version ultérieure.

Apparaît dans :

- «.spec.queueManager.availability», à la page 150

Zone	Description
Chaîne `cipherSpec`	Nom du CipherSpec pour une connexion TLS NativeHA.
Chaîne `secretName`	Nom du secret Kubernetes.

.spec.queueManager.ini

Source des fichiers de configuration INI.

Apparaît dans :

- «.spec.queueManager», à la page 149

Zone	Description
`configMap` ConfigMapINISource	ConfigMap représente une mappe de configuration Kubernetes qui contient des informations INI.
`secret` SecretINISource	Secret représente un secret Kubernetes qui contient des informations INI.

.spec.queueManager.ini.configMap

ConfigMap représente une mappe de configuration Kubernetes qui contient des informations INI.

Apparaît dans :

- «.spec.queueManager.ini», à la page 151

Zone	Description
Tableau `items`	Clés dans la source Kubernetes, qui doivent être appliquées.
Chaîne `name`	Nom de la source Kubernetes.

.spec.queueManager.ini.secret

Secret représente un secret Kubernetes qui contient des informations INI.

Apparaît dans :

- «.spec.queueManager.ini», à la page 151

Zone	Description
Tableau `items`	Clés dans la source Kubernetes, qui doivent être appliquées.
Chaîne `name`	Nom de la source Kubernetes.

.spec.queueManager.livenessProbe

Paramètres qui contrôlent la sonde de non-défaillance.

Apparaît dans :

- «.spec.queueManager», à la page 149

Zone	Description
Entier `failureThreshold`	Nombre minimal d'échecs consécutifs nécessaire pour que l'on considère que la sonde a échoué après une réussite. La valeur par défaut est 1.

Zone	Description
Entier `initialDelaySeconds`	Nombre de secondes après le démarrage du conteneur avant que la sonde ne soit initiée. La valeur par défaut est 90 secondes pour SingleInstance. La valeur par défaut est 0 seconde pour les déploiements MultiInstance et NativeHA. 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). La valeur par défaut est 10 secondes.
Entier `successThreshold`	Nombre minimal de réussites consécutives nécessaire pour que l'on considère que la sonde a abouti après un échec. La valeur par défaut est 1.
Entier `timeoutSeconds`	Délai d'expiration de la sonde, en secondes. La valeur par défaut est 5 secondes. Pour plus d'informations : https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes.

.spec.queueManager.metrics

Paramètres pour les métriques de style Prometheus.

Apparaît dans :

- «.spec.queueManager», à la page 149

Zone	Description
Valeur booléenne `enabled`	Indique si un noeud final doit être activé pour permettre à Prometheus de collecter des métriques compatibles avec Prometheus. La valeur par défaut est true.

.spec.queueManager.mqsc

Source des fichiers de configuration MQSC.

Apparaît dans :

- «.spec.queueManager», à la page 149

Zone	Description
`configMap` ConfigMapMQSCSource	ConfigMap représente une mappe de configuration Kubernetes qui contient des informations MQSC.
`secret` SecretMQSCSource	Secret représente un Secret Kubernetes qui contient des informations MQSC.

.spec.queueManager.mqsc.configMap

ConfigMap représente une mappe de configuration Kubernetes qui contient des informations MQSC.

Apparaît dans :

- «.spec.queueManager.mqsc», à la page 152

Zone	Description
Tableau `items`	Clés dans la source Kubernetes, qui doivent être appliquées.
Chaîne `name`	Nom de la source Kubernetes.

.spec.queueManager.mqsc.secret

Secret représente un Secret Kubernetes qui contient des informations MQSC.

Apparaît dans :

- «.spec.queueManager.mqsc», à la page 152

Zone	Description
Tableau `items`	Clés dans la source Kubernetes, qui doivent être appliquées.
Chaîne `name`	Nom de la source Kubernetes.

.spec.queueManager.readinessProbe

Paramètres qui contrôlent la sonde de vigilance.

Apparaît dans :

- «.spec.queueManager», à la page 149

Zone	Description
Entier `failureThreshold`	Nombre minimal d'échecs consécutifs nécessaire pour que l'on considère que la sonde a échoué après une réussite. La valeur par défaut est 1.
Entier `initialDelaySeconds`	Nombre de secondes après le démarrage du conteneur avant que la sonde ne soit initiée. La valeur par défaut est 10 secondes pour SingleInstance. La valeur par défaut est 0 pour les déploiements MultiInstance et NativeHA. 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). La valeur par défaut est 5 secondes.
Entier `successThreshold`	Nombre minimal de réussites consécutives nécessaire pour que l'on considère que la sonde a abouti après un échec. La valeur par défaut est 1.
Entier `timeoutSeconds`	Délai d'expiration de la sonde, en secondes. La valeur par défaut est 3 secondes. Pour plus d'informations : https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes.

.spec.queueManager.recoveryLogs

Paramètres des journaux de reprise MQ . Requiert MQ Operator 2.4.0 ou version ultérieure.

Apparaît dans :

- «.spec.queueManager», à la page 149

Zone	Description
Entier `logFilePages`	Les données du journal de reprise sont stockées dans une série de fichiers. La taille de ces fichiers est définie en unité de pages de 4 ko.

.spec.queueManager.resources

Paramètres qui contrôlent les exigences relatives aux ressources.

Apparaît dans :

- «.spec.queueManager», à la page 149

Zone	Description
`limits` Limits	Paramètres d'unité centrale et de mémoire.
`requests` Requests	Paramètres d'unité centrale et de mémoire.

.spec.queueManager.resources.limits

Paramètres d'unité centrale et de mémoire.

Apparaît dans :

- «.spec.queueManager.resources», à la page 153

Zone	Description
`cpu`	
`memory`	

.spec.queueManager.resources.requests

Paramètres d'unité centrale et de mémoire.

Apparaît dans :

- «.spec.queueManager.resources», à la page 153

Zone	Description
`cpu`	
`memory`	

.spec.queueManager.route

Paramètres de la route du gestionnaire de files d'attente. Requiert MQ Operator 1.4.0 ou version ultérieure.

Apparaît dans :

- «.spec.queueManager», à la page 149

Zone	Description
Valeur booléenne `enabled`	Indique si la route doit être activée ou non. La valeur par défaut est true.

.spec.queueManager.startupProbe

Paramètres qui contrôlent la sonde de démarrage. Ne s'applique qu'aux déploiements MultiInstance et NativeHA. Requiert MQ Operator 1.5.0 ou une version ultérieure.

Apparaît dans :

- «.spec.queueManager», à la page 149

Zone	Description
Entier `failureThreshold`	Nombre minimal d'échecs consécutifs nécessaire pour considérer que la sonde a échoué. La valeur par défaut est 24.
Entier `initialDelaySeconds`	Nombre de secondes après le démarrage du conteneur avant que la sonde ne soit initiée. La valeur par défaut est 0 seconde. 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). La valeur par défaut est 5 secondes.
Entier `successThreshold`	Nombre minimal de réussites consécutives nécessaire pour considérer que la sonde a réussi. La valeur par défaut est 1.

Zone	Description
Entier `timeoutSeconds`	Délai d'expiration de la sonde, en secondes. La valeur par défaut est 5 secondes. Pour plus d'informations : https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes.

.spec.queueManager.storage

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

Apparaît dans :

- «.spec.queueManager», à la page 149

Zone	Description
Valeur booléenne `allowVolumeExpansion`	Indique s'il faut ou non autoriser l'extension des volumes.
Chaîne `defaultClass`	Classe de stockage à appliquer à tous les volumes persistants de ce gestionnaire de files d'attente par défaut. Les volumes persistants spécifiques peuvent définir leur propre classe de stockage qui remplacera ce paramètre de classe de stockage par défaut. Si `type of availability` a pour valeur `SingleInstance` ou `NativeHA`, la classe de stockage peut être de type ReadWriteOnce ou ReadWriteMany. Si `type of availability` est `MultiInstance`, la classe de stockage doit être de type ReadWriteMany.
Valeur booléenne `defaultDeleteClaim`	Indique si tous les volumes doivent être supprimés lorsque le gestionnaire de files d'attente est supprimé. Les volumes persistants spécifiques peuvent définir leur propre valeur pour deleteClaim qui remplacera ce paramètre defaultDeleteClaim. La valeur par défaut est false.
`persistedData` QueueManagerOptionalVolume	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` QueueManagerVolume	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	Détails du volume persistant pour les journaux de reprise MQ. Requis en cas d'utilisation d'un gestionnaire de files d'attente multi-instance.
`scratch` Rebut	Paramètres du volume éphémère Scratch du gestionnaire de files d'attente. Ce volume sera monté en tant que dossier'/run'sur le conteneur. Applicable uniquement si le système de fichiers racine est défini en lecture seule. Requiert MQ Operator 3.0.0 ou version ultérieure.
`tmp` Tmp	Paramètres du volume temporaire Tmp du gestionnaire de files d'attente. Ce volume sera monté sur le conteneur en tant que dossier'/tmp'. Les fichiers de données de diagnostic, tels que le fichier zip généré par la commande runmqras, seront créés dans ce volume. Applicable uniquement si le système de fichiers racine est défini en lecture seule. Requiert MQ Operator 3.0.0 ou version ultérieure.

.spec.queueManager.storage.persistedData

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.

Apparaît dans :

- «.spec.queueManager.storage», à la page 155

Zone	Description
Chaîne `class`	Classe d'archivage à utiliser pour ce volume. Valide uniquement si `type` a pour valeur `persistent-claim`. Si `type of availability` a pour valeur `SingleInstance` ou `NativeHA`, la classe de stockage peut être de type ReadWriteOnce ou ReadWriteMany. Si `type of availability` est `MultiInstance`, la classe de stockage doit être de type ReadWriteMany.
Valeur booléenne `deleteClaim`	Indique si ce volume doit être supprimé lorsque le gestionnaire de files d'attente est supprimé.
Valeur booléenne `enabled`	Indique si ce volume doit être activé en tant que volume distinct ou placé dans le volume `queueManager` par défaut, ou non. La valeur par défaut est false.
Chaîne `size`	Taille du volume persistant à transmettre à Kubernetes, incluant les unités SI. Valide uniquement si `type` a pour valeur `persistent-claim`. Par exemple, `2Gi`. La valeur par défaut est `2Gi`.
Chaîne `sizeLimit`	Limite de taille en cas d'utilisation d'un volume de type `ephemeral`. 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 `ephemeral` et que le système de fichiers racine est en lecture seule. Requiert MQ Operator 3.0.0 ou version ultérieure.
Chaîne `type`	Type de volume à utiliser. Choisissez `ephemeral` pour utiliser un stockage non persistant ou `persistent-claim` pour utiliser un volume persistant. La valeur par défaut est `persistent-claim`.

.spec.queueManager.storage.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 :

- «.spec.queueManager.storage», à la page 155

Zone	Description
Chaîne `class`	Classe d'archivage à utiliser pour ce volume. Valide uniquement si `type` a pour valeur `persistent-claim`. Si `type of availability` a pour valeur `SingleInstance` ou `NativeHA`, la classe de stockage peut être de type ReadWriteOnce ou ReadWriteMany. Si `type of availability` est `MultiInstance`, la classe de stockage doit être de type ReadWriteMany.
Valeur booléenne `deleteClaim`	Indique si ce volume doit être supprimé lorsque le gestionnaire de files d'attente est supprimé.
Chaîne `size`	Taille du volume persistant à transmettre à Kubernetes, incluant les unités SI. Valide uniquement si `type` a pour valeur `persistent-claim`. Par exemple, `2Gi`. La valeur par défaut est `2Gi`.
Chaîne `sizeLimit`	Limite de taille en cas d'utilisation d'un volume de type `ephemeral`. 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 `ephemeral` et que le système de fichiers racine est en lecture seule. Requiert MQ Operator 3.0.0 ou version ultérieure.

Zone	Description
Chaîne `type`	Type de volume à utiliser. Choisissez `ephemeral` pour utiliser un stockage non persistant ou `persistent-claim` pour utiliser un volume persistant. La valeur par défaut est `persistent-claim`.

.spec.queueManager.storage.recoveryLogs

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 :

- «.spec.queueManager.storage», à la page 155

Zone	Description
Chaîne `class`	Classe d'archivage à utiliser pour ce volume. Valide uniquement si `type` a pour valeur `persistent-claim`. Si `type of availability` a pour valeur `SingleInstance` ou `NativeHA`, la classe de stockage peut être de type ReadWriteOnce ou ReadWriteMany. Si `type of availability` est `MultiInstance`, la classe de stockage doit être de type ReadWriteMany.
Valeur booléenne `deleteClaim`	Indique si ce volume doit être supprimé lorsque le gestionnaire de files d'attente est supprimé.
Valeur booléenne `enabled`	Indique si ce volume doit être activé en tant que volume distinct ou placé dans le volume `queueManager` par défaut, ou non. La valeur par défaut est false.
Chaîne `size`	Taille du volume persistant à transmettre à Kubernetes, incluant les unités SI. Valide uniquement si `type` a pour valeur `persistent-claim`. Par exemple, `2Gi`. La valeur par défaut est `2Gi`.
Chaîne `sizeLimit`	Limite de taille en cas d'utilisation d'un volume de type `ephemeral`. 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 `ephemeral` et que le système de fichiers racine est en lecture seule. Requiert MQ Operator 3.0.0 ou version ultérieure.
Chaîne `type`	Type de volume à utiliser. Choisissez `ephemeral` pour utiliser un stockage non persistant ou `persistent-claim` pour utiliser un volume persistant. La valeur par défaut est `persistent-claim`.

.spec.queueManager.storage.scratch

Paramètres du volume éphémère Scratch du gestionnaire de files d'attente. Ce volume sera monté en tant que dossier'/run'sur le conteneur. Applicable uniquement si le système de fichiers racine est défini en lecture seule. Requiert MQ Operator 3.0.0 ou version ultérieure.

Apparaît dans :

- «.spec.queueManager.storage», à la page 155

Zone	Description
Chaîne `sizeLimit`	Limite de taille du volume éphémère, y compris les unités SI. Par exemple, `2Gi`. Valide uniquement lorsque le système de fichiers racine est défini en lecture seule. Requiert MQ Operator 3.0.0 ou version ultérieure.

.spec.queueManager.storage.tmp

Paramètres du volume temporaire Tmp du gestionnaire de files d'attente. Ce volume sera monté sur le conteneur en tant que dossier'/tmp'. Les fichiers de données de diagnostic, tels que le fichier zip généré par la commande runmqras, seront créés dans ce volume. Applicable uniquement si le système de fichiers racine est défini en lecture seule. Requiert MQ Operator 3.0.0 ou version ultérieure.

Apparaît dans :

- «.spec.queueManager.storage», à la page 155

Zone	Description
Chaîne `sizeLimit`	Limite de taille du volume éphémère, y compris les unités SI. Par exemple, `2Gi`. Valide uniquement lorsque le système de fichiers racine est défini en lecture seule. Requiert MQ Operator 3.0.0 ou version ultérieure.

.spec.securityContext

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

Apparaît dans :

- «.spec», à la page 145

Zone	Description
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.
Valeur booléenne `initVolumeAsRoot`	Elle a un impact sur le contexte de sécurité utilisé par le conteneur qui initialise le volume persistant. Définissez la valeur `true` si vous utilisez un fournisseur de stockage qui exige que vous soyez l'utilisateur racine pour pouvoir accéder aux volumes nouvellement mis à disposition. La valeur `true` a un impact sur l'objet Contraintes de contexte de sécurité (SCC) que vous pouvez utiliser, et le démarrage du gestionnaire de files d'attente peut échouer si vous n'êtes pas autorisé à utiliser un objet SCC autorisant l'utilisateur racine. La valeur par défaut est false. Pour plus d'informations, voir https://docs.openshift.com/container-platform/latest/authentication/managing-security-context-constraints.html.
Valeur booléenne `readOnlyRootFilesystem`	Indique si les paramètres du système de fichiers racine en lecture seule doivent être activés pour le gestionnaire de files d'attente. La valeur par défaut est false. Requiert MQ Operator 3.0.0 ou version ultérieure.
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.

.spec.telemetry

Paramètres de configuration d'Open Telemetry. Requiert MQ Operator 2.2.0 ou version ultérieure.

Apparaît dans :

- «.spec», à la page 145

Zone	Description
`tracing` Fonction de trace	Paramètres du traçage Open Telemetry.

.spec.telemetry.tracing

Paramètres du traçage Open Telemetry.

Apparaît dans :

- «.spec.telemetry», à la page 158

Zone	Description
`instana` Instana	Paramètres du traçage Instana.

.spec.telemetry.tracing.instana

Paramètres du traçage Instana.

Apparaît dans :

- «.spec.telemetry.tracing», à la page 159

Zone	Description
Chaîne `agentHost`	Nom d'hôte de l'agent Instana auquel envoyer les données de trace. Cela ne devrait pas inclure de protocole.
Valeur booléenne `enabled`	Indique si le traçage d'Instana doit être activé ou non. La valeur par défaut est false.
Chaîne `protocol`	Protocole à utiliser pour la communication avec l'agent Instana. http et https sont pris en charge.

.spec.template

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 :

- «.spec», à la page 145

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.

.spec.tracing

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

Apparaît dans :

- «.spec», à la page 145

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

Zone	Description
`collector` TracingCollector	Dans Cloud Pak for Integration uniquement, vous pouvez configurer des paramètres pour le collecteur de trace facultatif.
Valeur booléenne `enabled`	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. La valeur par défaut est false.
Chaîne `namespace`	Espace de nom dans lequel le tableau de bord des opérations de Cloud Pak for Integration est installé.

.spec.tracing.agent

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

Apparaît dans :

- «.spec.tracing», à la page 159

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. La valeur par défaut est `IfNotPresent`.
`livenessProbe` TracingProbe	Paramètres qui contrôlent la sonde de non-défaillance.
`readinessProbe` TracingProbe	Paramètres qui contrôlent la sonde de vigilance.

.spec.tracing.agent.livenessProbe

Paramètres qui contrôlent la sonde de non-défaillance.

Apparaît dans :

- «.spec.tracing.agent», à la page 160

Zone	Description
Entier `failureThreshold`	Nombre minimal d'échecs consécutifs nécessaire pour que l'on considère que la sonde a échoué après une réussite. La valeur par défaut est 1.
Entier `initialDelaySeconds`	Nombre de secondes après le démarrage du conteneur avant que des sondes de non-défaillance ne soient initiées. La valeur par défaut est 10 secondes. 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). La valeur par défaut est 10 secondes.
Entier `successThreshold`	Nombre minimal de réussites consécutives nécessaire pour que l'on considère que la sonde a abouti après un échec. La valeur par défaut est 1.
Entier `timeoutSeconds`	Délai d'expiration de la sonde, en secondes. La valeur par défaut est 2 secondes. Pour plus d'informations : https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes.

.spec.tracing.agent.readinessProbe

Paramètres qui contrôlent la sonde de vigilance.

Apparaît dans :

- «.spec.tracing.agent», à la page 160

Zone	Description
Entier `failureThreshold`	Nombre minimal d'échecs consécutifs nécessaire pour que l'on considère que la sonde a échoué après une réussite. La valeur par défaut est 1.
Entier `initialDelaySeconds`	Nombre de secondes après le démarrage du conteneur avant que des sondes de non-défaillance ne soient initiées. La valeur par défaut est 10 secondes. 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). La valeur par défaut est 10 secondes.
Entier `successThreshold`	Nombre minimal de réussites consécutives nécessaire pour que l'on considère que la sonde a abouti après un échec. La valeur par défaut est 1.
Entier `timeoutSeconds`	Délai d'expiration de la sonde, en secondes. La valeur par défaut est 2 secondes. Pour plus d'informations : https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes.

.spec.tracing.collector

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

Apparaît dans :

- «.spec.tracing», à la page 159

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. La valeur par défaut est `IfNotPresent`.
`livenessProbe` TracingProbe	Paramètres qui contrôlent la sonde de non-défaillance.
`readinessProbe` TracingProbe	Paramètres qui contrôlent la sonde de vigilance.

.spec.tracing.collector.livenessProbe

Paramètres qui contrôlent la sonde de non-défaillance.

Apparaît dans :

- «.spec.tracing.collector», à la page 161

Zone	Description
Entier `failureThreshold`	Nombre minimal d'échecs consécutifs nécessaire pour que l'on considère que la sonde a échoué après une réussite. La valeur par défaut est 1.

Zone	Description
Entier `initialDelaySeconds`	Nombre de secondes après le démarrage du conteneur avant que des sondes de non-défaillance ne soient initiées. La valeur par défaut est 10 secondes. 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). La valeur par défaut est 10 secondes.
Entier `successThreshold`	Nombre minimal de réussites consécutives nécessaire pour que l'on considère que la sonde a abouti après un échec. La valeur par défaut est 1.
Entier `timeoutSeconds`	Délai d'expiration de la sonde, en secondes. La valeur par défaut est 2 secondes. Pour plus d'informations : https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes.

.spec.tracing.collector.readinessProbe

Paramètres qui contrôlent la sonde de vigilance.

Apparaît dans :

- «.spec.tracing.collector», à la page 161

Zone	Description
Entier `failureThreshold`	Nombre minimal d'échecs consécutifs nécessaire pour que l'on considère que la sonde a échoué après une réussite. La valeur par défaut est 1.
Entier `initialDelaySeconds`	Nombre de secondes après le démarrage du conteneur avant que des sondes de non-défaillance ne soient initiées. La valeur par défaut est 10 secondes. 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). La valeur par défaut est 10 secondes.
Entier `successThreshold`	Nombre minimal de réussites consécutives nécessaire pour que l'on considère que la sonde a abouti après un échec. La valeur par défaut est 1.
Entier `timeoutSeconds`	Délai d'expiration de la sonde, en secondes. La valeur par défaut est 2 secondes. Pour plus d'informations : https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes.

.spec.web

Paramètres pour le serveur Web MQ.

Apparaît dans :

- «.spec», à la page 145

Zone	Description
`console` Console	Paramètres de la console Web MQ . Requiert MQ Operator 3.0.0 ou version ultérieure.
Valeur booléenne `enabled`	Indique si le serveur Web doit être activé ou non. La valeur par défaut est false.
`manualConfig` ManualConfig	Paramètres permettant de fournir la configuration XML du serveur Web. Requiert MQ Operator 3.0.0 ou version ultérieure.

.spec.web.console

Paramètres de la console Web MQ . Requiert MQ Operator 3.0.0 ou version ultérieure.

Apparaît dans :

- «.spec.web», à la page 162

Zone	Description
`authentication` Authentification	Paramètres d'authentification de la console Web MQ . Requiert MQ Operator 3.0.0 ou version ultérieure.
`authorization` Autorisation	Paramètres d'autorisation pour la console Web MQ . Requiert MQ Operator 3.0.0 ou version ultérieure.

.spec.web.console.authentication

Paramètres d'authentification de la console Web MQ . Requiert MQ Operator 3.0.0 ou version ultérieure.

Apparaît dans :

- «.spec.web.console», à la page 163

Zone	Description
Chaîne `provider`	Fournisseur d'authentification à utiliser pour la console Web MQ . Utilisez `integration-keycloak` pour utiliser la connexion unique avec l'interface utilisateur de la plateforme Cloud Pak for Integration (Keycloak). La valeur par défaut est `integration-keycloak` si vous utilisez une licence Cloud Pak for Integration ou `manual` si vous utilisez une licence MQ . Utilisez `manual` si vous souhaitez fournir votre propre configuration.

.spec.web.console.authorization

Paramètres d'autorisation pour la console Web MQ . Requiert MQ Operator 3.0.0 ou version ultérieure.

Apparaît dans :

- «.spec.web.console», à la page 163

Zone	Description
Chaîne `provider`	Fournisseur d'autorisation à utiliser pour la console Web MQ . Utilisez `integration-keycloak` pour utiliser les rôles fournis par Cloud Pak for Integration Keycloak. Utilisez `manual` si vous souhaitez fournir votre propre configuration. La valeur par défaut est `integration-keycloak` si vous utilisez une licence Cloud Pak for Integration ou `manual` si vous utilisez une licence MQ .

.spec.web.manualConfig

Paramètres permettant de fournir la configuration XML du serveur Web. Requiert MQ Operator 3.0.0 ou version ultérieure.

Apparaît dans :

- «.spec.web», à la page 162

Zone	Description
`configMap` ConfigMap	ConfigMap représente un objet Kubernetes ConfigMap qui contient la configuration XML du serveur Web.

Zone	Description
`secret` Secret	Secret représente un secret Kubernetes qui contient la configuration XML du serveur Web. L'utilisation d'un secret protège toutes les données d'identification dans la couche Kubernetes , mais il est possible que les outils de surveillance ou de traitement des incidents exposent le fichier sous-jacent de manière non sécurisée. Pour améliorer la sécurité, codez les données d'identification à l'aide de "securityUtility.

.spec.web.manualConfig.configMap

ConfigMap représente un objet Kubernetes ConfigMap qui contient la configuration XML du serveur Web.

Apparaît dans :

- «.spec.web.manualConfig», à la page 163

Zone	Description
Chaîne `name`	Nom de la source Kubernetes.

.spec.web.manualConfig.secret

Secret représente un secret Kubernetes qui contient la configuration XML du serveur Web. L'utilisation d'un secret protège toutes les données d'identification dans la couche Kubernetes , mais il est possible que les outils de surveillance ou de traitement des incidents exposent le fichier sous-jacent de manière non sécurisée. Pour améliorer la sécurité, codez les données d'identification à l'aide de "securityUtility.

Apparaît dans :

- «.spec.web.manualConfig», à la page 163

Zone	Description
Chaîne `name`	Nom de la source Kubernetes.

.statut

Etat observé pour le gestionnaire de files d'attente.

Apparaît dans :

- «QueueManager», à la page 145

Zone	Description
Chaîne `adminUiUrl`	URL de l'interface utilisateur d'administration.
`availability` Availability	Statut de disponibilité du gestionnaire de files d'attente.
Tableau `conditions` QueueManagerStatusCondition	Les conditions représentent les dernières observations disponibles de l'état du gestionnaire de files d'attente.
Tableau `endpoints` QueueManagerStatusEndpoint	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.
`metadata` Métadonnées	Les métadonnées représentent des informations supplémentaires pour le gestionnaire de files d'attente, notamment le statut d'intégration-Keycloak .
Chaîne `name`	Nom du gestionnaire de files d'attente.
Chaîne `phase`	Phase de l'état du gestionnaire de files d'attente.

Zone	Description
`versions` QueueManagerStatusVersion	Version de MQ utilisée et autres versions disponibles depuis le registre autorisé IBM.

.status.availability

Statut de disponibilité du gestionnaire de files d'attente.

Apparaît dans :

- «.statut», à la page 164

Zone	Description
Valeur booléenne `initialQuorumEstablished`	Indique si un quorum initial a été établi pour NativeHA.

.status.conditions

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

Apparaît dans :

- «.statut», à la page 164

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

.status.endpoints

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

Apparaît dans :

- «.statut», à la page 164

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

.status.metadata

Les métadonnées représentent des informations supplémentaires pour le gestionnaire de files d'attente, notamment le statut d'intégration-Keycloak .

Apparaît dans :

- «.statut», à la page 164

Zone	Description
`integrationKeycloak` IntegrationKeycloak	QueueManagerStatusIntegrationKeycloak définit le statut d'intégration-Keycloak pour QueueManager.

.status.metadata.integrationKeycloak

QueueManagerStatusIntegrationKeycloak définit le statut d'intégration-Keycloak pour QueueManager.

Apparaît dans :

- «.status.metadata», à la page 165

Zone	Description
Chaîne `clientName`	

.status.versions

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

Apparaît dans :

- «.statut», à la page 164

Zone	Description
`available` QueueManagerStatusVersionAvailable	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.

.status.versions.available

Autres versions de MQ disponibles depuis le registre autorisé IBM.

Apparaît dans :

- «.status.versions», à la page 166

Zone	Description
Tableau `channels`	Canaux disponibles pour la mise à jour automatique de la version de MQ.
Tableau `versions` Versions	Versions spécifiques de MQ qui sont disponibles.

.status.versions.available.versions

QueueManagerStatusVersion définit une version de MQ.

Apparaît dans :

- «.status.versions.available», à la page 166

Zone	Description
Tableau `licenses` Licences	Licences applicables à cette version de QueueManager.
Chaîne `name`	Version `name` pour cette version de QueueManager. Valeurs valides pour la zone `spec.version`.

.status.versions.available.versions.licenses

QueueManagerStatusLicense définit une licence.

Apparaît dans :

- «.status.versions.available.versions», à la page 166

Zone	Description
Chaîne `displayName`	Nom d'affichage de la licence.
Chaîne `link`	Lien vers le contenu de la licence.
Valeur booléenne `matchesCurrentType`	Indique si la licence correspond ou non au type de licence actuellement utilisé.
Chaîne `name`	Nom de la licence.

Conditions de statut de QueueManager (mq.ibm.com/v1beta1)

Les zones **`status.conditions`** sont mises à jour pour refléter la condition de la ressource `QueueManager`. En général, les conditions décrivent des situations anormales. Un gestionnaire de files d'attente dans un état sain et prêt n'a pas de conditions **`Error`** ou **`Pending`** . Il peut comporter des conditions **`Warning`** de recommandation.

Les conditions suivantes sont définies pour une ressource `QueueManager` :

Tableau 2. Conditions de statut du gestionnaire de files d'attente

Composant	Type de condition	Code raison	Avertissement message
QueueManager[3]	Bloquée	OperatorDependency	Pour effectuer l'installation, cette instance nécessite que Keycloak soit configuré par [IBM Cloud Pak for Integration]. Cette instance restera à l'état [En attente] jusqu'à ce que Keycloak soit signalé comme [KeycloakReady] dans la ressource Cp4iServicesBinding pour ce QueueManager.
			Pour l'installation, cette instance requiert l'opérateur [IBM IAM]. Cette instance reste à l'état [Bloqué] jusqu'à ce que l'opérateur soit installé par [IBM Cloud Pak foundational services].
	En attente	Création	Le gestionnaire de files d'attente MQ est en cours de déploiement
	En attente	OidcPending	Le gestionnaire de files d'attente MQ attend l'enregistrement du client OIDC
	En attente	Arrêté	Le gestionnaire de files d'attente MQ a été arrêté car l'annotation'mq.ibm.com/stop'est présente et définie sur'true'dans la définition QueueManager . Une fois arrêté, le nombre de répliques QueueManager StatefulSet est défini sur zéro, ce qui supprime tous les pods de gestionnaire de files d'attente MQ .
	Erreur	Echec	Echec du déploiement du gestionnaire de files d'attente MQ
	Avertissement	UnsupportedVersion	Un facteur a été installé par un opérateur qui n'est pas pris en charge sur la version OCP *<ocp_version>*. Ce facteur n'est pas pris en charge.
	Avertissement	Support pour CP4I-LTS	Un CP4I-LTS opérande *< mq_version>* a été installé mais est géré par un opérateur qui ne se qualifie pas pour la durée de prise en charge étendue. Cet opérande ne correspond pas à la durée de prise en charge étendue.
	Avertissement	Support pour CP4I-LTS	Un CP4I-LTS opérande *< mq_version>* a été installé, mais la version d'OCP *< ocp_version>* ne se qualifie pas pour la durée de prise en charge étendue. Cet opérande ne correspond pas à la durée de prise en charge étendue.

[3] Les conditions `Creating` et `Failed` surveillent la progression globale du déploiement du gestionnaire de files d'attente. Si vous utilisez une licence IBM Cloud Pak for Integration et que la console Web est activée, la condition `OidcPending` consigne le statut du gestionnaire de files d'attente en attendant la fin de l'enregistrement du client OIDC avec IAM.

Tableau 2. Conditions de statut du gestionnaire de files d'attente (suite)			
Composant	**Type de condition**	**Code raison**	**Avertissement message**
Pod[4]	En attente	PodPending	Pod pour le gestionnaire de files d'attente MQ en cours de déploiement
	Erreur	PodFailed	Pod pour le gestionnaire de files d'attente MQ en cours de déploiement
Mémoire[5]	En attente	StoragePending	Le stockage du gestionnaire de files d'attente MQ est mis à disposition
	Avertissement	StorageEphemeral	Utilisation du stockage temporaire pour un gestionnaire de files d'attente MQ de production
	Avertissement	StorageExpansionen attente	L'extension de volume est en attente pour les réservations de volume persistant suivantes [< liste de pvcs>]
	Avertissement	StorageMismatch	Les tailles de stockage définies dans la ressource QueueManager ne correspondent pas à la capacité d'une ou de plusieurs PVC mises à disposition [< liste de pvcs>]. L'extension AllowVolumeest définie sur false dans la ressource QueueManager , de sorte que l'opérateur MQ ne tente pas de réconcilier ces différences.
	Erreur	StorageFailed	Echec de la mise à disposition du stockage du gestionnaire de files d'attente MQ

Linux Annotations de licence lors de la génération de votre propre image de conteneur IBM MQ

Les annotations de licence vous permettent de suivre l'utilisation en fonction des limites définies sur le conteneur, plutôt que sur la machine sous-jacente. Vous configurez vos clients pour déployer le conteneur avec des annotations spécifiques que IBM License Service utilise ensuite pour suivre l'utilisation.

Lors du déploiement d'une image de conteneur IBM MQ auto-générée, il existe deux approches communes pour l'octroi de licences :

- Licence sur l'ensemble de la machine qui exécute le conteneur.

[4] Les conditions du pod surveillent le statut des pods pendant le déploiement d'un gestionnaire de files d'attente. Si vous voyez une condition PodFailed, la condition globale du gestionnaire de files d'attente sera également définie sur Failed.

[5] Les conditions de stockage surveillent la progression (condition StoragePending) des demandes de création de volumes pour le stockage permanent, et signalent les erreurs de liaison et autres échecs. Les conditions de stockage surveillent également la progression des extensions de volume et alertent des non-concordances entre les tailles de stockage définies dans la définition du gestionnaire de files d'attente et la taille des réservations de volume persistant déployées. Si une erreur se produit lors de l'allocation de stockage, la condition StorageFailed est ajoutée à la liste des conditions et la condition globale du gestionnaire de files d'attente est définie sur Failed.

- Licence sur le conteneur en fonction des limites associées.

Les deux options sont disponibles pour les clients, et des détails supplémentaires sont disponibles sur la page des licences de conteneur IBM sous Passport Advantage.

Si le conteneur IBM MQ doit être concédé sous licence en fonction des limites de conteneur, IBM License Service doit être installé pour suivre l'utilisation. Pour plus d'informations sur les environnements pris en charge et les instructions d'installation, consultez la page ibm-licensing-operator sur GitHub.

IBM License Service est installé sur le cluster Kubernetes où le conteneur IBM MQ est déployé, et les annotations de pod sont utilisées pour le suivi de l'utilisation. Par conséquent, les clients doivent déployer le pod avec des annotations spécifiques que IBM License Service utilise ensuite. En fonction de votre autorisation d'utilisation et des capacités déployées dans le conteneur, utilisez une ou plusieurs des annotations suivantes.

Remarque : La plupart des annotations contiennent l'une des lignes suivantes ou les deux:

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

Vous devez éditer ces lignes avant d'utiliser l'annotation:

- Pour `productChargedContainers`, vous devez choisir `"All"`ou remplacer le nom réel du conteneur.
- Pour `productMetric`, vous devez choisir l'une des valeurs proposées.

Annotations à utiliser avec une autorisation d'utilisation du produit IBM MQ

Si vous disposez d'une autorisation d'utilisation du produit IBM MQ , sélectionnez l'annotation ci-dessous qui correspond à l'autorisation que vous avez achetée et que vous souhaitez utiliser.

- «IBM MQ», à la page 172
- «IBM MQ Avancé», à la page 172
- «IBM MQ pour l'environnement hors production», à la page 172
- «IBM MQ Advanced pour l'environnement de non-production», à la page 172
- «IBM MQ Advanced pour les développeurs», à la page 172

Les annotations IBM MQ à utiliser avec les configurations à haute disponibilité multi-instance IBM MQ sont les suivantes. Voir aussi «Sélection des annotations correctes pour les configurations à haute disponibilité», à la page 171.

- «Instance multiple de conteneur IBM MQ», à la page 173
- «IBM MQ Advanced Container multi-instance», à la page 173
- «IBM MQ Container Multi-Instance pour l'environnement de non-production», à la page 173
- «IBM MQ Advanced Container Multi Instance for Non-Production Environment», à la page 173

Annotations à utiliser avec l'autorisation d'utilisation du produit CP4I

Si vous disposez d'une autorisation IBM Cloud Pak for Integration (CP4I), sélectionnez l'annotation ci-dessous qui correspond à l'autorisation que vous avez achetée et que vous souhaitez utiliser.

- «IBM MQ avec autorisation d'utilisation CP4I», à la page 173
- «Autorisation d'utilisation de IBM MQ Advanced with CP4I», à la page 173
- «IBM MQ for Non-Production Environment avec autorisation d'utilisation CP4I», à la page 173
- «Autorisation d'utilisation d' IBM MQ Advanced for Non-Production Environment avec CP4I», à la page 174

Les annotations CP4I à utiliser avec les configurations à haute disponibilité multi-instance IBM MQ sont les suivantes. Voir aussi «Sélection des annotations correctes pour les configurations à haute disponibilité», à la page 171.

- «IBM MQ Container Multi Instance avec autorisation d'utilisation CP4I», à la page 174
- «IBM MQ Advanced Container Multi Instance avec autorisation d'utilisation CP4I», à la page 174
- «IBM MQ Container Multi Instance for Non-Production Environment avec autorisation d'utilisation CP4I», à la page 174
- «IBM MQ Advanced Container Multi Instance for Non-Production Environment avec autorisation d'utilisation CP4I», à la page 174

Sélection des annotations correctes pour les configurations à haute disponibilité

IBM MQ Multi-instance

Lorsque vous déployez une paire de gestionnaires de files d'attente dans une configuration à haute disponibilité multi-instance IBM MQ , vous devez utiliser la même annotation sur les deux instances. L'une des annotations suivantes doit être sélectionnée, en fonction de l'autorisation d'utilisation achetée:

- Autorisation d'utilisation autonome IBM MQ ou IBM MQ Advanced
  - «Instance multiple de conteneur IBM MQ», à la page 173
  - «IBM MQ Advanced Container multi-instance», à la page 173
  - «IBM MQ Container Multi-Instance pour l'environnement de non-production», à la page 173
  - «IBM MQ Advanced Container Multi Instance for Non-Production Environment», à la page 173
- IBM Cloud Pak for Integration autorisation
  - «IBM MQ Container Multi Instance avec autorisation d'utilisation CP4I», à la page 174
  - «IBM MQ Advanced Container Multi Instance avec autorisation d'utilisation CP4I», à la page 174
  - «IBM MQ Container Multi Instance for Non-Production Environment avec autorisation d'utilisation CP4I», à la page 174
  - «IBM MQ Advanced Container Multi Instance for Non-Production Environment avec autorisation d'utilisation CP4I», à la page 174

Lorsqu'ils sont utilisés avec les autorisations d'utilisation IBM Cloud Pak for Integration , les ratios d'autorisation dans les annotations garantissent que la consommation d'autorisations correcte est enregistrée. Lorsqu'elles sont utilisées avec des droits IBM MQ ou IBM MQ Advanced autonomes, les annotations signalées dans le License Service pour chaque instance doivent être mappées aux parties d'autorisation IBM MQ comme suit:

- Instances multiples IBM MQ Advanced container
  - 1 x IBM MQ Advanced **et** 1 x IBM MQ Advanced High Availability Replica **ou**
  - 2 x IBM MQ Advanced[6]
- IBM MQ Advanced container Multi-instance pour l'environnement de non-production
  - 1 x IBM MQ Advanced **et** 1 x IBM MQ Advanced High Availability Replica **ou**
  - 2 x IBM MQ Advanced pour l'environnement hors production)[6]
- Instance multiple de conteneur IBM MQ
  - 1 x IBM MQ **et** 1 x IBM MQ High Availability Replica **ou**
  - 2 x IBM MQ[6]
- IBM MQ Container Multi-Instance pour l'environnement de non-production
  - 1 x IBM MQ **et** 1 x IBM MQ High Availability Replica **ou**
  - 2 x IBM MQ pour l'environnement hors production) [6]

IBM MQ Native HA

[6] Cette option d'autorisation est sous-optimale et ne doit être utilisée que si aucune autorisation d'utilisation de la pièce High Availability Replica appropriée n'est disponible.

Si vous déployez trois gestionnaires de files d'attente dans un quorum Native HA, seule l'instance active consomme des droits. Toutes les instances doivent avoir la même annotation. L'une des options suivantes doit être sélectionnée, en fonction de l'autorisation d'utilisation achetée:

- Autorisation d'utilisation autonome IBM MQ ou IBM MQ Advanced
  - «IBM MQ Avancé», à la page 172
  - «IBM MQ Advanced pour l'environnement de non-production», à la page 172
- IBM Cloud Pak for Integration autorisation
  - «Autorisation d'utilisation de IBM MQ Advanced with CP4I», à la page 173
  - «Autorisation d'utilisation d' IBM MQ Advanced for Non-Production Environment avec CP4I», à la page 174

Annotations

Le reste de cette rubrique détaille le contenu de chaque annotation.

IBM MQ

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

IBM MQ Avancé

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

IBM MQ pour l'environnement hors production

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

IBM MQ Advanced pour l'environnement de non-production

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

IBM MQ Advanced pour les développeurs

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

Instance multiple de conteneur IBM MQ

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

IBM MQ Advanced Container multi-instance

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

IBM MQ Container Multi-Instance pour l'environnement de non-production

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

IBM MQ Advanced Container Multi Instance for Non-Production Environment

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

IBM MQ avec autorisation d'utilisation CP4I

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

Autorisation d'utilisation de IBM MQ Advanced with CP4I

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

IBM MQ for Non-Production Environment avec autorisation d'utilisation CP4I

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

Autorisation d'utilisation d' IBM MQ Advanced for Non-Production Environment avec CP4I

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

IBM MQ Container Multi Instance avec autorisation d'utilisation CP4I

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

IBM MQ Advanced Container Multi Instance avec autorisation d'utilisation CP4I

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

IBM MQ Container Multi Instance for Non-Production Environment avec autorisation d'utilisation CP4I

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

IBM MQ Advanced Container Multi Instance for Non-Production Environment avec autorisation d'utilisation CP4I

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

OpenShift CP4I Kubernetes IBM MQ Advanced for Developers image de conteneur

Une image de conteneur prégénérée est disponible pour IBM MQ Advanced for Developers. Cette image est disponible dans le IBM Container Registry. Cette image peut être utilisée avec Docker, Podman, Kuberneteset d'autres environnements de conteneur.

Images disponibles

Les images IBM MQ sont stockées dans IBM Container Registry:

- IBM MQ Advanced for Developers 9.4.0.0: `icr.io/ibm-messaging/mq:9.4.0.0-r1`

Aide-mémoire

- Licence :
  - Référence de licence pour mq.ibm.com/v1beta1 et Apache License 2.0. Notez que la licence IBM MQ Advanced for Developers n'autorise pas de distribution supplémentaire et que les dispositions limitent l'utilisation à une machine de développeur.
- Emplacement de la résolution des problèmes:
  - GitHub
- Disponible pour les architectures d'UC suivantes:
  - amd64
  - s390x
  - ppc64le

Utilisation

Exécutez IBM MQ Advanced for Developers dans un conteneur.

Voir la documentation sur l'utilisation pour plus de détails sur l'exécution d'un conteneur.

Pour pouvoir utiliser l'image, vous devez accepter les dispositions de la licence IBM MQ en définissant la variable d'environnement **LICENSE** .

Variables d'environnement prises en charge

LANG
Définissez la langue dans laquelle vous souhaitez que la licence soit imprimée.

Licence
Définissez `accept` pour accepter les conditions de licence IBM MQ Advanced for Developers .

Définissez `view` pour afficher les conditions de licence.

Deprecated **MQ_ADMIN_PASSWORD**
Indiquez le mot de passe de l'administrateur.

Doit comporter au moins 8 caractères.

Il n'existe pas de mot de passe par défaut pour l'administrateur.

V 9.4.0 V 9.4.0 Depuis la IBM MQ 9.4.0, cette variable n'est plus fournie. L'exemple de fichier YAML dans cette rubrique montre comment vous pouvez créer cette variable vous-même et la sécuriser avec un secret.

Deprecated **MQ_APP_PASSWORD**
Indiquez le mot de passe de l'utilisateur de l'application.

Si cette option est définie, le canal **DEV.APP.SVRCONN** devient sécurisé et autorise uniquement les connexions qui fournissent un ID utilisateur et un mot de passe valides.

Doit comporter au moins 8 caractères.

Il n'existe pas de mot de passe par défaut pour l'utilisateur de l'application.

V 9.4.0 V 9.4.0 Depuis la IBM MQ 9.4.0, cette variable n'est plus fournie. L'exemple de fichier YAML dans cette rubrique montre comment vous pouvez créer cette variable vous-même et la sécuriser avec un secret.

MQ_DEV
Définissez `false` pour arrêter la création des objets par défaut.

MQ_ENABLE_METRICS
Définissez `true` pour générer des métriques Prometheus pour votre gestionnaire de files d'attente.

MQ_LOGGING_CONSOLE_SOURCE
Spécifiez une liste de sources séparées par des virgules pour les journaux qui sont mis en miroir à l'emplacement **stdout** du conteneur.

Les valeurs valides sont `qmgr`, `web` et `mqsc`.

La valeur par défaut est `qmgr, web`.

La valeur facultative est `mqsc`. Cette option peut être utilisée pour refléter le contenu de `autocfgmqsc.LOG` dans le journal du conteneur.

MQ_LOGGING_CONSOLE_FORMAT
Modifiez le format des journaux imprimés à l'emplacement **stdout** du conteneur.

Définissez `basic` pour utiliser un format simple lisible par l'utilisateur. Il s'agit de la valeur par défaut.

Définissez `json` pour utiliser le format JSON (un objet JSON sur chaque ligne).

MQ_LOGGING_CONSOLE_EXCLUDE_ID
Spécifiez une liste d'ID de message séparés par des virgules pour les messages de journal qui sont exclus.

Les messages de journal apparaissent toujours dans le fichier journal sur le disque, mais ne sont pas imprimés à l'emplacement **stdout** du conteneur.

La valeur par défaut est `AMQ5041I,AMQ5052I,AMQ5051I,AMQ5037I,AMQ5975I`.

mq_qmgr_name
Définissez le nom avec lequel vous souhaitez que votre gestionnaire de files d'attente soit créé.

Pour plus d'informations sur la configuration de développeur par défaut prise en charge par l'image IBM MQ Advanced for Developers , voir la documentation sur la configuration de développeur par défaut.

Exemple de fichier YAML de gestionnaire de files d'attente qui décrit comment spécifier des mots de passe pour les utilisateurs admin et app

Pour les utilisateurs des ID utilisateur **admin** et **app** , vous devez fournir des mots de passe lors du déploiement d'un gestionnaire de files d'attente à l'aide de la licence `Development` . Voici un exemple de fichier YAML de gestionnaire de files d'attente qui explique comment utiliser IBM MQ Operator.

La commande suivante crée un secret contenant des mots de passe pour les utilisateurs **admin** et **app** .

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

Le fichier YAML suivant utilise ces mots de passe lors du déploiement d'un gestionnaire de files d'attente.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: qm-dev
spec:
  license:
    accept: false
    license: L-CLXQ-ADXTK3
    use: Development
  web:
    enabled: true
  template:
    pod:
      containers:
        - env:
            - name: MQ_DEV
              value: "true"
            - name: MQ_CONNAUTH_USE_HTP
              value: "true"
            - name: MQ_ADMIN_PASSWORD
```

```
              valueFrom:
                secretKeyRef:
                  name: my-mq-dev-passwords
                  key: dev-admin-password
            - name: MQ_APP_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: my-mq-dev-passwords
                  key: dev-app-password
          name: qmgr
  queueManager:
    storage:
      queueManager:
        type: persistent-claim
    name: QUICKSTART
  version: 9.4.0.0-r1
```

Traitement des incidents liés à IBM MQ dans les conteneurs

Si vous rencontrez des problèmes lors de l'exécution de IBM MQ dans un conteneur, vous pouvez utiliser les techniques décrites ici pour vous aider à diagnostiquer et à résoudre les problèmes.

Procédure

- «Traitement des incidents liés aux redémarrages non planifiés de IBM MQ dans des conteneurs», à la page 177.
- «Traitement des incidents liés à IBM MQ Operator», à la page 178.

OpenShift CP4I Kubernetes Traitement des incidents liés aux redémarrages non planifiés de IBM MQ dans des conteneurs

Dans la plupart des systèmes de gestion de conteneur tels que Red Hat OpenShift Container Platform et Kubernetes, les conteneurs sont généralement redémarrés. Il n'est pas normal qu'un conteneur ait une longue durée de vie. Cette rubrique explique le cycle de vie du conteneur, la manière dont vous pouvez examiner un redémarrage et les raisons d'un redémarrage de conteneur non planifié.

Si vous n'avez rencontré aucun problème avec votre déploiement IBM MQ et qu'il continue de s'exécuter comme prévu, il est probable que la solution fonctionne comme prévu. Vous pouvez voir un message de journal comme celui-ci dans le journal de conteneur:

```
Signal received: terminated
```

Cela signifie que le signal SIGTERM a été envoyé au conteneur MQ , lui demandant de s'arrêter.
Les conteneurs Linux ont la responsabilité de répondre aux signaux POSIX , qui sont des messages standardisés envoyés à un programme pour déclencher le comportement.

Lorsque le conteneur IBM MQ reçoit un signal SIGTERM, il émet une commande `endmqm -w -r -tp` pour arrêter le gestionnaire de files d'attente. Une fois le gestionnaire de files d'attente arrêté, le conteneur s'arrête. Si l'arrêt du gestionnaire de files d'attente prend beaucoup de temps, un signal SIGKILL peut être envoyé, ce qui interrompt immédiatement les processus Linux . Le délai entre un SIGTERM et un SIGKILL est appelé "délai de grâce de résiliation" dans Kuberneteset peut être configuré sur la ressource `QueueManager` (si vous utilisez le IBM MQ Operator) ou directement sur la ressource Pod. La valeur par défaut est de 30 secondes, dont une seconde est réservée pour l'arrêt du conteneur et le reste est attribué à IBM MQ. Par exemple, dans le cas par défaut, un `endmqm -w -tp 29` est émis, qui indique au gestionnaire de files d'attente qu'il a 29 secondes pour s'arrêter.

Raisons de l'expulsion de pod

Le signal SIGTERM est utilisé par Kubernetes (et donc par Red Hat OpenShift Container Platform) pour terminer correctement un pod. Voir Arrêt de pods dans la documentation Kubernetes . Kubernetes utilise les termes "Pod Disruption et" eviction " pour le processus par lequel les pods sur les noeuds sont arrêtés

volontairement ou involontairement. Il existe de nombreuses raisons pour lesquelles un pod peut être expulsé, notamment:

- **Résiliation par kubelet**. Cela peut être dû à un certain nombre de raisons, notamment:
  - Le pod peut être arrêté car le noeud est en cours d'arrêt (peut-être dans le cadre d'une mise à jour de cluster en continu)
  - Le pod peut être arrêté en raison de la "pression" du noeud (où le kubelet arrête proactivement les pods pour récupérer des ressources sur un noeud). L'administrateur de cluster Kubernetes peut configurer des seuils d'expulsion qui peuvent varier d'un cluster à l'autre.
  - Le pod peut être arrêté car il a échoué à sa sonde de vivacité. Une sonde de vivacité peut être configurée dans Kubernetes pour vérifier qu'un pod est toujours sain. Le IBM MQ Operator configure une sonde de vivacité de gestionnaire de files d'attente qui appelle la commande **dspmq** pour vérifier un état d'exécution valide. Si le gestionnaire de files d'attente n'est pas dans un état sain ou si l'exécution de la sonde elle-même prend trop de temps, le kubelet considère qu'il s'agit d'un échec. Les seuils du nombre d'échecs à tolérer sont configurables, soit sur la ressource `QueueManager` (si vous utilisez le IBM MQ Operator), soit directement sur la ressource de pod.
- **Préemption par le planificateur Kubernetes**. Cela peut se produire si le planificateur Kubernetes doit exécuter un pod de priorité plus élevée
- **Noeud entaché**. Un noeud peut être "taint" et les pods qui ne tolèrent pas la tache sont expulsés. Les teintes sont utilisées par les administrateurs Kubernetes pour "repousser" les pods de noeuds spécifiques. Par exemple, pour dire que les pods IBM MQ ne doivent plus s'exécuter sur des noeuds dotés d'un matériel spécial qui est désormais réservé à d'autres charges de travail.
- **Demande via l'API Eviction**. Cela peut être appelé par un administrateur pour expulser des pods
- **Récupération de place de pod**. Cela peut se produire si le noeud est mis hors service ou supprimé via l'API Kubernetes .

Détermination de la raison pour laquelle un pod de gestionnaire de files d'attente a été expulsé

Les sources d'information potentielles pour aider à comprendre pourquoi un pod a été expulsé sont les suivantes:

- **Evénements de cluster**. Par exemple, Affichage des informations d'événement système dans un cluster OpenShift Container Platform .
- **Evénements d'audit de cluster**. Voir Affichage des journaux d'audit dans Red Hat OpenShift Container Platform.
- **Noeuds sous pression**. Recherchez les noeuds sous la pression de l'unité centrale, du réseau ou de la mémoire. Vous pouvez voir cela dans le statut du noeud. Notez qu'au moment de la recherche, il se peut que le noeud ne soit plus sous pression.
- **Red Hat OpenShift Container Platform Monitoring** ou d'autres métriques de surveillance peuvent être en mesure d'afficher des éléments tels que des problèmes de temps d'attente de disque. Une métrique Prometheus utile est `ibmmq_qmgr_log_write_latency_seconds`. Ces informations proviennent des rubriques relatives aux statistiques de MQ .

Information associée

Documentation Kubernetes sur la planification, la préemption et l'expulsion

OpenShift CP4I Traitement des incidents liés à IBM MQ Operator

Si vous rencontrez des problèmes liés à IBM MQ Operator, appliquez les techniques de ce document pour les diagnostiquer et les résoudre.

Procédure

- «Collecte des informations d'identification et de résolution des problèmes pour les gestionnaires de files d'attente déployés avec IBM MQ Operator», à la page 179
- «Identification et résolution des problèmes: accès aux données du gestionnaire de files d'attente», à la page 181

OpenShift CP4I Collecte des informations d'identification et de résolution des problèmes pour les gestionnaires de files d'attente déployés avec IBM MQ Operator

Collecte des informations de traitement des incidents qui doivent être fournies au support IBM lors de la création d'un nouveau cas de support.

Procédure

1. Collectez des informations sur le fournisseur de cloud.

   Il s'agit du fournisseur de cloud qui héberge votre cluster Red Hat OpenShift (par exemple, IBM Cloud).
2. Collectez des informations sur l'architecture.

   L'architecture de votre cluster Red Hat OpenShift est l'une des suivantes:

   - Linux for x86-64
   - Linux on Power Systems (ppc64le)
   - Linux for IBM Z
3. Collectez les informations de déploiement IBM MQ .
   a) Connectez-vous à votre cluster Red Hat OpenShift à l'aide d'un interpréteur de commandes `bash/zsh` .
   b) Définissez les variables d'environnement suivantes :

   ```
   export QM=QueueManager_name
   export QM_NAMESPACE=QueueManager_namespace
   export MQ_OPERATOR_NAMESPACE=mq_operator_namespace
   ```

   Où *QueueManager_name* est le nom de votre ressource QueueManager , *QueueManager_namespace* est l'espace de nom dans lequel il est déployé et *mq_operator_namespace* est l'espace de nom dans lequel IBM MQ Operator est déployé. Il peut s'agir de l'espace de nom QueueManager .
   c) Exécutez les commandes suivantes et fournissez tous les fichiers de sortie résultants au support IBM .

   ```
   # OCP / Kubernetes: Version
   oc version -o yaml > ocversion.yaml

   # QueueManager: YAML
   oc get qmgr $QM -n $QM_NAMESPACE -o yaml > "queue-manager-$QM.yaml"

   # MQ Queue Manager: Pods
   oc get pods -n $QM_NAMESPACE -o wide --selector "app.kubernetes.io/instance=$QM" > "qm-
   pods-$QM.txt"

   # MQ Queue Manager: Pod YAML
   oc get pods -n $QM_NAMESPACE -o yaml --selector "app.kubernetes.io/instance=$QM" > "qm-
   pods-$QM.yaml"

   # MQ Queue Manager: Pod Logs
   for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/
   instance=$QM" | cut -d ' ' -f 1); do oc logs -n $QM_NAMESPACE --previous "$p" > "qm-logs-
   previous-$p.txt"; oc logs -n $QM_NAMESPACE $p > "qm-logs-$p.txt";done

   # MQ Queue Manager: Describe Pods
   for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/
   instance=$QM" | cut -d ' ' -f 1); do oc describe pod $p -n $QM_NAMESPACE > "qm-pod-
   ```

```
describe-$p.txt"; done

# MQ Web UI: Console Log
for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/
instance=$QM" | cut -d ' ' -f 1); do oc cp -n $QM_NAMESPACE --retries=10 "$p:var/mqm/web/
installations/Installation1/servers/mqweb/logs/console.log" "web-$p-console.log"; done

# MQ Web UI: Messages Log
for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/
instance=$QM" | cut -d ' ' -f 1); do oc cp -n $QM_NAMESPACE --retries=10 "$p:var/mqm/web/
installations/Installation1/servers/mqweb/logs/messages.log" "web-$p-messages.log"; done

# MQ Queue Manager: routes defined by operator
oc get routes -n $QM_NAMESPACE -o yaml --selector "app.kubernetes.io/instance=$QM" > "qm-
routes-$QM.yaml"

# MQ Queue Manager: routes to QM
oc get routes -n $QM_NAMESPACE -o yaml --field-selector "spec.to.name=$QM-ibm-mq" > "qm-
routes2-$QM.yaml"

# MQ Queue Manager: stateful set
oc get statefulset -n $QM_NAMESPACE -o yaml ${QM}-ibm-mq > "qm-statefulset-$QM.yaml"

# MQ Queue Manager: revisions of the stateful set
oc get controllerrevisions.apps -o yaml -n $QM_NAMESPACE --selector "app.kubernetes.io/
instance=$QM" > "qm-statefulset-revisions-$QM.yaml"

# MQ Queue Manager: Pod events
for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/
instance=$QM" | cut -d ' ' -f 1); do oc get -o custom-columns="LAST
SEEN:.lastTimestamp,TYPE:.type,REASON:.reason,KIND:.involvedObject.kind,NAME:.involvedObje
ct.name,MESSAGE:.message" event -n $QM_NAMESPACE --field-selector
involvedObject.name="$p" > "qm-pod-events-$p.txt"; done

# MQ Queue Manager: StatefulSet events
oc get events -n $QM_NAMESPACE -o custom-columns="LAST
SEEN:.lastTimestamp,TYPE:.type,REASON:.reason,KIND:.involvedObject.kind,NAME:.involvedObje
ct.name,MESSAGE:.message" --field-selector involvedObject.name="${QM}-ibm-mq" > "qm-
statefulset-events-$QM.txt"

# MQ Queue Manager: services
oc get services -n $QM_NAMESPACE -o yaml --selector "app.kubernetes.io/instance=$QM" >
"qm-services-$QM.yaml"

# MQ Queue Manager: PVCs
oc get pvc -n $QM_NAMESPACE -o yaml --selector "app.kubernetes.io/instance=$QM" > "qm-
pvcs-$QM.yaml"

# MQ Operator: Version
oc get csv -n $QM_NAMESPACE | grep "^ibm-mq\|NAME" > mq-operator-csv.txt

# Cloud Pak Foundational Services: Version
oc get csv -n $QM_NAMESPACE | grep "^ibm-common-service-operator\|NAME" > common-services-
csv.txt

# Cloud Pak for Integration: Version (if applicable)
oc get csv -n $QM_NAMESPACE | grep "^ibm-integration-platform-navigator\|NAME" > cp4i-
csv.txt

# Output from runmqras (this may take a while to execute)
for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/
instance=$QM" | cut -d ' ' -f 1); do timestamp=$(TZ=UTC date +"%Y%m%d_%H%M%S"); oc exec
-n $QM_NAMESPACE $p -- runmqras -workdirectory "/tmp/runmqras_$timestamp" -section
logger,mqweb,nativeha,trace; oc cp -n $QM_NAMESPACE --retries=10 "$p:tmp/
runmqras_$timestamp/" .; done

# MQ Operator: Pod Log
oc logs -n $MQ_OPERATOR_NAMESPACE $(oc get pods -n $MQ_OPERATOR_NAMESPACE --no-headers --
selector app.kubernetes.io/name=ibm-mq,app.kubernetes.io/managed-by=olm | cut -d ' ' -f
1) > mq-operator-log.txt
```

Remarque :

La majorité de ces commandes nécessitent un accès à l'espace de nom dans lequel le gestionnaire de files d'attente est déployé. Toutefois, la collecte du journal IBM MQ Operator peut également nécessiter un accès **administrateur de cluster** si IBM MQ Operator est installé **au niveau du cluster**.

Tâches associées

Collecte des informations de traitement des incidents pour le support IBM

OpenShift CP4I Identification et résolution des problèmes: accès aux données du gestionnaire de files d'attente

Utilisez l'outil d'inspection de réservation de volume persistant pour accéder aux fichiers d'une réservation de volume persistant de gestionnaire de files d'attente dans laquelle un shell distant ne peut pas être établi sur le pod du gestionnaire de files d'attente. Cela peut être dû au fait que le pod est à l'état **`Error`** ou **`CrashLoopBackOff`** . Cet outil est conçu pour être utilisé avec les gestionnaires de files d'attente déployés par IBM MQ Operator.

Avant de commencer

Pour utiliser l'outil d'inspecteur PVC. vous devez avoir accès à l'espace de nom de votre gestionnaire de files d'attente.

Pourquoi et quand exécuter cette tâche

Pour faciliter le traitement des incidents, vous pouvez accéder aux données stockées sur les réservations de volume persistant (PVC) associées à un gestionnaire de files d'attente donné. Pour ce faire, vous utilisez un outil pour monter les réservations de volume persistant sur un ensemble de pods inspector. Vous pouvez ensuite obtenir un shell distant dans n'importe lequel des pods inspector pour lire les fichiers.

Selon le type de déploiement, entre un et trois pods inspector sont créés. Les volumes spécifiques à un pod donné d'un gestionnaire de files d'attente Native-HA ou Multi-Instance sont disponibles sur le pod de l'inspecteur PVC associé. Les volumes partagés sont disponibles sur tous les inspecteurs. Le nom du pod inspector contient le nom du pod de gestionnaire de files d'attente associé.

Procédure

1. Téléchargez l'outil MQ PVC inspector.

   L'outil est disponible ici: https://github.com/ibm-messaging/mq-pvc-tool.
2. Vérifiez que vous êtes connecté à votre cluster.
3. Recherchez le nom du gestionnaire de files d'attente et l'espace de nom dans lequel le gestionnaire de files d'attente s'exécute.
4. Exécutez l'outil inspector sur votre gestionnaire de files d'attente.
   a) Exécutez la commande suivante en spécifiant le nom de votre gestionnaire de files d'attente et son nom d'espace de nom.

   ```
   ./pvc-tool.sh queue_manager_name queue_manager_namespace_name
   ```

   b) Une fois l'outil terminé, exécutez la commande suivante pour afficher les pods d'inspecteur en cours de création.

   ```
   oc get pods
   ```
5. Affichez les fichiers montés sur le pod inspector.
   a) Chaque pod d'inspecteur de PVC est associé à un pod de gestionnaire de files d'attente, de sorte qu'il peut y avoir plusieurs pods d'inspecteur. Accédez à l'un de ces pods en exécutant la commande suivante:

   ```
   oc rsh pvc-inspector-pod-name
   ```

   Vous êtes placé dans le répertoire contenant les répertoires PVC montés.

   b) Répertoriez les répertoires PVC en exécutant la commande suivante:

```
ls
```

c) Affichez la liste des réservations de volume persistant en exécutant la commande suivante en dehors de la session shell distante:

```
oc get pvc
```

d) Nettoyez les pods créés par l'outil en exécutant la commande suivante:

```
oc delete pods -l tool=mq-pvc-inspector
```

Remarques

:NONE.

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non annoncés dans ce pays. Consultez votre représentant IBM local pour obtenir des informations sur les produits et services actuellement disponibles dans votre région. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service IBM puisse être utilisé. Tout produit, programme ou service fonctionnellement équivalent qui ne porte pas atteinte à un droit de propriété intellectuelle IBM peut être utilisé à la place. 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 Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Pour obtenir des informations sur les licences relatives aux informations sur deux octets (DBCS), contactez le service de la propriété intellectuelle IBM de votre pays ou envoyez vos demandes de renseignements, par écrit, à :

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

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/ou programmes décrits dans ce document.

Les références à des sites Web non IBM sont fournies uniquement à titre d'information et n'impliquent en aucune façon une adhésion de ces sites Web. 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 d' IBM 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 certains 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 (https://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: