

9.1

IBM MQ nei contenitori

IBM

Nota

Prima di utilizzare queste informazioni e il prodotto che supportano, leggere le informazioni in [“Informazioni particolari” a pagina 53.](#)

Questa edizione si applica alla versione 9 release 1 di IBM® MQ e a tutte le successive release e modifiche se non diversamente indicato nelle nuove edizioni.

Quando si inviano informazioni a IBM, si concede a IBM un diritto non esclusivo di utilizzare o distribuire le informazioni in qualsiasi modo ritenga appropriato senza incorrere in alcun obbligo verso l'utente.

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

Indice

IBM MQ nei contenitori.....	5
Pianificazione per IBM MQ nei contenitori.....	5
Scelta della modalità di utilizzo di IBM MQ nei contenitori.....	5
Supporto per i contenitori certificati IBM MQ.....	6
Supporto per la creazione di grafici e immagini del contenitore IBM MQ.....	8
Considerazioni sulla memoria per IBM MQ Advanced certified container.....	9
Alta disponibilità per IBM MQ Advanced certified container.....	10
Autenticazione utente e autorizzazione per IBM MQ Advanced certified container.....	12
Installazione e disinstallazione dell'operatore IBM MQ su OpenShift.....	12
Installazione dell'operatore IBM MQ utilizzando la Console web di OpenShift.....	12
Installazione dell'operatore IBM MQ utilizzando la CLI OpenShift.....	14
Distribuzione di contenitori certificati IBM MQ.....	16
Preparazione del tuo progetto OpenShift per IBM MQ utilizzando la CLI di OpenShift.....	16
Distribuzione di un gestore code utilizzando IBM Cloud Pak for Integration Platform Navigator.....	17
Distribuzione di un gestore code utilizzando la Console web OpenShift.....	18
Distribuzione di un gestore code utilizzando la CLI OpenShift.....	19
Integrazione con IBM Cloud Pak for Integration Operations Dashboard.....	20
Creazione di un'immagine con file MQSC e INI personalizzati, utilizzando la CLI OpenShift.....	21
Distribuzione di contenitori certificati IBM MQ utilizzando Helm.....	23
Distribuzione di release CD precedenti di IBM MQ in un cluster IBM Cloud Private.....	27
Aggiunta di release CD precedenti di un'immagine di IBM MQ in un cluster IBM Cloud Private.....	29
Aggiunta di release CD precedenti di un'immagine di IBM MQ in un cluster IBM Cloud Kubernetes Service.....	29
Connessione a un gestore code distribuito in un cluster OpenShift.....	30
Connessione al IBM MQ Console distribuito in un cluster OpenShift.....	32
Backup e ripristino della configurazione del gestore code utilizzando la CLI OpenShift.....	33
Creazione del tuo proprio contenitore IBM MQ.....	34
Pianificazione della tua immagine del gestore code IBM MQ utilizzando un contenitore.....	34
Creazione di un'immagine del gestore code IBM MQ di esempio utilizzando Docker.....	35
Esecuzione di applicazioni di bind locali in contenitori separati.....	37
Riferimento API per l'operatore IBM MQ.....	39
Riferimento API per mq.ibm.com/v1beta1.....	40
Informazioni particolari.....	53
Informazioni sull'interfaccia di programmazione.....	54
Marchi.....	54

I contenitori ti consentono di impacchettare un gestore code IBM MQ o un'applicazione client IBM MQ, con tutte le sue dipendenze, in un'unità standardizzata per lo sviluppo software.

Puoi eseguire IBM MQ nel contenitore preconfezionato fornito in IBM MQ Advanced e IBM MQ Advanced for Developers. Questo IBM MQ Advanced certified container offre un'immagine supportata e un grafico Helm e può essere utilizzato per distribuire un'immagine IBM MQ pronta per la produzione in Red Hat® OpenShift®, IBM Cloud Private o IBM Cloud Kubernetes Service.

Puoi anche eseguire IBM MQ in un contenitore IBM Cloud Pak for Integration o in un contenitore che crei tu stesso.

  Per ulteriori informazioni su IBM MQ Advanced certified container, consultare i seguenti link.

Quando si pianifica IBM MQ nei contenitori, considerare il supporto fornito da IBM MQ per varie opzioni architetturali, ad esempio come viene gestita l'alta disponibilità e come proteggere i gestori code.

Informazioni su questa attività

Prima di pianificare il tuo IBM MQ nell'architettura dei contenitori, dovresti familiarizzare con i concetti di IBM MQ di base (vedi [Panoramica tecnica di IBM MQ](#)) e con i concetti di base di Kubernetes/OpenShift (vedi [Architettura di OpenShift Container Platform](#)).

Procedura

- [“Scelta della modalità di utilizzo di IBM MQ nei contenitori”](#) a pagina 5.
- [“Alta disponibilità per IBM MQ Advanced certified container”](#) a pagina 10.
- [“Autenticazione utente e autorizzazione per IBM MQ Advanced certified container”](#) a pagina 12.

Scelta della modalità di utilizzo di IBM MQ nei contenitori

Esistono più opzioni per l'utilizzo di IBM MQ nei contenitori: puoi scegliere di utilizzare contenitori certificati preconfezionati oppure puoi creare le tue immagini e il tuo codice di distribuzione.

Utilizzo dei contenitori certificati IBM MQ Advanced

Se stai pianificando di eseguire la distribuzione su Red Hat OpenShift Container Platform, probabilmente vuoi utilizzare i contenitori certificati. Ci sono tre varietà di contenitore certificato:

- IBM MQ Advanced certified container per IBM Cloud Pak for Integration. Si tratta di un prodotto IBM separato che include una versione di un contenitore certificato.
- IBM MQ Advanced certified container
- Contenitore certificato IBM MQ Advanced for Developers (non garantito)

Le release IBM MQ 9.1.4 e precedenti CD erano supportate anche su IBM Cloud Private e IBM Cloud Kubernetes Service.

Si noti che i contenitori certificati si stanno evolvendo rapidamente e sono pertanto supportati solo nelle release [Continuous Delivery](#).

I contenitori certificati includono sia le immagini del contenitore preintegrate, sia il codice di distribuzione per l'esecuzione su Red Hat OpenShift Container Platform. Da IBM MQ 9.1.5 in poi, i gestori code vengono gestiti utilizzando un operatore IBM MQ. Le versioni precedenti di IBM MQ, fino alla versione 9.1.5 inclusa, sono gestite utilizzando i grafici Helm.

Alcune funzioni IBM MQ non sono supportate quando si utilizzano i contenitori certificati. Sarà necessario creare le proprie immagini e grafici se si desidera effettuare una delle seguenti operazioni:

- Utilizzare le API REST per la gestione o la messaggistica
- Utilizzare uno dei seguenti componenti di MQ :
 - Managed File Transfer Agent e relative risorse. Tuttavia, è possibile utilizzare i contenitori certificati per fornire uno o più gestori code Coordinamento, Comando o Agent.
 - AMQP
 - IBM MQ Bridge to Salesforce
 - IBM MQ Bridge to blockchain (non supportato in contenitori)
- Utilizza il server web quando stai eseguendo la distribuzione utilizzando i grafici Helm (tranne IBM Cloud Pak for Integration)
- Personalizzare le opzioni utilizzate con `crtmqm`, `strmqm` e `endmqm`, come la configurazione dei log di ripristino

Creazione di immagini e grafici personalizzati

Questa è la soluzione del contenitore più flessibile, ma ti richiede di avere forti capacità nella configurazione dei contenitori e di "possedere" il contenitore risultante. Se non hai intenzione di utilizzare Red Hat OpenShift Container Platform, dovrai creare le tue immagini e il tuo codice di distribuzione.

Sono disponibili esempi per la creazione di immagini personalizzate. Consultare [“Creazione del tuo proprio contenitore IBM MQ”](#) a pagina 34. I grafici Helm forniti come parte dei contenitori certificati sono pubblicati su GitHub e possono essere utilizzati come esempi per quando stai creando le tue immagini:

- [Helm grafico per IBM MQ Advanced certified container](#)
- [Grafico Helm per contenitore certificato IBM MQ Advanced for Developers](#)

Concetti correlati

[“Supporto per i contenitori certificati IBM MQ”](#) a pagina 6

I contenitori certificati IBM MQ sono supportati solo in alcuni ambienti Kubernetes

[“Supporto per la creazione di grafici e immagini del contenitore IBM MQ”](#) a pagina 8

Informazioni da considerare se si utilizzano contenitori su un sistema Linux.

Linux

Supporto per i contenitori certificati IBM MQ

I contenitori certificati IBM MQ sono supportati solo in alcuni ambienti Kubernetes

 Per la release CD V9.1.4 e successive, IBM MQ Advanced certified container è supportato per l'utilizzo con Red Hat OpenShift. Consultare [“Distribuzione di un gestore code utilizzando la CLI Helm”](#) a pagina 24.

Le release CD precedenti a V9.1.4 erano supportate nei seguenti ambienti Kubernetes :

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

Per specifiche versioni supportate di Kubernetes, consulta i file `qualification.yaml` e `Chart.yaml` all'interno di un grafico IBM MQ Advanced Helm scaricato. Queste versioni variano da release a release.

IBM MQ Advanced certified container è supportato solo quando viene distribuito utilizzando l'operatore IBM MQ o quando si utilizza uno dei seguenti grafici Helm :

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

Nota: L'utilizzo dei diagrammi Helm è obsoleto, in seguito alla release dell'operatore IBM MQ .

Poiché la tecnologia del contenitore si sta evolvendo rapidamente, IBM MQ Advanced certified container è supportata solo sulla versione più recente delle piattaforme supportate da questo grafico al momento della release. Se si desidera utilizzare una versione della piattaforma precedente, potrebbe essere necessario utilizzare una versione precedente di IBM MQ Advanced certified container.

L'immagine IBM MQ Advanced certified container è basata sulle release IBM MQ Continuous Delivery (CD). Questi sono supportati per un massimo di un anno, o per due versioni di CD, a seconda di quale è più lungo. Le release Long Term Support di IBM MQ non sono disponibili come contenitore certificato.

Da IBM MQ Advanced certified container V4.0 in poi, l'immagine fornisce un'installazione di IBM MQ su un Red Hat Universal Base Image (UBI), che include le librerie e i programmi di utilità Linux chiave utilizzati da IBM MQ. L'UBI è supportato da Red Hat quando viene eseguito su un host Red Hat Enterprise Linux . Le versioni precedenti di IBM MQ Advanced certified container utilizzavano un'immagine base Ubuntu non supportata.

Concetti correlati

“Supporto per la creazione di grafici e immagini del contenitore IBM MQ” a pagina 8
 Informazioni da considerare se si utilizzano contenitori su un sistema Linux .

Linux → MQ Adv. → CD Supporto versione per IBM MQ Advanced certified container

Una serie di tabelle che mostra l'associazione tra le versioni supportate di IBM MQ Advanced certified container, IBM MQ, IBM Cloud Kubernetes Service, IBM Cloud Pak for Integration IBM Cloud Private.

IBM MQ Operatore

→ V 9.1.5

L'operatore IBM MQ è supportato per l'utilizzo come parte di IBM Cloud Pak for Integration versione 2020.2, o indipendentemente, con IBM MQ versione 9.1.5 e versioni successive.

L'operatore IBM MQ è supportato su Red Hat OpenShift Container Platform versione 4.4 o superiore.

IBM MQ Advanced certified container → V 9.1.5 (graficoHelm) - obsoleto

Includi il grafico Helm `ibm-mqadvanced-server-prod`.

→ V 9.1.5 Da IBM MQ Advanced certified container V5.0.x in poi, il grafico, l'immagine e le correzioni Helm vengono forniti tramite il catalogo e il registro di IBM Entitled. Le versioni precedenti sono state fornite tramite Passport Advantage e le release di fix sono disponibili da IBM Fix Central.

Tabella 1. Supporto per IBM MQ Advanced certified container			
Versione	IBM MQ Versione	Fine del supporto	Piattaforme supportate
6.0.x	9.1.5 Release di fornitura continua	Marzo 2021	Requisiti di sistema dettagliati
5.0.x	9.1.4 Release di fornitura continua	Dicembre 2020	Requisiti di sistema dettagliati

Tabella 1. Supporto per IBM MQ Advanced certified container (Continua)

Versione	IBM MQ Versione	Fine del supporto	Piattaforme supportate
4.1.x	9.1.3 Release di fornitura continua	Luglio 2020	Requisiti di sistema dettagliati

Software IBM MQ Advanced certified container per IBM Cloud Pak for Integration

V 9.1.5 (graficoHelm) - obsoleto

Includere il grafico Helm `ibm-mqadvanced-server-integration-prod`.

Tabella 2. Supporto versione per il software IBM MQ Advanced certified container per IBM Cloud Pak for Integration

Versione	IBM MQ Versione	IBM Cloud Pak for Integration Versione
6.0.x	9.1.4 Release di fornitura continua	2020.1.1 (Requisiti di sistema)
5.0.x	9.1.3 Release di fornitura continua	2019.4.1 (Requisiti di sistema)
4.1.x	9.1.3 Release di fornitura continua	2019.3.2.2 (Requisiti di sistema)
4.0.x	9.1.3 Release di fornitura continua	2019.3.2 (Requisiti di sistema)
3.0.x	9.1.3 Release di fornitura continua	2019.3.1 (Requisiti di sistema)

Consultare le [IBM Cloud Pak for Integration note sulla versione](#) per informazioni sulla versione supportata.

Linux

Supporto per la creazione di grafici e immagini del contenitore IBM

MQ

Informazioni da considerare se si utilizzano contenitori su un sistema Linux.

- L'immagine di base utilizzata dall'immagine contenitore deve utilizzare un sistema operativo Linux supportato.
- È necessario utilizzare i programmi di installazione IBM MQ per installare il prodotto all'interno dell'immagine del contenitore.
- Per un elenco dei pacchetti supportati, consultare [IBM MQ rpm components for Linux systems](#).
- **V 9.1.0** I seguenti pacchetti non sono supportati:
 - MQSeriesBCBridge
 - MQSeriesRDQM
- La directory dei dati del gestore code (`/var/mqm` per impostazione predefinita) deve essere memorizzata su un volume contenitore che mantiene lo stato persistente.

Importante: Non è possibile utilizzare il file system di unione.

È necessario montare una directory host come un volume di dati o utilizzare un contenitore di volumi di dati. Per ulteriori informazioni, vedi [Gestione dei dati nei contenitori](#).

- Devi essere in grado di eseguire i comandi di controllo IBM MQ, come `endmqm`, all'interno del contenitore.

- È necessario essere in grado di ottenere file e directory dall'interno del contenitore per scopi diagnostici.
- **V 9.1.0** È possibile utilizzare lo spazio dei nomi per condividere gli spazi dei nomi del contenitore per il gestore code con altri contenitori, in modo da collegare localmente le applicazioni a un gestore code in esecuzione in contenitori separati. Per ulteriori informazioni, consultare [“Esecuzione di applicazioni di bind locali in contenitori separati”](#) a pagina 37.

Concetti correlati

[“Supporto per i contenitori certificati IBM MQ”](#) a pagina 6

I contenitori certificati IBM MQ sono supportati solo in alcuni ambienti Kubernetes

V 9.1.5

Linux

MQ Adv.

CD

Considerazioni sulla memoria per IBM

MQ Advanced certified container

IBM MQ Advanced certified container viene eseguito in due modalità di memoria:

- L' **archiviazione temporanea** viene utilizzata quando è possibile eliminare tutto lo stato del contenitore al riavvio del contenitore. Viene comunemente utilizzato quando gli ambienti vengono creati per la dimostrazione o quando si sviluppano con gestori code autonomi.
- L' **archiviazione persistente** è la configurazione comune per IBM MQ e garantisce che se il contenitore viene riavviato, la configurazione esistente, i log e i messaggi persistenti sono disponibili nel contenitore riavviato.

L'operatore IBM MQ fornisce la capacità di personalizzare le caratteristiche di memoria che possono differire notevolmente a seconda dell'ambiente e della modalità di memoria desiderata.

Archiviazione effimera

IBM MQ è un'applicazione con stato e conserva questo stato nella memoria per il recupero in caso di riavvio. Se si utilizza la memoria temporanea, tutti gli stati del gestore code andranno persi al riavvio. Questo include:

- Tutti i messaggi
- Tutti i gestori code allo stato di comunicazione del gestore code (numeri di sequenza dei messaggi del canale)
- L'identità del cluster MQ del gestore code
- Stato di tutte le transazioni
- Configurazione di tutti i gestori code
- Tutti i dati diagnostici locali

Per questo motivo è necessario considerare se l'archiviazione effimera è un approccio adatto per uno scenario di produzione, test o sviluppo. Ad esempio, dove tutti i messaggi sono noti come non persistenti e il gestore code non è un membro di un cluster MQ . Oltre a disporre di tutto lo stato della messaggistica al riavvio, anche la configurazione del gestore code viene eliminata. Per abilitare un contenitore completamente effimero la configurazione IBM MQ deve essere aggiunta all'immagine del contenitore stessa (per ulteriori informazioni, consultare [“Creazione di un'immagine con file MQSC e INI personalizzati, utilizzando la CLI OpenShift”](#) a pagina 21). Se questo non viene completato, IBM MQ dovrà essere configurato ogni volta che il contenitore viene riavviato.

Ad esempio, per configurare IBM MQ con memoria effimera, il tipo di archiviazione di `QueueManager` deve includere quanto segue:

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

Archiviazione permanente

IBM MQ viene normalmente eseguito con la memoria persistente per garantire che il gestore code conservi i messaggi persistenti e la configurazione dopo un riavvio. Pertanto, questo è il comportamento predefinito. A causa dei vari provider di memoria e delle diverse funzionalità di ciascun supporto, ciò spesso significa che è richiesta la personalizzazione della configurazione. Di seguito vengono riportati i campi comuni che personalizzano la configurazione dell'archivio MQ nell'API v1beta1 :

- `spec.queueManager.availability` controlla la modalità di disponibilità. Se si utilizza `SingleInstance` , è necessaria solo l'archiviazione `ReadWriteOnce` , mentre `MultiInstance` richiede una classe di archiviazione che supporta `ReadWriteMany` con le caratteristiche di blocco file corrette. IBM MQ fornisce un' istruzione di supporto e un' istruzione di test. La modalità di disponibilità influenza anche il layout del volume persistente. Per ulteriori informazioni, consultare [“Alta disponibilità per IBM MQ Advanced certified container”](#) a pagina 10
- `spec.queueManager.storage` controlla le singole impostazioni di memoria. Un gestore code può essere configurato per utilizzare tra uno e quattro volumi persistenti

Il seguente esempio mostra un frammento di una configurazione semplice utilizzando un gestore code a istanza singola:

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

Il seguente esempio mostra un frammento di configurazione di un gestore code a più istanze, con una classe di memoria non predefinita e con l'archiviazione file che richiede gruppi supplementari:

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

Linux

MQ Adv.

CD

Alta disponibilità per IBM MQ Advanced certified container

Hai due scelte principali per l'alta disponibilità con IBM MQ Advanced certified container: **Gestore code a più istanze** (che è una coppia active - standby, che utilizza un file system di rete condiviso) e **Gestore code resiliente singolo** (che offre un approccio semplice per HA che utilizza la memoria di rete).

Dovresti considerare separatamente la disponibilità del **messaggio** e del **servizio** . Con IBM MQ for Multiplatforms, un messaggio viene memorizzato esattamente su un gestore code. Quindi, se il gestore code diventa non disponibile, si perde temporaneamente l'accesso ai messaggi in esso contenuti. Per ottenere un'elevata disponibilità di messaggi , è necessario essere in grado di ripristinare un gestore code il più rapidamente possibile. Puoi ottenere la disponibilità del servizio disponendo di più istanze di code per le applicazioni client da utilizzare, ad esempio utilizzando un cluster uniforme IBM MQ .

Un gestore code può essere pensato in due parti: i dati memorizzati sul disco e i processi in esecuzione che consentono l'accesso ai dati. Qualsiasi gestore code può essere spostato in un nodo Kubernetes diverso, purché conservi gli stessi dati (forniti dai volumi persistenti Kubernetes) ed è ancora indirizzabile nella rete dalle applicazioni client. In Kubernetes, un servizio è utilizzato per fornire un'identità di rete congruente.

IBM MQ si basa sulla disponibilità dei dati sui volumi persistenti. Pertanto, la disponibilità della memoria che fornisce i volumi persistenti è fondamentale per la disponibilità del gestore code, perché IBM MQ non può essere più disponibile della memoria che sta utilizzando. Se si desidera tollerare un'interruzione di un'intera zona di disponibilità, è necessario utilizzare un provider di volumi che replichi le scritture disco in un'altra zona.

gestore code a più istanze

I gestori code a più istanze coinvolgono un pod **attivo** e un pod **standby** Kubernetes , che vengono eseguiti come parte di un Kubernetes Stateful Set con esattamente due repliche e una serie di volumi persistenti Kubernetes . I dati e i log delle transazioni del gestore code sono conservati su due volumi permanenti, utilizzando un filesystem condiviso.

I gestori code a più istanze richiedono che i pod **attivi** e **standby** abbiano accesso simultaneo al volume persistente. Per configurare ciò, utilizzare Kubernetes Volumi persistenti con **access mode** impostato su `ReadWriteMany`. I volumi devono inoltre soddisfare i requisiti di IBM MQ [per i file system condivisi](#), poiché IBM MQ si basa sul rilascio automatico dei blocchi file per istigare un failover del gestore code. IBM MQ produce un [elenco di file system verificati](#).

I tempi di recupero per un gestore code a più istanze sono controllati dai seguenti fattori:

1. Il tempo impiegato dopo che si è verificato un malfunzionamento per il file system condiviso per rilasciare i blocchi originariamente presi dall'istanza attiva.
2. Il tempo impiegato dall'istanza standby per acquisire i blocchi e quindi avviarli.
3. Il tempo impiegato dal probe di disponibilità del pod Kubernetes per rilevare che il contenitore è pronto. È configurabile nel grafico Helm .
4. Il tempo impiegato dai client IBM MQ per riconnettersi.

Singolo gestore code resiliente

Un singolo gestore code resiliente è una singola istanza di un gestore code in esecuzione in un singolo pod Kubernetes , dove Kubernetes monitorizza il gestore code e sostituisce il pod come necessario.

I requisiti di IBM MQ [per i file system condivisi](#) si applicano anche quando si utilizza un singolo gestore code resiliente (ad eccezione del blocco basato sul lease), ma non è necessario utilizzare un file system condiviso. È possibile utilizzare l'archiviazione blocchi, con un file system adatto in cima. Ad esempio, `xfs` o `ext4`.

I tempi di recupero per un singolo gestore code resiliente sono controllati dai seguenti fattori:

1. Il tempo impiegato per l'esecuzione del probe di attività e il numero di errori tollerati. È configurabile nel grafico Helm .
2. Il tempo impiegato dallo scheduler Kubernetes per ripianificare il pod non riuscito su un nuovo nodo.
3. Quanto tempo ci vuole per scaricare l'immagine del contenitore sul nuovo Nodo. Se si utilizza un valore **imagePullPolicy** di `IfNotPresent`, l'immagine potrebbe essere già disponibile su tale Nodo.
4. Il tempo impiegato per l'avvio della nuova istanza del gestore code.
5. Il tempo impiegato dal probe di disponibilità del pod Kubernetes per rilevare che il contenitore è pronto. È configurabile nel grafico Helm .
6. Il tempo impiegato dai client IBM MQ per riconnettersi.

Importante:

Sebbene il singolo modello di gestore code resiliente offra alcuni vantaggi, è necessario comprendere se è possibile raggiungere i propri obiettivi di disponibilità con le limitazioni relative agli errori del nodo.

In Kubernetes, un pod malfunzionante viene generalmente ripristinato rapidamente, ma l'errore di un intero nodo viene gestito in modo diverso. Se il nodo master Kubernetes perde il contatto con un nodo di lavoro, non può determinare se il nodo ha avuto esito negativo o se ha semplicemente perso la

connettività di rete. Pertanto, Kubernetes non esegue **alcuna azione** in questo caso finché non si verifica uno dei seguenti eventi:

1. Il nodo viene ripristinato in uno stato in cui il nodo master Kubernetes può comunicare con esso.
2. Viene eseguita un'azione amministrativa per eliminare esplicitamente il pod sul nodo master Kubernetes . Ciò non arresta necessariamente l'esecuzione del pod, ma lo elimina semplicemente dall'archivio Kubernetes . Questa azione amministrativa deve quindi essere intrapresa con molta attenzione.

Concetti correlati

[Configurazioni HA \(High Availability\)](#)

Linux > MQ Adv. > CD Autenticazione utente e autorizzazione per IBM MQ Advanced certified container

IBM MQ può essere configurato per utilizzare utenti e gruppi LDAP per l'autorizzazione. Questo è l'approccio consigliato per IBM MQ Advanced certified container.

In un ambiente containerizzato a più tenant come Red Hat OpenShift Container Platform, vengono implementati dei vincoli di sicurezza per prevenire potenziali problemi di sicurezza. Ad esempio, in Red Hat OpenShift Container Platform il valore predefinito `SecurityContextConstraints` (denominato `restricted`) utilizza un ID utente casuale, sconsigliando gli utenti locali per il contenitore stesso. IBM MQ generalmente utilizza l'escalation dei privilegi per controllare le password degli utenti, che non è consigliata anche in ambienti contenitore a più tenant. Per questi motivi, l'utilizzo degli utenti definiti nelle librerie del sistema operativo all'interno di un contenitore in esecuzione non è supportato nei contenitori certificati IBM MQ .

È necessario configurare il gestore code per utilizzare LDAP per l'autenticazione e l'autorizzazione utente. Per informazioni sulla configurazione di IBM MQ a tale scopo, consultare [Autenticazione connessione: repository utente e autorizzazione LDAP](#)

V 9.1.5 > Linux > MQ Adv. > CD Installazione e disinstallazione dell'operatore IBM MQ su OpenShift

L'operatore IBM MQ può essere installato su OpenShift utilizzando l'hub operatore.

Prima di iniziare

Procedura

- [“Installazione dell'operatore IBM MQ utilizzando la CLI OpenShift”](#) a pagina 14.
- [“Installazione dell'operatore IBM MQ utilizzando la Console web di OpenShift”](#) a pagina 12.

V 9.1.5 > Linux > MQ Adv. > CD Installazione dell'operatore IBM MQ utilizzando la Console web di OpenShift

L'operatore IBM MQ può essere installato su OpenShift utilizzando l'hub operatore.

Prima di iniziare

Accedi alla console web del tuo cluster OpenShift .

Procedura

1. Aggiungere gli operatori IBM Common Services all'elenco di operatori installabili
 - a) Fai clic sull'icona Più. Viene visualizzata la finestra di dialogo **Importa YAM** .
 - b) Incollare la seguente definizione risorsa nella casella di dialogo.

```

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

```

- c) Fai clic su **Crea**.
2. Aggiungere gli operatori IBM all'elenco di operatori installabili
 - a) Fai clic sull'icona Più. Viene visualizzata la finestra di dialogo **Importa YAM** .
 - b) Incollare la seguente definizione risorsa nella casella di dialogo.

```

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

```

- c) Fai clic su **Crea**.
3. Creare uno spazio dei nomi da utilizzare per l'operatore IBM MQ

L'operatore IBM MQ può essere installato in un singolo spazio dei nomi o in tutti gli spazi dei nomi. Questa fase è necessaria solo se si desidera eseguire l'installazione in uno spazio dei nomi particolare che non esiste già.

 - a) Dal pannello di navigazione, fare clic su **Home > Progetti**.
Viene visualizzata la pagina Progetti.
 - b) Fare clic su **Crea progetto**. Viene visualizzata un'area Crea progetto.
 - c) Immetti i dettagli dello spazio dei nomi che stai creando. Ad esempio, è possibile specificare "ibm - mq" come nome.
 - d) Fai clic su **Crea**. Viene creato lo spazio dei nomi per il tuo operatore IBM MQ .
 4. Installare l'operatore IBM MQ .
 - a) Dal riquadro di navigazione, fare clic su **Operatori > OperatorHub**.
Viene visualizzata la pagina OperatorHub .
 - b) Nel campo **Tutti gli elementi** , immettere "IBM MQ".
Viene visualizzata la voce di catalogo IBM MQ .
 - c) Selezionare **IBM MQ**.
Viene visualizzata la finestra IBM MQ .
 - d) Fai clic su **Installa**.
Viene visualizzata la pagina Crea sottoscrizione operatore.
 - e) Impostare la modalità di installazione sullo spazio dei nomi specifico creato o sull'ambito a livello di cluster
 - f) Fai clic su **Sottoscrivi**.
Verrà visualizzato IBM MQ nella pagina Operatori installati.

- g) Controllare lo stato dell'operatore nella pagina Operatori installati; lo stato cambierà in Riuscito quando l'installazione è completa.

Operazioni successive

[“Distribuzione di contenitori certificati IBM MQ” a pagina 16](#)

V 9.1.5 Linux MQ Adv. CD **Installazione dell'operatore IBM MQ utilizzando la CLI OpenShift**

L'operatore IBM MQ può essere installato su OpenShift utilizzando l'hub operatore.

Prima di iniziare

Accedi alla CLI (command line interface) OpenShift utilizzando **oc login**. Per questa procedura, sarà necessario essere un amministratore del cluster.

Procedura

1. Creare un `OperatorSource` per gli operatori IBM Common Services

- a) Creare un file YAML che definisca la risorsa `OperatorSource`

Creare un file denominato "operator-source-cs.yaml" con il contenuto seguente:

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

- b) Applicare il `OperatorSource` al server.

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

2. Crea un `OperatorSource` per gli operatori IBM

- a) Creare un file YAML che definisca la risorsa `OperatorSource`

Creare un file denominato "operator-source-ibm.yaml" con il contenuto seguente:

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

- b) Applicare il `OperatorSource` al server.

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

3. Creare uno spazio dei nomi da utilizzare per l'operatore IBM MQ

L'operatore IBM MQ può essere installato in un singolo spazio dei nomi o in tutti gli spazi dei nomi. Questa fase è necessaria solo se si desidera eseguire l'installazione in uno spazio dei nomi particolare che non esiste già.

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

4. Visualizzare l'elenco di operatori disponibili per il cluster da OperatorHub

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

5. Esaminare l'operatore IBM MQ per verificarne le InstallModes supportate e i canali disponibili

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

6. Crea un file YAML dell'oggetto OperatorGroup

Una `OperatorGroup` è una risorsa OLM che seleziona gli spazi dei nomi di destinazione in cui generare l'accesso RBAC richiesto per tutti gli operatori nello stesso spazio dei nomi di `OperatorGroup`.

Lo spazio dei nomi a cui sottoscrivi l'operatore deve avere un `OperatorGroup` che corrisponda al `InstallMode` dell'operatore, la modalità `AllNamespaces` o `SingleNamespace`. Se l'operatore che si intende installare utilizza `AllNamespaces`, lo spazio dei nomi `openshift-operators` dispone già di un `OperatorGroup` appropriato.

Tuttavia, se l'operatore utilizza la modalità `SingleNamespace` e non si dispone già di un `OperatorGroup` appropriato, è necessario crearne uno.

a) Creare un file denominato "mq-operator-group.yaml" con il seguente contenuto:

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

b) Creare l'oggetto `OperatorGroup`

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

7. Creare un file YAML dell'oggetto sottoscrizione per sottoscrivere uno spazio dei nomi all'operatore MQ

a) Creare un file denominato "mq-sub.yaml" con il seguente contenuto:

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

Per l'uso `AllNamespaces` **InstallMode**, specifica lo spazio dei nomi `openshift-operators`. Altrimenti, specificare il singolo spazio dei nomi pertinente per l'utilizzo di `SingleNamespace` **InstallMode**.

b) Creare l'oggetto `Subscription`

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

8. Verificare lo stato dell'operatore

Una volta eseguita correttamente l'installazione dell'operatore, lo stato del pod viene visualizzato come *In esecuzione*. Per l'utilizzo `AllNamespaces` **InstallMode**, specificare **openshift-**

operators come spazio dei nomi. Altrimenti, specificare il singolo spazio dei nomi pertinente per l'utilizzo di `SingleNamespace` **InstallMode** .

Operazioni successive

[“Distribuzione di contenitori certificati IBM MQ” a pagina 16](#)

Linux

MQ Adv.

CD

Distribuzione di contenitori certificati IBM

MQ

IBM MQ versione 9.1.5 e successive possono essere distribuite a Red Hat OpenShift utilizzando l'operatore IBM MQ . Le IBM MQ versioni 9.1.5 e 9.1.4 possono essere distribuite a Red Hat OpenShift utilizzando Helm. Le versioni CD precedenti possono essere distribuite a un cluster IBM Cloud Private o a un cluster IBM Cloud Kubernetes Service , utilizzando Helm.

Informazioni su questa attività

Procedura

- [“Distribuzione di un gestore code utilizzando la CLI Helm” a pagina 24.](#)
- [“Distribuzione di release CD precedenti di IBM MQ in un cluster IBM Cloud Private” a pagina 27.](#)
- [“Aggiunta di release CD precedenti di un'immagine di IBM MQ in un cluster IBM Cloud Private” a pagina 29.](#)
- [“Aggiunta di release CD precedenti di un'immagine di IBM MQ in un cluster IBM Cloud Kubernetes Service” a pagina 29.](#)

Linux

MQ Adv.

CD

Preparazione del tuo progetto OpenShift per IBM

MQ utilizzando la CLI di OpenShift

Prepara il tuo cluster Red Hat OpenShift Container Platform in modo che sia pronto a distribuire un gestore code utilizzando l'operatore IBM MQ . Questa attività deve essere completata da un amministratore del progetto.

Prima di iniziare

Nota: Se si prevede di utilizzare IBM MQ in un progetto con altri componenti IBM Cloud Pak for Integration già installati, potrebbe non essere necessario seguire queste istruzioni.

Accedi al tuo cluster utilizzando **cloudctl login** (per IBM Cloud Pak for Integration) o **oc login**.

Informazioni su questa attività

Le immagini IBM MQ Advanced certified container vengono estratte da un registro del contenitore che esegue un controllo della titolarità della licenza. Questo controllo necessita di una chiave di titolarità memorizzata in un segreto di `pull docker-registry` . Se non si dispone ancora di una chiave di titolarità, seguire queste istruzioni per ottenere una chiave di titolarità e creare un segreto di pull.

Procedura

1. Ottieni la chiave di titolarità assegnata al tuo ID.
 - a) Accedere a [MyIBM Container Software Library](#) con l'ID e la password IBM associati al software autorizzato.
 - b) Nella sezione **Chiavi di titolarità** , selezionare **Copia chiave** per copiare la chiave di titolarità negli appunti.
2. Creare un segreto contenente la chiave di titolarità, nel progetto in cui si desidera distribuire il gestore code.

Eeguire il seguente comando, dove *< entitlement-key>* è la chiave richiamata nel passo 1 e *< user-email>* è l'ID IBM associato al software autorizzato.

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

Operazioni successive

[“Distribuzione di un gestore code utilizzando la CLI OpenShift” a pagina 19](#)

V 9.1.5

Linux

MQ Adv.

CD

Distribuzione di un gestore code utilizzando IBM Cloud Pak for Integration Platform Navigator

Utilizzare la risorsa personalizzata QueueManager per distribuire un gestore code su un cluster Red Hat OpenShift Container Platform utilizzando IBM Cloud Pak for Integration Platform Navigator. Questa attività deve essere completata da un amministratore del progetto

Prima di iniziare

In un browser, avviare IBM Cloud Pak for Integration Platform Navigator.

Se questa è la prima distribuzione di un gestore code in questo progetto Red Hat OpenShift , seguire la procedura per [“Preparazione del tuo progetto OpenShift per IBM MQ utilizzando la CLI di OpenShift” a pagina 16.](#)

Procedura

1. Distribuire un gestore code.

Il seguente esempio distribuisce un gestore code "avvio rapido", che utilizza memoria effimera (non persistente) e disattiva la sicurezza di MQ . I messaggi non saranno resi persistenti durante i riavvii del gestore code. È possibile modificare la configurazione per modificare molte impostazioni del gestore code.

a) In IBM Cloud Pak for Integration Platform Navigator, fare clic su **Runtime e istanze**.

b) Fai clic su **Crea istanza**.

c) Selezionare **Gestore code** e fare clic su **Avanti**.

Viene visualizzato il modulo per creare un'istanza di un QueueManager .

Nota: È anche possibile fare clic su **Codice** per visualizzare o modificare l'YAML di configurazione QueueManager .

d) Nella sezione **Dettagli** , controllare o aggiornare il campo **Nome** e specificare lo **Spazio dei nomi** in cui creare l'istanza del gestore code.

e) Se si accetta l'accordo di licenza IBM Cloud Pak for Integration , modificare **Accettazione licenza** in **On**.

È necessario accettare la licenza per distribuire un gestore code.

f) Nella sezione **Configurazione del gestore code** , controllare o aggiornare il **nome** del gestore code sottostante.

Per impostazione predefinita, il nome del gestore code utilizzato dalle applicazioni client IBM MQ sarà uguale al nome di QueueManager , ma con tutti i caratteri non validi (come i trattini) rimossi. Se si desidera forzare l'uso di un particolare nome, è possibile modificarlo qui.

g) Fai clic su **Crea**

Viene ora visualizzato l'elenco dei gestori code nel progetto corrente (spazio dei nomi). Il nuovo QueueManager deve avere lo stato Pending

2. Verificare che il gestore code sia in esecuzione

La creazione è completa quando lo stato di QueueManager è Running.

Attività correlate

[“Connessione a un gestore code distribuito in un cluster OpenShift” a pagina 30](#)

Una serie di esempi di configurazione per la connessione a un gestore code distribuito in un cluster Red Hat OpenShift .

[“Connessione al IBM MQ Console distribuito in un cluster OpenShift” a pagina 32](#)

Modalità di connessione a IBM MQ Console di un gestore code distribuito su un cluster Red Hat OpenShift Container Platform .

V 9.1.5 Linux MQ Adv. CD **Distribuzione di un gestore code utilizzando la Console web OpenShift**

Utilizzare la risorsa personalizzata QueueManager per distribuire un gestore code su un cluster Red Hat OpenShift Container Platform utilizzando la console web Red Hat OpenShift . Questa attività deve essere completata da un amministratore del progetto

Prima di iniziare

Accedi alla console web del tuo cluster OpenShift . Sarà necessario selezionare un progetto esistente (namespace) da utilizzare o crearne uno nuovo.

Se questa è la prima distribuzione di un gestore code in questo progetto Red Hat OpenShift , seguire la procedura per [“Preparazione del tuo progetto OpenShift per IBM MQ utilizzando la CLI di OpenShift” a pagina 16.](#)

Procedura

1. Distribuire un gestore code.

Il seguente esempio distribuisce un gestore code "avvio rapido", che utilizza memoria effimera (non persistente) e disattiva la sicurezza di MQ . I messaggi non saranno resi persistenti durante i riavvii del gestore code. È possibile modificare la configurazione per modificare molte impostazioni del gestore code.

- a) Nella console Web OpenShift , dal riquadro di navigazione, fare clic su **Operatori > Operatori installati**
- b) Fare clic su **IBM MQ**.
- c) Fare clic sulla scheda **Gestore code** .
- d) Fare clic sul pulsante **Crea QueueManager** .

Viene visualizzato un editor YAML, contenente YAML di esempio per una risorsa QueueManager .

Nota: È anche possibile fare clic su **Modifica modulo** per visualizzare o modificare la configurazione di QueueManager .

- e) Se si accetta l'accordo di licenza, modificare **Accettazione licenza** in **On**.

IBM MQ è disponibile con diverse licenze. Per ulteriori informazioni sulle licenze valide, consultare [“Riferimento per la licenza per mq.ibm.com/v1beta1” a pagina 40.](#) È necessario accettare la licenza per distribuire un gestore code.

- f) Fai clic su **Crea**

Viene ora visualizzato l'elenco dei gestori code nel progetto corrente (spazio dei nomi). Il nuovo QueueManager deve essere in uno stato Pending .

2. Verificare che il gestore code sia in esecuzione

La creazione è completa quando lo stato di QueueManager è Running.

Attività correlate

[“Connessione a un gestore code distribuito in un cluster OpenShift” a pagina 30](#)

Una serie di esempi di configurazione per la connessione a un gestore code distribuito in un cluster Red Hat OpenShift .

[“Connessione al IBM MQ Console distribuito in un cluster OpenShift” a pagina 32](#)

Modalità di connessione a IBM MQ Console di un gestore code distribuito su un cluster Red Hat OpenShift Container Platform .

V 9.1.5 Linux MQ Adv. CD **Distribuzione di un gestore code utilizzando la CLI OpenShift**

Utilizza la risorsa di personalizzazione QueueManager per distribuire un gestore code su un cluster Red Hat OpenShift Container Platform utilizzando la CLI (command line interface). Questa attività deve essere completata da un amministratore del progetto

Prima di iniziare

È necessario installare la CLI (command - line interface) [Red Hat OpenShift Container Platform](#).

Accedi al tuo cluster utilizzando **cloudctl login** (per IBM Cloud Pak for Integration) o **oc login**.

Se questa è la prima distribuzione di un gestore code in questo progetto Red Hat OpenShift , seguire la procedura per [“Preparazione del tuo progetto OpenShift per IBM MQ utilizzando la CLI di OpenShift” a pagina 16](#).

Procedura

1. Distribuire un gestore code.

Il seguente esempio distribuisce un gestore code "avvio rapido", che utilizza memoria effimera (non persistente) e disattiva la sicurezza di MQ . I messaggi non saranno resi persistenti durante i riavvii del gestore code. È possibile modificare il contenuto di YAML per modificare molte impostazioni del gestore code.

a) Crea un file YAML di QueueManager

Ad esempio, per installare un gestore code di base in IBM Cloud Pak for Integration, creare un file "mq-quickstart.yaml" con i seguenti contenuti:

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

Importante: se si accetta l'accordo di licenza IBM Cloud Pak for Integration , modificare `accept : false` in `accept : true`. Consultare [“Riferimento per la licenza per mq.ibm.com/v1beta1” a pagina 40](#) per i dettagli sulla licenza.

Questo esempio include anche un server Web distribuito con il gestore code, con la console web abilitata con Single Sign - On con Cloud Pak Identity and Access Manager.

Per installare un gestore code di base indipendentemente da IBM Cloud Pak for Integration, crea il file `mq-quickstart.yaml` con il seguente contenuto:

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

Importante: se si accetta l'accordo di licenza di MQ, modificare `accept: false` in `accept: true`. Consultare [“Riferimento per la licenza per mq.ibm.com/v1beta1”](#) a pagina 40 per i dettagli sulla licenza.

b) Creare l'oggetto QueueManager

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

2. Verificare che il gestore code sia in esecuzione

È possibile convalidare la distribuzione eseguendo

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

, e quindi controllare lo stato.

Ad esempio, eseguire

```
oc describe queuemanager quickstart
```

, e verificare che il campo `status.Phase` indichi `Running`

Attività correlate

[“Connessione a un gestore code distribuito in un cluster OpenShift”](#) a pagina 30

Una serie di esempi di configurazione per la connessione a un gestore code distribuito in un cluster Red Hat OpenShift .

[“Connessione al IBM MQ Console distribuito in un cluster OpenShift”](#) a pagina 32

Modalità di connessione a IBM MQ Console di un gestore code distribuito su un cluster Red Hat OpenShift Container Platform .

Linux

MQ Adv.

CD

V 9.1.4

Integrazione con IBM Cloud Pak for Integration Operations Dashboard

La capacità di tracciare le transazioni tramite IBM Cloud Pak for Integration è fornita dal dashboard Operazioni.

Informazioni su questa attività

L'abilitazione dell'integrazione con Operations Dashboard installa un'uscita dell'API MQ nel gestore code. L'uscita API invierà i dati di traccia all'archivio dati di Operations Dashboard, relativi ai messaggi che passano attraverso il gestore code.

Tenere presente che viene eseguita la traccia solo dei messaggi inviati utilizzando i collegamenti client MQ .

Procedura

1. Distribuisci un gestore code con la traccia abilitata

Per impostazione predefinita, la funzione di traccia è disabilitata.

Se si sta eseguendo la distribuzione utilizzando IBM Cloud Pak for Integration Platform Navigator, è possibile abilitare la traccia durante la distribuzione, impostando **Abilita traccia** su **One** impostando **Spazio dei nomi di traccia** sullo spazio dei nomi in cui è installato Operations Dashboard. Per ulteriori informazioni sulla distribuzione di un gestore code, consultare [“Distribuzione di un gestore code utilizzando IBM Cloud Pak for Integration Platform Navigator”](#) a pagina 17

Se stai eseguendo la distribuzione utilizzando la [CLIOpenShift](#) o la [console webOpenShift](#), puoi abilitare la traccia con il seguente frammento YAML:

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

Se stai eseguendo la distribuzione utilizzando Helm, puoi abilitare la traccia impostando `odTracingConfig.enabled=true` e `odTracingConfig.odTracingNamespace=<Operations_Dashboard_Namespace`. Se si desidera abilitare l'integrazione di Operations Dashboard su un gestore code esistente, è possibile applicare questa impostazione durante l'aggiornamento della release Helm .

Importante: il gestore code non verrà avviato fino a quando MQ non sarà stato registrato con il dashboard Operazioni (consultare il passo successivo).

Tenere presente che quando questa funzione è abilitata, eseguirà due contenitori sidecar ("Agent" e "Collector") in aggiunta al contenitore del gestore code. Le immagini per questi contenitori laterali saranno disponibili nello stesso registro dell'immagine MQ principale e utilizzeranno la stessa politica di pull e il segreto di pull. Sono disponibili ulteriori impostazioni per configurare i limiti di CPU e memoria.

2. Se questa è la prima volta che un gestore code con l'integrazione di Operations Dashboard viene distribuito in questo spazio dei nomi, è necessario [registrare](#) con Operations Dashboard.

La registrazione crea un oggetto segreto che il pod del gestore code deve avviare correttamente.

Creazione di un'immagine con file MQSC e INI personalizzati, utilizzando la CLI OpenShift

Utilizzare una pipeline Red Hat OpenShift Container Platform per creare una nuova immagine contenitore IBM MQ , con i file MQSC e INI che si desidera applicare ai gestori code che utilizzano questa immagine. Questa attività deve essere completata da un amministratore del progetto

Prima di iniziare

È necessario installare la CLI (command - line interface) [Red Hat OpenShift Container Platform](#).

Accedi al tuo cluster utilizzando **cloudctl login** (per IBM Cloud Pak for Integration) o **oc login**.

Se non hai un segreto OpenShift per IBM Entitled Registry nel tuo progetto Red Hat OpenShift , segui la procedura per [“Preparazione del tuo progetto OpenShift per IBM MQ utilizzando la CLI di OpenShift”](#) a pagina 16.

Procedura

1. Crea un ImageStream

Un flusso di immagini e i tag associati forniscono un'astrazione per fare riferimento alle immagini del contenitore da Red Hat OpenShift Container Platform. Il flusso di immagini e le relative tag consentono di vedere quali immagini sono disponibili e di verificare che si stia utilizzando l'immagine specifica necessaria anche se l'immagine nel repository cambia.

```
oc create imagestream mymq
```

2. Crea un BuildConfig per la nuova immagine

Un BuildConfig consentirà le build per la tua nuova immagine, che si baserà sulle immagini ufficiali di IBM , ma aggiungerà tutti i file MQSC o INI che vuoi eseguire all'avvio del contenitore.

a) Creare un file YAML che definisca la risorsa BuildConfig

Ad esempio, creare un file denominato "mq-build-config.yaml" con il seguente contenuto:

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

Sarà necessario sostituire le due posizioni in cui viene menzionato il IBM MQ di base, in modo da puntare all'immagine di base corretta per la versione e la correzione che si desidera utilizzare. Man mano che le correzioni vengono applicate, sarà necessario ripetere questi passi per ricreare l'immagine.

Questo esempio crea una nuova immagine basata sull'immagine ufficiale IBM e aggiunge i file denominati "my.mqsc" e "my.ini" nella directory /etc/mqm . Tutti i file MQSC o INI trovati in questa directory verranno applicati dal container all'avvio. I file INI vengono applicati utilizzando l'opzione **crtmqm -ii** e uniti ai file INI esistenti. I file MQSC vengono applicati in ordine alfabetico.

È importante che i comandi MQSC siano ripetibili, poiché verranno eseguiti *ad ogni* avvio del gestore code. Ciò in genere significa aggiungere il parametro REPLACE su qualsiasi comando DEFINE e aggiungere il parametro IGNSTATE (YES) a qualsiasi comando START o STOP .

b) Applicare il BuildConfig al server.

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

3. Esegui una build per creare la tua immagine

a) Avvia la build

```
oc start-build mymq
```

L'output dovrebbe essere simile al seguente:

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

b) Verificare lo stato della build

Ad esempio, è possibile eseguire il seguente comando, utilizzando l'identificativo di build restituito nel passaggio precedente:

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

4. Distribuisci un gestore code, utilizzando la nuova immagine

Segui la procedura descritta in [“Distribuzione di un gestore code utilizzando la CLI OpenShift”](#) a pagina 19, aggiungendo la nuova immagine personalizzata in YAML.

Puoi aggiungere il seguente frammento di YAML nel tuo normale QueueManager YAML, dove *my - namespace* è il progetto / spazio dei nomi OpenShift che stai utilizzando e *image* è il nome dell'immagine che hai creato in precedenza (ad esempio, "mymq:latest-amd64"):

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

Attività correlate

[“Distribuzione di un gestore code utilizzando la CLI OpenShift”](#) a pagina 19

Utilizza la risorsa di personalizzazione QueueManager per distribuire un gestore code su un cluster Red Hat OpenShift Container Platform utilizzando la CLI (command line interface). Questa attività deve essere completata da un amministratore del progetto

Linux

MQ Adv.

CD

Distribuzione di contenitori certificati IBM MQ utilizzando Helm

Da IBM MQ 9.1.5.0, il modo consigliato per distribuire un gestore code è utilizzare l'operatore IBM MQ. IBM MQ 9.1.5.0 e release CD precedenti possono essere distribuite utilizzando Helm, utilizzando le seguenti istruzioni.

Informazioni su questa attività

Procedura

- [“Preparazione del tuo cluster OpenShift per IBM MQ su OpenShift utilizzando Helm”](#) a pagina 23.
- [“Distribuzione di un gestore code utilizzando la CLI Helm”](#) a pagina 24.

Linux

MQ Adv.

CD

Preparazione del tuo cluster OpenShift per IBM MQ su OpenShift utilizzando Helm

Prepara il tuo cluster Red Hat OpenShift Container Platform in modo che sia pronto a distribuire un gestore code utilizzando Helm. Questa attività deve essere completata da un responsabile cluster.

Prima di iniziare

Nota: Se si utilizza IBM Cloud Pak for Integration, il programma di installazione dovrebbe aver preparato un progetto OpenShift (spazio dei nomi) da utilizzare con IBM MQ, quindi potrebbe non essere necessario seguire queste istruzioni.

Accedi al tuo cluster utilizzando **cloudctl login** (per IBM Cloud Pak for Integration) o **oc login**.

Procedura

1. Assicurati di aver aggiunto il repository IBM Helm alla copia locale di Helm.

Ad esempio, puoi immettere il seguente comando:

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

2. Assicurati di avere un server Helm (denominato "Tiller") installato sul tuo cluster.
Segui le istruzioni in [Introduzione a Helm su OpenShift](#) per installare Helm sul tuo cluster.
3. Assicurati che gli account di servizio nel progetto OpenShift (spazio dei nomi) siano autorizzati a utilizzare gli SCC (Security Context Constraint) corretti.

V 9.1.5 IBM MQ funziona con l'SCC predefinito di "restricted", quindi questo passo può essere normalmente ignorato.

L'applicazione di modifiche agli SCC deve essere effettuata dall'amministratore del cluster OpenShift. Ogni versione del grafico Helm ha requisiti differenti per gli SCC, documentati nei singoli file README per tale grafico Helm :

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

Ci sono istruzioni in ogni README per impostare l'autorizzazione per gli SCC. Tieni presente che i grafici IBM MQ Helm creano un account di servizio per il proprio utilizzo, il che significa che le autorizzazioni SCC devono essere applicate al livello "gruppo" (per tutti gli account di servizio nello spazio dei nomi).

4. Assicurarsi di disporre di un "segreto di pull dell'immagine" valido per eseguire il pull delle immagini dal registro del contenitore scelto

Le immagini IBM MQ Advanced certified container vengono estratte da un registro del contenitore che esegue un controllo della titolarità della licenza. Questo controllo necessita di una chiave di titolarità memorizzata in un segreto di pull `docker-registry`. Se non si dispone ancora di una chiave di titolarità, seguire queste istruzioni per ottenere una chiave di titolarità e creare un segreto di pull.

- a) Ottieni la chiave di titolarità assegnata al tuo ID.
 - i) Accedere a [MyIBM Container Software Library](#) con l'ID e la password IBM associati al software autorizzato.
 - ii) Nella sezione *Chiavi di titolarità*, selezionare **Copia chiave** per copiare la chiave di titolarità negli appunti.
- b) Crea il segreto nello spazio dei nomi in cui vuoi distribuire il tuo gestore code.
 - Eseguire il seguente comando, dove `<entitlement-key>` è la chiave richiamata nel passo 1 e `<user-email>` è l'ID IBM associato al software autorizzato.

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

Operazioni successive

[“Distribuzione di un gestore code utilizzando la CLI Helm” a pagina 24](#)

Linux > MQ Adv. > CD > V 9.1.4 **Distribuzione di un gestore code utilizzando la CLI Helm**

Utilizza Helm per distribuire un gestore code in un cluster Red Hat OpenShift Container Platform. Questa attività deve essere completata da un amministratore del progetto.

Prima di iniziare

Devi installare Helm V2 e la CLI (command - line interface) Red Hat OpenShift Container Platform. Se non stai usando IBM Cloud Pak for Integration, segui la procedura per [“Preparazione del tuo cluster OpenShift per IBM MQ su OpenShift utilizzando Helm”](#) a pagina 23.

Accedi al tuo cluster utilizzando **cloudctl login** (per IBM Cloud Pak for Integration) o **oc login**.

Procedura

1. Assicurati di aver aggiunto il repository IBM Helm alla copia locale di Helm.

Ad esempio, puoi immettere il seguente comando:

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

2. Esaminare le opzioni di configurazione per il gestore code

La fase di distribuzione comprende sia la procedura di installazione che la procedura di configurazione. Alcune impostazioni per il gestore code devono essere impostate al momento della distribuzione e la loro modifica richiede una nuova distribuzione.

Puoi visualizzare il README del diagramma Helm per dettagli su tutte le opzioni di distribuzione disponibili, eseguendo uno dei seguenti comandi:

- Per IBM MQ Advanced certified container in IBM Cloud Pak for Integration:

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

- Per IBM MQ Advanced certified container:

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

In genere sono necessari almeno i parametri seguenti:

- a. Nome release. Ad esempio: `my-release`
 - b. Repository Helm remoto. Ad esempio: `ibm-entitled-charts`
 - c. Grafico Helm : ad esempio `ibm-mqadvanced-server-prod` o `ibm-mqadvanced-server-integration-prod`
 - d. Nome segreto pull immagine. Ad esempio: `entitled-registry`. Notare che ciò non è necessario se si sta eseguendo la distribuzione nel progetto predefinito per MQ in IBM Cloud Pak for Integration
3. Distribuire un gestore code.

Tieni presente che, per impostazione predefinita, il grafico Helm presuppone che tu abbia una Classe di archiviazione predefinita impostata nel tuo cluster Red Hat OpenShift Container Platform.

Ad esempio, per installare un gestore code di base in IBM Cloud Pak for Integration, eseguire il seguente comando:

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

È possibile immettere qualsiasi nome host nel campo `tls.hostname` (questo è un campo obbligatorio ma non verrà utilizzato come in questo esempio si sta generando un nuovo certificato autofirmato)

Per installare un gestore code di base indipendentemente da IBM Cloud Pak for Integration, è possibile eseguire il seguente comando:

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

Attività correlate

“Connessione a un gestore code distribuito in un cluster OpenShift” a pagina 30

Una serie di esempi di configurazione per la connessione a un gestore code distribuito in un cluster Red Hat OpenShift .

“Connessione al IBM MQ Console distribuito in un cluster OpenShift” a pagina 32

Modalità di connessione a IBM MQ Console di un gestore code distribuito su un cluster Red Hat OpenShift Container Platform .

V 9.1.5 Linux MQ Adv. CD **Distribuzione di un gestore code con IBM Cloud File Storage, utilizzando la CLI Helm**

Scenario di esempio per utilizzare Helm per distribuire un gestore code in un cluster Red Hat OpenShift su IBM Cloud , utilizzando IBM Cloud File Storage. Questa attività deve essere completata da un amministratore del progetto

Prima di iniziare

Devi installare Helm V2 e la CLI (command - line interface) Red Hat OpenShift Container Platform. Se non stai usando IBM Cloud Pak for Integration, segui la procedura per [“Preparazione del tuo cluster OpenShift per IBM MQ su OpenShift utilizzando Helm”](#) a pagina 23.

Accedi al tuo cluster utilizzando **cloudctl login** (per IBM Cloud Pak for Integration) o **oc login**.

Procedura

1. Assicurati di aver aggiunto il repository IBM Helm alla copia locale di Helm.

Ad esempio, puoi immettere il seguente comando:

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

2. Distribuire un gestore code.

Quando si utilizza IBM Cloud File Storage, generalmente vengono visualizzati i migliori risultati utilizzando la classe di memoria `ibmc-file-gold-gid` . Questa classe di memorizzazione abilita la memorizzazione che può essere scritta dagli utenti nel corretto gruppo di file system.

Ad esempio, per installare un gestore code di base in IBM Cloud Pak for Integration, eseguire il seguente comando:

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

È possibile immettere qualsiasi nome host nel campo `tls.hostname` (questo è un campo obbligatorio ma non viene utilizzato qui, perché in questo esempio si sta generando un nuovo certificato autofirmato).

Per installare un gestore code di base indipendentemente da IBM Cloud Pak for Integration, è possibile eseguire il seguente comando:

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

Attività correlate

[“Connessione a un gestore code distribuito in un cluster OpenShift” a pagina 30](#)

Una serie di esempi di configurazione per la connessione a un gestore code distribuito in un cluster Red Hat OpenShift .

[“Connessione al IBM MQ Console distribuito in un cluster OpenShift” a pagina 32](#)

Modalità di connessione a IBM MQ Console di un gestore code distribuito su un cluster Red Hat OpenShift Container Platform .

Linux

MQ Adv.

CD

Distribuzione di release CD precedenti di IBM MQ in un cluster IBM Cloud Private

Per le versioni CD di IBM MQ precedenti a 9.1.4, utilizzare la console di gestione IBM Cloud Private per distribuire un gestore code in IBM Cloud Private.

Prima di iniziare



Attenzione: **V 9.1.4** Questa distribuzione non è supportata in IBM MQ 9.1.4 o versioni successive.

Questa attività presuppone che tu abbia già [aggiunto un'immagine IBM MQ in un cluster IBM Cloud Private](#).

Il file Helm chart README . md è disponibile dalla voce di catalogo IBM Cloud Private , che viene visualizzata dopo aver completato questo [passo secondario](#), o dalla riga di comando [aggiungendo il tuo repository **local-charts** di IBM Cloud Private come Helm repository remoto](#) ed eseguendo il seguente comando:

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

È necessario disporre di una [politica PodSecurity](#) di un [SecurityContextConstraint](#) (per IBM Cloud Private su Red Hat OpenShift) che supporta il contesto di sicurezza necessario. I dettagli, inclusi gli esempi, possono essere trovati dal file Helm chart README . md .

I dettagli su come configurare la tua release Helm sono disponibili anche nel file Helm chart README . md .

Nota:

- Se si sta eseguendo la distribuzione a un ambiente IBM Cloud Private che non supporta le impostazioni di sicurezza richieste per impostazione predefinita, abilitare la distribuzione seguendo le istruzioni riportate in [Distribuzione dei grafici Helm che richiedono privilegi elevati in uno spazio dei nomi non predefinito](#) nella documentazione del prodotto IBM Cloud Private .
- Se si utilizza SELinux, è necessario soddisfare i requisiti IBM MQ descritti in [IBM MQ per SELinux su Red Hat Enterprise Linux](#).

Informazioni su questa attività

IBM Cloud Private offre una piattaforma per la gestione in loco delle applicazioni inserite in un contenitore. Dopo aver aggiunto un'immagine IBM MQ a un cluster IBM Cloud Private , puoi utilizzare la console di gestione IBM Cloud Private o la riga di comando per distribuire un gestore code.

Procedura

- Utilizzo della console di gestione IBM Cloud Private
 - a) Apri la console di gestione IBM Cloud Private in un browser Web e fai clic su **Catalogo**.
Vedi [Accesso al tuo cluster IBM Cloud Private utilizzando la console di gestione](#) nella documentazione del prodotto IBM Cloud Private .
 - b) Selezionare il grafico `ibm - mqadvanced - server - prod` dall'elenco.
 - c) Selezionare **Configura**, quindi completare la seguente procedura di configurazione:
 - a. Immettere un nome di release.
 - b. Leggere e accettare gli accordi di licenza.
 - c. Nella sezione **dataPVC** , imposta **storageclass** sulla classe di archiviazione desiderata. Lasciare vuoto per selezionare la classe di memoria predefinita.
 - d. Nella sezione **image** , imposta il repository sul percorso dell'immagine completo. Ad esempio:

```
mycluster.icp:8500/namespace_name/ibm-mqadvanced-server-prod
```
 - e. Nella sezione **image** , imposta la tag sulla tag image. Ad esempio:

```
9.1.3.0-r1
```
 - f. Se hai bisogno di un segreto di pull Kubernetes per accedere al registro delle immagini, aggiungilo come **pullSecret**.
 - g. Nella sezione **queueManager** , impostare il nome del gestore code.
 - d) Fai clic su **Install** per distribuire il tuo gestore code come una *releaseHelm*.
- Utilizzo della riga di comando
 - a) Configurare **cloudctl** per accedere al cluster IBM Cloud Private .
Vedi [Installazione della CLI IBM Cloud Private](#) nella documentazione del prodotto IBM Cloud Private .
 - b) Assicurati di aver [aggiunto il tuo repository **local-charts** di IBM Cloud Private come repository Helm remoto](#).
 - c) Installa il grafico.
Esegui il seguente comando, specificando questi parametri:
 - a. Nome release (ad esempio `my-release`)
 - b. Nome del repository Helm remoto che contiene il grafico `ibm-mqadvanced-server-prod` (ad esempio `my-repo`)
 - c. Repository immagini (ad esempio `mycluster.icp:8500/namespace_name/ibm-mqadvanced-server-prod`)
 - d. Tag immagine (ad esempio `9.1.3.0-r1`)

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

Attività correlate

[“Distribuzione di un gestore code utilizzando la CLI Helm”](#) a pagina 24

Utilizza Helm per distribuire un gestore code in un cluster Red Hat OpenShift Container Platform. Questa attività deve essere completata da un amministratore del progetto.

[“Aggiunta di release CD precedenti di un'immagine di IBM MQ in un cluster IBM Cloud Private”](#) a pagina 29

Per le versioni CD di IBM MQ precedenti a 9.1.4, preparare il cluster IBM Cloud Private per distribuire un'immagine pronta per la produzione per IBM MQ.

[“Aggiunta di release CD precedenti di un'immagine di IBM MQ in un cluster IBM Cloud Kubernetes Service” a pagina 29](#)

Per le versioni CD di IBM MQ precedenti a 9.1.4, importa un'immagine pronta per la produzione per IBM MQ in IBM Cloud Kubernetes Service.

Linux > MQ Adv. > CD **Aggiunta di release CD precedenti di un'immagine di IBM MQ in un cluster IBM Cloud Private**

Per le versioni CD di IBM MQ precedenti a 9.1.4, preparare il cluster IBM Cloud Private per distribuire un'immagine pronta per la produzione per IBM MQ.

Informazioni su questa attività



Attenzione: **V 9.1.4** Questa importazione non è supportata in IBM MQ 9.1.4 o versioni successive.

Puoi scaricare un'immagine IBM MQ da Passport Advantage e importarla in un contenitore IBM Cloud Private .

Procedura

1. Scarica l'ultima immagine IBM MQ da [Sito Web Passport Advantage e Passport Advantage Express](#).

Per i dettagli dei download disponibili, visitare il sito [Download di IBM MQ 9.1](#) quindi fare clic sulla scheda per la release che si desidera scaricare. Il nome e il numero della parte da scaricare sono elencati in una tabella.

2. Importare il file di archivio scaricato in IBM Cloud Private.

Vedi [Aggiunta di software IBM a IBM Cloud Private Catalog](#) nella documentazione del prodotto IBM Cloud Private .

Operazioni successive

È ora possibile [distribuire un gestore code in IBM Cloud Private](#).

Attività correlate

[“Distribuzione di un gestore code utilizzando la CLI Helm” a pagina 24](#)

Utilizza Helm per distribuire un gestore code in un cluster Red Hat OpenShift Container Platform. Questa attività deve essere completata da un amministratore del progetto.

[“Distribuzione di release CD precedenti di IBM MQ in un cluster IBM Cloud Private” a pagina 27](#)

Per le versioni CD di IBM MQ precedenti a 9.1.4, utilizzare la console di gestione IBM Cloud Private per distribuire un gestore code in IBM Cloud Private.

[“Aggiunta di release CD precedenti di un'immagine di IBM MQ in un cluster IBM Cloud Kubernetes Service” a pagina 29](#)

Per le versioni CD di IBM MQ precedenti a 9.1.4, importa un'immagine pronta per la produzione per IBM MQ in IBM Cloud Kubernetes Service.

Linux > MQ Adv. > CD **Aggiunta di release CD precedenti di un'immagine di IBM MQ in un cluster IBM Cloud Kubernetes Service**

Per le versioni CD di IBM MQ precedenti a 9.1.4, importa un'immagine pronta per la produzione per IBM MQ in IBM Cloud Kubernetes Service.

Informazioni su questa attività



Attenzione: **V 9.1.4** Questa importazione non è supportata in IBM MQ 9.1.4 o versioni successive.

Puoi scaricare un'immagine IBM MQ da Passport Advantage e importarla in un cluster IBM Cloud Kubernetes Service .

Procedura

1. Scarica l'ultima immagine IBM MQ da [Sito Web Passport Advantage e Passport Advantage Express](#).

Per i dettagli dei download disponibili, visitare il sito [Download di IBM MQ 9.1](#) quindi fare clic sulla scheda per la release che si desidera scaricare. Il nome e il numero della parte da scaricare sono elencati in una tabella.

2. Importare il file di archivio scaricato in IBM Cloud Kubernetes Service.

Vedi [Esecuzione di immagini IBM Cloud Private in contenitori Kubernetes pubblici](#).

Attività correlate

[“Distribuzione di un gestore code utilizzando la CLI Helm” a pagina 24](#)

Utilizza Helm per distribuire un gestore code in un cluster Red Hat OpenShift Container Platform. Questa attività deve essere completata da un amministratore del progetto.

[“Distribuzione di release CD precedenti di IBM MQ in un cluster IBM Cloud Private” a pagina 27](#)

Per le versioni CD di IBM MQ precedenti a 9.1.4, utilizzare la console di gestione IBM Cloud Private per distribuire un gestore code in IBM Cloud Private.

[“Aggiunta di release CD precedenti di un'immagine di IBM MQ in un cluster IBM Cloud Private” a pagina 29](#)

Per le versioni CD di IBM MQ precedenti a 9.1.4, preparare il cluster IBM Cloud Private per distribuire un'immagine pronta per la produzione per IBM MQ.

Linux → MQ Adv. → CD → V 9.1.4 Connessione a un gestore code distribuito in un cluster OpenShift

Una serie di esempi di configurazione per la connessione a un gestore code distribuito in un cluster Red Hat OpenShift .

Informazioni su questa attività

Hai bisogno di un [instradamentoOpenShift](#) per connettere un'applicazione a un gestore code IBM MQ dall'esterno di un cluster Red Hat OpenShift .

È necessario abilitare TLS sul gestore code e sull'applicazione client IBM MQ , perché [SNI \(Server Name Indication\)](#) è disponibile solo nel protocollo TLS. Il router Red Hat OpenShift Container Platform usa SNI per instradare le richieste al gestore code IBM MQ .

La configurazione richiesta dell'instradamento OpenShift dipende dal comportamento SNI dell'applicazione client.

Per impostare l'intestazione SNI come TLS 1.2 o superiore, è necessario utilizzare un CipherSpec o CipherSuite per la comunicazione TLS.

SNI è impostato sul canale MQ se sono soddisfatte le condizioni riportate di seguito:

- Il client IBM MQ C è V8 o successivo.
- Il client Java/JMS è V9.1.1 o versione successiva e l'installazione Java supporta la classe `javax.net.ssl.SNIHostName` .
- Il client .NET è in modalità non gestita.

SNI viene impostato sul nome host se viene fornito un nome host come nome connessione e vengono soddisfatte le seguenti condizioni:

- Il client .NET è in modalità gestita.
- Viene utilizzato il client AMQP o XR.

- I client Java/JMS vengono utilizzati con **AllowOutboundSNI** impostato su NO.

SNI non è impostato ed è vuoto nelle condizioni seguenti:

- IBM MQ C Client è V7.5 o precedente.
- IBM MQ C Client viene utilizzato con **AllowOutboundSNI** impostato su NO.
- I client Java/JMS vengono utilizzati con un'installazione Java che non supporta la classe `javax.net.ssl.SNIHostName`.

Esempio

Instradamenti basati sul nome host OpenShift : per applicazioni client che impostano SNI sul nome host

I seguenti grafici Helm creano automaticamente un instradamento OpenShift basato sul nome host per la connessione di un'applicazione a un gestore code IBM MQ . Le applicazioni client che impostano SNI sul nome host possono utilizzare questo instradamento OpenShift .

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

Se non stai utilizzando questi grafici e devi creare il tuo percorso OpenShift basato sul tuo nome host, puoi applicare il seguente `yaml` nel tuo cluster:

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

Instradamenti MQ basati sul canale OpenShift : per le applicazioni client che impostano SNI sul canale MQ

Le applicazioni client che impostano SNI per il canale MQ richiedono la creazione di un nuovo instradamento OpenShift per ogni canale a cui si desidera connettersi. Devi anche utilizzare nomi di canale univoci nel tuo cluster Red Hat OpenShift , per consentire l'instradamento al gestore code corretto.

Per determinare il nome dell'host richiesto per ciascuno dei tuoi nuovi instradamenti OpenShift , devi associare ciascun nome del canale a un indirizzo SNI come documentato qui: <https://www.ibm.com/support/pages/ibm-websphere-mq-how-does-mq-provide-multiple-certificates-certlabl-capability>

Devi quindi creare un nuovo instradamento OpenShift (per ciascun canale) applicando il seguente `yaml` nel tuo cluster:

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

Configurazione dei dettagli di connessione dell'applicazione client

È possibile stabilire il nome host da utilizzare per la connessione client immettendo il seguente comando:

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

La porta per la connessione client deve essere impostata sulla porta utilizzata dal router OpenShift Container Platform (OCP) - Normalmente 443.

Attività correlate

[“Distribuzione di un gestore code utilizzando la CLI Helm” a pagina 24](#)

Utilizza Helm per distribuire un gestore code in un cluster Red Hat OpenShift Container Platform. Questa attività deve essere completata da un amministratore del progetto.

[“Connessione al IBM MQ Console distribuito in un cluster OpenShift” a pagina 32](#)

Modalità di connessione a IBM MQ Console di un gestore code distribuito su un cluster Red Hat OpenShift Container Platform .

Linux

MQ Adv.

CD

V 9.14

Connessione al IBM MQ Console distribuito in un cluster OpenShift

Modalità di connessione a IBM MQ Console di un gestore code distribuito su un cluster Red Hat OpenShift Container Platform .

Informazioni su questa attività

Se stai utilizzando l'operatore IBM MQ , l'URL IBM MQ Console può essere trovato nella pagina dei dettagli QueueManager nella console web OpenShift o in IBM Cloud Pak for Integration Platform Navigator. In alternativa, puoi trovarlo dalla CLI OpenShift immettendo il seguente comando:

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

Esempio

I seguenti grafici Helm creano automaticamente un instradamento OpenShift per accedere a IBM MQ Console

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

Puoi ottenere il nome host dell'instradamento OpenShift immettendo il seguente comando:

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

È possibile accedere a IBM MQ Console utilizzando il seguente URL:

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

Attività correlate

[“Distribuzione di un gestore code utilizzando la CLI Helm” a pagina 24](#)

Utilizza Helm per distribuire un gestore code in un cluster Red Hat OpenShift Container Platform. Questa attività deve essere completata da un amministratore del progetto.

[“Connessione a un gestore code distribuito in un cluster OpenShift” a pagina 30](#)

Una serie di esempi di configurazione per la connessione a un gestore code distribuito in un cluster Red Hat OpenShift .

Backup e ripristino della configurazione del gestore code utilizzando la CLI OpenShift

Il backup della configurazione del gestore code può essere utile per ricreare un gestore code dalle relative definizioni se la configurazione del gestore code viene persa. Questa procedura non esegue il backup dei dati di log del gestore code. A causa della natura transitoria dei messaggi, i dati di log cronologici sono probabilmente irrilevanti al momento del ripristino.

Prima di iniziare

Accedi al tuo cluster utilizzando **cloudctl login** (per IBM Cloud Pak for Integration) o **oc login**.

Procedura

- Eseguire il back up della configurazione del gestore code.

È possibile utilizzare il comando **dmpmqcfg** per eseguire il dump della configurazione di un gestore code IBM MQ .

- Ottenere il nome del pod per il gestore code.

Ad esempio, se si utilizza l'operatore, è possibile eseguire il seguente comando, dove *queue_manager_name* è il nome della risorsa QueueManager :

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

Ad esempio, se stai utilizzando Helm, puoi eseguire il seguente comando, dove *release_name* è il nome della tua release Helm .

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

- Eseguire il comando **dmpmqcfg** sul pod, indirizzando l'emissione in un file sulla macchina locale.

dmpmqcfg emette la configurazione MQSC del gestore code.

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

- Ripristinare la configurazione del gestore code.

Dopo aver seguito la procedura di backup descritta nel passo precedente, si dovrebbe avere un file `backup.mqsc` che contiene la configurazione del gestore code. È possibile ripristinare la configurazione applicando questo file a un nuovo gestore code.

- Ottenere il nome del pod per il gestore code.

Ad esempio, se si utilizza l'operatore, è possibile eseguire il seguente comando, dove *queue_manager_name* è il nome della risorsa QueueManager :

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

Ad esempio, se stai utilizzando Helm, puoi eseguire il seguente comando, dove *release_name* è il nome della tua release Helm .

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

- Eseguire il comando **runmqsc** sul pod, indirizzando il contenuto del file `backup.mqsc` .

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

Creazione del tuo proprio contenitore IBM MQ

Sviluppa un contenitore autonomo, precedentemente noto come "Docker". Questa è la soluzione del contenitore più flessibile, ma ti richiede di avere forti capacità nella configurazione dei contenitori e di "possedere" il contenitore risultante.

Prima di iniziare

Prima di sviluppare il tuo proprio contenitore, considera se puoi invece utilizzare uno dei contenitori preconfezionati forniti da IBM. Consultare [IBM MQ nei contenitori](#)

Informazioni su questa attività

Quando impacchetti IBM MQ come immagine del contenitore, le modifiche alla tua applicazione possono essere distribuite per testare e preparare i sistemi in modo rapido e semplice. Questo può essere un vantaggio importante per la fornitura continua nella tua azienda.

Procedura

- Per informazioni su come creare un'immagine del contenitore IBM MQ utilizzando Docker, vedi i seguenti topic secondari:
 -  [“Supporto per la creazione di grafici e immagini del contenitore IBM MQ” a pagina 8](#)
 - [“Pianificazione della tua immagine del gestore code IBM MQ utilizzando un contenitore” a pagina 34](#)
 - [“Creazione di un'immagine del gestore code IBM MQ di esempio utilizzando Docker” a pagina 35](#)
 - [“Esecuzione di applicazioni di bind locali in contenitori separati” a pagina 37](#)

Concetti correlati

[IBM MQ nei contenitori](#)

Pianificazione della tua immagine del gestore code IBM MQ utilizzando un contenitore

Esistono diversi requisiti da considerare quando si esegue un gestore code IBM MQ in un contenitore. L'immagine del contenitore di esempio fornisce un modo per gestire questi requisiti, ma se vuoi utilizzare la tua immagine, devi considerare come vengono gestiti questi requisiti.

Supervisione dei processi

Quando si esegue un contenitore, si sta essenzialmente eseguendo un singolo processo (PID 1 all'interno del contenitore), che può successivamente generare processi child.

Se il processo principale termina, il runtime del contenitore arresta il contenitore. Un gestore code IBM MQ richiede più processi in esecuzione in background.

Per questo motivo, è necessario assicurarsi che il processo principale rimanga attivo finché il gestore code è in esecuzione. Si consiglia di verificare che il gestore code sia attivo da questo processo, ad esempio, eseguendo query amministrative.

Popolamento di `/var/mqm`

I contenitori devono essere configurati con `/var/mqm` come volume.

Quando si esegue questa operazione, la directory del volume è vuota quando il contenitore viene avviato per la prima volta. Questa directory viene generalmente popolata al momento dell'installazione, ma l'installazione e il runtime sono ambienti separati quando si utilizza un contenitore.

Per risolvere questo problema, quando il contenitore viene avviato, puoi utilizzare il comando `crtmqdir` per popolare `/var/mqm` quando viene eseguito per la prima volta.

Creazione di un'immagine del gestore code IBM MQ di esempio utilizzando Docker

Utilizzare queste informazioni per creare un'immagine del contenitore di esempio per l'esecuzione di un gestore code IBM MQ in un contenitore.

Informazioni su questa attività

Innanzitutto, si crea un'immagine di base contenente un file system Red Hat Universal Base Image e un'installazione pulita di IBM MQ.

In secondo luogo, crei un altro livello di immagine del contenitore sopra la base, che aggiunge alcune configurazioni IBM MQ per consentire la sicurezza di ID utente e password di base.

Infine, si esegue un contenitore che utilizza questa immagine come file system, con il contenuto di `/var/mqm` fornito da un volume specifico del contenitore sul file system host.

Procedura

- Per informazioni su come creare un'immagine del contenitore di esempio per l'esecuzione di un gestore code IBM MQ in un contenitore, consultare i seguenti argomenti secondari:
 - [“Creazione di un'immagine del gestore code IBM MQ di base di esempio”](#) a pagina 35
 - [“Creazione di un'immagine del gestore code IBM MQ configurata di esempio”](#) a pagina 35

Creazione di un'immagine del gestore code IBM MQ di base di esempio

Per utilizzare IBM MQ nella tua immagine contenitore, devi inizialmente creare un'immagine base con un'installazione IBM MQ pulita. La seguente procedura ti mostra come creare un'immagine base di esempio, utilizzando il codice di esempio ospitato su GitHub.

Procedura

- Utilizza i file make forniti nel repository [mq - container GitHub](#) per creare la tua immagine contenitore di produzione.
Attieniti alle istruzioni in [Creazione di un'immagine del contenitore su GitHub](#).

Risultati

Hai ora un'immagine del contenitore di base con IBM MQ installato.

Creazione di un'immagine del gestore code IBM MQ configurata di esempio

Dopo aver creato la tua immagine del contenitore IBM MQ di base generica, devi applicare la tua propria configurazione per consentire l'accesso sicuro. Per fare ciò, creare il proprio livello di immagine contenitore, utilizzando l'immagine generica come elemento principale.

Prima di iniziare

Per un'immagine IBM MQ 9.1, non è possibile configurare l'accesso sicuro utilizzando l'SCC (Security Context Constraint) Red Hat OpenShift Container Platform "limitato". L'SCC "limitato" utilizza ID utente casuali e impedisce l'escalation dei privilegi passando a un utente diverso. Il programma di installazione basato su IBM MQ 9.1 RPM si basa su un utente e un gruppo `mqm` e utilizza anche `bit setuid` su programmi eseguibili.

Questa limitazione viene eliminata in IBM MQ 9.2.

Procedura

1. Creare una nuova directory e aggiungere un file denominato `config.mqsc`, con il seguente contenuto:

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

Tenere presente che l'esempio precedente utilizza l'autenticazione semplice di ID utente e password. Tuttavia, è possibile applicare qualsiasi configurazione di sicurezza richiesta dalla propria azienda.

2. Creare un file denominato `Dockerfile`, con il seguente contenuto:

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

dove:

- `johndoe` è l'ID utente che si desidera aggiungere
- `passw0rd` è la password originale

3. Crea la tua immagine contenitore personalizzata utilizzando il seguente comando:

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

dove `"."` è la directory contenente i due file appena creati.

Docker crea quindi un container temporaneo utilizzando tale immagine ed esegue i restanti comandi.

Il comando **RUN** aggiunge un utente denominato `johndoe` con password `passw0rd` e il comando **COPY** aggiunge il file `config.mqsc` in una specifica posizione nota all'immagine parent.

Nota: Su Red Hat Enterprise Linux (RHEL), utilizzi il comando **docker** (RHEL V7) o **podman** (RHEL V7 o RHEL V8). Nel caso di **podman**, non hai bisogno di **sudo** all'inizio del comando.

4. Esegui la tua nuova immagine personalizzata per creare un nuovo contenitore, con l'immagine disco che hai appena creato.

Il nuovo livello immagine non ha specificato alcun comando particolare da eseguire, quindi è stato ereditato dall'immagine principale. Il punto di immissione del parent (il codice è disponibile su GitHub):

- Crea un gestore code
- Avvia il gestore code
- Crea un listener predefinito
- Quindi, esegue i comandi MQSC da `/etc/mqm/config.mqsc`.

Immetti i comandi seguenti per eseguire la nuova immagine personalizzata:

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

dove:

Primo parametro `env`

Passa una variabile di ambiente nel contenitore, che riconosce l'accettazione della licenza per IBM IBM WebSphere MQ. È anche possibile impostare la variabile `LICENSE` da visualizzare per visualizzare la licenza.

Consultare [IBM MQ informazioni sulla licenza](#) per ulteriori dettagli sulle licenze IBM MQ .

Secondo parametro env

Imposta il nome del gestore code che si utilizza.

Parametro volume

Indica al contenitore che qualsiasi cosa MQ scriva in `/var/mqm` deve essere effettivamente scritta in `/var/example` sull'host.

Questa opzione significa che è possibile eliminare facilmente il contenitore in un secondo momento e conservare ancora i dati persistenti. Questa opzione rende inoltre più semplice la visualizzazione dei file di log.

Pubblica parametro

Associa le porte sul sistema host alle porte nel contenitore. Il contenitore viene eseguito per impostazione predefinita con il suo indirizzo IP interno, il che significa che devi associare in modo specifico tutte le porte che vuoi esporre.

In questo esempio, ciò significa associare la porta 1414 sull'host alla porta 1414 nel contenitore.

Scollega parametro

Esegue il contenitore in background.

Risultati

Hai creato un'immagine del contenitore configurata e puoi visualizzare i contenitori in esecuzione utilizzando il comando docker **ps** . Puoi vedere i processi IBM MQ in esecuzione nel contenitore utilizzando il comando docker **top** .



Attenzione:

Puoi visualizzare i log di un contenitore utilizzando il comando docker **logs** `#{CONTAINER_ID}` .

Operazioni successive

- Se il tuo contenitore non viene visualizzato quando utilizzi il comando docker **ps** , il contenitore potrebbe non essere riuscito. Puoi vedere i contenitori non riusciti utilizzando il comando docker **ps -a** .
- Quando utilizzi il comando docker **ps -a** , viene visualizzato l'ID contenitore. Questo ID è stato stampato anche quando hai immesso il comando docker **run** .
- Puoi visualizzare i log di un contenitore utilizzando il comando docker **logs** `#{CONTAINER_ID}` .
- È possibile impostare un massimo di file aperti utilizzando il comando **sysctl fs.file-max=524288**.

V 9.1.0

Esecuzione di applicazioni di bind locali in contenitori separati

Con la condivisione dello spazio dei nomi del processo tra contenitori in Docker, è possibile eseguire le applicazioni che richiedono una connessione di bind locale a IBM MQ in contenitori separati dal gestore code IBM MQ .

Informazioni su questa attività

Questa funzione è supportata in IBM MQ 9.0.3 e nei gestori code successivi.

È necessario rispettare le seguenti condizioni:

- Devi condividere lo spazio dei nomi PID dei contenitori utilizzando l'argomento `--pid` .
- Devi condividere lo spazio dei nomi IPC dei contenitori utilizzando l'argomento `--ipc` .
- È necessario:
 1. Condividere lo spazio dei nomi UTS dei contenitori con l'host utilizzando l'argomento `--uts` oppure
 2. Verificare che i contenitori abbiano lo stesso nome host utilizzando l'argomento `-h o --hostname` .

- È necessario montare la directory di dati IBM MQ in un volume disponibile per tutti i contenitori nella directory `/var/mqm`.

È possibile provare questa funzionalità completando la seguente procedura su un sistema Linux su cui è già installato Docker.

Il seguente esempio utilizza l'immagine del contenitore IBM MQ di esempio. Puoi trovare i dettagli di questa immagine su [Github](#).

Procedura

1. Creare una directory temporanea che agisca come volume, immettendo il seguente comando:

```
mkdir /tmp/dockerVolume
```

2. Creare un gestore code (QM1) in un contenitore, denominato `sharedNamespace`, immettendo il seguente comando:

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

3. Avviare un secondo contenitore denominato `secondaryContainer`, basato sul `ibmcom/mq`, ma non creare un gestore code immettendo il seguente comando:

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

4. Eseguire il comando **dspmqr** sul secondo contenitore, per visualizzare lo stato di entrambi i gestori code, immettendo il seguente comando:

```
docker exec secondaryContainer dspmqr
```

5. Eseguire il seguente comando per elaborare i comandi MQSC rispetto al gestore code in esecuzione sull'altro contenitore:

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

Risultati

Ora le applicazioni locali sono in esecuzione in contenitori separati e ora è possibile eseguire correttamente comandi come **dspmqr**, **amqsput**, **amqsgete** **runmqsc** come bind locali al gestore code QM1 dal contenitore secondario.

Se non viene visualizzato il risultato previsto, consultare [“Risoluzione dei problemi delle applicazioni dello spazio dei nomi”](#) a pagina 38 per ulteriori informazioni.

V 9.1.0 Risoluzione dei problemi delle applicazioni dello spazio dei nomi

Quando si utilizzano spazi dei nomi condivisi, è necessario assicurarsi di condividere tutti gli spazi dei nomi (IPC, PID e UTS/nomehost) e i volumi montati, altrimenti le applicazioni non funzioneranno.

Consultare [“Esecuzione di applicazioni di bind locali in contenitori separati”](#) a pagina 37 per un elenco delle limitazioni da seguire.

Se la tua applicazione non soddisfa tutte le limitazioni elencate, potresti riscontrare problemi nel punto in cui viene avviato il contenitore, ma la funzionalità prevista non funziona.

Il seguente elenco delinea alcune cause comuni e il comportamento che si sta probabilmente osservando se si è dimenticato di rispettare una delle restrizioni.

- Se dimentichi di condividere lo spazio dei nomi (UTS / PID/IPC) o il nome host dei contenitori e monti il volume, il tuo contenitore sarà in grado di vedere il gestore code ma non di interagire con il gestore code.

- Per i comandi **dspmq** , vedi quanto segue:

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

- Per i comandi **runmqsc** o altri comandi che tentano di connettersi al gestore code, è probabile che si riceva un messaggio di errore AMQ8146 :

```
docker exec -it container runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2024.
Starting MQSC for queue manager QM1.
AMQ8146: IBM MQ queue manager not available
```

- Se condividi tutti gli spazi dei nomi richiesti ma non monti un volume condiviso nella directory `/var/mqm` e hai un percorso dati IBM MQ valido, i tuoi comandi ricevono anche i messaggi di errore AMQ8146 .

Tuttavia, **dspmq** non è in grado di visualizzare il tuo gestore code e restituisce invece una risposta vuota:

```
docker exec container dspmq
```

- Se si condividono tutti gli spazi dei nomi richiesti ma non si monta un volume condiviso nella directory `/var/mqm` e non si dispone di un percorso dati IBM MQ valido (o nessun percorso dati IBM MQ), vengono visualizzati diversi errori poiché il percorso dati è un componente chiave di un'installazione IBM MQ . Senza il percorso dati, IBM MQ non può funzionare.

Se si esegue uno dei seguenti comandi e vengono visualizzate risposte simili a quelle mostrate in questi esempi, è necessario verificare di aver montato la directory o di aver creato una directory di dati 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]
lpi0btainQMDetails --> 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]
lpi0btainQMDetails --> 545261715
```

[V 9.1.5](#) [Linux](#) [MQ Adv.](#) [CD](#) **Riferimento API per l'operatore** **IBM MQ**

IBM MQ fornisce un operatore Kubernetes , che fornisce un'integrazione nativa con OpenShift Container Platform.

v1beta1

L'API v1beta1 può essere utilizzata per creare e gestire risorse QueueManager .

mq.ibm.com/v1beta1

Il campo `spec.license.license` deve contenere l'identificativo della licenza che si sta accettando. I valori validi sono:

Valore di <code>spec.license.license</code>	Valore di <code>spec.license.use</code>	Informazioni sulla licenza
L-RJON-BN7PN3	Production o NonProduction	IBM Cloud Pak for Integration 2020.2
L-RJON-BPHL2Y		IBM Cloud Pak for Integration Edizione limitata 2020.2
L-APIG-BJAKBF	Production o Development	IBM MQ Advanced 9.1.5
L-APIG-BM7GDH	Development	IBM MQ Advanced for Developers 9.1.5

Notare che la licenza *versione* è specificata, che non è sempre uguale alla versione di IBM MQ.

(mq.ibm.com/v1beta1)**QueueManager**

Un QueueManager è un IBM MQ che fornisce servizi di accodamento e pubblicazione / sottoscrizione alle applicazioni.

Campo	Descrizione
<code>apiVersion</code> Stringa	APIVersion definisce lo schema con versione di questa rappresentazione di oggetto. I server devono convertire gli schemi riconosciuti nell'ultimo valore interno e possono rifiutare i valori non riconosciuti. Ulteriori informazioni: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources .
<code>kind</code> Stringa	Il tipo è un valore stringa che rappresenta la risorsa REST rappresentata da questo oggetto. I server possono dedurre questo dall'endpoint a cui il client inoltra le richieste. Non può essere aggiornato. In CamelCase. Ulteriori informazioni: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-tipi .
<code>metadata</code>	
<code>spec</code> “Specifica QueueManager” a pagina 44	Lo stato desiderato del QueueManager.
<code>status</code> “QueueManagerStatus” a pagina 45	Lo stato osservato di QueueManager.

Disponibilità

Impostazioni di disponibilità per il gestore code, ad esempio se utilizzare o meno una coppia active - standby.

Viene visualizzato in:

- [“Configurazione di QueueManager” a pagina 42](#)

Campo	Descrizione
type Stringa	Il tipo di disponibilità da utilizzare. Utilizza "SingleInstance" per un singolo pod, che verrà riavviato automaticamente (in alcuni casi) da Kubernetes. Utilizza "MultiInstance" per una coppia di pod, uno dei quali è il gestore code "attivo" e l'altro è uno standby. Vedi Alta disponibilità per IBM MQ nei contenitori nella versione più recente di IBM MQ .

Licenza

Impostazioni che controllano l'accettazione della licenza e quali metriche di licenza utilizzare.

Viene visualizzato in:

- [“Specifica QueueManager” a pagina 44](#)

Campo	Descrizione
use Stringa	Impostazione che controlla il modo in cui verrà utilizzato il software, dove la licenza supporta più utilizzi. Consultare https://ibm.biz/BdqvCF per i valori validi.
accept booleano	Indica se si accetta o meno la licenza associata a questo software (obbligatorio).
license Stringa	L'identificativo della licenza che si sta accettando. Deve essere l'identificativo di licenza corretto per la versione di MQ che si utilizza. Consultare https://ibm.biz/BdqvCF per i valori validi.
metric Stringa	Impostazione che specifica quale metrica di licenza utilizzare. Ad esempio, "ProcessorValueUnit", "VirtualProcessorCore" o "ManagedVirtualServer".

Limiti

QueueManagerResourceList definisce le impostazioni di CPU & memoria.

Viene visualizzato in:

- [“Risorse” a pagina 48](#)

Campo	Descrizione
cpu	
memory	

Riferimento LocalObject

LocalObjectIl riferimento contiene informazioni sufficienti per individuare l'oggetto di riferimento all'interno dello stesso spazio dei nomi.

Viene visualizzato in:

- [“Specifica QueueManager” a pagina 44](#)

Campo	Descrizione
name Stringa	Nome del referente. Ulteriori informazioni: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names TODO: aggiungere altri campi utili. apiVersion, tipo, uid?.

PKI

Impostazioni Public Key Infrastructure, per la definizione di chiavi e certificati da utilizzare con TLS (Transport Layer Security) o AMS (Advanced Message Security) MQ Advanced Message Security .

Viene visualizzato in:

- [“Specifica QueueManager” a pagina 44](#)

Campo	Descrizione
Array keys “Origine PKIS” a pagina 42	Chiavi private da aggiungere al repository delle chiavi del gestore code.
Array trust “Origine PKIS” a pagina 42	Certificati da aggiungere al repository delle chiavi del gestore code.

Origine PKIS

PKISource definisce un'origine delle informazioni Public Key Infrastructure, come chiavi o certificati.

Viene visualizzato in:

- [“PKI” a pagina 42](#)

Campo	Descrizione
name Stringa	Il nome viene utilizzato come etichetta per la chiave o il certificato. Deve essere una stringa alfanumerica minuscola.
secret “Segreto” a pagina 48	Fornisci una chiave utilizzando un segreto Kubernetes .

Configurazione di QueueManager

La configurazione QueueManager definisce le impostazioni per il contenitore Gestore code e il gestore code sottostante.

Viene visualizzato in:

- [“Specifica QueueManager” a pagina 44](#)

Campo	Descrizione
logFormat Stringa	Quale formato di log utilizzare per questo contenitore. Utilizza "JSON" per i log formattati JSON dal contenitore. Utilizzare "Di base" per i messaggi in formato testo.
metrics “Metriche QueueManager” a pagina 43	Impostazioni per le metriche in stile Prometheus.
readinessProbe “QueueManagerReadinessProbe” a pagina 44	Impostazioni che controllano il probe di disponibilità.
resources “Risorse” a pagina 48	Impostazioni che controllano i requisiti delle risorse.

Campo	Descrizione
storage “Memoria QueueManager” a pagina 47	Impostazioni di archiviazione per controllare l'utilizzo da parte del gestore code di volumi persistenti e classi di archiviazione.
availability “Disponibilità” a pagina 41	Impostazioni di disponibilità per il gestore code, ad esempio se utilizzare o meno una coppia active - standby.
imagePullPolicy Stringa	Impostazione che controlla quando il kubelet tenta di estrarre l'immagine specificata.
livenessProbe “QueueManagerLivenessProbe” a pagina 43	Impostazioni che controllano il probe di attività.
debug booleano	Indica se registrare o meno i messaggi di debug dal codice specifico del contenitore al log del contenitore.
image Stringa	L'immagine contenitore che verrà utilizzata.
name Stringa	Nome del gestore code MQ sottostante, se diverso da metadata.name. Utilizzare questo campo se si desidera un nome gestore code non conforme alle regole Kubernetes per i nomi (ad esempio, un nome che include lettere maiuscole).

QueueManagerLivenessProbe

Impostazioni che controllano il probe di attività.

Viene visualizzato in:

- “Configurazione di QueueManager” a pagina 42

Campo	Descrizione
failureThreshold intero	Numero minimo di errori consecutivi perché l'analisi venga considerata non riuscita dopo l'esito positivo.
initialDelaySeconds intero	Numero di secondi dopo che il contenitore è stato avviato prima dell'avvio delle analisi di attività. Ulteriori informazioni: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
periodSeconds intero	La frequenza (in secondi) con cui eseguire l'analisi.
successThreshold intero	Numero minimo di esiti positivi consecutivi perché il probe venga considerato riuscito dopo l'errore.
timeoutSeconds intero	Numero di secondi dopo i quali l'analisi va in timeout. Ulteriori informazioni: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

Metriche QueueManager

Impostazioni per le metriche in stile Prometheus.

Viene visualizzato in:

- “Configurazione di QueueManager” a pagina 42

Campo	Descrizione
enabled booleano	Indica se abilitare o meno un endpoint di metriche compatibile con Prometheus.

QueueManagerOptionalVolume

Dettagli PersistentVolume per log di recupero di MQ . Obbligatorio quando si utilizza il gestore code a più istanze.

Viene visualizzato in:

- [“Memoria QueueManager” a pagina 47](#)

Campo	Descrizione
class Stringa	Classe di memoria da utilizzare per questo volume. Valido solo se "type" è "persistent-claim".
enabled booleano	Indica se questo volume deve essere abilitato o meno come volume separato o se deve essere posizionato sul volume "queueManager" predefinito.
size Stringa	Dimensione del PersistentVolume da passare a Kubernetes. Valido solo se "type" è "persistent-claim".
sizeLimit Stringa	Limite di dimensione quando si utilizza un volume "effimero". I file sono ancora scritti in una directory temporanea, quindi è possibile utilizzare questa opzione per limitare la dimensione. Valido solo se type è "effimero".
type Stringa	Tipo di volume da utilizzare. Scegliere ephemeral per creare un volume "emptyDir" non persistente oppure persistent-claim per utilizzare un volume persistente.

QueueManagerReadinessProbe

Impostazioni che controllano il probe di disponibilità.

Viene visualizzato in:

- [“Configurazione di QueueManager” a pagina 42](#)

Campo	Descrizione
failureThreshold intero	Numero minimo di errori consecutivi perché l'analisi venga considerata non riuscita dopo l'esito positivo.
initialDelaySeconds intero	Numero di secondi dopo che il contenitore è stato avviato prima dell'avvio delle analisi di attività. Ulteriori informazioni: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
periodSeconds intero	La frequenza (in secondi) con cui eseguire l'analisi.
successThreshold intero	Numero minimo di esiti positivi consecutivi perché il probe venga considerato riuscito dopo l'errore.
timeoutSeconds intero	Numero di secondi dopo i quali l'analisi va in timeout. Ulteriori informazioni: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

Specifica QueueManager

Lo stato desiderato del QueueManager.

Viene visualizzato in:

- [“QueueManager” a pagina 40](#)

Campo	Descrizione
<code>license</code> “Licenza” a pagina 41	Impostazioni che controllano l'accettazione della licenza e quali metriche di licenza utilizzare.
<code>pki</code> “PKI” a pagina 42	Impostazioni Public Key Infrastructure, per la definizione di chiavi e certificati da utilizzare con TLS (Transport Layer Security) o AMS (Advanced Message Security) MQ Advanced Message Security .
<code>queueManager</code> “Configurazione di QueueManager” a pagina 42	La configurazione QueueManager definisce le impostazioni per il contenitore Gestore code e il gestore code sottostante.
<code>securityContext</code> “SecurityContext” a pagina 48	Impostazioni di protezione da aggiungere al securityContext di Queue Manager Pod.
<code>tracing</code> “TracingConfig” a pagina 50	Impostazioni per l'integrazione di traccia con il dashboard Operazioni Cloud Pak for Integration .
<code>version</code> Stringa	Impostazione che controlla la versione di MQ che verrà utilizzata (obbligatorio). Ad esempio: "9.1.5.0-r2" specificherebbe MQ versione 9.1.5.0, utilizzando la seconda revisione dell'immagine del contenitore. Le correzioni specifiche del contenitore vengono spesso applicate nelle revisioni, come le correzioni all'immagine di base.
<code>affinity</code>	Regole di affinità Kubernetes standard. Per ulteriori informazioni, consultare https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#affinity-v1-core .
<code>Array imagePullSecrets</code> “Riferimento LocalObject” a pagina 41	Un elenco facoltativo di riferimenti ai segreti nello stesso spazio dei nomi da utilizzare per il pull delle immagini utilizzate da questo QueueManager. Se specificato, questi segreti verranno passati alle singole implementazioni del programma di estrattore per utilizzarli. Ad esempio, nel caso di docker, vengono rispettati solo i segreti di tipo DockerConfig . Per ulteriori informazioni, vedi https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod .
<code>template</code> “Modello” a pagina 49	Templating avanzato per risorse Kubernetes . Il template consente agli utenti di sovrascrivere il modo in cui IBM MQ genera le risorse Kubernetes sottostanti, come StatefulSet, Pods e Services. Questo è solo per gli utenti avanzati, poiché ha il potenziale di interrompere il normale funzionamento di MQ se utilizzato in modo non corretto. Tutti i valori specificati altrove nella risorsa QueueManager verranno sovrascritti dalle impostazioni nel modello.
<code>terminationGracePeriod</code> Seconds intero	Durata facoltativa in secondi di cui il pod ha bisogno per terminare correttamente. Il valore deve essere un numero intero non negativo. Il valore zero indica l'eliminazione immediata. L'ora di destinazione in cui si tenta di terminare il gestore code, eseguendo l'escalation delle fasi di disconnessione dell'applicazione. Le attività essenziali di manutenzione del gestore code vengono interrotte, se necessario.
<code>web</code> “Configurazione WebServer” a pagina 51	Impostazioni per il server web MQ .

QueueManagerStatus

Lo stato osservato di QueueManager.

Viene visualizzato in:

- [“QueueManager” a pagina 40](#)

Campo	Descrizione
Array endpoints “ QueueManagerStatusEndpoint ” a pagina 46	Informazioni sugli endpoint che questo gestore code sta esponendo, come gli endpoint API o UI.
name Stringa	Il nome del gestore code.
versions “ QueueManagerStatusVersion ” a pagina 46	Versione di MQ utilizzata e altre versioni disponibili da IBM Entitled Registry.
adminUiUrl Stringa	URL per l'interfaccia utente Admin.
Array conditions “ QueueManagerStatusCondition ” a pagina 46	Le condizioni rappresentano le ultime osservazioni disponibili dello stato del gestore code.

QueueManagerStatusCondition

QueueManagerStatusCondition definisce le condizioni del Gestore code.

Viene visualizzato in:

- “[QueueManagerStatus](#)” a pagina 45

Campo	Descrizione
message Stringa	Messaggio leggibile che indica i dettagli sull'ultima transizione.
type Stringa	Tipo di condizione.
lastTransitionTime Stringa	L'ultima volta in cui la condizione è passata da un stato all'altro.

QueueManagerStatusEndpoint

QueueManagerStatusEndpoint definisce gli endpoint per QueueManager.

Viene visualizzato in:

- “[QueueManagerStatus](#)” a pagina 45

Campo	Descrizione
name Stringa	Nome dell'endpoint.
type Stringa	Il tipo di endpoint, ad esempio 'UI' per un endpoint UI, 'API' per un endpoint API, 'OpenAPI' per la documentazione API.
uri Stringa	URI per l'endpoint.

QueueManagerStatusVersion

Versione di MQ utilizzata e altre versioni disponibili da IBM Entitled Registry.

Viene visualizzato in:

- “[QueueManagerStatus](#)” a pagina 45

Campo	Descrizione
available “ QueueManagerStatusVersion Disponibile ” a pagina 47	Altre versioni di MQ disponibili da IBM Entitled Registry.

Campo	Descrizione
reconciled Stringa	La specifica versione di IBM MQ utilizzata. Se viene specificata un'immagine personalizzata, potrebbe non corrispondere alla versione di MQ effettivamente utilizzata.

QueueManagerStatusVersionDisponibile

Altre versioni di MQ disponibili da IBM Entitled Registry.

Viene visualizzato in:

- [“QueueManagerStatusVersion” a pagina 46](#)

Campo	Descrizione
channels schiera	I canali disponibili per l'aggiornamento automatico della versione di MQ .
Array versions “Versioni” a pagina 51	Versioni specifiche di MQ disponibili.

Memoria QueueManager

Impostazioni di archiviazione per controllare l'utilizzo da parte del gestore code di volumi persistenti e classi di archiviazione.

Viene visualizzato in:

- [“Configurazione di QueueManager” a pagina 42](#)

Campo	Descrizione
persistedData “QueueManagerOptionalVolume” a pagina 44	Dettagli PersistentVolume per i dati persistenti di MQ , inclusi la configurazione, code e messaggi. Obbligatorio quando si utilizza il gestore code a più istanze.
queueManager “Volume QueueManager” a pagina 47	Il valore predefinito PersistentVolume per tutti i dati normalmente in /var/mqm. Conterrà tutti i dati persistenti e i log di recupero, se non vengono specificati altri volumi.
recoveryLogs “QueueManagerOptionalVolume” a pagina 44	Dettagli PersistentVolume per log di recupero di MQ . Obbligatorio quando si utilizza il gestore code a più istanze.

Volume QueueManager

Il valore predefinito PersistentVolume per tutti i dati normalmente in /var/mqm. Conterrà tutti i dati persistenti e i log di recupero, se non vengono specificati altri volumi.

Viene visualizzato in:

- [“Memoria QueueManager” a pagina 47](#)

Campo	Descrizione
class Stringa	Classe di memoria da utilizzare per questo volume. Valido solo se "type" è "persistent-claim".
size Stringa	Dimensione del PersistentVolume da passare a Kubernetes. Valido solo se "type" è "persistent-claim".
sizeLimit Stringa	Limite di dimensione quando si utilizza un volume "effimero". I file sono ancora scritti in una directory temporanea, quindi è possibile utilizzare questa opzione per limitare la dimensione. Valido solo se type è "effimero".

Campo	Descrizione
type Stringa	Tipo di volume da utilizzare. Scegliere ephemeral per creare un volume "emptyDir" non persistente oppure persistent-claim per utilizzare un volume persistente.

Richieste

QueueManagerResourceList definisce le impostazioni di CPU & memoria.

Viene visualizzato in:

- [“Risorse” a pagina 48](#)

Campo	Descrizione
memory	
cpu	

Risorse

Impostazioni che controllano i requisiti delle risorse.

Viene visualizzato in:

- [“Configurazione di QueueManager” a pagina 42](#)

Campo	Descrizione
limits “Limiti” a pagina 41	QueueManagerResourceList definisce le impostazioni di CPU & memoria.
requests “Richieste” a pagina 48	QueueManagerResourceList definisce le impostazioni di CPU & memoria.

Segreto

Fornisci una chiave utilizzando un segreto Kubernetes .

Viene visualizzato in:

- [“Origine PKIS” a pagina 42](#)

Campo	Descrizione
items schiera	Le chiavi all'interno del segreto Kubernetes che devono essere aggiunte al contenitore Gestore code.
secretName Stringa	Il nome del segreto Kubernetes .

SecurityContext

Impostazioni di protezione da aggiungere al securityContextdi Queue Manager Pod.

Viene visualizzato in:

- [“Specifica QueueManager” a pagina 44](#)

Campo	Descrizione
supplementalGroups schiera	Un elenco di gruppi applicati al primo processo eseguito in ogni contenitore, in aggiunta al GID primario del contenitore. Se non specificato, nessun gruppo verrà aggiunto ad alcun contenitore.

Campo	Descrizione
fsGroup intero	Un gruppo supplementare speciale che si applica a tutti i contenitori in un pod. Alcuni tipi di volume consentono a Kubelet di modificare la proprietà di tale volume in modo che appartenga al pod: 1. Il GID proprietario sarà il FSGroup 2. Il bit setgid è impostato (i nuovi file creati nel volume saranno di proprietà di FSGroup) 3. I bit di autorizzazione sono OR ' d con rw-rw ---- Se non impostato, Kubelet non modificherà la proprietà e le autorizzazioni di alcun volume.
initVolumeAsRoot booleano	Ciò influenza il securityContext utilizzato dal contenitore che inizializza il PersistentVolume. Impostare su "true" se si utilizza un provider di memoria che richiede di essere l'utente root per accedere ai volumi di cui è stato appena eseguito il provisioning. L'impostazione su "true" influenza l'oggetto SCC (Security Context Constraint) che è possibile utilizzare e il gestore code potrebbe non riuscire ad avviarsi se non si è autorizzati ad utilizzare un SCC che consente l'utente root. Per ulteriori informazioni, consultare https://docs.openshift.com/container-platform/latest/authentication/managing-security-context-constraints.html .

Modello

Templating avanzato per risorse Kubernetes . Il template consente agli utenti di sovrascrivere il modo in cui IBM MQ genera le risorse Kubernetes sottostanti, come StatefulSet, Pods e Services. Questo è solo per gli utenti avanzati, poiché ha il potenziale di interrompere il normale funzionamento di MQ se utilizzato in modo non corretto. Tutti i valori specificati altrove nella risorsa QueueManager verranno sovrascritti dalle impostazioni nel modello.

Viene visualizzato in:

- [“Specifica QueueManager” a pagina 44](#)

Campo	Descrizione
pod	Sovrascritture per il template utilizzato per il pod. Vedere https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#podspec-v1-core .

TracingAgent

Solo in Cloud Pak for Integration , è possibile configurare le impostazioni per l'agent di traccia facoltativo.

Viene visualizzato in:

- [“TracingConfig” a pagina 50](#)

Campo	Descrizione
image Stringa	L'immagine contenitore che verrà utilizzata.
imagePullPolicy Stringa	Impostazione che controlla quando il kubelet tenta di estrarre l'immagine specificata.
livenessProbe “TracingProbe” a pagina 50	Impostazioni che controllano il probe di attività.
readinessProbe “TracingProbe” a pagina 50	Impostazioni che controllano il probe di disponibilità.

TracingCollector

Solo in Cloud Pak for Integration , è possibile configurare le impostazioni per il Tracing Collector facoltativo.

Viene visualizzato in:

- [“TracingConfig” a pagina 50](#)

Campo	Descrizione
image Stringa	L'immagine contenitore che verrà utilizzata.
imagePullPolicy Stringa	Impostazione che controlla quando il kubelet tenta di estrarre l'immagine specificata.
livenessProbe “TracingProbe” a pagina 50	Impostazioni che controllano il probe di attività.
readinessProbe “TracingProbe” a pagina 50	Impostazioni che controllano il probe di disponibilità.

TracingConfig

Impostazioni per l'integrazione di traccia con il dashboard Operazioni Cloud Pak for Integration .

Viene visualizzato in:

- [“Specifica QueueManager” a pagina 44](#)

Campo	Descrizione
agent “TracingAgent” a pagina 49	Solo in Cloud Pak for Integration , è possibile configurare le impostazioni per l'agent di traccia facoltativo.
collector “TracingCollector” a pagina 50	Solo in Cloud Pak for Integration , è possibile configurare le impostazioni per il Tracing Collector facoltativo.
enabled booleano	Indica se abilitare o meno l'integrazione con il dashboard Operazioni di Cloud Pak for Integration , tramite la traccia.
namespace Stringa	Spazio dei nomi in cui è installato il dashboard Operazioni di Cloud Pak for Integration .

TracingProbe

Impostazioni che controllano il probe di disponibilità.

Viene visualizzato in:

- [“TracingCollector” a pagina 50](#)

Campo	Descrizione
failureThreshold intero	Numero minimo di errori consecutivi perché l'analisi venga considerata non riuscita dopo l'esito positivo.
initialDelaySeconds intero	Numero di secondi dopo che il contenitore è stato avviato prima dell'avvio delle analisi di attività. Ulteriori informazioni: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .
periodSeconds intero	La frequenza (in secondi) con cui eseguire l'analisi.
successThreshold intero	Numero minimo di esiti positivi consecutivi perché il probe venga considerato riuscito dopo l'errore.

Campo	Descrizione
timeoutSeconds intero	Numero di secondi dopo i quali l'analisi va in timeout. Ulteriori informazioni: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes .

Versioni

QueueManagerStatusVersion definisce una versione di MQ.

Viene visualizzato in:

- “QueueManagerStatusVersionDisponibile” a pagina 47

Campo	Descrizione
name Stringa	La versione "name" per questa versione di QueueManager. Questi sono valori validi per il campo spec.version.

Configurazione WebServer

Impostazioni per il server web MQ .

Viene visualizzato in:

- “Specifica QueueManager” a pagina 44

Campo	Descrizione
enabled booleano	Indica se abilitare o meno il server Web.

Informazioni particolari

Queste informazioni sono state sviluppate per i prodotti ed i servizi offerti negli Stati Uniti.

IBM potrebbe non offrire i prodotti, i servizi o le funzioni descritti in questo documento in altri paesi. Consultare il rappresentante IBM locale per informazioni sui prodotti e sui servizi disponibili nel proprio paese. Ogni riferimento relativo a prodotti, programmi o servizi IBM non implica che solo quei prodotti, programmi o servizi IBM possano essere utilizzati. In sostituzione a quelli forniti da IBM possono essere usati prodotti, programmi o servizi funzionalmente equivalenti che non comportino la violazione dei diritti di proprietà intellettuale o di altri diritti dell'IBM. È comunque responsabilità dell'utente valutare e verificare la possibilità di utilizzare altri programmi e/o prodotti, fatta eccezione per quelli espressamente indicati dall'IBM.

IBM potrebbe disporre di applicazioni di brevetti o brevetti in corso relativi all'argomento descritto in questo documento. La fornitura di tale documento non concede alcuna licenza a tali brevetti. Chi desiderasse ricevere informazioni relative a licenze può rivolgersi per iscritto a:

Director of Commercial Relations
IBM Corporation
Schoenaicher Str. 220
D-7030 Boeblingen
U.S.A.

Per richieste di licenze relative ad informazioni double-byte (DBCS), contattare il Dipartimento di Proprietà Intellettuale IBM nel proprio paese o inviare richieste per iscritto a:

Intellectual Property Licensing
Legge sulla proprietà intellettuale e legale
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

Il seguente paragrafo non si applica al Regno Unito o a qualunque altro paese in cui tali dichiarazioni sono incompatibili con le norme locali: INTERNATIONAL BUSINESS MACHINES CORPORATION FORNISCE LA PRESENTE PUBBLICAZIONE "NELLO STATO IN CUI SI TROVA" SENZA GARANZIE DI ALCUN TIPO, ESPRESSE O IMPLICITE, IVI INCLUSE, A TITOLO DI ESEMPIO, GARANZIE IMPLICITE DI NON VIOLAZIONE, DI COMMERCIALIZZABILITÀ E DI IDONEITÀ PER UNO SCOPO PARTICOLARE. Alcuni stati non consentono la rinuncia a garanzie esplicite o implicite in determinate transazioni; quindi la presente dichiarazione potrebbe non essere applicabile.

Questa pubblicazione potrebbe contenere imprecisioni tecniche o errori tipografici. Le informazioni incluse in questo documento vengono modificate su base periodica; tali modifiche vengono incorporate nelle nuove edizioni della pubblicazione. IBM si riserva il diritto di apportare miglioramenti o modifiche al prodotto/i e/o al programma/i descritti nella pubblicazione in qualsiasi momento e senza preavviso.

Qualsiasi riferimento a siti Web non IBM contenuto nelle presenti informazioni è fornito per consultazione e non vuole in alcun modo promuovere i suddetti siti Web. I materiali presenti in tali siti Web non sono parte dei materiali per questo prodotto IBM e l'utilizzo di tali siti Web è a proprio rischio.

Tutti i commenti e i suggerimenti inviati potranno essere utilizzati liberamente da IBM e diventeranno esclusiva della stessa.

Coloro che detengono la licenza su questo programma e desiderano avere informazioni su di esso allo scopo di consentire (i) uno scambio di informazioni tra programmi indipendenti ed altri (compreso questo) e (ii) l'uso reciproco di tali informazioni, dovrebbero rivolgersi a:

IBM Corporation
Coordinatore interoperabilità software, Dipartimento 49XA
Autostrada 3605 52 N

Rochester, MN 55901
U.S.A.

Queste informazioni possono essere rese disponibili secondo condizioni contrattuali appropriate, compreso, in alcuni casi, il pagamento di un addebito.

Il programma su licenza descritto in queste informazioni e tutto il materiale su licenza disponibile per esso sono forniti da IBM in base ai termini dell' IBM Customer Agreement, IBM International Program License Agreement o qualsiasi altro accordo equivalente tra le parti.

Tutti i dati relativi alle prestazioni contenuti in questo documento sono stati determinati in un ambiente controllato. Pertanto, i risultati ottenuti in altri ambienti operativi possono variare in modo significativo. Alcune misurazioni potrebbero essere state fatte su sistemi a livello di sviluppo e non vi è alcuna garanzia che queste misurazioni saranno le stesse sui sistemi generalmente disponibili. Inoltre, alcune misurazioni potrebbero essere state stimate mediante estrapolazione. I risultati quindi possono variare. Gli utenti di questo documento dovrebbero verificare i dati applicabili per il loro ambiente specifico.

Le informazioni relative a prodotti non IBM provengono dai fornitori di tali prodotti, dagli annunci pubblicati o da altre fonti pubblicamente disponibili. IBM non ha verificato tali prodotti e, pertanto, non può garantirne l'accuratezza delle prestazioni. Eventuali commenti relativi alle prestazioni dei prodotti non IBM devono essere indirizzati ai fornitori di tali prodotti.

Tutte le dichiarazioni riguardanti la direzione o l'intento futuro di IBM sono soggette a modifica o ritiro senza preavviso e rappresentano solo scopi e obiettivi.

Questa pubblicazione contiene esempi di dati e prospetti utilizzati quotidianamente nelle operazioni aziendali, Per illustrarle nel modo più completo possibile, gli esempi includono i nomi di individui, società, marchi e prodotti. Tutti questi nomi sono fittizi e qualsiasi somiglianza con nomi ed indirizzi adoperati da imprese realmente esistenti sono una mera coincidenza.

LICENZA SUL COPYRIGHT:

Queste informazioni contengono programmi applicativi di esempio in lingua originale, che illustrano le tecniche di programmazione su diverse piattaforme operative. È possibile copiare, modificare e distribuire questi programmi di esempio sotto qualsiasi forma senza alcun pagamento alla IBM, allo scopo di sviluppare, utilizzare, commercializzare o distribuire i programmi applicativi in conformità alle API (application programming interface) a seconda della piattaforma operativa per cui i programmi di esempio sono stati scritti. Questi esempi non sono stati testati approfonditamente tenendo conto di tutte le condizioni possibili. IBM, quindi, non può garantire o sottintendere l'affidabilità, l'utilità o il funzionamento di questi programmi.

Se si sta visualizzando queste informazioni in formato elettronico, le fotografie e le illustrazioni a colori potrebbero non apparire.

Informazioni sull'interfaccia di programmazione

Le informazioni sull'interfaccia di programmazione, se fornite, consentono di creare software applicativo da utilizzare con questo programma.

Questo manuale contiene informazioni sulle interfacce di programmazione che consentono al cliente di scrivere programmi per ottenere i servizi di WebSphere MQ.

Queste informazioni, tuttavia, possono contenere diagnosi, modifica e regolazione delle informazioni. La diagnosi, la modifica e la regolazione delle informazioni vengono fornite per consentire il debug del software applicativo.

Importante: Non utilizzare queste informazioni di diagnosi, modifica e ottimizzazione come interfaccia di programmazione poiché sono soggette a modifica.

Marchi

IBM, il logo IBM , ibm.com, sono marchi di IBM Corporation, registrati in molte giurisdizioni nel mondo. Un elenco aggiornato dei marchi IBM è disponibile sul web in "Copyright and trademark

information"www.ibm.com/legal/copytrade.shtml. Altri nomi di prodotti e servizi potrebbero essere marchi di IBM o altre società.

Microsoft e Windows sono marchi di Microsoft Corporation negli Stati Uniti e/o in altri paesi.

UNIX è un marchio registrato di The Open Group negli Stati Uniti e/o in altri paesi.

Linux è un marchio registrato di Linus Torvalds negli Stati Uniti e/o in altri paesi.

Questo prodotto include il software sviluppato da Eclipse Project (<http://www.eclipse.org/>).

Java e tutti i marchi e i logo Java sono marchi registrati di Oracle e/o di società affiliate.



Numero parte:

(1P) P/N: