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 309.

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.

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

© Copyright International Business Machines Corporation 2007, 2025.

Table des matières

IBM MQ en conteneurs et IBM Cloud Pak for Integration..... 5
A propos de..... 5
Historique des éditions de IBM MQ Operator..... 5
Historique des versions des images des conteneurs du gestionnaire de file d'attente..... 18
Planification..... 27
Comment utiliser IBM MQ dans des conteneurs..... 28
Prise en charge de IBM MQ dans les conteneurs..... 29
Planification de l'octroi de licence à IBM MQ dans des conteneurs..... 38
Planification du stockage pour le IBM MQ Operator..... 39
Planification de la haute disponibilité pour IBM MQ dans des conteneurs..... 41
Reprise après incident d'IBM MQ dans des conteneurs..... 47
Planification de la sécurité pour IBM MQ dans des conteneurs..... 50
Planification de l'évolutivité et des performances pour IBM MQ dans les conteneurs..... 56
Préparation, installation et mise à niveau..... 57
Installation et mise à niveau du IBM MQ Operator..... 57
Préparation de l'utilisation de 'Kubernetes par la création d'un secret d'extraction..... 88
Préparer l'utilisation de Kubernetes en installant le IBM License Service..... 89
Préparation à l'utilisation de 'Podman ou 'Docker en tirant l'image du conteneur préconstruit..... 89
Préparation pour IBM MQ en créant votre propre image de conteneur..... 90
Déploiement et configuration..... 99
Déploiement et configuration de gestionnaires de files d'attente à l'aide de IBM MQ Operator..... 100
Déploiement et configuration des gestionnaires de files d'attente sur Kubernetes..... 178
Déploiement d'un gestionnaire de file d'attente sur 'Podman..... 194
Migration vers IBM MQ Operator..... 199
Vérification de la disponibilité des fonctions requises..... 200
Extraction de la configuration du gestionnaire de files d'attente..... 201
Facultatif : extraction et acquisition des clés et certificats du gestionnaire de files d'attente..... 201
Facultatif : configuration de LDAP..... 203
Facultatif : modification des adresses IP et noms d'hôte dans la configuration IBM MQ..... 211
Mise à jour de la configuration du gestionnaire de files d'attente pour un environnement de conteneur..... 212
Sélection de l'architecture haute disponibilité cible pour IBM MQ exécuté en conteneurs..... 216
Création des ressources du gestionnaire de files d'attente..... 216
Création du gestionnaire de files d'attente dans Red Hat OpenShift..... 218
Vérification du nouveau déploiement de conteneur..... 222
Fonctionnement..... 223
Utilisation de IBM MQ à l'aide de IBM MQ Operator..... 223
Affichage du statut des gestionnaires de files d'attente Native HA..... 237
Basculement et basculement automatique en cas de panne de la HA CRR native..... 242
Fin des gestionnaires de files d'attente Native HA..... 247
Traitement des incidents..... 248
Traitement des incidents liés aux redémarrages non planifiés de IBM MQ dans des conteneurs..... 248
Traitement des incidents liés à IBM MQ Operator..... 250
Collecte d'informations de dépannage pour les gestionnaires de files d'attente déployés sur Kubernetes..... 255
Référence..... 256
Références de licences et d'API pour le IBM MQ Operator..... 256
IBM MQ annotations sur la licence du conteneur..... 289
IBM MQ les annotations de licence du conteneur qui sont appliquées par l'outil IBM MQ Operator..... 295
IBM MQ Advanced image de conteneur..... 296
IBM MQ image de conteneur sous licence..... 299

IBM MQ Advanced for Developers image de conteneur..... 300
IBM MQ MFT Agent for Developers image de conteneur.....302
IBM MQ CASE pour l'installation et la mise à jour de l'air-gap sur 'Red Hat OpenShift.....303

Remarques.....309
Documentation sur l'interface de programmation..... 310
Marques..... 310

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.

MQ Adv. CD 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 ?

OpenShift CP4I 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 3.6.0

CD

Version de IBM Cloud Pak for Integration

IBM Cloud Pak for Integration 16.1.2

Canal opérateur

`v3.6`

Valeurs autorisées pour `.spec.version`

9.4.3.0-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, 9.3.0.1-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, 9.3.0.20-r1, 9.3.0.20-r2, 9.3.0.21-r1, 9.3.0.21-r2, 9.3.0.21-r3, 9.3.0.25-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.3-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,

© Copyright IBM Corp. 2007, 2025

9.3.5.1-r2, 9.4.0.0-r1, 9.4.0.0-r2, 9.4.0.0-r3, 9.4.0.5-r1, 9.4.0.5-r2, 9.4.0.6-r1, 9.4.0.6-r2, 9.4.0.7-r1, 9.4.0.10-r1, 9.4.0.10-r2, 9.4.0.11-r1, 9.4.0.11-r2, 9.4.0.11-r3, 9.4.1.0-r1, 9.4.1.0-r2, 9.4.1.1-r1, 9.4.2.0-r1, 9.4.2.0-r2, 9.4.2.1-r1, 9.4.2.1-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

- Amélioration de la journalisation de la validation de l'accès aux webhooks, afin de faciliter la conformité avec les exigences d'audit en matière de journalisation des accès. Pour plus d'informations, voir «Configuration de la journalisation d'audit pour IBM MQ dans les conteneurs», à la page 155.
- Le site IBM MQ Operator a été validé pour les environnements à double pile IPv6, conformément aux dernières exigences fédérales américaines en matière de conformité.
- Vous pouvez maintenant configurer les paramètres `ipFamilyPolicy` et `ipFamilies` de IBM MQ queue manager Services. La configuration est disponible sous `spec.service`. Les paramètres s'appliquent à tous les services de gestion des files d'attente.
- Vous pouvez désormais configurer les gestionnaires de file d'attente déployés avec IBM MQ Operator avec des annotations de licence IBM MQ . Pour plus d'informations, voir «Configurer les gestionnaires de files d'attente avec les annotations de licence IBM MQ en utilisant la fonction IBM MQ Operator», à la page 166. Cette mise à jour s'applique à tous les gestionnaires de files d'attente IBM MQ 9.4
- Vous pouvez désormais utiliser les fonctionnalités Native HA et Cross-Region Replication sur les gestionnaires de file d'attente sous licence IBM MQ en ajoutant la nouvelle licence IBM MQ Native HA et Cross-Region Replication Add-on. Pour plus d'informations, voir «Configurer Native HA et Cross-Region Replication sur IBM MQ Queue Managers sous licence en utilisant la fonction IBM MQ Operator», à la page 119. Notez que cette mise à jour s'applique également aux gestionnaires de file d'attente IBM MQ 9.4.2.

IBM MQ Operator 3.5.3

CD

Version de IBM Cloud Pak for Integration
IBM Cloud Pak for Integration 16.1.1

Canal opérateur
`v3.5`

Valeurs autorisées pour `.spec.version`
9.4.2.1-r2

Valeurs autorisées pour `.spec.version` lors de la migration
9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-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, 9.3.0.20-r1, 9.3.0.20-r2, 9.3.0.21-r1, 9.3.0.21-r2, 9.3.0.21-r3, 9.3.0.25-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.3-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, 9.4.0.0-r1, 9.4.0.0-r2, 9.4.0.0-r3, 9.4.0.5-r1, 9.4.0.5-r2, 9.4.0.6-r1, 9.4.0.6-r2, 9.4.0.7-r1, 9.4.0.10-r1, 9.4.0.10-r2, 9.4.0.11-r1, 9.4.1.0-r1, 9.4.1.0-r2, 9.4.1.1-r1, 9.4.2.0-r2, 9.4.2.1-r1

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

- Les vulnérabilités corrigées sont détaillées dans ce bulletin de sécurité : https://www.ibm.com/support/pages/node/7234178

IBM MQ Operator 3.5.2

CD

Version de IBM Cloud Pak for Integration
IBM Cloud Pak for Integration 16.1.1

Canal opérateur
`v3.5`

Valeurs autorisées pour `.spec.version`
9.4.2.1-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, 9.3.0.1-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, 9.3.0.20-r1, 9.3.0.20-r2, 9.3.0.21-r1, 9.3.0.21-r2, 9.3.0.21-r3, 9.3.0.25-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.3-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, 9.4.0.0-r1, 9.4.0.0-r2, 9.4.0.0-r3, 9.4.0.5-r1, 9.4.0.5-r2, 9.4.0.6-r1, 9.4.0.6-r2, 9.4.0.7-r1, 9.4.0.10-r1, 9.4.0.10-r2, 9.4.0.11-r1, 9.4.1.0-r1, 9.4.1.0-r2, 9.4.1.1-r1, 9.4.2.0-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

- L'identifiant et le secret du client IBM MQ OpenID Connect (OIDC) sont désormais stockés dans un volume et non plus dans une variable d'environnement.
- Les vulnérabilités corrigées sont détaillées dans ce bulletin de sécurité : https://www.ibm.com/support/pages/node/7232272

IBM MQ Operator 3.5.1

CD

Version de IBM Cloud Pak for Integration
IBM Cloud Pak for Integration 16.1.1

Canal opérateur
`v3.5`

Valeurs autorisées pour `.spec.version`
9.4.2.0-r2

Valeurs autorisées pour `.spec.version` lors de la migration
9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-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, 9.3.0.20-r1, 9.3.0.20-r2, 9.3.0.21-r1, 9.3.0.21-r2, 9.3.0.21-r3, 9.3.0.25-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.3-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, 9.4.0.0-r1, 9.4.0.0-r2, 9.4.0.0-r3, 9.4.0.5-r1, 9.4.0.5-r2, 9.4.0.6-r1, 9.4.0.6-r2, 9.4.0.7-r1, 9.4.0.10-r1, 9.4.0.10-r2, 9.4.1.0-r1, 9.4.1.0-r2, 9.4.1.1-r1, 9.4.2.0-r1

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

- Correction d'un problème où, dans certaines conditions, la ressource IntegrationKeycloakClient était mise à jour avec un identifiant de client incorrect.
- Les vulnérabilités corrigées sont détaillées dans ce bulletin de sécurité : https://www.ibm.com/support/pages/node/7228945

IBM MQ Operator 3.5.0

CD

Version de IBM Cloud Pak for Integration
IBM Cloud Pak for Integration 16.1.1

Canal opérateur
`v3.5`

Valeurs autorisées pour `.spec.version`
9.4.2.0-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, 9.3.0.1-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, 9.3.0.20-r1, 9.3.0.20-r2, 9.3.0.21-r1, 9.3.0.21-r2, 9.3.0.21-r3, 9.3.0.25-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.3-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, 9.4.0.0-r1, 9.4.0.0-r2, 9.4.0.0-r3, 9.4.0.5-r1, 9.4.0.5-r2, 9.4.0.6-r1, 9.4.0.6-r2, 9.4.0.7-r1, 9.4.0.10-r1, 9.4.1.0-r1, 9.4.1.0-r2, 9.4.1.1-r1

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

- Des fichiers supplémentaires peuvent désormais être fournis lorsqu'un gestionnaire de file d'attente est déployé. Ces fichiers sont montés dans le conteneur d' IBM MQ s aux emplacements spécifiés. Pour plus d'informations, voir «Exemple : Fournir des fichiers supplémentaires à un gestionnaire de file d'attente», à la page 105.
- Des variables d'environnement supplémentaires peuvent désormais être fournies lors du déploiement d'un gestionnaire de file d'attente. Pour plus d'informations, voir «Exemple : Fournir des variables d'environnement supplémentaires pour un gestionnaire de file d'attente», à la page 106.
- IBM MQ les gestionnaires de file d'attente d' 9.4.2.0-r1 peuvent être configurés par l' IBM MQ Operator pour servir des métriques via HTTPS en utilisant des certificats créés par OpenShift. Pour plus d'informations, voir «Surveillance lors de l'utilisation de IBM MQ Operator», à la page 224.
- IBM MQ CASE 3.5.0 et contiennent ensuite uniquement la dernière version du gestionnaire de file d'attente. Par exemple IBM MQ CASE 3.5.0 ne contient que la version du gestionnaire de file d'attente 9.4.2.0-r1. Lors de l'installation d'un espace d'air, cela permet d'accélérer l'étape de mise en miroir de l'image, mais vous devrez peut-être effectuer des étapes supplémentaires de mise en miroir de l'image pour installer ou mettre à niveau un gestionnaire de file d'attente qui n'est pas le plus récent. Pour plus d'informations, voir «IBM MQ CASE pour l'installation et la mise à jour de l'air-gap sur 'Red Hat OpenShift», à la page 303.

Modifications
Les vulnérabilités corrigées sont détaillées dans ce bulletin de sécurité : https://www.ibm.com/support/pages/node/7184453

IBM MQ Operator 3.4.1

CD

Version de IBM Cloud Pak for Integration
IBM Cloud Pak for Integration 16.1.1

Canal opérateur
`v3.4`

Valeurs autorisées pour `.spec.version`
9.4.1.1-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, 9.3.0.1-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, 9.3.0.20-r1, 9.3.0.20-r2, 9.3.0.21-r1, 9.3.0.21-r2, 9.3.0.21-r3, 9.3.0.25-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.3-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, 9.4.0.0-r1, 9.4.0.0-r2, 9.4.0.0-r3, 9.4.0.5-r1, 9.4.0.5-r2, 9.4.0.6-r1, 9.4.0.6-r2, 9.4.0.7-r1, 9.4.1.0-r1, 9.4.1.0-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
Les vulnérabilités corrigées sont détaillées dans ce bulletin de sécurité : https://www.ibm.com/support/pages/node/7182196

IBM MQ Operator 3.4.0

CD

Version de IBM Cloud Pak for Integration
IBM Cloud Pak for Integration 16.1.1

Canal opérateur
`v3.4`

Valeurs autorisées pour `.spec.version`
9.4.1.0-r2

Valeurs autorisées pour `.spec.version` lors de la migration
9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-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, 9.3.0.20-r1, 9.3.0.20-r2, 9.3.0.21-r1, 9.3.0.21-r2, 9.3.0.21-r3, 9.3.0.25-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.3-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, 9.4.0.0-r1, 9.4.0.0-r2, 9.4.0.0-r3, 9.4.0.5-r1, 9.4.0.5-r2, 9.4.0.6-r1, 9.4.0.6-r2, 9.4.0.7-r1, 9.4.1.0-r1

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

- Un certain nombre d'anciens opérateurs qui n'étaient plus soutenus ont été retirés de la CASE.
- Les vulnérabilités corrigées sont détaillées dans ce bulletin de sécurité : https://www.ibm.com/support/pages/node/7178065

IBM MQ Operator 3.3.0

CD

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

Canal opérateur
`v3.3`

Valeurs autorisées pour `.spec.version`
9.4.1.0-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, 9.3.0.1-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, 9.3.0.20-r1, 9.3.0.20-r2, 9.3.0.21-r1, 9.3.0.21-r2, 9.3.0.21-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, 9.3.3.3-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, 9.4.0.0-r1, 9.4.0.0-r2, 9.4.0.0-r3, 9.4.0.5-r1, 9.4.0.5-r2, 9.4.0.6-r1

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

- La licence " IBM MQ Advanced ne peut plus être configurée avec l'utilisation `Développement`.
- Les gestionnaires de files d'attente sont désormais dotés des annotations de licence correctes. Pour plus d'informations ou pour des instructions sur la mise à jour des gestionnaires de file d'attente précédents, voir " «Correction des annotations de licence pour les gestionnaires de file d'attente déployés», à la page 177.
- Les règles concernant la valeur du '**`spec.license.metric`** sont maintenant appliquées par le webhook. Les gestionnaires de file d'attente existants doivent suivre ces règles lors de la mise à niveau vers la version 9.4.1.0-r1 et les versions ultérieures.
- Ajout de la prise en charge de la licence " IBM MQ Advanced for Non-Production Environment
- Suppression de la possibilité de déployer avec une licence 'IBM MQ Advanced et un '**`Use`** réglé sur `Développement`.
- Réduire l'empreinte mémoire du " IBM MQ Operator dans les grandes grappes.
- Correction d'un problème avec les rapprochements '`ServiceAccount` lors de l'utilisation de l'OCP 4.16
- Les vulnérabilités corrigées sont détaillées dans ce bulletin de sécurité : https://www.ibm.com/support/pages/node/7174358

IBM MQ Operator 3.2.13

CP4I-SC2

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`
9.4.0.11-r3

Valeurs autorisées pour `.spec.version` lors de la migration

9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-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, 9.3.0.20-r1, 9.3.0.20-r2, 9.3.0.21-r1, 9.3.0.21-r2, 9.3.0.21-r3, 9.3.0.25-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.3-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, 9.4.0.0-r1, 9.4.0.0-r2, 9.4.0.0-r3, 9.4.0.5-r1, 9.4.0.5-r2, 9.4.0.6-r1, 9.4.0.6-r2, 9.4.0.7-r1, 9.4.0.10-r1, 9.4.0.10-r2, 9.4.0.11-r1, 9.4.0.11-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.

IBM MQ Operator 3.2.12

CP4I-SC2

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`
9.4.0.11-r2

Valeurs autorisées pour `.spec.version` lors de la migration
9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-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, 9.3.0.20-r1, 9.3.0.20-r2, 9.3.0.21-r1, 9.3.0.21-r2, 9.3.0.21-r3, 9.3.0.25-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.3-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, 9.4.0.0-r1, 9.4.0.0-r2, 9.4.0.0-r3, 9.4.0.5-r1, 9.4.0.5-r2, 9.4.0.6-r1, 9.4.0.6-r2, 9.4.0.7-r1, 9.4.0.10-r1, 9.4.0.10-r2, 9.4.0.11-r1

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

- Les vulnérabilités corrigées sont détaillées dans ce bulletin de sécurité : https://www.ibm.com/support/pages/node/7234178

IBM MQ Operator 3.2.11

CP4I-SC2

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`
9.4.0.11-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, 9.3.0.1-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, 9.3.0.20-r1, 9.3.0.20-r2, 9.3.0.21-r1, 9.3.0.21-r2, 9.3.0.21-r3, 9.3.0.25-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.3-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, 9.4.0.0-r1, 9.4.0.0-r2, 9.4.0.0-r3, 9.4.0.5-r1, 9.4.0.5-r2, 9.4.0.6-r1, 9.4.0.6-r2, 9.4.0.7-r1, 9.4.0.10-r1, 9.4.0.10-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

- L'identifiant et le secret du client IBM MQ OpenID Connect (OIDC) sont désormais stockés dans un volume et non plus dans une variable d'environnement.
- Suppression du secret d'extraction de l'image `sa-<namespace>` inutilisé afin d'éviter les avertissements concernant Kubernetes.
- Les vulnérabilités corrigées sont détaillées dans ce bulletin de sécurité : https://www.ibm.com/support/pages/node/7232272

IBM MQ Operator 3.2.10

CP4I-SC2

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`
9.4.0.10-r2

Valeurs autorisées pour `.spec.version` lors de la migration
9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-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, 9.3.0.20-r1, 9.3.0.20-r2, 9.3.0.21-r1, 9.3.0.21-r2, 9.3.0.21-r3, 9.3.0.25-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.3-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, 9.4.0.0-r1, 9.4.0.0-r2, 9.4.0.0-r3, 9.4.0.5-r1, 9.4.0.5-r2, 9.4.0.6-r1, 9.4.0.6-r2, 9.4.0.7-r1, 9.4.0.10-r1

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

- Correction d'un problème où, dans certaines conditions, la ressource IntegrationKeycloakClient était mise à jour avec un identifiant de client incorrect.
- Les vulnérabilités corrigées sont détaillées dans ce bulletin de sécurité : https://www.ibm.com/support/pages/node/7228945

IBM MQ Operator 3.2.9

CP4I-SC2

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`
9.4.0.10-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, 9.3.0.1-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, 9.3.0.20-r1, 9.3.0.20-r2, 9.3.0.21-r1, 9.3.0.21-r2, 9.3.0.21-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, 9.3.3.3-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, 9.4.0.0-r1, 9.4.0.0-r2, 9.4.0.0-r3, 9.4.0.5-r1, 9.4.0.5-r2, 9.4.0.6-r1, 9.4.0.6-r2, 9.4.0.7-r1

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
Les vulnérabilités corrigées sont détaillées dans ce bulletin de sécurité : https://www.ibm.com/support/pages/node/7184453

IBM MQ Operator 3.2.8

CP4I-SC2

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`
9.4.0.7-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, 9.3.0.1-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, 9.3.0.20-r1, 9.3.0.20-r2, 9.3.0.21-r1, 9.3.0.21-r2, 9.3.0.21-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, 9.3.3.3-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, 9.4.0.0-r1, 9.4.0.0-r2, 9.4.0.0-r3, 9.4.0.5-r1, 9.4.0.5-r2, 9.4.0.6-r1, 9.4.0.6-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
Les vulnérabilités corrigées sont détaillées dans ce bulletin de sécurité : https://www.ibm.com/support/pages/node/7182196

IBM MQ Operator 3.2.7

CP4I-SC2

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`
9.4.0.6-r2

Valeurs autorisées pour `.spec.version` lors de la migration
9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-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, 9.3.0.20-r1, 9.3.0.20-r2, 9.3.0.21-r1, 9.3.0.21-r2, 9.3.0.21-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, 9.3.3.3-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, 9.4.0.0-r1, 9.4.0.0-r2, 9.4.0.0-r3, 9.4.0.5-r1, 9.4.0.5-r2, 9.4.0.6-r1

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
Les vulnérabilités corrigées sont détaillées dans ce bulletin de sécurité : https://www.ibm.com/support/pages/node/7178065

IBM MQ Operator 3.2.6

CP4I-SC2

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`
9.4.0.6-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, 9.3.0.1-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, 9.3.0.20-r1, 9.3.0.20-r2, 9.3.0.21-r1, 9.3.0.21-r2, 9.3.0.21-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, 9.3.3.3-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, 9.4.0.0-r1, 9.4.0.0-r2, 9.4.0.0-r3, 9.4.0.5-r1, 9.4.0.5-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
Les vulnérabilités corrigées sont détaillées dans ce bulletin de sécurité : https://www.ibm.com/support/pages/node/7174358

IBM MQ Operator 3.2.5

CD CP4I-SC2

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`
9.4.0.5-r2

Valeurs autorisées pour `.spec.version` lors de la migration
9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-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, 9.3.0.20-r1, 9.3.0.20-r2, 9.3.0.21-r1, 9.3.0.21-r2, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2,

9.3.3.3-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, 9.4.0.0-r1, 9.4.0.0-r2, 9.4.0.0-r3, 9.4.0.5-r1

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
Les vulnérabilités corrigées sont détaillées dans ce bulletin de sécurité : https://www.ibm.com/support/pages/node/7171536

IBM MQ Operator 3.2.4

CD CP4I-SC2

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`
9.4.0.5-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, 9.3.0.1-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, 9.3.0.20-r1, 9.3.0.20-r2, 9.3.0.21-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.3-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, 9.4.0.0-r1, 9.4.0.0-r2, 9.4.0.0-r3

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
Les vulnérabilités corrigées sont détaillées dans ce bulletin de sécurité : https://www.ibm.com/support/pages/node/7167732

IBM MQ Operator 3.2.3

CD CP4I-SC2

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-r3

Valeurs autorisées pour `.spec.version` lors de la migration
9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-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, 9.3.0.20-r1, 9.3.0.20-r2, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.3-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, 9.4.0.0-r1, 9.4.0.0-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
Les vulnérabilités corrigées sont détaillées dans ces bulletins de sécurité :

- https://www.ibm.com/support/pages/node/7162095
- https://www.ibm.com/support/pages/node/7162272

IBM MQ Operator 3.2.2

CD CP4I-SC2

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-r2

Valeurs autorisées pour `.spec.version` lors de la migration
9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3, 9.3.0.1-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, 9.3.0.20-r1, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.3-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, 9.4.0.0-r1

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

- Les vulnérabilités qui sont prises en compte sont détaillées dans ce bulletin de sécurité.

IBM MQ Operator 3.2.1

CD CP4I-SC2

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

CD CP4I-SC2

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 173 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 235

  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 Image de base universelle 9.4-949.1716471857. **Note :** UBI 9 est en attente de la certification FIPS 140-3 . UBI 9 n'est pas supporté par l'architecture Power 8.
- Les vulnérabilités qui sont prises en compte sont détaillées dans ce bulletin de sécurité.

OpenShift CP4I Historique des versions des images des conteneurs du gestionnaire de file d'attente

Remarque : Pour plus d'informations sur les images antérieures du conteneur du gestionnaire de file d'attente, voir l'historique des versions pour 'IBM MQ Operator dans la documentation 'IBM MQ 9.3

Images prises en charge de manière autonome ou avec le " IBM MQ Operator

9.4.3.0-r1

CD

Version de l'opérateur compatible

Pris en charge de manière autonome ou avec IBM MQ Operator 3.6.0 ou plus.

Architectures prises en charge

`amd64`, `s390x`, `ppc64le`

Images

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.4.3.0-r1` (IBM Cloud Pak for Integration)
- `cp.icr.io/cp/ibm-mqadvanced-server:9.4.3.0-r1` (IBM MQ Advanced)

  Remarque : Vous pouvez également utiliser cette image pour déployer des gestionnaires de file d'attente sous licence IBM MQ , à condition de respecter les conditions de licence appropriées et d'annoter ces gestionnaires de file d'attente. Pour plus d'informations, reportez-vous aux sections «Configurer les gestionnaires de files d'attente avec les annotations de licence IBM MQ en utilisant la fonction IBM MQ Operator», à la page 166 et «IBM MQ annotations sur la licence du conteneur», à la page 289.
- `icr.io/ibm-messaging/mq:9.4.3.0-r1` (IBM MQ Advanced for Developers)

Nouveautés

- Vous pouvez désormais activer l'enregistrement d'audit de l'accès aux points de terminaison des mesures Prometheus. Pour plus d'informations, voir «Configuration de la journalisation d'audit pour IBM MQ dans les conteneurs», à la page 155.
- Vous pouvez désormais configurer les gestionnaires de file d'attente déployés avec IBM MQ Operator avec des annotations de licence IBM MQ . Pour plus d'informations, voir «Configurer les gestionnaires de files d'attente avec les annotations de licence IBM MQ en utilisant la fonction IBM MQ Operator», à la page 166. Cette mise à jour s'applique à tous les gestionnaires de files d'attente IBM MQ 9.4
- Vous pouvez désormais utiliser les fonctionnalités Native HA et Cross-Region Replication sur les gestionnaires de file d'attente sous licence IBM MQ en ajoutant la nouvelle licence IBM MQ Native HA et Cross-Region Replication Add-on. Pour plus d'informations, voir «Configurer Native HA et Cross-Region Replication sur IBM MQ Queue Managers sous licence en utilisant la fonction IBM MQ Operator», à la page 119 ou «Configuration de Native HA et de Cross-Region Replication sur IBM MQ gestionnaires de files d'attente sous licence dans Kubernetes», à la page 192. Notez que cette mise à jour s'applique également aux gestionnaires de file d'attente IBM MQ 9.4.2.

Modifications

- Qu'est-ce qui a changé dansIBM MQ 9.4.3 .
- Les vulnérabilités corrigées sont détaillées dans ce bulletin de sécurité : https://www.ibm.com/support/pages/node/7236608

9.4.2.1-r2

CD

Version de l'opérateur compatible
Pris en charge de manière autonome ou avec IBM MQ Operator 3.5.3 ou plus.

Architectures prises en charge
amd64, s390x, ppc64le

Images

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.4.2.1-r2` (IBM Cloud Pak for Integration)
- `cp.icr.io/cp/ibm-mqadvanced-server:9.4.2.1-r2` (IBM MQ Advanced)
- `icr.io/ibm-messaging/mq:9.4.2.1-r2` (IBM MQ Advanced for Developers)

Nouveautés

- Quoi de neuf dans le 'IBM MQ 9.4.2 pour Multiplatforms - base et Advanced entitlement.

Modifications

- Qu'est-ce qui a changé dansIBM MQ 9.4.2 .
- Les vulnérabilités corrigées sont détaillées dans ce bulletin de sécurité : https://www.ibm.com/support/pages/node/7234178

9.4.2.1-r1

CD

Version de l'opérateur compatible
Pris en charge de manière autonome ou avec IBM MQ Operator 3.5.2 ou plus.

Architectures prises en charge
amd64, s390x, ppc64le

Images

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.4.2.1-r1` (IBM Cloud Pak for Integration)
- `cp.icr.io/cp/ibm-mqadvanced-server:9.4.2.1-r1` (IBM MQ Advanced)
- `icr.io/ibm-messaging/mq:9.4.2.1-r1` (IBM MQ Advanced for Developers)

Nouveautés

- Quoi de neuf dans le 'IBM MQ 9.4.2 pour Multiplatforms - base et Advanced entitlement.

Modifications

- Qu'est-ce qui a changé dansIBM MQ 9.4.2 .
- Les vulnérabilités corrigées sont détaillées dans ce bulletin de sécurité : https://www.ibm.com/support/pages/node/7232272

9.4.2.0-r2

CD

Version de l'opérateur compatible
Pris en charge de manière autonome ou avec IBM MQ Operator 3.5.1 ou plus.

Architectures prises en charge
amd64, s390x, ppc64le

Images

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.4.2.0-r2` (IBM Cloud Pak for Integration)
- `cp.icr.io/cp/ibm-mqadvanced-server:9.4.2.0-r2` (IBM MQ Advanced)
- `icr.io/ibm-messaging/mq:9.4.2.0-r2` (IBM MQ Advanced for Developers)

Nouveautés

- Quoi de neuf dans le 'IBM MQ 9.4.2 pour Multiplatforms - base et Advanced entitlement.

Modifications

- Qu'est-ce qui a changé dansIBM MQ 9.4.2 .
- Les vulnérabilités corrigées sont détaillées dans ce bulletin de sécurité : https://www.ibm.com/support/pages/node/7228945

9.4.2.0-r1

CD

Version de l'opérateur compatible
Prise en charge autonome ou avec l' IBM MQ Operator 3.5.0 ou supérieur.

Architectures prises en charge
`amd64`, `s390x`, `ppc64le`

Images

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.4.2.0-r1` (IBM Cloud Pak for Integration)
- `cp.icr.io/cp/ibm-mqadvanced-server:9.4.2.0-r1` (IBM MQ Advanced)
- `icr.io/ibm-messaging/mq:9.4.2.0-r1` (IBM MQ Advanced for Developers)

Nouveautés

- Les mesures du gestionnaire de file d'attente peuvent être fournies via HTTPS en renseignant `/etc/mqm/metrics/pki/keys` avec les clés TLS. Pour plus d'informations, voir «Système de fichiers et points de montage», à la page 298.
- Quoi de neuf dans le 'IBM MQ 9.4.2 pour Multiplatforms - base et Advanced entitlement.

Modifications

- Qu'est-ce qui a changé dansIBM MQ 9.4.2 .
- Les vulnérabilités corrigées sont détaillées dans ce bulletin de sécurité : https://www.ibm.com/support/pages/node/7184453

9.4.1.1-r1

CD

Version de l'opérateur compatible
Pris en charge de manière autonome ou avec IBM MQ Operator 3.4.1 ou plus

Architectures prises en charge
`amd64`, `s390x`, `ppc64le`

Images

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.4.1.1-r1` (IBM Cloud Pak for Integration)
- `cp.icr.io/cp/ibm-mqadvanced-server:9.4.1.1-r1` (IBM MQ Advanced)
- `icr.io/ibm-messaging/mq:9.4.1.1-r1` (IBM MQ Advanced for Developers)

Nouveautés

- Quoi de neuf dans le 'IBM MQ 9.4.1 pour Multiplatforms - base et Advanced entitlement.

Modifications

- Qu'est-ce qui a changé dansIBM MQ 9.4.1 .

- Les vulnérabilités corrigées sont détaillées dans ce bulletin de sécurité : https://www.ibm.com/support/pages/node/7182196

9.4.1.0-r2

CD

Version de l'opérateur compatible
Pris en charge de manière autonome ou avec 'IBM MQ Operator 3 3.4.0 ou supérieur

Architectures prises en charge
`amd64`, `s390x`, `ppc64le`

Images

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.4.1.0-r2` (IBM Cloud Pak for Integration)
- `cp.icr.io/cp/ibm-mqadvanced-server:9.4.1.0-r2` (IBM MQ Advanced)
- `icr.io/ibm-messaging/mq:9.4.1.0-r2` (IBM MQ Advanced for Developers)

Nouveautés

- Quoi de neuf dans le 'IBM MQ 9.4.1 pour Multiplatforms - base et Advanced entitlement.

Modifications

- Qu'est-ce qui a changé dansIBM MQ 9.4.1 .
- Les vulnérabilités corrigées sont détaillées dans ce bulletin de sécurité : https://www.ibm.com/support/pages/node/7178065

9.4.1.0-r1

CD

Version de l'opérateur compatible
Pris en charge de manière autonome ou avec 'IBM MQ Operator 3 3.3.0 ou supérieur

Architectures prises en charge
`amd64`, `s390x`, `ppc64le`

Images

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.4.1.0-r1` (IBM Cloud Pak for Integration)
- `cp.icr.io/cp/ibm-mqadvanced-server:9.4.1.0-r1` (IBM MQ Advanced)
- `icr.io/ibm-messaging/mq:9.4.1.0-r1` (IBM MQ Advanced for Developers)

Nouveautés

- La licence 'IBM MQ Advanced for Non-Production Environment est maintenant configurable pour les gestionnaires de file d'attente 9.4.1.0-r1 et ultérieurs. Pour des informations sur la licence, voir '«Références de licences et d'API pour le IBM MQ Operator», à la page 256.
- Nouveautés d' IBM MQ 9.4.1 for Multiplatforms-autorisation d'utilisation de base et Advanced

Modifications

- Correction pour APAR IT46430.
- Correction d'un problème dans le traitement des certificats intermédiaires.
- Correction de la désactivation du mode FIPS pour la génération de la base de données de clés **`runmqakm`** , lorsque FIPS n'est pas activé.
- Correction de la vérification des chemins d'accès aux fichiers.
- Mises à jour pour réduire la taille de l'image.

- l'image 9.4.1.0-r1 du conteneur du gestionnaire de files d'attente IBM MQ inclut la version 3.1.112024.3.0) de l'exit utilisateur de traçage 'IBM MQ Voir le suivi de 'IBM MQ dans la documentation de 'IBM Instana
- Permet à la configuration Native HA d'être fournie de l'extérieur plutôt que d'être générée à partir d'un modèle.
  - Suppression de l'utilisation de la configuration des variables d'environnement de Native HA (à l'exception de 'MQ_NATIVE_HA=true, qui est toujours nécessaire).
  - Clarifier le comportement de la configuration de la variable d'environnement dans 'IBM Documentation.
- Qu'est-ce qui a changé dansIBM MQ 9.4.1 .
- Les vulnérabilités corrigées sont détaillées dans ce bulletin de sécurité : https://www.ibm.com/support/pages/node/7174358

Images compatibles avec le " IBM MQ Operator uniquement

9.4.0.11-r3

CP4I-SC2

Version de l'opérateur requise

«IBM MQ Operator 3.2.13», à la page 10 ou plus

Architectures prises en charge

amd64, s390x, ppc64le

Images

- cp.icr.io/cp/ibm-mqadvanced-server-integration:9.4.0.11-r3 (IBM Cloud Pak for Integration)
- cp.icr.io/cp/ibm-mqadvanced-server:9.4.0.11-r3 (IBM MQ Advanced)

  Remarque : Vous pouvez également utiliser cette image pour déployer des gestionnaires de file d'attente sous licence IBM MQ , à condition de respecter les conditions de licence appropriées et d'annoter ces gestionnaires de file d'attente. Pour plus d'informations, reportez-vous aux sections «Configurer les gestionnaires de files d'attente avec les annotations de licence IBM MQ en utilisant la fonction IBM MQ Operator», à la page 166 et «IBM MQ annotations sur la licence du conteneur», à la page 289.
- icr.io/ibm-messaging/mq:9.4.0.11-r3 (IBM MQ Advanced for Developers)

Nouveautés

- Vous pouvez désormais configurer les gestionnaires de file d'attente déployés avec IBM MQ Operator avec des annotations de licence IBM MQ . Pour plus d'informations, voir «Configurer les gestionnaires de files d'attente avec les annotations de licence IBM MQ en utilisant la fonction IBM MQ Operator», à la page 166. Cette mise à jour s'applique à tous les gestionnaires de files d'attente IBM MQ 9.4

Modifications

- Qu'est-ce qui a changé dansIBM MQ 9.4.0 .
- Les vulnérabilités traitées sont détaillées dans ce bulletin de sécurité : https://www.ibm.com/support/pages/node/7236608

9.4.0.11-r2

CP4I-SC2

Version de l'opérateur requise

«IBM MQ Operator 3.2.12», à la page 11 ou plus

Architectures prises en charge
amd64, s390x, ppc64le

Images

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.4.0.11-r2` (IBM Cloud Pak for Integration)
- `cp.icr.io/cp/ibm-mqadvanced-server:9.4.0.11-r2` (IBM MQ Advanced)
- `icr.io/ibm-messaging/mq:9.4.0.11-r2` (IBM MQ Advanced for Developers)

Nouveautés

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

Modifications

- Qu'est-ce qui a changé dansIBM MQ 9.4.0 .
- Les vulnérabilités traitées sont détaillées dans ce bulletin de sécurité : https://www.ibm.com/support/pages/node/7234178

9.4.0.11-r1

CP4I-SC2

Version de l'opérateur requise
3.2.11 ou plus

Architectures prises en charge
amd64, s390x, ppc64le

Images

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.4.0.11-r1` (IBM Cloud Pak for Integration)
- `cp.icr.io/cp/ibm-mqadvanced-server:9.4.0.11-r1` (IBM MQ Advanced)
- `icr.io/ibm-messaging/mq:9.4.0.11-r1` (IBM MQ Advanced for Developers)

Nouveautés

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

Modifications

- Qu'est-ce qui a changé dansIBM MQ 9.4.0 .

9.4.0.10-r2

CP4I-SC2

Version de l'opérateur requise
3.2.10 ou plus

Architectures prises en charge
amd64, s390x, ppc64le

Images

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.4.0.10-r2` (IBM Cloud Pak for Integration)
- `cp.icr.io/cp/ibm-mqadvanced-server:9.4.0.10-r2` (IBM MQ Advanced)
- `icr.io/ibm-messaging/mq:9.4.0.10-r2` (IBM MQ Advanced for Developers)

Nouveautés

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

Modifications

- Qu'est-ce qui a changé dansIBM MQ 9.4.0 .

9.4.0.10-r1

CP4I-SC2

Version de l'opérateur requise
3.2.9 ou supérieur

Architectures prises en charge
amd64, s390x, ppc64le

Images

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.4.0.10-r1` (IBM Cloud Pak for Integration)
- `cp.icr.io/cp/ibm-mqadvanced-server:9.4.0.10-r1` (IBM MQ Advanced)
- `icr.io/ibm-messaging/mq:9.4.0.10-r1` (IBM MQ Advanced for Developers)

Nouveautés

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

Modifications

- Qu'est-ce qui a changé dansIBM MQ 9.4.0 .

9.4.0.7-r1

CP4I-SC2

Version de l'opérateur requise
3.2.8 ou plus

Architectures prises en charge
amd64, s390x, ppc64le

Images

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.4.0.7-r1` (IBM Cloud Pak for Integration)
- `cp.icr.io/cp/ibm-mqadvanced-server:9.4.0.7-r1` (IBM MQ Advanced)
- `icr.io/ibm-messaging/mq:9.4.0.7-r1` (IBM MQ Advanced for Developers)

Nouveautés

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

Modifications

- Qu'est-ce qui a changé dansIBM MQ 9.4.0 .

9.4.0.6-r2

CP4I-SC2

Version de l'opérateur requise
3.3.2.7 ou plus

Architectures prises en charge
amd64, s390x, ppc64le

Images

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.4.0.6-r2` (IBM Cloud Pak for Integration)

- `cp.icr.io/cp/ibm-mqadvanced-server:9.4.0.6-r2` (IBM MQ Advanced)
- `icr.io/ibm-messaging/mq:9.4.0.6-r2` (IBM MQ Advanced for Developers)

Nouveautés

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

Modifications

- Qu'est-ce qui a changé dansIBM MQ 9.4.0 .

9.4.0.6-r1

CP4I-SC2

Version de l'opérateur requise
3.3.2.6 ou plus

Architectures prises en charge
amd64, s390x, ppc64le

Images

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.4.0.6-r1 (IBM Cloud Pak for Integration)`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.4.0.6-r1 (IBM MQ Advanced)`
- `icr.io/ibm-messaging/mq:9.4.0.6-r1 (IBM MQ Advanced for Developers)`

Nouveautés

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

Modifications

- Qu'est-ce qui a changé dansIBM MQ 9.4.0 .
- Correction appliquée pour résoudre le problème de blocage du conteneur lors de l'exécution de la commande '**endmqm** sur un Mac utilisant Rosetta.

9.4.0.5-r2

CD CP4I-SC2

Version de l'opérateur requise
3.2.5 ou supérieur

Architectures prises en charge
amd64, s390x, ppc64le

Images

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.4.0.5-r2 (IBM Cloud Pak for Integration)`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.4.0.5-r2 (IBM MQ Advanced)`
- `icr.io/ibm-messaging/mq:9.4.0.5-r2 (IBM MQ Advanced for Developers)`

Nouveautés

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

Modifications

- Qu'est-ce qui a changé dansIBM MQ 9.4.0 .
- Basé sur Red Hat Universal Base Image 9.4-1227.1725849298.

9.4.0.5-r1

CD CP4I-SC2

Version de l'opérateur requise
3.2.4 ou supérieur

Architectures prises en charge
amd64, s390x, ppc64le

Images

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.4.0.5-r1 (IBM Cloud Pak for Integration)`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.4.0.5-r1 (IBM MQ Advanced)`
- `icr.io/ibm-messaging/mq:9.4.0.5-r1 (IBM MQ Advanced for Developers)`

Nouveautés

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

Modifications

- Qu'est-ce qui a changé dansIBM MQ 9.4.0 .
- Basé sur Red Hat Universal Base Image 9.4-1194.

9.4.0.0-r3

CD CP4I-SC2

Version de l'opérateur requise
3.2.3 ou supérieur

Architectures prises en charge
amd64, s390x, ppc64le

Images

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.4.0.0-r3 (IBM Cloud Pak for Integration)`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.4.0.0-r3 (IBM MQ Advanced)`
- `icr.io/ibm-messaging/mq:9.4.0.0-r3 (IBM MQ Advanced for Developers)`

Nouveautés

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

Modifications

- Qu'est-ce qui a changé dansIBM MQ 9.4.0 .
- Basé surRed Hat Image de base universelle9.4-1134 .
- Résout un problème observé uniquement sur OpenShift Container Platform (OCP) 4.16, où le rapprochement `ServiceAccount` crée une boucle indéfinie de secrets d'extraction d'images. OCP 4.16 a introduit un changement de comportement où OCP injecte `.metadata.annotation` dans le `ServiceAccount`.

9.4.0.0-r2

CD CP4I-SC2

Version de l'opérateur requise
3.2.2 ou plus

Architectures prises en charge
amd64, s390x, ppc64le

Images

- `cp.icr.io/cp/ibm-mqadvanced-server-integration:9.4.0.0-r2 (IBM Cloud Pak for Integration)`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.4.0.0-r2 (IBM MQ Advanced)`
- `icr.io/ibm-messaging/mq:9.4.0.0-r2 (IBM MQ Advanced for Developers)`

Nouveautés

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

Modifications

- Qu'est-ce qui a changé dansIBM MQ 9.4.0 .
- Basé surRed Hat Image de base universelle9.4-1134 .

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 (IBM Cloud Pak for Integration)`
- `cp.icr.io/cp/ibm-mqadvanced-server:9.4.0.0-r1 (IBM MQ Advanced)`
- `icr.io/ibm-messaging/mq:9.4.0.0-r1 (IBM MQ Advanced for Developers)`

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 Image de base universelle 9.4-949.1716471857. **Note :** UBI 9 est en attente de la certification FIPS 140-3 . UBI 9 n'est pas supporté par 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 28.
- «Prise en charge de IBM MQ dans les conteneurs», à la page 29.
- «Planification du stockage pour le IBM MQ Operator», à la page 39.
- «Planification de la haute disponibilité pour IBM MQ dans des conteneurs», à la page 41.
- «Reprise après incident d'IBM MQ dans des conteneurs», à la page 47.
- «Authentification et autorisation des utilisateurs pour IBM MQ dans les conteneurs», à la page 50.

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

OpenShift

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 45

Certaines fonctions de IBM MQ ne sont pas prises en charge lors de l'utilisation de IBM MQ Operator. Voir «Prise en charge du IBM MQ Operator», à la page 29 pour plus de détails sur ce qui est pris en charge lors de l'utilisation du IBM MQ Operator.

Utilisation des images pré-construites IBM MQ Advanced

V 9.4.1

Vous pouvez utiliser les images préconstruites disponibles dans le IBM Container Registry. Pour plus de détails, voir «Prise en charge des images du conteneur IBM MQ Advanced», à la page 30.

Utilisation des images pré-construites sous licence IBM MQ

V 9.4.1

À partir de IBM MQ 9.4.1, vous pouvez utiliser les images préconstruites disponibles sur IBM Container Registry. Pour plus d'informations, voir «Prise en charge des images de conteneurs sous licence IBM MQ », à la page 32.

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.

Des exemples de génération d'images sont disponibles. Voir «Préparation pour IBM MQ en créant votre propre image de conteneur», à la page 90.

Voir «Prise en charge de la construction de vos propres images de conteneurs IBM MQ», à la page 33 pour plus de détails sur ce qui est pris en charge lors de la création de votre propre image et du code de déploiement.

Référence associée

«Prise en charge de IBM MQ dans les conteneurs», à la page 29
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.

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

Addendum au cycle de vie du support du logiciel IBM Cloud Pak for Integration

Information associée

Prise en charge de SELinux et AppArmor par IBM MQ

OpenShift CP4I MQ Adv. CD CP4I-SC2 Prise en charge du IBM MQ Operator

Le IBM MQ Operator est pris en charge dans le cadre des IBM MQ Advanced et IBM Cloud Pak for Integration

Plateformes prises en charge

Le IBM MQ Operator n'est pris en charge que par le Red Hat OpenShift Container Platform. Les versions de Red Hat OpenShift Container Platform ne sont plus prises en charge par IBM MQ dès que Red Hat cesse d'être prise en charge. Voir «Versions prises en charge pour IBM MQ Operator», à la page 35 et Red Hat OpenShift Container Platform Life Cycle Dates pour plus de détails

Architectures CPU prises en charge

Le IBM MQ Operator est supporté sur `amd64` et `s390x` z/Linux, et sur `ppc64le` Power Systems version 9 et plus. Notez que tous les nœuds du cluster Red Hat OpenShift Container Platform doivent utiliser la même architecture de CPU.

Durée de l'aide

IBM fournit une assistance en matière de défauts et d'utilisation pour le IBM MQ Operator et les images de gestionnaire de file d'attente associées. La durée de l'assistance dépend de la licence que vous utilisez :

- Si vous utilisez une licence **IBM Cloud Pak for Integration**, la IBM MQ Operator est prise en charge pour la même durée que la version IBM Cloud Pak for Integration correspondante. Certaines versions sont IBM Cloud Pak for Integration - Support Cycle 2 (formerly Long Term Support) et d'autres Continuous Delivery (CD). L'opérateur CD et les gestionnaires de file d'attente sont pris en charge jusqu'à la prochaine version IBM Cloud Pak for Integration CD ou CP4I-LTS L'opérateur CP4I-LTS et les gestionnaires de file d'attente déployés avec la licence IBM Cloud Pak for Integration sont pris en charge jusqu'à la prochaine version CP4I-LTS, plus une période de grâce pour permettre la mise à niveau.

  Par exemple, si vous créez un gestionnaire de file d'attente à l'aide de la licence IBM Cloud Pak for Integration “L-JTPV-KYG8TF”, celle-ci est associée à une version CP4I-LTS. Cela permettra de déployer IBM MQ Advanced 9.4.0, et sera pris en charge conformément à cette IBM Cloud Pak for Integration version (deux ans).
- Si vous utilisez un **IBM MQ Advanced** ou une licence **IBM MQ** seul le flux IBM MQ Continuous Delivery est pris en charge. Chaque IBM MQ Operator et gestionnaire de file d'attente déployé avec une licence IBM MQ Advanced ou IBM MQ est pris en charge pendant un an à partir du premier

opérateur qui prend en charge cette version du gestionnaire de file d'attente. Les versions reçoivent de manière proactive des correctifs de sécurité jusqu'à la prochaine version de Continuous Delivery . Par exemple, si vous créez un gestionnaire de file d'attente en utilisant la licence IBM MQ Advanced “L-EHXT-MQCRN9”, il utilisera IBM MQ Advanced 9.4.0 en tant que version Continuous Delivery , et sera pris en charge pendant un an, et des correctifs de sécurité seront publiés jusqu'à la sortie de IBM MQ 9.4.1. Après la publication de la version 9.4.1, sur demande, IBM évaluera la possibilité de faire des efforts raisonnables pour fournir un correctif au niveau de la version précédente du CD

Voir IBM MQ FAQ pour le support à long terme et les versions de livraison continue pour plus d'informations.

Fixer la disponibilité

Des correctifs périodiques et provisoires du gestionnaire de file d'attente sont disponibles auprès de IBM Container Registry. Des correctifs provisoires pour le IBM MQ Operator ne sont pas disponibles.

Caractéristiques non disponibles

Les fonctionnalités suivantes de IBM MQ Advanced ne sont pas disponibles pour les gestionnaires de file d'attente gérés par IBM MQ Operator :

- Les gestionnaires de files d'attente de données répliquées (RDQM) ne sont pas disponibles - cette fonctionnalité est étroitement liée au noyau Linux et n'est donc pas prise en charge dans les conteneurs
- La journalisation linéaire pour les gestionnaires de file d'attente n'est pas disponible
- La spécification d'options de ligne de commande personnalisées pour **crtmqdir**, **crtmqm**, **strmqm** et **endmqm** n'est pas disponible - La plupart des options peuvent être configurées à l'aide d'un fichier INI, mais certaines ne peuvent pas l'être, comme l'utilisation de la journalisation linéaire.
- Les utilisateurs du système d'exploitation (OS) ne sont pas disponibles
- AMQP n'est pas installé
- MQTT n'est pas installé
- L'agent de transfert de fichiers géré n'est pas disponible. Cependant, vous pouvez utiliser l'opérateur IBM MQ pour fournir un ou plusieurs gestionnaires de file d'attente de coordination, de commande ou d'agent.
- La sécurité avancée des messages avec interception MCA à l'aide d'un magasin de clés séparé n'est pas disponible. L'interception MCA peut toujours être configurée pour utiliser le magasin de clés du gestionnaire de la file d'attente principale.

Vous ne pouvez utiliser que les images de conteneurs de gestionnaires de files d'attente fournies par IBM avec l'opérateur MQ et vous ne pouvez pas créer votre propre image avec un code personnalisé. L'exception à cette règle est l'ajout d'une couche d'image de conteneur à une image fournie par IBM, afin d'ajouter des fichiers de configuration ou des sorties.

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

Addendum au cycle de vie du support du logiciel IBM Cloud Pak for Integration

Information associée

Dates du cycle de vie d'IBM MQ dans les conteneurs

MQ Adv. Linux CD V 9.4.1 Prise en charge des images du conteneur IBM MQ Advanced

À partir de la version IBM MQ Advanced 9.4.1, vous pouvez utiliser l'image conteneur préconstruite IBM MQ Advanced sans le IBM MQ Operator

L'image conteneur préconstruite IBM MQ Advanced est composée de plusieurs couches logiques :

- IBM MQ code de validation du conteneur (`runmqserver`, `chkmqhealthy`, `chkmqready` et `chkmqstarted`)
- IBM MQ Advanced lui-même
- Red Hat des paquets de la Red Hat Universal Base Image

Plateformes prises en charge

L'image préconstruite IBM MQ Advanced peut être déployée dans des environnements d'exécution de conteneurs fournis par containerd, cri-o, Podman ou Docker. Pour plus de détails, voir Conditions requises pour IBM MQ. IBM fournit une assistance en cas de défaut et d'utilisation du code d'activation du conteneur MQ IBM et de " IBM MQ Advanced lui-même, mais n'aide pas les utilisateurs à configurer et à déboguer les problèmes liés à leur environnement de conteneurs ou à leur pipeline de déploiement.

Important : Red Hat assure la prise en charge des paquets 'Red Hat uniquement pour les conteneurs fonctionnant sur les hôtes 'Red Hat Enterprise Linux® ou 'Red Hat Enterprise Linux CoreOS Voir Red Hat Container Support Policy pour plus d'informations.

Octroi de licence

L'image préconstruite IBM MQ Advanced est éligible pour les licences de conteneurs si elle est utilisée sur Kubernetes avec le IBM License Service. Si vous n'utilisez pas l'orchestration Kubernetes (par exemple, en utilisant Podman ou Docker), vous ne pouvez pas bénéficier de ce modèle de licence optimisé. Voir Foire aux questions sur les licences de conteneurs pour plus de détails.

Architectures CPU prises en charge

L'image préconstruite IBM MQ Advanced est supportée sur `amd64` et `s390x` z/Linux, et sur `ppc64le` Power Systems version 9 et plus. Notez que tous les conteneurs qui font partie du même gestionnaire de file d'attente (par exemple, les répliques Native HA) doivent utiliser la même architecture de CPU.

Durée de l'aide

Seul le flux IBM MQ Advanced Continuous Delivery est pris en charge Chaque version du gestionnaire de file d'attente est prise en charge conformément à la durée normale de l'assistance (un an).

Les versions reçoivent de manière proactive des correctifs de sécurité jusqu'à la prochaine Continuous Delivery version. Voir IBM MQ FAQ pour le support à long terme et les versions de livraison continue pour plus d'informations.

Fixer la disponibilité

Des correctifs périodiques et provisoires sont disponibles auprès de IBM Container Registry.

Caractéristiques non disponibles

Les fonctionnalités suivantes de IBM MQ Advanced ne sont pas disponibles avec l'image de conteneur préconstruite :

- Les gestionnaires de files d'attente de données répliquées (RDQM) ne sont pas disponibles - cette fonctionnalité est étroitement liée au noyau Linux et n'est donc pas prise en charge dans les conteneurs
- La journalisation linéaire pour les gestionnaires de file d'attente n'est pas disponible
- La spécification d'options de ligne de commande personnalisées pour **`crtmqdir`**, **`crtmqm`**, **`strmqm`** et **`endmqm`** n'est pas disponible - La plupart des options peuvent être configurées à l'aide d'un fichier INI, mais certaines ne peuvent pas l'être, comme l'utilisation de la journalisation linéaire.
- Les utilisateurs du système d'exploitation (OS) ne sont pas disponibles
- AMQP n'est pas installé
- MQTT n'est pas installé

- L'agent de transfert de fichiers géré n'est pas disponible. Cependant, vous pouvez utiliser l'image de conteneur préconstruite pour fournir un ou plusieurs gestionnaires de file d'attente de coordination, de commande ou d'agent.

Si vous souhaitez utiliser ces fonctionnalités, vous pouvez créer votre propre image de 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)

Red Hat Politique de soutien aux conteneurs

Information associée

Dates du cycle de vie d'IBM MQ dans les conteneurs

V 9.4.1 Prise en charge des images de conteneurs sous licence IBM MQ

À partir de IBM MQ 9.4.1, vous pouvez utiliser l'image de conteneur préconstruite IBM MQ sans IBM MQ Operator.

L'image préconstruite du conteneur IBM MQ est composée de plusieurs couches logiques :

- IBM MQ code d'habilitation du conteneur (`runmqserver`, `chkmqhealthy`, `chkmqready` et `chkmqstarted`)
- IBM MQ lui-même
- Red Hat des paquets de la Red Hat Universal Base Image

Remarque : L'image de conteneur sous licence IBM MQ utilise l'image IBM MQ Advanced , bien que les fonctionnalités spécifiques à IBM MQ Advanced ne puissent pas être utilisées sans les licences appropriées. Pour plus d'informations, voir la section «Fonctionnalités indisponibles de IBM MQ Advanced», à la page 33 plus loin dans cette rubrique.

Plateformes prises en charge

L'image préconstruite IBM MQ peut être déployée dans des environnements d'exécution de conteneurs fournis par containerd, cri-o, Podman ou Docker. Pour plus de détails, voir la configuration requise pour IBM MQ. IBM fournit une assistance pour les défauts et l'utilisation du code d'activation du conteneur IBM MQ fourni par IBM et IBM MQ lui-même, mais n'aide pas les utilisateurs à configurer et à déboguer les problèmes liés à leur environnement de conteneurs ou à leur pipeline de déploiement.

Important : Red Hat ne prend en charge les paquets Red Hat que pour les conteneurs fonctionnant sur des hôtes Red Hat Enterprise Linux ou Red Hat Enterprise Linux CoreOS . Voir Red Hat Container Support Policy pour plus d'informations.

Octroi de licence

L'image préconstruite IBM MQ est éligible pour les licences de conteneurs si elle est utilisée sur Kubernetes avec l'image IBM License Service. Si vous n'utilisez pas l'orchestration Kubernetes (par exemple, en utilisant Podman ou Docker), vous ne pouvez pas bénéficier de ce modèle de licence optimisé. Pour plus de détails, voir la FAQ sur les licences de conteneurs.

Architectures CPU prises en charge

L'image préconstruite IBM MQ est prise en charge sur `amd64` et `s390x` z/ Linux, et sur `ppc64le` Power Systems version 9 et supérieure. Notez que tous les conteneurs qui font partie du même gestionnaire de file d'attente (par exemple, les répliques Native HA) doivent utiliser la même architecture CPU.

Durée de l'aide

Seul le flux de livraison continue IBM MQ est pris en charge. Chaque version du gestionnaire de file d'attente est prise en charge conformément à la durée normale de l'assistance (un an). Les versions reçoivent de manière proactive des correctifs de sécurité jusqu'à la prochaine version de Continuous

Delivery . Pour plus d'informations, voir IBM MQ FAQ pour les versions de support à long terme et de livraison continue.

Fixer la disponibilité

Des correctifs périodiques et provisoires sont disponibles à l'adresse suivante : IBM Container Registry.

Fonctionnalités indisponibles de IBM MQ Advanced

Les fonctionnalités suivantes de IBM MQ Advanced ne peuvent pas être utilisées avec l'image pré-construite du conteneur sous licence IBM MQ :

- L'HA native et la réplication interrégionale. Pour utiliser les fonctions Native HA et Cross-Region Replication sur un gestionnaire de files d'attente sous licence IBM MQ , voir «Configurer Native HA et Cross-Region Replication sur IBM MQ Queue Managers sous licence en utilisant la fonction IBM MQ Operator», à la page 119.
- AMS
- Managed File Transfer prise en charge du type OpenSSH Clé privée pour les serveurs SFTP
- Restauration automatique des objets endommagés pour les gestionnaires de file d'attente Native HA
- Planification de l'image des médias - journalisation répliquée
- RDQM disponible sur Red Hat Enterprise Linux 9
- Kafka Connexion

Si vous souhaitez utiliser ces fonctionnalités, vous pouvez déployer IBM MQ Advanced. Pour plus d'informations, voir Prise en charge des images de conteneurs IBM MQ Advanced.

Linux Prise en charge de la construction de vos propres images de conteneurs IBM MQ

L'exemple de conteneur " IBM MQ fournit un exemple de code non pris en charge pour activer " IBM MQ dans un conteneur. 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.

Téléchargements

L'exemple de conteneur " IBM MQ est disponible ici : https://github.com/ibm-messaging/mq-container.

Remarque : L'exemple de conteneur, et tout code que vous ajoutez ou modifiez, n'est pas pris en charge par " IBM".

Pour l'utilisation de l'exemple de conteneur, vous téléchargez une image " IBM MQ sans installation.

- Si vous disposez d'une licence " IBM MQ Advanced for Developers, utilisez la version " Continuous Delivery (CD) de " IBM MQ". Ces images non installées sont disponibles sur le CD Downloading IBM MQ 9.4
- Si vous disposez d'une licence de production, utilisez la version 'Long Term Support (LTS) ou 'CD de 'IBM MQ. Ces images non installées sont disponibles auprès de Fix Central.

Plateformes prises en charge

Voir Conditions requises pour IBM MQ pour des informations détaillées sur les environnements d'exécution de conteneurs pris en charge et les Linux distributions. IBM MQ peut être utilisé avec les environnements d'exécution de conteneurs fournis par containerd, cri-o, Podman ou 'Docker. IBM fournit une assistance en cas de défaut et d'utilisation du logiciel de base 'IBM MQ, mais ne vous aide pas à configurer et à déboguer les problèmes liés à votre environnement de conteneurs ou à votre pipeline de déploiement.

L'exemple de conteneur utilise une Red Hat Universal Base Image comme image de base. Red Hat ne prend en charge les paquets Red Hat que pour les conteneurs fonctionnant sur des hôtes Red Hat Enterprise Linux ou Red Hat Enterprise Linux CoreOS. Voir Red Hat Container Support Policy pour plus d'informations. Vous pouvez créer votre propre image de conteneur sur n'importe quelle image de base " Linux correspondant à une distribution " Linux prise en charge par " IBM MQ, par exemple :

- Red Hat Universal Base Image
- Ubuntu
- SUSE Linux Enterprise Server Image du conteneur de base

Architectures CPU prises en charge

Voir Configuration requise pour IBM MQ.

Durée de l'aide pour le " IBM MQ

Le support est fourni pour les versions de production des images de non-installation 'IBM MQ Pour le " LTS, la durée de l'assistance est de 5 ans à compter de la disponibilité générale. Pour le " CD, chaque version incrémentale est prise en charge pendant un an à compter de la date de disponibilité générale. Pour plus d'informations, voir la FAQIBM MQ pour les versions Long Term Support et Continuous Delivery.

Fixer la disponibilité pour 'IBM MQ

Pour les versions de production des images non installées " IBM MQ, des correctifs sont fournis pendant toute la durée de l'assistance. Vous pouvez également demander des corrections. Si une mise à jour " CD n'est plus éligible pour les corrections de défauts, mettez à jour vers une version qui est éligible, et si le défaut peut être recréé, une correction sera fournie. Pour plus d'informations sur les versions 'LTS et 'CD disponibles et les derniers correctifs, voir IBM MQ fixes by version.

Caractéristiques non disponibles

Vous ne pouvez pas utiliser les utilisateurs du système d'exploitation (OS) avec une image de non-installation " IBM MQ Si vous souhaitez utiliser les utilisateurs du système d'exploitation, vous devez créer votre propre image à l'aide du programme d'installation RPM, pour lequel il n'existe pas d'exemple.

Voir " Configuration requise pour " IBM MQ pour plus de détails sur les composants pris en charge. Des échantillons sont disponibles, mais pas pour tous les composants.

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)

Tâches associées

«Préparation pour IBM MQ en créant votre propre image de conteneur», à la page 90
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.

MQ Adv. Linux Disponibilité de l'image du conteneur IBM MQ Advanced for Developers

L'image préconstruite IBM MQ Advanced for Developers est disponible à partir de IBM Container Registry : `icr.io/ibm-messaging/mq`. L'image est téléchargée gratuitement et n'est pas justifiée. **L'image n'est pas prise en charge.**

Plates-formes disponibles

L'image préconstruite IBM MQ Advanced for Developers peut être déployée dans des environnements d'exécution de conteneurs fournis par containerd, cri-o, Podman ou Docker. Pour plus de détails, voir Conditions requises pour IBM MQ.

Architectures CPU disponibles

L'image pré-construite IBM MQ Advanced for Developers est disponible sur `amd64` et `s390x` z/Linux, et sur `ppc64le` Power Systems version 9 et supérieures. Notez que tous les conteneurs qui font partie du même gestionnaire de file d'attente (par exemple, les répliques Native HA) doivent utiliser la même architecture de CPU.

Caractéristiques non disponibles

Les fonctionnalités suivantes de IBM MQ Advanced for Developers ne sont pas disponibles avec l'image de conteneur préconstruite :

- Les gestionnaires de files d'attente de données répliquées (RDQM) ne sont pas disponibles - cette fonctionnalité est étroitement liée au noyau Linux et n'est donc pas prise en charge dans les conteneurs
- La journalisation linéaire pour les gestionnaires de file d'attente n'est pas disponible
- La spécification d'options de ligne de commande personnalisées pour **`crtmqdir`**, **`crtmqm`**, **`strmqm`** et **`endmqm`** n'est pas disponible - La plupart des options peuvent être configurées à l'aide d'un fichier INI, mais certaines ne peuvent pas l'être, comme l'utilisation de la journalisation linéaire.
- Les utilisateurs du système d'exploitation (OS) ne sont pas disponibles
- AMQP n'est pas installé
- MQTT n'est pas installé
- L'agent de transfert de fichiers géré n'est pas disponible. Cependant, vous pouvez utiliser l'image de conteneur préconstruite pour fournir un ou plusieurs gestionnaires de file d'attente de coordination, de commande ou d'agent.

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)

«IBM MQ Advanced for Developers image de conteneur», à la page 300

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, Kubernetes et d'autres environnements de conteneurs.

OpenShift CP4I CD CP4I-SC2 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», à la page 36
- «Versions Red Hat OpenShift Container Platform compatibles», à la page 36
- «Versions IBM Cloud Pak for Integration», à la page 37
- «Versions IBM MQ disponibles dans les anciens opérateurs», à la page 37
- «Versions de OpenShift Container Platform compatibles avec les opérateurs plus anciens», à la page 37

Versions IBM MQ disponibles

Canal opérateur	Version de l'opérateur	Versions IBM MQ									
		9.4.3	9.4.2	9.4.1	9.4.0	9.3.5	9.3.4	9.3.3	9.3.2	9.3.1	9.3.0
v3.6	3.6	CD	DEP	DEP	MIG	DEP	DEP	DEP	DEP	DEP	MIG
v3.5	3.5		CD	DEP	MIG	DEP	DEP	DEP	DEP	DEP	MIG
v3.4	3.4			CD	MIG	DEP	DEP	DEP	DEP	DEP	MIG
v3.3	3.3			CD	MIG	DEP	DEP	DEP	DEP	DEP	MIG
v3.2-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[1]					
		4.18	4.17	4.16	4.15	4.14	4.12
v3.6	3.6.0 à partir de	CD	CD	CD	CD	CD	CD
v3.5	3.5.1 à partir de	CD	CD	CD	CD	CD	CD
	3.5.0 et au-delà		CD	CD	CD	CD	CD
v3.4	3.3.4.0 et plus		CD	CD	CD	CD	CD
v3.3	3.3.3.0 et plus		CD	CD	CD	CD	CD

[1] 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.

Canal opérateur	Version de l'opérateur	Versions OpenShift Container Platform[1]					
		4.18	4.17	4.16	4.15	4.14	4.12
v3.2-sc2	3.2.10 à partir de	SC2	SC2	SC2	SC2	SC2	SC2
	3.3.2.6 et suivants		SC2	SC2	SC2	SC2	SC2
	3.2.3 et suivantes			SC2	SC2	SC2	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 .

Versions IBM Cloud Pak for Integration

Chaque version de IBM MQ Operator peut être utilisée dans le cadre d'une version de IBM Cloud Pak for Integration (comme indiqué dans le tableau suivant) ou indépendamment.

Version de IBM Cloud Pak for Integration	Versions IBM MQ Operator
16.1.2	3.6.x
16.1.1	3.4.x, 3.5.x
16.1.0	3.2.x, 3.3.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 .

OpenShift MQ Adv. 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
```

[1] 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.

```
      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 `security Context`, 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 des licences de conteneurs, le site IBM License Service doit être utilisé pour suivre l'utilisation des licences et déterminer les droits dont vous avez besoin.

Notez que les licences basées sur des conteneurs ne sont disponibles que si vous êtes sur Red Hat OpenShift Container Platform ou Kubernetes. Pour plus d'informations, consultez la FAQ sur les licences de conteneurs.

Pour utiliser le site IBM License Service:

- Si vous utilisez le IBM MQ Operator, voir l'étape Déployer le License Service du IBM MQ Operator processus d'installation.
- Si vous déployez sur Kubernetes, voir Installer License Service sur le cluster Kubernetes dans la documentation IBM Cloud Pak foundational services .

Si vous n'utilisez pas Red Hat OpenShift Container Platform ou Kubernetes, vous devez utiliser IBM License Metric Tool. Cet outil est nécessaire pour mesurer et rapporter l'utilisation des licences des produits IBM qui sont déployés dans des environnements de virtualisation traditionnels afin de tirer profit des licences de sous-capacité. Pour plus d'informations, voir IBM License Metric Tool documentation.

Référence associée

«IBM MQ annotations sur la licence du conteneur», à la page 289
Dans les environnements Kubernetes et Red Hat OpenShift Container Platform , IBM License Service utilise les annotations Pod pour suivre l'utilisation en fonction des limites définies pour le conteneur, plutôt que pour la machine sous-jacente. Le site IBM MQ Operator définit ces annotations Pod pour vous, mais lorsque vous utilisez d'autres moyens de déploiement, vous devez les définir vous-même.

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 164). 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

OpenShift CP4I

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 41.
- **`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 43.

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 173.

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.

OpenShift Kubernetes 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

MQ Adv.

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 43.

HA CRR natif (réplication interrégionale)

MQ Adv. V 9.4.2

Vous pouvez implémenter une solution de reprise après incident qui comprend deux groupes Native HA situés dans des régions distinctes. L'un des groupes Native HA a le rôle d'**exécution**, tandis que l'autre groupe a le rôle de **récupération**. Lorsqu'une panne planifiée ou imprévue survient dans la région où se trouve le groupe actif, une action de l'utilisateur peut faire basculer le rôle du groupe de la région distante de la récupération à l'activité. Ce groupe reprend ensuite le travail de l'instance de gestionnaire de files d'attente d'origine, en acceptant les connexions d'application, etc.

Pour plus d'informations, voir «Réplication inter-régions native HA», à la page 47.

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. Lors de l'utilisation d'une charge de travail avec état comme " IBM MQ avec un " Kubernetes " StatefulSet, ", si un nœud maître " Kubernetes perd le contact avec un nœud de travail, il ne peut pas déterminer si le nœud est tombé en panne ou s'il a simplement perdu la connectivité 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 115
vous pouvez configurer la haute disponibilité pour les gestionnaires de files d'attente en utilisant soit la solution Native HA, soit la solution de gestionnaire de files d'attente multi-instances.

CP4I MQ Adv. Kubernetes 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 performance et de fiabilité, il est recommandé d'utiliser le stockage persistant RWOReadWriteOnce) 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 le provisionnement dynamique des volumes et le "volumeBindingMode: WaitForFirstConsumer".

Les fournisseurs suivants sont explicitement interdits:

- NFS
- GlusterFS
- Autres fournisseurs non bloqués.

Remarque : V 9.4.2 Si vous configurez la réplication interrégionale de Native HA, vous aurez besoin d'au moins deux fois plus d'espace disque que pour Native HA. Ceci est dû à la nécessité de disposer d'un répertoire de sauvegarde des journaux lors du rebasage (voir «Réplication inter-régions native HA», à la page 47)

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

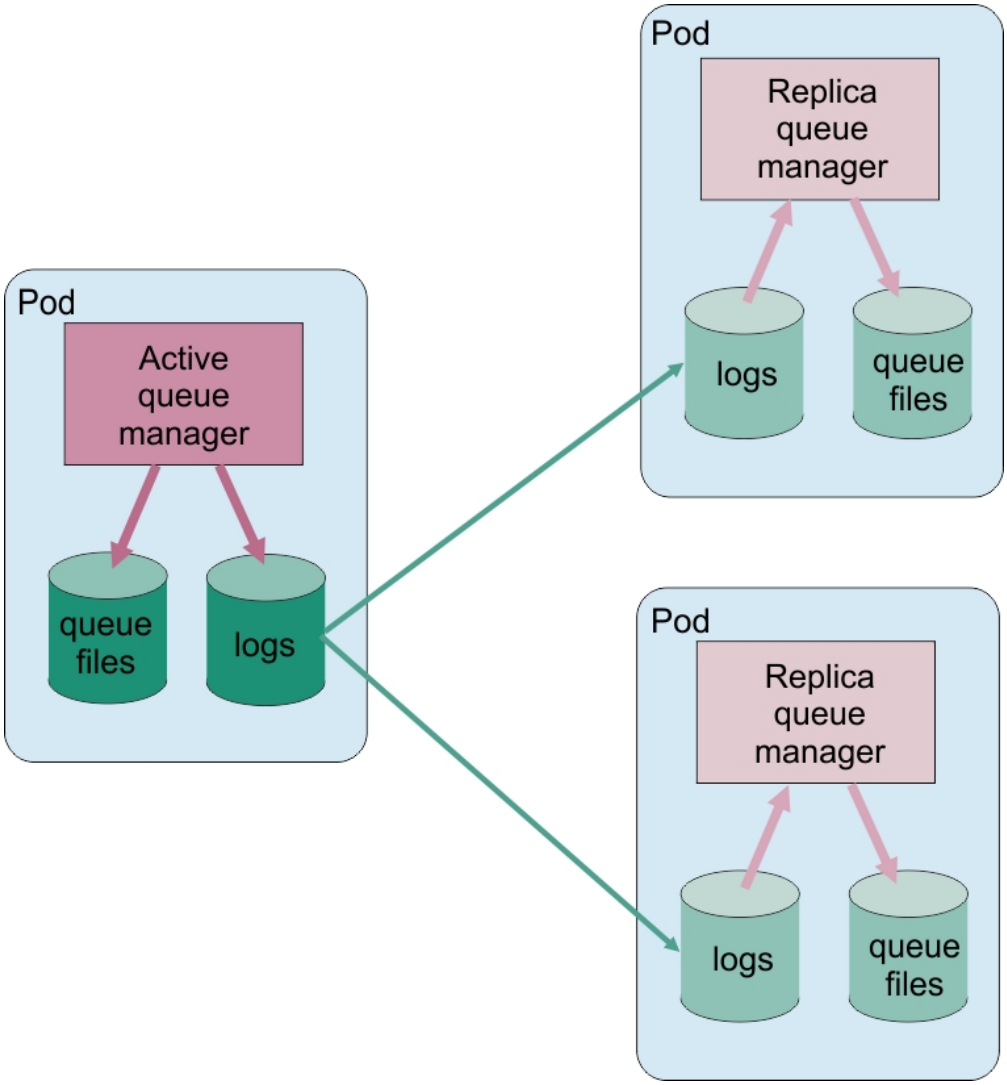


Figure 1. Exemple de configuration Native HA

MQ Adv. Kubernetes 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 28
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

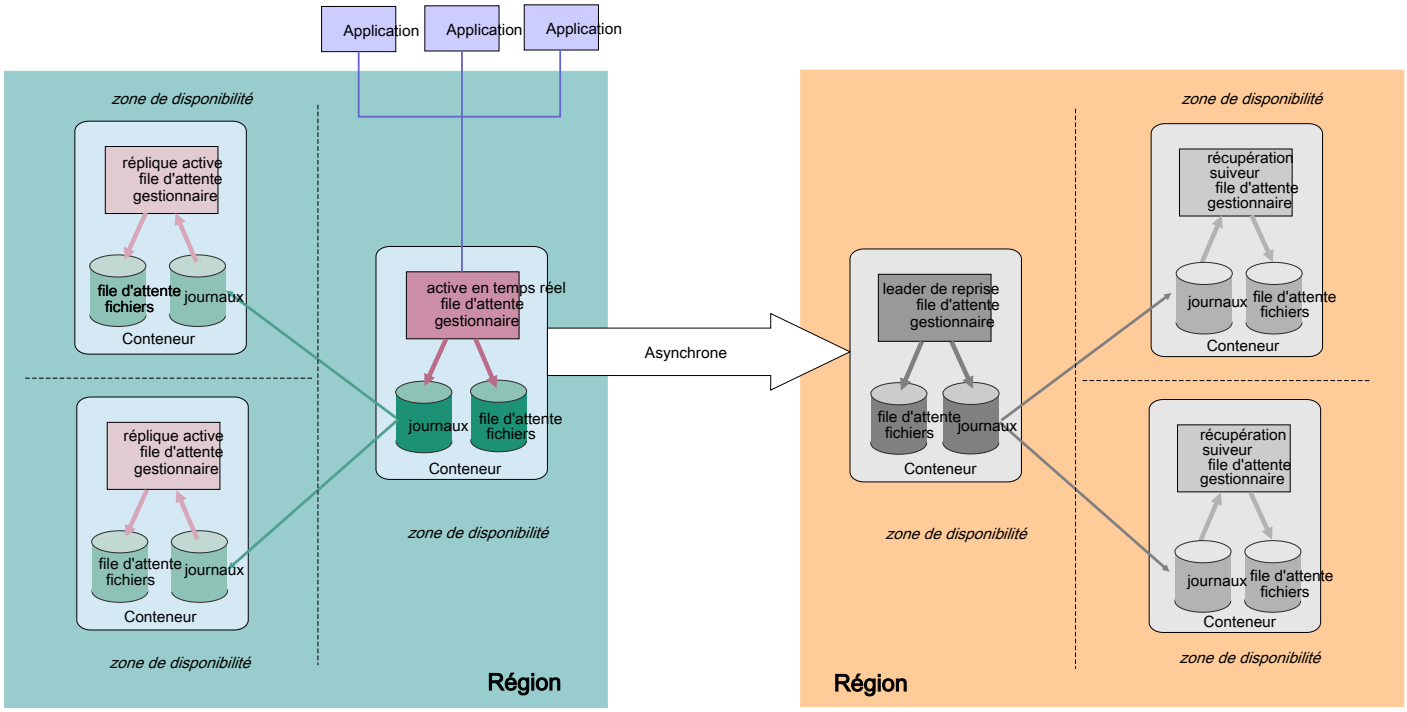
ha_for_ctr.dita

V 9.4.2 Réplication inter-régions native HA

Vous pouvez mettre en œuvre une configuration Native HA Cross-Region Replication (CRR) basée sur la solution Native HA. La configuration CRR permet la reprise après sinistre.

Consultez le site «Native HA», à la page 43 pour obtenir une présentation de la solution Native HA.

Native HA utilise un groupe de trois instances de gestionnaire de file d'attente pour conserver une copie répliquée des données de journalisation. Le groupe élit automatiquement une instance active en comparant et en choisissant la meilleure copie des données. Le CRR étend la réplication des journaux pour inclure un groupe (région) potentiellement éloigné géographiquement, mais également hautement disponible. Lorsqu'une panne planifiée ou non est détectée sur le site principal, le groupe présent sur le site de secours peut prendre le relais. Vous définissez d'abord le rôle d'un groupe, puis vous changez manuellement le rôle d'un groupe si nécessaire pour effectuer un basculement planifié ou lorsqu'une panne imprévue se produit. Les données du journal sont répliquées de manière asynchrone vers le groupe de reprise, de sorte qu'il peut y avoir une certaine perte de données lors d'un basculement à la suite d'une panne non planifiée.



Rôles demandés et rôles effectifs

Un rôle demandé (Requested role) est le paramètre de configuration d' GroupRole s pour une instance Native HA qui est spécifiée dans la strophe d' `NativeHALocalInstance` s de son fichier d' `qm.ini`. Le paramètre demande à une instance de coopérer avec les autres instances de son groupe pour remplir un rôle. Une fois qu'une majorité de cas partagent le même rôle demandé, une élection peut avoir lieu pour choisir un chef de groupe. Lorsqu'une élection est gagnée, le rôle demandé est comparé au dernier rôle que l'instance a utilisé pour dériver un rôle effectif. Voir la strophe NativeHAInstance du fichier qm.ini et la strophe NativeHALocalInstance du fichier qm.ini.

Rôle en direct

Un groupe Live est un gestionnaire de file d'attente Native HA standard composé de trois instances. Une instance est choisie comme instance active et cette instance accepte les nouvelles tâches des applications. Un gestionnaire de file d'attente Native HA existant peut être migré vers une configuration Native CRR.

Si un rôle demandé n'est pas explicitement spécifié dans l' `qm.ini`, le rôle par défaut est considéré comme étant Live. Un leader élu d'un groupe Live est également connu sous le nom de gestionnaire de file d'attente actif. Cette instance peut accepter des connexions d'applications et d'outils pour effectuer de nouveaux travaux de messagerie et gérer les objets du gestionnaire de file d'attente.

Rôle de recouvrement

Un groupe de récupération se compose également de trois instances et l'une de ces instances est élue leader. Le leader agit uniquement en tant que mandataire, acceptant le travail envoyé par un groupe Live associé. Aucune connexion d'application n'est autorisée au groupe Recovery, bien que certaines commandes de contrôle soient autorisées afin que des opérations telles que la fin de l'instance du gestionnaire de file d'attente puissent être effectuées.

La strophe « `NativeHARecoveryGroup` » du fichier de configuration « `qm.ini` » du groupe «Live» identifie l'adresse réseau du groupe «Recovery».

Transitions de fonctions en cours

Lors d'un basculement planifié où le groupe Actif actuel devient le nouveau groupe de Reprise (et le groupe de Reprise actuel devient le nouveau groupe Actif), une coordination entre les groupes est

nécessaire. Cette coordination garantit l'identité des journaux et permet d'atteindre un RPO (objectif de point de récupération) de zéro.

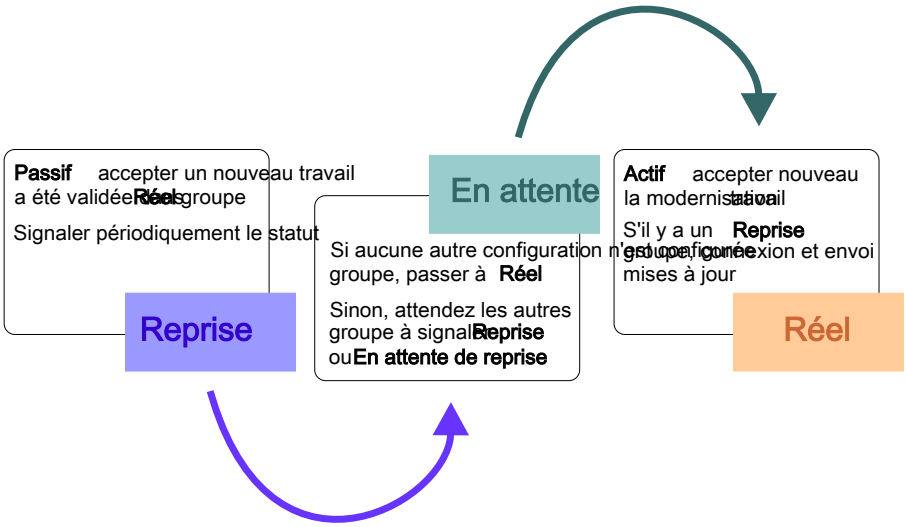
Lorsque deux groupes coordonnent le passage entre Recovery et Live, un rôle Pending efficace est temporairement adopté par les deux groupes.

Rôle de reprise vers rôle de production (en attente de production)

Un groupe qui a précédemment opéré dans un rôle de récupération et qui a l'intention de passer à un rôle de direct prend la décision d'utiliser le rôle en fonction de la présence ou non d'un autre groupe configuré.

S'il n'y a pas d'autre groupe configuré, il passe immédiatement au rôle Live.

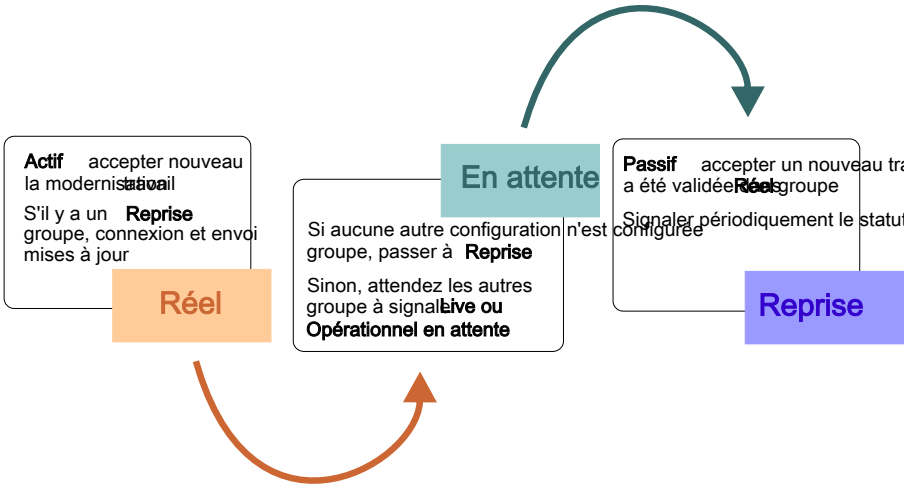
Si un autre groupe est configuré et activé, la transition vers un rôle Live est bloquée jusqu'à ce que l'autre groupe adopte un rôle Recovery ou Pending Recovery, et jusqu'à ce que les deux groupes soient synchronisés.



Rôle actif vers Rôle de récupération (récupération en attente)

Un groupe qui avait précédemment opéré dans un rôle Live et qui a l'intention de passer à un rôle Recovery décidera du moment où il utilisera le rôle Recovery, en fonction de la présence ou non d'un autre groupe configuré.

Si un autre groupe est configuré et activé, la transition vers un rôle de récupération est bloquée jusqu'à ce que l'autre groupe adopte un rôle actif ou en attente d'activation et jusqu'à ce que les deux groupes soient synchronisés.



Réplication et rebasage

Lorsque les deux groupes d'une configuration Native HA CRR sont connectés, les mises à jour du journal sont transmises au leader du groupe de récupération, puis aux autres instances de récupération. Il s'agit de la **réplication**.

Si la connexion réseau entre les deux groupes est perdue, il se peut que le groupe de récupération ne puisse pas rattraper son retard en utilisant des journaux répliqués (par exemple, parce que le groupe actif a réutilisé des extensions de journaux qui seraient nécessaires pour le rattrapage). Dans ce cas, le leader de la reprise effectue **un rebasage**. Un rebasage est le processus qui consiste à rejeter les données de journal reçues et à reconstruire le gestionnaire de file d'attente à partir d'un ensemble complet de données de journal envoyées par le groupe en direct. Une fois que le leader de la reprise a terminé le rebasage, les journaux sont répliqués vers les instances de reprise.

Si un problème survient lors d'une opération de rebasage, il est possible que le gestionnaire de file d'attente ne puisse démarrer sur aucun des membres du groupe de récupération. Pour éviter cette éventualité, une sauvegarde du journal est effectuée avant le début de l'opération de rebasage. Le gestionnaire de file d'attente peut alors être redémarré en utilisant les données du journal de sauvegarde si la réinitialisation échoue. Le journal de sauvegarde est supprimé une fois la rebase réussie.

En raison de l'exigence d'un journal de sauvegarde, une configuration Native HA CRR nécessite au moins deux fois l'espace de stockage d'une configuration Native HA.

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 50
  - «Contraintes de sécurité liées à l'utilisation des utilisateurs du système d'exploitation dans les conteneurs», à la page 51
- «Remarques relatives à la restriction du trafic réseau à IBM MQ dans les conteneurs», à la page 51

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 51.

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 109
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
```

FIPS conformité pour IBM MQ dans les conteneurs

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

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. Le NIST produit des recommandations et des normes, notamment le site Federal Information Processing Standards (FIPS).

Une norme importante de FIPS est FIPS 140-2 , qui exige l'utilisation d'algorithmes cryptographiques puissants. FIPS 140-2 spécifie également les exigences relatives aux algorithmes de hachage à utiliser pour protéger les paquets contre les modifications en transit.

IBM MQ fournit un support FIPS 140-2 s'il a été configuré à cet effet.

Remarque : Sur AIX, Linux, and Windows, IBM MQ assure la conformité à FIPS 140-2 grâce au module cryptographique IBM Crypto for C (ICC) . Le certificat de ce module a été déplacé vers le statut Historique. Les clients doivent consulter le certificat IBM Crypto for C (ICC) et prendre connaissance de tous les conseils fournis par le NIST.

V 9.4.2 Le module cryptographique FIPS 140-3 au sein de IBM Semeru Runtime a été approuvé par le NIST en août 2024. IBM MQ 9.4.2 ajoute la prise en charge du traitement des connexions clients IBM MQ classes for JMS et IBM MQ classes for Java utilisant TLS pour FIPS 140-3 dans Java 8 et IBM Semeru Runtime 11+. La certification NIST associée au module FIPS 140-3 peut être consultée à l'adresse suivante : https://csrc.nist.gov/projects/cryptographic-module-validation-program/certificate/4755. Le fournisseur FIPS 140-2 reste le profil par défaut. IBM MQ 9.4.2 ne modifie pas le comportement par défaut mais permet de configurer des connexions avec FIPS 140-3.

Pour IBM MQ in Containers, l'image du conteneur IBM MQ Operator 3.2.0 et du gestionnaire de file d'attente 9.4.0.0 sont basées sur UBI 9. FIPS 140-3 la conformité de IBM MQ dans les conteneurs est actuellement en cours.

Exigences

Pour les exigences relatives à la configuration des clusters et d'autres considérations, voir FIPS Wall : Approche actuelle de IBM pour la conformité à FIPS .

IBM MQ dans des conteneurs peuvent fonctionner en mode de conformité avec 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 FIPS . Si le système d'exploitation hôte est conforme à FIPS et que les clés privées et les certificats ont été fournis, le conteneur IBM MQ configure le gestionnaire de file d'attente, le serveur web IBM MQ et le transfert de données entre les nœuds dans le cadre d'un déploiement Native High Availability, afin qu'ils fonctionnent en mode de conformité à 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 file 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 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 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 point d'accès est uniquement accessible à l'adresse HTTP. Vous pouvez désactiver le point de terminaison des mesures pour rendre le reste de IBM MQ FIPS conforme.
- 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. 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 respect de FIPS .
- IBM MQ Operator l'accès aux services de gestion de l'identité et de l'accès (IAM)/Zen de IBM n'est pas conforme à FIPS .

Comment la conformité à FIPS est détectée et comment le support de FIPS est configuré automatiquement

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

Remarque : Sur AIX, Linux, and Windows, IBM MQ assure la conformité à FIPS 140-2 grâce au module cryptographique IBM Crypto for C (ICC) . Le certificat de ce module a été déplacé vers le statut Historique. Les clients doivent consulter le certificat IBM Crypto for C (ICC) et prendre connaissance de tous les conseils fournis par le NIST.

V 9.4.2 Le module cryptographique FIPS 140-3 au sein de IBM Semeru Runtime a été approuvé par le NIST en août 2024. IBM MQ 9.4.2 ajoute la prise en charge du traitement des connexions clients IBM MQ classes for JMS et IBM MQ classes for Java utilisant TLS pour FIPS 140-3 dans Java 8 et IBM Semeru Runtime 11+. La certification NIST associée au module FIPS 140-3 peut être consultée à l'adresse suivante : https://csrc.nist.gov/projects/cryptographic-module-validation-program/certificate/4755. Le fournisseur FIPS 140-2 reste le profil par défaut. IBM MQ 9.4.2 ne modifie pas le comportement par défaut mais permet de configurer des connexions avec FIPS 140-3.

Pour IBM MQ in Containers, l'image du conteneur IBM MQ Operator 3.2.0 et du gestionnaire de file d'attente 9.4.0.0 sont basées sur UBI 9. FIPS 140-3 la conformité de IBM MQ dans les conteneurs est actuellement en cours.

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

Gestionnaire de files d'attente
: Si le système d'exploitation hôte est compatible avec FIPS et que la clé privée et les certificats sont fournis, l'attribut du gestionnaire de file d'attente **SSLFIPS** est défini sur YES. Dans le cas contraire, l'attribut **SSLFIPS** prend la valeur NO .

IBM MQ serveur Web
: Le serveur web IBM MQ fournit une interface HTTP / HTTPS pour administrer IBM MQ. Si le système d'exploitation hôte est conforme à FIPS , les options de la JVM sont mises à jour pour que le serveur web utilise une cryptographie conforme à 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 Yes :

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

```
    CipherSpec=ECDHE_RSA_AES_256_GCM_SHA384
    SSLFipsRequired=Yes
```

Si le conteneur est exécuté dans un cluster OpenShift sans support FIPS , le gestionnaire de file d'attente, le serveur web IBM MQ et les composants Native HA n'ont pas leur support FIPS automatiquement activé. Seule l'architecture x86-64 est actuellement prise en charge par la plateforme OpenShift pour FIPS. Pour les architectures Power et Linux on IBM Z , OpenShift n'offre pas de support 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 gestionnaire de file d'attente YAML. L'extrait 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
```

Remplacer le mode automatique FIPS 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 du conteneur.

Avant de commencer

Remarque : Sur AIX, Linux, and Windows, IBM MQ assure la conformité à FIPS 140-2 grâce au module cryptographique IBM Crypto for C (ICC) . Le certificat de ce module a été déplacé vers le statut Historique. Les clients doivent consulter le certificat IBM Crypto for C (ICC) et prendre connaissance de tous les conseils fournis par le NIST.

V 9.4.2 Le module cryptographique FIPS 140-3 au sein de IBM Semeru Runtime a été approuvé par le NIST en août 2024. IBM MQ 9.4.2 ajoute la prise en charge du traitement des connexions clients IBM MQ classes for JMS et IBM MQ classes for Java utilisant TLS pour FIPS 140-3 dans Java 8 et IBM Semeru Runtime 11+. La certification NIST associée au module FIPS 140-3 peut être consultée à l'adresse suivante : https://csrc.nist.gov/projects/cryptographic-module-validation-program/certificate/4755. Le fournisseur FIPS 140-2 reste le profil par défaut. IBM MQ 9.4.2 ne modifie pas le comportement par défaut mais permet de configurer des connexions avec FIPS 140-3.

Pour IBM MQ in Containers, l'image du conteneur IBM MQ Operator 3.2.0 et du gestionnaire de file d'attente 9.4.0.0 sont basées sur UBI 9. FIPS 140-3 la conformité de IBM MQ dans les conteneurs est actuellement en cours.

Pourquoi et quand exécuter cette tâche

MQ_ENABLE_FIPS prend en charge trois valeurs:

Automatique

Il s'agit de la valeur par défaut.

Si le système d'exploitation hôte est compatible avec FIPS , tous les composants (gestionnaire de file d'attente, serveur web IBM MQ et Native HA) fonctionnent en mode FIPS .

Si le système d'exploitation hôte n'est pas compatible avec FIPS , tous les composants ne fonctionnent pas en mode FIPS .

Oui

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

L'attribut du gestionnaire de file d'attente **SSLFIPS** est défini sur YES même si IBM MQ dans les conteneurs est exécuté sur un système d'exploitation hôte qui n'est pas conforme à FIPS . Par exemple, si le gestionnaire de file d'attente IBM MQ , le serveur web et Native HA sont conformes à FIPS , mais que le système d'exploitation du conteneur ne l'est pas.

false

Cette valeur désactive la conformité à FIPS .

L'attribut du gestionnaire de file d'attente **SSLFIPS** est défini sur NO, même si IBM MQ dans les conteneurs est exécuté sur une machine hôte conforme à FIPS . Cependant, 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 point final HTTPS si la clé privée et les certificats sont fournis.

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

Exemple

Voici un exemple de YAML de gestionnaire de file d'attente qui décrit l'activation de TLS et de FIPS pour le composant de gestionnaire de file d'attente.

Remarque : Cet exemple utilise une licence IBM MQ Advanced , mais vous pouvez également utiliser une licence IBM MQ . Pour plus d'informations, voir «Configurer les gestionnaires de files d'attente avec les annotations de licence IBM MQ en utilisant la fonction IBM MQ Operator», à la page 166.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  namespace: ibm-mq-fips
  name: exampleqm
spec:
  license:
    accept: true
    license: L-NUUP-23NH8Y
    use: Production
  queueManager:
    name: EXAMPLEQM
    storage:
      queueManager:
        type: ephemeral
  template:
    pod:
      containers:
        - env:
            - name: MQ_ENABLE_FIPS
              value: "true"
          name: qmgr
  version: 9.4.3.0-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 57

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 90

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 58
- «Droits d'accès au cluster requis par IBM MQ Operator», à la page 58
- «Vérification des signatures d'image», à la page 58
- «Installation d'IBM MQ Operator», à la page 59
- «Mise à niveau de IBM MQ Operator et des gestionnaires de files d'attente», à la page 71
- «Désinstallation de IBM MQ Operator», à la page 86

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 59 pour plus de détails sur les dépendances.

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

Le site IBM MQ Operator a toujours besoin d'autorisations à l'échelle du cluster pour lire les informations relatives à la classe de stockage et à la version du cluster.

Le site IBM MQ Operator peut être installé dans deux portées différentes :

1. L' **étendue du cluster** signifie que l'opérateur se verra attribuer des autorisations pour utiliser les ressources `ClusterRole` et `ClusterRoleBinding` . Ceux-ci permettront à l'opérateur d'agir dans tous les espaces de noms.
2. L' **étendue de l'espace de noms** signifie que l'opérateur se verra attribuer des autorisations à l'aide de `Role` et `RoleBinding`, qui n'accorderont des autorisations que dans l'espace de noms donné. En outre, il y aura un `ClusterRole` et un `ClusterRoleBinding` pour les autorisations en lecture seule suivantes :

- 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`**
  - Instructions : **`get, list, watch`**
- Droit de lecture des classes de stockage sur le cluster. Cela permet à l'opérateur de tenter de faire remonter les problèmes liés aux classes de stockage sélectionnées dans les conteneurs.
  - Groupes d'API : **`storage.k8s.io`**
  - Ressources : **`storageclasses`**
  - Instructions : **`get, list`**

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 :
  - Pour une image " IBM MQ Operator de la version 3.4.0 ou ultérieure, ou une image de conteneur de gestionnaire de file d'attente " IBM MQ de la version 9.4.1.0-r2 ou ultérieure, voir Vérification des signatures d'image dans la documentation " IBM Cloud Pak for Integration (CP4I) 16.1.1
  - Pour une image " IBM MQ Operator à 3.2.x ou 3.3.x, ou une image de conteneur de gestionnaire de file d'attente " IBM MQ entre les versions 9.4.0.0-r1 et 9.4.1.0-r1, 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 67.
- 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 27.

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 3.6 ou version ultérieure, vous devez installer OpenShift Container Platform 4.12 ou version ultérieure. Pour plus d'informations, voir IBM Cloud Pak et Red Hat OpenShift Container Platform compatibilité.

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 39. 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 (air gap), vous devez dupliquer les images d' IBM MQ .

   Important : À partir de la version 3.5.0, l'adresse IBM MQ CASE a changé et ne contient plus qu'une seule version du gestionnaire de file d'attente. Le processus de mise en miroir des images est beaucoup plus rapide, mais il peut être nécessaire de l'exécuter plusieurs fois, une fois pour chaque version du gestionnaire de file d'attente en cours d'installation.

   Pour plus d'informations, voir «IBM MQ CASE pour l'installation et la mise à jour de l'air-gap sur 'Red Hat OpenShift», à la page 303.

   Vous devez rechercher les versions d' IBM, MQ et CASE qui contiennent les versions de l'opérateur et du gestionnaire de file d'attente dont vous avez besoin, et répéter le processus de mise en miroir de l'air gap pour chacune d'entre elles. Consultez les tables de recherche IBM MQ CASE pour connaître les mappages de versions.

   Par exemple, lorsque vous suivez les instructions liées ci-dessous pour mettre en miroir des images, vous installez la dernière version de l'opérateur et de la version correspondante du gestionnaire de file d'attente en utilisant les valeurs suivantes :

   ```
   export OPERATOR_PACKAGE_NAME=ibm-mq
   export OPERATOR_VERSION=3.6.0
   ```

   Pour plus d'informations sur le contenu du CASE IBM MQ, qui est utilisé pour les installations et les mises à niveau de l'intervalle aérien, voir " «IBM MQ CASE pour l'installation et la mise à jour de l'air-gap sur 'Red Hat OpenShift», à la page 303.

   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 61.

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 100.

Tâches associées

«Désinstallation de IBM MQ Operator», à la page 86
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 59.

  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=CHOSEN_OPERATOR_VERSION
     export ARCH=ARCHITECTURE
     ```

Où

– *ARCHITECTURE* est l'architecture du système sur lequel vous déployez le IBM MQ Operator, et a une valeur de `amd64`, `ppc64le`, ou `s390x`.
– *CHOSEN_OPERATOR_VERSION* est la version requise de l'IBM MQ Operator La version actuelle CD est 3.6.0. La version actuelle SC2 est 3.2.13.

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 63 "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 59.

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 59.

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 35 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 59).

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 59.

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. Consultez la liste des opérateurs disponibles pour le cluster à partir du 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 35 pour déterminer le canal d'opérateur à sélectionner.
7. Installez l'opérateur.

   Utilisez la commande suivante, en modifiant les valeurs suivantes :

   - Changez *ibm_mq_operator_channel* pour qu'il corresponde au canal de la version du IBM MQ Operator que vous voulez installer.
   - Remplacez *namespace_name* par **openshift-operators** si vous utilisez le mode "AllNamespaces", ou par l'espace de noms dans lequel vous souhaitez déployer le IBM MQ Operator si vous utilisez le mode "SingleNamespace".
   - Remplacer *catalog_source_name* par le nom de la source de catalogue utilisée pour installer le IBM MQ Operator, tel qu'ajouté dans l'étape d'installation précédente «Ajout de la source de catalogue IBM MQ Operator», à la page 61

```
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: catalog_source_name
  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 *namespace_name* est **openshift-operators** si vous utilisez le mode "AllNamespaces", ou le nom du projet (espace de noms) 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 59).

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 59.
- 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 27.

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 3.6 ou version ultérieure, vous devez installer OpenShift Container Platform 4.12 ou version ultérieure. Pour plus d'informations, voir IBM Cloud Pak et Red Hat OpenShift Container Platform compatibilité.

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 39. 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 .

     Important : À partir de la version 3.5.0, l'adresse IBM MQ CASE a changé et ne contient plus qu'une seule version du gestionnaire de file d'attente. Le processus de mise en miroir des images est beaucoup plus rapide, mais il peut être nécessaire de l'exécuter plusieurs fois, une fois pour chaque version du gestionnaire de file d'attente en cours d'installation.

     Pour plus d'informations, voir «IBM MQ CASE pour l'installation et la mise à jour de l'air-gap sur 'Red Hat OpenShift», à la page 303.

     Vous devez rechercher les versions d' IBM, MQ et CASE qui contiennent les versions de l'opérateur et du gestionnaire de file d'attente dont vous avez besoin, et répéter le processus de miroir d'air-gap pour chacune d'entre elles. Consultez les tables de recherche IBM MQ CASE pour connaître les mappages de versions.

     Par exemple, lorsque vous suivez les instructions liées ci-dessous pour mettre en miroir des images, vous installez la dernière version de l'opérateur et la version unique correspondante du gestionnaire de file d'attente en utilisant les valeurs suivantes :

     ```
     export OPERATOR_PACKAGE_NAME=ibm-mq
     export OPERATOR_VERSION=3.6.0
     ```

     Pour plus d'informations sur le contenu du CASE IBM MQ, qui est utilisé pour les installations et les mises à niveau de l'intervalle aérien, voir " «IBM MQ CASE pour l'installation et la mise à jour de l'air-gap sur 'Red Hat OpenShift», à la page 303.

- 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, en plus de l' 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=CHOSEN_OPERATOR_VERSION
   export ARCH=ARCHITECTURE
   ```

   Où

   - *ARCHITECTURE* est l'architecture du système sur lequel vous déployez le IBM MQ Operator, et a une valeur de `amd64`, `ppc64le`, ou `s390x`.
   - *CHOSEN_OPERATOR_VERSION* est la version requise de l'IBM MQ Operator La version actuelle CD est 3.6.0. La version actuelle SC2 est 3.2.13.

   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 100.

Tâches associées

«Désinstallation de IBM MQ Operator», à la page 86
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 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 72.
- 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 71.
- 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 81.

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 81.

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 3.6.0 du IBM MQ Operator est la dernière version CD
- La version 3.2.13 du IBM MQ Operator est la dernière version SC2

Procédure

- Option 1 :«Mise à niveau vers une édition de sécurité la plus récente du canal IBM MQ Operator», à la page 72
- Option 2 :«Mise à jour vers le canal 3.2.x SC2 de l'application IBM MQ Operator», à la page 74

- Option 3 :«Upgrading to the current CD channel of the IBM MQ Operator», à la page 76

OpenShift CP4I *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 71.

Pourquoi et quand exécuter cette tâche

Il existe différentes procédures, en fonction de la source de catalogue utilisée pour déployer l'IBM MQ Operator qui est mis à niveau.

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

Une nouvelle version 'IBM MQ Operator devient disponible dans un cluster OpenShift **seulement** après que vous ayez mis à jour la source du 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 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 80.

Procédure

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

  a) Images en miroir (entrefer uniquement).

  Si votre cluster se trouve dans un environnement réseau restreint (air gap), vous devez dupliquer les images d' IBM MQ .

  Important : À partir de la version 3.5.0, l'adresse IBM MQ CASE a changé et ne contient plus qu'une seule version du gestionnaire de file d'attente. Le processus de mise en miroir des images est beaucoup plus rapide, mais il peut être nécessaire de l'exécuter plusieurs fois, une fois pour chaque version du gestionnaire de file d'attente en cours d'installation.

  Pour plus d'informations, voir «IBM MQ CASE pour l'installation et la mise à jour de l'air-gap sur 'Red Hat OpenShift», à la page 303.

  Vous devez rechercher les versions d' IBM, MQ et CASE qui contiennent les versions de l'opérateur et du gestionnaire de file d'attente dont vous avez besoin, et répéter le processus de mise en

miroir de l'air gap pour chacune d'entre elles. Consultez les tables de recherche IBM MQ CASE pour connaître les mappages de versions.

Par exemple, lorsque vous suivez les instructions liées ci-dessous pour mettre en miroir des images, vous installez la dernière version de l'opérateur et la version unique correspondante du gestionnaire de file d'attente en utilisant les valeurs suivantes :

```
export OPERATOR_PACKAGE_NAME=ibm-mq
export OPERATOR_VERSION=3.6.0
```

Pour plus d'informations sur le contenu du CASE IBM MQ, qui est utilisé pour les installations et les mises à niveau de l'intervalle aérien, voir " «IBM MQ CASE pour l'installation et la mise à jour de l'air-gap sur 'Red Hat OpenShift», à la page 303.

Pour créer des images miroir, voir Mise en miroir d'images pour un cluster air-gapped.

b) 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.

c) Mettez à niveau IBM MQ Operator.

Si le statut d'**approbation de la mise à jour** pour le IBM MQ Operator est défini sur **Automatique**, votre opérateur est mis à niveau. Sinon, mettez à jour manuellement 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.

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

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

- **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. Sinon, approuvez manuellement votre IBM MQ Operator mise à niveau :

– 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 section Mise à niveau IBM MQ des gestionnaires de file d'attente.

- **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 dernière version du gestionnaire de files d'attente IBM MQ pour chaque canal d'opérateur actif. En utilisant la version correspondante, suivez la procédure décrite dans «Mise à niveau d'un gestionnaire de files d'attente IBM MQ avec Red Hat OpenShift», à la page 85.

Canal opérateur	Dernier gestionnaire de files d'attente IBM MQ
3.2SC2)	9.4.0.11-r3
3.6CD)	9.4.3.0-r1

OpenShift CP4I *Mise à jour vers le canal 3.2.x SC2 de l'application IBM MQ Operator*

Mettre à jour les déploiements du IBM MQ Operator où **seules les** IBM MQ licences 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 IntegrationCP4I), ou si vous avez déployé au moins un de vos gestionnaires de file d'attente en utilisant une licence CP4I, voir «Mise à niveau du IBM MQ Operator pour les utilisateurs CP4I», à la page 81.
- This task is for upgrading deployments of the IBM MQ Operator to the Support Cycle 2 (SC2) channel of IBM MQ Operator 3.2.x **seulement**. Notez qu'il s'agit également d'une étape intermédiaire potentielle lors de la mise à niveau vers la dernière CD version (3.6.0). Si cela ne s'applique pas à votre déploiement, consultez les autres voies de mise à niveau décrites dans «Mise à niveau de IBM MQ Operator et des gestionnaires de files d'attente», à la page 71.

Pourquoi et quand exécuter cette tâche

Complétez l'une des options suivantes en fonction du déploiement du IBM MQ Operator que vous mettez à niveau vers le canal 3.2.x SC2

Procédure

- **Option 1**: Mise à niveau des déploiements de la version 2.0.x Long Term SupportLTS) IBM MQ Operator.

  Suivez les étapes de la section «Mise à niveau d'un 2.0.x LTS IBM MQ Operator vers le 3.2.x SC2 canal», à la page 74.
- **Option 2**: Mise à niveau d'un déploiement CD du IBM MQ Operator.

  Suivez les étapes de la section «Mise à jour d'un CD IBM MQ Operator vers le canal v3.2.x SC2 », à la page 75.

OpenShift V 9.4.0 CP4I-SC2 *Mise à niveau d'un 2.0.x LTS IBM MQ Operator vers le 3.2.x SC2 canal*

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 81.
- 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 71.

Pour effectuer une mise à niveau vers IBM MQ Operator 3.2.13 , 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 36. Pour mettre à niveau la plateforme, voir Mise à niveau d' Red Hat OpenShift.

Procédure

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

   Si votre cluster se trouve dans un environnement réseau restreint (air gap), vous devez dupliquer les images d' IBM MQ .

   Important : À partir de la version 3.5.0, l'adresse IBM MQ CASE a changé et ne contient plus qu'une seule version du gestionnaire de file d'attente. Le processus de mise en miroir des images est beaucoup plus rapide, mais il peut être nécessaire de l'exécuter plusieurs fois, une fois pour chaque version du gestionnaire de file d'attente en cours d'installation.

   Pour plus d'informations, voir «IBM MQ CASE pour l'installation et la mise à jour de l'air-gap sur 'Red Hat OpenShift», à la page 303.

   Vous devez rechercher les versions d' IBM, MQ et CASE qui contiennent les versions de l'opérateur et du gestionnaire de file d'attente dont vous avez besoin, et répéter le processus de miroir d'air-gap pour chacune d'entre elles. Consultez les tables de recherche IBM MQ CASE pour connaître les mappages de versions.

   Par exemple, lorsque vous suivez les instructions liées ci-dessous pour mettre en miroir des images, vous installez la dernière version de l'opérateur et la version unique correspondante du gestionnaire de file d'attente en utilisant les valeurs suivantes :

   ```
   export OPERATOR_PACKAGE_NAME=ibm-mq
   export OPERATOR_VERSION=3.6.0
   ```

   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 3.2.13.**

   Voir «Mise à niveau du IBM MQ Operator à l'aide de Red Hat OpenShift», à la page 83.
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' SC2 (9.4.0.11). Voir «Mise à niveau d'un gestionnaire de files d'attente IBM MQ avec Red Hat OpenShift», à la page 85.

OpenShift CP4I V 9.4.0 *Mise à jour d'un CD IBM MQ Operator vers le canal v3.2.x SC2*

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 81.
- Cette tâche est destinée à mettre à niveau les déploiements de " Continuous Delivery (CD) de " IBM MQ Operator antérieurs à la version 3.2.0, vers la version " 3.2.13 **uniquement.** Notez qu'il s'agit également d'une étape intermédiaire potentielle lors de la mise à niveau vers la dernière version " CD " (v3.6.0). Si cela ne s'applique pas à votre déploiement, consultez les chemins de mise à niveau alternatifs décrits dans le " «Mise à niveau de IBM MQ Operator et des gestionnaires de files d'attente», à la page 71.

Pour effectuer une mise à niveau vers IBM MQ Operator 3.2.13 , 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 36. 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 3.2.13.

2. **Mise à jour vers l'opérateur le plus récent de votre chaîne actuelle**

   Suivez les étapes de la section «Mise à niveau vers une édition de sécurité la plus récente du canal IBM MQ Operator», à la page 72.

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

   Si votre cluster se trouve dans un environnement réseau restreint (air gap), vous devez dupliquer les images d' IBM MQ .

   Important : À partir de la version 3.5.0, l'adresse IBM MQ CASE a changé et ne contient plus qu'une seule version du gestionnaire de file d'attente. Le processus de mise en miroir des images est beaucoup plus rapide, mais il peut être nécessaire de l'exécuter plusieurs fois, une fois pour chaque version du gestionnaire de file d'attente en cours d'installation.

   Pour plus d'informations, voir «IBM MQ CASE pour l'installation et la mise à jour de l'air-gap sur 'Red Hat OpenShift», à la page 303.

   Vous devez rechercher les versions d' IBM, MQ et CASE qui contiennent les versions de l'opérateur et du gestionnaire de file d'attente dont vous avez besoin, et répéter le processus de mise en miroir de l'air gap pour chacune d'entre elles. Consultez les tables de recherche IBM MQ CASE pour connaître les mappages de versions.

   Par exemple, lorsque vous suivez les instructions liées ci-dessous pour mettre en miroir des images, vous installez la dernière version de l'opérateur et la version correspondante du gestionnaire de file d'attente en utilisant les valeurs suivantes :

   ```
   export OPERATOR_PACKAGE_NAME=ibm-mq
   export OPERATOR_VERSION=3.6.0
   ```

   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.

4. **Mettez à niveau votre IBM MQ Operator vers 3.2.13.**

   Voir «Mise à niveau du IBM MQ Operator à l'aide de Red Hat OpenShift», à la page 83.

5. **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' SC2 (9.4.0.11). Voir «Mise à niveau d'un gestionnaire de files d'attente IBM MQ avec Red Hat OpenShift», à la page 85.

OpenShift *Upgrading to the current CD channel of the IBM MQ Operator*

Mettre à jour 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 IntegrationCP4I), ou si vous avez déployé au moins un de vos gestionnaires de file d'attente en utilisant une licence CP4I, voir «Mise à niveau du IBM MQ Operator pour les utilisateurs CP4I», à la page 81.
- Ce sujet concerne **uniquement** la mise à niveau vers le canal Continuous DeliveryCD) du canal IBM MQ Operator (3.6). Si cela ne s'applique pas à votre déploiement, consultez les autres voies de mise à niveau décrites dans «Mise à niveau de IBM MQ Operator et des gestionnaires de files d'attente», à la page 71.

Pourquoi et quand exécuter cette tâche

Complétez l'une des options suivantes, en fonction du déploiement du " IBM MQ Operator que vous mettez à niveau vers le canal CD " 3.6 " :

Procédure

- **Option 1 :** mise à niveau des déploiements de la 3.2.x " SC2 " IBM MQ Operator.

  Suivez les étapes de la section «Mise à niveau d'un " SC2 " IBM MQ Operator 3.2.x vers le canal " CD " actuel», à la page 77.
- **Option 2 :** mise à niveau d'un déploiement " CD du " IBM MQ Operator.

  Suivez les étapes de la section «Mise à niveau d'un canal " CD " IBM MQ Operator vers le canal " CD " actuel», à la page 78.

OpenShift *Mise à niveau d'un " SC2 " IBM MQ Operator 3.2.x vers le canal " CD " actuel*

Mettre à jour un déploiement 3.2.x 'SC2 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 **seulement** " 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 en utilisant une licence " CP4I, voir " «Mise à niveau du IBM MQ Operator pour les utilisateurs CP4I», à la page 81.
- Cette tâche consiste à mettre à niveau une version 3.2.x " SC2 du " IBM MQ Operator vers le canal actuel " Continuous Delivery (CD) du " IBM MQ Operator (v3.6) **uniquement.** Si cela ne s'applique pas à votre déploiement, consultez le chemin de mise à niveau alternatif décrit dans " «Mise à niveau de IBM MQ Operator et des gestionnaires de files d'attente», à la page 71.

Pour passer à 'IBM MQ Operator '3.6.0, vous devez utiliser 'Red Hat OpenShift Container Platform '4.12 ' ou une version ultérieure. Pour vérifier les versions compatibles pour chaque canal " IBM MQ Operator, voir " «Versions Red Hat OpenShift Container Platform compatibles», à la page 36. Pour mettre à niveau la plate-forme, voir 'Mise à niveau de " Red Hat OpenShift dans la documentation 'IBM Cloud Pak for Integration

Procédure

1. Facultatif : **Mettre à jour un " IBM MQ Operator qui est actuellement dans une version antérieure à 3.2.0**

   Si votre " IBM MQ Operator est actuellement dans une version antérieure à 3.2.0, suivez les étapes appropriées dans le " «Mise à jour vers le canal 3.2.x SC2 de l'application IBM MQ Operator», à la page 74, puis revenez ici pour passer à la dernière version du " CD. Notez que si vous utilisez déjà une

version v3.2.x du " IBM MQ Operator, vous pouvez sauter cette étape. Cependant, la mise à niveau vers le canal v3.2 du " IBM MQ Operator " est une étape préalable obligatoire avant la mise à niveau vers la version " 3.6.0.

2. **Mise à jour vers l'opérateur le plus récent de votre chaîne actuelle**

   Suivez les étapes de la section «Mise à niveau vers une édition de sécurité la plus récente du canal IBM MQ Operator», à la page 72.

3. **Images en miroir (entrefer uniquement).**

   Si votre cluster se trouve dans un environnement réseau restreint (air gap), vous devez dupliquer les images d' IBM MQ .

   Important : À partir de la version 3.5.0, l'adresse IBM MQ CASE a changé et ne contient plus qu'une seule version du gestionnaire de file d'attente. Le processus de mise en miroir des images est beaucoup plus rapide, mais il peut être nécessaire de l'exécuter plusieurs fois, une fois pour chaque version du gestionnaire de file d'attente en cours d'installation.

   Pour plus d'informations, voir «IBM MQ CASE pour l'installation et la mise à jour de l'air-gap sur 'Red Hat OpenShift», à la page 303.

   Vous devez rechercher les versions d' IBM, MQ et CASE qui contiennent les versions de l'opérateur et du gestionnaire de file d'attente dont vous avez besoin, et répéter le processus de mise en miroir de l'air gap pour chacune d'entre elles. Consultez les tables de recherche IBM MQ CASE pour connaître les mappages de versions.

   Par exemple, lorsque vous suivez les instructions liées ci-dessous pour mettre en miroir des images, vous installez la dernière version de l'opérateur et la version correspondante du gestionnaire de file d'attente en utilisant les valeurs suivantes :

   ```
   export OPERATOR_PACKAGE_NAME=ibm-mq
   export OPERATOR_VERSION=3.6.0
   ```

   Vous devriez omettre la section 3.5 "Configurer le cluster", car la connexion au registre d'images devrait avoir été établie lors d'installations ou de mises à jour antérieures.

   Pour plus d'informations sur le contenu du CASE IBM MQ, qui est utilisé pour les installations et les mises à niveau de l'intervalle aérien, voir " «IBM MQ CASE pour l'installation et la mise à jour de l'air-gap sur 'Red Hat OpenShift», à la page 303.

   Lien : Miroir d'images pour un cluster à air comprimé.

4. **Passez du " IBM MQ Operator au " 3.6.0.**

   Suivez les instructions présentées dans «Mise à niveau du IBM MQ Operator à l'aide de Red Hat OpenShift», à la page 83.

5. **Mettre à niveau les instances.**

   Pour bénéficier des dernières fonctionnalités et des correctifs de sécurité, mettez à niveau l'opérande IBM MQ (image du conteneur Queue Manager) vers la dernière version CD (9.4.3.0-r1). Voir '«Mise à niveau d'un gestionnaire de files d'attente IBM MQ avec Red Hat OpenShift», à la page 85.

OpenShift *Mise à niveau d'un canal " CD " IBM MQ Operator vers le canal " CD " actuel*

Mettre à niveau un déploiement " CD 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 **seulement** " 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 en utilisant une licence " CP4I, voir " «Mise à niveau du IBM MQ Operator pour les utilisateurs CP4I», à la page 81.

- Cette tâche consiste à mettre à niveau une version " Continuous Delivery (CD) du " IBM MQ Operator vers le canal " CD actuel du " IBM MQ Operator (v3.6) **uniquement.** Si cela ne s'applique pas à votre déploiement, consultez le chemin de mise à niveau alternatif décrit dans " «Mise à niveau de IBM MQ Operator et des gestionnaires de files d'attente», à la page 71.

Pour passer à 'IBM MQ Operator '3.6.0, vous devez utiliser 'Red Hat OpenShift Container Platform '4.12 ' ou une version ultérieure. Pour vérifier les versions compatibles pour chaque canal " IBM MQ Operator, voir " «Versions Red Hat OpenShift Container Platform compatibles», à la page 36. Pour mettre à niveau la plate-forme, voir 'Mise à niveau de " Red Hat OpenShift dans la documentation 'IBM Cloud Pak for Integration

Procédure

1. Facultatif : **Mettre à jour un " IBM MQ Operator qui est actuellement dans une version antérieure à 3.2.0**

   Si votre " IBM MQ Operator est actuellement dans une version antérieure à 3.2.0, suivez les étapes appropriées dans " «Mise à jour vers le canal 3.2.x SC2 de l'application IBM MQ Operator», à la page 74, puis revenez ici pour suivre l'étape " «2», à la page 79 afin de passer à la dernière version " CD Notez que si vous utilisez déjà une version " 3.2.x du " IBM MQ Operator ", vous pouvez sauter cette étape. Cependant, la mise à niveau vers le canal v3.2 du " IBM MQ Operator est une étape préalable obligatoire avant la mise à niveau vers la version " 3.6.0.
2. Facultatif : **Mise à niveau d'un " IBM MQ Operator qui est actuellement à la version " 3.2.x.**

   Si votre " IBM MQ Operator est actuellement à la version " 3.2.x, suivez les étapes pertinentes du " «Mise à niveau d'un " SC2 " IBM MQ Operator 3.2.x vers le canal " CD " actuel», à la page 77.
3. **Mettre à niveau une version d' IBM MQ Operator , actuellement à la version 3.3.0 ou supérieure.**

   Si votre version d' IBM MQ Operator s est actuellement la version 3.3.0 ou une version ultérieure, suivez les étapes ci-dessous pour passer à la version IBM MQ Operator 3.6.0.

   a) **Mise à jour vers l'opérateur le plus récent de votre chaîne actuelle**

   Suivez les étapes de la section «Mise à niveau vers une édition de sécurité la plus récente du canal IBM MQ Operator», à la page 72.

   b) **Passez au canal opérateur suivant.**

   i) Images miroir (espace d'air uniquement).

   Si votre cluster se trouve dans un environnement réseau restreint (air gap), vous devez dupliquer les images d' IBM MQ . Complétez les étapes du lien suivant, en utilisant uniquement ces valeurs :

   ```
   export OPERATOR_PACKAGE_NAME=ibm-mq
   export OPERATOR_VERSION=[operator_version]
   ```

   où *[operator_version]* est la version de l' IBM MQ Operator . Pour connaître les dernières informations (IBM MQ Operator) sur le canal vers lequel vous effectuez la mise à niveau, consultez le site «Historique des éditions de IBM MQ Operator», à la page 5.

   Vous devriez omettre la section 3.5 "Configurer le cluster", car la connexion au registre d'images devrait avoir été établie lors d'installations ou de mises à jour antérieures.

   Pour plus d'informations sur le contenu du CASE IBM MQ, qui est utilisé pour les installations et les mises à niveau de l'intervalle aérien, voir " «IBM MQ CASE pour l'installation et la mise à jour de l'air-gap sur 'Red Hat OpenShift», à la page 303.

   Lien : Miroir d'images pour un cluster à air comprimé.

   ii) Mettez à niveau votre système d' IBM MQ Operator.

   Suivez les instructions présentées dans «Mise à niveau du IBM MQ Operator à l'aide de Red Hat OpenShift», à la page 83.

   iii) Mettre à niveau les instances.

Pour recevoir les dernières fonctionnalités et correctifs de sécurité, mettez à jour l'opérande d' IBM MQ . Voir «Mise à niveau d'un gestionnaire de files d'attente IBM MQ avec Red Hat OpenShift», à la page 85.

c) Répétez les sous-étapes (a) et (b) précédentes jusqu'à ce que vous atteigniez le dernier canal d' CD s (v3.6.0).

OpenShift *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
   ```

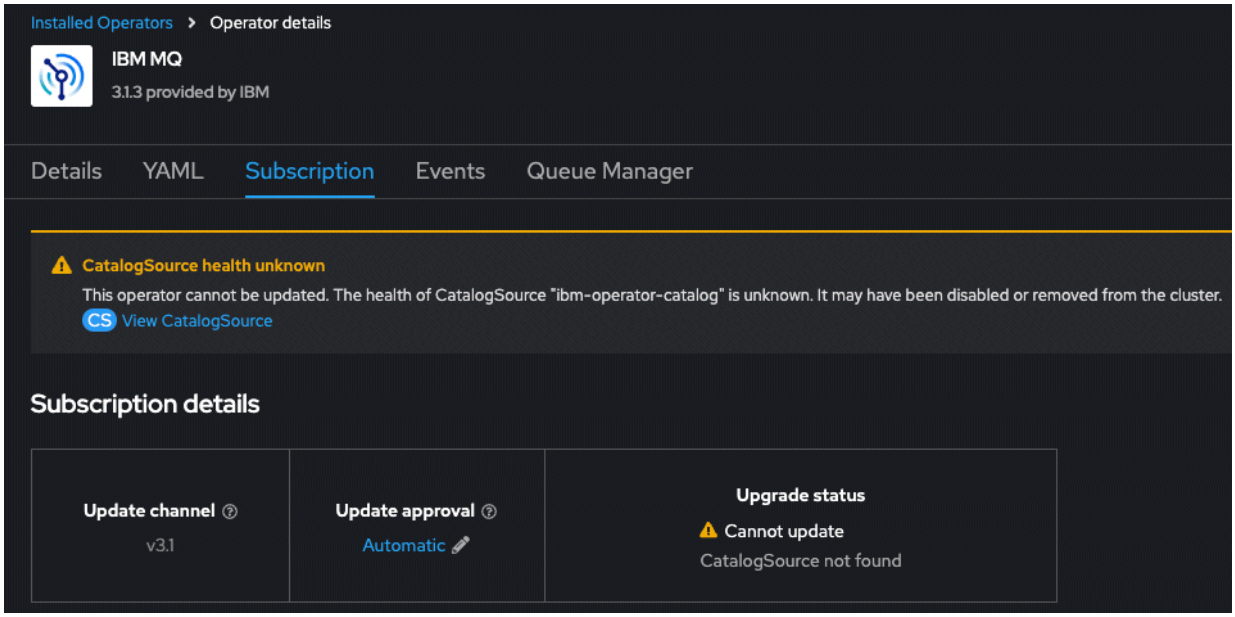
3. Facultatif : Supprimez la source de catalogue de l'opérateur IBM .

   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 80.

      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, l'état passe de "CatalogSource introuvable" à "Réussi".

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 71.

Pourquoi et quand exécuter cette tâche

Si vous passez à la version CD du IBM MQ Operator (version 3.6.0), vous devez d'abord passer au canal v3.2 SC2 comme décrit dans les instructions suivantes.

Si votre version d' IBM MQ Operator s est la version 3.3.0 ou supérieure, passez à l'étape (c) de l' `Upgrade to the CD Channel`.

Procédure

1. Facultatif : **Mise à jour vers le canal v3.2 SC2**

   Complétez une ou plusieurs des options suivantes pour mettre à niveau votre IBM MQ Operator vers le canal Support Cycle 2SC2) du IBM MQ Operator **uniquement :**

   - **Option 1 :** Mise à niveau des déploiements de la version 2.0.x Long Term SupportLTS) 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 de l'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 la solution IBM MQ Operator.

     Suivez les étapes de la migration vers le canal CD actuel de la IBM MQ Operator (IBM MQ 9.3 documentation), puis revenez ici et passez à l'**option 2.** Notez qu'il s'agit d'une étape prérequise obligatoire.
2. **Passer au canal CD (3.6).**

   a) Facultatif : **Mise à niveau vers le déploiement d' 3.2.x IBM MQ Operator**

   Suivez l'étape principale précédente pour passer au déploiement 3.2.x du " IBM MQ Operator.

   b) Facultatif : **Mettre à jour un déploiement 3.2.x du " IBM MQ Operator**

   Suivez les étapes de la section Mise à niveau à partir de la 16.1.0 en générant un plan de mise à niveau.

   c) Facultatif : **Mise à jour d'un déploiement 3.3.x de l'application IBM MQ Operator**

   Si votre IBM MQ Operator est actuellement à la version 3.3.x, suivez les étapes suivantes pour mettre à jour votre opérateur à 3.4.x avant de passer à la mise à jour suivante à 3.6.0.

   i) Images miroir (espace d'air uniquement).

   Si votre cluster se trouve dans un environnement réseau restreint (air gap), vous devez dupliquer les images d' IBM MQ . Effectuez les étapes du lien suivant, en utilisant uniquement ces valeurs :

   ```
   export OPERATOR_PACKAGE_NAME=ibm-mq
   export OPERATOR_VERSION=3.4.1
   ```

   Vous devez omettre la section « Configurer le cluster » (3.5), car la connexion au registre d'images aurait dû être configurée lors des installations ou mises à niveau précédentes.

   Pour plus d'informations sur le contenu de l' IBM ez MQ CASE, qui est utilisé pour les installations et les mises à niveau d'air-gap, consultez «IBM MQ CASE pour l'installation et la mise à jour de l'air-gap sur 'Red Hat OpenShift», à la page 303.

   Lien : images miroir pour un cluster isolé.

   ii) Mettez à niveau votre système d' IBM MQ Operator.

   Suivez les instructions présentées dans «Mise à niveau du IBM MQ Operator à l'aide de Red Hat OpenShift», à la page 83.

   iii) Mettre à niveau les instances.

Pour recevoir les dernières fonctionnalités et correctifs de sécurité, mettez à jour l'opérande d' IBM MQ . Voir «Mise à niveau d'un gestionnaire de files d'attente IBM MQ avec Red Hat OpenShift», à la page 85.

d) **Mettre à jour un déploiement de 3.4.0 ou supérieur de l'application IBM MQ Operator**

i) Images miroir (espace d'air uniquement).

Si votre cluster se trouve dans un environnement réseau restreint (air gap), vous devez dupliquer les images d' IBM MQ . Effectuez les étapes du lien suivant, en utilisant uniquement ces valeurs :

```
export OPERATOR_PACKAGE_NAME=ibm-mq
export OPERATOR_VERSION=[operator_version]
```

où *[operator_version]* est la version de l' IBM MQ Operator . Pour connaître les dernières informations (IBM MQ Operator) sur le canal vers lequel vous effectuez la mise à niveau, consultez le site «Historique des éditions de IBM MQ Operator», à la page 5.

Vous devez omettre la section « Configurer le cluster » (3.5), car la connexion au registre d'images aurait dû être configurée lors des installations ou mises à niveau précédentes.

Pour plus d'informations sur le contenu de l' IBM ez MQ CASE, qui est utilisé pour les installations et les mises à niveau d'air-gap, consultez «IBM MQ CASE pour l'installation et la mise à jour de l'air-gap sur 'Red Hat OpenShift», à la page 303.

Lien : images miroir pour un cluster isolé.

ii) Passez à l'opérateur le plus récent de votre chaîne actuelle.

Suivez les étapes de la section «Mise à niveau vers une édition de sécurité la plus récente du canal IBM MQ Operator», à la page 72.

iii) Mettez à niveau le déploiement IBM MQ Operator.

Suivez les étapes de la section Mise à niveau à partir de 16.1.1 en générant un plan de mise à niveau.

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

Le " IBM MQ Operator peut être mis à jour à partir de la console ou de la ligne de commande du " Red Hat OpenShift

Avant de commencer

Remarque :

- La dernière CD version du IBM MQ Operator est 3.6.0.
- La dernière SC2 version du IBM MQ Operator est 3.2.13.

Pour les dernières notes de version de IBM MQ Operator , voir Historique des versions de IBM MQ Operator.

Procédure

- **Option 1 : Mettre à jour le " IBM MQ Operator à l'aide de la console " Red Hat OpenShift.**

a) Connectez-vous à la console du cluster " Red Hat OpenShift

b) Images en miroir (entrefer uniquement).

Si votre cluster se trouve dans un environnement réseau restreint (air-gapped), vous devez mettre en miroir les images 'IBM MQ en utilisant les valeurs suivantes :

```
export OPERATOR_PACKAGE_NAME=ibm-mq
export OPERATOR_VERSION=3.6.0
```

Pour plus d'informations sur le contenu du CASE IBM MQ, qui est utilisé pour les installations et les mises à niveau de l'intervalle aérien, voir " «IBM MQ CASE pour l'installation et la mise à jour de l'air-gap sur 'Red Hat OpenShift», à la page 303.

Pour créer des images miroir, voir Mise en miroir d'images pour un cluster air-gapped.

c) Appliquer la dernière source du catalogue.

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

Pour passer de l'utilisation du catalogue de l'opérateur " IBM à l'utilisation de la source de catalogue spécifique pour le " IBM MQ Operator, et mieux contrôler les mises à niveau, reportez-vous aux étapes du " «Passage à la source de catalogue spécifique pour le IBM MQ Operator», à la page 80 avant de revenir à l'étape suivante.

Si vous utilisez le catalogue Opérateur 'IBM (certaines installations en ligne uniquement), passez à l'étape suivante.

Sinon, suivez les instructions du " «Ajout de la source de catalogue IBM MQ Operator», à la page 61 et passez à l'étape suivante.

d) 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 jour 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 **Abonnement** .

d. Cliquez sur **Canal**.

La fenêtre **Change Subscription Update Channel** s'ouvre.

e. Sélectionnez le canal souhaité, puis cliquez sur **Enregistrer.**

L'opérateur passe à la dernière version disponible pour le nouveau canal. Voir «Versions prises en charge pour IBM MQ Operator», à la page 35.

- **Option 2 : Mettre à jour le 'IBM MQ Operator en utilisant la ligne de commande 'Red Hat OpenShift**

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

b) Images en miroir (entrefer uniquement).

Si votre cluster se trouve dans un environnement réseau restreint (air-gapped), vous devez mettre en miroir les images 'IBM MQ en utilisant les valeurs suivantes :

```
export OPERATOR_PACKAGE_NAME=ibm-mq
export OPERATOR_VERSION=3.6.0
```

Pour plus d'informations sur le contenu du CASE IBM MQ, qui est utilisé pour les installations et les mises à niveau de l'intervalle aérien, voir " «IBM MQ CASE pour l'installation et la mise à jour de l'air-gap sur 'Red Hat OpenShift», à la page 303.

Pour créer des images miroir, voir Mise en miroir d'images pour un cluster air-gapped.

c) Appliquer la dernière source du catalogue.

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

Pour passer de l'utilisation du catalogue de l'opérateur " IBM à l'utilisation de la source de catalogue spécifique pour le " IBM MQ Operator, et mieux contrôler les mises à niveau, reportez-

vous aux étapes du " «Passage à la source de catalogue spécifique pour le IBM MQ Operator», à la page 80 avant de revenir à l'étape suivante.

Si vous utilisez le catalogue Opérateur 'IBM (certaines installations en ligne uniquement), passez à l'étape suivante.

Sinon, suivez les instructions du " «Ajout de la source de catalogue IBM MQ Operator», à la page 61 et passez à l'étape suivante.

d) Mettez à niveau IBM MQ Operator. Les nouvelles versions principales ou mineures de IBM MQ Operator sont livrées via de nouveaux canaux d'abonnement. Pour mettre à jour votre opérateur vers une nouvelle version majeure ou mineure, suivez les étapes suivantes pour 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*

Un gestionnaire de file 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'une des interfaces suivantes : Operator Hub ; Ligne de commande ; IBM Cloud Pak for Integration Platform UI.

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 .

Remarque :

- La dernière CD version du IBM MQ gestionnaire de file d'attente est 9.4.3.0-r1.
- La dernière SC2 version du IBM MQ gestionnaire de file d'attente est 9.4.0.11-r3.

Pour les dernières IBM MQ notes de mise à jour du gestionnaire de file d'attente, voir «Historique des versions des images des conteneurs du gestionnaire de file d'attente», à la page 18.

Assurez-vous que votre IBM MQ Operator est à la version requise. Pour mettre à jour votre IBM MQ Operator, voir «Mise à niveau du IBM MQ Operator», à la page 71.

Avant de pouvoir mettre à niveau le gestionnaire de file d'attente dans un environnement air-gap, vous devez mettre en miroir les dernières images IBM Cloud Pak for Integration en suivant les étapes spécifiques à l'air-gap qui sont effectuées lors d'une mise à niveau de l'IBM MQ Operator. Pour un exemple de ces étapes, voir «Upgrading to the current CD channel of the IBM MQ Operator», à la page 76.

Pourquoi et quand exécuter cette tâche

Cette tâche propose trois options pour mettre à niveau un gestionnaire de files d'attente IBM MQ déployé sur OpenShift:

1. Mise à jour d'un IBM MQ gestionnaire de file d'attente à l'aide de la Red Hat OpenShift console.
2. Mise à niveau d'un IBM MQ gestionnaire de files d'attente à l'aide de la Red Hat OpenShift CLI
3. Mise à jour d'un IBM MQ gestionnaire de file d'attente dans Red Hat OpenShift à l'aide de l'interface utilisateur de la plate-forme

Procédure

- **Option 1**: Mise à niveau d'un IBM MQ gestionnaire de files d'attente à l'aide de la Red Hat OpenShift console.
  a) Connectez-vous à votre console Web de cluster Red Hat OpenShift.
  b) Depuis le panneau de navigation, cliquez sur **Operators** > **Installed Operators**.
     Tous les opérateurs installés dans le projet spécifié sont affichés.
  c) Sélectionnez **IBM MQ Operator**.
     La fenêtre **IBM MQ Operator** s'affiche.
  d) Accédez à l'onglet **Queue Manager** .
     La fenêtre **Queue Manager Details** s'affiche.
  e) Sélectionnez le gestionnaire de files d'attente à mettre à niveau.
  f) Accédez à l'onglet YAML.
  g) 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 versions des images des conteneurs du gestionnaire de file d'attente», à la page 18 pour une correspondance entre IBM MQ Operator les versions et IBM MQ les images du conteneur du gestionnaire de file d'attente.
  h) Sauvegardez le gestionnaire de files d'attente mis à jour YAML.
- **Option 2**: Mise à niveau d'un IBM MQ gestionnaire de files d'attente à l'aide de la Red Hat OpenShift CLI
  a) Connectez-vous à l'interface de ligne de commande (CLI Red Hat OpenShift en utilisant `oc login`.
  b) 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 35 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.
- **Option 3**: Mise à jour d'un IBM MQ gestionnaire de file d'attente en Red Hat OpenShift utilisant l'interface utilisateur de la plate-forme.

  Suivez les instructions de la section Gérer les versions et les mises à jour à l'aide de l'interface utilisateur de la plateforme dans la documentation IBM Cloud Pak for Integration

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 .

Kubernetes V 9.4.1 Préparation de l'utilisation de 'Kubernetes par la création d'un secret d'extraction

L'image préconstruite " IBM MQ Advanced peut être extraite de l'image " IBM Container Registry, mais le registre vérifie les droits de licence. Ce contrôle requiert une clé d'autorisation qui est stockée dans un secret d'extraction `docker-registry`.

Avant de commencer

Installez l'outil de ligne de commande " **kubectl** et connectez-vous à votre cluster " Kubernetes

Procédure

1. Obtenez la clé de droit qui est attribuée à votre ID " IBM

a) Connectez-vous à Mon IBM - Bibliothèque des logiciels de conteneur avec l'ID et le mot de passe IBM associés au logiciel autorisé.

b) Dans la section **Clé d'autorisation**, cliquez sur **Copier la clé** pour copier la clé d'autorisation dans le presse-papiers.

2. Créer le secret de la clé de droit.

   a. À partir de l'interface de gestion " OpenShift, exécutez la commande suivante pour créer un secret d'extraction d'image appelé " `ibm-entitlement-key`.

```
kubectl 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 de droit que vous avez copiée à l'étape b, *user_email* est l'ID 'IBM associé au logiciel ayant droit, et *namespace* est l'espace de noms que vous prévoyez d'utiliser pour les gestionnaires de file d'attente.

Kubernetes Préparer l'utilisation de Kubernetes en installant le IBM License Service

La licence de conteneur vous permet d'obtenir une licence uniquement pour la capacité disponible de vos conteneurs IBM MQ individuels, plutôt que d'exiger une licence pour l'ensemble du serveur sur lequel vos conteneurs s'exécutent. Pour tirer parti des licences de conteneurs, le site IBM License Service doit être utilisé pour suivre l'utilisation des licences et déterminer les droits dont vous avez besoin.

Procédure

Installez le site IBM License Service en suivant les instructions de la section Installation de License Service sur le cluster Kubernetes dans la documentation IBM Cloud Pak foundational services .

Linux V 9.4.1 Préparation à l'utilisation de 'Podman ou 'Docker en tirant l'image du conteneur préconstruit

L'image préconstruite " IBM MQ Advanced peut être extraite du " IBM Container Registry à l'aide d'outils de ligne de commande, tels que " Podman ou " Docker.

Avant de commencer

Installer " Podman ou " Docker.

Remarque : Les exemples de commandes suivants utilisent le " Podman, mais vous pouvez utiliser le " Podman ou le " Docker avec les mêmes paramètres de commande.

Procédure

1. Obtenir la clé de droit attribuée à votre IBM ID.

   a) Connectez-vous à Mon IBM - Bibliothèque des logiciels de conteneur avec l'ID et le mot de passe IBM associés au logiciel autorisé.

   b) Dans la section **Clé d'autorisation**, cliquez sur **Copier la clé** pour copier la clé d'autorisation dans le presse-papiers.

2. Se connecter au registre des conteneurs

   Par exemple, en utilisant " Podman:

```
podman login --username cp --password ... cp.icr.io
```

   Où "..." est remplacée par la clé de droit que vous avez copiée à l'étape précédente.

3. Tirer l'image

Par exemple, en utilisant " Podman:

```
podman pull cp.icr.io/cp/ibm-mqadvanced-server:9.4.3.0-r1
```

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 28

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 90
- «Génération d'un exemple d'image de conteneur de gestionnaire de files d'attente IBM MQ», à la page 91
- «Exécution d'applications de liaison locale dans des conteneurs distincts», à la page 93
- 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 91
  - «Génération d'un exemple d'image de gestionnaire de files d'attente IBM MQ configurée», à la page 92

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 94 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 93 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, 2025.
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. Kubernetes 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)
```

MQ Adv. Kubernetes V 9.4.2 Création d'une configuration Native HA CRR si vous créez vos propres conteneurs

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

Pourquoi et quand exécuter cette tâche

La méthode recommandée pour créer une solution Native HA CRR consiste à utiliser l'opérateur « IBM MQ » (voir «Configuration de Native HA CRR à l'aide de l' IBM MQ Operator», à la page 118). Si vous créez vos propres conteneurs, vous pouvez également suivre ces instructions.

Pour créer une configuration Native HA CRR, vous devez créer deux groupes Native HA pour un gestionnaire de file d'attente, un sur le site « actif » et un sur le site « de récupération ». Vous éditez ensuite le fichier d' `qm.ini` s pour que le gestionnaire de file d'attente configure la réplication inter-régions entre eux.

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

Procédure

1. Sur votre site Live, créez un groupe Native HA, voir «Création du groupe Native HA si vous créez vos propres conteneurs», à la page 96.
2. Sur votre site de récupération, créez un deuxième groupe Native HA, voir «Création du groupe Native HA si vous créez vos propres conteneurs», à la page 96.
3. Spécifiez que les deux groupes sur les sites de production et de reprise travaillent ensemble dans une configuration Native HA CRR en apportant les modifications suivantes au fichier d' qm.ini s du gestionnaire de file d'attente.

   Sur le groupe de récupération, spécifiez les valeurs suivantes dans la strophe d' `NativeHALocalInstance` :

   `GroupName`
   Spécifiez un nom pour le groupe de récupération, par exemple « NHACRR_recovery ».

   `GroupRole`
   Préciser le rôle actuel du groupe. Dans ce cas, réglez sur « Récupération ».

   Spécifiez les valeurs suivantes dans la strophe d' `NativeHARecoveryGroup` :

   `GroupName`
   Le nom du groupe sur le site en ligne, par exemple, « NHACRR_live ».

ReplicationAddress

Spécifiez un nom d'hôte et un port qui adressent le port de trafic de réplication sur l'autre groupe dans la configuration de réplication inter-régions Native HA. L'adresse peut être une adresse de service unique et un port ou une liste séparée par des virgules.

Sur le groupe live, spécifiez les valeurs suivantes dans la strophe d' `NativeHALocalInstance` :

GroupName
Spécifiez un nom pour le groupe live, par exemple « NHACRR_live ».

GroupRole
Préciser le rôle actuel du groupe. Dans ce cas, réglez sur « Live ».

Spécifiez les valeurs suivantes dans la strophe d' `NativeHARecoveryGroup` :

GroupName
Le nom du groupe sur le site de récupération, par exemple « NHACRR_recovery ».

ReplicationAddress

Spécifiez un nom d'hôte et un port qui adressent le port de trafic de réplication sur l'autre groupe dans la configuration de réplication inter-régions Native HA. L'adresse peut être une adresse de service unique et un port ou une liste séparée par des virgules.

4. Vous devez sécuriser la connexion utilisée pour la réplication en utilisant TLS. Si vous avez spécifié la sécurité pour la réplication interne lors de la création des deux groupes, vous pouvez utiliser les mêmes paramètres pour sécuriser la réplication externe. Vous pouvez également spécifier les valeurs suivantes dans la strophe d' `NativeHALocalInstance` s pour le groupe de récupération et le groupe de production.

   GroupCipherSpec
   Remplace les spécifications de chiffrement spécifiées par `CipherSpec`.

   GroupCertificateLabel
   Remplace l'étiquette de certification spécifiée par `CertificateLabel`.

   Si vous souhaitez utiliser ces valeurs pour la réplication externe, tout en utilisant du texte brut pour la réplication interne, définissez ces deux valeurs pour les groupes de production et de récupération, tout en définissant `CipherSpec` sur NULL.

Exemple

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

```
NativeHALocalInstance:
  ...
  GroupName=NHACRR_live
  GroupRole=live

NativeHARecoveryGroup:
  GroupName=NHACRR_recovery
  ReplicationAddress=recovery-cluster.fyre.ibm.com(4445)
```

Information associée

NativeHALocalInstance strophe du fichier d' qm.ini

NativeHARecoveryGroup strophe du fichier d' qm.ini

NativeHAInstance strophe du fichier d' qm.ini

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 100
- «Déploiement et configuration des gestionnaires de file d'attente à l'aide de l'exemple de diagramme Helm », à la page 194

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 103.
- «Configuration de la haute disponibilité pour les gestionnaires de files d'attente à l'aide de IBM MQ Operator», à la page 115.
- «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 153.
- «Intégration d' IBM MQ à la fonction de trace IBM Instana», à la page 156.
- «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 164.
- «Ajout d'annotations et d'étiquettes personnalisées aux ressources du gestionnaire de files d'attente», à la page 168.
- «Désactivation des vérifications des webhooks d'exécution», à la page 169.
- «Désactivation des mises à jour des valeurs par défaut de la spécification du gestionnaire de files d'attente», à la page 169.

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 «mq.ibm.com/v1beta1: Versions actuelles de la licence», à la page 256. 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 «mq.ibm.com/v1beta1: Versions actuelles de la licence», à la page 256 .

     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: 9.4.3.0-r1
  license:
    accept: false
    license: L-CYPF-CRPF3H
    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 «mq.ibm.com/v1beta1: Versions actuelles de la licence», à la page 256 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 67.

  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 :

  Remarque : Cet exemple utilise une licence IBM MQ Advanced , mais vous pouvez également utiliser une licence IBM MQ . Pour plus d'informations, voir «Configurer les gestionnaires de files d'attente avec les annotations de licence IBM MQ en utilisant la fonction IBM MQ Operator», à la page 166.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart
spec:
  version: 9.4.3.0-r1
  license:
    accept: false
    license: L-NUUP-23NH8Y
  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 «mq.ibm.com/v1beta1: Versions actuelles de la licence», à la page 256 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 153
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 224
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 103
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 103
- «Exemple: configuration d'un gestionnaire de files d'attente avec l'authentification TLS mutuelle», à la page 109

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. Celles-ci doivent être créées dans l'espace de noms (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 ConfigMap: Appliquez le 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 file d'attente avec la configuration suivante, en utilisant la ligne de commande ou la console web Red Hat OpenShift Container Platform .

Remarque : Cet exemple utilise une licence IBM MQ Advanced , mais vous pouvez également utiliser une licence IBM MQ . Pour plus d'informations, voir «Configurer les gestionnaires de files d'attente avec les annotations de licence IBM MQ en utilisant la fonction IBM MQ Operator», à la page 166.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: mqsc-ini-qm
spec:
  version: 9.4.3.0-r1
  license:
    accept: false
    license: L-NUUP-23NH8Y
    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`. Voir «mq.ibm.com/v1beta1: Versions actuelles de la licence», à la page 256 pour des détails sur la licence.

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` est toujours traité avant `example2.mqsc`, quel que soit l'ordre dans lequel ils apparaissent dans la configuration du gestionnaire de file 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 la ConfigMap, par exemple des modifications aux commandes MQSC ou aux fichiers INI, vous devez alors 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 *Exemple : Fournir des fichiers supplémentaires à un gestionnaire de file d'attente*

Cet exemple crée un dossier Kubernetes ConfigMap qui contient deux fichiers. Un gestionnaire de file d'attente est alors déployé pour monter ces fichiers dans le conteneur d' IBM MQ s aux emplacements spécifiés.

Pourquoi et quand exécuter cette tâche

Des fichiers supplémentaires peuvent être fournis lors du déploiement d'un gestionnaire de file d'attente. Ces fichiers doivent être définis dans un ou plusieurs fichiers Kubernetes ConfigMaps et Secrets. Celles-ci doivent être créées dans l'espace de noms (projet) où vous déployez le gestionnaire de file d'attente. Un secret d' Kubernetes s doit être utilisé lorsque les fichiers contiennent des données sensibles.

Remarque : Cette fonctionnalité n'est prise en charge qu'avec l' IBM MQ Operator 3.5.0 ou version ultérieure.

Exemple

L'exemple suivant crée un fichier Kubernetes ConfigMap qui contient deux fichiers. Un gestionnaire de file d'attente est alors déployé pour monter ces fichiers dans le conteneur d' IBM MQ s aux emplacements spécifiés.

Exemple ConfigMap: Appliquez le YAML suivant dans votre cluster.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: files-example
data:
  file1.txt: |
    Test File 1
  file2.txt: |
    Test File 2
```

Exemple QueueManager: Déployez votre gestionnaire de file d'attente avec la configuration suivante, en utilisant la ligne de commande ou la console web Red Hat OpenShift Container Platform .

Remarque : Cet exemple utilise une licence IBM MQ Advanced , mais vous pouvez également utiliser une licence IBM MQ . Pour plus d'informations, voir «Configurer les gestionnaires de files d'attente avec les annotations de licence IBM MQ en utilisant la fonction IBM MQ Operator», à la page 166.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: files-qm
spec:
  version: 9.4.3.0-r1
  license:
    accept: false
    license: L-NUUP-23NH8Y
    use: Production
  queueManager:
```

```
    files:
    - configMap:
        name: files-example
        defaultMountPath: "/etc/mqm/myfiles"
        items:
        - item: file1.txt
        - item: file2.txt
          mountPath: "/etc/mqm/otherfiles"
    storage:
      queueManager:
        type: ephemeral
```

Important : Si vous acceptez le contrat de licence IBM MQ Advanced, modifiez `accept: false` par `accept: true`. Voir «mq.ibm.com/v1beta1: Versions actuelles de la licence», à la page 256 pour des détails sur la licence.

Emplacement des fichiers :

- Le fichier `file1.txt` est monté à l'emplacement `/etc/mqm/myfiles/file1.txt`, qui est l' **`defaultMountPath`** e spécifié pour cette ConfigMap.
- Le fichier `file2.txt` est monté à l'emplacement `/etc/mqm/otherfiles/file2.txt`, qui est la substitution d' **`mountPath`** s pour ce fichier spécifique.

Informations supplémentaires :

- Lorsque **`defaultMountPath`** n'est pas défini (pour un ConfigMap ou Secret) et que la priorité d' **`mountPath`** s n'est pas spécifiée, les fichiers sont montés par défaut à l'emplacement `/etc/mqm/customfiles`
- Si vous modifiez le contenu de l'un des fichiers de l' ConfigMap, vous devez redémarrer manuellement les gestionnaires de file d'attente pour prendre en compte ces modifications. Pour les gestionnaires de file d'attente à instance unique, supprimez le pod pour déclencher le redémarrage requis. Pour les déploiements Native HA, redémarrez d'abord les pods en veille en les supprimant. Lorsqu'ils sont à nouveau en état de fonctionnement, supprimez le pod actif pour le redémarrer. Cet ordre de redémarrage garantit un temps d'arrêt minimal pour le gestionnaire de file d'attente.

OpenShift CP4I *Exemple : Fournir des variables d'environnement supplémentaires pour un gestionnaire de file d'attente*

Cet exemple déploie un gestionnaire de file d'attente avec deux variables d'environnement supplémentaires. Ces variables d'environnement sont définies sur le conteneur d' IBM MQ .

Pourquoi et quand exécuter cette tâche

Des variables d'environnement supplémentaires peuvent être fournies lorsqu'un gestionnaire de file d'attente est déployé. Cela permet de définir des variables d'environnement qui ne sont pas déjà exposées par l' IBM MQ Operator.

Remarque : Cette fonctionnalité n'est prise en charge qu'avec l' IBM MQ Operator , version 3.5.0 ou ultérieure.

Exemple

L'exemple suivant déploie un gestionnaire de file d'attente avec deux variables d'environnement supplémentaires. Ces variables d'environnement sont définies sur le conteneur d' IBM MQ .

Exemple QueueManager: Déployez votre gestionnaire de file d'attente avec la configuration suivante, en utilisant la ligne de commande ou la console web Red Hat OpenShift Container Platform .

Remarque : Cet exemple utilise une licence IBM MQ Advanced , mais vous pouvez également utiliser une licence IBM MQ . Pour plus d'informations, voir «Configurer les gestionnaires de files d'attente avec les annotations de licence IBM MQ en utilisant la fonction IBM MQ Operator», à la page 166.

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

```
  name: env-qm
spec:
  version: 9.4.3.0-r1
  license:
    accept: false
    license: L-NUUP-23NH8Y
    use: Production
  queueManager:
    env:
    - name: TEST_ENV_1
      value: "true"
    - name: TEST_ENV_2
      value: "false"
    storage:
      queueManager:
        type: ephemeral
```

Important : Si vous acceptez le contrat de licence IBM MQ Advanced, modifiez `accept: false` par `accept: true`. Voir «mq.ibm.com/v1beta1: Versions actuelles de la licence», à la page 256 pour des détails sur la licence.

Informations supplémentaires :

- Si vous spécifiez une variable d'environnement qui aurait autrement été définie par l' IBM MQ Operator, cela pourrait entraîner un comportement imprévisible. Dans ce cas, un état d'avertissement est ajouté à votre fichier d' QueueManager.

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 la commande " 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

```
kubectl 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 109
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 116
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 150
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.

OpenShift CP4I Linux *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 107.
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 109. 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`.

   Remarque : Cet exemple utilise une licence IBM MQ Advanced , mais vous pouvez également utiliser une licence IBM MQ . Pour plus d'informations, voir «Configurer les gestionnaires de files d'attente avec les annotations de licence IBM MQ en utilisant la fonction IBM MQ Operator», à la page 166.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: exampleqm
spec:
  license:
    accept: false
    license: L-NUUP-23NH8Y
    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: 9.4.3.0-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 109et que l' *exemple-tls-configmap* ConfigMap a été créé à l'étape «2», à la page 109

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 111.

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 107:
  - `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 109

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 153.

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 168.

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

vous pouvez configurer la haute disponibilité pour les gestionnaires de files d'attente en utilisant soit la solution Native HA, soit la solution de gestionnaire de files d'attente multi-instances.

Pourquoi et quand exécuter cette tâche

Suivez la procédure correspondant à la solution de haute disponibilité que vous avez choisie.

Procédure

- «Native HA», à la page 43.
- «Exemple: Configuration de Native HA à l'aide de IBM MQ Operator», à la page 116.
- «Configuration d'un gestionnaire de files d'attente multi-instance à l'aide de IBM MQ Operator», à la page 150.

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-NUUP-23NH8Y
    use: Production
  queueManager:
    availability:
      type: NativeHA
  version: 9.4.3.0-r1
```

Remarque : Cet exemple utilise une licence IBM MQ Advanced . A partir de 9.4.2, les gestionnaires de files d'attente sous licence IBM MQ peuvent également être configurés pour utiliser les fonctions Native HA et Cross-Region Replication en ajoutant des licences supplémentaires à un gestionnaire de files d'attente. Pour plus d'informations, voir «Configurer Native HA et Cross-Region Replication sur IBM MQ Queue Managers sous licence en utilisant la fonction IBM MQ Operator», à la page 119.

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 116.

Tâches associées

«Exemple: Configuration de Native HA à l'aide de IBM MQ Operator», à la page 116
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.

OpenShift MQ Adv. Kubernetes *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 107.
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 116. 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`.

   Remarque :

   Cet exemple utilise une licence IBM MQ Advanced . A partir de IBM MQ 9.4.2, les fonctionnalités Native HA et Cross-Region Replication sont disponibles sur les gestionnaires de file d'attente qui utilisent une licence IBM MQ . Pour plus d'informations, voir «Configurer Native HA et Cross-Region Replication sur IBM MQ Queue Managers sous licence en utilisant la fonction IBM MQ Operator», à la page 119.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: exampleqm
spec:
  license:
    accept: false
    license: L-NUUP-23NH8Y
    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: 9.4.2.0-r2
  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 116et que l' *exemple-nativeha-configmap* ConfigMap a été créé à l'étape «2», à la page 116

   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 111.

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 du <pod actif>. Le pod actif doit être prêt, c'est-à-dire le pod avec la mention ready 1/1 indiquée par la commande de l'étape c :

      ```
      oc exec -t <active-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 111.

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. V 9.4.2 *Configuration de Native HA CRR à l'aide de l' IBM MQ Operator*

La réplication inter-régions (CRR) native HA est configurée à l'aide de l'API d' `QueueManager` , et des options avancées sont disponibles à l'aide d'un fichier INI.

Remarque : Cet exemple utilise une licence IBM MQ Advanced . De IBM MQ 9.4.2, IBM MQ Les gestionnaires de file d'attente sous licence peuvent également être configurés pour utiliser les fonctions Native HA CRR en ajoutant des licences supplémentaires à un gestionnaire de file d'attente. Pour plus d'informations, voir «Configurer Native HA et Cross-Region Replication sur IBM MQ Queue Managers sous licence en utilisant la fonction IBM MQ Operator», à la page 119.

Native HA CRR est configuré en utilisant `.spec.queueManager.availability` dans l'API d' `QueueManager` . La zone `.spec.queueManager.availability.type` doit être définie sur `NativeHA`. Le champ `.spec.queueManager.availability.nativeHAGroups.local.role` est défini sur `Live` ou `Recovery` pour spécifier le rôle du groupe local.

Vous spécifiez les détails du groupe distant dans la configuration de la section « `.spec.queueManager.availability.nativeHAGroups.remotes` ». Vous indiquez le nom d'hôte et le port de la route OpenShift utilisée par le groupe distant où les données seront répliquées. Le groupe actif doit inclure au minimum la section d' `remotes` s pour pouvoir se répliquer vers le groupe de récupération. Le groupe de récupération doit inclure la section d' `remotes` s afin de pouvoir répliquer vers l'autre site s'il devient un jour actif. (Voir «Basculement et basculement automatique en cas de panne de la HA CRR native», à la page 242.)

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 116.

Vous devez configurer un secret TLS et des chiffrements à utiliser entre les groupes live et recovery lors de la réplication. Vous pouvez utiliser le même secret que celui spécifié pour la réplication entre instances, ou vous pouvez spécifier un autre secret à cette fin.

Tâches associées

«Exemple: Configuration de Native HA à l'aide de IBM MQ Operator», à la page 116
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.

OpenShift CP4I V 9.4.2 ***Configurer Native HA et Cross-Region Replication sur IBM MQ Queue Managers sous licence en utilisant la fonction IBM MQ Operator***

De IBM MQ 9.4.2, IBM MQ Les gestionnaires de file d'attente sous licence peuvent être configurés pour utiliser les fonctions de réplication transrégionale (CRR) de Native HA en ajoutant une licence supplémentaire à un gestionnaire de file d'attente.

Avant de commencer

Remarque : Ces instructions ne s'appliquent que si vous utilisez le site IBM MQ Operator. Si vous déployez des gestionnaires de file d'attente sur Kubernetes, consultez plutôt «Configuration de Native HA et de Cross-Region Replication sur IBM MQ gestionnaires de files d'attente sous licence dans Kubernetes», à la page 192 .

Le site IBM License Service est nécessaire pour utiliser les fonctions Native HA CRR sur les gestionnaires de file d'attente sous licence IBM MQ . Consultez l'étape Déployer le License Service de la IBM MQ Operator processus d'installation.

Important : Cette tâche implique la mise à jour du YAML du gestionnaire de file d'attente, ce qui déclenche un redémarrage du pod.

Procédure

1. Ajoutez l'une des CR `IBMLicensingDefinition` , ou les deux, selon que vous avez besoin d'une utilisation en production ou hors production. Ces CR ajoutent des annotations de licence supplémentaires à votre gestionnaire de file d'attente pour les fonctionnalités Native HA CRR.

Répétez cette étape dans chaque espace de noms dans lequel vous utiliserez la licence supplémentaire pour les fonctionnalités Native HA CRR.

- **Option 1 :** ajouter des CR en utilisant la console OpenShift :
  - Connectez-vous à la console OpenShift avec vos identifiants d'administrateur du cluster Red Hat OpenShift Container Platform .
  - Changez le **projet** pour l'espace de noms dans lequel votre gestionnaire de file d'attente est déployé en sélectionnant l'espace de noms dans la liste déroulante **Projet.**
  - Cliquez sur l'icône **Quick Create** (signe plus) en haut à droite de l'écran pour ouvrir la page **Import YAML**.
  - Ajoutez le CR `IBMLicensingDefinition` YAML approprié (voir les exemples à la fin de cette étape).
  - Cliquez sur **Créer**.
  - Répétez ces étapes pour ajouter l'autre `IBMLicensingDefinition` CR si nécessaire.
- **Option 2 :** ajouter des CR en utilisant la ligne de commande :
  - Connectez-vous à l'interface de ligne de commande (CLI) Red Hat OpenShift en utilisant la commande `oc login` .
  - Définissez votre `project` (espace de noms) sur celui dans lequel votre gestionnaire de file d'attente est déployé :

    ```
    oc project <QUEUE-MANAGER-NAMESPACE>
    ```

  - Créez un fichier `IBMLicensingDefinition` YAML (voir les exemples à la fin de cette étape).
  - Appliquer le fichier `IBMLicensingDefinition` YAML :

    ```
    oc apply -f <IBMLicensingDefinition-UNIQUE-NAME>.yaml
    ```

  - Répétez ces étapes pour ajouter l'autre `IBMLicensingDefinition` CR si nécessaire.

Dans l'exemple suivant `IBMLicensingDefinition` CR YAML files, changez la valeur `<QUEUE-MANAGER-NAMESPACE>` pour le projet dans lequel votre gestionnaire de file d'attente est déployé.

- Pour la production :

```
apiVersion: operator.ibm.com/v1
kind: IBMLicensingDefinition
metadata:
  name: ibmlicensingdefinition-mq-nha-prod
  namespace: <QUEUE-MANAGER-NAMESPACE>
spec:
  action: cloneModify
  condition:
    metadata:
      annotations:
        additionalLicense: L-EZBK-Y86SSH
        additionalLicenseUse: Production
  scope: cluster
  set:
    productID: 46a57a4a66ee4b904e9b5544b53a2ba2
    productName: IBM MQ Native HA and Cross-Region Replication Add-on
```

- Pour une utilisation hors production :

```
apiVersion: operator.ibm.com/v1
kind: IBMLicensingDefinition
metadata:
  name: ibmlicensingdefinition-mq-nha-non-prod
  namespace: <QUEUE-MANAGER-NAMESPACE>
spec:
  action: cloneModify
  condition:
    metadata:
      annotations:
        additionalLicense: L-EZBK-Y86SSH
        additionalLicenseUse: NonProduction
```

```
  scope: cluster
  set:
    productID: 525b1331a74d4a209d9f6e9c908e374e
    productName: IBM MQ Native HA and Cross-Region Replication Add-on for Non-Production
Env
```

2. Ajoutez les annotations pertinentes au site `spec.annotations` de votre gestionnaire de file d'attente.

   Important : La mise à jour du YAML du gestionnaire de file d'attente déclenche un redémarrage du pod.

   Suivez les instructions à l'adresse «Ajout d'annotations aux gestionnaires de file d'attente à l'aide de la fonction IBM MQ Operator», à la page 167 pour ajouter l'une des annotations suivantes, selon vos besoins :

   - Pour IBM MQ et IBM MQ Native HA et Cross-Region Replication Add-on entitlements :

     ```
     spec:
       annotations:
         productID: c661609261d5471fb4ff8970a36bccea
         productName: IBM MQ
         additionalLicense: L-EZBK-Y86SSH
         additionalLicenseUse: Production
     ```

   - Pour IBM MQ for Non-Production Environment et IBM MQ Native HA and Cross-Region Replication Add-on for Non-Production entitlements

     ```
     spec:
       annotations:
         productID: 151bec68564a4a47a14e6fa99266deff
         productName: IBM MQ for Non-Production Environment
         additionalLicense: L-EZBK-Y86SSH
         additionalLicenseUse: NonProduction
     ```

   Répétez cette étape pour d'autres gestionnaires de file d'attente si nécessaire.

OpenShift MQ Adv. Kubernetes *Exemple : Déploiement d'une configuration simple de CRR Native HA à l'aide de l'opérateur IBM MQ*

Cet exemple déploie un gestionnaire de files d'attente à l'aide d'une configuration simple de réplication transrégionale Native HA (CRR) sur le site OpenShift® Container Platform à l'aide de l'opérateur IBM MQ .

Avant de commencer

Pour réaliser cet exemple, vous devez d'abord suivre les étapes suivantes :

- Sur la ligne de commande, connectez-vous aux clusters OpenShift Container Platform (OCP) que vous utilisez.
- Créez deux projets/espaces de noms OCP pour cet exemple. L'exemple se réfère aux sites de Londres et de Rome. Ils renvoient soit à deux espaces de noms pour Londres et Rome dans une seule grappe, soit à un espace de noms dans chaque grappe pour chaque site.
- Assurez-vous que IBM MQ Operator 3.5.0 ou une version supérieure est installé dans les clusters et disponible dans les espaces de noms.
- Assurez-vous que l'outil de ligne de commande OpenSSL est installé.
- Complétez la tâche «Préparation de l'utilisation de 'Kubernetes par la création d'un secret d'extraction», à la page 88 .
- Lisez «Déploiement d'un gestionnaire de files d'attente simple à l'aide de IBM MQ Operator», à la page 100 afin de vous familiariser avec l'utilisation de l'opérateur IBM MQ .

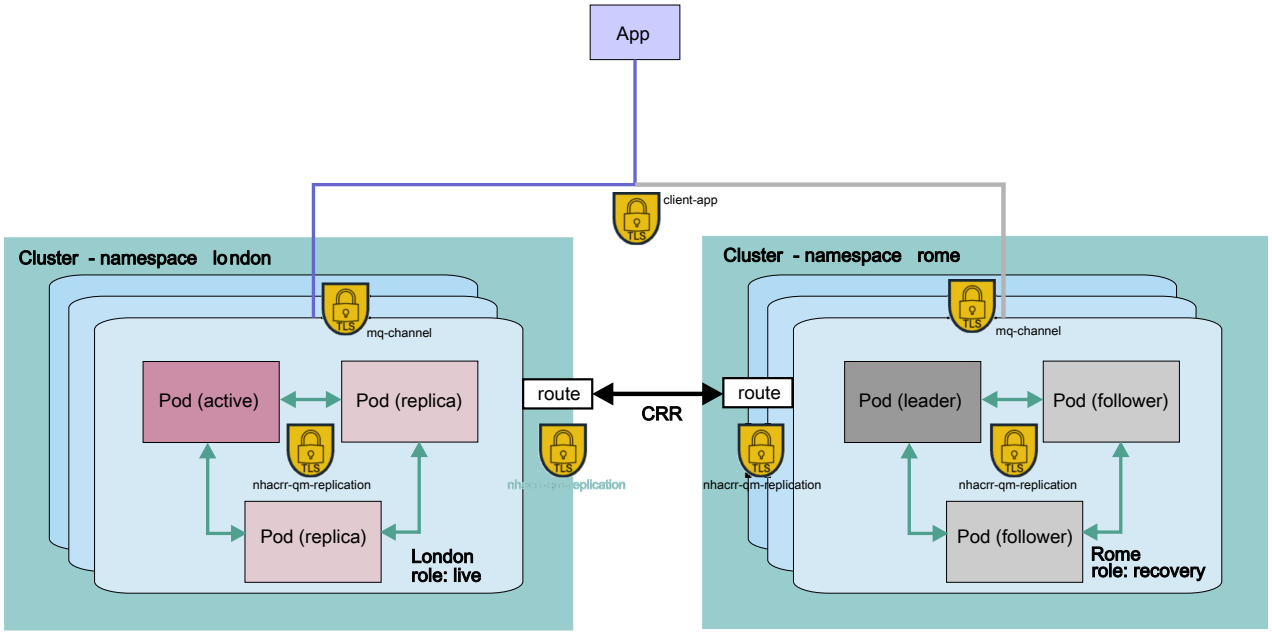
Pourquoi et quand exécuter cette tâche

La configuration Native HA CRR définit deux groupes Native HA, qui forment ensemble un gestionnaire de file d'attente hautement disponible. Ces groupes sont généralement déployés dans deux clusters OCP

situés à des endroits différents, mais cet exemple vous permet de déployer les deux groupes dans le même cluster si vous le souhaitez. Voir Native HA Cross-Region Replication pour une vue d'ensemble de Native HA CRR.

Pour des raisons de simplicité, cet exemple utilise un seul certificat pour sécuriser l'ensemble du trafic de réplication du gestionnaire de file d'attente, nommé nhacrr-qm-replication.

Pour comprendre comment les certificats sont utilisés dans Native HA CRR et comment utiliser différents certificats pour sécuriser le trafic des groupes et des instances, voir «Exemple : utilisation de plusieurs certificats TLS avec Native HA CRR», à la page 134.



Important : Ces exemples sont destinés à vous aider à démarrer, ils ne conviennent pas à un environnement de production. La gestion des certificats est un sujet complexe, réservé aux utilisateurs avancés. Pour la production, vous devez prendre en compte des éléments tels que la rotation, la révocation, la longueur des clés, la reprise après sinistre et bien d'autres encore.

Cet exemple fournit la ressource personnalisée YAML pour définir deux groupes Native HA fournissant une réplication interrégionale.

Procédure

1. Créez les certificats à utiliser pour le trafic de réplication du gestionnaire de file d'attente.

   Utilisez les commandes suivantes pour créer une clé privée et l'utiliser pour émettre un certificat auto-signé pour votre autorité de certification interne. Créez ensuite une clé privée et un certificat pour le gestionnaire de file d'attente, signés par votre autorité de certification interne :

   ```
   openssl genpkey -algorithm rsa -pkeyopt rsa_keygen_bits:4096 -out nhacrr-ca.key
   openssl req -x509 -new -nodes -key "nhacrr-ca.key" -sha512 -days 30 -subj "/CN=nhacrr-
   selfsigned-ca" -out nhacrr-ca.crt
   openssl req -new -nodes -out nhacrr-qm-replication.csr -newkey rsa:4096 -keyout nhacrr-qm-
   replication.key -subj "/CN=nhacrr-qm-replication"
   openssl x509 -req -in nhacrr-qm-replication.csr -CA nhacrr-ca.crt -CAkey nhacrr-ca.key
   -CAcreateserial -out nhacrr-qm-replication.crt -days 7 -sha512
   ```

   Pour plus d'informations sur ces commandes, voir «Création d'une infrastructure PKI autosignée à l'aide de OpenSSL», à la page 107.
2. Créez deux secrets Kubernetes contenant la clé et les certificats du gestionnaire de file d'attente. Créez un secret nommé nhacrr-qm-replication sur les sites de Rome et de Londres, pour stocker la clé et le certificat des instances du gestionnaire de file d'attente.

Utilisez la commande suivante pour créer les secrets. Passez à chaque espace de noms (Rome puis Londres) avant d'exécuter la commande.

```
kubectl create secret generic nhacrr-qm-replication --type="kubernetes.io/tls" --from-
file=tls.key=nhacrr-qm-replication.key --from-file=tls.crt=nhacrr-qm-replication.crt --from-
file=nhacrr-ca.crt
```

3. Créez une paire de certificats pour le trafic du gestionnaire de file d'attente comme décrit dans «Création d'une infrastructure PKI autosignée à l'aide de OpenSSL», à la page 107. Créez le secret sur les sites de Rome et de Londres (il s'agit des certificats nommés mq-channel et client-app dans le diagramme).
4. Créer une carte de configuration contenant des commandes MQSC et un fichier INI pour Rome et Londres. Créer une ressource ConfigMap contenant les commandes MQSC pour créer une nouvelle file d'attente de messages persistante et un canal SVRCONN, et pour ajouter un enregistrement d'authentification de canal qui autorise l'accès au canal. Sur le site de Rome, entrez le YAML suivant dans la console web de l'OCP, sélectionnez ConfigMaps puis Create ConfigMap, ou utilisez la commande **oc apply** . Répétez cette opération pour le site de Londres :

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: nhacrr-configmap
data:
  nhacrr.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') DEFPSIST(YES) REPLACE
    SET AUTHREC PROFILE('EXAMPLE.QUEUE') PRINCIPAL('app1') OBJTYPE(QUEUE)
AUTHADD(BROWSE,PUT,GET,INQ)
  nhacrr.ini: |
    Service:
        Name=AuthorizationService
        EntryPoints=14
        SecurityPolicy=UserExternal
```

Le MQSC définit un canal nommé MTLS.SVRCONN et une file d'attente nommée EXAMPLE.QUEUE. Le canal est configuré pour n'autoriser l'accès qu'aux clients qui présentent un certificat avec un CN de example-app1 comme celui créé à l'étape 3.

5. Déployer le groupe Native HA Recovery. Créez un nouveau groupe de récupération sur le site de Rome. Assurez-vous d'être dans l'espace de noms rome, puis entrez le YAML suivant dans la console web de l'OCP pour créer le site QueueManager, ou utilisez la commande **oc apply** . Vérifiez que la licence est correcte et acceptez la licence en modifiant `false` par `true`.

   Remarque : Vous devez utiliser le même nom de gestionnaire de file d'attente dans le groupe actif et le groupe de secours. Par défaut, si vous n'indiquez pas `.spec.queuemanager.name`, il s'agit de `.metadata.name`. Vous pouvez obtenir des noms correspondants en utilisant l'une des méthodes suivantes :

   - Nommez votre ressource gestionnaire de file d'attente de la même manière dans chaque cluster (c'est-à-dire `.metadata.name` correspond) et ne remplacez pas le nom
   - Nommez votre gestionnaire de file d'attente individuellement dans `.metadata.name` et remplacez le nom dans `.spec.queuemanager.name` afin que les noms correspondent dans les deux sites.

   Dans cet exemple, le nom du gestionnaire de file d'attente est défini sur `exampleqm` en utilisant `.metadata.name`.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: exampleqm
  labels:
    role: Recovery
spec:
```

```
  license:
    accept: false
    license: L-NUUP-23NH8Y
    use: Production
  queueManager:
    name: EXAMPLEQM
    availability:
      type: NativeHA
      tls:
        secretName: nhacrr-qm-replication
      nativeHAGroups:
        local:
          name: "rome"
          role: "Recovery"
    mqsc:
    - configMap:
        name: nhacrr-configmap
        items:
        - nhacrr.mqsc
    ini:
    - configMap:
        name: nhacrr-configmap
        items:
        - nhacrr.ini
    storage:
      queueManager:
        type: persistent-claim
  version:  9.4.2.0-r1
  pki:
    keys:
      - name: default
        secret:
          secretName: example-qm-tls
          items:
            - tls.key
            - tls.crt
            - ca.crt
```

6. Vérifiez que le groupe de reprise a été déployé correctement :
   a) Exécutez la commande `oc get qmgr exampleqm` pour afficher la ressource du gestionnaire de files d'attente et attendez que le gestionnaire de files d'attente soit en cours d'exécution. Vous devez voir le message suivant :

```
NAME            PHASE
exampleqm   Running
```

   b) Lorsque le gestionnaire de files d'attente est en cours d'exécution, vérifiez l'état des pods à l'aide de la commande `oc get pods` . Un pod doit être prêt à l'emploi :

```
NAME                 READY   STATUS    RESTARTS   AGE
exampleqm-ibm-mq-0   1/1     Running   0          2m45s
exampleqm-ibm-mq-1   0/1     Running   0          2m45s
exampleqm-ibm-mq-2   0/1     Running   0          2m45s
```

   c) Vérifiez l'état du groupe de reprise en utilisant la commande `oc exec -t <leader-pod> -- dspmq -o nativeha -g` où `<leader-pod>` est le nom du pod prêt 1/1 (exampleqm-ibm-mq-0 dans l'exemple).

```
QMNAME(EXAMPLEQM)                                          ROLE(Leader)
INSTANCE(exampleqm-ibm-mq-0) INSYNC(yes) QUORUM(1/3) GRPLSN() GRPNAME(rome)
GRPROLE(Recovery)
 GRPNAME(rome) GRPROLE(Recovery) GRPADDR(Unknown) GRPVER(9.4.2.0) GRSTATUS(Waiting
for connection) RCOVLSN() RCOVTIME() BACKLOG(Unknown) INSYNC(Unknown) SYNCTIME()
ALTDATE(2025-04-25) ALTTIME(13.48.15)
```

   Les éléments clés de ce statut sont les suivants :

   - `Role(Leader)`
   - `QUORUM(1/3)` - cela reste 1/3 jusqu'à ce que les groupes Live et Recovery se connectent
   - `GRSTATUS(Waiting for connection)` - il est mis à jour lorsque le groupe Live se connecte

7. Déployer le groupe Native HA Live. Le groupe Live est déployé sur le site de Londres :

Pour créer le nouveau groupe live sur le site de Londres, assurez-vous d'être dans le bon espace de noms, puis entrez le YAML suivant dans la console web de l'OCP, ou en utilisant la ligne de commande. Vérifiez que la licence spécifiée est correcte et acceptez-la en remplaçant false par true.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: exampleqm
  labels:
    role: Live
spec:
  license:
    accept: false
    license: L-NUUP-23NH8Y
    use: Production
  queueManager:
    name: EXAMPLEQM
    availability:
      type: NativeHA
      tls:
        secretName: nhacrr-qm-replication
      nativeHAGroups:
        local:
          name: "london"
          role: "Live"
        remotes:
          - name: "rome"
            addresses:
              - "<rome group hostname>:443"
    mqsc:
    - configMap:
        name: nhacrr-configmap
        items:
        - nhacrr.mqsc
    ini:
    - configMap:
        name: nhacrr-configmap
        items:
        - nhacrr.ini
    storage:
      queueManager:
        type: persistent-claim
  version: 9.4.2.0-r1
  pki:
    keys:
      - name: default
        secret:
          secretName: example-qm-tls
          items:
            - tls.key
            - tls.crt
            - ca.crt
```

Mettez à jour l'adresse dans le YAML qui spécifie l'adresse que le groupe en direct utilise pour se connecter au groupe de récupération ("<rome group hostname>:443 " dans l'exemple). Remplacez-le par l'attribut host de la route exampleqm-ibm-mq-nhacrr sur le site de Rome. Pour récupérer cet attribut, assurez-vous d'être sur le site de Rome et utilisez la commande suivante pour récupérer le nom d'hôte de la route nhacrr :

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

Remarque :

Deux routes sont créées lorsque les groupes CRR sont déployés, l'une se terminant par `qm` et l'autre par `nhacrr`. La route `qm` est utilisée pour envoyer des messages au gestionnaire de file d'attente, tandis que la route `nhacrr` est utilisée pour la communication de groupe à groupe. Ajoutez toujours les adresses d'hôtes de la route `nhacrr` comme adresse distante pour le groupe dans le YAML.

8. Vérifiez le groupe Live et la réplication interrégionale. Répétez les opérations de l'étape 6, mais cette fois pour le groupe de Londres.
   a) Exécutez la commande `oc get qmgr exampleqm` pour afficher la ressource du gestionnaire de files d'attente et attendez que le gestionnaire de files d'attente soit en cours d'exécution. Vous devez voir le message suivant :

```
NAME            PHASE
exampleqm   Running
```

b) Lorsque le gestionnaire de files d'attente est en cours d'exécution, vérifiez l'état des pods à l'aide de la commande `oc get pods` . Un pod doit être prêt à l'emploi :

```
NAME                 READY   STATUS    RESTARTS   AGE
exampleqm-ibm-mq-0   1/1     Running   0          2m45s
exampleqm-ibm-mq-1   0/1     Running   0          2m45s
exampleqm-ibm-mq-2   0/1     Running   0          2m45s
```

c) Vérifiez l'état du groupe Live en utilisant la commande `oc exec -t <active-pod> -- dspmq -o nativeha -g` où `<active-pod>` est le nom du pod prêt 1/1 (exampleqm-ibm-mq-0 dans l'exemple).

```
QMNAME(EXAMPLEQM)                                          ROLE(Active)
INSTANCE(exampleqm-ibm-mq-0) INSYNC(yes) QUORUM(3/3) GRPLSN(<0:0:87:58322>)
GRPNAME(london) GRPROLE(Live)
 GRPNAME(london) GRPROLE(Live) GRPADDR(Unknown) GRPVER(9.4.2.0) GRSTATUS(Normal)
RCOVLSN(<0:0:87:58322>) RCOVTIME(2025-05-16T14:27:09.051799Z) INITLSN(<0:0:9:4696>)
INITTIME(2025-05-12T16:58:13.582274Z) LIVETIME(2025-05-12T16:58:16.319844Z)
ALTDATE(2025-05-16) ALTTIME(14.27.10)
 GRPNAME(rome) GRPROLE(Recovery) GRPADDR(exampleqm-ibm-mq-nhacrr-
<namespace>.apps.<cluster>) GRPVER(9.4.2.0) CONNGRP(yes) GRSTATUS(Normal)
RCOVLSN(<0:0:87:58322>) RCOVTIME(2025-05-16T14:27:09.051799Z) BACKLOG(0) INSYNC(yes)
SYNCTIME(2025-05-16T14:28:38.084565Z) ALTDATE(2025-05-16) ALTTIME(14.28.31)
```

Les éléments clés de ce statut sont les suivants :

- `Role(Active)`
- `QUORUM(3/3)` - le groupe en direct dispose d'un quorum complet
- `GRSTATUS(Normal)`

d) Vérifiez à nouveau l'état du groupe de reprise en exécutant la commande `oc exec -t <leader-pod> -- dspmq -o nativeha -g` sur le site de Rome.

```
QMNAME(EXAMPLEQM)                                          ROLE(Leader)
INSTANCE(exampleqm-ibm-mq-0) INSYNC(yes) QUORUM(3/3) GRPLSN(<0:0:87:58322>)
GRPNAME(rome) GRPROLE(Recovery)
 GRPNAME(rome) GRPROLE(Recovery) GRPADDR(exampleqm-ibm-mq-nhacrr-
<namespace>.apps.<cluster>) GRPVER(9.4.3.0) GRSTATUS(Normal)
RCOVLSN(<0:0:87:58322>) RCOVTIME(2025-05-16T14:27:09.051799Z) BACKLOG(0) INSYNC(yes)
SYNCTIME(2025-05-16T14:30:40.134627Z) ALTDATE(2025-05-16) ALTTIME(14.27.17)
 GRPNAME(london) GRPROLE(Live) GRPADDR(Unknown) GRPVER(9.4.3.0) CONNGRP(yes)
GRSTATUS(Normal) RCOVLSN(<0:0:87:58322>) RCOVTIME(2025-05-16T14:27:09.051799Z)
INITLSN(<0:0:9:4696>) INITTIME(2025-05-12T16:58:13.582274Z)
LIVETIME(2025-05-12T16:58:16.319844Z) ALTDATE(2025-05-16) ALTTIME(14.30.41)
```

Notez que les champs `QUORUM(3/3)` et `GRSTATUS(Normal)` ont été mis à jour.

9. Testez la connexion au gestionnaire de file d'attente. Pour confirmer que le gestionnaire de files d'attente est configuré et disponible, suivez les étapes décrites à l'adresse «Test d'une connexion TLS mutuelle à un gestionnaire de files d'attente à partir de votre ordinateur portable», à la page 111, en vous assurant que le nom d'hôte du gestionnaire de files d'attente, déterminé à l'étape 2, est le nom d'hôte de la route du site de Londres.
10. Ajouter une adresse distante au groupe Native HA Recovery. Lorsque Rome devient le groupe actif, il doit connaître l'adresse distante de Londres afin de pouvoir se connecter à Londres en tant que groupe de récupération. Pour l'ajouter, mettez à jour le site `spec.queueManager.availability.nativeHAGroups` pour le site de Rome, avec l'ajout suivant, sous la strophe locale :

```
          remotes:
            - name: "london"
              addresses:
              - "<london group hostname>:443"
```

Remplacez l'adresse par l'attribut host de la route `exampleqm-ibm-mq-nhacrr` sur le site de Londres. Assurez-vous d'être sur le site de Londres et utilisez la commande suivante pour récupérer le nom d'hôte de la route nhacrr :

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

Une fois la mise à jour effectuée, attendez que les pods redémarrent, qu'ils fonctionnent et qu'un pod soit le pod 1/1 prêt.

11. Ajoutez le nom d'hôte du site de Rome au fichier `ccdt.json` . Après un basculement, lorsque Rome devient le site en direct, les clients doivent se connecter au groupe de Rome. L'ajout du nom d'hôte de Rome au fichier `ccdt.json` facilite cette reconnexion.

    Assurez-vous d'être sur le site de Rome et utilisez la commande suivante pour récupérer le nom d'hôte de la route `qm` :

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

    Remarque : Il y a deux routes, dans ce cas c'est la route pour le trafic de messages qui doit être utilisée. Vous devez ajouter les adresses des hôtes de la route `qm` au fichier `ccdt.json` .

    Dans le fichier `ccdt.json` , ajoutez une deuxième paire de noms d'hôtes et de ports à la liste des connexions, de sorte que votre liste de connexions ressemble maintenant à ceci :

    ```
    "connection":
                    [
                        {
                            "host": "<hostname from London qm route>",
                            "port": 443
                        },
                        {
                            "host": "<hostname from Rome qm route>",
                            "port": 443
                        }
                    ],
    ```

12. Ajouter des messages persistants à la file d'attente. La file d'attente définie dans le fichier `mqsc` spécifié dans le fichier ConfigMap avait l'attribut `DEFPSIST(YES)` , de sorte que tous les messages placés dans la file d'attente seront persistants et répliqués dans le groupe de reprise. Placez les messages dans la file d'attente à l'aide de 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**.

13. Effectuer un basculement géré du groupe Live à Londres vers le groupe Recovery à Rome pour démontrer que la configuration Native HA - CRR est correcte. Suivez les instructions présentées dans «Effectuer un basculement planifié sur une configuration Native HA CRR», à la page 243. Lorsque vous modifiez les rôles des groupes Live et Recovery, il peut être utile de mettre à jour `metadata.labels.role` dans le YAML pour refléter vos changements, mais cela n'est pas nécessaire pour le basculement. Une fois le basculement effectué, vous pouvez récupérer les messages du nouveau groupe en direct, à Rome.

    Pour plus d'informations sur les bascules et les commutations, voir «Basculement et basculement automatique en cas de panne de la HA CRR native», à la page 242.

14. Récupérez les messages envoyés avant le basculement en exécutant la commande suivante :

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

    Les messages que vous avez ajoutés à l'étape 11 sont disponibles et sont édités. Après quelques secondes, la commande prend fin.

Que faire ensuite

Lorsque la maintenance est terminée, vous pouvez revenir au groupe d'origine en répétant les étapes 12 et 13.

Vous pouvez également essayer un basculement non planifié en suivant les instructions données à l'adresse «Effectuer un basculement non planifié sur une configuration Native HA CRR», à la page 244.

Pour en savoir plus sur l'utilisation de différents certificats pour sécuriser les différentes parties de votre configuration CRR, voir «Exemple : utilisation de plusieurs certificats TLS avec Native HA CRR», à la page 134.

Exemple : Ajout d'un groupe de reprise à une configuration Native HA existante à l'aide de IBM MQ Operator

Cet exemple déploie un gestionnaire de files d'attente à l'aide d'une configuration Native HA simple sur le site OpenShift® Container Platform en utilisant l'opérateur IBM MQ . L'exemple ajoute un groupe de reprise, puis met à jour le gestionnaire de file d'attente Native HA pour en faire un groupe actif, étendant ainsi la configuration à une réplication Cross-Region (CRR) Native HA.

Avant de commencer

Pour réaliser cet exemple, vous devez d'abord suivre les étapes suivantes :

- Sur la ligne de commande, connectez-vous aux clusters OpenShift Container Platform (OCP) que vous utilisez.
- Créez deux projets/espaces de noms OCP pour cet exemple. L'exemple se réfère aux sites de Londres et de Rome. Ils renvoient soit à deux espaces de noms pour Londres et Rome dans une seule grappe, soit à un espace de noms dans chaque grappe pour chaque site.
- Assurez-vous que IBM MQ Operator 3.5.0 ou une version supérieure est installé dans les clusters et disponible dans les espaces de noms.
- Assurez-vous que l'outil de ligne de commande OpenSSL est installé.
- Complétez la tâche «Préparation de l'utilisation de 'Kubernetes par la création d'un secret d'extraction», à la page 88 .
- Lisez «Déploiement d'un gestionnaire de files d'attente simple à l'aide de IBM MQ Operator», à la page 100 afin de vous familiariser avec l'utilisation de l'opérateur IBM MQ .

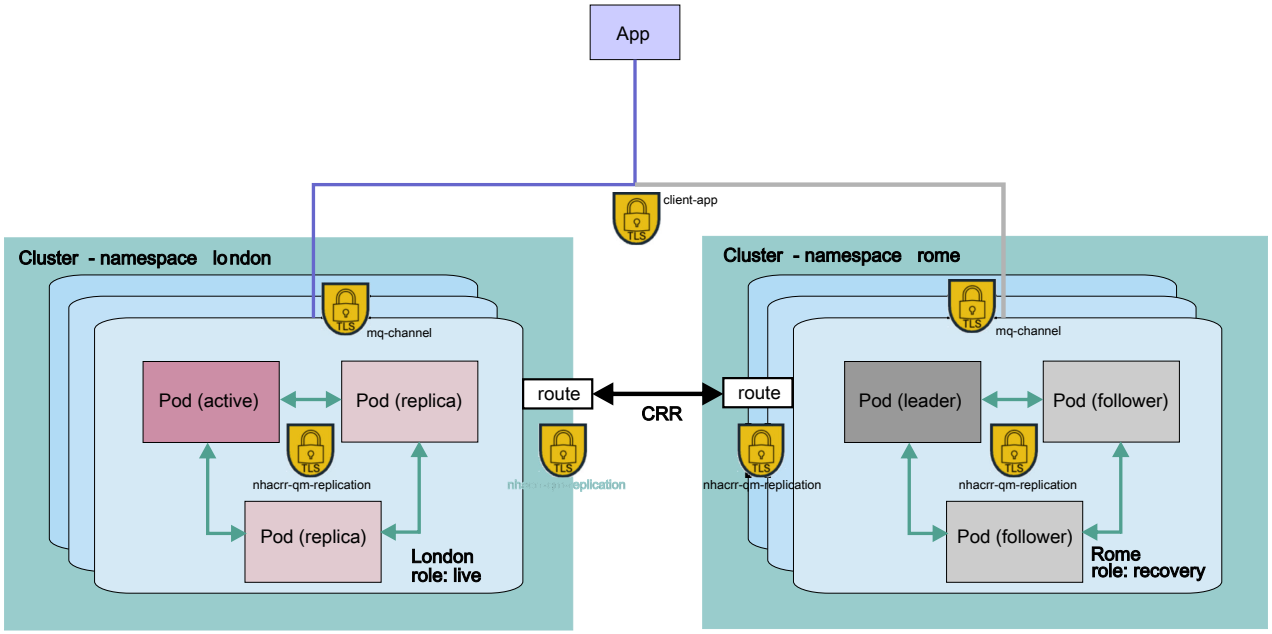
Pourquoi et quand exécuter cette tâche

Les groupes CRR sont généralement déployés dans deux clusters OCP situés à des endroits différents, mais cet exemple vous permet de déployer les deux groupes dans le même cluster si vous le souhaitez. Voir Native HA Cross-Region Replication pour une vue d'ensemble de Native HA CRR.

Pour des raisons de simplicité, cet exemple utilise un seul certificat pour sécuriser l'ensemble du trafic de réplication du gestionnaire de file d'attente, nommé nhacrr-qm-replication.

Pour comprendre comment les certificats sont utilisés dans Native HA CRR et utiliser différents certificats pour sécuriser le trafic des groupes et des instances, veuillez consulter «Exemple : utilisation de plusieurs certificats TLS avec Native HA CRR», à la page 134 Deploying Native HA CRR with multiple certificates.

Important : Ces exemples sont destinés à vous aider à démarrer, ils ne conviennent pas à un environnement de production. La gestion des certificats est un sujet complexe, réservé aux utilisateurs avancés. Pour la production, vous devez prendre en compte des éléments tels que la rotation, la révocation, la longueur des clés, la reprise après sinistre et bien d'autres encore.



Cet exemple fournit la ressource personnalisée YAML pour définir deux gestionnaires de file d'attente Native HA et un groupe de récupération utilisés dans la réplication interrégionale.

Procédure

1. Créer un gestionnaire de file d'attente Native HA sur le site de Londres. Assurez-vous que vous êtes dans l'espace de noms London sur le bon cluster et créez une configuration Native HA en suivant les instructions sur «Exemple: Configuration de Native HA à l'aide de IBM MQ Operator», à la page 116. (Vous créez les deux certificats nommés client-app et mq-channel dans le diagramme lorsque vous effectuez cette étape)
2. Créez les certificats à utiliser pour le trafic de réplication du gestionnaire de file d'attente dans la configuration CRR. Utilisez les commandes suivantes pour créer une clé privée et l'utiliser pour émettre un certificat auto-signé pour votre autorité de certification interne. Créez ensuite une clé privée et un certificat pour le gestionnaire de file d'attente, signés par votre autorité de certification interne.

   ```
   openssl genpkey -algorithm rsa -pkeyopt rsa_keygen_bits:4096 -out nhacrr-ca.key
   openssl req -x509 -new -nodes -key "nhacrr-ca.key" -sha512 -days 30 -subj "/CN=nhacrr-
   selfsigned-ca" -out nhacrr-ca.crt
   openssl req -new -nodes -out nhacrr-qm-replication.csr -newkey rsa:4096 -keyout nhacrr-qm-
   replication.key -subj "/CN=nhacrr-qm-replication"
   openssl x509 -req -in nhacrr-qm-replication.csr -CA nhacrr-ca.crt -CAkey nhacrr-ca.key
   -CAcreateserial -out nhacrr-qm-replication.crt -days 7 -sha512
   ```

   Pour plus d'informations sur ces commandes, voir «Création d'une infrastructure PKI autosignée à l'aide de OpenSSL», à la page 107.
3. Créez deux secrets Kubernetes contenant la clé et les certificats du gestionnaire de file d'attente. Créez un secret appelé nhacrr-qm-replication sur les sites de Rome et de Londres, afin de stocker la clé et le certificat des instances du gestionnaire de file d'attente. Utilisez la commande ci-dessous pour créer les secrets, passez à chaque espace de noms (Rome que Londres) avant d'exécuter la commande.

   ```
   kubectl create secret generic nhacrr-qm-replication --type="kubernetes.io/tls" --from-
   file=tls.key=nhacrr-qm-replication.key --from-file=tls.crt=nhacrr-qm-replication.crt --from-
   file=nhacrr-ca.crt
   ```
4. Mettez à jour le gestionnaire de files d'attente Native HA afin d'utiliser le nouveau secret pour le trafic répliqué. Pour mettre à jour le certificat utilisé pour le trafic de groupe interne, et plus tard pour le trafic de groupe à groupe, assurez-vous que vous êtes dans l'espace de noms de Londres,

puis modifiez l'attribut `.spec.queuemanager.availability.tls.secretname` dans le YAML QueueManager dans la console Web de l'OCP en :

```
      tls:
        secretName: nhacrr-qm-replication
```

5. Créer un secret dans le site de Rome, contenant le certificat et la clé générés précédemment à l'étape 1 de «Exemple: Configuration de Native HA à l'aide de IBM MQ Operator», à la page 116. Créez le secret en vous connectant au cluster, passez à l'espace de noms Rome et exécutez la commande suivante dans le répertoire où le fichier de certificat a été créé :

```
   kubectl 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
```

6. Créer une carte de configuration contenant des commandes MQSC et un fichier INI dans Rome. Dupliquer à Rome la carte de configuration créée sur le site de Londres. Assurez-vous d'être dans l'espace de noms Rome, entrez le YAML suivant dans la console web OCP, sélectionnez `ConfigMaps` puis `Create ConfigMap`, ou utilisez la commande `oc apply` .

```
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 n'autoriser l'accès qu'aux clients qui présentent un certificat avec un CN de example-app1 comme celui stocké dans l'exemple-qm-tls créé à l'étape 4.

7. Déployer le groupe Native HA Recovery. Créez un nouveau groupe de récupération sur le site de Rome, en utilisant la ressource personnalisée YAML suivante. Assurez-vous que vous êtes dans l'espace de noms Rome, puis entrez le YAML suivant dans la console web OCP pour créer le gestionnaire de files d'attente, ou à l'aide de la commande `oc apply` . Vérifiez que la licence spécifiée est correcte et acceptez-la en remplaçant false par true.

   Remarque : Vous devez utiliser le même nom de gestionnaire de file d'attente dans le groupe actif et le groupe de secours. Par défaut, si vous n'indiquez pas `.spec.queuemanager.name`, il s'agit de `.metadata.name`. Vous pouvez obtenir des noms correspondants en utilisant l'une des méthodes suivantes :

   - Nommez votre ressource gestionnaire de file d'attente de la même manière dans chaque cluster (c'est-à-dire `.metadata.name` correspond) et ne remplacez pas le nom.
   - Nommez votre gestionnaire de file d'attente individuellement dans `.metadata.name` et remplacez le nom dans `.spec.queuemanager.name` afin que les noms correspondent dans les deux sites.

   Dans cet exemple, le nom du gestionnaire de file d'attente est défini sur `exampleqm` en utilisant `.metadata.name`.

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

```
    role: Recovery
spec:
  license:
    accept: false
    license: L-NUUP-23NH8Y
    use: Production
  queueManager:
    name: EXAMPLEQM
    availability:
      type: NativeHA
      tls:
        secretName: nhacrr-qm-replication
      nativeHAGroups:
        local:
          name: "rome"
          role: "Recovery"
    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: 9.4.2.0-r1
  pki:
    keys:
      - name: default
        secret:
          secretName: example-qm-tls
          items:
            - tls.key
            - tls.crt
            - ca.crt
```

8. Vérifier que le groupe a été déployé correctement (cette commande doit être exécutée sur le site de Rome).
   a) Exécutez la commande `oc get qmgr exampleqm` pour afficher la ressource du gestionnaire de files d'attente et attendez que le gestionnaire de files d'attente soit en cours d'exécution :

```
NAME            PHASE
exampleqm   Running
```

   b) Lorsque le gestionnaire de file d'attente est en cours d'exécution, vous pouvez vérifier l'état des pods en utilisant `oc get pods`, un pod devrait être en état de préparation :

```
NAME                 READY   STATUS    RESTARTS   AGE
exampleqm-ibm-mq-0   1/1     Running   0          2m45s
exampleqm-ibm-mq-1   0/1     Running   0          2m45s
exampleqm-ibm-mq-2   0/1     Running   0          2m45s
```

   c) Vérifiez l'état du groupe de récupération en exécutant la commande suivante :

```
oc exec -t <leader-pod> -- dspmq -o nativeha -g
```

   Où `<leader-pod>` est le nom du pod prêt 1/1 (`exampleqm-ibm-mq-0` dans l'exemple) :

```
QMNAME(EXAMPLEQM)                                          ROLE(Leader)
INSTANCE(exampleqm-ibm-mq-0) INSYNC(yes) QUORUM(1/3) GRPLSN() GRPNAME(rome)
GRPROLE(Recovery)
 GRPNAME(rome) GRPROLE(Recovery) GRPADDR(Unknown) GRPVER(9.4.2.0) GRSTATUS(Waiting
for connection) RCOVLSN() RCOVTIME() BACKLOG(Unknown) INSYNC(Unknown) SYNCTIME()
ALTDATE(2025-04-25) ALTTIME(13.48.15)
```

   Les éléments clés de ce statut sont les suivants :

   - `Role(Leader)`
   - `QUORUM(1/3)` - cela reste 1/3 jusqu'à ce que les groupes Live et Recovery se connectent

- `GRSTATUS(Waiting for connection)` - il est mis à jour lorsque le groupe Live se connecte

9. Mettre à jour le gestionnaire de files d'attente Native HA pour utiliser le groupe Live.
   a) Récupérez l'adresse du groupe de récupération à l'aide de la commande suivante sur le site de Rome :

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

   Remarque : Deux routes sont créées lorsque les groupes CRR sont déployés, l'une se terminant par `qm` et l'autre par `nhacrr`. La route `qm` est utilisée pour envoyer des messages au gestionnaire de file d'attente, tandis que la route `nhacrr` est utilisée pour la communication de groupe à groupe. Assurez-vous d'ajouter les adresses des hôtes de la route `nhacrr` comme adresse distante pour le groupe dans le YAML ci-dessous.

   b) Remplacez l'adresse dans le fichier YAML ci-dessous par le nom d'hôte renvoyé par la commande, qui spécifie l'adresse utilisée par le groupe actif pour se connecter au groupe de reprise. Ajoutez le YAML suivant au YAML London QueueManager dans la console web de l'OCP, directement sous l'attribut `type: NativeHA` :

```
      nativeHAGroups:
        local:
          name: "london"
          role: "Live"
        remotes:
          - name: "rome"
            addresses:
              - "<rome group hostname>:443"
```

   Avant d'enregistrer le fichier YAML, ajoutez une étiquette "Live role" en haut des strophes de métadonnées :

```
metadata:
  labels:
    role: Live
```

10. Vérifiez le groupe Live et la réplication interrégionale.
   a) Sur le site de Londres, lancez la commande `oc get qmgr exampleqm` pour afficher la ressource du gestionnaire de files d'attente et attendez que le gestionnaire de files d'attente soit en cours d'exécution. Vous devez voir le message suivant :

```
NAME            PHASE
exampleqm   Running
```

   b) Lorsque le gestionnaire de files d'attente est en cours d'exécution, vérifiez l'état des pods à l'aide de la commande `oc get pods` . Un pod doit être prêt à l'emploi :

```
NAME                 READY   STATUS    RESTARTS   AGE
exampleqm-ibm-mq-0   1/1     Running   0          2m45s
exampleqm-ibm-mq-1   0/1     Running   0          2m45s
exampleqm-ibm-mq-2   0/1     Running   0          2m45s
```

   c) Vérifiez l'état du groupe en direct en émettant la commande suivante :

```
oc exec -t <active-pod> -- dspmq -o nativeha -g
```

   Où `<active-pod>` est le nom du pod prêt 1/1 (`exampleqm-ibm-mq-0` dans l'exemple) :

```
QMNAME(EXAMPLEQM)                                          ROLE(Active)
INSTANCE(exampleqm-ibm-mq-0) INSYNC(yes) QUORUM(3/3) GRPLSN(<0:0:13:21327>)
GRPNAME(london) GRPROLE(Live)
GRPNAME(london) GRPROLE(Live) GRPADDR(Unknown) GRPVER(9.4.3.0) GRSTATUS(Normal)
RCOVLSN(<0:0:13:21327>) RCOVTIME(2025-06-04T16:22:54.414645Z) INITLSN(<0:0:9:5720>)
INITTIME(2025-06-04T16:12:09.629132Z) LIVETIME(2025-06-04T16:12:11.825065Z)
ALTDATE(2025-06-04) ALTTIME(16.23.01)
GRPNAME(rome) GRPROLE(Recovery) GRPADDR(exampleqm-ibm-mq-nhacrr-
<namespace>.apps.<cluster>) GRPVER(9.4.3.0) CONNGRP(yes) GRSTATUS(Normal)
RCOVLSN(<0:0:13:21327>) RCOVTIME(2025-06-04T16:22:54.414645Z) BACKLOG(0) INSYNC(yes)
SYNCTIME(2025-06-04T16:29:17.206996Z) ALTDATE(2025-06-04) ALTTIME(16.29.12)
```

Les éléments clés de ce statut sont les suivants :

- `Role(Active)`
- `QUORUM(3/3)` - le groupe en direct dispose d'un quorum complet
- `GRSTATUS(Normal)`

d) Exécutez la même commande sur le site de Rome et observez que les adresses `QUORUM(3/3)` et `GRSTATUS(Normal)` ont également été mises à jour :

```
QMNAME(EXAMPLEQM)                                          ROLE(Leader)
INSTANCE(exampleqm-ibm-mq-0) INSYNC(yes) QUORUM(3/3) GRPLSN(<0:0:87:58322>)
GRPNAME(rome) GRPROLE(Recovery)
 GRPNAME(rome) GRPROLE(Recovery) GRPADDR(exampleqm-ibm-mq-nhacrr-
rome.apps.destructive-1.cp.fyre.ibm.com) GRPVER(9.4.3.0) GRSTATUS(Normal)
RCOVLSN(<0:0:87:58322>) RCOVTIME(2025-05-16T14:27:09.051799Z) BACKLOG(0) INSYNC(yes)
SYNCTIME(2025-05-16T14:30:40.134627Z) ALTDATE(2025-05-16) ALTTIME(14.27.17)
 GRPNAME(london) GRPROLE(Live) GRPADDR(Unknown) GRPVER(9.4.3.0) CONNGRP(yes)
GRSTATUS(Normal) RCOVLSN(<0:0:87:58322>) RCOVTIME(2025-05-16T14:27:09.051799Z)
INITLSN(<0:0:9:4696>) INITTIME(2025-05-12T16:58:13.582274Z)
LIVETIME(2025-05-12T16:58:16.319844Z) ALTDATE(2025-05-16) ALTTIME(14.30.41)
```

11. Ajouter une adresse distante au groupe Native HA Recovery. Lorsque Rome devient le groupe actif, il doit connaître l'adresse distante de Londres afin de pouvoir se connecter à Londres en tant que groupe de récupération. Pour l'ajouter, mettez à jour le site `spec.queueManager.availability.nativeHAGroups` pour le site de Rome, avec ce qui suit, en dessous de la strophe locale :

```
        remotes:
          - name: "london"
            addresses:
              - "<london group hostname>:443"
```

Remplacez <nom d'hôte du groupe Londres> par l'attribut d'hôte de la route `exampleqm-ibm-mq-nhacrr` sur le site de Londres. Assurez-vous d'être sur le site de Londres et utilisez la commande suivante pour récupérer le nom d'hôte de la route nhacrr :

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

Une fois la mise à jour effectuée, attendez que les pods redémarrent, qu'ils fonctionnent et qu'un pod soit le pod 1/1 prêt.

12. Ajoutez le nom d'hôte du site de Rome au fichier `ccdt.json` . Après un basculement, lorsque Rome devient le site principal, les clients doivent se connecter au groupe de Rome, et l'ajout du nom d'hôte de Rome au fichier `ccdt.json` permet de le faire.

a) Assurez-vous d'être sur le site de Rome et utilisez la commande suivante pour récupérer le nom d'hôte de la route qm :

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

Remarque : Il existe deux itinéraires, dans ce cas, utilisez l'itinéraire pour le trafic de messages. Ajoutez les adresses des hôtes de la route `qm` à la route `ccdt.json`.

b) Dans le fichier `ccdt.json` , ajoutez une deuxième paire de noms d'hôtes et de ports à la liste des connexions, de sorte que votre liste de connexions ressemble maintenant à ceci :

```
                "connection":
                [
                    {
                        "host": "<hostname from London qm route>",
                        "port": 443
                    },
                    {
                        "host": "<hostname from Rome qm route>",
                        "port": 443
                    }
                ],
```

13. Effectuez un basculement géré de Londres à Rome pour démontrer que la configuration Native HA - CRR est correcte, puis effectuez un basculement géré entre le groupe Live à Londres et le groupe

Recovery à Rome. Suivez les instructions présentées dans «Effectuer un basculement planifié sur une configuration Native HA CRR», à la page 243. Lorsque vous mettez à jour les rôles des groupes Live et Recovery, il peut être utile de mettre à jour `metadata.labels.role` dans le YAML pour refléter le changement, mais cela n'est pas nécessaire pour le basculement. Une fois le basculement effectué, vous pouvez récupérer les messages du nouveau groupe en direct, à Rome.

Pour plus d'informations sur les bascules et les commutations, voir «Basculement et basculement automatique en cas de panne de la HA CRR native», à la page 242.

Que faire ensuite

Lorsque la maintenance est terminée, vous pouvez revenir au groupe Live d'origine en répétant les étapes 12 et 13.

Vous pouvez également essayer un basculement non planifié en suivant les instructions données à l'adresse «Effectuer un basculement non planifié sur une configuration Native HA CRR», à la page 244.

Enfin, pour en savoir plus sur l'utilisation de différents certificats pour sécuriser les différentes parties de votre configuration CRR, voir Configuration de TLS pour Native HA CRR.

Exemple : utilisation de plusieurs certificats TLS avec Native HA CRR

Avant de commencer

Avant de commencer cet exemple, vous devez d'abord terminer le site «Exemple : Déploiement d'une configuration simple de CRR Native HA à l'aide de l'opérateur IBM MQ», à la page 121 (cet exemple suppose que le déploiement existe et que le groupe de Londres est le groupe actif).

Suivez ensuite les étapes suivantes :

- Assurez-vous que vous pouvez vous connecter au(x) cluster(s) OpenShift Container Platform (OCP) que vous utilisez, et sélectionnez le projet/l'espace de noms approprié. (L'exemple décrit à l'adresse «Exemple : Déploiement d'une configuration simple de CRR Native HA à l'aide de l'opérateur IBM MQ», à la page 121 a créé deux projets/espaces de noms OpenShift Container Platform (OCP) soit dans deux espaces de noms pour Londres et Rome dans un seul cluster, soit dans un espace de noms dans chaque cluster pour chaque site. Cet exemple se réfère également aux sites de Londres et de Rome)
- Assurez-vous que IBM MQ Operator 3.5.0 ou une version supérieure est installé dans les clusters et disponible dans les espaces de noms.
- Assurez-vous que l'outil de ligne de commande OpenSSL est installé.
- Complétez la tâche «Préparation de l'utilisation de 'Kubernetes par la création d'un secret d'extraction», à la page 88 .
- Lisez «Déploiement d'un gestionnaire de files d'attente simple à l'aide de IBM MQ Operator», à la page 100 afin de vous familiariser avec l'utilisation de l'opérateur IBM MQ .

Pourquoi et quand exécuter cette tâche

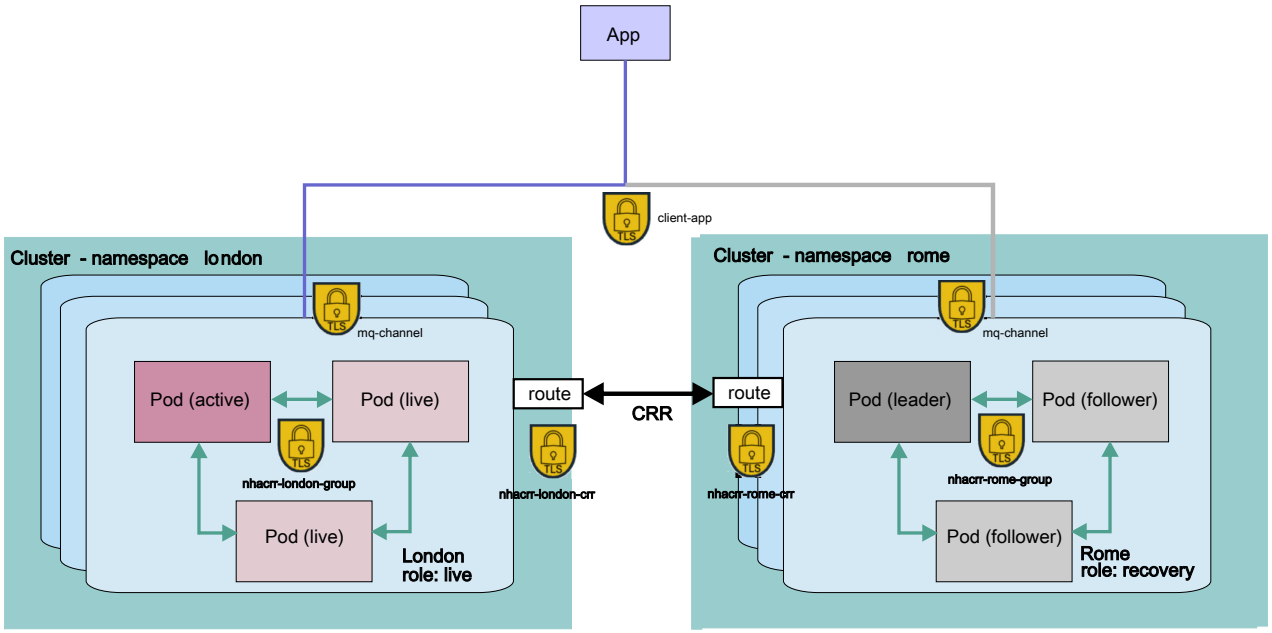
Cet exemple commence par une présentation de la façon dont TLS et les certificats sont utilisés dans une configuration Native HA CRR, puis utilise des étapes simples pour démontrer comment le trafic `nhacrr-london-group`, `nhacrr-rome-group`, `nhacrr-london-crr` et `nhacrr-rome-crr` est sécurisé.

Les étapes de la section de procédure ci-dessous vous expliquent comment modifier le déploiement de «Exemple : Déploiement d'une configuration simple de CRR Native HA à l'aide de l'opérateur IBM MQ», à la page 121 afin d'utiliser des certificats différents pour chaque type de trafic et de contrôler quel CipherSpec est utilisé.

Configuration de TLS pour le CRR HA natif

Le diagramme suivant montre où les certificats sont utilisés dans la configuration du CRR Native HA. Les certificats sont utilisés comme suit :

1. Pour sécuriser la communication entre le gestionnaire de file d'attente et les applications clientes - `client-app` et `mq-channel`
2. Pour sécuriser le trafic de réplication du gestionnaire de file d'attente - `nhacrr-london-group`, `nhacrr-rome-group`, `nhacrr-london-crr` et `nhacrr-rome-crr`



Communication sécurisée entre le gestionnaire de file d'attente et les applications

Les deux groupes Native HA, illustrés dans le diagramme, forment ensemble un gestionnaire de file d'attente hautement disponible, auquel les applications clientes peuvent se connecter via le groupe en direct. Lors d'un basculement, les applications sont déconnectées et doivent se reconnecter au nouveau groupe actif. (L'ajout des détails de connexion des deux groupes à la liste de connexion d'une application facilite la reconnexion)

Si les applications qui se connectent via une route ne font pas partie de l'un des groupes ou des deux, ou se trouvent dans un cluster différent, elles doivent communiquer via TLS à l'aide de clés privées et de certificats. Dans le diagramme, le certificat et la clé du client sont étiquetés `client-app` et ceux du gestionnaire de file d'attente sont étiquetés `mq-channel`. Le certificat et la clé privée de `mq-channel` doivent être ajoutés comme deux secrets, un pour chaque groupe.

«Exemple : Déploiement d'une configuration simple de CRR Native HA à l'aide de l'opérateur IBM MQ», à la page 121 vous guide dans la création de ces certificats, clés et secrets. Les étapes suivantes se concentrent sur les certificats pour le trafic de réplication du gestionnaire de file d'attente et ne modifient pas les certificats `mq-channel` ou `client-app` .

Sécuriser le trafic de réplication du gestionnaire de file d'attente

Le trafic est répliqué à la fois au sein des groupes et entre eux. Le trafic réseau entre les deux groupes doit toujours être sécurisé, mais le trafic à l'intérieur d'un groupe peut être en texte clair, car le trafic à l'intérieur du groupe n'est généralement pas accessible.

1. `nhacrr-london-group` et `nhacrr-rome-group` représentent les certificats et les clés privées utilisés pour sécuriser le trafic au sein de chaque groupe.
2. `nhacrr-london-crr` et `nhacrr-rome-crr` représentent les certificats et les clés privées utilisés pour sécuriser chaque extrémité du trafic intergroupe.

Dans l' exemple : Déploiement d'une configuration CRR Native HA simple utilisant l'opérateur IBM MQ, le même certificat et la même clé privée, stockés dans un secret `nhacrr-qm-replication` , dans chaque espace de noms, sont utilisés pour sécuriser le trafic `nhacrr-london-group`, `nhacrr-rome-group`, `nhacrr-london-crr` et `nhacrr-rome-crr` .

Important : Ces exemples sont destinés à vous aider à démarrer, ils ne conviennent pas à un environnement de production. La gestion des certificats est un sujet complexe, réservé aux utilisateurs avancés. Pour la production, vous devez prendre en compte des éléments tels que la rotation, la révocation, la longueur des clés, la reprise après sinistre et bien d'autres encore.

Cet exemple montre comment utiliser des certificats différents pour chaque type de trafic.

Procédure

1. Créer quatre jeux de clés et de certificats à utiliser pour le trafic de réplication du gestionnaire de file d'attente. Pour les créer dans le cadre de cet exemple, exécutez les commandes suivantes. Chaque série de commandes crée une clé privée, l'utilise pour émettre un certificat auto-signé pour une autorité de certification interne, puis crée une clé privée et un certificat signé avec l'autorité de certification interne. Cet exercice est répété pour créer des certificats pour les trafics `nhacrr-london-group`, `nhacrr-rome-group`, `nhacrr-london-crr` et `nhacrr-rome-crr` .

   Pour une explication plus détaillée des commandes, voir «Création d'une infrastructure PKI autosignée à l'aide de OpenSSL», à la page 107.

   Remarque : Cet exemple crée des autorités de certification distinctes pour chaque chaîne de certificats, afin de permettre à l'exemple de démontrer la mise en place d'une confiance entre les groupes, si nécessaire. Les certificats d'une même organisation utilisent souvent une autorité de certification commune. Dans ce cas, les strophes de confiance du Queue Manager YAML ne sont pas nécessaires, car l'autorité de certification est déjà approuvée.

```
openssl genpkey -algorithm rsa -pkeyopt rsa_keygen_bits:4096 -out nhacrr-london-group-
ca.key openssl req -x509 -new -nodes -key
"nhacrr-london-group-ca.key" -sha512 -days 30 -subj "/CN=nhacrr-london-group-selfsigned-ca"
-out nhacrr-london-group-ca.crt openssl
req -new -nodes -out nhacrr-london-group.csr -newkey rsa:4096 -keyout nhacrr-london-
group.key -subj "/CN=nhacrr-london-group"
openssl x509 -req -in nhacrr-london-group.csr -CA nhacrr-london-group-ca.crt -CAkey nhacrr-
london-group-ca.key -CAcreateserial -out
nhacrr-london-group.crt -days 7 -sha512
```

```
openssl genpkey -algorithm rsa -pkeyopt rsa_keygen_bits:4096 -out nhacrr-rome-group-ca.key
openssl req -x509 -new -nodes -key
"nhacrr-rome-group-ca.key" -sha512 -days 30 -subj "/CN=nhacrr-rome-group-selfsigned-ca"
-out nhacrr-rome-group-ca.crt openssl req -new
-nodes -out nhacrr-rome-group.csr -newkey rsa:4096 -keyout nhacrr-rome-group.key -subj "/
CN=nhacrr-rome-group" openssl x509
-req -in nhacrr-rome-group.csr -CA nhacrr-rome-group-ca.crt -CAkey nhacrr-rome-group-ca.key
-CAcreateserial -out nhacrr-rome-group.crt
-days 7 -sha512
```

```
openssl genpkey -algorithm rsa -pkeyopt rsa_keygen_bits:4096 -out nhacrr-london-crr-ca.key
openssl req -x509 -new -nodes -key
"nhacrr-london-crr-ca.key" -sha512 -days 30 -subj "/CN=nhacrr-london-crr-selfsigned-ca"
-out nhacrr-london-crr-ca.crt openssl req -new -nodes
-out nhacrr-london-crr.csr -newkey rsa:4096 -keyout nhacrr-london-crr.key -subj "/CN=nhacrr-
london-crr" openssl x509 -req -in
nhacrr-london-crr.csr -CA nhacrr-london-crr-ca.crt -CAkey nhacrr-london-crr-ca.key
-CAcreateserial -out nhacrr-london-crr.crt -days 7 -sha512
```

```
openssl genpkey -algorithm rsa -pkeyopt rsa_keygen_bits:4096 -out nhacrr-rome-crr-ca.key
openssl req -x509 -new -nodes -key
"nhacrr-rome-crr-ca.key" -sha512 -days 30 -subj "/CN=nhacrr-rome-crr-selfsigned-ca" -out
nhacrr-rome-crr-ca.crt openssl req -new -nodes -out
nhacrr-rome-crr.csr -newkey rsa:4096 -keyout nhacrr-rome-crr.key -subj "/CN=nhacrr-rome-
crr" openssl x509 -req -in nhacrr-rome-crr.csr
-CA nhacrr-rome-crr-ca.crt -CAkey nhacrr-rome-crr-ca.key -CAcreateserial -out nhacrr-rome-
crr.crt -days 7 -sha512
```

2. Créez des secrets Kubernetes contenant les clés et les certificats pour Rome. Utilisez les commandes suivantes pour créer deux secrets TLS sur le site de Rome :

   a. Un certificat pour sécuriser le trafic interne du groupe Rome.

b. Un certificat pour sécuriser le trafic entre les groupes du groupe de Rome.

```
kubectl create secret generic nhacrr-rome-group --type="kubernetes.io/tls" --
from-file=tls.key=nhacrr-rome-group.key --from-file=tls.crt=nhacrr-rome-group.crt --from-
file=nhacrr-rome-group-ca.crt
kubectl create secret generic nhacrr-rome-crr --type="kubernetes.io/tls" --from-
file=tls.key=nhacrr-rome-crr.key --from-file=tls.crt=nhacrr-rome-crr.crt --from-file=nhacrr-
rome-crr-ca.crt
```

Utilisez maintenant les commandes suivantes pour créer deux secrets contenant les certificats Rome auxquels le groupe de Londres doit faire confiance :

```
kubectl create secret generic nhacrr-london-group-trust --from-file=tls.crt=nhacrr-london-
group.crt --from-file=ca.crt=nhacrr-london-group-ca.crt --type=opaque
kubectl create secret generic nhacrr-london-crr-trust --from-file=tls.crt=nhacrr-london-
crr.crt  --from-file=ca.crt=nhacrr-london-crr-ca.crt --type=opaque
```

3. Créez des secrets Kubernetes contenant les clés et les certificats pour Londres. Utilisez les commandes suivantes pour créer deux secrets TLS sur le site de Londres :

   a. Un certificat pour sécuriser le trafic interne du groupe de Londres.

   b. Un certificat pour sécuriser le trafic entre les groupes à partir du groupe de Londres.

```
kubectl create secret generic nhacrr-london-group --type="kubernetes.io/tls" --from-
file=tls.key=nhacrr-london-group.key --from-file=tls.crt=nhacrr-london-group.crt --from-
file=nhacrr-london-group-ca.crt
kubectl create secret generic nhacrr-london-crr --type="kubernetes.io/tls" --
from-file=tls.key=nhacrr-london-crr.key --from-file=tls.crt=nhacrr-london-crr.crt --from-
file=nhacrr-london-crr-ca.crt
```

Utilisez maintenant les commandes suivantes pour créer deux secrets contenant les certificats de Londres auxquels le groupe de Rome doit faire confiance :

```
kubectl create secret generic nhacrr-rome-group-trust --from-file=tls.crt=nhacrr-rome-
group.crt --from-file=ca.crt=nhacrr-rome-group-ca.crt --type=opaque
kubectl create secret generic nhacrr-rome-crr-trust --from-file=tls.crt=nhacrr-rome-
crr.crt  --from-file=ca.crt=nhacrr-rome-crr-ca.crt --type=opaque
```

4. Sécuriser le trafic sur les sites de Rome et de Londres avec des certificats différents. Le déploiement créé à l'adresse «Exemple : Déploiement d'une configuration simple de CRR Native HA à l'aide de l'opérateur IBM MQ», à la page 121 a sécurisé tout le trafic de réplication avec le secret `nhacrr-qm-replication` . Cette étape modifie la configuration pour sécuriser le trafic au sein de chaque groupe avec les secrets `nhacrr-rome-group` et `nhacrr-london-group` respectivement.

   L'attribut `.spec.queuemanager.availability.tls.secretName` spécifie le secret à utiliser pour sécuriser le trafic au sein du groupe du site. Lorsqu'aucun autre attribut TLS n'est défini, le secret est également utilisé pour sécuriser le trafic entre les groupes.

   a) Sur le site de Rome, modifiez l'attribut `.spec.queuemanager.availability.tls.secretName` dans le fichier YAML QueueManager de la console Web de l'OCP et remplacez-le par nhacrr-rome-group :

```
      tls:
        secretName: nhacrr-rome-group
```

   b) Ajoutez une strophe `spec.queueManager.availability.NativeHAGroups.remotes.trust` au YAML et ajoutez le secret `nhacrr-london-group-trust` comme celui auquel il faut faire confiance :

```
...
      nativeHAGroups:
        local:
          name: "rome"
          role: "Recovery"
        remotes:
          - name: "london"
            addresses:
              - "exampleqm-ibm-mq-nhacrr-london.apps.trouble.cp.fyre.ibm.com:443"
            trust:
```

```
              - secret:
                  items:
                    - tls.crt
                    - ca.crt
                  secretName: nhacrr-london-group-trust
...
```

Le secret `nhacrr-london-group-trust` contient le certificat et l'autorité de certification stockés dans le secret `nhacrr-london-group` sur le site de Londres. Le secret `nhacrr-london-group` est utilisé pour sécuriser le trafic du groupe de Londres. Le groupe de Rome utilise donc ce secret pour confirmer que le certificat fourni est fiable.

c) Effectuez les étapes équivalentes sur le site de Londres. Modifier l'attribut `.spec.queuemanager.availability.tls.secretName` en :

```
      tls:
        secretName: nhacrr-london-group
```

d) Ajouter le secret `nhacrr-rome-group-trust` à une strophe `spec.queueManager.availability.NativeHAGroups.remotes.trust` :

```
...
      nativeHAGroups:
        local:
          name: "london"
          role: "Live"
        remotes:
          - name: "rome"
            addresses:
              - "<rome group hostname>:443"
            trust:
              - secret:
                  items:
                    - tls.crt
                    - ca.crt
                  secretName: nhacrr-rome-group-trust
...
```

5. Vérifiez la réplication interrégionale avec les nouveaux certificats. Sur le site de Londres, lancez la commande `oc get qmgr exampleqm` pour afficher la ressource du gestionnaire de files d'attente et attendez que le gestionnaire de files d'attente soit en cours d'exécution. Vérifiez ensuite l'état des pods à l'aide de la commande `oc get pods` . Un pod doit être prêt, c'est le pod actif.

   Vous pouvez ensuite lancer la commande `oc exec -t <active-pod> -- dspmq -o nativeha -g` pour vérifier l'état du groupe en direct. Remplacez `<active-pod>` par le nom du pod prêt 1/1 renvoyé par la commande `oc get pods` .

```
QMNAME(EXAMPLEQM)                                          ROLE(Active) INSTANCE(exampleqm-
ibm-mq-0) INSYNC(yes) QUORUM(3/3) GRPLSN(<0:0:87:58322>) GRPNAME(london) GRPROLE(Live)
 GRPNAME(london) GRPROLE(Live) GRPADDR(Unknown) GRPVER(9.4.3.0) GRSTATUS(Normal)
RCOVLSN(<0:0:87:58322>) RCOVTIME(2025-05-16T14:27:09.051799Z) INITLSN(<0:0:9:4696>)
INITTIME(2025-05-12T16:58:13.582274Z) LIVETIME(2025-05-12T16:58:16.319844Z)
ALTDATE(2025-05-16) ALTTIME(14.27.10)
 GRPNAME(rome) GRPROLE(Recovery) GRPADDR(exampleqm-ibm-mq-nhacrr-
<namespace>.apps.<cluster>) GRPVER(9.4.3.0) CONNGRP(yes) GRSTATUS(Normal)
RCOVLSN(<0:0:87:58322>) RCOVTIME(2025-05-16T14:27:09.051799Z) BACKLOG(0) INSYNC(yes)
SYNCTIME(2025-05-16T14:28:38.084565Z) ALTDATE(2025-05-16) ALTTIME(14.28.31)
```

   Les éléments clés de ce statut sont les suivants :

   - `Role(Active)`
   - `QUORUM(3/3)` - le groupe en direct dispose d'un quorum complet
   - `GRSTATUS(Normal)`

6. Sécuriser le trafic entre les groupes à l'aide de certificats distincts. Vous pouvez utiliser des certificats distincts pour le trafic à l'intérieur du groupe et pour le trafic entre les groupes. Dans cette étape, les certificats des secrets `nhacrr-rome-crr` et `nhacrr-london-crr` sont utilisés pour sécuriser le trafic entre les groupes, tout en laissant les secrets `nhacrr-rome-group` et `nhacrr-london-group` sécuriser le trafic dans chaque groupe. Pour ce faire, deux modifications

sont nécessaires sur chaque site, l'une pour spécifier le secret pour le trafic entre les groupes et l'autre pour modifier les trusts afin qu'ils correspondent aux nouveaux certificats utilisés.

Remarque : Si les secrets utilisés pour les attributs `.spec.queuemanager.availability.NativeHAGroups.local.tls.secretName` et `.spec.queuemanager.availability.tls.secretName` contiennent les mêmes certificats, il y a conflit et les instances du gestionnaire de file d'attente ne peuvent pas démarrer. Si vous souhaitez utiliser un seul certificat, définissez uniquement `.spec.queuemanager.availability.tls.secretName` .

a) Sur le site de Rome, ajoutez l'attributspec.queuemanager.availability.NativeHAGroups.local.tls.key.secretName dans le YAML QueueManager dans la console web OCP et définissez-le à `nhacrr-rome-crr`. Ensuite, changez le `.spec.queueManager.availability.NativeHAGroups.remotes.trust.secret.secretName` en `nhacrr-london-crr-trust`:

```
...
      nativeHAGroups:
        local:
          name: "rome"
          role: "Recovery"
          tls:
            key:
              secretName: nhacrr-rome-crr
        remotes:
          - name: "london"
            addresses:
              - "<london group hostname>:443"
            trust:
              - secret:
                  items:
                    - tls.crt
                    - ca.crt
                  secretName: nhacrr-london-crr-trust
...
```

b) Effectuez les étapes équivalentes sur le site de Londres. Ajoutez l'attribut `.spec.queuemanager.availability.NativeHAGroups.local.tls.key.secretName` dans le YAML QueueManager dans la console web de l'OCP et définissez-le à `nhacrr-london-crr`. Ensuite, changez le `.spec.queueManager.availability.NativeHAGroups.remotes.trust.secret.secretName` en `nhacrr-rome-crr-trust`.

```
...
      nativeHAGroups:
        local:
          name: "london"
          role: "Live"
          tls:
            key:
              secretName: nhacrr-london-crr
        remotes:
          - name: "london"
            addresses:
              - "<london group hostname>:443"
            trust:
              - secret:
                  items:
                    - tls.crt
                    - ca.crt
                  secretName: nhacrr-rome-crr-trust
...
```

7. Vérifiez la réplication interrégionale avec les nouveaux certificats. Dans le site de Londres, lancez la commande `oc get qmgr exampleqm` pour afficher la ressource du gestionnaire de files d'attente et attendez que le gestionnaire de files d'attente soit en cours d'exécution. Vérifiez ensuite l'état des pods à l'aide de `oc get pods`, un pod doit être prêt, c'est le pod actif.

Emettez la commande `oc exec -t <active-pod> -- dspmq -o nativeha -g` pour vérifier l'état du groupe en direct. Remplacez `<active-pod>` par le nom du pod prêt 1/1 renvoyé par la commande `oc get pods` .

```
QMNAME(EXAMPLEQM)                                          ROLE(Active) INSTANCE(exampleqm-
ibm-mq-0) INSYNC(yes) QUORUM(3/3) GRPLSN(<0:0:87:58322>) GRPNAME(london) GRPROLE(Live)
 GRPNAME(london) GRPROLE(Live) GRPADDR(Unknown) GRPVER(9.4.3.0) GRSTATUS(Normal)
RCOVLSN(<0:0:87:58322>) RCOVTIME(2025-05-16T14:27:09.051799Z) INITLSN(<0:0:9:4696>)
INITTIME(2025-05-12T16:58:13.582274Z) LIVETIME(2025-05-12T16:58:16.319844Z)
ALTDATE(2025-05-16) ALTTIME(14.27.10)
 GRPNAME(rome) GRPROLE(Recovery) GRPADDR(exampleqm-ibm-mq-nhacrr-
<namespace>.apps.<cluster>) GRPVER(9.4.3.0) CONNGRP(yes) GRSTATUS(Normal)
RCOVLSN(<0:0:87:58322>) RCOVTIME(2025-05-16T14:27:09.051799Z) BACKLOG(0) INSYNC(yes)
SYNCTIME(2025-05-16T14:28:38.084565Z) ALTDATE(2025-05-16) ALTTIME(14.28.31)
```

Les éléments clés de ce statut sont les suivants :

- `Role(Active)`
- `QUORUM(3/3)` - le groupe en direct dispose d'un quorum complet
- `GRSTATUS(Normal)`

Enfin, si vous lancez la commande `kubectl logs <active-pod>` , vous devriez voir les messages suivants dans le journal, confirmant que les bons certificats sont utilisés pour les différents types de trafic. Pour le trafic inter-groupes, vous devriez voir :

```
AMQ3307I: The Native HA secure connection between the local group and 'rome' is using
certificate DN 'CN=nhacrr-london-crr'
```

Et pour le trafic du groupe interne, vous devriez voir les messages suivants, où `<replica-pod>` est le nom de l'un des pods avec le statut ready 0/1 :

```
AMQ3306I: The Native HA secure connection between the local instance and '<replica-pod>'
accepted remote certificate DN 'CN=nhacrr-london-group'
```

8. Ajouter des adresses CipherSpecs spécifiques pour le trafic répliqué.

   Si aucun CipherSpecs spécifique n'est défini pour le trafic répliqué, ANY_TLS13_OR_HIGHER est utilisé pour que les protocoles et le cryptage soient négociés entre les instances. Pour plus d'informations sur le site CipherSpecs pris en charge par IBM MQ , consultez le site CipherSpecs.

   Pour spécifier un CipherSpec pour le trafic au sein des groupes et entre les groupes, l'attribut `.spec.queuemanager.availability.tls.cipherSpec` est ajouté au YAML QueueManager.

   Ajoutez le YAML ci-dessous au YAML QueueManager dans la console web de l'OCP sur les sites de Rome et de Londres afin d'utiliser le TLS_AES_256_GCM_SHA384 CipherSpec pour tout le trafic de réplication :

```
      tls:
        secretName: nhacrr-rome-group
        cipherSpec: "TLS_AES_256_GCM_SHA384"
```

   Si vous devez spécifier une adresse CipherSpec distincte pour le trafic entre les groupes, l'attribut `.spec.queuemanager.availability.NativeHAGroups.local.tls.cipherSpec` peut être ajouté au fichier YAML QueueManager.

   Ajoutez le YAML ci-dessous au YAML QueueManager dans la console web de l'OCP pour les sites de Rome et de Londres afin d'utiliser le TLS_AES_128_GCM_SHA256 CipherSpec pour le trafic entre les groupes et le TLS_AES_256_GCM_SHA384 CipherSpec pour le trafic à l'intérieur des groupes :

```
      nativeHAGroups:
        local:
       ...
          tls:
            key:
              secretName: nhacrr-rome-crr
            cipherSpec: "TLS_AES_128_GCM_SHA256"
```

Après le redémarrage des instances, lancez la commande suivante sur le site de Londres :

```
kubectl logs
        <active-pod>
```

Vous devriez voir apparaître dans le journal des messages similaires à l'exemple ci-dessous. L'exemple montre que la communication entre les instances du groupe se fait par le biais de CipherSpec TLS_AES_256_GCM_SHA384:

```
[CommentInsert1(exampleqm-ibm-mq-2), CommentInsert2(exampleqm-ibm-mq-replica-2(9414)),
        CommentInsert3(TLS_AES_256_GCM_SHA384)]
```

Cet exemple montre que la connexion au groupe Rome utilise CipherSpec TLS_AES_128_GCM_SHA256:

```
AMQ3253I: A Native HA recovery group connection to 'rome' has
        been established using the 'TLS_AES_128_GCM_SHA256' CipherSpec.
```

9. Autoriser le trafic de texte en clair au sein d'un groupe. Le trafic entre les groupes doit toujours être sécurisé et crypté, mais le trafic au sein d'un groupe peut être non crypté. Définissez l'adresse `.spec.queuemanager.availability.tls.cipherSpec` sur blank dans le site de Londres pour supprimer le cryptage du trafic interne du groupe de Londres :

```
      tls:
        cipherSpec: ''
        secretName: nhacrr-london-group
```

Après le redémarrage des instances, lancez la commande suivante sur le site de Londres :

```
kubectl logs
        <active-pod>
```

Le message suivant apparaît dans le journal :

`AMQ9722W: Plain text communication is enabled.`

10. Désactiver la sécurité du trafic au sein d'un groupe. La suppression de la strophe `.spec.queuemanager.availability.tls` désactive complètement la sécurité pour le trafic interne d'un groupe.

Remarque : Le trafic entre les groupes doit être sécurisé, de sorte que si le site `.spec.queuemanager.availability.tls.secretName` est supprimé, le site `.spec.queuemanager.availability.NativeHAGroups.local.tls.key.secretName` doit être spécifié.

Désactivez la sécurité pour le groupe Rome en supprimant la strophe `.spec.queuemanager.availability.tls` du YAML QueueManager dans la console web OCP sur le site Rome :

```
      tls:
        secretName: nhacrr-rome-group
        cipherSpec: "TLS_AES_256_GCM_SHA384"
```

Après le redémarrage des instances, lancez la commande `kubectl logs <active-pod>` sur le site de Rome pour voir des messages similaires à ceux-ci dans le journal :

```
AMQ3213I: Native HA inbound connection accepted from 'exampleqm-ibm-mq-2'.
[CommentInsert1(exampleqm-ibm-mq-2), CommentInsert2(<ip>)]
AMQ3211I: Native HA outbound connection established to 'exampleqm-ibm-mq-2'.
[CommentInsert1(exampleqm-ibm-mq-2), CommentInsert2(<replica>(9414))]
```

Au lieu de :

```
AMQ3214I: Native HA inbound secure connection accepted from 'exampleqm-ibm-mq-0'.
[CommentInsert1(exampleqm-ibm-mq-0), CommentInsert2(<ip>), CommentInsert3(<cipherspec>)]
AMQ3212I: Native HA outbound secure connection established to 'exampleqm-
```

```
ibm-mq-0'. [CommentInsert1(exampleqm-ibm-mq-0), CommentInsert2(<replica>(9414)),
CommentInsert3(<cipherspec)
```

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, les informations provenant des instances répliques peuvent être obsolètes.

V 9.4.2 Pour une configuration Native HA Cross-Region Replication (CRR), les informations fournies dépendent également du fait que l'instance fait partie d'un groupe Live ou d'un groupe Recovery. Les informations provenant du groupe Live sont plus définitives, mais peuvent toujours être périmées car le groupe Recovery ne renvoie que des informations périodiques.

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.
- V 9.4.2 Afficher le statut des groupes auxquels l'instance appartient ou auxquels elle est connectée.

Les champs d'état suivants sont utilisés pour signaler l'état des configurations Native HA et Native HA Cross-Region Replication (CRR) d'une instance de gestionnaire de file d'attente. Ces champs apparaissent toujours :

ROLE
: Spécifie le rôle actuel de l'instance. Est l'un des sites `Active`, `Replica`, `Unknown`, (pour un groupe de récupération) `Leader`, ou `Not Configured`.

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*.

V 9.4.2 **GRPLSN**
: Spécifie le numéro de séquence du journal (LSN) qui a été répliqué entre un quorum d'instances dans le groupe HA, au format « *nnnnn : nnnnn : nnnnn : nnnnn* ». Blanc si le LSN n'est pas connu.

V 9.4.2 **GRPNAME**
: Spécifie le nom du groupe HA. Blanc si aucun nom n'a été configuré.

V 9.4.2 **GRPROLE**
: Spécifie le rôle actuel du groupe HA et est l'un des suivants :

  - `Live` - Un groupe vivant
  - `Recovery` - Un groupe de rétablissement

- `Pending live` - Un groupe en devenir `Live`
- `Pending recovery` - Un groupe en devenir `Recovery`
- `Not configured` - Native HA n'a pas été configuré
- `Unknown` - Le rôle actuel ne peut être déterminé

Ces champs apparaissent lorsque des informations complémentaires sont demandées pour toutes les instances en utilisant le paramètre `-x` :

REPLADDR
Adresse de réplication de l'instance de gestionnaire de files d'attente.

CONNACTV
Indique si l'instance est connectée à l'instance active.

BACKLOG
Indique le nombre de Ko que l'instance est en retard par rapport à l'instance active.

CONNINST
Indique si l'instance désignée est connectée à cette instance.

ALTDATE
Indique la date de la dernière mise à jour de ces informations. Blanc s'il n'a jamais été mis à jour.

ALTTIME
Indique l'heure à laquelle cette information a été mise à jour pour la dernière fois. Blanc s'il n'a jamais été mis à jour.

V 9.4.2 **ACKLSN**
Indique le LSN que l'instance a reconnu comme ayant été écrit dans le journal de récupération, au format *nnnnn:nnnnn:nnnnn:nnnnn*. Blanc si la valeur du LSN n'est pas connue.

V 9.4.2 **HAStatus**

Indique l'état opérationnel de cette instance et correspond à l'une des valeurs suivantes :

- `Normal` - l'instance fonctionne normalement.
- `Checking` - l'instance est vérifiée pour assurer la cohérence du journal de récupération.
- `Synchronizing` - l'instance reçoit des données pour la synchroniser.
- `Rebasing` - un nouveau journal de récupération est envoyé à l'instance.
- `Disk full` - l'instance a un disque plein.
- `Disconnected` - ne peut pas communiquer avec l'instance.
- `Unknown` - le statut de l'instance est inconnu.

V 9.4.2 **SYNCTIME**
Indique l'heure à laquelle cette instance a été synchronisée pour la dernière fois avec le gestionnaire de file d'attente actif, au format ISO 8601. Blanc si l'heure n'est pas connue.

V 9.4.2 Ces champs apparaissent lorsque des informations sur le groupe sont demandées en utilisant le paramètre `-g`. Ces champs sont utiles pour visualiser le statut lorsque le groupe fait partie d'une configuration Native HA CRR :

V 9.4.2 **GRPNAME**
Spécifie le nom du groupe HA. Blanc si aucun nom n'a été configuré.

V 9.4.2 **GRPROLE**
Spécifie le rôle actuel du groupe HA et est l'un des suivants :

- `Live` - Un groupe vivant
- `Recovery` - Un groupe de rétablissement
- `Pending live` - Un groupe en devenir `Live`

- `Pending recovery` - Un groupe en devenir `Recovery`
- `Unknown` - Le rôle actuel ne peut être déterminé
- `Not configured` - L'HA natif n'a pas été configuré

V 9.4.2 **GRPADDR**

L'adresse IP utilisée pour se connecter au groupe. La valeur est fixée à Unknown si l'adresse n'est pas connue parce qu'il n'y a pas eu de connexion au groupe.

V 9.4.2 **GRPVER**

Indique la version du chef de groupe actuel au format `V.R.M.F`, où `V.R.M.F` est le numéro de Version, Release, Modification et Fix Pack. Si la valeur n'est pas connue, elle est fixée à `?.?.?.?`

V 9.4.2 **CONNGRP**

Indique si le groupe est connecté au groupe local et s'il est l'un des suivants :

- `yes` - le groupe est connecté.
- no - le groupe est déconnecté.
- `unknown` - l'état de la connexion est inconnu.
- `suspended` - le groupe est connecté, mais il ne peut pas répliquer les données tant qu'une incompatibilité de configuration entre les groupes n'est pas résolue.

Ce champ n'est affiché que pour les groupes autres que le groupe local.

V 9.4.2 **INSYNC**

Ceci n'est affiché que pour un groupe ayant le rôle `Recovery` ou `Pending live`. Il indique si le groupe peut devenir `Live` sans perte de données en cas de basculement. (Notez qu'un groupe d' `Recovery` s peut ne pas être très souvent synchronisé avec le groupe d' `Live` s car le journal est répliqué de manière asynchrone. Utilisez les valeurs RCOVLSN et RCOVTIME pour indiquer si une perte de données peut se produire en cas de basculement non planifié).

V 9.4.2 **SYNCTIME**

Ceci n'est affiché que pour un groupe ayant le rôle `Recovery` ou `Pending live`. Il indique l'heure à laquelle ce groupe a été synchronisé pour la dernière fois avec le groupe d' `Live` , au format ISO 8601 (blanc si elle n'est pas connue). (Notez qu'un groupe d' `Recovery` s peut ne pas être très souvent synchronisé avec le groupe d' `Live` s car le journal est répliqué de manière asynchrone. Utilisez les valeurs RCOVLSN et RCOVTIME pour indiquer si une perte de données peut se produire en cas de basculement non planifié).

V 9.4.2 **BACKLOG**

Ceci n'est affiché que pour un groupe ayant le rôle `Recovery` ou `Pending live`. Il indique le nombre de Ko de retard du groupe par rapport au groupe en direct.

V 9.4.2 **GRSTATUS**

Indique l'état opérationnel de ce groupe et correspond à l'une des valeurs suivantes :

- `Normal` - le groupe fonctionne normalement.
- `Checking` - le groupe est vérifié pour assurer la cohérence du journal des récupérations.
- `Synchronizing` - le groupe reçoit des données pour le synchroniser.
- `Rebasing` - un nouveau journal de récupération est envoyé à l'instance.
- `Waiting for connection` - le groupe attend une connexion d'un groupe ayant le rôle `Live` ou `Pending Recovery`.
- `Partitioned` - deux groupes croient qu'ils sont le groupe Live (c'est-à-dire un statut de "cerveau divisé").
- `Unknown` - le statut du groupe est inconnu.

V 9.4.2 **RCOVLSN**
Un numéro de séquence d'enregistrement (LSN) que le groupe pourrait récupérer, au format *nnnnn:nnnnn:nnnnn:nnnnn*. Toute donnée écrite dans le journal de récupération après ce point pourrait être perdue en cas de basculement imprévu. Cette valeur peut être transmise à l'option -s de la commande **dmpmqlog**.

V 9.4.2 **RCOVTIME**
Heure à laquelle le groupe pourrait récupérer, au format ISO 8601. Toute donnée écrite dans le journal de récupération après ce délai pourrait être perdue en cas de basculement imprévu. Cette valeur peut être transmise dans les options -t ou -u de la commande **dmpmqlog**.

V 9.4.2 **INITLSN**
Ceci n'est affiché que pour un groupe ayant le rôle `Live` ou `Pending recovery`. Il indique le numéro de séquence du journal (LSN) du dernier enregistrement récupéré lorsque le groupe Native HA est devenu actif, au format *nnnnn:nnnnn:nnnnn:nnnnn*, ou vide si le groupe n'est pas actif.

V 9.4.2 **initTime**
Ceci n'est affiché que pour un groupe ayant le rôle `Live` ou `Pending recovery`. Il indique l'heure du dernier enregistrement récupéré lorsque le groupe Native HA a été mis en service, au format ISO 8601. Blanc, si le groupe n'est pas en direct.

V 9.4.2 **LIVETIME**
Ceci n'est affiché que pour un groupe ayant le rôle `Live` ou `Pending recovery`. Il indique l'heure à laquelle le groupe est devenu le groupe en direct, dans le format ISO 8601. Blanc, si le groupe n'est pas en direct.

V 9.4.2 **ALTDATE**
Indique la date de la dernière mise à jour de ces informations.

V 9.4.2 **ALTTIME**
Indique l'heure à laquelle cette information a été mise à jour pour la dernière fois.

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)
```

  V 9.4.2 Une instance de BOB qui est un chef de groupe de récupération dans une configuration CRR Native HA signalerait l'état suivant :

```
QMNAME(BOB)                    STATUS(Recovery group leader)
```

  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)
GRPLSN(<0:25:365:1000>) GRPNAME() GRPROLE(Live)
```

V 9.4.2

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)
GRPLSN(<0:25:365:1000>) GRPNAME() GRPROLE(Live)
```

V 9.4.2

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)
GRPLSN() GRPNAME() GRPROLE(Live)
```

V 9.4.2

V 9.4.2 Une instance active d'un gestionnaire de file d'attente nommé BOB appartenant au groupe Live dans une configuration Native HA CRR peut signaler l'état suivant :

```
QMNAME(BOB)         ROLE(Active) INSTANCE(inst1) INSYNC(yes) QUORUM(3/3)
GRPLSN(<0:25:365:1000>) GRPNAME(alpha) GRPROLE(Live)
```

V 9.4.2 La même instance, lorsque le groupe attend de devenir le groupe de recouvrement, peut signaler l'état suivant :

```
QMNAME(BOB)         ROLE(Active) INSTANCE(inst1) INSYNC(yes) QUORUM(3/3)
GRPLSN(<0:25:365:1000>) GRPNAME(alpha) GRPROLE(Pending recovery)
```

V 9.4.2 Un gestionnaire de file d'attente nommé CAROL qui n'est pas une instance d'une configuration Native HA ou Native HA CRR devrait signaler l'état suivant :

```
QMNAME(CAROL)       ROLE(Not configured) INSTANCE() INSYNC() QUORUM() GRPLSN() GRPNAME()
GRPROLE(Not configured)
```

- 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
```

Par exemple, si vous lancez cette commande sur le nœud exécutant l'instance active du gestionnaire de file d'attente BOB, où toutes les instances sont synchronisées, vous pouvez obtenir l'état suivant :

```
QMNAME(BOB)         ROLE(Active) INSTANCE(inst1) INSYNC(yes) QUORUM(2/3)
GRPLSN(<0:25:365:1000>) GRPROLE(Live) GRPNAME()
 INSTANCE(inst1) ROLE(Active)  REPLADDR(localhost) CONNACTV(yes) INSYNC(yes)
BACKLOG(0)  CONNINST(yes) ACKLSN(<0:26:365:1000>)  HASTATUS(Normal)
SYNCTIME(2023-12-20T17:13:27.853412Z) ALTDATE(2023-12-20) ALTTIME(17.13.27)
 INSTANCE(inst3) ROLE(Replica) REPLADDR(localhost) CONNACTV(yes) INSYNC(yes)
BACKLOG(0)  CONNINST(yes) ACKLSN (<0:24:365:1000>) HASTATUS(Normal)
SYNCTIME(2023-12-20T17:13:27.853412Z) ALTDATE(2023-12-20) ALTTIME(17.13.27)
 INSTANCE(inst2) ROLE(Replica) REPLADDR(localhost) CONNACTV(yes) INSYNC(yes)
BACKLOG(0)  CONNINST(yes) ACKLSN (<0:25:365:1000>) HASTATUS(Synchronizing)
SYNCTIME(2023-12-20T17:13:22.123321Z) ALTDATE(2023-12-20) ALTTIME(17.13.27)
```

Dans un autre exemple, si vous lancez cette commande sur un nœud exécutant une instance répliquée du gestionnaire de file d'attente BOB, qui est à la traîne, vous pouvez obtenir l'état suivant :

```
QMNAME(BOB)         ROLE(Replica) INSTANCE(inst2) INSYNC(no) QUORUM(2/3)
GRPLSN(<0:25:365:1000>) GRPROLE(Live) GRPNAME()
 INSTANCE(inst1) ROLE(Active)  REPLADDR(localhost) CONNACTV(yes) INSYNC(yes)
BACKLOG(0)  CONNINST(yes) ACKLSN(<0:26:365:1000>)  HASTATUS(Normal)
SYNCTIME(2023-12-20T17:13:27.853412Z) ALTDATE(2023-12-20) ALTTIME(17.13.27)
 INSTANCE(inst3) ROLE(Replica) REPLADDR(localhost) CONNACTV(yes) INSYNC(yes)
BACKLOG(0)  CONNINST(yes) ACKLSN (<0:24:365:1000>) HASTATUS(Normal)
SYNCTIME(2023-12-20T17:13:27.853412Z) ALTDATE(2023-12-20) ALTTIME(17.13.27)
 INSTANCE(inst2) ROLE(Replica) REPLADDR(localhost) CONNACTV(yes) INSYNC(no)
BACKLOG(435) CONNINST(yes) ACKLSN (<0:25:365:1000>) HASTATUS(Synchronizing)
SYNCTIME(2023-12-20T17:13:22.123321Z) ALTDATE(2023-12-20) ALTTIME(17.13.27)
```

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) GRPLSN()
GRPROLE() GRPNAME()
 INSTANCE(inst1) ROLE(Unknown) REPLADDR(9.20.123.45) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ACKLSN()  HASTATUS(Unknown) SYNCTIME() ALTDATE() ALTTIME()
ALTDATE() ALTTIME()
 INSTANCE(inst2) ROLE(Unknown) REPLADDR(9.20.123.46) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ACKLSN()  HASTATUS(Unknown) SYNCTIME() ALTDATE() ALTTIME()
ALTDATE() ALTTIME()
 INSTANCE(inst3) ROLE(Unknown) REPLADDR(9.20.123.47) CONNACTV(No)      INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ACKLSN()  HASTATUS(Unknown) SYNCTIME() ALTDATE() ALTTIME()
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)
```

- V 9.4.2

  Pour déterminer l'état opérationnel de tous les groupes Native HA dans une configuration Native HA CRR :

  ```
  oc exec -t Pod dspmq -o nativeha -g -m QMgrName
  ```

  Si vous émettez cette commande sur un membre du groupe Live, nommé alpha, vous pourriez recevoir le statut suivant pour alpha, et le groupe Recovery, beta :

  ```
  GRPNAME(alpha) GRPROLE(Live) GRPADDR(9.20.135.129) GRPVER(09040200)
  GRSTATUS(Normal) RCOVLSN(<0:0:104:65163>) RCOVTIME(2023-12-20T17:37:27.114787Z)
  INITLSN(<0:0:101:352>) INITTIME(2023-12-19T09:13:27.512294)
  LIVETIME(2023-12-19T06:14:27.512294) ALTDATE(2023-12-20) ALTTIME(17.13.27)
  ```

  ```
  GRPNAME(beta) GRPROLE(Recovery) GRPADDR(9.20.135.130) GRPVER(09040300)
  CONNGRP(yes) GRSTATUS(Synchronizing) RCOVLSN(<0:0:19:16723>)
  RCOVTIME(2023-12-20T17:36:23.463209Z) BACKLOG(5619) INSYNC(no) SYNCTIME()
  ALTDATE(2023-12-20) ALTTIME(17.13.27)
  ```

Tâches associées

«Exemple: Configuration de Native HA à l'aide de IBM MQ Operator», à la page 116
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 103 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. V 9.4.2 *Réglage avancé pour Native HA CRR*

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 pour configurer Native HA CRR sont gérées à l'aide de l'API d' `QueueManager` , que l' IBM MQ Operator utilise pour configurer les fichiers INI du gestionnaire de file d'attente sous-jacent à votre place. Il existe des options plus avancées qui ne sont configurables qu'à l'aide d'un fichier INI, dans la strophe « NativeHARecoveryGroup » du fichier « qm.ini ». Voir également «Exemple : fourniture de fichiers MQSC et INI», à la page 103 pour plus d'informations sur la configuration d'un fichier INI.

`ShortRetryTimer`
: Spécifie l'intervalle à respecter entre les tentatives de connexion à un groupe de récupération dans la phase de réessai court. Spécifié en millisecondes, la valeur par défaut est de 2500 (2.5 s de secondes).

`ShortRetryCount`
: Spécifie le nombre de tentatives pour établir une connexion avec le groupe de récupération pendant la phase de réessai court.

`LongRetryTimer`
: Spécifie l'intervalle à respecter entre les tentatives de connexion à un groupe de récupération dans la phase de réessai longue. La longue phase de réessai se poursuit indéfiniment jusqu'à ce que la connexion soit établie ou jusqu'à ce que la configuration soit modifiée pour commenter ou supprimer l' `ReplicationAddress`. Spécifié en millisecondes, la valeur par défaut est 60000 (60 secondes).

`RetryReporting`
: Définissez `Phase` ou `All` pour spécifier si chaque tentative de relance est signalée, ou seulement la phase (longue ou courte). Si non spécifié, la valeur par défaut est `Phase`.

`StatusInterval`
: Indique le délai maximal que le groupe de récupération attendra avant d'envoyer un état du journal de récupération. Spécifié en millisecondes, la valeur par défaut est 10000 (10 secondes).

StatusCommitThreshold
Spécifie le nombre maximal d'octets que le groupe de récupération valide avant d'envoyer un état du journal de récupération. Spécifié en octets et par défaut à 0, ce qui signifie que seul l' `StatusInterval` e est utilisé pour déterminer quand l'état du journal de récupération est envoyé.

OpenShift MQ Adv. *Arrêt des gestionnaires de files d'attente Native HA*

Pour IBM MQ dans des conteneurs, vous pouvez utiliser la commande **endmqm** pour mettre fin à un gestionnaire de file d'attente actif ou répliqué qui fait partie d'un groupe Native HA.

Avant de commencer

Remarque : Ces informations ne s'appliquent qu'aux environnements de conteneurs.

Pourquoi et quand exécuter cette tâche

La procédure d'arrêt d'un gestionnaire de files d'attente faisant partie d'un groupe Native HA dépend du fait qu'il s'agit d'une instance active ou répliquée. Lorsque vous terminez l'un ou l'autre type d'instance, un contrôle est effectué pour s'assurer que la fin de l'instance ne brise pas le quorum du groupe Native HA. Si le quorum est rompu, la commande **endmqm** échoue.

Lorsque vous lancez une commande **endmqm**, les autres instances du groupe en sont averties, de sorte qu'elles ne signalent pas d'erreurs lorsque la connexion est interrompue.

Si une instance active perd le quorum en raison de l'arrêt ou de la déconnexion d'un trop grand nombre d'instances répliques, l'instance active attend pendant une durée configurable avant de s'arrêter complètement. Cela permet de disposer d'un certain temps pour arrêter le traitement de manière gracieuse, plutôt que d'interrompre les connexions des applications. Cette valeur peut être spécifiée par l'attribut `QuorumConnectivityTimeout` dans la strophe `NativeHALocalInstance` du fichier `qm.ini`. La valeur par défaut est 0 seconde.

Procédure

- Pour mettre fin à l'instance active d'un gestionnaire de files d'attente, exécutez la commande suivante sur le nœud où l'instance active est en cours d'exécution :

  ```
  endmqm -s QMgrName
  ```

  – Spécifiez l'option `-r` pour aider les applications clientes à se reconnecter à une autre instance.
  – Si cette instance n'est pas l'instance active dans le groupe Native HA, la commande échoue.
  – Si l'arrêt de cette instance active entraîne l'échec du quorum du groupe, la commande échoue. (Si d'autres instances se terminent ou deviennent indisponibles en même temps que vous exécutez cette commande, la vérification du quorum peut ne pas le détecter, le groupe Native HA se termine et ne peut être redémarré que lorsque suffisamment d'instances sont disponibles)

  Lorsque le gestionnaire de file d'attente actif prend fin, l'une des instances de réplique reprend le rôle actif. Vous ne pouvez pas spécifier quelle réplique prend le relais, cela est déterminé par négociation au sein du groupe et dépend de celle qui a les journaux de transactions les plus à jour.
- Pour mettre fin à une instance répliquée d'un gestionnaire de files d'attente, exécutez la commande suivante :

  ```
  endmqm -x QMgrName
  ```

  – Si cette instance est l'instance active, la commande échoue.
  – Si l'arrêt de cette instance de réplique entraîne l'échec du quorum du groupe, la commande échoue. (Si d'autres instances se terminent ou deviennent indisponibles en même temps que vous exécutez cette commande, la vérification du quorum peut ne pas le détecter, le groupe Native HA se termine et ne peut être redémarré que lorsque suffisamment d'instances sont disponibles)

Remarque : Vous pouvez également utiliser les commutateurs -c, -i, -p ou -w avec la commande **endmqm** sur les instances Native HA, quel que soit leur rôle. L'instance du gestionnaire de file d'attente se termine, sans tenir compte de l'effet sur le quorum du groupe. Toutefois, les informations sont toujours partagées avec les autres instances du groupe. Vous pouvez utiliser ces commutateurs avec le -s pour l'instance active. Vous ne pouvez pas utiliser ces commutateurs en même temps que le commutateur -x pour les instances de réplique.

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 107.
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 150. 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. Assurez-vous d'être dans l'espace de noms que vous avez créé avant de commencer cette tâche, puis entrez le YAML suivant dans la console web de l'OCP, ou en utilisant la ligne de commande. Vérifiez que la licence est correcte et acceptez-la en remplaçant `false` par `true`.

   Remarque : Cet exemple utilise une licence IBM MQ Advanced , mais vous pouvez également utiliser une licence IBM MQ . Pour plus d'informations, voir «Configurer les gestionnaires de files d'attente avec les annotations de licence IBM MQ en utilisant la fonction IBM MQ Operator», à la page 166.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: exampleqm
spec:
  license:
    accept: false
    license: L-NUUP-23NH8Y
    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: 9.4.3.0-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 150.

Notez que la valeur confidentielle *example-qm-tls* a été créée à l'étape «2», à la page 150et que la valeur confidentielle ConfigMap *example-miqm-configmap* a été créée à l'étape «3», à la page 150

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 111.

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 111.

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 224
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 .

Configuration de la journalisation d'audit pour IBM MQ dans les conteneurs

IBM MQ dans les conteneurs offre diverses possibilités d'enregistrement d'audit de différents événements.

En fonction des exigences de votre environnement en matière de journalisation des audits, vous devrez peut-être configurer une ou plusieurs de ces fonctions, ainsi que des mécanismes appropriés de collecte et de conservation des journaux d'événements. Choisissez l'une des options décrites ci-après.

IBM MQ Operator Webhook de validation

À partir de la version 3.6.0, l'application IBM MQ Operator émet par défaut les événements d'audit d'accès aux points d'extrémité exposés dans le journal du conteneur du pod. Aucune configuration n'est nécessaire.

événements IBM MQ

Vous pouvez configurer IBM MQ pour qu'il produise des messages d'événements à des fins d'audit. Pour plus d'informations, voir Audit.

V 9.4.3 MQ métriques des conteneurs

Le conteneur IBM MQ expose un point d'accès aux métriques Prometheus (pour plus d'informations, voir «Surveillance lors de l'utilisation de IBM MQ Operator», à la page 224). À partir de IBM MQ 9.4.3, vous pouvez activer la journalisation de l'accès à ce point de terminaison en définissant la variable d'environnement `MQ_LOGGING_METRICS_AUDIT_ENABLED` sur `true`. Si vous utilisez IBM MQ Operator, vous pouvez définir cette variable dans la section `env` de la configuration `queueManager` :

```
[...]
spec:
  queueManager:
    env:
    - name: MQ_LOGGING_METRICS_AUDIT_ENABLED
      value: "true"
[...]
```

Lorsque la journalisation est activée, les fichiers journaux sont générés dans le répertoire `/var/mqm/errors` . Vous pouvez conserver les données qui se trouvent dans ce répertoire en montant une unité de stockage externe sur `/mnt/mqm` (voir la référence de l'image du conteneur). Si vous utilisez le site IBM MQ Operator, vous pouvez mapper le stockage externe en utilisant les champs «.spec.queueManager.storage.persistedData», à la page 275 . Les fichiers journaux d'accès aux métriques sont écrits au format JSON. Les noms de fichiers ont la forme `metricaudit<nn>.json`, où nn est un nombre. Le fichier `metricaudit01.json` contient les événements les plus récents.

Serveur web intégré IBM WebSphere Application Server Liberty

Vous pouvez configurer le serveur web intégré WebSphere Liberty pour qu'il produise des événements d'audit. Les fonctionnalités `audit-1.0` et `audit-2.0` sont installées et disponibles pour la configuration, en fonction de vos besoins. Pour plus d'informations, voir Audit 1.0 et Audit 2.0 dans la documentation WebSphere Liberty .

Vous pouvez activer et configurer la fonction à l'aide d'un fichier de configuration. Si vous utilisez IBM MQ Operator, vous pouvez charger le fichier en utilisant les montages «.spec.web.manualConfig», à la page 283 . Par exemple, le code suivant active la configuration par défaut de la fonction Audit 2.0.

Créez un fichier de configuration appelé `mqwebuser.xml`, pour activer la fonction `audit-2.0` :

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: example-enable-audit
data:
  mqwebuser.xml: |
    <?xml version="1.0" encoding="UTF-8"?>
    <server>
      <featureManager>
        <feature>audit-2.0</feature>
```

```
      </featureManager>
    </server>
```

Ajoutez le code précédent à la configuration queueManager en utilisant le montage spec.web.manualConfig :

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
[...]
spec:
  [...]
  web:
    enabled: true
    manualConfig:
      configMap:
        name: example-enable-audit
```

Remarque : Par défaut, les journaux d'audit sont mis en miroir dans le journal du conteneur au format de base (texte brut). Ce format contient un ensemble réduit d'informations d'audit. Pour obtenir plus d'informations, vous pouvez soit modifier le journal du conteneur IBM MQ au format JSON, soit désactiver la mise en miroir du journal web et collecter directement le fichier journal d'audit JSON de WebSphere Liberty . Vous pouvez contrôler le format et les sources de mise en miroir des journaux en utilisant les variables d'environnement MQ_LOGGING_CONSOLE_FORMAT et MQ_LOGGING_CONSOLE_SOURCE . Pour les valeurs valides, voir «Variables d'environnement prises en charge», à la page 296.

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 158.

Remarque :

- Cette fonction n'est prise en charge que pour les opérandes de la version IBM MQ 9.3.1.0-r2 ou ultérieure.
- Vous pouvez exécuter IBM Instana tracing sur les versions précédentes de IBM MQ Operator et du gestionnaire de file d'attente, mais pas de manière native. Voir Configuration de IBM MQ Tracing dans la documentation IBM Instana .
- Cette fonction n'est pas prise en charge lors de l'utilisation du réseau IPv6 .

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 **Opérateurs** > **Opérateurs installés** > **IBM MQ** > **Détails** > **Créer une instance**. Fournissez toutes les informations nécessaires à votre gestionnaire de file d'attente.
  2. Sous le même emplacement, cliquez sur **Telemetry** > **Tracing** > **Instana**.
  3. 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 **Opérateurs** > **Opérateurs installés** > **IBM MQ** > **Détails** > **Créer une instance**. Fournissez toutes les informations nécessaires à votre gestionnaire de file d'attente.
  2. Sous le même emplacement, cliquez sur **Telemetry** > **Instana**.
  3. Développez la liste déroulante **Configuration avancée** .
  4. 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 **Opérateurs** > **Opérateurs installés** > **IBM MQ** > **Détails** > **Créer une instance**. Fournissez toutes les informations nécessaires à votre gestionnaire de file d'attente.
2. Sous le même emplacement, cliquez sur **Telemetry** > **Instana**.
3. Développez la liste déroulante **Configuration avancée** .
4. 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 100.

Operator2.2.0 OpenShift CP4I 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 156
IBM Instana peut être utilisé pour tracer des transactions dans IBM Cloud Pak for Integration.

Operator2.2.0 OpenShift CP4I *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.

Operator2.2.0 OpenShift CP4I *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. Montez le secret en tant que volume dans le pod de l'agent Instana.

   Pour ce faire, spécifiez le volume et les montures de volume dans l'agent `CustomResource` ou dans le tableau `values.yaml` Helm :

   ```
   agent:
     pod:
       volumeMounts:
       - mountPath: /opt/instana/agent/etc/jks-file-name.jks
         name: mq-key-jks-name
         subPath: jks-file-name.jks
       volumes:
       - name: mq-key-jks-name
         secret:
           secretName: keystore_secret_name
   ```

   Pour effectuer cette modification sur un agent Instana déployé par l'opérateur d'agent Instana, exécutez une commande similaire à la suivante :

   ```
   oc edit agent [instana-agent-name] -n [instana-agent-namespace]
   ```

   Pour plus d'informations sur l'intégration de secrets dans le pod de l'agent Instana, consultez Volumes et volumeMounts dans la documentation de la charte d' Helm s de l'agent Instana.

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.

Operator2.2.0 OpenShift CP4I *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 103.

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 156.
7. Déployez le gestionnaire de files d'attente.

Que faire ensuite

Vous êtes maintenant prêt à vérifier et déboguer la surveillance IBM Instana.

Operator2.2.0 OpenShift CP4I

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:9.4.3.0-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:9.4.3.0-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 100 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 100
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 Configurer les gestionnaires de files d'attente avec les annotations de licence IBM MQ en utilisant la fonction IBM MQ Operator

Des annotations doivent être appliquées aux gestionnaires de file d'attente pour les configurer avec une licence IBM MQ .

Avant de commencer

Remarque : Ces instructions ne s'appliquent que si vous utilisez le site IBM MQ Operator. Si vous déployez des gestionnaires de file d'attente sur Kubernetes, consultez plutôt IBM MQ annotations de licence de conteneur.

Pourquoi et quand exécuter cette tâche

Vous pouvez ajouter des annotations aux gestionnaires de file d'attente soit lorsque vous créez les gestionnaires de file d'attente, soit en mettant à jour les gestionnaires de file d'attente existants via le site `spec.annotations`.

Important : La mise à jour du YAML du gestionnaire de file d'attente déclenche un redémarrage du pod.

Procédure

Mettez à jour les annotations de licence de vos gestionnaires de file d'attente existants en suivant les instructions sur «Ajout d'annotations aux gestionnaires de file d'attente à l'aide de la fonction IBM MQ Operator», à la page 167, en utilisant les annotations de licence requises pour chaque gestionnaire de file d'attente, en fonction de vos droits.

Annotations requises sur les licences

IBM MQ

```
spec:
  annotations:
    productID: c661609261d5471fb4ff8970a36bccea
    productName: IBM MQ
```

IBM MQ for Non-Production Environment

```
spec:
  annotations:
    productID: 151bec68564a4a47a14e6fa99266deff
    productName: IBM MQ for Non-Production Environment
```

IBM MQ Conteneur Multi Instance

```
spec:
  annotations:
    productID: 2dea73b866b648b6b4abe2a85eb76964
    productName: IBM MQ Container Multi Instance
```

IBM MQ Container Multi Instance pour l'environnement de non-production :

```
spec:
  annotations:
    productID: af11b093f16a4a26806013712b860b60
    productName: IBM MQ Container Multi Instance for Non-Production Environment
```

IBM MQ avec CP4I droit

```
spec:
  annotations:
    cloudpakId: c8b82d189e7545f0892db9ef2731b90d
    cloudpakName: IBM Cloud Pak for Integration
    productID: c661609261d5471fb4ff8970a36bccea
    productName: IBM MQ
    productMetric: VIRTUAL_PROCESSOR_CORE
    productCloudpakRatio: 4:1
```

IBM MQ pour l'environnement de non-production avec le droit CP4I

```
spec:
  annotations:
    cloudpakId: c8b82d189e7545f0892db9ef2731b90d
    cloudpakName: IBM Cloud Pak for Integration
    productID: 151bec68564a4a47a14e6fa99266deff
    productName: IBM MQ for Non-Production Environment
    productMetric: VIRTUAL_PROCESSOR_CORE
    productCloudpakRatio: 8:1
```

IBM MQ Container Multi Instance avec CP4I entitlement

:NONE.

```
spec:
  annotations:
    productName: IBM MQ Container Multi Instance
    productID: 2dea73b866b648b6b4abe2a85eb76964
    productMetric: VIRTUAL_PROCESSOR_CORE
    productCloudpakRatio: 10:3
    cloudpakName: IBM Cloud Pak for Integration
    cloudpakId: c8b82d189e7545f0892db9ef2731b90d
```

IBM MQ Container Multi Instance pour l'environnement de non-production avec CP4I entitlement

```
spec:
  annotations:
    cloudpakId: c8b82d189e7545f0892db9ef2731b90d
    cloudpakName: IBM Cloud Pak for Integration
    productID: af11b093f16a4a26806013712b860b60
    productName: IBM MQ Container Multi Instance for Non-Production Environment
    productMetric: VIRTUAL_PROCESSOR_CORE
    productCloudpakRatio: 20:3
```

OpenShift CP4I Ajout d'annotations aux gestionnaires de file d'attente à l'aide de la fonction IBM MQ Operator

Vous pouvez ajouter des annotations aux gestionnaires de file d'attente lors de leur création ou en mettant à jour les gestionnaires de file d'attente existants à l'aide de la page `spec.annotations`.

Pourquoi et quand exécuter cette tâche

Ces instructions ne s'appliquent que si les deux affirmations suivantes sont vraies :

- Vous utilisez le site IBM MQ Operator. Si vous déployez des gestionnaires de file d'attente sur Kubernetes, consultez plutôt «IBM MQ annotations sur la licence du conteneur», à la page 289 .
- Vous modifiez un gestionnaire de file d'attente existant. Pour les nouveaux gestionnaires de file d'attente, ajoutez plutôt les annotations lors de la création.

Important : La mise à jour du YAML du gestionnaire de file d'attente déclenche un redémarrage du pod.

Procédure

- **Option 1 :** Ajouter des annotations en utilisant la console OpenShift .
  a) Accédez à votre gestionnaire de file d'attente

a. Connectez-vous à la console OpenShift avec vos identifiants d'administrateur du cluster Red Hat OpenShift Container Platform .
b. Modifiez le **projet** dans l'espace de noms où vous avez installé le site IBM MQ Operator. Sélectionnez l'espace de noms dans la liste déroulante **Projet.**
c. Dans le volet de navigation, cliquez sur **OpérateursInstallés** > **.**
d. Dans la liste du panneau Opérateurs installés, recherchez et cliquez sur **IBM MQ**.
e. Cliquez sur l'onglet **Gestionnaire de files d'attente**.
f. Cliquez sur le **nom du gestionnaire de files d'attente** requis.

b) Ajoutez les annotations nécessaires à votre gestionnaire de file d'attente.

a. Cliquez sur l'onglet **YAML**.
b. Localisez le champ `spec` .
c. Ajouter les annotations nécessaires au gestionnaire de file d'attente. Créez le champ `annotations` s'il n'existe pas déjà.
d. Cliquez sur **Sauvegarder**.

Le pod du gestionnaire de file d'attente redémarre et les annotations sont appliquées.

c) Répétez ces étapes pour d'autres gestionnaires de file d'attente si nécessaire.

- **Option 2 :** Ajouter des annotations en utilisant l'interface de ligne de commande (CLI) de OpenShift .

a) Connectez-vous au CLI de Red Hat OpenShift en utilisant la commande `oc login` .

b) Définissez votre `project` (espace de noms) sur celui dans lequel votre gestionnaire de files d'attente est déployé :

```
oc project <QUEUE-MANAGER-NAMESPACE>
```

c) Identifiez le nom du gestionnaire de file d'attente que vous souhaitez mettre à jour :

```
oc get queuemanager
```

d) Ajoutez les annotations requises à votre gestionnaire de file d'attente en modifiant le gestionnaire de file d'attente :

```
oc edit queuemanager <QUEUE-MANAGER-NAME>
```

e) Répétez ces étapes pour d'autres gestionnaires de file d'attente si nécessaire.

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 277 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 275, 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 276. La valeur par défaut est "100M".
   - Pour le volume éphémère `Tmp` , définissez la propriété **`sizeLimit`** dans «.spec.queueManager.storage.tmp», à la page 277. 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 275. 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 301. 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 224.

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 171.

  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 171.

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 171.

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 9.4.1 Correction des annotations de licence pour les gestionnaires de file d'attente déployés

Avant la version 9.4.1.0-r1 du gestionnaire de file d'attente 'IBM MQ, certains gestionnaires de file d'attente étaient déployés avec des annotations de licence incorrectes. Depuis la version 9.4.1.0-r1, tous les gestionnaires de file d'attente sont automatiquement déployés avec les annotations de licence correctes.

Pourquoi et quand exécuter cette tâche

Pour les gestionnaires de file d'attente " IBM MQ déjà déployés, il existe deux options pour corriger les annotations de licence :

- Mettez à jour votre gestionnaire de file d'attente vers la version 9.4.1.0-r1 (nécessite 'IBM MQ Operator 3 3.3.0 ou plus récent). Voir «Mise à niveau de IBM MQ Operator et des gestionnaires de files d'attente», à la page 71. Notez que si votre gestionnaire de file d'attente YAML contient des annotations de licence non valides, vous devez les résoudre comme demandé lors de la mise à niveau de chaque gestionnaire de file d'attente.
- Pour les anciens gestionnaires de file d'attente qui n'ont pas été mis à jour et qui sont déployés en utilisant les licences suivantes, les annotations de licence peuvent être corrigées en mettant à jour le YAML du gestionnaire de file d'attente. Effectuez les étapes suivantes pour ajouter les annotations spécifiées au gestionnaire de file d'attente YAML. Notez que cette procédure doit être effectuée individuellement pour chaque gestionnaire de file d'attente.

Important : La mise à jour du YAML du gestionnaire de file d'attente déclenche un redémarrage du pod.

Utilisations de licences affectées :

- IBM MQ Advanced for Non-Production Environment avec droit " CP4I
- IBM MQ Advanced for Developers
- IBM MQ Advanced Container Multi Instance
- IBM MQ Advanced Container Multi Instance avec droit " CP4I
- IBM MQ Advanced Container Multi Instance for Non-Production Environment avec droit " CP4I

Procédure

Mettez à jour les annotations de licence de vos gestionnaires de file d'attente existants en suivant les instructions sur «Ajout d'annotations aux gestionnaires de file d'attente à l'aide de la fonction IBM MQ Operator», à la page 167, en utilisant les annotations de licence requises pour chaque gestionnaire de file d'attente, en fonction de vos droits.

Annotations requises sur les licences

IBM MQ Advanced for Non-Production Environment avec droit " CP4I

```
spec:
  annotations:
    productName: IBM MQ Advanced for Non-Production Environment
```

IBM MQ Advanced for Developers

```
spec:
  annotations:
    productName: IBM MQ Advanced for Developers (Non-Warranted)
    productChargedContainers: [container_name]
```

où le *nom du conteneur* se trouve dans le fichier YAML du gestionnaire de file d'attente, spécifié comme " `spec.containers[0].name`

IBM MQ Advanced Container Multi Instance

```
spec:
  annotations:
    productID: bd35bff411bb47c2a3f3a4590f33a8ef
    productName: IBM MQ Advanced Container Multi Instance
```

IBM MQ Advanced Container Multi Instance avec droit " CP4I

```
spec:
  annotations:
    productID: bd35bff411bb47c2a3f3a4590f33a8ef
    productName: IBM MQ Advanced Container Multi Instance
    productCloudpakRatio: 5:3
```

IBM MQ Advanced Container Multi Instance for Non-Production Environment avec droit " CP4I

```
spec:
  annotations:
    productID: 31f844f7a96b49749130cd0708fdbb17
    productName: IBM MQ Advanced Container Multi Instance for Non-Production Environments
    productCloudpakRatio: 10:3
```

Kubernetes Déploiement et configuration des gestionnaires de files d'attente sur Kubernetes

Comment déployer des gestionnaires de file d'attente sur Kubernetes en utilisant des ressources natives.

Pourquoi et quand exécuter cette tâche

MQ Adv. Kubernetes V 9.4.1 Exemple : Configuration d'un gestionnaire de file d'attente simple dans " Kubernetes

Cet exemple déploie un gestionnaire de file d'attente simple, de type "démarrage rapide", qui utilise un stockage éphémère (non persistant) et désactive la sécurité d'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.

Avant de commencer

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

- Créez un espace de noms " Kubernetes pour cet exemple.
- Sur la ligne de commande, connectez-vous au cluster " Kubernetes et passez à l'espace de noms ci-dessus.
- Assurez-vous d'avoir suivi les étapes du " «Préparation de l'utilisation de 'Kubernetes par la création d'un secret d'extraction», à la page 88 dans l'espace de noms ci-dessus.

Pourquoi et quand exécuter cette tâche

Cet exemple fournit les fichiers YAML " `Service`, " `ServiceAccount` et " `StatefulSet` pour définir un gestionnaire de file d'attente à déployer dans " Kubernetes. Cet exemple requiert un minimum de " IBM MQ Advanced 9.4.1.

Procédure

1. Créez un " `Service` pour fournir une adresse IP cohérente au trafic du " IBM MQ

   Assurez-vous d'être dans l'espace de noms que vous avez créé précédemment (voir Avant de commencer), puis appliquez le YAML suivant à l'aide de la ligne de commande 'Kubernetes ou de la console.

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app.kubernetes.io/component: integration
    app.kubernetes.io/instance: quickstart
    app.kubernetes.io/name: ibm-mq
    app.kubernetes.io/version: 9.4.3.0
  name: quickstart-ibm-mq
spec:
  ports:
  - name: qmgr
    port: 1414
    protocol: TCP
    targetPort: 1414
  - name: metrics
    port: 9157
    protocol: TCP
    targetPort: 9157
  selector:
    app.kubernetes.io/component: integration
    app.kubernetes.io/instance: quickstart
    app.kubernetes.io/name: ibm-mq
  type: ClusterIP
```

2. Créer un " `ServiceAccount` à utiliser pour le gestionnaire de file d'attente.

L'exemple " `ServiceAccount` ci-dessous renvoie à un " `imagePullSecret` qui est utilisé pour vérifier les droits lorsque l'image est tirée. Pour plus d'informations, voir «Préparation de l'utilisation de 'Kubernetes par la création d'un secret d'extraction», à la page 88.

```
kind: ServiceAccount
apiVersion: v1
metadata:
  name: quickstart-ibm-mq
  labels:
    app.kubernetes.io/component: integration
    app.kubernetes.io/instance: quickstart
    app.kubernetes.io/name: ibm-mq
    app.kubernetes.io/version: 9.4.3.0
imagePullSecrets:
  - name: ibm-entitlement-key
```

3. Créer le gestionnaire de file d'attente Pod, géré par un " `StatefulSet`.

   Dans le fichier YAML suivant, le nom " `namespace` est supposé être " `mq-test`. Modifiez-le pour qu'il corresponde au nom que vous utilisez dans votre environnement " Kubernetes pour cet exemple.

   Ce YAML crée une instance unique de gestionnaire de file d'attente dans votre environnement 'Kubernetes dans l'espace de noms spécifié. Aucun volume de persistance n'est créé pour cette instance car notre exemple est un stockage éphémère.

   Remarque : Cet exemple utilise les annotations IBM MQ Advanced . Pour utiliser une licence IBM MQ , modifiez les annotations sous `spec.template.metadata` avec les valeurs spécifiées sous IBM MQ ou IBM MQ for Non-Production Environment dans «IBM MQ annotations sur la licence du conteneur», à la page 289.

```
kind: StatefulSet
apiVersion: apps/v1
metadata:
  name: quickstart-ibm-mq
  namespace: mq-test
  labels:
    app.kubernetes.io/component: integration
    app.kubernetes.io/instance: quickstart
    app.kubernetes.io/name: ibm-mq
    app.kubernetes.io/version: 9.4.3.0
spec:
  persistentVolumeClaimRetentionPolicy:
    whenDeleted: Retain
    whenScaled: Retain
  replicas: 1
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app.kubernetes.io/component: integration
      app.kubernetes.io/instance: quickstart
      app.kubernetes.io/name: ibm-mq
  template:
    metadata:
      labels:
        app.kubernetes.io/component: integration
        app.kubernetes.io/instance: quickstart
        app.kubernetes.io/name: ibm-mq
        app.kubernetes.io/version: 9.4.3.0
        statefulSetName: quickstart-ibm-mq
      annotations:
# These annotations are critical for the IBM License Service to charge you correctly
        productChargedContainers: qmgr
        productID: 208423bb063c43288328b1d788745b0c
        productMetric: PROCESSOR_VALUE_UNIT
        productName: IBM MQ Advanced
        productVersion: 9.4.3.0
    spec:
      restartPolicy: Always
      serviceAccountName: quickstart-ibm-mq
      terminationGracePeriodSeconds: 30
      securityContext: {}
      containers:
        - resources:
            limits:
              cpu: '1'
              memory: 1Gi
```

```
            requests:
              cpu: '1'
              memory: 1Gi
          readinessProbe:
            exec:
              command:
                - chkmqready
            initialDelaySeconds: 10
            timeoutSeconds: 3
            periodSeconds: 5
            successThreshold: 1
            failureThreshold: 1
          terminationMessagePath: /run/termination-log
          name: qmgr
          livenessProbe:
            exec:
              command:
                - chkmqhealthy
            initialDelaySeconds: 90
            timeoutSeconds: 5
            periodSeconds: 10
            successThreshold: 1
            failureThreshold: 1
          env:
            - name: LICENSE
              # if you accept the license, change the following value to "accept"
              value: view
            - name: MQ_QMGR_NAME
              value: QUICKSTART
            - name: MQ_MULTI_INSTANCE
              value: 'false'
            - name: MQ_ENABLE_METRICS
              value: 'true'
            - name: MQ_ENABLE_EMBEDDED_WEB_SERVER
              value: 'false'
            - name: MQ_LOGGING_CONSOLE_FORMAT
              value: basic
            - name: DEBUG
              value: 'false'
            - name: MQ_ENABLE_TRACE_STRMQM
              value: 'false'
            - name: MQ_EPHEMERAL_PREFIX
              value: /run/mqm
            - name: MQ_GRACE_PERIOD
              value: '29'
            - name: MQ_NATIVE_HA
              value: 'false'
          securityContext:
            allowPrivilegeEscalation: false
            capabilities:
              drop:
                - ALL
            privileged: false
            runAsNonRoot: true
            readOnlyRootFilesystem: false
          ports:
            - containerPort: 1414
              protocol: TCP
            - containerPort: 9157
              protocol: TCP
          terminationMessagePolicy: File
# The image to pull.  This example uses a prebuilt MQ Advanced image from IBM Container
Registry
          image: cp.icr.io/cp/ibm-mqadvanced-server:9.4.1.0-r1
          imagePullPolicy: IfNotPresent
      serviceAccount: quickstart-ibm-mq
      dnsPolicy: ClusterFirst
  serviceName: qm
  podManagementPolicy: OrderedReady
  updateStrategy:
    type: OnDelete
```

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 est dans l'état " `Running`, par exemple en listant les pods du gestionnaire de file d'attente :

```
kubectl get pods --selector app.kubernetes.io/instance=quickstart
```

Notez que cette opération doit renvoyer un seul pod et que le champ " READY doit afficher la valeur " 1/1.

5. Afficher le statut du gestionnaire de files d'attente

   Exécutez la commande suivante, en spécifiant le nom complet du module de gestion de la file d'attente :

   ```
   kubectl exec -t <qmgr_pod_name> -- dspmq -m QUICKSTART
   ```

   Vous devriez obtenir une réponse similaire à la suivante :

   ```
   QMNAME(QUICKSTART) STATUS(Running)
   ```

Résultats

Vous avez déployé avec succès un gestionnaire de file d'attente " IBM MQ simple, de type instance unique, sur votre plate-forme " Kubernetes.

MQ Adv. Kubernetes V 9.4.1 Exemple : Configuration de Native HA dans " Kubernetes

Cet exemple déploie un gestionnaire de file d'attente à l'aide de la fonctionnalité native de haute disponibilité dans le " Kubernetes en utilisant directement les ressources YAML. TLS mutuel est utilisé pour l'authentification, afin de faire correspondre un certificat TLS à une identité dans le gestionnaire de files d'attente.

Avant de commencer

Remarque : Cet exemple utilise une licence IBM MQ Advanced . A partir de IBM MQ 9.4.2, vous pouvez également configurer les gestionnaires de files d'attente sous licence IBM MQ pour utiliser les fonctionnalités Native HA et Cross-Region Replication en ajoutant des licences supplémentaires à un gestionnaire de files d'attente. Pour plus d'informations, voir «Configuration de Native HA et de Cross-Region Replication sur IBM MQ gestionnaires de files d'attente sous licence dans Kubernetes», à la page 192.

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

- Créez un espace de noms " Kubernetes pour cet exemple.
- Sur la ligne de commande, connectez-vous au cluster " Kubernetes et passez à l'espace de noms ci-dessus.
- Assurez-vous d'avoir suivi les étapes du " «Préparation de l'utilisation de 'Kubernetes par la création d'un secret d'extraction», à la page 88 dans l'espace de noms ci-dessus.

Pourquoi et quand exécuter cette tâche

Cet exemple fournit les fichiers YAML " `Service`, " `ServiceAccount` et " `StatefulSet` pour définir un gestionnaire de file d'attente à déployer dans " Kubernetes. Il détaille également les étapes supplémentaires requises pour déployer le gestionnaire de files d'attente avec TLS activé. Cet exemple requiert un minimum de " IBM MQ Advanced 9.4.1.

Procédure

1. Créez une paire de certificats comme décrit dans le " «Création d'une infrastructure PKI autosignée à l'aide de OpenSSL», à la page 107.
2. Créer une carte de configuration contenant des commandes MQSC et un fichier INI.

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

Assurez-vous d'être dans l'espace de noms que vous avez créé précédemment (voir Avant de commencer), puis appliquez le YAML suivant en utilisant la ligne de commande 'Kubernetes

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: example-nativeha-configmap
data:
  example.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.ini: |
    Service:
        Name=AuthorizationService
        EntryPoints=14
        SecurityPolicy=UserExternal
    NativeHAInstance:
      Name=exampleqm-ibm-mq-0
      ReplicationAddress=exampleqm-ibm-mq-replica-0(9414)
    NativeHAInstance:
      Name=exampleqm-ibm-mq-1
      ReplicationAddress=exampleqm-ibm-mq-replica-1(9414)
    NativeHAInstance:
      Name=exampleqm-ibm-mq-2
      ReplicationAddress=exampleqm-ibm-mq-replica-2(9414)
```

Le MQSC définit un canal appelé *MTLS.SVRCONN* et une file d'attente appelée *EXAMPLE.QUEUE*. Le canal est configuré pour n'autoriser l'accès qu'aux clients qui présentent un certificat portant le "nom commun" *example-app1*. Il s'agit du nom commun utilisé dans l'un des certificats créés à l'étape " «1», à la page 182. Les connexions sur ce canal avec ce nom commun sont associées à l'identifiant *app1*, qui est autorisé à se connecter au gestionnaire de file d'attente et à accéder à la file d'attente de l'exemple. Le fichier INI active une politique de sécurité qui signifie que l'identifiant de l'utilisateur *app1* n'a pas besoin d'exister dans un registre d'utilisateurs externe - il n'existe que sous forme de nom dans cette configuration. Il configure également les adresses de réplication à utiliser avec Native HA.

3. Créez un " `Service` pour fournir une adresse IP cohérente au trafic du " IBM MQ

```
apiVersion: v1
kind: Service
metadata:
  labels:
    app.kubernetes.io/component: integration
    app.kubernetes.io/instance: exampleqm
    app.kubernetes.io/name: ibm-mq
    app.kubernetes.io/version: 9.4.3.0
  name: exampleqm-ibm-mq
spec:
  ports:
  - name: qmgr
    port: 1414
    protocol: TCP
    targetPort: 1414
  - name: metrics
    port: 9157
    protocol: TCP
    targetPort: 9157
  selector:
    app.kubernetes.io/component: integration
    app.kubernetes.io/instance: exampleqm
    app.kubernetes.io/name: ibm-mq
  type: ClusterIP
```

4. Créez un " `Service` interne pour chacune des répliques Native HA afin qu'elles communiquent entre elles.

```
---
apiVersion: v1
kind: Service
metadata:
  labels:
    app.kubernetes.io/component: integration
    app.kubernetes.io/instance: exampleqm
    app.kubernetes.io/name: ibm-mq
    app.kubernetes.io/version: 9.4.3.0
  name: exampleqm-ibm-mq-replica-0
spec:
  ports:
  - port: 9414
    protocol: TCP
    targetPort: 9414
  publishNotReadyAddresses: true
  selector:
    app.kubernetes.io/component: integration
    app.kubernetes.io/instance: exampleqm
    app.kubernetes.io/name: ibm-mq
    statefulset.kubernetes.io/pod-name: exampleqm-ibm-mq-0
  type: ClusterIP
---
apiVersion: v1
kind: Service
metadata:
  labels:
    app.kubernetes.io/component: integration
    app.kubernetes.io/instance: exampleqm
    app.kubernetes.io/name: ibm-mq
    app.kubernetes.io/version: 9.4.3.0
  name: exampleqm-ibm-mq-replica-1
spec:
  ports:
  - port: 9414
    protocol: TCP
    targetPort: 9414
  publishNotReadyAddresses: true
  selector:
    app.kubernetes.io/component: integration
    app.kubernetes.io/instance: exampleqm
    app.kubernetes.io/name: ibm-mq
    statefulset.kubernetes.io/pod-name: exampleqm-ibm-mq-1
  type: ClusterIP
---
apiVersion: v1
kind: Service
metadata:
  labels:
    app.kubernetes.io/component: integration
    app.kubernetes.io/instance: exampleqm
    app.kubernetes.io/name: ibm-mq
    app.kubernetes.io/version: 9.4.3.0
  name: exampleqm-ibm-mq-replica-2
spec:
  ports:
  - port: 9414
    protocol: TCP
    targetPort: 9414
  publishNotReadyAddresses: true
  selector:
    app.kubernetes.io/component: integration
    app.kubernetes.io/instance: exampleqm
    app.kubernetes.io/name: ibm-mq
    statefulset.kubernetes.io/pod-name: exampleqm-ibm-mq-2
  type: ClusterIP
```

5. Créer un " `ServiceAccount` à utiliser pour le gestionnaire de file d'attente.

   L'exemple " `ServiceAccount` ci-dessous renvoie à un " `imagePullSecret` qui est utilisé pour vérifier les droits lorsque l'image est tirée. Pour plus d'informations, voir «Préparation de l'utilisation de 'Kubernetes par la création d'un secret d'extraction», à la page 88.

```
kind: ServiceAccount
apiVersion: v1
metadata:
  name: exampleqm-ibm-mq
  labels:
    app.kubernetes.io/component: integration
```

```
    app.kubernetes.io/instance: exampleqm
    app.kubernetes.io/name: ibm-mq
    app.kubernetes.io/version: 9.4.3.0
imagePullSecrets:
  - name: ibm-entitlement-key
```

6. Créer le gestionnaire de file d'attente Pod, géré par un " `StatefulSet`.

   Utilisez le fichier YAML suivant comme exemple, en effectuant ces changements :

   - Modifiez le nom `namespace` de `mq-test` pour qu'il corresponde au nom que vous utilisez dans votre environnement Kubernetes pour cet exemple.
   - Si vous essayez l'exemple sur un cluster Kubernetes avec moins de trois nœuds disponibles, supprimez la section `affinity` .
   - Optionnellement, changer la licence. L'exemple utilise une licence IBM MQ Advanced . A partir de IBM MQ 9.4.2, vous pouvez également configurer les gestionnaires de files d'attente sous licence IBM MQ pour utiliser les fonctionnalités Native HA et Cross-Region Replication, en ajoutant des licences supplémentaires à un gestionnaire de files d'attente.

     a. Remplacer les annotations sous `spec.template.metadata` par les valeurs spécifiées sous IBM MQ ou IBM MQ for Non-Production Environment dans «IBM MQ annotations sur la licence du conteneur», à la page 289.

     b. Complétez l'étape «3», à la page 193 dans «Configuration de Native HA et de Cross-Region Replication sur IBM MQ gestionnaires de files d'attente sous licence dans Kubernetes», à la page 192. Cette étape ajoute une CR `IBMLicensingDefinition` qui ajoute des annotations de licence supplémentaires à votre gestionnaire de file d'attente pour les fonctions Native HA et Cross-Region Replication

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  labels:
    app.kubernetes.io/component: integration
    app.kubernetes.io/instance: exampleqm
    app.kubernetes.io/name: ibm-mq
    app.kubernetes.io/version: 9.4.3.0
  name: exampleqm-ibm-mq
spec:
  persistentVolumeClaimRetentionPolicy:
    whenDeleted: Retain
    whenScaled: Retain
  updateStrategy:
    type: OnDelete
  podManagementPolicy: Parallel
  replicas: 3
  revisionHistoryLimit: 10
  selector:
    matchLabels:
      app.kubernetes.io/component: integration
      app.kubernetes.io/instance: exampleqm
      app.kubernetes.io/name: ibm-mq
  serviceName: qm
  template:
    metadata:
      # These annotations are critical for the IBM License Service to charge you correctly
      annotations:
        productChargedContainers: All
        productID: 208423bb063c43288328b1d788745b0c
        productMetric: PROCESSOR_VALUE_UNIT
        productName: IBM MQ Advanced
        productVersion: 9.4.3.0
      labels:
        app.kubernetes.io/component: integration
        app.kubernetes.io/instance: exampleqm
        app.kubernetes.io/name: ibm-mq
        app.kubernetes.io/version: 9.4.3.0
        statefulSetName: exampleqm-ibm-mq
    spec:
      # Create an anti-affinity rule, so the Native HA replicas don't end up on the same Node
      affinity:
        podAntiAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
          - labelSelector:
```

```
              matchExpressions:
              - key: statefulSetName
                operator: In
                values:
                - exampleqm-ibm-mq
            topologyKey: kubernetes.io/hostname
      containers:
      # Environment variables to supply to MQ container
      - env:
        - name: LICENSE
          # if you accept the license, change the following value to "accept"
          value: view
        - name: MQ_QMGR_NAME
          value: EXAMPLEQM
        - name: MQ_MULTI_INSTANCE
          value: "false"
        - name: MQ_ENABLE_METRICS
          value: "true"
        - name: MQ_ENABLE_EMBEDDED_WEB_SERVER
          value: "false"
        - name: MQ_LOGGING_CONSOLE_FORMAT
          value: basic
        - name: DEBUG
          value: "false"
        - name: MQ_ENABLE_TRACE_STRMQM
          value: "false"
        - name: MQ_EPHEMERAL_PREFIX
          value: /run/mqm
        - name: MQ_GRACE_PERIOD
          value: "29"
        - name: MQ_NATIVE_HA
          value: "true"
        # The image to pull.  This example uses a prebuilt MQ Advanced image from IBM
Container Registry
        image: cp.icr.io/cp/ibm-mqadvanced-server:9.4.3.0-r1
        imagePullPolicy: IfNotPresent
        livenessProbe:
          exec:
            command:
            - chkmqhealthy
          failureThreshold: 1
          periodSeconds: 10
          successThreshold: 1
          timeoutSeconds: 5
        name: qmgr
        ports:
        - containerPort: 1414
          protocol: TCP
        - containerPort: 9157
          protocol: TCP
        - containerPort: 9414
          protocol: TCP
        readinessProbe:
          exec:
            command:
            - chkmqready
          failureThreshold: 1
          periodSeconds: 5
          successThreshold: 1
          timeoutSeconds: 3
        # CPU and memory resources to allocate
        resources:
          limits:
            cpu: "1"
            memory: 1Gi
          requests:
            cpu: "1"
            memory: 1Gi
        # Container security context.  Disable all root capabilities and privileges.
        securityContext:
          allowPrivilegeEscalation: false
          capabilities:
            drop:
            - ALL
          privileged: false
          readOnlyRootFilesystem: true
          runAsNonRoot: true
        startupProbe:
          exec:
            command:
            - chkmqstarted
          failureThreshold: 24
```

```
          periodSeconds: 5
          successThreshold: 1
          timeoutSeconds: 5
        terminationMessagePath: /run/termination-log
        terminationMessagePolicy: File
        volumeMounts:
        # Mount the main queue manager persistent volume.  Sym-linked from /var/mqm
        - mountPath: /mnt/mqm
          name: data
        # Mount space for temporary files
        - mountPath: /run
          name: scratch
        - mountPath: /tmp
          name: tmp
        # Mount the "default" key for TLS
        - mountPath: /etc/mqm/pki/keys/default
          name: default
          readOnly: true
        # Mount the key to use for Native HA replication traffic
        - mountPath: /etc/mqm/ha/pki/keys/ha
          name: ha-tls
          readOnly: true
        # Mount MQSC to be applied when the queue manager starts
        - mountPath: /etc/mqm/example.mqsc
          name: cm-example-nativeha-configmap
          readOnly: true
          subPath: example.mqsc
        # Mount an INI file to be applied when the queue manager starts
        - mountPath: /etc/mqm/example.ini
          name: cm-example-nativeha-configmap
          readOnly: true
          subPath: example.ini
      restartPolicy: Always
      securityContext: {}
      serviceAccount: exampleqm-ibm-mq
      serviceAccountName: exampleqm-ibm-mq
      automountServiceAccountToken: false
      terminationGracePeriodSeconds: 30
      volumes:
      - emptyDir:
          sizeLimit: 100M
        name: scratch
      # Temporary directory is used by default to store runmqras output, so make it larger
      - emptyDir:
          sizeLimit: 2Gi
        name: tmp
      - name: default
        secret:
          defaultMode: 288
          items:
          - key: tls.key
            path: tls.key
          - key: tls.crt
            path: tls.crt
          - key: ca.crt
            path: ca.crt
          secretName: example-qm-tls
      - name: ha-tls
        secret:
          defaultMode: 288
          secretName: example-qm-tls
      - configMap:
          defaultMode: 420
          items:
          - key: example.mqsc
            path: example.mqsc
          - key: example.ini
            path: example.ini
          name: example-nativeha-configmap
        name: cm-example-nativeha-configmap
  volumeClaimTemplates:
  - apiVersion: v1
    kind: PersistentVolumeClaim
    metadata:
      labels:
        app.kubernetes.io/component: integration
        app.kubernetes.io/instance: exampleqm
        app.kubernetes.io/name: ibm-mq
      name: data
      namespace: mq-test
    spec:
      accessModes:
```

```
      - ReadWriteOnce
      resources:
        requests:
          storage: 2Gi
      volumeMode: Filesystem
```

Notez que le " `Secret` " *exemple-qm-tls* a été créé à l'étape " «1», à la page 182 et que le " `ConfigMap` " *exemple-nativeha-configmap* " a été créé à l'étape " «2», à la page 182.

Un volume persistant est créé pour chaque instance. La classe de stockage par défaut configurée dans votre cluster " Kubernetes est utilisée. Si aucune classe de stockage n'est configurée par défaut ou si vous souhaitez utiliser une autre classe de stockage, ajoutez `storageClassName:` *`storage_class_name`* sous `spec.volumeClaimTemplates[0].spec`.

Les trois pods d'un gestionnaire de files d'attente Native HA répliquent les données sur le réseau. Ce lien n'est pas crypté par défaut, mais cet exemple utilise le certificat du gestionnaire de file d'attente pour crypter le trafic. Vous pouvez spécifier un autre certificat pour plus de sécurité. Le secret TLS de l'AH natif doit être un secret TLS " Kubernetes, qui a une structure particulière (par exemple, la clé privée doit s'appeler *tls.key*

Notez que le " `updateStrategy` est remplacé par le " `OnDelete`, ce qui a une incidence sur la manière dont les mises à jour sont diffusées. Pour plus d'informations, 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 45

7. 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 est dans l'état " `Running` avant de continuer, par exemple en listant les pods du gestionnaire de file d'attente :

   ```
   kubectl get pods --selector app.kubernetes.io/instance=exampleqm
   ```

   Idéalement, vous devriez vous connecter au gestionnaire de file d'attente pour confirmer qu'il fonctionne. Cependant, la connexion au gestionnaire de file d'attente depuis l'extérieur du cluster " Kubernetes n'entre pas dans le cadre de ces instructions, car elle peut varier de manière significative entre les distributions " Kubernetes et les environnements en nuage.

8. 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

      Réexécuter la commande précédente :

      ```
      kubectl 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 :

      ```
      kubectl delete pod exampleqm-ibm-mq-<active_pod_value>
      ```

   c) Afficher de nouveau le statut du pod

      Exécutez ensuite la commande suivante :

      ```
      kubectl 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 :

      ```
      kubectl exec -t <qmgr_pod_name> -- 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)
```

Résultats

Vous avez déployé avec succès un gestionnaire de files d'attente avec une haute disponibilité native et une authentification TLS mutuelle, et vérifié qu'il se rétablit automatiquement en cas de défaillance du pod actif.

V 9.4.2 Exemple : Configuration de la réplication interrégionale native HA dans Kubernetes

Cet exemple déploie un gestionnaire de file d'attente en utilisant la fonctionnalité native de réplication inter-régions haute disponibilité (Native HA CRR) dans l' Kubernetes, en utilisant directement les ressources YAML.

Avant de commencer

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

- Créer un espace de noms (Kubernetes) dans chacun des clusters où les groupes Live et Recovery seront déployés.
- Sur la ligne de commande, pouvoir se connecter à l'un ou l'autre des clusters d' Kubernetes, et passer à l'espace de noms correspondant.
- Assurez-vous d'avoir suivi les étapes décrites dans «Préparation de l'utilisation de 'Kubernetes par la création d'un secret d'extraction», à la page 88 dans les espaces de noms ci-dessus.

Pourquoi et quand exécuter cette tâche

Cet exemple déploie deux gestionnaires de file d'attente en utilisant la fonctionnalité Native HA CRR dans l' Kubernetes, en utilisant directement les ressources YAML. L'un des gestionnaires de file d'attente Native HA sera Live et l'autre Recovery. Cet exemple s'appuie sur les fichiers YAML créés dans «Exemple : Configuration de Native HA dans " Kubernetes», à la page 182.

Cet exemple fournit les modifications requises en plus de celles définies pour NativeHA: ConfigMap, Service, StatefulSet, et Route.

Procédure

1. Créez une nouvelle configuration ou définissez une nouvelle définition de données.ini dans l' ConfigMap e existant (voir l'étape 2 dans l'exemple Native HA) pour définir le fichier INI d'un gestionnaire de file d'attente Live ou Recovery.

   Créer un fichier Kubernetes ConfigMap contenant les définitions pour configurer les groupes suivants :

   a) Un groupe natif à haute disponibilité comme récupération :

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: example-nativeha-configmap
data:
...
  example-crr: |
    NativeHALocalInstance:
      GroupName=beta
      GroupRole=Recovery
```

```
      GroupLocalAddress=(9415)
      GroupCipherSpec=ANY_TLS12_OR_HIGHER
```

Lorsque le groupe Recovery est déployé, l'adresse de réplication de ce groupe peut être déterminée et fournie dans l' ConfigMap e pour le groupe Live. Par exemple, cela peut être fait en émettant la commande suivante vers le cluster de récupération déployé :

```
kubectl get route example-ibm-mq-crr-route -n <recovery-qm-namespace> -o
jsonpath="{.spec.host}"
```

b) Un groupe natif HA CRR en tant que Live (pointant vers le groupe Recovery) :

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: example-nativeha-configmap
data:
...
  example-crr: |
    NativeHALocalInstance:
      GroupName=alpha
      GroupRole=Live
      GroupLocalAddress=(9415)
      GroupCipherSpec=ANY_TLS12_OR_HIGHER
    NativeHARecoveryGroup:
      Enabled=Yes
      GroupName=beta
      ReplicationAddress=<address.to.recovery.cluster>(443)
```

2. Exposer un port d' ClusterIP s de service supplémentaire pour que le trafic CRR HA natif circule dans le groupe CRR. Cela peut être fait en créant une nouvelle définition YAML, ou en étendant la définition du Service à l'étape 3 de l'exemple Native HA :

```
apiVersion: v1
kind: Service
metadata:
...
  name: exampleqm-ibm-mq
spec:
  ports:
...
  - name: ha-crr
    port: 9415
    protocol: TCP
    targetPort: 9415
...
  selector:
...
  type: ClusterIP
```

3. Définissez une nouvelle Route dans le Cluster pour permettre la communication de l'autre groupe Native HA CRR pour atteindre ce groupe Native HA CRR et activer le « passthrough » TLS vers le nouveau port Service ClusterIP défini à l'étape 2 :

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: "exampleqm-ibm-mq-crr-route"
spec:
  to:
    kind: Service
    name: exampleqm-ibm-mq
  port:
    targetPort: 9415
  tls:
    termination: passthrough
```

4. Modifiez l' StatefulSet, créée à l'étape 6 de l'exemple Native HA, pour vous assurer que les exigences suivantes sont respectées :

   a) La nouvelle définition INI d' ConfigMap, pour Native HA CRR, est montée de manière à ce que le gestionnaire de file d'attente puisse la récupérer au démarrage :

```
...
        volumeMounts:
...
        # Mount the NativeHA CRR INI file to be applied when the queue manager starts
        - mountPath: /etc/mqm/example-crr.ini
          name: cm-example-nativeha-configmap
          readOnly: true
          subPath: example-crr.ini
...
      volumes:
...
      - configMap:
          defaultMode: 420
          items:
...
          - key: example-crr.ini
            path: example-crr.ini
          name: example-nativeha-configmap
        name: cm-example-nativeha-configmap
```

b) La communication entre les deux groupes Native HA CRR doit être cryptée. Par conséquent, les déclarations de la clé secrète TLS pour le groupe CRR actuel d' NativeHA, et le certificat de confiance pour l'autre groupe CRR d' NativeHA, doivent être définies et montées dans le répertoire d' `/etc/mqm/groupha` .

En utilisant les noms de groupe alpha (pour Live) et beta (pour Recovery), les modifications d'exemple pour l' StatefulSet e YAML pour le groupe Live seraient :

```
...
        volumeMounts:
...
          - mountPath: /etc/mqm/groupha/pki/keys/ha-group
            name: groupha-tls
            readOnly: true
          - name: groupha-tls-trust-0
            mountPath: /etc/mqm/groupha/pki/trust/0
            readOnly: true
...
      volumes:
...
      - name: groupha-tls
        secret:
          secretName: nha-crr-secret-alpha
          defaultMode: 288
      - name: groupha-tls-trust-0
        secret:
          secretName: nha-crr-secret-beta
          defaultMode: 420
          items:
            - key: tls.crt
              path: native-crr-nha-crr-secret-beta-0-tls.crt
```

5. Confirmez que les groupes Native HA CRR sont en cours d'exécution, synchronisés, connectés et en quorum, en utilisant la commande suivante **dspmq** sur les pods Ready and Running de l'instance du gestionnaire de file d'attente Live and Recovery :

```
kubectl exec -t <qmgr_pod_name> -- dspmq -o nativeha -g -m EXAMPLEQM
```

Le statut doit indiquer que les deux groupes Native HA CRR sont connectés et agissent en tant que groupes Live et Recovery, par exemple :

```
QMNAME(EXAMPLEQM) ROLE(Active) INSTANCE(testmqqm-ibm-mq-0) INSYNC(yes) QUORUM(3/3)
GRPLSN(<0:0:10:30665>) GRPNAME(alpha) GRPROLE(Live)
 GRPNAME(alpha) GRPROLE(Live) GRPADDR(Unknown) GRPVER(9.4.2.0) GRSTATUS(Normal)
RCOVLSN(<0:0:10:30665>) RCOVTIME(2025-02-17T16:25:16.527089Z) INITLSN(<0:0:9:4684>)
INITTIME(2025-02-17T16:25:01.226650Z) LIVETIME(2025-02-17T16:25:02.974677Z)
ALTDATE(2025-02-17) ALTTIME(16.25.19)
 GRPNAME(beta) GRPROLE(Recovery) GRPADDR(<address.to.recovery.cluster>) GRPVER(9.4.2.0)
CONNGRP(yes) GRSTATUS(Normal) RCOVLSN(<0:0:10:30665>) RCOVTIME(2025-02-17T16:25:16.527089Z)
BACKLOG(0) INSYNC(yes) SYNCTIME(2025-02-17T16:29:38.643916Z) ALTDATE(2025-02-17)
ALTTIME(16.29.29)
```

Résultats

Vous avez déployé avec succès deux groupes de gestionnaires de file d'attente avec une haute disponibilité native et une authentification TLS au sein des groupes et entre eux. Vous avez également vérifié que les groupes Live et Recovery sont connectés et effectuent une réplication inter-régions.

V 9.4.2 Kubernetes Configuration de Native HA et de Cross-Region Replication sur IBM MQ gestionnaires de files d'attente sous licence dans Kubernetes

De IBM MQ 9.4.2, IBM MQ Les gestionnaires de file d'attente sous licence peuvent être configurés pour utiliser les fonctions de réplication transrégionale (CRR) de Native HA en ajoutant une licence supplémentaire à un gestionnaire de file d'attente.

Avant de commencer

Remarque : Ces instructions ne s'appliquent que si vous déployez des gestionnaires de file d'attente sur Kubernetes. Si vous avez déployé le site IBM MQ Operator, consultez plutôt le site «Configurer Native HA et Cross-Region Replication sur IBM MQ Queue Managers sous licence en utilisant la fonction IBM MQ Operator», à la page 119 .

- Le site IBM License Service est nécessaire pour utiliser les fonctions Native HA CRR sur les gestionnaires de file d'attente sous licence IBM MQ . Pour plus d'informations, voir Installer License Service sur le cluster Kubernetes dans la documentation IBM Cloud Pak foundational services .
- Sur le site `StatefulSet` qui gère votre pod de gestion des files d'attente, le label `metadata.labels.app.kubernetes.io/instance` doit être défini sur une valeur unique.

Procédure

1. Sur la ligne de commande, connectez-vous au cluster Kubernetes puis passez à l'espace de noms dans lequel votre gestionnaire de files d'attente est déployé.
2. Ajoutez les annotations pertinentes aux annotations de votre gestionnaire de file d'attente.

   Notez que les instructions de cette étape concernent la modification d'un gestionnaire de file d'attente existant. Pour les nouveaux gestionnaires de file d'attente, ajouter les annotations lors de la création.

   Important : La mise à jour des annotations du gestionnaire de file d'attente nécessite un redémarrage du pod.

   a) Identifiez le site `StatefulSet` du gestionnaire de files d'attente que vous souhaitez mettre à jour :

   ```
   kubectl get statefulset
   ```

   b) Ouvrez le site `StatefulSet` du gestionnaire de file d'attente pour le modifier :

   ```
   kubectl edit statefulset <QUEUE-MANAGER-STATEFUL-SET>
   ```

   c) Ajoutez l'annotation correspondant à votre composante imposable sur votre site `StatefulSet`. Notez qu'il y a d'autres annotations sous `spec.template.metadata`; ne modifiez que les annotations suivantes :

   - Pour IBM MQ :

     ```
     spec:
       template:
         metadata:
           productID: c661609261d5471fb4ff8970a36bccea
           productName: IBM MQ
     ```

   - Pour IBM MQ for Non-Production Environment :

     ```
     spec:
       template:
     ```

```
    metadata:
      productID: 151bec68564a4a47a14e6fa99266deff
      productName: IBM MQ for Non-Production Environment
```

d) Enregistrez vos modifications. Un message similaire au suivant s'affiche sur la ligne de commande :

```
statefulset.apps/<QUEUE-MANAGER-STATEFUL-SET> edited
```

e) Supprimez tous les pods de gestion de file d'attente qui sont gérés par le site `StatefulSet` que vous avez modifié.

f) Répétez ces étapes pour d'autres gestionnaires de file d'attente si nécessaire.

3. Ajoutez la CR `IBMLicensingDefinition` pertinente qui ajoutera des annotations de licence supplémentaires à votre gestionnaire de file d'attente pour les fonctionnalités Native HA CRR.

a) Créer un fichier YAML `IBMLicensingDefinition` .

Dans les exemples suivants, modifiez ces valeurs :

- `<IBMLicensingDefinition-UNIQUE-NAME>`: Un nom unique pour le CR `IBMLicensingDefinition` . Une bonne pratique consiste à inclure le nom de votre gestionnaire de file d'attente dans ce nom.
- `<QUEUE-MANAGER-NAMESPACE>`: L'espace de noms dans lequel votre gestionnaire de file d'attente est déployé.
- `<QUEUE-MANAGER-NAME>`: La valeur de `metadata.labels.app.kubernetes.io/instance` de votre gestionnaire de file d'attente.
- Exemple de YAML pour un `IBMLicensingDefinition` CR destiné à être utilisé en production :

```
apiVersion: operator.ibm.com/v1
kind: IBMLicensingDefinition
metadata:
  name: <IBMLicensingDefinition-UNIQUE-NAME>
  namespace: <QUEUE-MANAGER-NAMESPACE>
spec:
  action: cloneModify
  condition:
    metadata:
      labels:
        app.kubernetes.io/instance: <QUEUE-MANAGER-NAME>
  scope: cluster
  set:
    productID: 46a57a4a66ee4b904e9b5544b53a2ba2
    productName: IBM MQ Native HA and Cross-Region Replication Add-on
```

- Exemple de YAML pour un CR `IBMLicensingDefinition` destiné à une utilisation hors production :

```
apiVersion: operator.ibm.com/v1
kind: IBMLicensingDefinition
metadata:
  name: <IBMLicensingDefinition-UNIQUE-NAME>
  namespace: <QUEUE-MANAGER-NAMESPACE>
spec:
  action: cloneModify
  condition:
    metadata:
      labels:
        app.kubernetes.io/instance: <QUEUE-MANAGER-NAME>
  scope: cluster
  set:
    productID: 525b1331a74d4a209d9f6e9c908e374e
    productName: IBM MQ Native HA and Cross-Region Replication Add-on for Non-
Production Env
```

b) Appliquer le fichier YAML `IBMLicensingDefinition` :

```
kubectl apply -f <IBMLicensingDefinition-UNIQUE-NAME>.yaml
```

c) Répétez ces étapes pour d'autres gestionnaires de file d'attente si nécessaire.

Kubernetes Déploiement et configuration des gestionnaires de file d'attente à l'aide de l'exemple de diagramme 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 savoir comment utiliser " Helm pour déployer votre propre image de conteneur " IBM MQ, voir " Exemple de graphique 'IBM MQ 'Helm .

Référence associée

«Prise en charge de IBM MQ dans les conteneurs», à la page 29
Toutes les fonctions IBM MQ ne sont pas disponibles et prises en charge de la même manière dans les conteneurs.

Podman Déploiement d'un gestionnaire de file d'attente sur 'Podman

Cet exemple exécute une image préconstruite " IBM MQ Advanced Container pour déployer un gestionnaire de file d'attente dans " Podman. Le même exemple peut être utilisé avec " Docker.

Avant de commencer

Pour valider la configuration TLS à la fin de cet exemple :

- Installez le client IBM MQ. Vous avez besoin des commandes '**amqsputc** et '**amqsgetc** pour tester l'envoi et la réception de messages. Ces commandes peuvent être installées dans le cadre du client " IBM MQ
  - Pour 'Windows et 'Linux, installez le client redistribuable 'IBM MQ pour votre système d'exploitation à partir de 'https://ibm.biz/mq94redistclients.
  - Pour le " Mac, téléchargez et configurez le " IBM MQ MacOS Toolkit à partir du " https://developer.ibm.com/tutorials/mq-macos-dev/.
- Notez qu'il n'y a pas d'image 'ARM64 'IBM MQ Advanced Container préconstruite.

Cet exemple utilise le " Podman, mais peut être utilisé avec le " Docker

Pourquoi et quand exécuter cette tâche

Cet exemple exécute une image préconstruite " IBM MQ Advanced Container pour déployer un gestionnaire de file d'attente dans " Podman. Il détaille les étapes supplémentaires nécessaires à la configuration de TLS pour la messagerie, ainsi qu'à la configuration du " IBM MQ Console et du " REST API. Vous pouvez choisir de configurer l'un ou l'autre de ces éléments, en sautant les étapes qui ne sont pas nécessaires pour répondre à vos besoins.

Lorsque le gestionnaire de files d'attente est en cours d'exécution, cet exemple utilise les programmes d'exemple " IBM MQ exécutés sur une machine extérieure à " Podman, comme votre ordinateur portable, pour valider l'utilisation de TLS mutuel entre le client d'exemple et le gestionnaire de files d'attente pour envoyer et recevoir des messages. Il valide également la connexion au " IBM MQ Console et les appels administratifs et de messagerie du " REST API

Remarque : Vous pouvez à la place exécuter une image de conteneur préconstruite " IBM MQ Advanced for Developers, qui dispose d'une configuration par défaut supplémentaire pour les développeurs. Voir «IBM MQ Advanced for Developers image de conteneur», à la page 300.

Procédure

- **Configurer TLS pour la messagerie entre un client et le gestionnaire de file d'attente**

  a) Créez des certificats comme décrit dans le " «Création d'une infrastructure PKI autosignée à l'aide de OpenSSL», à la page 107, en suivant toutes les étapes mais en sautant la commande " **kubectl**

     a. Créez un répertoire, par exemple appelé " `qm-pki`, contenant les fichiers de clés et de certificats générés pour le gestionnaire de files d'attente :

        i) `example-qm.key`

        ii) `example-qm.crt`

        iii) `ca.crt`

        L'utilisateur qui exécute le conteneur (uid 1001) a besoin d'un accès en lecture à '`example-qm.key`. Par défaut, " OpenSSL la crée de manière à ce que l'utilisateur qui crée la clé y ait accès.

     b. Créez un répertoire, par exemple appelé " `mq-client`, contenant les fichiers générés pour le client :

        i) `example-app1.p12`

        ii) Pour le " Mac, vous avez également besoin du " `example-app1-chain.crt`

  b) Créez un fichier contenant les commandes MQSC, par exemple '**`example-tls.mqsc`**, pour créer une nouvelle file d'attente et un canal 'SVRCONN, et pour ajouter un enregistrement d'authentification de canal qui autorise l'accès au canal.

     ```
     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)
     ```

     La commande MQSC définit un canal appelé " `MTLS.SVRCONN` et une file d'attente appelée " `EXAMPLE.QUEUE`. Le canal est configuré pour n'autoriser l'accès qu'aux clients qui présentent un certificat dont le "nom commun" est " `example-app1`". Il s'agit du nom commun utilisé dans l'un des certificats créés à l'étape précédente. Les connexions sur ce canal avec ce nom commun sont associées à un identifiant d'utilisateur " `app1`, qui est autorisé à se connecter au gestionnaire de file d'attente et à accéder à la file d'attente de l'exemple.

  c) Créez un fichier INI, par exemple " `example-tls.ini`, qui active une politique de sécurité telle que l'identifiant de l'utilisateur " `app1` n'a pas besoin d'exister dans un registre d'utilisateurs externe - il n'existe qu'en tant que nom dans cette configuration.

     ```
     Service:
         Name=AuthorizationService
         EntryPoints=14
         SecurityPolicy=UserExternal
     ```

- **Configurer le " IBM MQ Console et le " REST API**

  Dans cet exemple, vous configurez un registre de base dans le fichier " `mqwebuser.xml`. The user names, passwords, and roles in the XML file are used to authenticate and authorize users of the IBM MQ Console and REST API. Pour plus d'informations, voir Configuration d'un registre de base pour les IBM MQ Console et 'REST API.

  Vous devez vous demander si l'utilisation d'un registre de base est appropriée pour vos environnements de production. Voir " sécurité IBM MQ Console et " REST API .

  a) Créez un fichier appelé " `mqwebuser.xml`, pour configurer le " IBM MQ Console.

Dans cet exemple, un registre de base est créé, contenant des utilisateurs ayant les rôles d'administrateur web et d'utilisateur web.

Vous **devez** remplacer *ADMIN_PASSWORD* et *APP_PASSWORD* par vos propres valeurs. Vous pouvez également modifier les noms d'utilisateur admin et app si vous le souhaitez, mais notez que le nom d'utilisateur app doit correspondre au nom d'utilisateur app spécifié dans la commande MQSC précédente. Notez que les noms d'utilisateurs sont utilisés deux fois dans l'exemple.

```
<?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>
    <enterpriseApplication id="com.ibm.mq.rest">
        <application-bnd>
            <security-role name="MQWebAdmin">
                <group name="MQWebAdminGroup" realm="defaultRealm"/>
            </security-role>
            <security-role name="MQWebUser">
                <group name="MQWebMessaging" realm="defaultRealm"/>
            </security-role>
        </application-bnd>
    </enterpriseApplication>
    <basicRegistry id="basic" realm="defaultRealm">
        <user name="admin" password="ADMIN_PASSWORD"/>
        <user name="app1" password="APP_PASSWORD"/>
        <group name="MQWebAdminGroup">
            <member name="admin"/>
        </group>
        <group name="MQWebMessaging">
            <member name="app1"/>
        </group>
    </basicRegistry>
    <sslDefault sslRef="mqDefaultSSLConfig"/>
</server>
```

b) Connectez-vous au registre des conteneurs.

Obtenir une clé d'habilitation, puis l'utiliser pour se connecter au registre du conteneur. Voir «Préparation à l'utilisation de 'Podman ou 'Docker en tirant l'image du conteneur préconstruit», à la page 89.

c) Exécuter le gestionnaire de file d'attente dans " Podman.

Choisissez l'une des options suivantes :

– **Option 1 : Exécuter le gestionnaire de files d'attente dans " Podman en montant le répertoire contenant les certificats du gestionnaire de files d'attente.**

La commande suivante comprend la configuration de la messagerie TLS ainsi que la configuration des IBM MQ Console et 'REST API. Si vous avez ignoré une partie de la configuration décrite précédemment, parce qu'elle n'était pas nécessaire pour vos besoins, supprimez les paramètres correspondant aux sections que vous avez ignorées.

Assurez-vous que la commande pointe vers l'emplacement correct des fichiers et du répertoire créés précédemment.

Pour exécuter le gestionnaire de file d'attente dans " Podman, exécutez la commande suivante :

```
podman run --env MQ_ENABLE_EMBEDDED_WEB_SERVER=true  \
-v "./mqwebuser.xml:/etc/mqm/web/installations/Installation1/servers/mqweb/
mqwebuser.xml" \
-v "./example-tls.ini:/etc/mqm/example-tls.ini"  \
-v "./example-tls.mqsc:/etc/mqm/example-tls.mqsc"  \
-v "./qm-pki:/etc/mqm/pki/keys/qmlabel" --env LICENSE=accept --env MQ_QMGR_NAME=QM1 \
```

```
--publish 1414:1414 --publish 9443:9443 --detach --name QM1 cp.icr.io/cp/ibm-
mqadvanced-server:9.4.1.0-r1
```

La commande extrait l'image du gestionnaire de file d'attente 'IBM MQ Advanced et démarre le conteneur.

– **Option 2 : Exécuter le gestionnaire de files d'attente dans " Podman en utilisant les secrets de " Podman**

Vous pouvez monter les certificats du gestionnaire de file d'attente en utilisant les secrets " Podman, plutôt que de monter directement les répertoires qui les contiennent dans le conteneur.

Remarque : Cette option ne s'applique pas lorsque vous utilisez " Docker, à moins que vous n'utilisiez " Docker Swarm, car les secrets " Docker ne peuvent être utilisés qu'avec " Docker Swarm. Voir la commande docker secret create dans dockerdocs.

Procédez comme suit :

a. Créez des secrets " Podman pour chacun des fichiers TLS du gestionnaire de file d'attente. Par exemple, à partir du répertoire de création " `qm-pki`:

```
podman secret create example-qm.key ./example-qm.key
podman secret create example-qm.crt ./example-qm.crt
podman secret create ca.crt ./ca.crt
```

La commande suivante comprend la configuration de la messagerie TLS ainsi que la configuration des IBM MQ Console et 'REST API. Si vous avez ignoré une partie de la configuration décrite précédemment, parce qu'elle n'était pas nécessaire pour vos besoins, supprimez les paramètres correspondant aux sections que vous avez ignorées.

Assurez-vous que la commande pointe vers l'emplacement correct des fichiers et du répertoire créés précédemment.

Pour exécuter le gestionnaire de files d'attente dans " Podman, exécutez la commande suivante :

```
podman run --env MQ_ENABLE_EMBEDDED_WEB_SERVER=true \
-v "./mqwebuser.xml:/etc/mqm/web/installations/Installation1/servers/mqweb/
mqwebuser.xml" \
-v "./example-tls.ini:/etc/mqm/example-tls.ini" \
-v "./example-tls.mqsc:/etc/mqm/example-tls.mqsc" \
--secret example-qm.key,target=/etc/mqm/pki/keys/qmlabel/example-
qm.key,uid=1001,mode=0440 \
--secret example-qm.crt,target=/etc/mqm/pki/keys/qmlabel/example-qm.crt,uid=1001 \
--secret ca.crt,target=/etc/mqm/pki/keys/qmlabel/ca.crt,uid=1001 --env
LICENSE=accept --env MQ_QMGR_NAME=QM1 \
--publish 1414:1414 --publish 9443:9443 --detach --name QM1 cp.icr.io/cp/ibm-
mqadvanced-server:9.4.1.0-r1
```

Remarque : Si l'utilisateur sous lequel le conteneur est exécuté est remplacé par un autre utilisateur, modifiez l'UID pour qu'il corresponde.

Un gestionnaire de file d'attente est en cours d'exécution dans " Podman.

```
podman ps
```

Validation

- **Valider la messagerie avec TLS**

a) Créez un fichier INI client, appelé " `mqclient.ini`, dans le répertoire client créé ci-dessus, par exemple " `mq-client`. Spécifiez le *MOT DE PASSE* utilisé pour le magasin de clés p12 à l'étape 1 de la configuration de TLS pour la messagerie entre un client et le gestionnaire de files d'attente:

```
Channels:
  ChannelDefinitionDirectory=.
  ChannelDefinitionFile=CCDT.json
SSL:
  OutboundSNI=HOSTNAME
  SSLKeyRepository=example-app1.p12
  SSLKeyRepositoryPassword=PASSWORD
```

b) Dans le même répertoire, créez un fichier CCDT dont le nom correspond à la référence du fichier " `mqclient.ini` défini précédemment, par exemple " `CCDT.json`:

```
{
    "channel":
    [
        {
            "name": "MTLS.SVRCONN",
            "clientConnection":
            {
                "connection":
                [
                    {
                        "host": "localhost",
                        "port": 1414
                    }
                ],
                "queueManager": "QM1"
            },
            "transmissionSecurity":
            {
              "cipherSpecification": "ANY_TLS13",
              "certificateLabel": "example-app1"
            },
            "type": "clientConnection"
        }
   ]
}
```

c) Sous Mac, exécutez la commande suivante :

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

d) Dans le même répertoire, par exemple '`mq-client`, utilisez l'exemple d'application pour envoyer un message :

```
amqsputc EXAMPLE.QUEUE QM1
```

e) Utilisez l'exemple d'application pour obtenir un message :

```
amqsgetc EXAMPLE.QUEUE QM1
```

- **Valider le " IBM MQ Console**

  Se connecter à la console dans un navigateur :

  a) Ouvert https://localhost:9443/ibmmq/console

  b) Acceptez le certificat autosigné.

  c) Se connecter :

  – Utilisateur : `admin`

  – Mot de passe : La valeur que vous avez précédemment spécifiée pour *ADMIN_PASSWORD*.

- **Valider le " REST API**

  Dans les exemples suivants, spécifiez les valeurs que vous avez précédemment configurées pour *ADMIN_PASSWORD* et *APP_PASSWORD*.

  Exemple d'appel au " administrative REST API pour obtenir le statut du gestionnaire de file d'attente :

```
curl -k -X GET -u admin:ADMIN_PASSWORD https://localhost:9443/ibmmq/rest/v1/admin/qmgr/QM1
```

  Exemple d'appel à la messagerie " REST API pour placer puis récupérer un message dans une file d'attente.

  Insérez un message :

```
curl -k -X POST -u app1:APP_PASSWORD \
-H 'Content-Type: application/json' -H "ibm-mq-rest-csrf-token: value" \
-d 'msg from REST' https://localhost:9443/ibmmq/rest/v2/messaging/qmgr/QM1/queue/
EXAMPLE.QUEUE/message
```

Parcourir un message :

```
curl -k -X GET -u app1:APP_PASSWORD \
-H "ibm-mq-rest-csrf-token: value" https://localhost:9443/ibmmq/rest/v2/messaging/qmgr/QM1/
queue/EXAMPLE.QUEUE/message
```

Remarque : L'opération " REST API " **GET** est en fait une opération de navigation " IBM MQ - le message est laissé dans la file d'attente. Pour faire un 'IBM MQ '**GET**', utilisez le verbe 'REST API '**DELETE** '.

Traitement des incidents

- Si les exemples d'application " **amqputc** ou " **amqgetc** échouent, consultez les journaux d'erreur à " `~/IBM/MQ/data/errors/AMQERR01.LOG` sur la machine cliente, par exemple votre ordinateur portable.
- Affichez les journaux d'erreurs du gestionnaire de file d'attente à leurs emplacements habituels, en exécutant le conteneur dans 'Podman.

  Exemple :

  ```
  podman exec -it QM1 /bin/bash
  ```

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 200
2. «Extraction de la configuration du gestionnaire de files d'attente», à la page 201
3. Facultatif : «Facultatif : extraction et acquisition des clés et certificats du gestionnaire de files d'attente», à la page 201
4. Facultatif : «Facultatif : configuration de LDAP», à la page 203
5. Facultatif : «Facultatif : modification des adresses IP et noms d'hôte dans la configuration IBM MQ», à la page 211
6. «Mise à jour de la configuration du gestionnaire de files d'attente pour un environnement de conteneur», à la page 212
7. «Sélection de l'architecture haute disponibilité cible pour IBM MQ exécuté en conteneurs», à la page 216
8. «Création des ressources du gestionnaire de files d'attente», à la page 216
9. «Création du gestionnaire de files d'attente dans Red Hat OpenShift», à la page 218
10. «Vérification du nouveau déploiement de conteneur», à la page 222

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 199.

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 28.
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 164.
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 41.

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 203.

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 :

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

2. 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')` :

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

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 203.

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` :

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

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.

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

2. 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 218).

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 41.
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 201, 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 216, 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 :

  Remarque : Cet exemple utilise une licence IBM MQ Advanced , mais vous pouvez également utiliser une licence IBM MQ . Pour plus d'informations, voir «Configurer les gestionnaires de files d'attente avec les annotations de licence IBM MQ en utilisant la fonction IBM MQ Operator», à la page 166.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: qm1
spec:
  version: 9.4.3.0-r1
  license:
    accept: true
    license: L-CYPF-CRPF3H
    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: 9.4.3.0-r1
  license:
    accept: true
    license: L-CYPF-CRPF3H
    use: "Production"
```

Correspond aux informations de version et de licence du déploiement. Pour les personnaliser, utilisez les informations fournies dans la rubrique «mq.ibm.com/v1beta1: Versions actuelles de la licence», à la page 256.

```
  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.

  Remarque : Cet exemple utilise une licence IBM MQ Advanced , mais vous pouvez également utiliser une licence IBM MQ . Pour plus d'informations, voir «Configurer les gestionnaires de files d'attente avec les annotations de licence IBM MQ en utilisant la fonction IBM MQ Operator», à la page 166.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: qm1mi
spec:
  version: 9.4.3.0-r1
  license:
    accept: true
    license: L-CYPF-CRPF3H
    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 116.

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 218. 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 199.

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 153.

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 223.
- «Affichage du statut des gestionnaires de files d'attente Native HA», à la page 237.
- «Fin des gestionnaires de files d'attente Native HA», à la page 247.

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 224.
- «Surveillance lors de l'utilisation de IBM MQ Operator», à la page 224.
- «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 236.

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

Choisissez l'option appropriée en fonction de la licence utilisée pour votre gestionnaire de file d'attente.

Procédure

- **Option 1 :** Configurer les utilisateurs et les rôles via Keycloak.

  Si vous utilisez une licence IBM Cloud Pak for Integration , IBM MQ Console utilise Keycloak pour la gestion des identités et des accès.

  – Pour plus d'informations sur la configuration Keycloak, voir Gestion des identités et des accès dans la documentation IBM Cloud Pak for Integration

- Pour plus d'informations sur les *rôlesMQ* et leurs rôles Keycloak correspondants, voir Cloud Pak for Integration roles and permissions dans la IBM Cloud Pak for Integration documentation. Les rôles peuvent être configurés pour l'ensemble de l'installation ou pour chaque instance individuelle (gestionnaire de file d'attente). Les équivalents des *rôlesMQ* sont répertoriés sous **Queue manager.**
- Si vous avez précédemment configuré des utilisateurs avec IAM sur des versions antérieures de IBM MQ Operator, voir Migrer les utilisateurs de IAM vers Keycloak dans la documentation IBM Cloud Pak for Integration

- **Option 2 :** Configurer manuellement les utilisateurs et les rôles.

  Si vous utilisez une licence IBM MQ, la IBM MQ Console n'est pas préconfigurée et vous devez la configurer vous-même.

  - Pour plus d'informations sur les utilisateurs et les rôles, voir Configuration des utilisateurs et des rôles.
  - Pour obtenir un exemple simple, voir «Configuration de IBM MQ Console avec un registre de base à l'aide de IBM MQ Operator», à la page 171.
  - Alternativement, si vous changez un gestionnaire de file d'attente pour utiliser une licence IBM Cloud Pak for Integration, vous pouvez alors configurer Keycloak comme décrit précédemment.

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 .

Avant de commencer

Vous avez besoin d'une autorisation pour accéder au IBM MQ Console Voir «Octroi de droits pour le IBM MQ Console», à la page 223.

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}'
```

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 153
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 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. Les mesures du gestionnaire de file d'attente sont activées et fournies par défaut via HTTP.

Avant de commencer

Utilisation d' HTTPS s pour les mesures du gestionnaire de file d'attente

Depuis 9.4.2.0-r1, les gestionnaires de file d'attente peuvent servir des métriques via HTTPS en utilisant un certificat créé par OpenShift . Avant l' 9.4.2.0-r1, les gestionnaires de file d'attente ne peuvent servir les métriques que par l'intermédiaire d' HTTP.

Remarque : Tous les gestionnaires de file d'attente fournissent des mesures par défaut via HTTP.

Pourquoi et quand exécuter cette tâche

Prometheus est une base de données de séries chronologiques open source et un moteur d'évaluation de règles pour les métriques. Les conteneurs d' IBM MQ s exposent un point de terminaison de métriques qui peut être interrogé par Prometheus. Les métriques sont générées à partir des sujets du système d' IBM MQ Operator , pour le suivi et la traçabilité des activités.

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. Voir Activation du suivi des projets définis par l'utilisateur.

Lorsque vous créez un `QueueManager` avec des métriques activées, l' IBM MQ Operator crée un `ServiceMonitor` que l' Prometheus Operator peut ensuite découvrir.

Procédure

- Affichez les mesures du gestionnaire de files d'attente à l'aide de la pile de surveillance Red Hat OpenShift Container Platform (OCP).
  a) Ouvrez **l'onglet Métriques** dans l' OCP.
  b) Cliquez sur **Observer** > **les métriques**.
- Fournir des mesures de gestion de file d'attente par le biais de HTTPS

  Définissez **`.spec.queueManager.metrics.tls.provider`** sur `openshift`.

  ```
  spec:
    queueManager:
      metrics:
        enabled: true
        tls:
          provider: openshift
  ```

  Remarque :

  L'utilisation de ce paramètre crée un certificat d' OpenShift , utilisé par le serveur de métriques d' HTTPS s d'un gestionnaire de file d'attente. Aucun autre fournisseur ne peut être configuré pour être utilisé par l' IBM MQ Operator.
- Désactiver les mesures du gestionnaire de file d'attente.

  Définissez **`.spec.queueManager.metrics.enabled`** sur `false`.

Tâches associées

«Troubleshooting container clusters with the OpenShift Monitoring CLI», à la page 253
As a troubleshooting aid, you can query the available container cluster metrics using the OpenShift Monitoring CLI.

OpenShift CP4I *Monitoring container clusters with the OpenShift Monitoring console*

You can query the available container cluster metrics using the OpenShift Monitoring web console.

Before you begin

OpenShift Monitoring provides a web console and a CLI. For day-to-day operations you might prefer to use the console, as described in this task. For gathering troubleshooting information you might prefer to

use the CLI, as described in “Troubleshooting container clusters with the OpenShift Monitoring CLI” on page 253.

About this task

OpenShift Container Platform includes a pre-configured, pre-installed and self-updating monitoring stack that uses an open source Prometheus server. Queue managers running under an IBM MQ Operator can produce metrics compatible with Prometheus. For a complete list of the metrics available for IBM MQ, see “Indicateurs publiés par l' IBM MQ ” on page 227.

OpenShift Monitoring is the default cluster monitoring method. It is installed with the cluster. It provides Prometheus, Thanos Querier, and a Grafana dashboard that shows queue manager metrics.

To query the available container cluster metrics using the OpenShift Monitoring console, complete the following steps.

Procedure

1. Enable OpenShift Monitoring for the operands deployed in an IBM MQ container cluster.
   a) Log in to the cluster as a cluster administrator.
   b) Add "`enableUserWorkload: true`" to the configMap `cluster-monitoring-config` in the `openshift-monitoring` namespace. To do this, follow the steps in Enabling monitoring for user-defined projects in the Red Hat OpenShift documentation. If the `cluster-monitoring-config` ConfigMap does not exist in the `openshift-monitoring` namespace, follow the instructions to create it.
2. Query the IBM MQ metrics using the OpenShift Monitoring console.

   Use the static Grafana dashboard, which presents cluster metrics to cluster administrators.

   Query IBM MQ metrics, and examine the data plotted on a chart, as described in Querying metrics in the Red Hat OpenShift documentation.

   For example, if you query the metric **`ibmmq_qmgr_cpu_load_fifteen_minute_average_percentage`** you get a chart similar to the following screen capture:

   This screen capture also shows the web console prompt. You can run `PromQL` queries from this prompt. For example:

   Get a list of queue managers that have a positive FDC count:

```
ibmmq_qmgr_fdc_files > 0
```

Pass the `namespace` filter in the query to get the results for a specific namespace:

```
ibmmq_qmgr_fdc_files{namespace='your-namespace'} > 0
```

Return the count of all such queue managers:

```
count(ibmmq_qmgr_fdc_files{namespace='your-namespace'} > 0)
```

Get the Write latency of queue manager pods over the last 5 hours:

```
ibmmq_qmgr_log_write_latency_seconds{namespace='your-namespace'}[5h]
```

Get the RAM usage estimate in bytes for a queue manager for a particular namespace summed by "`Job`":

```
sum by (job)(ibmmq_qmgr_ram_usage_estimate_for_queue_manager_bytes{qmgr='your-queue-
manager', namespace='your-namespace'})
```

The previous queries illustrate use of filters "`namespace`", "`job`" and "`qmgr`". For this metric you can also filter on "`container`", "`endpoint`", "`instance`", "`pod`", and "`prometheus`". Note that the available filters are the column headings shown in the screen capture, except for “Name” which is the name of the metric itself.

Related tasks

“Troubleshooting container clusters with the OpenShift Monitoring CLI” on page 253
As a troubleshooting aid, you can query the available container cluster metrics using the OpenShift Monitoring CLI.

Thanos Querier Versus Thanos Querier

Related reference

“Indicateurs publiés par l' IBM MQ ” on page 227
Les conteneurs du gestionnaire de files d'attente peuvent publier des mesures compatibles avec le monitoring Red Hat OpenShift.

OpenShift CP4I *Indicateurs publiés par l' IBM MQ*

Les conteneurs du gestionnaire de files d'attente peuvent publier des mesures compatibles avec le monitoring Red Hat OpenShift.

Métrique	Tapez	Description
`ibmmq_nha_recovery_average_network_round_trip_time_seconds`	gauge	Durée moyenne des allers-retours réseau
`ibmmq_nha_recovery_backlog_average_bytes`	gauge	Nombre moyen d'octets en attente
`ibmmq_nha_recovery_backlog_bytes`	gauge	Octets en attente
`ibmmq_nha_recovery_compressed_log_sent_bytes`	gauge	Octets de journal compressés envoyés
`ibmmq_nha_recovery_log_data_average_compression_time_seconds`	gauge	Temps de compression moyen des données de journal

Métrique	Tapez	Description
ibmmq_nha_recovery_log_data_average_decompression_time_seconds	gauge	Temps de décompression moyen des données de journal
ibmmq_nha_recovery_log_decompressed_bytes	gauge	Octets de journal décompressés
ibmmq_nha_recovery_log_sent_bytes	counter	Octets de journal envoyés
ibmmq_nha_recovery_rebase_count	gauge	Nombre de resynchronisations
ibmmq_nha_recovery_recovery_log_sequence_number	gauge	Numéro de séquence de journal de reprise
ibmmq_nha_replication_acknowledged_log_sequence_number_total	gauge	Numéro de séquence de journal pris en compte
ibmmq_nha_replication_average_network_round_trip_time_seconds	gauge	Durée moyenne des allers-retours réseau
ibmmq_nha_replication_backlog_average_bytes	gauge	Nombre moyen d'octets en attente
ibmmq_nha_replication_backlog_bytes	gauge	Octets en attente
ibmmq_nha_replication_catch_up_compressed_log_sent_bytes	counter	Octets de journal compressés de rattrapage envoyés
ibmmq_nha_replication_catch_up_log_data_average_compression_time_seconds	gauge	Temps de compression moyen des données de journal de rattrapage
ibmmq_nha_replication_catch_up_log_data_average_decompression_time_seconds	gauge	Temps de décompression moyen des données de journal de rattrapage
ibmmq_nha_replication_catch_up_log_decompressed_bytes	gauge	Octets de journal de rattrapage décompressés
ibmmq_nha_replication_catch_up_uncompressed_log_sent_bytes	counter	Octets de journal non compressés de rattrapage envoyés

Métrique	Tapez	Description
ibmmq_nha_replication_catchup_log_sent_bytes	counter	Octets de journal de rattrapage envoyés
ibmmq_nha_replication_log_file_system_free_space_percent	gauge	Système de fichiers journaux - espace libre
ibmmq_nha_replication_log_file_system_in_use_bytes	gauge	Système de fichier journal - octets utilisés
ibmmq_nha_replication_log_write_average_acknowledgement_latency_seconds	gauge	Temps d'attente moyen d'accusé de réception d'écriture de journal
ibmmq_nha_replication_log_write_average_acknowledgement_size_bytes	gauge	Taille de l'accusé de réception d'écriture de journal
ibmmq_nha_replication_mq_fdc_file_count	gauge	Nombre de fichiers FDC MQ
ibmmq_nha_replication_queue_manager_file_system_free_space_percent	gauge	Système de fichiers du gestionnaire de files d'attente - espace disponible
ibmmq_nha_replication_queue_manager_file_system_in_use_bytes	gauge	Système de fichiers du gestionnaire de files d'attente - octets utilisés
ibmmq_nha_replication_synchronous_compressed_log_sent_bytes	counter	Octets de journal compressés synchrones envoyés
ibmmq_nha_replication_synchronous_log_data_average_compression_time_seconds	gauge	Temps de compression moyen des données de journal synchrones
ibmmq_nha_replication_synchronous_log_data_average_decompression_time_seconds	gauge	Temps de décompression moyen des données de journal synchrones
ibmmq_nha_replication_synchronous_log_decompressed_bytes	gauge	Octets de journal synchrones décompressés

Métrique	Tapez	Description
ibmmq_nha_replication_synchronous_log_sent_bytes	counter	Octets de journal synchrones envoyés
ibmmq_nha_replication_synchronous_uncompressed_log_sent_bytes	counter	Octets de journal non compressés synchrones envoyés
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
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

Métrique	Tapez	Description
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_free_space_percentage	gauge	Système de fichiers journaux - espace libre
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_extents_waiting_to_be_archived_bytes	gauge	Journal - occupé par les extensions en attente d'archivage
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_sequence_number_disk_total	gauge	Journal - numéro de séquence de journal écrit sur le disque
ibmmq_qmgr_log_sequence_number_quorum_total	gauge	Journal - numéro de séquence de journal du quorum
ibmmq_qmgr_log_slowest_write_since_restart_seconds	gauge	Journal - Ecriture la plus lente depuis le redémarrage
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

Métrique	Tapez	Description
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
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

Métrique	Tapez	Description
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
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

Métrique	Tapez	Description
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
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 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`'.

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** 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, les informations provenant des instances répliques peuvent être obsolètes.

V 9.4.2 Pour une configuration Native HA Cross-Region Replication (CRR), les informations fournies dépendent également du fait que l'instance fait partie d'un groupe Live ou d'un groupe Recovery. Les informations provenant du groupe Live sont plus définitives, mais peuvent toujours être périmées car le groupe Recovery ne renvoie que des informations périodiques.

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.
- V 9.4.2 Afficher le statut des groupes auxquels l'instance appartient ou auxquels elle est connectée.

Les champs d'état suivants sont utilisés pour signaler l'état des configurations Native HA et Native HA Cross-Region Replication (CRR) d'une instance de gestionnaire de file d'attente. Ces champs apparaissent toujours :

ROLE
: Spécifie le rôle actuel de l'instance. Est l'un des sites `Active`, `Replica`, `Unknown`, (pour un groupe de récupération) `Leader`, ou `Not Configured`.

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*.

V 9.4.2 **GRPLSN**
: Spécifie le numéro de séquence du journal (LSN) qui a été répliqué entre un quorum d'instances dans le groupe HA, au format « *nnnnn : nnnnn : nnnnn : nnnnn* ». Blanc si le LSN n'est pas connu.

V 9.4.2 **GRPNAME**
: Spécifie le nom du groupe HA. Blanc si aucun nom n'a été configuré.

V 9.4.2 **GRPROLE**
: Spécifie le rôle actuel du groupe HA et est l'un des suivants :

- `Live` - Un groupe vivant
- `Recovery` - Un groupe de rétablissement

- `Pending live` - Un groupe en devenir `Live`
- `Pending recovery` - Un groupe en devenir `Recovery`
- `Not configured` - Native HA n'a pas été configuré
- `Unknown` - Le rôle actuel ne peut être déterminé

Ces champs apparaissent lorsque des informations complémentaires sont demandées pour toutes les instances en utilisant le paramètre `-x` :

REPLADDR
Adresse de réplication de l'instance de gestionnaire de files d'attente.

CONNACTV
Indique si l'instance est connectée à l'instance active.

BACKLOG
Indique le nombre de Ko que l'instance est en retard par rapport à l'instance active.

CONNINST
Indique si l'instance désignée est connectée à cette instance.

ALTDATE
Indique la date de la dernière mise à jour de ces informations. Blanc s'il n'a jamais été mis à jour.

ALTTIME
Indique l'heure à laquelle cette information a été mise à jour pour la dernière fois. Blanc s'il n'a jamais été mis à jour.

V 9.4.2 **ACKLSN**
Indique le LSN que l'instance a reconnu comme ayant été écrit dans le journal de récupération, au format *nnnnn:nnnnn:nnnnn:nnnnn*. Blanc si la valeur du LSN n'est pas connue.

V 9.4.2 **HAStatus**

Indique l'état opérationnel de cette instance et correspond à l'une des valeurs suivantes :

- `Normal` - l'instance fonctionne normalement.
- `Checking` - l'instance est vérifiée pour assurer la cohérence du journal de récupération.
- `Synchronizing` - l'instance reçoit des données pour la synchroniser.
- `Rebasing` - un nouveau journal de récupération est envoyé à l'instance.
- `Disk full` - l'instance a un disque plein.
- `Disconnected` - ne peut pas communiquer avec l'instance.
- `Unknown` - le statut de l'instance est inconnu.

V 9.4.2 **SYNCTIME**
Indique l'heure à laquelle cette instance a été synchronisée pour la dernière fois avec le gestionnaire de file d'attente actif, au format ISO 8601. Blanc si l'heure n'est pas connue.

V 9.4.2 Ces champs apparaissent lorsque des informations sur le groupe sont demandées en utilisant le paramètre `-g`. Ces champs sont utiles pour visualiser le statut lorsque le groupe fait partie d'une configuration Native HA CRR :

V 9.4.2 **GRPNAME**
Spécifie le nom du groupe HA. Blanc si aucun nom n'a été configuré.

V 9.4.2 **GRPROLE**
Spécifie le rôle actuel du groupe HA et est l'un des suivants :

- `Live` - Un groupe vivant
- `Recovery` - Un groupe de rétablissement
- `Pending live` - Un groupe en devenir `Live`

- `Pending recovery` - Un groupe en devenir `Recovery`
- `Unknown` - Le rôle actuel ne peut être déterminé
- `Not configured` - L'HA natif n'a pas été configuré

V 9.4.2 **GRPADDR**

L'adresse IP utilisée pour se connecter au groupe. La valeur est fixée à Unknown si l'adresse n'est pas connue parce qu'il n'y a pas eu de connexion au groupe.

V 9.4.2 **GRPVER**

Indique la version du chef de groupe actuel au format `V.R.M.F`, où `V.R.M.F` est le numéro de Version, Release, Modification et Fix Pack. Si la valeur n'est pas connue, elle est fixée à `?.?.?.?`

V 9.4.2 **CONNGRP**

Indique si le groupe est connecté au groupe local et s'il est l'un des suivants :

- `yes` - le groupe est connecté.
- no - le groupe est déconnecté.
- `unknown` - l'état de la connexion est inconnu.
- `suspended` - le groupe est connecté, mais il ne peut pas répliquer les données tant qu'une incompatibilité de configuration entre les groupes n'est pas résolue.

Ce champ n'est affiché que pour les groupes autres que le groupe local.

V 9.4.2 **INSYNC**

Ceci n'est affiché que pour un groupe ayant le rôle `Recovery` ou `Pending live`. Il indique si le groupe peut devenir `Live` sans perte de données en cas de basculement. (Notez qu'un groupe d' `Recovery` s peut ne pas être très souvent synchronisé avec le groupe d' `Live` s car le journal est répliqué de manière asynchrone. Utilisez les valeurs RCOVLSN et RCOVTIME pour indiquer si une perte de données peut se produire en cas de basculement non planifié).

V 9.4.2 **SYNCTIME**

Ceci n'est affiché que pour un groupe ayant le rôle `Recovery` ou `Pending live`. Il indique l'heure à laquelle ce groupe a été synchronisé pour la dernière fois avec le groupe d' `Live` , au format ISO 8601 (blanc si elle n'est pas connue). (Notez qu'un groupe d' `Recovery` s peut ne pas être très souvent synchronisé avec le groupe d' `Live` s car le journal est répliqué de manière asynchrone. Utilisez les valeurs RCOVLSN et RCOVTIME pour indiquer si une perte de données peut se produire en cas de basculement non planifié).

V 9.4.2 **BACKLOG**

Ceci n'est affiché que pour un groupe ayant le rôle `Recovery` ou `Pending live`. Il indique le nombre de Ko de retard du groupe par rapport au groupe en direct.

V 9.4.2 **GRSTATUS**

Indique l'état opérationnel de ce groupe et correspond à l'une des valeurs suivantes :

- `Normal` - le groupe fonctionne normalement.
- `Checking` - le groupe est vérifié pour assurer la cohérence du journal des récupérations.
- `Synchronizing` - le groupe reçoit des données pour le synchroniser.
- `Rebasing` - un nouveau journal de récupération est envoyé à l'instance.
- `Waiting for connection` - le groupe attend une connexion d'un groupe ayant le rôle `Live` ou `Pending Recovery`.
- `Partitioned` - deux groupes croient qu'ils sont le groupe Live (c'est-à-dire un statut de "cerveau divisé").
- `Unknown` - le statut du groupe est inconnu.

V 9.4.2 **RCOVLSN**
Un numéro de séquence d'enregistrement (LSN) que le groupe pourrait récupérer, au format *nnnnn:nnnnn:nnnnn:nnnnn*. Toute donnée écrite dans le journal de récupération après ce point pourrait être perdue en cas de basculement imprévu. Cette valeur peut être transmise à l'option -s de la commande **dmpmqlog**.

V 9.4.2 **RCOVTIME**
Heure à laquelle le groupe pourrait récupérer, au format ISO 8601. Toute donnée écrite dans le journal de récupération après ce délai pourrait être perdue en cas de basculement imprévu. Cette valeur peut être transmise dans les options -t ou -u de la commande **dmpmqlog**.

V 9.4.2 **INITLSN**
Ceci n'est affiché que pour un groupe ayant le rôle `Live` ou `Pending recovery`. Il indique le numéro de séquence du journal (LSN) du dernier enregistrement récupéré lorsque le groupe Native HA est devenu actif, au format *nnnnn:nnnnn:nnnnn:nnnnn*, ou vide si le groupe n'est pas actif.

V 9.4.2 **initTime**
Ceci n'est affiché que pour un groupe ayant le rôle `Live` ou `Pending recovery`. Il indique l'heure du dernier enregistrement récupéré lorsque le groupe Native HA a été mis en service, au format ISO 8601. Blanc, si le groupe n'est pas en direct.

V 9.4.2 **LIVETIME**
Ceci n'est affiché que pour un groupe ayant le rôle `Live` ou `Pending recovery`. Il indique l'heure à laquelle le groupe est devenu le groupe en direct, dans le format ISO 8601. Blanc, si le groupe n'est pas en direct.

V 9.4.2 **ALTDATE**
Indique la date de la dernière mise à jour de ces informations.

V 9.4.2 **ALTTIME**
Indique l'heure à laquelle cette information a été mise à jour pour la dernière fois.

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)
```

  V 9.4.2 Une instance de BOB qui est un chef de groupe de récupération dans une configuration CRR Native HA signalerait l'état suivant :

```
QMNAME(BOB)                    STATUS(Recovery group leader)
```

  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)
GRPLSN(<0:25:365:1000>) GRPNAME() GRPROLE(Live)
```

V 9.4.2

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)
GRPLSN(<0:25:365:1000>) GRPNAME() GRPROLE(Live)
```

V 9.4.2

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)
GRPLSN() GRPNAME() GRPROLE(Live)
```

V 9.4.2

V 9.4.2 Une instance active d'un gestionnaire de file d'attente nommé BOB appartenant au groupe Live dans une configuration Native HA CRR peut signaler l'état suivant :

```
QMNAME(BOB)         ROLE(Active) INSTANCE(inst1) INSYNC(yes) QUORUM(3/3)
GRPLSN(<0:25:365:1000>) GRPNAME(alpha) GRPROLE(Live)
```

V 9.4.2 La même instance, lorsque le groupe attend de devenir le groupe de recouvrement, peut signaler l'état suivant :

```
QMNAME(BOB)         ROLE(Active) INSTANCE(inst1) INSYNC(yes) QUORUM(3/3)
GRPLSN(<0:25:365:1000>) GRPNAME(alpha) GRPROLE(Pending recovery)
```

V 9.4.2 Un gestionnaire de file d'attente nommé CAROL qui n'est pas une instance d'une configuration Native HA ou Native HA CRR devrait signaler l'état suivant :

```
QMNAME(CAROL)       ROLE(Not configured) INSTANCE() INSYNC() QUORUM() GRPLSN() GRPNAME()
GRPROLE(Not configured)
```

- 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
```

Par exemple, si vous lancez cette commande sur le nœud exécutant l'instance active du gestionnaire de file d'attente BOB, où toutes les instances sont synchronisées, vous pouvez obtenir l'état suivant :

```
QMNAME(BOB)         ROLE(Active) INSTANCE(inst1) INSYNC(yes) QUORUM(2/3)
GRPLSN(<0:25:365:1000>) GRPROLE(Live) GRPNAME()
 INSTANCE(inst1) ROLE(Active)  REPLADDR(localhost) CONNACTV(yes) INSYNC(yes)
BACKLOG(0)  CONNINST(yes) ACKLSN(<0:26:365:1000>)  HASTATUS(Normal)
SYNCTIME(2023-12-20T17:13:27.853412Z) ALTDATE(2023-12-20) ALTTIME(17.13.27)
 INSTANCE(inst3) ROLE(Replica) REPLADDR(localhost) CONNACTV(yes) INSYNC(yes)
BACKLOG(0)  CONNINST(yes) ACKLSN (<0:24:365:1000>) HASTATUS(Normal)
SYNCTIME(2023-12-20T17:13:27.853412Z) ALTDATE(2023-12-20) ALTTIME(17.13.27)
 INSTANCE(inst2) ROLE(Replica) REPLADDR(localhost) CONNACTV(yes) INSYNC(yes)
BACKLOG(0)  CONNINST(yes) ACKLSN (<0:25:365:1000>) HASTATUS(Synchronizing)
SYNCTIME(2023-12-20T17:13:22.123321Z) ALTDATE(2023-12-20) ALTTIME(17.13.27)
```

Dans un autre exemple, si vous lancez cette commande sur un nœud exécutant une instance répliquée du gestionnaire de file d'attente BOB, qui est à la traîne, vous pouvez obtenir l'état suivant :

```
QMNAME(BOB)         ROLE(Replica) INSTANCE(inst2) INSYNC(no) QUORUM(2/3)
GRPLSN(<0:25:365:1000>) GRPROLE(Live) GRPNAME()
 INSTANCE(inst1) ROLE(Active)  REPLADDR(localhost) CONNACTV(yes) INSYNC(yes)
BACKLOG(0)  CONNINST(yes) ACKLSN(<0:26:365:1000>)  HASTATUS(Normal)
SYNCTIME(2023-12-20T17:13:27.853412Z) ALTDATE(2023-12-20) ALTTIME(17.13.27)
 INSTANCE(inst3) ROLE(Replica) REPLADDR(localhost) CONNACTV(yes) INSYNC(yes)
BACKLOG(0)  CONNINST(yes) ACKLSN (<0:24:365:1000>) HASTATUS(Normal)
SYNCTIME(2023-12-20T17:13:27.853412Z) ALTDATE(2023-12-20) ALTTIME(17.13.27)
 INSTANCE(inst2) ROLE(Replica) REPLADDR(localhost) CONNACTV(yes) INSYNC(no)
BACKLOG(435) CONNINST(yes) ACKLSN (<0:25:365:1000>) HASTATUS(Synchronizing)
SYNCTIME(2023-12-20T17:13:22.123321Z) ALTDATE(2023-12-20) ALTTIME(17.13.27)
```

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) GRPLSN()
GRPROLE() GRPNAME()
 INSTANCE(inst1) ROLE(Unknown) REPLADDR(9.20.123.45) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ACKLSN()  HASTATUS(Unknown) SYNCTIME() ALTDATE() ALTTIME()
ALTDATE() ALTTIME()
 INSTANCE(inst2) ROLE(Unknown) REPLADDR(9.20.123.46) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ACKLSN()  HASTATUS(Unknown) SYNCTIME() ALTDATE() ALTTIME()
ALTDATE() ALTTIME()
 INSTANCE(inst3) ROLE(Unknown) REPLADDR(9.20.123.47) CONNACTV(No)      INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ACKLSN()  HASTATUS(Unknown) SYNCTIME() ALTDATE() ALTTIME()
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)
```

- V 9.4.2

  Pour déterminer l'état opérationnel de tous les groupes Native HA dans une configuration Native HA CRR :

  ```
  oc exec -t Pod dspmq -o nativeha -g -m QMgrName
  ```

  Si vous émettez cette commande sur un membre du groupe Live, nommé alpha, vous pourriez recevoir le statut suivant pour alpha, et le groupe Recovery, beta :

  ```
  GRPNAME(alpha) GRPROLE(Live) GRPADDR(9.20.135.129) GRPVER(09040200)
  GRSTATUS(Normal) RCOVLSN(<0:0:104:65163>) RCOVTIME(2023-12-20T17:37:27.114787Z)
  INITLSN(<0:0:101:352>) INITTIME(2023-12-19T09:13:27.512294)
  LIVETIME(2023-12-19T06:14:27.512294) ALTDATE(2023-12-20) ALTTIME(17.13.27)
  ```

  ```
  GRPNAME(beta) GRPROLE(Recovery) GRPADDR(9.20.135.130) GRPVER(09040300)
  CONNGRP(yes) GRSTATUS(Synchronizing) RCOVLSN(<0:0:19:16723>)
  RCOVTIME(2023-12-20T17:36:23.463209Z) BACKLOG(5619) INSYNC(no) SYNCTIME()
  ALTDATE(2023-12-20) ALTTIME(17.13.27)
  ```

Référence associée

Commande dspmq (display queue managers)

V 9.4.2 Basculement et basculement automatique en cas de panne de la HA CRR native

Une configuration Native HA Cross-Region Replication (CRR) vous permet de basculer l'exécution d'un gestionnaire de file d'attente vers une autre région.

Vous pouvez utiliser une configuration Native HA CRR pour fournir une solution de reprise après sinistre. Si un sinistre se produit sur votre site de production, vous pouvez effectuer un basculement et continuer à exécuter le gestionnaire de files d'attente sur le site de récupération. Dans cette situation, vous ne pouvez pas garantir que les journaux de récupération seront synchronisés lors du basculement, ce qui pourrait

entraîner une perte de données. De même, lorsque le site défaillant est restauré, vous pouvez avoir deux instances actives du gestionnaire de file d'attente et donc accumuler des données partitionnées (split-brain).

Vous pouvez utiliser une configuration Native HA CRR pour assurer la continuité pendant les opérations de maintenance ou de test. Dans ce cas, le basculement est contrôlé de manière à ce qu'aucune perte de données ne se produise. De même, lorsque vous revenez en arrière, il n'y a pas de données partitionnées à résoudre.

V 9.4.2 Effectuer un basculement planifié sur une configuration Native HA CRR

Vous pouvez effectuer une commutation planifiée entre les groupes de production et de reprise dans une configuration de réplication inter-régions (CRR) Native HA.

Pourquoi et quand exécuter cette tâche

Vous pourriez avoir besoin de passer de votre groupe actif à votre groupe de secours si vous souhaitez tester votre configuration ou si vous devez effectuer une maintenance.

Lorsque vous effectuez une commutation planifiée, le système prend des mesures pour s'assurer que vous ne perdez pas de données et que vous ne rencontrez pas de problème de partitionnement (split-brain). Vous demandez le basculement en modifiant le rôle de groupe sur chacun des groupes. Pendant un certain temps, les deux groupes sont en attente pendant que leurs journaux de récupération sont synchronisés. Lorsque cela est fait, les groupes adoptent leurs nouveaux rôles, un nouveau leader est élu dans le nouveau groupe actif et prend le rôle actif et gère le gestionnaire de file d'attente.

Procédure

1. Spécifiez que le groupe actif doit devenir le groupe de récupération en définissant son `GroupRole` sur `Recovery` dans la strophe `NativeHALocalInstance` du fichier `qm.ini` .

   Si vous utilisez l'opérateur " MQ ", l'élément " `GroupRole` " est modifié en définissant le champ " `.spec.queueManager.availability.nativeHAGroups.local.role` " dans le fichier YAML " QueueManager ". Par exemple, pour modifier le rôle du groupe Live en Recovery, définissez la valeur du champ `.spec.queueManager.availability.nativeHAGroups.local.role` sur `Recovery` pour le groupe actuellement en ligne.

   Pour vérifier que le basculement a réussi, obtenez l'état des pods à l'aide de la commande **`oc get pods`** , un pod devrait être dans l'état "ready" :

   ```
   NAME                  READY   STATUS    RESTARTS   AGE
   exampleqm-ibm-mq-0    1/1     Running   0          2m45s
   exampleqm-ibm-mq-1    0/1     Running   0          2m45s
   exampleqm-ibm-mq-2    0/1     Running   0          2m45s
   ```

   Vérifiez l'état du groupe de récupération en exécutant la commande suivante :

   ```
   oc exec -t <leader-pod> -- dspmq -o nativeha -g
   ```

   Où <leader-pod> est le nom du pod prêt 1/1 (exampleqm-ibm-mq-0 dans l'exemple) :

   Vérifier que le GRPROLE est en attente de récupération.
2. Spécifiez que le groupe de récupération doit devenir le groupe actif en définissant son attribut `GroupRole` sur `Live` dans la strophe `NativeHALocalInstance` du fichier `qm.ini` .

   Si vous utilisez l'opérateur " MQ ", l'élément " `GroupRole` " est modifié en définissant le champ " `.spec.queueManager.availability.nativeHAGroups.local.role` " dans le fichier YAML " QueueManager ". Par exemple, pour modifier le rôle du groupe Live en Live, définissez la valeur du champ `.spec.queueManager.availability.nativeHAGroups.local.role` sur `Live` pour le groupe qui est actuellement en cours de récupération.

Si vous émettez une commande **dspmq -o nativeha -g** pour ce gestionnaire de file d'attente après le redémarrage de chaque instance du groupe, vous verrez un GRPROLE de `Pending live`.

Une fois que chaque instance du groupe a redémarré, lancez la commande suivante :

```
oc exec -t <active-pod> -- dspmq -o nativeha -g
```

Où <pod actif> est le nom du pod prêt 1/1 affiché par la commande `oc get pods` . Vous voyez que GRPROLE est `Pending live` suivi de `live` une fois que les groupes sont synchronisés.

3. Lorsque les journaux de reprise sur le site de reprise et le site actif sont synchronisés, le groupe de reprise devient actif et élit un leader qui sera l'instance active du gestionnaire de file d'attente. À partir de là, les applications se reconnecteront automatiquement à cette instance et toute charge de travail reprendra normalement.

Que faire ensuite

Lorsque la maintenance est terminée, vous pouvez revenir au groupe actif d'origine en répétant le processus.

V 9.4.2 Effectuer un basculement non planifié sur une configuration Native HA CRR

Vous effectuez un basculement non planifié entre les groupes de production et de reprise dans une configuration de réplication inter-régions (CRR) native HA lorsqu'une catastrophe survient sur le site de production et que celui-ci ne peut plus fonctionner.

Pourquoi et quand exécuter cette tâche

Lorsque le groupe actif tombe en panne, le groupe de secours ne peut pas le détecter car il cesse simplement de recevoir les données de journal répliquées, ce qui pourrait simplement signifier qu'il n'y a pas de charge de travail en cours. Un basculement non planifié doit donc être effectué manuellement en changeant le rôle du groupe de reprise en direct et en l'informant que le groupe précédemment en direct est désactivé afin qu'il ne tente pas de répliquer les données vers celui-ci.

Le groupe de reprise peut ne pas être synchronisé avec le groupe actif car les données du journal sont répliquées de manière asynchrone. Si le groupe de récupération n'est pas synchronisé avec le groupe actif, des données sont perdues en cas de basculement imprévu. Avant de lancer un basculement non planifié, exécutez la commande **dspmq -o nativeha -g** sur le groupe de récupération et utilisez les valeurs RCOVLSN et RCOVTIME pour afficher le numéro de séquence du journal (LSN) et l'heure à laquelle le groupe peut récupérer.

Procédure

1. S'assurer que les applications se sont déconnectées du groupe actif et sont configurées pour se reconnecter au groupe qui est actuellement le groupe de récupération.
2. Spécifiez que le groupe de récupération doit devenir le groupe actif en effectuant les étapes suivantes :

   a. Définissez son `GroupRole` sur `Live` dans la strophe `NativeHALocalInstance` du fichier `qm.ini` . Par exemple, si vous utilisez l'opérateur « MQ », modifiez le paramètre « `.spec.queueManager.availability.nativeHAGroups.local.role` » en « `Live` ».

   b. Définissez le champ `Enabled` sur `False` dans la strophe `NativeHARecoveryGroup` du fichier `qm.ini` . Par exemple, si vous utilisez l'opérateur « MQ », modifiez le paramètre « `.spec.queueManager.availability.nativeHAGroups.remotes.enabled` » en « `false` ».

   Si vous émettez une commande **dspmq -o nativeha -g** pour ce gestionnaire de file d'attente après le redémarrage de chaque instance du groupe, vous verrez un GRPROLE de `Live`. Un vote est organisé parmi les membres du groupe et l'un d'entre eux devient l'instance active et se charge du travail du gestionnaire de file d'attente.

3. Pour relancer la réplication après le basculement, vous pouvez soit utiliser le groupe actif d'origine comme groupe de récupération, soit créer un nouveau groupe de récupération sur un nouveau site.
   - Si le site de production d'origine est restauré, avant de redémarrer le groupe de production d'origine, modifiez le rôle du groupe en récupération. Cela permet de s'assurer que le groupe ne reprenne pas son rôle actif.
   - Si le site d'origine ne peut pas être restauré, créez un nouveau groupe de récupération Native HA sur un nouveau site.

   Pour configurer le groupe de récupération, suivez les étapes décrites dans «Restauration de la réplication après un basculement imprévu», à la page 245.
4. Si la configuration du site de production d'origine n'est pas modifiée et que le groupe reprend son rôle de production lorsqu'il est restauré, vous aurez deux instances actives du gestionnaire de file d'attente. Si tel est le cas, l'état partitionné est détecté et aucune autre réplication n'a lieu entre les deux groupes. Vous devez déterminer quelles données conserver avant de restaurer les rôles originaux de production et de reprise. Pour plus d'instructions, voir «Résolution d'un problème de partitionnement (split brain)», à la page 246.

Information associée

dspmq (affichage des gestionnaires de files d'attente)

V 9.4.2 Restauration de la réplication après un basculement imprévu

Restaurer la réplication du groupe actif vers un groupe de récupération après un basculement imprévu.

Pourquoi et quand exécuter cette tâche

Après un basculement non planifié, si le site actif d'origine est restauré, il peut être reconfiguré pour devenir le groupe de récupération. Pour empêcher le gestionnaire de file d'attente de passer en mode partitionné (« split-brain »), terminez cette tâche avant de redémarrer le groupe Native HA sur le site de production d'origine.

Vous pouvez également suivre ces étapes lorsque vous créez un nouveau groupe de récupération après un basculement non planifié, si le site actif d'origine ne peut pas être restauré.

Procédure

1. Spécifiez que le groupe doit devenir le groupe de récupération en suivant les étapes suivantes :
   a) Définissez son état de récupération (GroupRole) sur Recovery dans la strophe d' NativeHALocalInstance s du fichier d' `qm.ini` . Par exemple, si vous utilisez l'opérateur « MQ », modifiez le paramètre du champ « `.spec.queueManager.availability.nativeHAGroups.local.role` » dans le fichier « QueueManager » YAML en « Recovery ».
   b) Définissez le champ Enabled sur False dans la strophe « NativeHARecoveryGroup » du fichier « qm.ini ». Par exemple, si vous utilisez l'opérateur « MQ », modifiez le paramètre du champ « `.spec.queueManager.availability.nativeHAGroups.remotes.enabled` » dans le YAML « QueueManager » sur «false».

      Si vous exécutez une commande **`dspmq -o nativeha -g`** sur une instance du groupe après le redémarrage de chaque instance, vous verrez un GRPROLE de `Recovery`. Un vote est organisé parmi les membres du groupe et l'un d'entre eux devient le chef du groupe de recouvrement.
2. Redémarrez la réplication du groupe actif vers le groupe de récupération en définissant le champ `Enabled` sur `True` dans la strophe NativeHARecoveryGroup du fichier `qm.ini` du groupe actif. Par exemple, si vous utilisez l'opérateur « MQ », modifiez le paramètre du champ « `.spec.queueManager.availability.nativeHAGroups.remotes.enabled` » dans le fichier « QueueManager YAML » en « true ».

V 9.4.2 Résolution d'un problème de partitionnement (split brain)

Une condition de partitionnement, ou « split-brain », peut se produire si un gestionnaire de file d'attente dans une configuration de réplication inter-régions (CRR) native HA s'exécute dans les deux groupes HA en même temps.

Cette situation peut se produire en cas de basculement imprévu. La communication avec le groupe Live est perdue et vous faites donc du gestionnaire de file d'attente le principal sur le groupe Recovery et il commence à fonctionner. Pendant ce temps, le gestionnaire de file d'attente est toujours en cours d'exécution sur le groupe Live d'origine. Il est possible que l'ancien groupe Live soit toujours capable de mettre à jour son journal, ce qui crée une branche insoluble dans les données gérées par les deux groupes. Vous devez supprimer les données de l'un des groupes avant de pouvoir les réunir.

Pour résoudre la condition de partitionnement, vous choisissez les données de groupe à conserver et celles à supprimer. Il se peut que les groupes n'aient pas été synchronisés lorsque le basculement non planifié a été déclenché, ou que la charge de travail de l'application ait pu être traitée par les deux groupes. La définition des « meilleures » données est largement subjective. Par exemple, alors qu'un groupe a pu stocker davantage de données dans son journal, il peut s'agir en grande partie d'images multimédias d'objets d' IBM MQ , enregistrées automatiquement, tandis que l'autre groupe a pu traiter des transactions commerciales plus importantes. Quel que soit le groupe choisi, il est probable que, lors de la reprise après partitionnement, les messages non répliqués et les résultats des transactions sur le groupe à supprimer seront utiles pour compenser la perte éventuelle de données.

IBM MQ détecte une condition de partitionnement lorsqu'un groupe avec un rôle Live se connecte à un autre groupe qui s'exécute avec un rôle Live. Si vous émettez la commande `dspmq -o nativeha -g` dans les deux groupes, le statut `GRSTATUS(Partitioned)` est renvoyé. La réplication des données de journalisation est suspendue entre les deux groupes. L'état d' `Partitioned` s continue d'être signalé jusqu'à ce que le groupe soit en mesure de répliquer suffisamment de données de journal vers un autre groupe ayant un rôle de récupération.

S'il n'est pas évident de choisir le groupe à conserver et si des volumes de journaux sont encore disponibles dans les deux groupes au moment du basculement imprévu, vous pouvez comparer le contenu des journaux.

Utilisez la commande « **dspmq -o nativeha -g** » pour identifier le numéro de séquence du journal (LSN) du point exact dans le journal où un groupe est devenu actif. Le LSN est renvoyé sous la forme `INITLSN (nnnnn:nnnnn:nnnnn:nnnnn)`. L'heure de cet événement est également indiquée par l'attribut « `INITTIME` ». Utilisez ensuite la commande **dmpmqlog -s** *startLSN* pour examiner le contenu d'un journal à partir de ce point. Utilisez **dspmq** et **dmpmqlog** en combinaison pour identifier les enregistrements de journal qui ont été écrits par un groupe et non répliqués dans l'autre. La commande « **dmpmqlog** » fournit également un résumé de l'activité de messagerie et de transaction, ce qui peut être utile pour faire son choix.

Exemple

Les groupes « alpha » et « bêta » ont été configurés initialement dans une configuration Live and Recovery. La décision de convertir le groupe « bêta » de Recovery à Live a été prise et un basculement imprévu s'est produit. Cependant, le groupe « alpha » a pu continuer à traiter certaines charges de travail applicatives qui n'ont pas été répliquées vers « bêta ». Pendant ce temps, certaines applications connectées à la « bêta » ont traité une certaine charge de travail de messagerie. L'INITLSN du groupe « bêta » indique le point du journal où le groupe « bêta » s'est séparé du contenu du groupe « alpha ».

Une copie des données du gestionnaire de file d'attente est nécessaire pour le processus de comparaison et le gestionnaire de file d'attente doit être inactif. Cela peut être réalisé dans un environnement de type « Kubernetes » en mettant à zéro la taille de l'ensemble avec état du gestionnaire de file d'attente ou en effectuant une copie des données, par exemple, au moyen d'un instantané de volume persistant.

Une fois l'analyse des données du journal terminée et le choix des données de groupe à supprimer effectué, vous pouvez les supprimer en modifiant le rôle du groupe sur Récupération et en supprimant soit tous les volumes persistants du gestionnaire de file d'attente, soit les informations du groupe Récupération.

Information associée

dspmq

dmpmqlog

OpenShift MQ Adv. Fin des gestionnaires de files d'attente Native HA

Pour IBM MQ dans des conteneurs, vous pouvez utiliser la commande **endmqm** pour mettre fin à un gestionnaire de file d'attente actif ou répliqué qui fait partie d'un groupe Native HA.

Avant de commencer

Remarque : Ces informations ne s'appliquent qu'aux environnements de conteneurs.

Pourquoi et quand exécuter cette tâche

La procédure d'arrêt d'un gestionnaire de files d'attente faisant partie d'un groupe Native HA dépend du fait qu'il s'agit d'une instance active ou répliquée. Lorsque vous terminez l'un ou l'autre type d'instance, un contrôle est effectué pour s'assurer que la fin de l'instance ne brise pas le quorum du groupe Native HA. Si le quorum est rompu, la commande **endmqm** échoue.

Lorsque vous lancez une commande **endmqm**, les autres instances du groupe en sont averties, de sorte qu'elles ne signalent pas d'erreurs lorsque la connexion est interrompue.

Si une instance active perd le quorum en raison de l'arrêt ou de la déconnexion d'un trop grand nombre d'instances répliques, l'instance active attend pendant une durée configurable avant de s'arrêter complètement. Cela permet de disposer d'un certain temps pour arrêter le traitement de manière gracieuse, plutôt que d'interrompre les connexions des applications. Cette valeur peut être spécifiée par l'attribut `QuorumConnectivityTimeout` dans la strophe `NativeHALocalInstance` du fichier `qm.ini`. La valeur par défaut est 0 seconde.

Procédure

- Pour mettre fin à l'instance active d'un gestionnaire de files d'attente, exécutez la commande suivante sur le nœud où l'instance active est en cours d'exécution :

  ```
  endmqm -s QMgrName
  ```

  - Spécifiez l'option `-r` pour aider les applications clientes à se reconnecter à une autre instance.
  - Si cette instance n'est pas l'instance active dans le groupe Native HA, la commande échoue.
  - Si l'arrêt de cette instance active entraîne l'échec du quorum du groupe, la commande échoue. (Si d'autres instances se terminent ou deviennent indisponibles en même temps que vous exécutez cette commande, la vérification du quorum peut ne pas le détecter, le groupe Native HA se termine et ne peut être redémarré que lorsque suffisamment d'instances sont disponibles)

  Lorsque le gestionnaire de file d'attente actif prend fin, l'une des instances de réplique reprend le rôle actif. Vous ne pouvez pas spécifier quelle réplique prend le relais, cela est déterminé par négociation au sein du groupe et dépend de celle qui a les journaux de transactions les plus à jour.
- Pour mettre fin à une instance répliquée d'un gestionnaire de files d'attente, exécutez la commande suivante :

  ```
  endmqm -x QMgrName
  ```

  - Si cette instance est l'instance active, la commande échoue.
  - Si l'arrêt de cette instance de réplique entraîne l'échec du quorum du groupe, la commande échoue. (Si d'autres instances se terminent ou deviennent indisponibles en même temps que vous exécutez cette commande, la vérification du quorum peut ne pas le détecter, le groupe Native HA se termine et ne peut être redémarré que lorsque suffisamment d'instances sont disponibles)

Remarque : Vous pouvez également utiliser les commutateurs -c, -i, -p ou -w avec la commande **endmqm** sur les instances Native HA, quel que soit leur rôle. L'instance du gestionnaire de file d'attente se termine, sans tenir compte de l'effet sur le quorum du groupe. Toutefois, les informations sont toujours partagées avec les autres instances du groupe. Vous pouvez utiliser ces commutateurs avec le -s pour l'instance active. Vous ne pouvez pas utiliser ces commutateurs en même temps que le commutateur -x pour les instances de réplique.

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 248.
- «Traitement des incidents liés à IBM MQ Operator», à la page 250.

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 se terminer. Linux les conteneurs ont la responsabilité de répondre aux signaux POSIX, qui sont des messages standardisés envoyés à un programme pour déclencher un comportement. Le signal SIGTERM est utilisé par Kubernetes (et donc Red Hat OpenShift Container Platform) pour mettre fin à un conteneur de manière élégante.

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'arrêt des conteneurs

Kubernetes peut redémarrer un conteneur dans un pod existant ou supprimer le pod. Certains champs de la section "status" du Pod permettent d'indiquer s'il s'agit d'un redémarrage de conteneur, par exemple :

```
status:
  containerStatuses:
  - restartCount: 2
    started: true
    state:
      running:
```

```
        startedAt: "2025-01-20T12:03:15Z"
  startTime: "2024-12-28T13:46:12Z"
```

Dans cet exemple, vous pouvez voir que le conteneur a redémarré et qu'il est beaucoup plus récent que le pod. Cela indique qu'un seul conteneur a redémarré, ce qui signifie probablement que le conteneur n'a pas répondu à l'enquête sur l'état de conservation. Une sonde d'intégrité peut être configurée dans Kubernetes pour vérifier qu'un conteneur est toujours en bonne santé. 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.

Si vous ne voyez pas de redémarrage de conteneur, alors le Pod entier a probablement été remplacé. Voir Termination of Pods dans la documentation de Kubernetes. Kubernetes utilise les termes "Pod Disruption" et "eviction" pour désigner le processus par lequel les pods sur les nœuds sont arrêtés volontairement ou involontairement. Il existe de nombreuses raisons pour lesquelles un pod peut être expulsé, notamment:

- **Redémarrage en douceur**. Le contrôleur IBM MQ Operator ou Stateful Set peut déclencher une mise à jour continue des pods du gestionnaire de file d'attente, si un changement de configuration est effectué et nécessite une mise à jour continue. Par exemple, modification de la version de l'image, des fichiers montés, des variables d'environnement, des annotations, des étiquettes, etc.
- **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.
- **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 250
- «Identification et résolution des problèmes: accès aux données du gestionnaire de files d'attente», à la page 252

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 on 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 (`metadata.name` dans le YAML du gestionnaire de file d'attente), *QueueManager_namespace* est l'espace de noms dans lequel elle est déployée (`metadata.namespace` dans le YAML du gestionnaire de files d'attente), et *mq_operator_namespace* est l'espace de noms dans lequel la ressource IBM MQ Operator est déployée. Il peut s'agir de l'espace de noms 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
```

Remarques :

- 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**.
- Si vous avez un gestionnaire de file d'attente en cours d'exécution, assurez-vous que les fichiers de sortie produits par ces commandes ne sont pas vides (à l'exception des journaux du pod `qm-logs-previous-`*`pod_name`*`.txt` , qui sont censés être vides si le conteneur n'a pas redémarré).

Tâches associées

Collecte des informations de dépannage

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
```

OpenShift CP4I Troubleshooting container clusters with the OpenShift Monitoring CLI

As a troubleshooting aid, you can query the available container cluster metrics using the OpenShift Monitoring CLI.

Before you begin

OpenShift Monitoring provides a web console and a CLI. For day-to-day operations you might prefer to use the console, as described in “Monitoring container clusters with the OpenShift Monitoring console” on page 225. For gathering troubleshooting information you might prefer to use the CLI, as described in the rest of this task.

About this task

OpenShift Container Platform includes a pre-configured, pre-installed and self-updating monitoring stack that uses an open source Prometheus server. Queue managers running under an IBM MQ Operator can produce metrics compatible with Prometheus. For a complete list of the metrics available for IBM MQ, see “Indicateurs publiés par l' IBM MQ ” on page 227.

OpenShift Monitoring is the default cluster monitoring method. It is installed with the cluster. To query the available container cluster metrics using the OpenShift Monitoring CLI, complete the following steps.

Procedure

1. Enable OpenShift Monitoring for the operands deployed in an IBM MQ container cluster.

a) Log in to the cluster as a cluster administrator.

b) Add "`enableUserWorkload: true`" to the configMap `cluster-monitoring-config` in the `openshift-monitoring` namespace. To do this, follow the steps in Enabling monitoring for user-defined projects in the Red Hat OpenShift documentation. If the `cluster-monitoring-config` ConfigMap does not exist in the `openshift-monitoring` namespace, follow the instructions to create it.

2. Install the OpenShift command line tool (oc).

   See Getting started with the OpenShift CLI.

3. Query the IBM MQ metrics using the OpenShift Monitoring CLI.

   When using the CLI, first declare the variables by using the **export** command. For example:

```
export OCP_LOGIN_TOKEN=$(oc whoami -t)
export THANOS_QUERIER_HOST=$(oc -n openshift-monitoring get route thanos-querier -o
jsonpath={.spec.host})
export QM_NAMESPACE=namespace-in-which-the-target-queue-manager-is-installed
export QM_METRIC_NAME=IBM-MQ-metric
```

   To run PromQL queries from the CLI, use the **PROMQL_QUERY** command. For example:

   - Get a list of queue managers that have a positive FDC count:

```
export QM_METRIC_NAME=ibmmq_qmgr_fdc_files
export PROMQL_QUERY="$QM_METRIC_NAME{namespace=\"$QM_NAMESPACE\"} > 0"
```

   - Get data for a specific time period:

```
export START=start_time
export END=end_time
export STEP=step_interval
curl -G -s -k -H "Authorization: Bearer $OCP_LOGIN_TOKEN" --data-urlencode
"query=$PROMQL_QUERY" \
--data-urlencode "start=$START" --data-urlencode "end=$END" \
--data-urlencode "step=$STEP" "https://$THANOS_QUERIER_HOST/api/v1/query_range"
```

     where *start_time* and *end_time* specify the time in ISO 8601 format, and *step_interval* is (for example) 15m, or 30s, or 1h.

   You can also export the data to a CSV file:

   a. The following command uses the same headers that display in the web console output shown in the screen capture in "Monitoring container clusters with the OpenShift Monitoring console" on page 225. To export to a CSV file, and pass the same headers that display in the web console output, first download and install jq.

   b. Run the following command:

```
(
echo
"metric_name,container,endpoint,instance,job,namespace,pod,prometheus,qmgr,service,timesta
mp,value"
curl -G -s -k -H "Authorization: Bearer $OCP_LOGIN_TOKEN" --data-urlencode
"query=$PROMQL_QUERY" \
--data-urlencode "start=$START" --data-urlencode "end=$END" --data-urlencode "step=$STEP"
\
"https://$THANOS_QUERIER_HOST/api/v1/query_range"| jq -r '.data.result[] | .metric as
$metric | .values[] | \
[$metric.__name__,
$metric.container,
$metric.endpoint,
$metric.instance,
$metric.job,
$metric.namespace,
$metric.pod,
$metric.prometheus,
$metric.qmgr,
$metric.service,
(.[0] | strftime("%Y-%m-%dT%H:%M:%SZ")), /* converts the timestamp from milliseconds to
ISO 8601 format */
.[1] ] | @csv'
) > your_filename.csv
```

CLI example: Write latency of Queue manager pods over the last 5 hours with a resolution of 10 minutes

```
export QM_METRIC_NAME=ibmmq_qmgr_log_write_latency_seconds
export PROMQL_QUERY="$QM_METRIC_NAME{namespace=\"$QM_NAMESPACE\"} [5h:10m]"
```

Export the data to a CSV file:

```
(
echo
"metric_name,container,endpoint,instance,job,namespace,pod,prometheus,qmgr,service,timestamp,val
ue"
curl -G -s -k -H "Authorization: Bearer $OCP_LOGIN_TOKEN" \
--data-urlencode "query=$PROMQL_QUERY" \
"https://$THANOS_QUERIER_HOST/api/v1/query" | jq -r '.data.result[] | .metric as $metric
| .values[] | \
[$metric.__name__,
$metric.container,
$metric.endpoint,
$metric.instance,
$metric.job,
$metric.namespace,
$metric.pod,
$metric.prometheus,
$metric.qmgr,
$metric.service,
(.[0] | strftime("%Y-%m-%dT%H:%M:%SZ")), /* converts the timestamp from milliseconds to ISO
8601 format */
.[1] ] | @csv'
) > your_filename.csv
```

Related tasks

"Surveillance lors de l'utilisation de IBM MQ Operator" on page 224
Les gestionnaires de files d'attente gérés par le IBM MQ Operator peuvent produire des métriques compatibles avec Prometheus. Les mesures du gestionnaire de file d'attente sont activées et fournies par défaut via HTTP.

"Monitoring container clusters with the OpenShift Monitoring console" on page 225
You can query the available container cluster metrics using the OpenShift Monitoring web console.

Related reference

"Indicateurs publiés par l' IBM MQ " on page 227
Les conteneurs du gestionnaire de files d'attente peuvent publier des mesures compatibles avec le monitoring Red Hat OpenShift.

Kubernetes V 9.4.1 Collecte d'informations de dépannage pour les gestionnaires de files d'attente déployés sur Kubernetes

Collecte des informations de dépannage à fournir à l'équipe d'assistance IBM lors de l'ouverture d'un nouveau dossier d'assistance.

Procédure

1. Recueillir des informations sur les fournisseurs de services en nuage.

   Il s'agit du fournisseur de cloud qui héberge votre Kubernetes cluster (par exemple, IBM Cloud).
2. Collecter les informations sur la version de Kubernetes
   Par exemple, exécutez `kubectl version -o yaml > kubeversion.yaml`
3. Collecter les ressources YAML décrivant les éléments clés de la manière dont IBM MQ est déployé.

   Inclure les éléments suivants :

   a. StatefulSet

   b. Pod (pour chaque instance de gestionnaire de file d'attente)

   c. Service

   d. PersistentVolumeClaim

   e. ConfigMaps

   Par exemple, exécutez `kubectl get pod` *PodName* `-o yaml` pour chaque gestionnaire de file d'attente Pod.
4. Recueillir une description de chaque Cosse, y compris des événements.

Par exemple, exécutez `kubectl describe pod` pour chaque module de gestion de la file d'attente.

5. Collecte de tous les journaux Pod
   Par exemple, exécutez `kubectl logs` *PodName* pour chaque gestionnaire de file d'attente Pod.
6. Recueillir les résultats de la commande **runmqras**
   Utilisez `kubectl exec` pour exécuter la commande **runmqras** à l'intérieur de chaque conteneur de gestionnaire de file d'attente.

Tâches associées

Collecte des informations de dépannage

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 Références de licences et d'API pour le IBM MQ Operator

IBM MQ fournit un Kubernetes Opérateur, qui fournit une intégration native avec Red Hat OpenShift Container Platform. L'API `v1beta1` peut être utilisée pour créer et gérer les ressources du gestionnaire de file d'attente.

mq.ibm.com/v1beta1: Versions actuelles de la licence

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-CYPF-CRPF3H`	`Production` ou `NonProduction`	IBM Cloud Pak for Integration 16.1.2	9.4.3
`L-MQQP-KBWMYE`	`Production` ou `NonProduction`	IBM Cloud Pak for Integration Édition limitée 16.1.2	9.4.3
`L-NUUP-23NH8Y`	`Production`	IBM MQ Advanced 9.4 - 02/2025	9.4.3 ou 9.4.2
`L-QWBN-T3PKLE`	`NonProduction`	IBM MQ Advanced for Non-Production Environment 9.4 - 02/2025	9.4.3 ou 9.4.2
`L-HYGL-6STWD6`	`Development`	IBM MQ Advanced for Developers 9.4, non garanti, 10/2024	9.4.3, 9.4.2, ou 9.4.1
`L-QYVA-B365MB`	`Production` ou `NonProduction`	IBM Cloud Pak for Integration 16.1.1	9.4.2 ou 9.4.1
`L-JVML-UFQVM4`	`Production` ou `NonProduction`	IBM Cloud Pak for Integration Edition limitée 16.1.1	9.4.2 ou 9.4.1
`L-JTPV-KYG8TF`	`Production` ou `NonProduction`	IBM Cloud Pak for Integration 16.1.0	9.9.4.1 ou 9.4.0
`L-BMSF-5YDSLR`	`Production` ou `NonProduction`	IBM Cloud Pak for Integration Edition limitée 16.1.0	9.9.4.1 ou 9.4.0
`L-EHXT-MQCRN9`	`Production`	IBM MQ Advanced 9.4	9.9.4.1 ou 9.4.0
`L-DRNQ-NGCA76`	`NonProduction`	IBM MQ Advanced for Non-Production Environment 9.4	9.4.1

Valeur de `spec.license.license`	Valeur de `spec.license.use`	Informations sur la licence	Versions IBM MQ applicables
`L-CLXQ-ADXTK3`	`Development`	IBM MQ Advanced for Developers (non justifié) 9.4	9.9.4.1 ou 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.

mq.ibm.com/v1beta1: Anciennes versions de la licence

Voir Anciennes versions de la licence dans la IBM MQ 9.3 documentation.

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 Documentation : https://ibm.biz/BdPZqj. Référence de la licence : https://ibm.biz/Bdm9be.

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 257

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.

Zone	Description
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.
`service` Service	Paramètres qui s'appliquent à tous les services QueueManager. La modification ou la suppression de cette section ou des champs qu'elle contient peut entraîner la régénération de tous les services QueueManager. La régénération du service perturbera la communication, ce qui entraînera des temps d'arrêt pour tous les types de QueueManager, y compris NativeHA. Les services régénérés se voient attribuer de nouvelles adresses ClusterIP par Kubernetes. Requiert MQ Operator 3.6.0 ou une version plus récente.
`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.
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.

Zone	Description
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 257

.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 257

Zone	Description
Chaîne `name`	Nom du référent. Ce champ est effectivement obligatoire, mais pour des raisons de compatibilité ascendante, il peut être vide. Les instances de ce type avec une valeur vide ici sont presque certainement erronées. Plus d'informations : https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names. Valeur par défaut : " ".

.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 257

.spec.license

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

Apparaît dans :

- «.spec», à la page 257

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. Voir https://ibm.biz/Bdm9be pour les valeurs valables.

Zone	Description
Chaîne `metric`	Paramètre qui spécifie la métrique de licence à utiliser. Par exemple, `ProcessorValueUnit`, `VirtualProcessorCore`, `ManagedVirtualServer` ou FREE. La valeur par défaut est `ProcessorValueUnit` lorsque vous utilisez une licence MQ Advanced ou une licence MQ Advanced for Non-Production, `VirtualProcessorCore` lorsque vous utilisez une licence Cloud Pak for Integration, et FREE lorsque vous utilisez une licence MQ Advanced for Developers.
Chaîne `use`	Paramètre qui contrôle la façon dont le logiciel sera utilisé, où la licence prend en charge plusieurs utilisations. Voir https://ibm.biz/Bdm9be pour les valeurs valables.

.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 257

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 260

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 260

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 260

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 261

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 257

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.
`env` Env tableau	Paramètres pour fournir des variables d'environnement pour le gestionnaire de files d'attente. Nécessite l' MQ. Opérateur 3.5.0 ou supérieur.
`files` Tableau des fichiers	Paramètres de fourniture des fichiers pour le gestionnaire de files d'attente. Nécessite l' MQ. Opérateur 3.5.0 ou supérieur.
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.

Zone	Description
Chaîne `logFormat`	Indique le format de journal à utiliser pour ce conteneur. Utilisez `JSON` pour les journaux au format JSON du conteneur. Utilisez `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.
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` Journaux de reprise	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 261

Zone	Description
`nativeHAGroups` Groupes autochtones	Paramètres des groupes HA natifs. Le site `type of availability` doit être `NativeHA`. Nécessite l' MQ. Opérateur 3.5.0 ou supérieur.
`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`. Voir https://ibm.biz/Bdm9TW pour plus de détails.

Zone	Description
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.nativeHAGroups

Paramètres des groupes HA natifs. Le site `type of availability` doit être `NativeHA`. Nécessite l' MQ. Opérateur 3.5.0 ou supérieur.

Apparaît dans :

- «.spec.queueManager.availability», à la page 262

Zone	Description
`local` Local	Paramètres de ce groupe d' NativeHA s.
`remotes` Tableau de télécommandes	Paramètres de connexion aux groupes d' NativeHA s distants.

.spec.queueManager.availability.nativeHAGroups.local

Paramètres de ce groupe d' NativeHA s.

Apparaît dans :

- «.spec.queueManager.availability.nativeHAGroups», à la page 263

Zone	Description
Chaîne `name`	Le nom de ce groupe d' NativeHA.
Chaîne `role`	Le rôle que ce groupe d' NativeHA s devrait adopter. Soit en direct, soit en différé. Les valeurs valides sont : `"Live"; "Recovery"`.
`route` Route	Paramètres de l'itinéraire du groupe d' NativeHA.
`tls` Tls	Paramètres TLS pour configurer une communication sécurisée entre les groupes d' NativeHA. Requis si TLS n'est pas configuré dans la section parent Availability.

.spec.queueManager.availability.nativeHAGroups.local.route

Paramètres de l'itinéraire du groupe d' NativeHA.

Apparaît dans :

- «.spec.queueManager.availability.nativeHAGroups.local», à la page 263

Zone	Description
Valeur booléenne `enabled`	Indique si la route doit être activée ou non. La valeur par défaut est true.

.spec.queueManager.availability.nativeHAGroups.local.tls

Paramètres TLS pour configurer une communication sécurisée entre les groupes d' NativeHA. Requis si TLS n'est pas configuré dans la section parent Availability.

Apparaît dans :

- «.spec.queueManager.availability.nativeHAGroups.local», à la page 263

Zone	Description
Chaîne `cipherSpec`	Le nom de l' CipherSpec e pour le groupe TLS d' NativeHA.
`key` Clé	Paramètres de la clé TLS utilisée dans la communication sécurisée entre les groupes Native HA.

.spec.queueManager.availability.nativeHAGroups.local.tls.key

Paramètres de la clé TLS utilisée dans la communication sécurisée entre les groupes Native HA.

Apparaît dans :

- «.spec.queueManager.availability.nativeHAGroups.local.tls», à la page 264

Zone	Description
Chaîne `secretName`	Nom du secret Kubernetes. Le secret doit être de type " kubernetes.io/tls " et contenir une clé privée unique, un certificat public et une chaîne de confiance.

.spec.queueManager.availability.nativeHAGroups.télécommandes

Apparaît dans :

- «.spec.queueManager.availability.nativeHAGroups», à la page 263

Zone	Description
Tableau `addresses`	Un ensemble de noms d'hôtes et d'adresses de ports pour communiquer avec le groupe d' NativeHA s à distance. Uniquement efficace lorsque le rôle de groupe local est « `Live` ».
Valeur booléenne `enabled`	S'il faut ou non commencer la communication avec la télécommande. Uniquement compatible avec le groupe d' NativeHA s local ayant le rôle « Live ». Si ce paramètre n'est pas défini, le conteneur d' MQ s utilisera la valeur par défaut interne.
Chaîne `name`	Nom du groupe d' NativeHA s à distance. Ce nom doit correspondre au nom du groupe d' NativeHA s configuré dans le cluster distant.
`retries` Nouvelles tentatives	Configuration du comportement de la télécommande en cas de nouvel essai.
Entier `statusInterval`	Intervalle de temps le plus long en millisecondes que le groupe de récupération attendra avant d'envoyer un statut de journal de récupération.
`trust` Tableau de confiance	Certificats à ajouter au référentiel de clés du gestionnaire de file d'attente pour la communication avec le groupe d' NativeHA s à distance.

.spec.queueManager.availability.nativeHAGroups.remotes.retries

Configuration du comportement de la télécommande en cas de nouvel essai.

Apparaît dans :

- «.spec.queueManager.availability.nativeHAGroups.télécommandes», à la page 264

Zone	Description
`long` Longue	Réessayer la configuration pour la longue phase de réessais.
Chaîne `reporting`	S'il faut signaler toutes les tentatives de relance ou seulement la phase de relance.
`short` Court	Réessayer la configuration pour la phase de réessai courte.

.spec.queueManager.availability.nativeHAGroups.remotes.retries.long

Réessayer la configuration pour la longue phase de réessais.

Apparaît dans :

- «.spec.queueManager.availability.nativeHAGroups.remotes.retries», à la page 264

Zone	Description
Entier `interval`	Intervalle en millisecondes à respecter entre les tentatives de connexion au Remote pendant la phase de réessais longs.

.spec.queueManager.availability.nativeHAGroups.remotes.retries.short

Réessayer la configuration pour la phase de réessai courte.

Apparaît dans :

- «.spec.queueManager.availability.nativeHAGroups.remotes.retries», à la page 264

Zone	Description
Entier `count`	Le nombre de tentatives pour établir une connexion pendant la courte phase de réessai.
Entier `interval`	Intervalle en millisecondes à respecter entre les tentatives de connexion au Remote pendant la phase de réessai court.

.spec.queueManager.availability.nativeHAGroups.remotes.trust

NativeHAGroupTrust définit une source de clés publiques d'autorité de certification.

Apparaît dans :

- «.spec.queueManager.availability.nativeHAGroups.télécommandes», à la page 264

Zone	Description
`configMap` Carte de configuration	Fournir un certificat CA en utilisant une carte de configuration Kubenetes.
`secret` Secret	Fournir un certificat CA utilisant un secret de l' Kubernetes.

.spec.queueManager.availability.nativeHAGroups.remotes.trust.configMap

Fournir un certificat CA en utilisant une carte de configuration Kubenetes.

Apparaît dans :

- «.spec.queueManager.availability.nativeHAGroups.remotes.trust», à la page 265

Zone	Description
Tableau `items`	Clés à l'intérieur de la source configmap d' Kubernetes, qui doivent être appliquées. Si non spécifié, utilise toutes les clés.

Zone	Description
Chaîne `name`	Le nom de la source de configmap d' Kubernetes.

.spec.queueManager.availability.nativeHAGroups.remotes.trust.secret

Fournir un certificat CA utilisant un secret de l' Kubernetes.

Apparaît dans :

- «.spec.queueManager.availability.nativeHAGroups.remotes.trust», à la page 265

Zone	Description
Tableau `items`	Clés à l'intérieur de l' Kubernetes e secret qui devrait être appliqué. Si non spécifié, utilise toutes les clés.
Chaîne `secretName`	Nom du secret Kubernetes.

.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 262

Zone	Description
Chaîne `cipherSpec`	Nom du CipherSpec pour une connexion TLS NativeHA.
Chaîne `secretName`	Nom du secret Kubernetes.

.spec.queueManager.env

EnvVar représente une variable d'environnement présente dans un Conteneur.

Apparaît dans :

- «.spec.queueManager», à la page 261

Zone	Description
Chaîne `name`	Nom de la variable d'environnement. Doit être un C_IDENTIFIER.
Chaîne `value`	Les références variables $(VAR_NAME) sont développées à l'aide des variables d'environnement précédemment définies dans le conteneur et de toutes les variables d'environnement de service. Si une variable ne peut pas être résolue, la référence dans la chaîne d'entrée restera inchangée. Les doubles $$ sont réduits à un simple $, ce qui permet d'échapper à la syntaxe $(VAR_NAME) : c'est-à-dire `$$(VAR_NAME)` produira le littéral de chaîne `$(VAR_NAME)`. Les références échappées ne seront jamais développées, que la variable existe ou non. Par défaut, .
`valueFrom` Valeur de	Source de la valeur de la variable d'environnement. Ne peut pas être utilisé si la valeur n'est pas vide.

.spec.queueManager.env.valueFrom

Source de la valeur de la variable d'environnement. Ne peut pas être utilisé si la valeur n'est pas vide.

Apparaît dans :

- «.spec.queueManager.env», à la page 266

Zone	Description
`configMapKeyRef` Configmapkeyref	Sélectionne une touche d'un clavier (ConfigMap).
`fieldRef` Fieldref	Sélectionne un champ du pod : prend en charge metadata.name, metadata.namespace, `metadata.labels['<KEY>']`, `metadata.annotations['<KEY>']`, spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.
`resourceFieldRef` Ressourcefieldref	Sélectionne une ressource du conteneur : seules les limites et les demandes de ressources (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory et requests.ephemeral-storage) sont actuellement prises en charge.
`secretKeyRef` Secretkeyref	Sélectionne une clé d'un secret dans l'espace de noms du pod.

.spec.queueManager.env.valueFrom.configMapKeyRef

Sélectionne une touche d'un clavier (ConfigMap).

Apparaît dans :

- «.spec.queueManager.env.valueFrom», à la page 266

Zone	Description
Chaîne `key`	La clé pour choisir.
Chaîne `name`	Nom du référent. Ce champ est effectivement obligatoire, mais pour des raisons de compatibilité ascendante, il peut être vide. Les instances de ce type avec une valeur vide ici sont presque certainement erronées. Plus d'informations : https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names. Valeur par défaut : `" "`.
Valeur booléenne `optional`	Préciser si l' ConfigMap e ou sa clé doit être définie.

.spec.queueManager.env.valueFrom.fieldRef

Sélectionne un champ du pod : prend en charge metadata.name, metadata.namespace, `metadata.labels['<KEY>']`, `metadata.annotations['<KEY>']`, spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.

Apparaît dans :

- «.spec.queueManager.env.valueFrom», à la page 266

Zone	Description
Chaîne `apiVersion`	La version du schéma dans lequel l' FieldPath e est écrite, par défaut, est `v1`.
Chaîne `fieldPath`	Chemin du champ à sélectionner dans la version d'API spécifiée.

.spec.queueManager.env.valueFrom.resourceFieldRef

Sélectionne une ressource du conteneur : seules les limites et les demandes de ressources (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory et requests.ephemeral-storage) sont actuellement prises en charge.

Apparaît dans :

- «.spec.queueManager.env.valueFrom», à la page 266

Zone	Description
Chaîne `containerName`	Nom du conteneur : obligatoire pour les volumes, facultatif pour les variables d'environnement.
`divisor`	Spécifie le format de sortie des ressources exposées, par défaut 1.
Chaîne `resource`	Requis : ressource à sélectionner.

.spec.queueManager.env.valueFrom.secretKeyRef

Sélectionne une clé d'un secret dans l'espace de noms du pod.

Apparaît dans :

- «.spec.queueManager.env.valueFrom», à la page 266

Zone	Description
Chaîne `key`	La clé du secret pour choisir. Doit être une clé secrète valide.
Chaîne `name`	Nom du référent. Ce champ est effectivement obligatoire, mais pour des raisons de compatibilité ascendante, il peut être vide. Les instances de ce type avec une valeur vide ici sont presque certainement erronées. Plus d'informations : https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names. Valeur par défaut : " ".
Valeur booléenne `optional`	Indiquez si le Secret ou sa clé doivent être définis.

.spec.queueManager.fichiers

Source des fichiers.

Apparaît dans :

- «.spec.queueManager», à la page 261

Zone	Description
`configMap` Carte de configuration	ConfigMap représente un Kubernetes ConfigMap qui contient des fichiers.
`secret` Secret	Secret représente un secret d' Kubernetes s qui contient des fichiers.

.spec.queueManager.files.configMap

ConfigMap représente un Kubernetes ConfigMap qui contient des fichiers.

Apparaît dans :

- «.spec.queueManager.fichiers», à la page 268

Zone	Description
Chaîne `defaultMountPath`	Chemin de montage par défaut à utiliser pour les clés dans la source d' Kubernetes. Il est possible de passer outre en spécifiant des chemins de montage individuels pour les clés. Si non spécifié, `/etc/mqm/customfiles` sera utilisé.
`items` Matrice d'éléments	Les clés à l'intérieur de la source d' Kubernetes s qui doivent être appliquées.
Chaîne `name`	Nom de la source Kubernetes.

.spec.queueManager.files.configMap.articles

Apparaît dans :

- «.spec.queueManager.files.configMap», à la page 268

Zone	Description
Chaîne `item`	Nom de la clé.
Chaîne `mountPath`	Le chemin de montage à utiliser pour la clé remplace le chemin de montage par défaut.

.spec.queueManager.files.secret

Secret représente un secret d' Kubernetes s qui contient des fichiers.

Apparaît dans :

- «.spec.queueManager.fichiers», à la page 268

Zone	Description
Chaîne `defaultMountPath`	Chemin de montage par défaut à utiliser pour les clés dans la source d' Kubernetes. Il est possible de passer outre en spécifiant des chemins de montage individuels pour les clés. Si non spécifié, `/etc/mqm/customfiles` sera utilisé.
`items` Matrice d'éléments	Les clés à l'intérieur de la source d' Kubernetes s qui doivent être appliquées.
Chaîne `name`	Nom de la source Kubernetes.

.spec.queueManager.files.secret.items

Apparaît dans :

- «.spec.queueManager.files.secret», à la page 269

Zone	Description
Chaîne `item`	Nom de la clé.
Chaîne `mountPath`	Le chemin de montage à utiliser pour la clé remplace le chemin de montage par défaut.

.spec.queueManager.ini

Source des fichiers de configuration INI.

Apparaît dans :

- «.spec.queueManager», à la page 261

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 269

Zone	Description
Tableau `items`	Les clés à l'intérieur de la source d' Kubernetes s 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 269

Zone	Description
Tableau `items`	Les clés à l'intérieur de la source d' Kubernetes s 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 261

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 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 261

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.
`serviceMonitor` Servicemoniteur	Paramètres du moniteur de service pour les mesures compatibles avec Prometheus. Requiert MQ Operator 3.6.0 ou une version plus récente.
`tls` Tls	Paramètres TLS pour configurer une communication sécurisée pour les métriques compatibles avec l' Prometheus. Nécessite l' MQ. Opérateur 3.5.0 ou supérieur.

.spec.queueManager.metrics.serviceMonitor

Paramètres du moniteur de service pour les mesures compatibles avec Prometheus. Requiert MQ Operator 3.6.0 ou une version plus récente.

Apparaît dans :

- «.spec.queueManager.metrics», à la page 270

Zone	Description
Valeur booléenne `enabled`	Activation ou non du moniteur de service. La valeur par défaut est true.

.spec.queueManager.metrics.tls

Paramètres TLS pour configurer une communication sécurisée pour les métriques compatibles avec l' Prometheus. Nécessite l' MQ. Opérateur 3.5.0 ou supérieur.

Apparaît dans :

- «.spec.queueManager.metrics», à la page 270

Zone	Description
Chaîne `provider`	Le fournisseur à utiliser pour configurer la communication sécurisée pour les métriques compatibles avec l' Prometheus. La valeur par défaut est none. Utilisez none pour des mesures d' HTTP s non sécurisées. Utilisez `openshift` pour utiliser des certificats OpenShift-provided avec un ServiceMonitor.

.spec.queueManager.mqsc

Source des fichiers de configuration MQSC.

Apparaît dans :

- «.spec.queueManager», à la page 261

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 271

Zone	Description
Tableau `items`	Les clés à l'intérieur de la source d' Kubernetes s 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 271

Zone	Description
Tableau `items`	Les clés à l'intérieur de la source d' Kubernetes s 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 261

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 261

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 261

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 273

Zone	Description
`cpu`	
`memory`	

.spec.queueManager.resources.requests

Paramètres d'unité centrale et de mémoire.

Apparaît dans :

- «.spec.queueManager.resources», à la page 273

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 261

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 261

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.
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 261

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.

Zone	Description
`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 274

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 274

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.

Zone	Description
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.
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 274

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 274

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 274

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 257

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 combinés par OU avec 0660. Si cette option n'est pas définie, la Kubelet ne modifiera pas la propriété et les autorisations d'un 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.service

Paramètres qui s'appliquent à tous les services QueueManager. La modification ou la suppression de cette section ou des champs qu'elle contient peut entraîner la régénération de tous les services QueueManager. La régénération du service perturbera la communication, ce qui entraînera des temps d'arrêt pour tous les types de QueueManager, y compris NativeHA. Les services régénérés se voient attribuer de nouvelles adresses ClusterIP par Kubernetes. Requiert MQ Operator 3.6.0 ou une version plus récente.

Apparaît dans :

- «.spec», à la page 257

Zone	Description
Tableau `ipFamilies`	Spécifie les familles IP à utiliser pour tous les services QueueManager. Les options valables sont IPv4 et IPv6. Si elle n'est pas spécifiée, Kubernetes définira une valeur par défaut basée sur le paramètre ipFamilyPolicy. S'il est défini manuellement, la création du service échouera si la famille IP n'est pas disponible sur le cluster ou si les valeurs sont incompatibles avec le paramètre ipFamilyPolicy. La modification de la première valeur du tableau (la famille IP primaire) ou la suppression de ce champ entraînera la régénération de tous les services QueueManager. La régénération du service perturbera la communication, ce qui entraînera des temps d'arrêt pour tous les types de QueueManager, y compris NativeHA. Les services régénérés se voient attribuer de nouvelles adresses ClusterIP par Kubernetes.
Chaîne `ipFamilyPolicy`	Spécifie la politique de la famille IP à utiliser pour tous les services QueueManager. Les options valables sont SingleStack, PreferDualStack et RequireDualStack. SingleStack est pour une seule famille IP. PreferDualStack est pour deux familles IP sur des clusters configurés en double pile ou une seule famille IP sur des clusters à pile unique. RequireDualStack est pour deux familles IP sur des clusters configurés en double pile, sinon il échoue. La suppression de ce champ entraînera la régénération de tous les services QueueManager. La régénération du service perturbera la communication, ce qui entraînera des temps d'arrêt pour tous les types de QueueManager, y compris NativeHA. Les services régénérés se voient attribuer de nouvelles adresses ClusterIP par Kubernetes.

.spec.telemetry

Paramètres de configuration d'Open Telemetry. Requiert MQ Operator 2.2.0 ou version ultérieure.

Apparaît dans :

- «.spec», à la page 257

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 278

Zone	Description
`instana` Instana	Paramètres du traçage Instana.

.spec.telemetry.tracing.instana

Paramètres du traçage Instana.

Apparaît dans :

- «.spec.telemetry.tracing», à la page 278

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 257

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 257

Zone	Description
`agent` TracingAgent	Dans Cloud Pak for Integration uniquement, vous pouvez configurer des paramètres pour l'agent de trace facultatif.
`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 279

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 279

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 279

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.

Zone	Description
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 279

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 281

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.readinessProbe

Paramètres qui contrôlent la sonde de vigilance.

Apparaît dans :

- «.spec.tracing.collector», à la page 281

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 257

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` Configuration manuelle	Paramètres permettant de fournir la configuration XML du serveur Web. Requiert MQ Operator 3.0.0 ou version ultérieure.
`route` Route	Paramètres de la route du serveur web. Requiert MQ Operator 3.6.0 ou une version plus récente.

.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 282

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 282

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 282

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 282

Zone	Description
`configMap` Carte de configuration	ConfigMap représente un objet Kubernetes ConfigMap qui contient la configuration XML du serveur Web.
`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 une meilleure sécurité, encodez les informations 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 283

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 une meilleure sécurité, encodez les informations d'identification à l'aide de “securityUtility”

Apparaît dans :

- «.spec.web.manualConfig», à la page 283

Zone	Description
Chaîne `name`	Nom de la source Kubernetes.

.spec.web.route

Paramètres de la route du serveur web. Requiert MQ Operator 3.6.0 ou une version plus récente.

Apparaît dans :

- «.spec.web», à la page 282

Zone	Description
Valeur booléenne `enabled`	Indique si la route doit être activée ou non. La valeur par défaut est true.

.statut

Etat observé pour le gestionnaire de files d'attente.

Apparaît dans :

- «QueueManager», à la page 257

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.
Valeur booléenne `customImages`	Indique si l'image du gestionnaire de files d'attente sélectionnée par l'opérateur a été remplacée par une image personnalisée.
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.
`nativeHACRR` Nativehacrr	Informations sur la réplication interrégionale native HA configurée.
Chaîne `phase`	Phase de l'état du gestionnaire de files d'attente.
`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 284

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 284

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. Les valeurs valides sont : `"True"; "False"; "Unknown"`.
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 284

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 284

Zone	Description
`integrationKeycloak` Intégrationkeycloak	QueueManagerStatusIntegrationKeycloak définit l'état de la clé d'Keycloak pour le QueueManager.

.status.metadata.integrationKeycloak

QueueManagerStatusIntegrationKeycloak définit l'état de la clé d'Keycloak pour le QueueManager.

Apparaît dans :

- «.status.metadata», à la page 285

Zone	Description
Chaîne `clientName`	

.status.nativeHACRR

Informations sur la réplication interrégionale native HA configurée.

Apparaît dans :

- «.statut», à la page 284

Zone	Description
`local` Local	Statut du cluster local.

.status.nativeHACRR.local

Statut du cluster local.

Apparaît dans :

- «.status.nativeHACRR», à la page 286

Zone	Description
Chaîne `address`	Adresse du cluster local.
Chaîne `groupName`	Nom de groupe du cluster local.

.status.versions

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

Apparaît dans :

- «.statut», à la page 284

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 286

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 286

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 287

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.

OpenShift CP4I 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[2]	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.

[2] 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[3]	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[4]	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 IBM MQ annotations sur la licence du conteneur

Dans les environnements Kubernetes et Red Hat OpenShift Container Platform , IBM License Service utilise les annotations Pod pour suivre l'utilisation en fonction des limites définies pour le conteneur, plutôt que pour la machine sous-jacente. Le site IBM MQ Operator définit ces annotations Pod pour vous, mais lorsque vous utilisez d'autres moyens de déploiement, vous devez les définir vous-même.

Pour plus d'informations sur la façon dont IBM MQ Operator annote les pods en fonction des valeurs du YAML d'un gestionnaire de file d'attente, voir «IBM MQ les annotations de licence du conteneur qui sont appliquées par l'outil IBM MQ Operator», à la page 295.

---

[3] 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`.

[4] 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`.

Lors du déploiement d'un conteneur IBM MQ , il existe deux approches courantes en matière de licences :

- Licence sur l'ensemble de la machine qui exécute le conteneur.
- Octroyer une licence au conteneur en fonction des limites qui lui sont associées (Kubernetes et Red Hat OpenShift Container Platform uniquement)

Les deux options sont disponibles pour les clients et de plus amples informations sont disponibles sur la page IBM Container Licenses à l 'adresse suivante : 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 292
- «IBM MQ Avancé», à la page 292
- «IBM MQ pour l'environnement hors production», à la page 292
- «IBM MQ Advanced pour l'environnement de non-production», à la page 293
- «IBM MQ Advanced pour les développeurs», à la page 293

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 291.

- «Instance multiple de conteneur IBM MQ», à la page 293
- «IBM MQ Advanced Container multi-instance», à la page 293
- «IBM MQ Container Multi-Instance pour l'environnement de non-production», à la page 293
- «IBM MQ Conteneur avancé Multi Instance pour l'environnement de non-production», à la page 293

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 293
- «Autorisation d'utilisation de IBM MQ Advanced with CP4I», à la page 293
- «IBM MQ for Non-Production Environment avec autorisation d'utilisation CP4I», à la page 294
- «Autorisation d'utilisation d' IBM MQ Advanced for Non-Production Environment avec CP4I», à la page 294

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 291.

- «IBM MQ Container Multi Instance avec autorisation d'utilisation CP4I», à la page 294
- «IBM MQ Advanced Container Multi Instance avec autorisation d'utilisation CP4I», à la page 294
- «IBM MQ Container Multi Instance for Non-Production Environment avec autorisation d'utilisation CP4I», à la page 294
- «IBM MQ Advanced Container Multi Instance for Non-Production Environment avec autorisation d'utilisation CP4I», à la page 294

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 293
  - «IBM MQ Advanced Container multi-instance», à la page 293
  - «IBM MQ Container Multi-Instance pour l'environnement de non-production», à la page 293
  - «IBM MQ Conteneur avancé Multi Instance pour l'environnement de non-production», à la page 293
- IBM Cloud Pak for Integration autorisation
  - «IBM MQ Container Multi Instance avec autorisation d'utilisation CP4I», à la page 294
  - «IBM MQ Advanced Container Multi Instance avec autorisation d'utilisation CP4I», à la page 294
  - «IBM MQ Container Multi Instance for Non-Production Environment avec autorisation d'utilisation CP4I», à la page 294
  - «IBM MQ Advanced Container Multi Instance for Non-Production Environment avec autorisation d'utilisation CP4I», à la page 294

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[5]
- 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)[5]
- Instance multiple de conteneur IBM MQ
  - 1 x IBM MQ **et** 1 x IBM MQ High Availability Replica **ou**
  - 2 x IBM MQ[5]
- IBM MQ Container Multi-Instance pour l'environnement de non-production
  - 1 x IBM MQ **et** 1 x IBM MQ High Availability Replica **ou**

[5] 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.

– 2 x IBM MQ pour l'environnement hors production) [5]

IBM MQ HA autochtone

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 292
  - «IBM MQ Advanced pour l'environnement de non-production», à la page 293
- IBM Cloud Pak for Integration autorisation
  - «Autorisation d'utilisation de IBM MQ Advanced with CP4I», à la page 293
  - «Autorisation d'utilisation d' IBM MQ Advanced for Non-Production Environment avec CP4I», à la page 294

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"
```

Pour utiliser les fonctionnalités Native HA et Cross-Region Replication sur les gestionnaires de file d'attente qui utilisent ces annotations de licence, consultez la rubrique correspondant à votre environnement :

- «Configurer Native HA et Cross-Region Replication sur IBM MQ Queue Managers sous licence en utilisant la fonction IBM MQ Operator», à la page 119
- «Configuration de Native HA et de Cross-Region Replication sur IBM MQ gestionnaires de files d'attente sous licence dans Kubernetes», à la page 192

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"
```

Pour utiliser les fonctionnalités Native HA et Cross-Region Replication sur les gestionnaires de file d'attente qui utilisent ces annotations de licence, consultez la rubrique correspondant à votre environnement :

- «Configurer Native HA et Cross-Region Replication sur IBM MQ Queue Managers sous licence en utilisant la fonction IBM MQ Operator», à la page 119
- «Configuration de Native HA et de Cross-Region Replication sur IBM MQ gestionnaires de files d'attente sous licence dans Kubernetes», à la page 192

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

IBM MQ Conteneur avancé Multi Instance pour l'environnement de non-production

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

Linux IBM MQ les annotations de licence du conteneur qui sont appliquées par l'outil IBM MQ Operator

Dans les environnements Kubernetes et Red Hat OpenShift Container Platform , IBM License Service utilise les annotations Pod pour suivre l'utilisation en fonction des limites définies pour le conteneur, plutôt que pour la machine sous-jacente. Le site IBM MQ Operator définit ces annotations Pod pour vous, mais lorsque vous utilisez d'autres moyens de déploiement, vous devez les définir vous-même.

Pour les annotations disponibles, voir «IBM MQ annotations sur la licence du conteneur», à la page 289. Le tableau suivant décrit les annotations qui sont appliquées à un pod par IBM MQ Operator, en fonction de ces valeurs dans le YAML d'un gestionnaire de file d'attente :

- La valeur de `spec.license.license`
- La valeur de `spec.license.use`
- La valeur de `spec.queueManager.availability.type`

Remarque : Ce tableau se réfère aux dernières licences de CD, pour IBM MQ 9.4.3.0. Pour les licences des autres versions de IBM MQ , voir «Références de licences et d'API pour le IBM MQ Operator», à la page 256.

Commentaire	Value of `spec.license.license`	Valeur de `spec.license.use`	Valeur de `spec.queueManager.availability.type`
IBM MQ Advanced	L-NUUP-23NH8Y	`Production`	`SingleInstance` ou `NativeHA`
IBM MQ Advanced for Non-Production Environment	L-QWBN-T3PKLE	`NonProduction`	`SingleInstance` ou `NativeHA`
IBM MQ Advanced for Developers	L-HYGL-6STWD6	`Development`	`SingleInstance` ou `NativeHA`
IBM MQ Advanced Container Multi Instance	L-NUUP-23NH8Y	`Production`	`MultiInstance`
IBM MQ Advanced Container Multi Instance for Non-Production Environment	L-QWBN-T3PKLE	`NonProduction`	`MultiInstance`
IBM MQ Advancedavec CP4I droit	L-CYPF-CRPF3H	`Production`	`SingleInstance` ou `NativeHA`
IBM MQ Advanced for Non-Production Environment avec CP4I droit	L-CYPF-CRPF3H	`NonProduction`	`SingleInstance` ou `NativeHA`
IBM MQ Advanced Container Multi Instance avec CP4I droit	L-CYPF-CRPF3H	`Production`	`MultiInstance`
IBM MQ Advanced Container Multi Instance for Non-Production Environment avec CP4I droit	L-CYPF-CRPF3H	`NonProduction`	`MultiInstance`

Linux V 9.4.1 IBM MQ Advanced image de conteneur

Une image de conteneur préconstruite est disponible pour IBM MQ Advanced. Cette image est disponible sur le site IBM Container Registry.

Utilisation

Pour pouvoir utiliser l'image, vous devez accepter les termes de la licence IBM MQ Advanced 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 IBM MQ Advanced for Developers conditions de licence.

Définissez `view` pour afficher les conditions de licence.

MQ_CMDLEVEL
Définit le niveau de commande MQ à utiliser.

MQ_ENABLE_EMBEDDED_WEB_SERVER
Défini à `true` pour démarrer le serveur web au démarrage du conteneur.

MQ_ENABLE_FIPS
La valeur " `true`" permet d'activer le mode FIPS . Régler à `false` pour désactiver le mode FIPS . Par défaut, le mode FIPS est automatiquement activé si l'hôte Linux sous-jacent a activé FIPS , mais cette option peut être utile pour les tests.

MQ_ENABLE_METRICS
Définissez `true` pour générer des métriques Prometheus pour votre gestionnaire de file d'attente.

MQ_ENABLE_TRACE_CRTMQDIR
La valeur "true" permet de créer une trace de diagnostic MQ lors de l'exécution de la commande **crtmqdir**.

MQ_ENABLE_TRACE_CRTMQM
La valeur "true" permet de créer une trace de diagnostic MQ lors de l'exécution de la commande **crtmqm**.

MQ_ENABLE_TRACE_STRMQM
La valeur "true" permet de créer une trace de diagnostic MQ lors de l'exécution de la commande **strmqm**.

MQ_GRACE_PERIOD
L'heure cible à laquelle la fin du gestionnaire de file d'attente est tentée, en escaladant les phases de déconnexion de l'application. Définit l'option **endmqm -tp** utilisée lorsque le conteneur se termine. Voir endmqm (end queue manager) pour plus d'informations.

MQ_LOGGING_CONSOLE_SOURCE
Spécifiez une liste de sources séparées par des virgules pour les journaux qui sont mis en miroir à l'emplacement du conteneur **stdout** .

Les valeurs valides sont `qmgr`, `web` et `mqsc`.

La valeur par défaut est `qmgr,web`.

La valeur optionnelle est `mqsc`. Cette option peut être utilisée pour refléter le contenu de `autocfgmqsc.LOG` dans le journal du conteneur.

MQ_LOGGING_CONSOLE_FORMAT
Modifier le format des journaux qui sont imprimés à l'emplacement **stdout** du conteneur.

Définissez `basic` pour utiliser un format simple et lisible par l'homme. 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 messages séparés par des virgules pour les messages à exclure.

Les messages 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`.

V 9.4.3 **MQ_LOGGING_METRICS_AUDIT_ENABLED**

La valeur "true" permet d'activer la journalisation de l'accès au point de terminaison Prometheus metrics. La sortie du journal est envoyée dans un fichier JSON à l'adresse `/var/mqm/errors/`. Vous devez également définir `MQ_ENABLE_METRICS=true`, pour générer les mesures Prometheus.

MQ_MULTI_INSTANCE

La valeur "true" permet de fonctionner en tant que gestionnaire de file d'attente multi-instances. Ceci modifie les options utilisées avec **endmqm**.

MQ_NATIVE_HA

La valeur "true" permet d'activer Native HA. Ceci définit l'option **crtmqm -lr**. Voir crtmqm (create queue manager) pour plus d'informations. Vous pouvez configurer d'autres paramètres de Native HA en montant un fichier INI. Voir la strophe 'NativeHAInstance du fichier qm.ini et la strophe 'NativeHALocalInstance du fichier qm.ini.

Deprecated **MQ_NATIVE_HA_INSTANCE_0_NAME**

Définit l'attribut 'Name dans la strophe INI 'NativeHAInstance pour l'une des trois instances Native HA. Voir la strophe 'NativeHAInstance du fichier qm.ini. A partir de 'IBM MQ 9.4.1, cette variable est obsolète, et vous devez à la place fournir un fragment de fichier INI dans '/etc/mqm.

Deprecated **MQ_NATIVE_HA_INSTANCE_1_NAME**

Définit l'attribut 'Name dans la strophe INI 'NativeHAInstance pour l'une des trois instances Native HA. Voir la strophe 'NativeHAInstance du fichier qm.ini. A partir de 'IBM MQ 9.4.1, cette variable est obsolète, et vous devez à la place fournir un fragment de fichier INI dans '/etc/mqm.

Deprecated **MQ_NATIVE_HA_INSTANCE_2_NAME**

Définit l'attribut 'Name dans la strophe INI 'NativeHAInstance pour l'une des trois instances Native HA. Voir la strophe 'NativeHAInstance du fichier qm.ini. A partir de 'IBM MQ 9.4.1, cette variable est obsolète, et vous devez à la place fournir un fragment de fichier INI dans '/etc/mqm.

Deprecated **MQ_NATIVE_HA_INSTANCE_0_REPLICATION_ADDRESS**

Définit l'attribut 'ReplicationAddress dans la strophe INI 'NativeHAInstance pour l'une des trois instances Native HA. Voir la strophe 'NativeHAInstance du fichier qm.ini. A partir de 'IBM MQ 9.4.1, cette variable est obsolète, et vous devez à la place fournir un fragment de fichier INI dans '/etc/mqm.

Deprecated **MQ_NATIVE_HA_INSTANCE_1_REPLICATION_ADDRESS**

Définit l'attribut 'ReplicationAddress dans la strophe INI 'NativeHAInstance pour l'une des trois instances Native HA. Voir la strophe 'NativeHAInstance du fichier qm.ini. A partir de 'IBM MQ 9.4.1, cette variable est obsolète, et vous devez à la place fournir un fragment de fichier INI dans '/etc/mqm.

Deprecated **MQ_NATIVE_HA_INSTANCE_2_REPLICATION_ADDRESS**

Définit l'attribut 'ReplicationAddress dans la strophe INI 'NativeHAInstance pour l'une des trois instances Native HA. Voir la strophe 'NativeHAInstance du fichier qm.ini. A partir de 'IBM MQ 9.4.1, cette variable est obsolète, et vous devez à la place fournir un fragment de fichier INI dans '/etc/mqm.

Deprecated **MQ_NATIVE_HA_CIPHERSPEC**

Définit l'attribut 'CipherSpec dans la strophe INI 'NativeHALocalInstance pour l'une des trois instances Native HA. Voir la strophe 'NativeHALocalInstance du fichier qm.ini. A partir de 'IBM MQ 9.4.1, cette variable est obsolète, et vous devez à la place fournir un fragment de fichier INI dans '/etc/mqm.

Deprecated **MQ_NATIVE_HA_KEY_REPOSITORY**
Ignorer le dépôt de clés généré automatiquement pour Native HA et utiliser celui spécifié. À partir de " IBM MQ 9.4.1, cette variable est obsolète, et vous devez plutôt fournir l'emplacement du dépôt de clés dans un fragment de fichier INI dans " `/etc/mqm`.

MQ_QMGR_LOG_FILE_PAGES
MQ Les données du journal de récupération sont conservées dans une série de fichiers appelés fichiers journaux. La taille de ces fichiers est définie en unité de pages de 4 ko. Cette variable d'environnement définit l'option **`crtmqm -lf`**. Voir crtmqm (create queue manager) pour plus d'informations.

mq_qmgr_name
Définissez le nom avec lequel vous souhaitez que votre gestionnaire de file d'attente soit créé.

Système de fichiers et points de montage

`/etc/mqm`
Tous les fichiers MQSC ou INI contenus dans ce répertoire seront traités par le gestionnaire de files d'attente lors de son démarrage. Il s'agit du répertoire utilisé par la fonction de configuration automatique. Voir Configuration automatique à partir d'un script MQSC au démarrage et Configuration automatique de qm.ini au démarrage pour plus d'informations.

`/etc/mqm/pki/keys`
Contient des sous-répertoires avec des clés privées sous forme PKCS #1, ASN.1 DER ou PKCS #8, ASN.1 DER non chiffrées. Chaque sous-répertoire situé sous `/etc/mqm/pki/keys` est analysé à la recherche de fichiers portant l'extension `.key`. La chaîne de certificats X.509 publics doit se trouver dans le même répertoire, dans un ou plusieurs fichiers portant l'extension " `.crt`, au format ASN.1 DER. Si les clés sont montées de cette manière, un dépôt de clés est automatiquement généré dans " `/run/runmqserver/tls`. Le référentiel de clés sera défini dans le paramètre " SSLKEYR du gestionnaire de file d'attente, et la clé par défaut (CERTLABL) sera le premier nom après le tri lexicographique de la liste des noms de clés.

`/etc/mqm/pki/trust`
Contient des sous-répertoires contenant des certificats publics X.509 sous forme ASN.1 DER. Chaque sous-répertoire sera analysé à la recherche de fichiers `.crt` et ajouté en tant que certificats de confiance. Les certificats des clés privées sont automatiquement approuvés et n'ont pas besoin d'être ajoutés de cette manière.

`/etc/mqm/ha/pki/keys`
Contient des sous-répertoires avec des clés privées sous forme PKCS #1, ASN.1 DER ou PKCS #8, ASN.1 DER non chiffrées. Chaque sous-répertoire situé sous `/etc/mqm/ha/pki/keys` est analysé à la recherche de fichiers portant l'extension `.key`. La chaîne de certificats X.509 publics doit se trouver dans le même répertoire, dans un ou plusieurs fichiers portant l'extension " `.crt`, au format ASN.1 DER. Si les clés sont montées de cette manière, un dépôt de clés est automatiquement généré, ainsi qu'un fragment de fichier INI permettant d'utiliser ce dépôt de clés. La clé utilisée sera le premier nom après le tri lexicographique de la liste des noms de clés. Le référentiel de clés généré est écrit dans " `/run/runmqserver/ha/tls`.

`/etc/mqm/metrics/pki/keys`
Depuis 9.4.2.0-r1, si les clés TLS sont fournies dans ce répertoire et que les métriques sont activées pour un gestionnaire de file d'attente, un serveur d' HTTPS s est démarré sur le port 9157 de `/metrics` . Les fichiers de ce répertoire doivent être des certificats encodés en PEM :

- `tls.crt` certificat public du serveur
- `tls.key` Clé privée de serveur
- `ca.crt` Certificat public CA (facultatif)

Si aucune clé TLS n'est fournie ou si des versions antérieures d' IBM MQ sont utilisées, un serveur d' HTTP s est lancé.

/etc/mqm/web
L'arborescence des répertoires sous `/etc/mqm/web` est copiée au-dessus de `/var/mqm/web` lorsque le conteneur démarre. Par exemple, vous pouvez créer `/etc/mqm/web/installations/Installation1/servers/mqweb/mqwebuser.xml` pour configurer le serveur web. Ce fichier sera copié au bon endroit sous le " `/var/mqm/web`, qui est généralement un volume monté. Voir Configuration des codes 'IBM MQ Console et 'REST API.

/mnt/mqm
Le répertoire `/var/mqm` est symboliquement lié à cet emplacement. En montant un volume persistant à cet emplacement, vous pouvez conserver les données MQ lors de l'exécution des conteneurs.

/mnt/mqm-log
Répertoire utilisé pour contenir les fichiers journaux de récupération du gestionnaire de files d'attente. Si un volume est monté à cet emplacement, le code du conteneur définit l'option **crtmqm -ld**. Voir crtmqm (create queue manager) pour plus d'informations.

/mnt/mqm-data
Répertoire utilisé pour contenir les fichiers de données du gestionnaire de file d'attente. Si un volume est monté à cet emplacement, le code du conteneur définit l'option **crtmqm -md**. Voir crtmqm (create queue manager) pour plus d'informations.

/run
Le conteneur écrit des fichiers temporaires dans ce répertoire. Si vous souhaitez utiliser un système de fichiers en lecture seule pour le conteneur, vous devez monter un volume à cet emplacement et dans `/tmp`

/run/termination-log
Si le code du conteneur rencontre une erreur qui entraînera l'arrêt du gestionnaire de file d'attente, il en écrira la raison dans ce fichier. Cet emplacement de fichier est utilisé par défaut dans Kubernetes pour récupérer les informations de terminaison.

/tmp
Le conteneur écrit des fichiers temporaires dans ce répertoire. Si vous souhaitez utiliser un système de fichiers en lecture seule pour le conteneur, vous devez monter un volume à cet emplacement et dans `/run`

Linux IBM MQ image de conteneur sous licence

Une image de conteneur préconstruite est disponible pour IBM MQ. Cette image est disponible sur le site IBM Container Registry.

Utilisation

Pour pouvoir utiliser l'image, vous devez accepter les termes de la licence IBM MQ en définissant la variable d'environnement **LICENSE** . Pour plus d'informations, voir «Configurer les gestionnaires de files d'attente avec les annotations de licence IBM MQ en utilisant la fonction IBM MQ Operator», à la page 166.

Variables d'environnement prises en charge

Les variables d'environnement prises en charge par les images de conteneurs sous licence IBM MQ sont les mêmes que celles prises en charge par l'image de conteneur IBM MQ Advanced . Pour une liste complète, voir Variables d'environnement supportées par l'image du conteneur IBM MQ Advanced.

Remarque : A partir de IBM MQ 9.4.2, IBM MQ les gestionnaires de file d'attente sous licence peuvent être configurés pour utiliser les fonctions Native HA et Cross-Region Replication en ajoutant des licences supplémentaires à un gestionnaire de file d'attente. Pour plus d'informations, voir «Configurer Native HA et Cross-Region Replication sur IBM MQ Queue Managers sous licence en utilisant la fonction IBM MQ Operator», à la page 119. Les fonctions HA natives ne peuvent pas être utilisées sur les gestionnaires de file d'attente antérieurs à IBM MQ 9.4.2 qui utilisent une licence IBM MQ .

Système de fichiers et points de montage

Le système de fichiers et les points de montage pris en charge par les images de conteneurs sous licence IBM MQ sont les mêmes que ceux pris en charge par l'image de conteneur IBM MQ Advanced . Pour une liste complète, voir Système de fichiers et points de montage dans l'image du conteneur IBM MQ Advanced.

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, Kubernetes et d'autres environnements de conteneurs.

Images disponibles

Les images IBM MQ sont stockées dans IBM Container Registry:

- IBM MQ Advanced for Developers 9.4.0.11: `icr.io/ibm-messaging/mq:9.4.0.11-r3`
- IBM MQ Advanced for Developers 9.4.3.0: `icr.io/ibm-messaging/mq:9.4.3.0-r1`

Aide-mémoire

- Licence :
  - «mq.ibm.com/v1beta1: Versions actuelles de la licence», à la page 256 et Apache Licence 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

Toutes les variables d'environnement de l'article «IBM MQ Advanced image de conteneur», à la page 296 sont disponibles, ainsi que les variables supplémentaires suivantes :

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 YAML de cette rubrique montre comment vous pouvez fournir le mot de passe de l'administrateur en montant 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 YAML de cette rubrique montre comment vous pouvez fournir le mot de passe de l'application en montant un secret...

MQ_DEV

Définissez `false` pour arrêter la création des objets par défaut.

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=mqAdminPassword=passw0rd --from-
literal=mqAppPassword=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
  annotations: com.ibm.mq/write-defaults
  spec: 'false'
spec:
  license:
    accept: true
    license: L-HYGL-6STWD6
    use: Development
  web:
    enabled: true
  queueManager:
    env:
      - name: MQ_DEV
        value: "true"
      - name: MQ_CONNAUTH_USE_HTP
        value: "true"
    files:
      - secret:
          items:
            - item: mqAdminPassword
            - item: mqAppPassword
      name: my-mq-dev-passwords
      defaultMountPath: /run/secrets
    storage:
      queueManager:
        type: persistent-claim
    name: QUICKSTART
  version: 9.4.3.0-r1
```

IBM MQ MFT Agent for Developers image de conteneur

Une image de conteneur préconstruite est disponible pour IBM MQ MFT Agent for Developers. Cette image est disponible sur le site IBM Container Registry. Cette image peut être utilisée avec Docker, Podman, Kubernetes, et d'autres environnements de conteneurs.

Avertissement : Deprecated IBM MQ MFT Agent for Developers étaient auparavant disponibles sur le site Docker Hub, mais cette méthode est obsolète et aucune mise à jour n'est disponible sur le site Docker Hub.

Images disponibles

IBM MQ MFT Agent for Developers les images sont stockées sur le site IBM Container Registry:

- IBM MQ 9.3.0.0: `icr.io/ibm-messaging/mqmft:latest`

Aide-mémoire

- Licence :
  - IBM MQ Advanced for Developers et Apache Licence 2.0. La licence IBM MQ Advanced for Developers ne permet pas de distribution ultérieure, et les conditions limitent l'utilisation à une machine de développement.
- Où déposer les questions :
  - GitHub
- Disponible pour les architectures CPU suivantes :
  - amd64

Génération d'une image de conteneur

Voir la documentation d'utilisation pour construire votre propre image de conteneur d'agent avec le paquetage redistribuable MQ Managed File Transfer sur les architectures amd64..

Utilisation

Voir la documentation d'utilisation pour plus de détails sur la façon d'exécuter le conteneur d'images avec Podman runtime.

Voir la documentation d'utilisation pour plus de détails sur la manière de déployer l'image dans un site OpenShift Container Platform.

Pour pouvoir utiliser l'image, vous devez accepter les termes de la licence IBM MQ en définissant la variable d'environnement **`LICENSE`** .

Variables d'environnement

Licence

Sélectionnez `accepter` pour accepter les conditions de la licence IBM MQ Advanced for Developers .

Définir la `vue` pour afficher les conditions de licence.

MFT_AGENT_CONFIG_FILE

Obligatoire.

Chemin du fichier json contenant les informations nécessaires à la mise en place d'un agent. Le chemin d'accès doit se trouver sur un point de montage. Par exemple, un `configMap` sur OpenShift. Pour une description détaillée des attributs, voir les fichiers de configuration supplémentaires de l'agent MFT .

MFT_AGENT_NAME

Spécifiez le mot de passe de l'utilisateur admin.

Nom de l'agent à configurer.

BFG_JVM_PROPERTIES
Facultatif.

Toute propriété de la JVM qui doit être définie lors de l'exécution de la JVM de l'agent.

MFT_LOG_LEVEL
Facultatif.

Niveau d'information affiché. `info` et `verbose` sont les valeurs prises en charge, `info` étant la valeur *par défaut*. Le contenu de l'agent `output0.log` est affiché si MFT_LOG_LEVEL est réglé sur `verbose`.

MFT_AGENT_START_WAIT_TIME
Facultatif.

Le démarrage d'un agent peut prendre un certain temps après l'émission de la commande **`fteStartAgent`** . C'est le temps, en secondes, pendant lequel le conteneur attendra le démarrage d'un agent. Si un agent ne respecte pas le temps d'attente spécifié, le conteneur se termine.

MFT_MOUNT_PATH
Facultatif.

Variable d'environnement indiquant le chemin à partir duquel l'agent lira les fichiers ou les écrira.

MFT_COORD_QMGR_CIPHER
Le nom de l'adresse CipherSpec à utiliser pour se connecter en toute sécurité au gestionnaire de file d'attente de coordination.

MFT_CMD_QMGR_CIPHER
Le nom de l'adresse CipherSpec à utiliser pour se connecter de manière sécurisée au gestionnaire de file d'attente de commandes.

MFT_AGENT_QMGR_CIPHER
Le nom de l'adresse CipherSpec à utiliser pour se connecter en toute sécurité au gestionnaire de file d'attente de l'agent.

Pour plus d'informations sur la configuration par défaut du développeur prise en charge par l'image IBM MQ Advanced for Developers , voir la documentation sur la configuration par défaut du développeur.

Emplacement des fichiers de configuration de l'agent

L'agent dans le conteneur créera la configuration de l'agent et les fichiers journaux dans le répertoire fixe `/mnt/mftdata`. Ce dossier peut également se trouver sur un volume persistant, auquel cas le volume doit être monté en tant que point de montage `/mnt/mftdata` dans le conteneur.

OpenShift CP4I IBM MQ CASE pour l'installation et la mise à jour de l'air-gap sur 'Red Hat OpenShift

Lors de l'installation des gestionnaires de files d'attente " IBM MQ Operator et " IBM MQ dans un cluster " Red Hat OpenShift " à air comprimé, IBM MQ CASE est utilisé. L'IBM MQ CASE contient les détails de toutes les images et métadonnées qui composent une version particulière de l'IBM MQ CASE.

Air-gap et IBM MQ CASE 3.5.0 modifications

À partir de IBM MQ CASE 3.5.0, le site IBM MQ CASE contient des images pour une seule version du gestionnaire de file d'attente. Il s'agit de la dernière version du gestionnaire de files d'attente que le site IBM MQ Operator correspondant peut déployer. Avant IBM MQ CASE 3.5.0, le site CASE contenait toutes les versions du gestionnaire de file d'attente que l'opérateur correspondant pouvait déployer. Voir l' IBM MQ CASE Lookup Tables pour la version exacte disponible dans chaque IBM MQ CASE.

Vous pouvez utiliser un IBM MQ CASE pour installer ou mettre à niveau le dernier gestionnaire de file d'attente disponible que l'opérateur peut déployer, c'est-à-dire le gestionnaire de file d'attente CD ou SC2 publié en même temps que le CASE et l'opérateur correspondant. Dans ce cas, le processus d'installation ou de mise à niveau reste le même, mais il est plus rapide, car il doit refléter moins d'images.

Cependant, si vous installez une ou plusieurs versions de gestionnaire de files d'attente qui ne sont pas les plus récentes que l'opérateur peut déployer, vous devez répéter le processus de mise en miroir de l'image de l'air gap en utilisant un IBM MQ CASE distinct pour chaque version de gestionnaire de files d'attente requise. Voir l' IBM MQ CASE Lookup Tables pour la version exacte disponible dans chaque IBM MQ CASE.

Remarque : IBM MQ Operator , qui connaît toujours les anciennes versions du gestionnaire de files d'attente, peut donc toujours être sollicité pour déployer ces anciennes versions. C'est pourquoi il est important de déterminer les versions du gestionnaire de files d'attente que vous devez mettre en miroir.

Images relatives au gestionnaire de file d'attente

À partir de IBM MQ CASE 3.5.0, le site IBM MQ CASE contient uniquement les images de la dernière version du gestionnaire de files d'attente que l'opérateur correspondant peut déployer. Auparavant, le site CASE contenait toutes les versions du gestionnaire de file d'attente que l'opérateur correspondant pouvait déployer.

Remarque : Depuis 3.5.0, IBM MQ CASE n'inclut plus les images du tableau de bord des opérations, car les dernières versions du gestionnaire de file d'attente ne permettent pas de déployer le tableau de bord des opérations.

Utilisez les tables de recherche IBM MQ CASE pour identifier quelle version de IBM MQ CASE contient la version de gestionnaire de file d'attente IBM MQ dont vous avez besoin.

Images relatives à l'opérateur

Utilisez les tables de recherche IBM MQ CASE pour identifier quelle IBM MQ CASE contient la version IBM MQ Operator dont vous avez besoin.

Important :

- Lorsque vous installez le site IBM MQ Operator dans un environnement à espace aérien, vous devez toujours terminer le processus de mise en miroir de l'espace aérien pour le site IBM MQ CASE qui correspond à la version IBM MQ Operator que vous avez l'intention d'installer. N'installez pas, ni ne mettez à niveau, une version opérateur provenant d'une autre IBM MQ CASE.
- Le site IBM MQ CASE contient des versions supplémentaires de IBM MQ Operator, mais n'installez que l'opérateur correspondant à IBM MQ CASE que vous avez mis en miroir.
- Lors de la mise à jour du site IBM MQ Operator, si vous changez de canal d'opérateur, suivez la meilleure pratique de mise à jour de l'espace aérien pour passer à la dernière version de IBM MQ Operator disponible sur le canal actuel. Il est important de le faire avant de changer d'abonnement pour une nouvelle chaîne IBM MQ Operator.

IBM MQ CASE Tableaux de consultation

Tableau 3. Contenu des instances récentes du CD IBM MQ CASE

IBM MQ CASE	IBM MQ Opérateur	Gestionnaire de file d'attente IBM MQ
3.6.0	3.6.0	9.4.3.0-r1
3.5.3	3.5.3	9.4.2.1-r2
3.5.2	3.5.2	9.4.2.1-r1
3.5.1	3.5.1	9.4.2.0-r2
3.5.0	3.5.0	9.4.2.0-r1

Tableau 3. Contenu des instances récentes du CD IBM MQ CASE (suite)		
IBM MQ CASE	**IBM MQ Opérateur**	**Gestionnaire de file d'attente IBM MQ**
3.4.1	3.4.1	9.4.1.1-r1, 9.4.1.0-r2, 9.4.1.0-r1, 9.4.0.7-r1, 9.4.0.6-r2, 9.4.0.6-r1, 9.4.0.5-r2, 9.4.0.5-r1, 9.4.0.0-r3, 9.4.0.0-r2, 9.4.0.0-r1, 9.3.5.1-r2, 9.3.5.1-r1, 9.3.5.0-r2, 9.3.5.0-r1, 9.3.4.1-r1, 9.3.4.0-r1, 9.3.3.3-r2, 9.3.3.3-r1, 9.3.3.2-r3, 9.3.3.2-r2, 9.3.3.2-r1, 9.3.3.1-r2, 9.3.3.1-r1, 9.3.3.0-r2, 9.3.3.0-r1, 9.3.2.1-r2, 9.3.2.1-r1, 9.3.2.0-r2, 9.3.2.0-r1, 9.3.1.1-r1, 9.3.1.0-r3, 9.3.1.0-r2, 9.3.1.0-r1, 9.3.0.25-r1, 9.3.0.21-r3, 9.3.0.21-r2, 9.3.0.21-r1, 9.3.0.20-r2, 9.3.0.20-r1, 9.3.0.17-r3, 9.3.0.17-r2, 9.3.0.17-r1, 9.3.0.16-r2, 9.3.0.16-r1, 9.3.0.15-r1, 9.3.0.11-r2, 9.3.0.11-r1, 9.3.0.10-r2, 9.3.0.10-r1, 9.3.0.6-r1, 9.3.0.5-r3, 9.3.0.5-r2, 9.3.0.5-r1, 9.3.0.4-r2, 9.3.0.4-r1, 9.3.0.3-r1, 9.3.0.1-r4, 9.3.0.1-r3, 9.3.0.1-r2, 9.3.0.1-r1, 9.3.0.0-r3, 9.3.0.0-r2, 9.3.0.0-r1

Tableau 4. Contenu des récentes SC2 instances de IBM MQ CASE		
IBM MQ CASE	**IBM MQ Opérateur**	**Gestionnaire de file d'attente IBM MQ**
3.2.13	3.2.13	9.4.0.11-r3, 9.4.0.11-r2, 9.4.0.11-r1, 9.4.0.10-r2, 9.4.0.10-r1, 9.4.0.7-r1, 9.4.0.6-r2, 9.4.0.6-r1, 9.4.0.5-r2, 9.4.0.5-r1, 9.4.0.0-r3, 9.4.0.0-r2, 9.4.0.0-r1, 9.3.5.1-r2, 9.3.5.1-r1, 9.3.5.0-r2, 9.3.5.0-r1, 9.3.4.1-r1, 9.3.4.0-r1, 9.3.3.3-r2, 9.3.3.3-r1, 9.3.3.2-r3, 9.3.3.2-r2, 9.3.3.2-r1, 9.3.3.1-r2, 9.3.3.1-r1, 9.3.3.0-r2, 9.3.3.0-r1, 9.3.2.1-r2, 9.3.2.1-r1, 9.3.2.0-r2, 9.3.2.0-r1, 9.3.1.1-r1, 9.3.1.0-r3, 9.3.1.0-r2, 9.3.1.0-r1, 9.3.0.25-r1, 9.3.0.21-r3, 9.3.0.21-r2, 9.3.0.21-r1, 9.3.0.20-r2, 9.3.0.20-r1, 9.3.0.17-r3, 9.3.0.17-r2, 9.3.0.17-r1, 9.3.0.16-r2, 9.3.0.16-r1, 9.3.0.15-r1, 9.3.0.11-r2, 9.3.0.11-r1, 9.3.0.10-r2, 9.3.0.10-r1, 9.3.0.6-r1, 9.3.0.5-r3, 9.3.0.5-r2, 9.3.0.5-r1, 9.3.0.4-r2, 9.3.0.4-r1, 9.3.0.3-r1, 9.3.0.1-r4, 9.3.0.1-r3, 9.3.0.1-r2, 9.3.0.1-r1, 9.3.0.0-r3, 9.3.0.0-r2, 9.3.0.0-r1

Suppression de l'opérateur précédent

Pour référence, à partir de IBM MQ CASE 3.4.0, les images d' IBM MQ Operator s pour les chaînes suivantes ont été supprimées de IBM MQ CASE :

- v1.1
- v1.2
- v1.4
- v1.5
- v1.6
- v1.7
- v2.1

Recréation de grappes et de registres d'images

Vous pouvez utiliser la commande suivante pour lister tous les gestionnaires de file d'attente déployés dans un cluster d' Red Hat OpenShift , ainsi que leurs versions.

```
oc get queuemanager --all-namespaces -o=jsonpath='{range .items[*]}{.metadata.name}{":\t"}
{.spec.version}{"\n"}{end}'
```

Vous devez exécuter cette commande périodiquement et enregistrer le résultat. Cette liste est utile si vous souhaitez recréer le cluster d' Red Hat OpenShift , ou le registre d'images stockant les images en miroir pour l'air gap.

Tâches associées

«Mise à niveau d'un canal " CD " IBM MQ Operator vers le canal " CD " actuel», à la page 78
Mettre à niveau un déploiement " CD du " IBM MQ Operator où **seules les** licences " IBM MQ sont utilisées.

«Mise à niveau d'un " SC2 " IBM MQ Operator 3.2.x vers le canal " CD " actuel», à la page 77
Mettre à jour un déploiement 3.2.x 'SC2 du 'IBM MQ Operator où **seules les** licences 'IBM MQ sont utilisées.

«Installation d'IBM MQ Operator», à la page 59
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).

«Installation du IBM MQ Operator pour une utilisation avec CP4I», à la page 67
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).

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

© Copyright IBM Corp. 2007, 2025

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: