9.4

IBM MQ nei contenitori

Nota

Prima di utilizzare queste informazioni e il prodotto che supportano, leggere le informazioni in “Informazioni particolari” a pagina 175.

Questa edizione si applica alla versione 9 release 4 di IBM® MQ e a tutte le successive release e modifiche se non diversamente indicato nelle nuove edizioni.

Indice

IBM MQ nei contenitori e IBM Cloud Pak for Integration

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.

È possibile eseguire IBM MQ utilizzando IBM MQ Operator su Red Hat® OpenShift®. Questa operazione può essere eseguita utilizzando IBM Cloud Pak for Integration, IBM MQ Advanced o IBM MQ Advanced for Developers.

Puoi anche eseguire IBM MQ in un contenitore che crei da solo.

Per ulteriori informazioni su IBM MQ Operator, consultare i seguenti link.

Informazioni su IBM MQ nei contenitori

Informazioni introduttive per aiutarti a iniziare a utilizzare IBM MQ nei contenitori.

I contenitori sono una tecnologia per consentire l'impacchettamento e l'isolamento del codice con il suo ambiente di runtime, che può essere eseguito in modo isolato da altri software sulla stessa infrastruttura. Ciò semplifica lo spostamento di un gestore code o di un'applicazione tra ambienti (ad esempio, sviluppo, test e produzione). I moderni orchestratori di contenitori, come Red Hat OpenShift Container Platform e Kubernetes , possono eseguire molti tipi di contenitori sulla stessa macchina, ognuno isolato l'uno dall'altro in termini di risorse, sicurezza e errori.

È possibile eseguire i gestori code IBM MQ o le applicazioni IBM MQ nei contenitori.

Informazioni correlate

Cosa sono i contenitori?

Cronologia delle release per IBM MQ Operator

Note:

- Per informazioni sugli operatori IBM MQ precedenti, consultare Release history for IBM MQ Operator nella documentazione IBM MQ 9.3 .
- Per informazioni sui futuri aggiornamenti di IBM MQ , consultare la pagina IBM MQ Correzioni consigliate e date di rilascio della manutenzione pianificata .

IBM MQ Operator 3.2.1

IBM Cloud Pak for Integration Versione
IBM Cloud Pak for Integration 16.1.0

Canale operatore
`v3.2-sc2`

Valori consentiti per `.spec.version`
9.4.0.0-r1

Valori consentiti per `.spec.version` durante la migrazione
9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3,9.3.3.2-r3 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1, 9.3.0.16-r2, 9.3.0.17-r1, 9.3.0.17-r2, 9.3.0.17-r3, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3, 9.3.3.3-r1, 9.3.3.3-r2, 9.3.4.0-r1, 9.3.4.1-r1, 9.3.5.0-r1, 9.3.5.0-r2, 9.3.5.1-r1, 9.3.5.1-r2

Red Hat OpenShift Container Platform versioni
OpenShift Container Platform 4.12 e successive.

IBM Cloud Pak foundational services versioni
Solo IBM Cloud Pak foundational services versione 4.6 .

Elementi modificati

- Risolve un problema su OpenShift Container Platform 4.12 in cui l'aggiornamento al Canale v3.2-sc2 potrebbe causare un comportamento imprevisto per gli utenti IBM Cloud Pak for Integration . Per ulteriori informazioni, vedi Aggiornamento da 2023.4 nella documentazione di IBM Cloud Pak for Integration .

IBM MQ Operator 3.2.0

IBM Cloud Pak for Integration Versione
IBM Cloud Pak for Integration 16.1.0

Canale operatore
`v3.2-sc2`

Valori consentiti per `.spec.version`
9.4.0.0-r1

Valori consentiti per `.spec.version` durante la migrazione
9.3.0.0-r1, 9.3.0.0-r2, 9.3.0.0-r3,9.3.3.2-r3 9.3.0.1-r1, 9.3.0.1-r2, 9.3.0.1-r3, 9.3.0.1-r4, 9.3.0.3-r1, 9.3.0.4-r1, 9.3.0.4-r2, 9.3.0.5-r1, 9.3.0.5-r2, 9.3.0.5-r3, 9.3.0.6-r1, 9.3.0.10-r1, 9.3.0.10-r2, 9.3.0.11-r1, 9.3.0.11-r2, 9.3.0.15-r1, 9.3.0.16-r1, 9.3.0.16-r2, 9.3.0.17-r1, 9.3.0.17-r2, 9.3.0.17-r3, 9.3.1.0-r1, 9.3.1.0-r2, 9.3.1.0-r3, 9.3.1.1-r1, 9.3.2.0-r1, 9.3.2.0-r2, 9.3.2.1-r1, 9.3.2.1-r2, 9.3.3.0-r1, 9.3.3.0-r2, 9.3.3.1-r1, 9.3.3.1-r2, 9.3.3.2-r1, 9.3.3.2-r2, 9.3.3.2-r3, 9.3.3.3-r1, 9.3.3.3-r2, 9.3.4.0-r1, 9.3.4.1-r1, 9.3.5.0-r1, 9.3.5.0-r2, 9.3.5.1-r1, 9.3.5.1-r2

Red Hat OpenShift Container Platform versioni
OpenShift Container Platform 4.12 e successive.

IBM Cloud Pak foundational services versioni
Solo IBM Cloud Pak foundational services versione 4.6 .

Novità

- “Espansione dei volumi persistenti” a pagina 98 è ora supportata.
- I gestori code possono ora essere arrestati aggiungendo l'annotazione `mq.ibm.com/stop` e impostandola su `true`. Vedi“Arresto di un gestore code (mq.ibm.com/stop)” a pagina 102

  Note:

  - Un gestore code arrestato ha il campo *.replicas* nel relativo `StatefulSet` impostato su 0.
  - Poiché IBM MQ Operator ora gestisce attivamente il campo *.replicas* in `StatefulSet`, se si modifica questo campo viene immediatamente ripristinato dall'operatore.
  - Le versioni precedenti di IBM MQ immettono uno stato 'Non riuscito' se si modifica il campo *.replicas* , ma si conserva ancora il valore modificato. Se le tue procedure operative esistenti si basano su questo comportamento, da IBM MQ 9.4 devi utilizzare l'annotazione `mq.ibm.com/stop` .

Elementi modificati

- Sono ora supportate le release con numero dispari di Red Hat OpenShift Container Platform .
- IBM MQ L'immagine di catalogo è stata spostata nel formato di catalogo basato su file dal formato database SQLite .
- Basato su Red Hat Universal Base Image 9.4-949.1716471857. **Nota:** UBI 9 ha una certificazione FIPS 140-3 in sospeso. UBI 9 non è supportato sull'architettura di Power 8.
- Le vulnerabilità risolte sono descritte in questo Bollettino sulla sicurezza.

OpenShift CP4I Cronologia delle release per le immagini del contenitore del gestore code da utilizzare con IBM MQ Operator

Nota: Per informazioni sulle immagini del contenitore del gestore code precedenti, consultare Release history for IBM MQ Operator nella documentazione IBM MQ 9.3 .

9.4.0.0-r1

CD CP4I-SC2

Versione operatore richiesta
3.2.0 o superiore

Architetture supportate
amd64, s390x, ppc64le

Immagini

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

Novità

- Novità in IBM MQ 9.4.0 for Multiplatforms - base e titolarità avanzata

Elementi modificati

- Cosa è cambiato in IBM MQ 9.4.0
- Deprecated Quando utilizzi IBM MQ Advanced for Developers, l'impostazione delle parole d'ordine per gli utenti `admin` e `app` tramite una variabile di ambiente è obsoleta. Utilizzare invece i segreti.
- È stato aggiunto un nuovo valore facoltativo `mqsc` per la variabile di ambiente *MQ_LOGGING_CONSOLE_SOURCE*. Questa opzione può essere utilizzata per riflettere i contenuti di `autocfgmqsc.LOG` nel log del contenitore.
- Basato su Red Hat Universal Base Image 9.4-949.1716471857. **Nota:** UBI 9 ha una certificazione FIPS 140-3 in sospeso. UBI 9 non è supportato sull'architettura di Power 8.

Pianificazione per IBM MQ nei contenitori

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 IBM MQ di base (vedi IBM MQ Technical overview) e con i concetti di base di Kubernetes/Red Hat OpenShift (vedi OpenShift Container Platform architecture).

Procedura

- “Scelta della modalità di utilizzo di IBM MQ nei contenitori” a pagina 8.
- “Supporto per IBM MQ nei contenitori” a pagina 8.
- “Pianificazione dell'archiviazione per IBM MQ Operator” a pagina 15.
- “Pianificazione dell'alta disponibilità per IBM MQ nei contenitori” a pagina 17.
- “Ripristino di emergenza per IBM MQ nei contenitori” a pagina 23.
- “Autenticazione e autorizzazione utente per IBM MQ nei contenitori” a pagina 23.

Scelta della modalità di utilizzo di IBM MQ nei contenitori

Ci sono diverse opzioni per l'utilizzo di IBM MQ nei contenitori: puoi scegliere di utilizzare IBM MQ Operator, che utilizza le immagini del contenitore preconfezionate, oppure puoi creare le tue immagini e il tuo codice di distribuzione.

Utilizzo di IBM MQ Operator

Se si sta pianificando la distribuzione su Red Hat OpenShift Container Platform, probabilmente si desidera utilizzare IBM MQ Operator.

IBM MQ Operator estende l'API Red Hat OpenShift Container Platform per aggiungere una nuova risorsa personalizzata `QueueManager` . L'operatore controlla le nuove definizioni del gestore code e le trasforma in risorse di basso livello necessarie, come le risorse `StatefulSet` e `Service` . Nel caso della HA nativa, l'operatore può anche eseguire l'aggiornamento progressivo complesso delle istanze del gestore code. Vedere “Considerazioni sull'esecuzione di un aggiornamento continuo di un gestore code HA nativo” a pagina 21

Alcune funzioni IBM MQ non sono supportate quando si utilizza IBM MQ Operator. Consultare “Supporto per IBM MQ nei contenitori” a pagina 8 per dettagli su ciò che è supportato quando si utilizza IBM MQ Operator.

Creazione di immagini e codice di distribuzione 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 “Preparazione per IBM MQ creando la propria immagine contenitore” a pagina 54.

Consultare “Supporto per IBM MQ nei contenitori” a pagina 8 per i dettagli su cosa è supportato quando si crea la tua immagine e il tuo codice di distribuzione.

Riferimenti correlati

“Supporto per IBM MQ nei contenitori” a pagina 8
Non tutte le funzioni IBM MQ sono disponibili e supportate nello stesso modo nei contenitori.


 Supporto per IBM MQ nei contenitori

Non tutte le funzioni IBM MQ sono disponibili e supportate nello stesso modo nei contenitori.

Di seguito è riportata una tabella che mostra in dettaglio come le funzioni IBM MQ sono supportate con IBM MQ Operatoro quando crei i tuoi propri contenitori e il tuo codice di distribuzione.

Note:

- Le immagini del contenitore IBM MQ precompilate su IBM Container Registry (icr.io e cp.icr.io) sono supportate e idonee per le fix solo se utilizzate con IBM MQ Operator.
- Dal canale IBM MQ Operator v3.2, Long Term Support (LTS) viene ridenominato in Support Cycle 2 (SC2). Ciò è dovuto al fatto che l'unico LTS percorso disponibile per IBM MQ nei contenitori è il supporto di due anni sotto la titolarità IBM Cloud Pak for Integration e IBM Cloud Pak for Integration ha adottato il termine SC2. Ecco il quadro completo della titolarità:
  - Con la titolarità IBM MQ , IBM MQ Operator può distribuire solo immagini IBM MQ Continuous Delivery (CD).
  - Con la titolarità IBM Cloud Pak for Integration , IBM MQ Operator può distribuire immagini CD o SC2 (formerly LTS) .

Non è possibile "aggiornare" la licenza dell'immagine IBM MQ Advanced for Developers precostruita a una licenza diversa. IBM MQ Operator distribuirà immagini differenti, a seconda della licenza selezionata.

In questa tabella, si applicano i termini seguenti:

"Codice abilitazione contenitore"
Gli eseguibili **runmqserver**, **runmqintegrationserver**, **chkmqhealthy**, **chkmqready** e **chkmqstarted**. Questo codice viene fornito come esempio ed è supportato solo come parte dei contenitori preintegrati quando viene utilizzato con IBM MQ Operator.

	Utilizzo di IBM MQ Operator e di una licenza IBM Cloud Pak for Integration	**Utilizzo di IBM MQ Operator e di una licenza IBM MQ Advanced**	**Utilizzo di IBM MQ Operator e di una licenza IBM MQ Advanced for Developers**	**Immagine IBM MQ Advanced for Developers precostruita**	**Crea il tuo contenitore**
Piattaforma supportate	Supportato solo su Red Hat OpenShift Container Platform . Le release di Red Hat OpenShift Container Platform non vengono più supportate da IBM MQ una volta Red Hat arrestato il supporto. Per ulteriori dettagli, vedere “Supporto versione per IBM MQ Operator” a pagina 13.		Disponibile solo su Red Hat OpenShift Container Platform , ma non supportato.	Funziona su qualsiasi piattaforma Docker, containerd o cri-o, ma non è supportata. Consultare Requisiti di sistema per IBM MQ per i dettagli.	Qualsiasi piattaforma Docker, containerd o cri-o. Consultare Requisiti di sistema per IBM MQ per i dettagli. La HA nativa è supportata solo su Kubernetes o Red Hat OpenShift Container Platform. L'immagine del contenitore di esempio utilizza un Red Hat Universal Base Image (UBI), che include le librerie Linux® e i programmi di utilità utilizzati da IBM MQ. L'UBI è supportato da Red Hat quando viene eseguito su Red Hat OpenShift. Il *codice di abilitazione del contenitore* non è supportato.

	Utilizzo di IBM MQ Operator e di una licenza IBM Cloud Pak for Integration	Utilizzo di IBM MQ Operator e di una licenza IBM MQ Advanced	Utilizzo di IBM MQ Operator e di una licenza IBM MQ Advanced for Developers	Immagine IBM MQ Advanced for Developers precostruita	Crea il tuo contenitore
Architetture CPU	Supportato su `amd64` e `s390x` z /Linux. Supportato anche su sistemi `ppc64le` Power Systems versione 9 e successive. Nota che tutti i nodi nel cluster Red Hat OpenShift Container Platform devono utilizzare la stessa architettura CPU.		Disponibile su `amd64` e `s390x` z /Linux, ma non supportato. Disponibile anche su sistemi `ppc64le` Power Systems versione 9 e superiori, ma non supportato. Nota che tutti i nodi nel cluster Red Hat OpenShift Container Platform devono utilizzare la stessa architettura CPU.		Come per il software IBM MQ .
Durata del supporto	IBM Cloud Pak for Integration - Support Cycle 2 (formerly Long Term Support) o Continuous Delivery.[1] I gestori code e gli operatori CD sono supportati fino alla successiva release IBM Cloud Pak for Integration CD o CP4I-SC2 . I gestori code e gli operatori CP4I-SC2 sono supportati fino alla release IBM Cloud Pak for Integration CP4I-SC2 successiva, più un periodo di tolleranza per consentire l'aggiornamento.	Solo per il flusso Continuous Delivery , sia per i gestori code che per IBM MQ Operator. Ogni versione di IBM MQ Operator e del gestore code è supportata solo fino alla release CD successiva.	Non supportato		Come per il software IBM MQ . Consultare IBM MQ FAQ per il supporto a lungo termine e le release di Continuous Delivery. Il *codice di abilitazione del contenitore* non è supportato.

	Utilizzo di IBM MQ Operator e di una licenza IBM Cloud Pak for Integration	Utilizzo di IBM MQ Operator e di una licenza IBM MQ Advanced	Utilizzo di IBM MQ Operator e di una licenza IBM MQ Advanced for Developers	Immagine IBM MQ Advanced for Developers precostruita	Crea il tuo contenitore
Disponibilità delle correzioni di protezione	Correzioni periodiche disponibili come immagini del contenitore su IBM Container Registry				Le correzioni al software IBM MQ sono disponibili come software su Fix Central. Il *codice di abilitazione del contenitore* non è supportato.
Disponibilità fix temporanea	Le correzioni del gestore code sono disponibili come software ed è necessario creare un'immagine personalizzata. Le correzioni IBM MQ Operator non sono disponibili come correzioni temporanee.		Nessuna fix temporanea disponibile.		Le fix per il software IBM MQ sono disponibili come software su Fix Central o tramite il supporto IBM . Il *codice di attivazione contenitore* non è supportato.
Funzione: Advanced Message Security	Supportato. Tenere presente che non è facile utilizzare la codifica lato server, perché IBM MQ Operator non consente direttamente di specificare il proprio keystore per Advanced Message Security.		Disponibile ma non supportato.		Supportato per il software IBM MQ , ma nessun esempio disponibile.
Funzione: Managed File Transfer	Non disponibile e non supportato. Tuttavia, è possibile utilizzare IBM MQ Operator per fornire uno o più gestori code Coordinamento, Comando o Agent.			Non disponibile e non supportato.	Supportato come da software IBM MQ , con esempio per l'agent.

[1] Il IBM MQ Operator è supportato come release IBM MQ CD o come release IBM Cloud Pak for Integration - Support Cycle 2 (formerly Long Term Support) :

- Le immagini del contenitore IBM MQ 9.4.0.x distribuite con IBM MQ Operator 3.2.x, quando utilizzate come parte di IBM Cloud Pak for Integration 16.1.0, sono idonee per il supporto CP4I-LTS . L'ultima release Support Cycle 2 (SC2) di IBM MQ Operator è 3.2.1e l'ultima SC2 immagine del contenitore è 9.4.0.0-r1.
- Le immagini del contenitore IBM MQ 9.4.0.x distribuite con IBM MQ Operator 3.2.x, quando utilizzate come parte di IBM Cloud Pak for Integration 16.1.0, sono idonee per il supporto CD . L'ultima release Continuous Delivery (CD) di IBM MQ Operator è 3.2.1e l'ultima CD immagine del contenitore è 9.4.0.0-r1.

	Utilizzo di IBM MQ Operator e di una licenza IBM Cloud Pak for Integration	Utilizzo di IBM MQ Operator e di una licenza IBM MQ Advanced	Utilizzo di IBM MQ Operator e di una licenza IBM MQ Advanced for Developers	Immagine IBM MQ Advanced for Developers precostruita	Crea il tuo contenitore
Funzione: MQTT	Non disponibile e non supportato.				Supportato per il software IBM MQ , ma nessun esempio disponibile.
Funzione: AMQP	Non disponibile e non supportato.				Supportato per il software IBM MQ , ma nessun esempio disponibile.
Funzione: REST API	Disponibile e supportato.				Disponibile e supportato secondo il software IBM MQ .
Funzione: gestori code di dati replica	Non disponibile e non supportato. I gestori code di dati replicati (RDQM) sono strettamente associati al kernel Linux e non sono supportati nei contenitori.				
Funzione: HA nativa	Disponibile e supportato.		Disponibile ma non supportato.		Disponibile solo su Kubernetes e Red Hat OpenShift Container Platform. Supportato secondo il software IBM MQ .
Funzione: gestori code a più istanze	Disponibile e supportato.		Disponibile ma non supportato.		Disponibile e supportato secondo il software IBM MQ .
Funzione: tipi di log di ripristino	Registrazione circolare o solo log replicati. La registrazione lineare non è supportata.				Disponibile e supportato secondo il software IBM MQ . È necessario configurare le opzioni **crtmqm** .

	Utilizzo di IBM MQ Operator e di una licenza IBM Cloud Pak for Integration	Utilizzo di IBM MQ Operator e di una licenza IBM MQ Advanced	Utilizzo di IBM MQ Operator e di una licenza IBM MQ Advanced for Developers	Immagine IBM MQ Advanced for Developers precostruita	Crea il tuo contenitore
Funzione: specifica delle opzioni della linea di comando personalizzata per `crtmqdir`, `crtmqm`, `strmqm` e `endmqm`	Non disponibile e non supportato. La maggior parte delle opzioni possono essere configurate utilizzando un file INI, ma alcune non possono essere configurate, ad esempio l'utilizzo della registrazione lineare.				Facoltativo, a seconda di come implementi il tuo *codice di abilitazione del contenitore*.
Funzione: utenti del sistema operativo	Non disponibile e non supportato.				Possibile e supportato come per il software IBM MQ , se si installa IBM MQ utilizzando RPM, ma non è disponibile alcun esempio. Non consigliato a causa del rischio di sicurezza.

Nota: La frase "supported as per IBM MQ software" significa che il supporto tecnico IBM è limitato al software IBM MQ principale in esecuzione all'interno del contenitore.

Concetti correlati

Domande frequenti di IBM MQ per le release Long Term Support e Continuous Delivery

Riferimenti correlati

IBM Cloud Pak for Integration Software Support Lifecycle Addendum

OpenShift CP4I CD CP4I-SC2 Supporto versione per IBM MQ Operator

Un'associazione tra le versioni supportate di IBM MQ, OpenShift Container Platform e IBM Cloud Pak for Integration.

- “Versioni IBM MQ disponibili” a pagina 14
- “Versioni Red Hat OpenShift Container Platform compatibili” a pagina 14
- “IBM Cloud Pak for Integration versioni” a pagina 14
- “Versioni IBM MQ disponibili in operatori meno recenti” a pagina 14
- “Versioni OpenShift Container Platform compatibili per gli operatori meno recenti” a pagina 14

Versioni IBM MQ disponibili

Canale operatore	Versione dell'operatore	IBM MQ versioni						
		9.4.0	9.3.5	9.3.4	9.3.3	9.3.2	9.3.1	9.3.0
v32-sc2	3.2	CD e SC2	DEP	DEP	DEP	DEP	DEP	MIG

Chiave:

- CD: il supporto Continuous Delivery è disponibile.
- SC2: IBM Cloud Pak for Integration - Support Cycle 2 (formerly Long Term Support) è disponibile.
- MIG: disponibile solo durante la migrazione da un operando IBM Cloud Pak for Integration - Support Cycle 2 (formerly Long Term Support) ad uno Continuous Delivery .
- DEP: Deprecated Obsoleto. Poiché le release IBM MQ non sono più supportate, potrebbero essere ancora configurabili nell'operatore, ma non sono più idonee per il supporto e potrebbero essere rimosse nelle release future.

Consultare “Cronologia delle release per IBM MQ Operator” a pagina 5 per i dettagli completi di ciascuna versione, incluse le funzioni dettagliate, le modifiche e le correzioni in ogni versione.

Versioni Red Hat OpenShift Container Platform compatibili

Canale operatore	Versione dell'operatore	OpenShift Container Platform versioni[2]		
		4.15	4.14	4.12
v3.2-sc2	3.2.0 e versioni successive	SC2	SC2	SC2

Chiave:

- CD: il supporto Continuous Delivery è disponibile.
- SC2: IBM Cloud Pak for Integration - Support Cycle 2 (formerly Long Term Support) è disponibile.
- EOS: non più supportato. Eseguire la migrazione a una versione OpenShift Container Platform successiva.

IBM Cloud Pak for Integration versioni

Supportato per l'utilizzo come parte di IBM Cloud Pak for Integration versione 16.1.0o indipendentemente:

- IBM MQ Operator 3.2.x

Versioni IBM MQ disponibili in operatori meno recenti

Vedi Available IBM MQ versions nella documentazione di IBM MQ 9.3 .

Versioni OpenShift Container Platform compatibili per gli operatori meno recenti

Consultare Compatible OpenShift Container Platform versions nella documentazione IBM MQ 9.3 .

OpenShift MQ Adv. Modifica delle risorse create da IBM MQ Operator

IBM MQ Operator riconcilia una risorsa personalizzata QueueManager creando e gestendo le risorse Kubernetes native. Queste risorse gestite **non devono** essere modificate direttamente.

[2] Le versioni di OpenShift Container Platform sono soggette alle proprie date di supporto. Per ulteriori informazioni, consultare OpenShift Container Platform Politica del ciclo di vita .

Generalmente, è possibile determinare se una risorsa è di proprietà di un'altra risorsa di livello superiore, esaminando `ownerReferences`. Ad esempio, i seguenti metadati presi da un `StatefulSet` mostrano che appartengono alla risorsa `QueueManager` "qm1":

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

Notare che non tutte le risorse hanno questi metadati.

È responsabilità di IBM MQ Operator gestire le risorse sottostanti, come `StatefulSet`, `Service` e `Route`. Se modifichi una di queste risorse sottostanti, IBM MQ Operator le modificherà e potresti riscontrare un tempo di inattività se tale modifica richiede un aggiornamento progressivo.

La maggior parte delle impostazioni importanti per i gestori code sono disponibili sulla risorsa `QueueManager` . Tuttavia, se si ritiene che sia necessario il controllo completo delle risorse sottostanti, sono disponibili alcune opzioni:

- Se devi sovrascrivere le impostazioni sul pod creato da IBM MQ Operator, puoi aggiungere un modello di sovrascrittura pod nella sezione `.spec.template` di `QueueManager` YAML.
- Se devi sovrascrivere le impostazioni sul gestore code `Route` creato da IBM MQ Operator, devi disabilitare l'impostazione dell'instradamento `.spec.route.enabled` su "false" e quindi creare il tuo percorso.
- Le impostazioni come le etichette e le annotazioni, così come le impostazioni Pod come `security Context`, possono essere tutte impostate sulla risorsa `QueueManager` .
- In altri casi, il IBM MQ Operator potrebbe non essere appropriato per il tuo caso d'uso se hai bisogno di un controllo completo.

Pianificazione per la licenza di IBM MQ nei contenitori

La licenza del contenitore ti consente di concedere in licenza solo la capacità disponibile dei tuoi singoli contenitori IBM MQ , piuttosto che richiedere la licenza dell'intero server in cui sono in esecuzione i tuoi contenitori. Per trarre vantaggio dalla licenza del contenitore, è necessario utilizzare IBM License Service per tenere traccia dell'utilizzo della licenza e determinare la titolarità richiesta.

Riferimenti correlati

“Annotazioni di licenza durante la creazione della propria immagine del contenitore IBM MQ” a pagina 161

Le annotazioni di licenza ti permettono di tenere traccia dell'utilizzo in base ai limiti definiti sul contenitore, piuttosto che sulla macchina sottostante. Configura i tuoi client per distribuire il contenitore con annotazioni specifiche che IBM License Service utilizza per tracciare l'utilizzo.

Informazioni correlate

Licenze di IBM Container

Domande frequenti sulle licenze dei contenitori

Installazione di License Service

Visualizzazione e traccia dell'utilizzo della licenza

OpenShift CP4I Kubernetes Pianificazione dell'archiviazione per IBM MQ Operator

IBM MQ Operator viene eseguito in due modalità di memoria:

- L' **archiviazione temporanea** viene utilizzata quando tutte le informazioni sullo stato del contenitore possono essere eliminate quando il contenitore viene riavviato. 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.

IBM MQ Operator fornisce la possibilità di personalizzare le caratteristiche di archiviazione che possono differire notevolmente a seconda dell'ambiente e della modalità di archiviazione 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, tutte le informazioni sullo stato per il gestore code vengono perse al riavvio. Ciò comprende:

- Tutti i messaggi
- Tutti i gestori code allo stato di comunicazione del gestore code (numeri di sequenza dei messaggi del canale)
- L'identità 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 all'eliminazione di tutto lo stato di messaggistica al riavvio, viene eliminata anche la configurazione del gestore code. 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 Red Hat OpenShift” a pagina 92). 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
```

Storage persistente

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. Questo è il funzionamento predefinito. Poiché ci sono diversi provider di memoria, ognuno dei quali supporta diverse funzionalità, ciò spesso significa che è richiesta la personalizzazione della configurazione. Il seguente esempio descrive i campi comuni che personalizzano la configurazione dell'archivio di IBM MQ nell'API v1beta1 :

- **`spec.queueManager.availability`** controlla la modalità di disponibilità. Se si utilizza `SingleInstance` o `NativeHA`, è necessaria solo l'archiviazione `ReadWriteOnce` . Per `multiInstance` è necessaria una classe di memoria che supporti `ReadWriteMany` con le caratteristiche di blocco file corrette. IBM MQ fornisce un' istruzione di supporto e una istruzione di test. La modalità di disponibilità influenza anche il layout del volume persistente. Per ulteriori informazioni, fare riferimento a “Pianificazione dell'alta disponibilità per IBM MQ nei contenitori” a pagina 17.
- **`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:
         class: ibmc-file-gold-gid
      persistedData:
        enabled: true
        class: ibmc-file-gold-gid
      recoveryLogs:
        enabled: true
        class: ibmc-file-gold-gid
  securityContext:
    supplementalGroups: [65534] # Change to 99 for clusters with RHEL7 or earlier worker nodes
```

Per informazioni sulle considerazioni sulla memoria per i gestori code della HA nativa, consultare “HA nativa” a pagina 19.

Nota: È inoltre possibile configurare gruppi supplementari con gestori code a istanza singola.

Capacità di memoria

OpenShift CP4I

Quando si utilizza IBM MQ Operator , è necessario provare ad assicurarsi di richiedere volumi sufficientemente grandi per le esigenze in corso. Tuttavia, se è necessario aumentare la capacità di memoria di uno o più volumi, questi volumi possono essere espansi se la classe di memoria supporta l'espansione dei volumi. I volumi possono essere espansi mediante una procedura online o offline. Una procedura offline richiede il riavvio dei pod `QueueManager` , mentre una procedura online non lo richiede. Per determinare se la classe di memoria supporta l'espansione del volume e quale procedura seguire per l'espansione del volume, fare riferimento alla documentazione del provider di memoria. È necessario considerare queste informazioni quando si seleziona una classe di memoria. Per una guida all'espansione del volume, consultare “Espansione dei volumi persistenti” a pagina 98.

Crittografia

OpenShift CP4I

IBM MQ non crittografa attivamente i dati inattivi. Pertanto, è necessario utilizzare l'archiviazione codificata passivamente o IBM MQ Advanced Message Security, o entrambi, per codificare i messaggi. Su IBM Cloud sia l'archiviazione blocchi che l'archiviazione file sono disponibili con la crittografia passiva inattiva.

OpenShift Kubernetes Pianificazione dell'alta disponibilità per IBM MQ nei contenitori

Esistono tre opzioni per l'alta disponibilità con IBM MQ Operator: **gestore code HA nativo** (che ha una replica attiva e due repliche in standby), **gestore code a più istanze** (che è una coppia attivo - standby, che utilizza un file system condiviso di rete) o **Gestore code resiliente singolo** (che offre un approccio semplice per l'HA utilizzando la memoria di rete). Gli ultimi due si basano sul sistema di file per assicurare la disponibilità dei dati recuperabili, tuttavia Native HA non lo fa. Pertanto, quando non si utilizza la HA nativa, la disponibilità del file system è critica per la disponibilità del gestore code. Quando il recupero dei dati è importante, il file system deve garantire la ridondanza attraverso la replica.

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 HA nativo

I gestori code della HA nativa coinvolgono un **attivo** e due pod di **replica** Kubernetes , che vengono eseguiti come parte di uno Kubernetes StatefulSet con esattamente tre repliche ciascuna con la propria serie di volumi persistenti Kubernetes . I requisiti IBM MQ per i file system condivisi si applicano anche quando si utilizza un gestore code HA nativo (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 gestore code HA nativo sono controllati dai seguenti fattori:

1. Il tempo necessario alle istanze di replica per rilevare che l'istanza attiva ha avuto esito negativo. Questo è configurabile.
2. Il tempo impiegato dal probe Pod di Kubernetes per rilevare che il contenitore pronto è stato modificato e reindirizzare il traffico di rete. Questo è configurabile.
3. Il tempo necessario ai client IBM MQ per riconnettersi.

Per ulteriori informazioni, consultare “HA nativa” a pagina 19.

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 Pod di Kubernetes per rilevare che il contenitore pronto è stato modificato e reindirizzare il traffico di rete. Questo è configurabile.
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. Questo è configurabile.
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. Questo è configurabile.
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. Quando si utilizza un carico di lavoro con stato come IBM MQ con uno Kubernetes StatefulSet, se un 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.

Nota: La modifica dei dettagli dello StatefulSet di un gestore code IBM MQ , incluso il numero di repliche, non è supportata quando il gestore code viene creato tramite il IBM MQ Operator.

Concetti correlati

Configurazioni HA (High Availability)

Attività correlate

“Configurazione dell'alta disponibilità per i gestori code utilizzando IBM MQ Operator” a pagina 74

HA nativa

La HA nativa è una soluzione di alta disponibilità nativa (integrata) per IBM MQ adatta per l'utilizzo con l'archiviazione blocchi cloud.

Una configurazione della HA nativa fornisce un gestore code ad alta disponibilità in cui i dati MQ recuperabili (ad esempio, i messaggi) vengono replicati su più serie di memoria, impedendo la perdita di memoria a causa di errori di memoria. Il gestore code è costituito da più istanze in esecuzione, una è il leader, le altre sono pronte a subentrare rapidamente in caso di errore, massimizzando l'accesso al gestore code e ai relativi messaggi.

Una configurazione della HA nativa è composta da tre pod Kubernetes , ciascuno con un'istanza del gestore code. Un'istanza è il gestore code attivo, che elabora i messaggi e scrive nel log di ripristino. Ogni

volta che viene scritto il log di ripristino, il gestore code attivo invia i dati alle altre due istanze, note come repliche. Ogni replica scrive nel proprio log di ripristino, riconosce i dati e quindi aggiorna i propri dati della coda dal log di ripristino replicato. Se il pod su cui è in esecuzione il gestore code attivo ha esito negativo, una delle istanze di replica del gestore code assume il ruolo attivo e dispone dei dati correnti con cui operare.

Il tipo di log è noto come 'log replicato '. Un log replicato è essenzialmente un log lineare, con la gestione automatica dei log e le immagini di supporto automatiche abilitate. Consultare Tipi di registrazione. Per gestire il log replicato si utilizzano le stesse tecniche utilizzate per la gestione di un log lineare.

Un Kubernetes Service viene utilizzato per instradare connessioni client TCP/IP all'istanza attiva corrente, che è identificata come l'unico pod pronto per il traffico di rete. Ciò si verifica senza che l'applicazione client sia a conoscenza delle diverse istanze.

Tre baccelli sono utilizzati per ridurre notevolmente la possibilità che si verifichi una situazione di divisione del cervello. In un sistema ad alta disponibilità a due pod, lo split - brain può verificarsi quando la connettività tra i due pod si rompe. Senza connettività, entrambi i pod possono eseguire il gestore code contemporaneamente, accumulando dati diversi. Quando la connessione viene ripristinata, ci sarebbero due versioni differenti dei dati (un 'split-brain ') e l'intervento manuale è richiesto per decidere quale serie di dati conservare e quale scartare.

La HA nativa utilizza un sistema a tre pod con quorum per evitare la situazione di split - brain. I pod che possono comunicare con almeno uno degli altri pod formano un quorum. Un gestore code può diventare solo l'istanza attiva su un pod che ha quorum. Il gestore code non può diventare attivo su un pool che non è connesso ad almeno un altro pod, quindi non possono mai esserci due istanze attive contemporaneamente:

- Se un singolo pod ha esito negativo, il gestore code su uno degli altri due pod può eseguire il controllo. Se due pod hanno esito negativo, il gestore code non può diventare l'istanza attiva sul pod rimanente perché il pod non ha quorum (il pod rimanente non può indicare se gli altri due pod hanno avuto esito negativo o se sono ancora in esecuzione e hanno perso la connettività).
- Se un singolo pod perde la connettività, il gestore code non può diventare attivo su questo pod perché il pod non ha quorum. Il gestore code su uno dei due pod rimanenti può assumere il controllo, che hanno quorum. Se tutti i pod perdono la connettività, il gestore code non è in grado di diventare attivo su nessuno dei pod, perché nessuno dei pod ha il quorum.

Se un pod attivo ha esito negativo e successivamente viene ripristinato, può unirsi nuovamente al gruppo in un ruolo di replica.

Per prestazioni e affidabilità, l'archiviazione persistente RWO (ReadWriteOnce) è consigliata per l'utilizzo con una configurazione della HA nativa. I volumi RWO da qualsiasi provider di memoria sono supportati se soddisfano le condizioni riportate di seguito:

- Ottenuto da un provider di archiviazione blocchi.
- Formattato come ext4 o XFS (che garantisce la conformità POSIX).
- Supporta il provisioning del volume dinamico e la modalità "volumeBinding: WaitForFirstConsumer".

I seguenti fornitori sono esplicitamente vietati:

- NFS
- GlusterFS
- Altri provider non di blocco.

La seguente figura mostra una tipica distribuzione con tre istanze di un gestore code distribuite in tre contenitori.

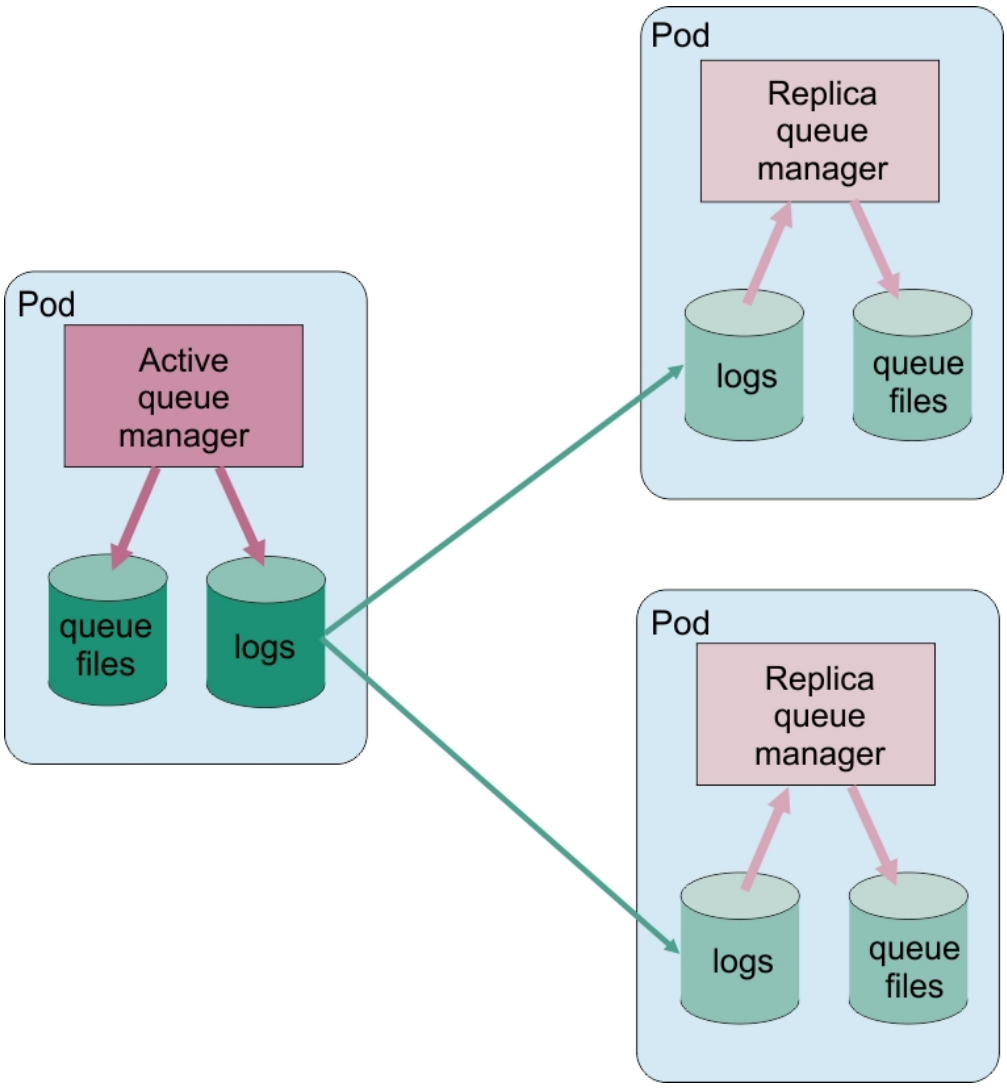


Figura 1. Esempio di configurazione della HA nativa

MQ Adv. Considerazioni sull'esecuzione di un aggiornamento continuo di un gestore code HA nativo

Qualsiasi aggiornamento alla versione o alla specifica Pod di IBM MQ per un gestore code HA nativo richiederà l'esecuzione di un aggiornamento progressivo delle istanze del gestore code. Il IBM MQ Operator lo gestisce automaticamente, ma se stai creando il tuo codice di distribuzione, ci sono alcune considerazioni importanti.

Nota: Il grafico Helm di esempio include uno script di shell per eseguire un aggiornamento progressivo, ma lo script **non** è adatto per l'utilizzo in produzione, poiché non affronta le considerazioni in questo argomento.

Kubernetes In Kubernetes `StatefulSet` , le risorse vengono utilizzate per gestire l'avvio ordinato e gli aggiornamenti continui. Parte della procedura di avvio è quella di avviare ogni Pod individualmente, attendere che diventi pronto, e quindi passare al Pod successivo. Questo non funzionerà per l'HA nativa, poiché tutti i Pods devono essere avviati in modo che possano eseguire un'elezione di leader. Pertanto, il campo `.spec.podManagementPolicy` su `StatefulSet` deve essere impostato su `Parallel`. Ciò significa anche che tutti i pod saranno aggiornati in parallelo, il che è particolarmente indesiderabile. Per questo motivo, `StatefulSet` deve utilizzare anche la strategia di aggiornamento `OnDelete` .

L'impossibilità di utilizzare il codice di aggiornamento continuo `StatefulSet` rende necessario il codice di aggiornamento continuo personalizzato, che dovrebbe considerare quanto segue:

- Procedura generale di aggiornamento a rotazione
- Ridurre al minimo i tempi di inattività aggiornando i pod nell'ordine migliore
- Gestione delle modifiche nello stato del cluster
- Gestione degli errori
- Gestione dei problemi di temporizzazione

Procedura generale di aggiornamento a rotazione

Il codice di aggiornamento progressivo deve attendere che ogni istanza mostri uno stato di REPLICA da dspmq. Ciò significa che l'istanza ha eseguito un certo livello di avvio (ad esempio, il contenitore è avviato e i processi MQ sono in esecuzione), ma non è necessariamente riuscita a comunicare con le altre istanze. Ad esempio: il Pod A viene riavviato e non appena è in stato REPLICA , il Pod B viene riavviato. Una volta che il Pod B inizia con la nuova configurazione, dovrebbe essere in grado di parlare con il Pod A, e può formare il quorum, e A o B diventeranno la nuova istanza attiva.

Come parte di questo, è utile avere un ritardo dopo che ogni pod ha raggiunto lo stato REPLICA , per consentirgli di connettersi ai suoi peer e stabilire il quorum.

Ridurre al minimo i tempi di inattività aggiornando i pod nell'ordine migliore

Il codice di aggiornamento continuo deve eliminare i pod uno alla volta, iniziando con i pod che si trovano in un stato di errore noto, seguiti da tutti i pod che non sono stati avviati correttamente. Il pod del gestore code attivo deve essere generalmente aggiornato per ultimo.

È anche importante mettere in pausa l'eliminazione dei pod se l'ultimo aggiornamento ha portato un pod in uno stato di errore noto. Ciò impedisce il roll-out di un aggiornamento interrotto su tutti i pod. Ad esempio, questo può accadere se il pod viene aggiornato per utilizzare una nuova immagine del contenitore che non è accessibile (o contiene un errore di battitura).

Gestione delle modifiche nello stato del cluster

Il codice di aggiornamento progressivo deve reagire in maniera appropriata alle modifiche in tempo reale nello stato del cluster. Ad esempio, uno dei pod del gestore code potrebbe essere eliminato a seguito di un riavvio del nodo o a causa della pressione del nodo. È possibile che un pod sfrattato non venga immediatamente ripianificato se il cluster è occupato. In questo caso, il codice di aggiornamento scorrevole dovrebbe attendere in modo appropriato prima di riavviare qualsiasi altro pod.

Gestione degli errori

Il codice di aggiornamento continuo deve essere robusto per gli errori quando si richiama l'API Kubernetes e un altro comportamento del cluster non previsto.

Inoltre, il codice di aggiornamento continuo deve essere tollerante al riavvio. Un aggiornamento a rotazione può essere di lunga durata e potrebbe essere necessario riavviare il codice.

Gestione dei problemi di temporizzazione

Il codice di aggiornamento continuo deve controllare le revisioni di aggiornamento del pod, in modo che possa garantire che il pod sia stato riavviato. Ciò evita problemi di sincronizzazione in cui un pod può indicare che è "Avviato", ma in realtà non è ancora terminato.

Concetti correlati

“Scelta della modalità di utilizzo di IBM MQ nei contenitori” a pagina 8
Ci sono diverse opzioni per l'utilizzo di IBM MQ nei contenitori: puoi scegliere di utilizzare IBM MQ Operator, che utilizza le immagini del contenitore preconfezionate, oppure puoi creare le tue immagini e il tuo codice di distribuzione.

OpenShift CP4I Kubernetes Ripristino di emergenza per IBM MQ nei contenitori

Devi considerare a quale tipo di disastro ti stai preparando. Negli ambienti cloud, l'utilizzo delle zone di disponibilità fornisce un certo livello di tolleranza per i disastri e sono molto più facili da usare. Se hai un numero dispari di data center (per quorum) e un link di rete a bassa latenza, puoi potenzialmente eseguire un singolo cluster Red Hat OpenShift Container Platform o Kubernetes con più zone di disponibilità, ognuna in un'ubicazione fisica separata. Questo argomento illustra le considerazioni per il ripristino di emergenza in cui questi criteri non possono essere soddisfatti, ovvero un numero pari di data center o un link di rete ad alta latenza.

Per il ripristino di emergenza, è necessario considerare quanto segue:

- Replica dei dati IBM MQ (conservati in una o più risorse `PersistentVolume`) nell'ubicazione di ripristino di emergenza
- Ricreazione del gestore code utilizzando i dati replicati
- L'ID di rete del gestore code visibile alle applicazioni client IBM MQ e ad altri gestori code. Questo ID potrebbe essere una voce DNS, ad esempio.

I dati persistenti devono essere replicati, in modo sincrono o asincrono, sul sito di ripristino di emergenza. Ciò è in genere specifico del provider di memoria, ma può essere eseguito anche utilizzando un `VolumeSnapshot`. Consultare Istantanee volume CSI per ulteriori informazioni sulle istantanee volume.

Quando si esegue il ripristino da un'emergenza, sarà necessario ricreare l'istanza del gestore code sul nuovo cluster Kubernetes , utilizzando i dati replicati. Se stai utilizzando IBM MQ Operator, avrai bisogno dello YAML `QueueManager` e dello YAML per altre risorse di supporto come `ConfigMap` o `Secret`.

Informazioni correlate

ha_for_ctr.dita

OpenShift CP4I Pianificazione della sicurezza per IBM MQ nei contenitori

Considerazioni sulla sicurezza quando pianifichi il tuo IBM MQ nella configurazione dei contenitori.

Procedura

- “Autenticazione e autorizzazione utente per IBM MQ nei contenitori” a pagina 23
  - “Vincoli di sicurezza sull'utilizzo degli utenti del sistema operativo nei contenitori” a pagina 24
- “Considerazioni per limitare il traffico di rete a IBM MQ nei contenitori” a pagina 24

Autenticazione e autorizzazione utente per IBM MQ nei contenitori

IBM MQ nei contenitori può essere configurato per autenticare gli utenti tramite LDAP, Mutual TLS o un plugin MQ personalizzato.

Tieni presente che l'operatore IBM MQ non consente l'uso di utenti e gruppi del sistema operativo all'interno dell'immagine del contenitore. Per ulteriori informazioni, consultare “Vincoli di sicurezza sull'utilizzo degli utenti del sistema operativo nei contenitori” a pagina 24.

LDAP

Per informazioni sulla configurazione di IBM MQ per l'utilizzo di un repository utente LDAP, consultare Autenticazione connessione: Repository utente e Autorizzazione LDAP.

TLS reciproco

Se si configurano le connessioni in entrata a un gestore code per richiedere un certificato TLS (TLS reciproco), è possibile associare il DN del certificato a un nome utente. Devi fare due cose:

- Configurare un record di autenticazione di canale per creare l'associazione a un nome utente, utilizzando SSLPEER. Per ulteriori informazioni, consultare Associazione di un DN (Distinguished Name) SSL o TLS a un ID utente MCAUSER.
- Configurare il gestore code per definire i record di autorizzazioni per un nome utente non riconosciuto dal sistema. Per ulteriori informazioni, consultare Stanza di servizio del file qm.ini.

Token Web JSON

Per informazioni sulla configurazione di IBM MQ per utilizzare JWT (JSON Web Tokens), vedi Gestione dei token di autenticazione.

Plug-in MQ personalizzato

Questa è una tecnica avanzata, e richiede molto più lavoro. Per ulteriori informazioni, vedi Utilizzo di un servizio di autorizzazione personalizzato.

Attività correlate

"Esempio: configurazione di un gestore code con autenticazione TLS reciproca" a pagina 68
Questo esempio distribuisce un gestore code in OpenShift Container Platform utilizzando IBM MQ Operator. Il TLS reciproco viene utilizzato per l'autenticazione, per eseguire l'associazione da un certificato TLS a un'identità nel gestore code.

Vincoli di sicurezza sull'utilizzo degli utenti del sistema operativo nei contenitori

L'utilizzo di utenti del sistema operativo nei contenitori non è consigliato e non è consentito con l'operatore IBM MQ .

In un ambiente containerizzato a più tenant, i vincoli di sicurezza vengono di norma messi in atto per impedire potenziali problemi di sicurezza, ad esempio:

- **Come impedire l'utilizzo dell'utente "root" in un contenitore**
- **forzatura dell'uso di un UID casuale**. Ad esempio, in Red Hat OpenShift Container Platform il valore predefinito `SecurityContextConstraints` (denominato `restricted`) utilizza un ID utente casuale per ogni contenitore.
- **Impedire l'utilizzo dell'escalation di privilegi**. IBM MQ on Linux utilizza l'escalation dei privilegi per controllare la password degli utenti - utilizza un programma "setuid" in modo da diventare l'utente "root" per eseguire questa operazione.

OpenShift CP4I Per garantire la conformità a queste misure di sicurezza, IBM MQ Operator non consente l'utilizzo di ID definiti nelle librerie del sistema operativo all'interno di un contenitore. Nessun ID utente o gruppo `mqm` definito nel contenitore.

Considerazioni per limitare il traffico di rete a IBM MQ nei contenitori

Puoi definire le politiche di rete per limitare il traffico ai pod nel tuo cluster in OpenShift Container Platform e Kubernetes. Questo argomento descrive alcune considerazioni su come le politiche di rete possono essere applicate a IBM MQ.

Per l'ingresso di rete in un gestore code, ci sono diverse porte da prendere in considerazione:

- Porta 1414 per il traffico del gestore code
- Porta 9414 per HA nativo
- Porta 9157 per le metriche
- Porta 9443 per la console web e API REST

L'uscita dalla rete è più complessa. Esempi di uscita di rete che si potrebbe voler considerare:

- DNS - se si dispone di canali o altre configurazioni che utilizzano nomi DNS
- Altri gestori code

- OCSP (Online Certificate Status Protocol) e CRL (Certificate Revocation Lists) - determinati dal provider di certificati.
- Provider di autenticazione:
  - LDAP
  - Aprire ID Connect o un altro provider di login configurato per il server Web IBM MQ . Ciò include IBM Cloud Pak Keycloak.
- Provider di traccia:
  - IBM Instana

  Nota: Per le versioni precedenti di IBM MQ , IBM Cloud Pak for Integration Operations Dashboard era disponibile anche come provider di traccia. Tuttavia, il dashboard Operazioni è stato rimosso in IBM MQ 9.3.3 CD e IBM MQ 9.4.0 LTS.

Esempio di NetworkPolicy

Di seguito è riportata una politica di rete di esempio per controllare l'ingresso per un gestore code denominato "myqm", da utilizzare su Red Hat OpenShift Container Platform.

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: myqm
spec:
  podSelector:
    matchLabels:
      app.kubernetes.io/instance: myqm
      app.kubernetes.io/name: ibm-mq
  ingress:
    # Allow access to queue manager listener from anywhere
    - ports:
        - protocol: TCP
          port: 1414
    # Allow access to Native HA port from other instances of the same queue manager
    - from:
        - podSelector:
            matchLabels:
              app.kubernetes.io/instance: myqm
              app.kubernetes.io/name: ibm-mq
      ports:
        - protocol: TCP
          port: 9414
    # Allow access to metrics from monitoring project
    - from:
        - namespaceSelector:
            matchLabels:
              network.openshift.io/policy-group: monitoring
      ports:
        - protocol: TCP
          port: 9157
    # Allow access to web server via Route
    - from:
        - namespaceSelector:
            matchLabels:
              network.openshift.io/policy-group: ingress
      ports:
        - protocol: TCP
          port: 9443
```

Conformità FIPS per IBM MQ nei contenitori

All'avvio, IBM MQ nei contenitori rileva se il sistema operativo su cui si sta avviando il contenitore è conforme a FIPS e (in tal caso) configura automaticamente il supporto FIPS. I requisiti e le limitazioni sono indicati qui.

FIS (Federal Information Processing Standards)

Il governo degli Stati Uniti fornisce consulenza tecnica sui sistemi IT e sulla sicurezza, inclusa la crittografia dei dati. Il National Institute for Standards and Technology (NIST) è un ente governativo che si occupa dei sistemi informatici e della sicurezza. NIST produce raccomandazioni e standard, inclusi FIPS (Federal Information Processing Standards).

Uno standard FIPS significativo è FIPS 140-2, che richiede l'utilizzo di forti algoritmi crittografici. FIPS 140-2 specifica anche i requisiti per gli algoritmi di hash da utilizzare per proteggere i pacchetti dalle modifiche in transito.

IBM MQ fornisce il supporto FIPS 140-2 se è stato configurato per farlo.

Nota: Su AIX, Linux, and Windows, IBM MQ fornisce la conformità FIPS 140-2 tramite il modulo crittografico IBM Crypto for C (ICC) . Il certificato per questo modulo è stato spostato nello stato cronologico. I clienti devono visualizzare il certificatoIBM Crypto for C (ICC) ed essere a conoscenza di eventuali consigli forniti da NIST. Un modulo FIPS 140-3 di sostituzione è attualmente in corso e il relativo stato può essere visualizzato ricercandolo in NIST CMVP modules in process list.

IBM MQ Operator 3.2.0 e l'immagine del contenitore del gestore code 9.4.0.0 sono basati su UBI 9. La conformità FIPS 140-3 è attualmente in sospeso e il suo stato può essere visualizzato ricercando "Red Hat Enterprise Linux 9 - OpenSSL FIPS Provider" in NIST CMVP modules in process list.

Requisiti

Per i requisiti relativi alla configurazione del cluster e altre considerazioni, consultare FIPS Wall: Current IBM approach to FIPS compliance.

IBM MQ nei contenitori può essere eseguito in modalità di conformità FIPS 140-2. Durante l'avvio, IBM MQ nei contenitori rileva se il sistema operativo host su cui viene avviato il contenitore è conforme a FIPS. Se il sistema operativo dell'host è conforme a FIPS e sono state fornite chiavi private e certificati, il contenitore IBM MQ configura il gestore code, il server web IBM MQ e il trasferimento dati tra i nodi in una distribuzione Native High Availability, per l'esecuzione in modalità di conformità FIPS.

Quando si utilizza IBM MQ Operator per distribuire i gestori code, l'operatore crea un instradamento con tipo di terminazione **`Passthrough`**. Ciò significa che il traffico viene inviato direttamente alla destinazione senza che il router fornisca la terminazione TLS. Il gestore code IBM MQ e il server web IBM MQ sono le destinazioni in questo caso e forniscono già comunicazioni protette conformi a FIPS.

Requisiti chiave:

1. Una chiave privata e i certificati, forniti in un segreto al gestore code e al server Web, che consentono ai client esterni di connettersi in modo sicuro al gestore code e al server web.
2. Una chiave privata e i certificati per trasferimento dati tra nodi differenti in una configurazione Native High Availability.

Limitazioni

Per una distribuzione conforme a FIPS di IBM MQ nei contenitori, considerare quanto segue:

- IBM MQ nei contenitori fornisce un endpoint per la raccolta di metriche. Attualmente questo endpoint è solo HTTP. È possibile disattivare l'endpoint delle metriche per rendere il resto di IBM MQ compatibile con FIPS.
- IBM MQ nei contenitori consente sovrascritture di immagini personalizzate. In altre parole, puoi creare immagini personalizzate utilizzando l'immagine del contenitore IBM MQ come immagine di base. La conformità FIPS potrebbe non essere applicabile per tali immagini personalizzate.
- Per la traccia dei messaggi mediante IBM Instana, la comunicazione tra IBM MQ e IBM Instana è HTTP o HTTPS, senza conformità FIPS.
- L'accesso IBM MQ Operator ai servizi IBM identity and access management (IAM) /Zen non è conforme a FIPS.

Modalità di rilevamento della conformità FIPS e configurazione automatica del supporto FIPS

Se il sistema operativo su cui viene avviato il contenitore è conforme a FIPS, il supporto FIPS viene configurato automaticamente.

Nota: Su AIX, Linux, and Windows, IBM MQ fornisce la conformità FIPS 140-2 tramite il modulo crittografico IBM Crypto for C (ICC) . Il certificato per questo modulo è stato spostato nello stato cronologico. I clienti devono visualizzare il certificatoIBM Crypto for C (ICC) ed essere a conoscenza di eventuali consigli forniti da NIST. Un modulo FIPS 140-3 di sostituzione è attualmente in corso e il relativo stato può essere visualizzato ricercandolo in NIST CMVP modules in process list.

IBM MQ Operator 3.2.0 e l'immagine del contenitore del gestore code 9.4.0.0 sono basati su UBI 9. La conformità FIPS 140-3 è attualmente in sospeso e il suo stato può essere visualizzato ricercando "Red Hat Enterprise Linux 9 - OpenSSL FIPS Provider" in NIST CMVP modules in process list.

Durante l'avvio, IBM MQ nei contenitori rileva se il sistema operativo su cui viene avviato il contenitore è conforme a FIPS. In tal caso, vengono eseguite automaticamente le seguenti operazioni:

Gestore code
Se il sistema operativo dell'host è conforme a FIPS e vengono forniti la chiave privata e i certificati, l'attributo del gestore code **SSLFIPS** è impostato su YES. Altrimenti, l'attributo **SSLFIPS** è impostato su NO.

IBM MQ Server Web
Il server web IBM MQ fornisce un'interfaccia HTTP/HTTPS per la gestione di IBM MQ. Se il sistema operativo host è compatibile con FIPS, le opzioni JVM vengono aggiornate per fare in modo che il server Web utilizzi la crittografia compatibile con FIPS. Per poter utilizzare FIPS, è necessario fornire la chiave privata e i certificati durante l'avvio del contenitore.

HA nativa
La sicurezza dei dati replicati tra nodi è controllata dalla sezione **NativeHALocalInstance** del file `qm.ini` . Ad esempio:

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

Se FIPS è abilitato, l'attributo **SSLFipsRequired** viene aggiunto alla stanza, con il valore impostato su `Sì`:

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

Se il contenitore è in esecuzione in un cluster OpenShift senza supporto FIPS, il gestore code, il server web IBM MQ e i componenti della HA nativa non hanno il supporto FIPS abilitato automaticamente. Solo l'architettura x86-64 è attualmente supportata dalla piattaforma OpenShift per FIPS. Per le architetture Power e Linux for IBM Z OpenShift non offre il supporto FIPS. Per abilitare esplicitamente il supporto FIPS nei componenti IBM MQ per queste architetture, impostare la variabile di ambiente *MQ_ENABLE_FIPS* su `true` nel gestore code YAML. Il seguente frammento YAML descrive l'utilizzo della variabile di ambiente *MQ_ENABLE_FIPS* :

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

Sovrascrittura della modalità FIPS automatica per IBM MQ nei contenitori

Utilizzare la variabile di ambiente *MQ_ENABLE_FIPS* per abilitare o disabilitare esplicitamente la modalità FIPS per i componenti IBM MQ nel contenitore.

Prima di iniziare

Nota: Su AIX, Linux, and Windows, IBM MQ fornisce la conformità FIPS 140-2 tramite il modulo crittografico IBM Crypto for C (ICC) . Il certificato per questo modulo è stato spostato nello stato cronologico. I clienti devono visualizzare il certificatoIBM Crypto for C (ICC) ed essere a conoscenza di eventuali consigli forniti da NIST. Un modulo FIPS 140-3 di sostituzione è attualmente in corso e il relativo stato può essere visualizzato ricercandolo in NIST CMVP modules in process list.

IBM MQ Operator 3.2.0 e l'immagine del contenitore del gestore code 9.4.0.0 sono basati su UBI 9. La conformità FIPS 140-3 è attualmente in sospeso e il suo stato può essere visualizzato ricercando "Red Hat Enterprise Linux 9 - OpenSSL FIPS Provider" in NIST CMVP modules in process list.

Informazioni su questa attività

MQ_ENABLE_FIPS supporta tre valori:

automatico
Questo è il valore predefinito.

Se il sistema operativo host è abilitato a FIPS, tutti i componenti (gestore code, server Web IBM MQ e HA nativa) vengono eseguiti in modalità FIPS.

Se il sistema operativo host non è abilitato a FIPS, tutti i componenti non vengono eseguiti in modalità FIPS.

vero, true
Questo valore attiva FIPS per i componenti selezionati nel contenitore.

L'attributo del gestore code **SSLFIPS** è impostato su YES anche se IBM MQ nei contenitori è in esecuzione su un sistema operativo host non conforme a FIPS. Vale a dire, se il gestore code IBM MQ , il server web e la HA nativa sono conformi a FIPS, ma il sistema operativo del contenitore non lo è.

No
Questo valore disattiva la conformità FIPS.

L'attributo del gestore code **SSLFIPS** è impostato su NO, anche se IBM MQ nei contenitori è in esecuzione su una macchina host conforme a FIPS. Tuttavia, IBM MQ protegge ancora le connessioni se vengono forniti la chiave privata e i certificati.

Le opzioni JVM non vengono aggiornate per il server Web IBM MQ . Tuttavia, il server web IBM MQ esegue ancora un endpoint HTTPS se vengono forniti la chiave privata e i certificati.

La replica dei dati nella HA nativa non utilizza la codifica FIPS.

Esempio

Di seguito è riportato un esempio di YAML del gestore code che descrive l'attivazione di TLS e FIPS per il componente gestore code:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  namespace: ibm-mq-fips
  name: ibm-mq-qm-ppcle
spec:
  license:
    accept: true
    license: L-EHXT-MQCRN9
    use: Production
  queueManager:
    name: PPCLEQM
    storage:
      queueManager:
        type: ephemeral
  template:
```

```
    pod:
      containers:
        - env:
            - name: MQ_ENABLE_FIPS
              value: "true"
          name: qmgr
  version: 9.4.0.0-r1
  web:
    enabled: false
  pki:
    keys:
      - name: ibm-mq-tls-certs
        secret:
          secretName: ibm-mq-tls-secret
          items:
          - tls.key
          - tls.crt
```

Pianificazione della scalabilità e delle prestazioni per IBM MQ nei contenitori

Nella maggior parte dei casi, la scalabilità e le prestazioni di IBM MQ nei contenitori sono le stesse di IBM MQ for Multiplatforms. Tuttavia, ci sono alcuni limiti aggiuntivi che possono essere imposti dalla piattaforma del contenitore.

Informazioni su questa attività

Quando si pianifica la scalabilità e le prestazioni per IBM MQ nei contenitori, considerare le opzioni riportate di seguito:

Procedura

- **Limitare il numero di thread e processi.**

  IBM MQ utilizza i thread per gestire la simultaneità. In Linux, i thread vengono implementati come processi, in modo che sia possibile incontrare i limiti imposti dalla piattaforma del contenitore o dal sistema operativo, sul numero massimo di processi. Da Red Hat OpenShift Container Platform 4.11, esiste un limite predefinito di 4096 processi per contenitore. Sebbene ciò sia adeguato per la maggior parte degli scenari, in alcuni casi potrebbe influire sul numero di connessioni client per un gestore code.

  Il limite di processi in Kubernetes può essere configurato dall'amministratore del cluster utilizzando l'impostazione di configurazione kubelet **podPidsLimit**. Vedi Limiti e prenotazioni dell'ID processo nella documentazione di Kubernetes . In Red Hat OpenShift Container Platform, puoi anche creare una **ContainerRuntimeConfig** risorsa personalizzata per modificare i CRI-O.

  Nella configurazione IBM MQ , è anche possibile impostare il numero massimo di connessioni client per un gestore code. Consultare Limiti del canale di connessione server per l'applicazione dei limiti a un singolo canale di connessione server e l' attributo MAXCHANNELS INI per applicare i limiti all'intero gestore code.
- **Limita numero di volumi.**

  Nei sistemi cloud e contenitore, i volumi di archiviazione collegati alla rete sono comunemente utilizzati. Esistono dei limiti al numero di volumi che possono essere collegati ai nodi Linux . Ad esempio, AWS EC2 limita a non più di 30 volumi per VM. Red Hat OpenShift Container Platform ha un limite simile, come Microsoft Azure e Google Cloud Platform.

  Un gestore code HA nativo richiede un volume per ognuna delle tre istanze e impone la diffusione delle istanze tra i nodi. Tuttavia, è possibile configurare il gestore code in modo che utilizzi tre volumi per istanza (dati del gestore code, log di ripristino e dati persistenti).
- **Utilizza tecniche di scaling IBM MQ .**

  Invece di un numero ridotto di gestori code di grandi dimensioni, può essere utile utilizzare le tecniche di scalabilità IBM MQ come i cluster uniformi IBM MQ per eseguire più gestori code con la stessa configurazione. Ciò ha il vantaggio aggiunto che l'impatto del riavvio di un singolo contenitore (ad esempio, come parte della manutenzione della piattaforma del contenitore) è ridotto.

Preparazione, installazione e aggiornamento del tuo ambiente per IBM MQ nei contenitori

È possibile eseguire una gamma di attività per preparare il proprio ambiente per IBM MQ

Informazioni su questa attività

Se si utilizza IBM MQ Operator, preparare il cluster Red Hat OpenShift Container Platforminstallando l'operatore. Vedere “Installazione e aggiornamento di IBM MQ Operator” a pagina 30

Altrimenti, prepari il tuo ambiente contenitore creando le tue proprie immagini contenitore. Vedere “Preparazione per IBM MQ creando la propria immagine contenitore” a pagina 54

Installazione e aggiornamento di IBM MQ Operator

È possibile eseguire una serie di attività per installare, disinstallare e aggiornare IBM MQ Operator.

Informazioni su questa attività

Per iniziare a installare e aggiornare IBM MQ Operator su Red Hat OpenShift Container Platform, consultare i seguenti argomenti.

Procedura

OpenShift CP4I Dipendenze per IBM MQ Operator

Nessun altro operatore viene installato automaticamente quando si installa IBM MQ Operator.

IBM Licensing Operator deve essere installato separatamente per tenere traccia dell'utilizzo della licenza. Consultare Distribuzione del servizio di licenza License Service nella documentazione IBM Cloud Pak for Integration .

Quando si crea un `QueueManager` utilizzando una licenza IBM Cloud Pak for Integration , è possibile scegliere se si desidera o meno utilizzare SSO (single sign - on) con l'istanza IBM Cloud Pak for Integration di Keycloak. L'utilizzo di Keycloak è abilitato per impostazione predefinita con una licenza IBM Cloud Pak for Integration , ma se non è installato, `QueueManager` entrerà in uno stato "Bloccato" fino a quando non saranno installate le dipendenze corrette. Consultare “Installazione del IBM MQ Operator” a pagina 31 per ulteriori dettagli sulle dipendenze.

OpenShift CP4I Autorizzazioni nell'ambito del cluster richieste da IBM MQ Operator

IBM MQ Operator richiede autorizzazioni nell'ambito del cluster per gestire i webhook di ammissione e gli esempi e per leggere le informazioni sulla classe di archiviazione e sulla versione cluster.

IBM MQ Operator richiede le seguenti autorizzazioni nell'ambito del cluster:

- Autorizzazione a gestire i webhook di ammissione. Ciò consente di creare, richiamare e aggiornare specifici webhook utilizzati nel processo di creazione e gestione dei contenitori forniti dall'operatore.
  - Gruppi API: **`admissionregistration.k8s.io`**

  - Risorse: **validatingwebhookconfigurations**
  - verbs: **get, delete**
- Autorizzazione per creare e gestire le risorse utilizzate nella console Red Hat OpenShift per fornire esempi e frammenti quando si creano risorse personalizzate.
  - Gruppi API: **console.openshift.io**
  - Risorse: **consoleyamlsamples**
  - verbs: **create, get, update, delete**
- Autorizzazione a leggere la versione del cluster. Ciò consente all'operatore di eseguire il feed back di eventuali problemi con l'ambiente cluster.
  - Gruppi API: **config.openshift.io**
  - Risorse: **clusterversions**
  - verbs: **get, list, watch**
- Autorizzazione a leggere le classi di memoria sul cluster. Ciò consente all'operatore di eseguire il feed di eventuali problemi con le classi di archiviazione selezionate nei contenitori.
  - Gruppi API: **storage.k8s.io**
  - Risorse: **storageclasses**
  - verbs: **get, list**

Nota: IBM MQ Operator richiede anche autorizzazioni nell'ambito dello spazio dei nomi. Se IBM MQ Operator è installato in un ambito cluster, le autorizzazioni nell'ambito dello spazio dei nomi sono presenti in tutti gli spazi dei nomi.

OpenShift CP4I Verifica delle firme di immagine

Le immagini del contenitore del gestore code IBM MQ Operator e IBM MQ sono firmate digitalmente.

Informazioni su questa attività

Le firme digitali forniscono un modo ai consumatori di contenuti per garantire che ciò che scaricano sia autentico (proviene dalla fonte prevista) e abbia integrità (è ciò che ci aspettiamo che sia).

Procedura

- Verificare le firme delle immagini del contenitore del gestore code IBM MQ Operator e IBM MQ :
  - Vedi Verifica delle firme dell'immagine nella IBM Cloud Pak for Integration (CP4I) 16.1.0 documentazione.

OpenShift CP4I Installazione del IBM MQ Operator

Il IBM MQ Operator può essere installato su Red Hat OpenShift utilizzando la console OpenShift o la CLI (command line interface).

Prima di iniziare

Importante:

- Questo argomento è per l'installazione di IBM MQ Operator per uso autonomo **solo**. Se si intende utilizzare l'SSO IBM Cloud Pak for Integrationo Keycloak per uno o più gestori code, fare riferimento a “Installazione di IBM MQ Operator per l'utilizzo con CP4I” a pagina 39.
- Esamina le istruzioni su come strutturare la tua distribuzione prima di installare IBM MQ Operator.

Per assicurarsi che l'installazione avvenga nel modo più semplice possibile, accertarsi di aver compreso tutti i prerequisiti e i requisiti prima di avviare l'installazione. Consultare “Pianificazione per IBM MQ nei contenitori” a pagina 7.

Informazioni su questa attività

I seguenti passi rappresentano il tipico flusso di attività per installare IBM MQ Operator:

1. Installare Red Hat OpenShift Container Platform.
2. Configurare l'archiviazione.
3. Immagini mirror (solo air - gap).
4. Aggiungi il catalogo IBM MQ Operator.
5. Installa IBM MQ Operator.
6. Creare il segreto della chiave di titolarità (solo installazioni online).
7. Distribuire il servizio di licenza License Service.
8. Distribuire un gestore code.

Procedura

1. Installa Red Hat OpenShift Container Platform.

   Per la procedura dettagliata per l'installazione di OpenShift, consultare Installazione del software Red Hat 4.6 o successiva.

   Importante: Assicurarsi di installare una versione supportata di OpenShift Container Platform. Ad esempio, per utilizzare IBM MQ Operator 3.2 o versioni successive, è necessario installare OpenShift Container Platform 4.12 o versioni successive. Per ulteriori informazioni, vedi IBM Cloud Pak and Red Hat OpenShift Container Platform compatibility.

   Per qualsiasi passo che utilizza la CLI Red Hat OpenShift Container Platform , devi essere collegato al tuo cluster OpenShift con `oc login`. Per installare la CLI, vedi Introduzione alla CLI OpenShift.

   Dopo aver installato OpenShift, puoi verificare e ottenere l'accesso al tuo software del contenitore utilizzando la IBM chiave di titolarità che crei in Crea il segreto della chiave di titolarità.
2. Configurare la memoria.

   È necessario definire le classi di memoria in Red Hat OpenShift Container Platform e impostare la propria configurazione di memoria per soddisfare i propri requisiti di dimensione.

   Importante: I gestori code IBM MQ a istanza singola e HA nativa possono utilizzare la modalità di accesso RWO, mentre i gestori code a più istanze richiedono RWX come descritto in “Pianificazione dell'archiviazione per IBM MQ Operator” a pagina 15. I gestori code a più istanze IBM MQ richiedono particolari caratteristiche del file system, che possono essere verificate utilizzando le istruzioni per Verifica di un file system condiviso per IBM MQ.

   Un elenco di file system noti, conformi e non conformi, e note su altri limiti o limitazioni, è disponibile nell' istruzione di test per i file system IBM MQ.

   I provider di memoria consigliati possono essere trovati nella pagina CP4I Considerazioni sulla memoria .
3. Immagini speculari (solo air - gap).

   Se il cluster si trova in un ambiente di rete limitato (con air-gapping), è necessario eseguire il mirroring delle immagini IBM MQ utilizzando i seguenti valori:

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

   Per creare immagini di mirroring, consultare Mirroring delle immagini per un cluster air - gapped.
4. Aggiungere l'origine del catalogo IBM MQ Operator .

   Aggiungi l'origine del catalogo che rende gli operatori disponibili al tuo cluster. Consultare “Aggiunta dell'origine del catalogo IBM MQ Operator” a pagina 33.
5. Installare IBM MQ Operator.

Scegli una delle seguenti due opzioni (utilizza la console o utilizza la CLI):

- Opzione 1: Installare IBM MQ Operator utilizzando la console OpenShift.
- Opzione 2 Installa IBM MQ Operator utilizzando la CLI OpenShift.

6. Creare il segreto della chiave di titolarità (solo installazioni in linea).

   IBM MQ Operator distribuisce le immagini del gestore code estratte da un registro del contenitore che esegue un controllo di titolarità della licenza. Questo controllo richiede una chiave di titolarità memorizzata in un segreto di pull `docker-registry` . Se non si dispone ancora di una chiave di titolarità nello spazio dei nomi in cui verranno installati i gestori code, seguire queste istruzioni per ottenere una chiave di titolarità e creare un segreto di pull.

   Nota: La chiave di titolarità non è richiesta se verranno distribuiti solo i gestori code IBM MQ Advanced for Developers (non Warranted).

   Puoi creare il segreto della chiave di titolarità utilizzando la console OpenShift o la CLI. Il seguente esempio utilizza la CLI:

   a. Ottieni la chiave di titolarità assegnata al tuo ID IBM . 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.
   c. Dalla CLI OpenShift , immetti il seguente comando per creare un segreto di pull dell'immagine denominato `ibm-entitlement-key`.

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

      Dove *entitlement_key* è la chiave di titolarità che hai copiato nel passaggio b, *user_email* è l' IBM ID associato al software autorizzato e *namespace* è lo spazio dei nomi in cui hai installato il tuo IBM MQ Operator .

7. Distribuire License Service.

   Ciò è necessario per il monitoraggio dell'utilizzo della licenza dei gestori code. Seguire le istruzioni riportate in Distribuzione del servizio di licenza License Service.

8. Distribuire un gestore code.

   Per istruzioni sulla distribuzione di un gestore code di esempio "avvio rapido" , consultare “Distribuzione di un gestore code semplice utilizzando IBM MQ Operator” a pagina 62.

Attività correlate

“Disinstallazione di IBM MQ Operator” a pagina 52
Puoi utilizzare la console o la CLI Red Hat OpenShift per disinstallare IBM MQ Operator da Red Hat OpenShift.

OpenShift *Aggiunta dell'origine del catalogo IBM MQ Operator*

Aggiungere l'origine del catalogo IBM MQ Operator al proprio cluster OpenShift per rendere IBM MQ Operator disponibile per l'installazione. Questa attività è richiesta anche se si applicano i fix pack di origine del catalogo prima di completare un aggiornamento.

Informazioni su questa attività

Un catalogo di operatori è un indice di operatori disponibili per estendere l'API di un cluster Red Hat OpenShift Container Platform per abilitare i prodotti software IBM .

Sono disponibili le seguenti origini catalogo:

Opzione 1: origine catalogo specifica per IBM MQ Operator.

Utilizzando un'origine del catalogo IBM MQ Operator specifica, si ottiene il controllo completo della versione del software su un cluster e quando si verificano gli aggiornamenti. Una nuova versione di IBM MQ Operator diventa disponibile **solo** in un cluster OpenShift dopo aver aggiornato l'origine del catalogo. Questo processo fornisce in modo efficace il controllo manuale degli aggiornamenti, quindi non è necessario utilizzare l'opzione `Manuale` per l'impostazione **Update approval** per gli operatori. L'opzione **Manuale** forza l'esecuzione di tutti i possibili aggiornamenti contemporaneamente e può bloccare gli aggiornamenti, quindi utilizzare solo l'opzione **Automatico** . Per ulteriori informazioni, consulta la sezione "Limitazione degli aggiornamenti automatici con una strategia di approvazione" di Installazione degli operatori utilizzando la console Red Hat OpenShift.

Scegliere questa opzione se si sta completando un aggiornamento e si deve aggiungere l'origine del catalogo IBM MQ Operator di una versione più recente.

Per utilizzare questa opzione, passare a Opzione 1: aggiungere origini di catalogo specifiche per IBM MQ Operator.

Opzione 2: IBM Operator Catalog.

Con questa opzione, le nuove versioni dell'operatore diventano disponibili e vengono applicate **senza** alcun intervento da parte dell'utente. Utilizzare questa opzione **solo** per le installazioni in linea in cui si desiderano gli aggiornamenti **automatici** di IBM MQ Operatore in cui le installazioni deterministiche non sono necessarie.

Nota: Questa opzione può essere utile per ambienti di prova, ma **non è adatta per ambienti di produzione**.

Per utilizzare questa opzione, passare a Opzione 2: aggiungere il IBM Catalogo operatore.

Procedura

- **Opzione 1: aggiungere origini di catalogo specifiche per IBM MQ Operator.**

  Questa attività presuppone che siano stati completati i primi 3 passi di “Installazione del IBM MQ Operator” a pagina 31.

  Questa attività deve essere eseguita da un amministratore cluster e deve essere eseguita utilizzando la CLI.

  a) Solo aggiornamento: se si stanno applicando i fix pack di origine del catalogo prima di un aggiornamento, completare la seguente procedura:

  – Confermare che gli operatori siano in esecuzione correttamente.

  – Se sono presenti aggiornamenti IBM MQ Operator in sospeso che richiedono l'approvazione manuale, approvarli prima di avviare questa procedura. Per ulteriori informazioni, vedi "Limitazione degli aggiornamenti automatici con una strategia di approvazione" in Installazione degli operatori utilizzando la console Red Hat OpenShift.

  b) Se non l'hai già installato o se è necessario aggiornarlo, scarica il plug-in IBM Catalog Management (versione 1.6.0 o successiva) da GitHub.

  Questo plug-in consente di eseguire i comandi **oc ibm-pak** sul cluster.

  c) Accedi al tuo cluster utilizzando il comando **oc login** e le credenziali utente:

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

  d) Esportare le seguenti variabili di ambiente per IBM MQ Operator:

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

  dove *ARCHITECTURE* si riferisce all'architettura del sistema su cui si sta distribuendo IBM MQ Operatore ha un valore di `amd64`, `ppc64leo` `s390x`.

Importante: Se si sta spostando dal catalogo operatore IBM all'origine del catalogo specifico per IBM MQ Operator, impostare `OPERATOR_VERSION` sulla versione della distribuzione di IBM MQ Operator.

e) Scaricare i file per l'operatore IBM MQ .

Nota: Se si sta completando un'installazione **air - gapped** , è necessario disporre già dei file necessari dopo aver completato il passo "Immagini di mirroring" di "Installazione di IBM MQ Operator", nel qual caso è possibile passare al passo “8” a pagina 35 "Applicare l'origine del catalogo IBM MQ Operator al cluster".

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

f) Generare l'origine del catalogo richiesta per IBM MQ Operator:

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

g) Opzionale: Generare le origini del catalogo e salvarle in un'altra directory.

a. Ottieni l'origine del catalogo:

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

b. (Facoltativo) Passare alla directory nel browser di file per copiare queste risorse utente in file che è possibile conservare per il riutilizzo o per le pipeline.

h) Applica l'origine del catalogo IBM MQ Operator al cluster.

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

i) Conferma che l'origine del catalogo IBM MQ Operator è stata creata nel namespace `openshift-marketplace` :

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

Output di esempio:

```
oc get catalogsource -n openshift-marketplace

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

Si è ora pronti a completare il Passo 5 dell'installazione di IBM MQ Operator.

- **Opzione 2: aggiungere il IBM Catalogo operatore.**

Importante: Utilizzare il IBM Catalogo operatore **solo** per le installazioni in linea in cui si desiderano gli aggiornamenti **automatici** di IBM MQ Operatore in cui le installazioni deterministiche non sono necessarie. Questa opzione può essere utile per ambienti di prova, ma **non è adatta per ambienti di produzione**.

Il IBM Operator Catalog è un indice di operatori disponibili per estendere l'API di un cluster Red Hat OpenShift Container Platform per abilitare i prodotti software IBM . L'aggiunta delle origini del catalogo al tuo cluster OpenShift aggiunge gli operatori IBM all'elenco di operatori che puoi installare.

Questa attività presuppone che siano stati completati i primi 3 passi di “Installazione del IBM MQ Operator” a pagina 31.

Questa attività può essere eseguita utilizzando la CLI o la console web OpenShift .

Utilizzo della CLI

1. Copiare la seguente definizione risorsa per gli operatori IBM in un file locale sul computer:

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
```

```
  name: ibm-operator-catalog
  namespace: openshift-marketplace
spec:
  displayName: IBM Operator Catalog
  publisher: IBM
  sourceType: grpc
  image: icr.io/cpopen/ibm-operator-catalog:latest
  updateStrategy:
    registryPoll:
      interval: 45m
```

2. Immetti il seguente comando. Sostituisci *filename*.yaml con il nome del file che hai creato nel passaggio precedente:

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

Utilizzo della console web OpenShift

1. Accedi alla console web OpenShift con le tue credenziali di amministratore del cluster OpenShift .
2. Nel banner, fai clic sul segno più ("+") per aprire la casella di dialogo **Importa YAML** .

   Nota: Non è necessario selezionare un valore per **Progetto**. Il codice YAML nel passo successivo include già il valore corretto per `metadata:namespace`, che garantisce che l'origine del catalogo sia installata nel progetto corretto (spazio dei nomi).
3. Incollare la seguente definizione della risorsa nella finestra:

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

4. Fai clic su **Crea**.

Si è ora pronti a completare il Passo 5 dell'installazione di IBM MQ Operator.

OpenShift CP4I *Installazione di IBM MQ Operator utilizzando la console OpenShift*

IBM MQ Operator può essere installato su Red Hat OpenShift utilizzando OperatorHub.

Prima di iniziare

Questa attività presuppone che siano stati completati i passi da 1 a 4 di “Installazione del IBM MQ Operator” a pagina 31.

Procedura

1. Accedi alla tua console del cluster Red Hat OpenShift .
2. Dal riquadro di navigazione, fare clic su **Operatori** > **OperatorHub**.
   Viene visualizzata la pagina OperatorHub .
3. Nel campo **Tutti gli elementi** , immettere "IBM MQ".
   Viene visualizzata la voce di catalogo IBM MQ .
4. Selezionare **IBM MQ**.
   Viene visualizzata la finestra IBM MQ .
5. Fai clic su **Installa**.
   Viene visualizzata la pagina Operatore di installazione.

6. Immetti i seguenti valori:
   a) Impostare **Canale** sulla versione scelta.

      Consultare “Supporto versione per IBM MQ Operator” a pagina 13 per stabilire quale canale operatore scegliere.

   b) Impostare la **Modalità di installazione** su "uno spazio dei nomi specifico sul cluster" (che è possibile creare nel passo successivo) o sull'ambito a livello di cluster.

      Si consiglia di scegliere l'ambito a livello di cluster, poiché l'installazione di versioni differenti di un operatore in spazi dei nomi differenti può causare problemi. Gli operatori sono progettati per essere estensioni del piano di controllo.

   c) Opzionale: Se si sceglie "uno spazio dei nomi specifico sul cluster", impostare lo **Spazio dei nomi** sul valore del progetto (spazio dei nomi) in cui si desidera installare l'operatore.

      Nota: Quando si utilizza la console per installare l'operatore, è possibile utilizzare uno spazio dei nomi esistente, lo spazio dei nomi predefinito fornito dall'operatore o creare un nuovo spazio dei nomi. Se si desidera creare un nuovo spazio dei nomi è possibile crearlo da questo modulo, come segue: dal riquadro di navigazione, fare clic su **Home** > **Progetti**, selezionare **Crea progetto**, specificare il **Nome** del progetto (lo spazio dei nomi) che si desidera creare, quindi fare clic su **Crea**.

   d) Impostare **Strategia di approvazione** su Automatico.
7. Fare clic su **Installa** e attendere l'installazione dell'operatore.

   Viene fornita una conferma al termine dell'installazione.

   Per verificare l'installazione, passare a **Operatori** > **Operatori installati**e selezionare il proprio progetto dall'elenco a discesa **Progetti** . Lo stato dell'operatore cambia in Riuscito quando l'installazione è completa.

Operazioni successive

Sei ora pronto a Creare il segreto della chiave di titolarità (passo 6 di “Installazione del IBM MQ Operator” a pagina 31).

OpenShift CP4I *Installazione di IBM MQ Operator utilizzando la CLI di Red Hat OpenShift*

Il IBM MQ Operator può essere installato su Red Hat OpenShift utilizzando la CLI (command line interface).

Prima di iniziare

Questa attività presuppone che siano stati completati i passi da 1 a 4 di “Installazione del IBM MQ Operator” a pagina 31.

Procedura

1. Accedi alla CLI (command line interface) Red Hat OpenShift utilizzando **`oc login`**.
2. Opzionale: Crea uno spazio dei nomi da utilizzare per IBM MQ Operator.

   Il IBM MQ Operator può essere installato nell'ambito di un singolo spazio dei nomi o di tutti gli spazi dei nomi. Questa operazione è necessaria solo se si desidera eseguire l'installazione in un determinato spazio dei nomi che non esiste già.

   Per creare un nuovo spazio dei nomi nella CLI, esegui questo comando:

   ```
   oc create namespace namespace_name
   ```

   Dove *namespace_name* è il nome dello spazio dei nomi che si desidera creare.
3. Visualizza l'elenco di operatori disponibili al cluster da OperatorHub:

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

4. Esaminare IBM MQ Operator per verificarne il **InstallModes** supportato e disponibile **Channels**.

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

5. Opzionale: Creare un **OperatorGroup**.

   Un **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 alla modalità **InstallMode**dell'operatore, `AllNamespaces` o `SingleNamespace` .

   Se l'operatore che si desidera installare utilizza la modalità `AllNamespaces` , lo spazio dei nomi `openshift-operators` dispone già di un **OperatorGroup** appropriato ed è possibile ignorare questo passo.

   Se l'operatore utilizza la modalità `SingleNamespace` e non si dispone già di un **OperatorGroup** appropriato, crearne uno immettendo il seguente comando:

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

6. Consultare “Supporto versione per IBM MQ Operator” a pagina 13 per stabilire quale canale operatore scegliere.
7. Installare l'operatore.

   Utilizza il seguente comando, modificando *ibm - mq - operator - channel* per corrispondere al canale per la versione dell'operatore IBM MQ che vuoi installare e modificando *namespace_name* in **openshift-operators** se stai utilizzando la modalità "AllNamespaces" o allo spazio dei nomi a cui vuoi distribuire l'operatore IBM MQ se stai utilizzando la modalità "SingleNamespace".

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

8. Dopo pochi minuti, l'operatore viene installato. Eseguire il seguente comando per verificare che tutti i componenti siano nello stato Riuscito:

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

   Dove *nome_spazio dei nomi* è **openshift-operators** se si utilizza la modalità "AllNamespaces" o il nome del progetto (spazio dei nomi) se si utilizza la modalità "SingleNamespace".

Operazioni successive

Sei ora pronto a Creare il segreto della chiave di titolarità (passo 6 di “Installazione del IBM MQ Operator” a pagina 31).

Installazione di IBM MQ Operator per l'utilizzo con CP4I

Per l'utilizzo con IBM Cloud Pak for Integration (CP4I), IBM MQ Operator può essere installato su Red Hat OpenShift tramite la console OpenShift o la CLI (command line interface).

Prima di iniziare

Importante:

- Questo argomento è per l'installazione di IBM MQ Operator da utilizzare con CP4Io se si intende distribuire almeno uno dei gestori code utilizzando una CP4I licenza **solo**. Per istruzioni sull'installazione di IBM MQ Operator per uso autonomo, consultare “Installazione del IBM MQ Operator” a pagina 31.
- Esamina le istruzioni su strutturare la tua distribuzione prima di installare IBM MQ Operator.

Per assicurarsi che l'installazione avvenga nel modo più fluido possibile, accertarsi di aver compreso tutti i prerequisiti e i requisiti prima di avviare l'installazione. Consultare “Pianificazione per IBM MQ nei contenitori” a pagina 7.

Informazioni su questa attività

I seguenti passi rappresentano il tipico flusso di attività per l'installazione di IBM MQ Operator:

1. Installare Red Hat OpenShift Container Platform.
2. Configurare l'archiviazione.
3. Immagini mirror (solo air - gap).
4. Aggiungi il catalogo IBM MQ Operator e preparare il cluster.
5. Installa IBM MQ Operator.
6. Creare il segreto della chiave di titolarità (solo installazioni online).
7. Facoltativo: installare IBM Cloud Pak for Integration (CP4I) e le relative dipendenze.
8. Distribuire il servizio di licenza License Service.
9. Distribuire un gestore code.

Procedura

1. Installa Red Hat OpenShift Container Platform.

   Per la procedura dettagliata per l'installazione di OpenShift, consultare Installazione del software Red Hat 4.6 o successiva.

   Importante: Assicurarsi di installare una versione supportata di OpenShift Container Platform. Ad esempio, per utilizzare IBM MQ Operator 3.2 o versioni successive, è necessario installare OpenShift Container Platform 4.12 o versioni successive. Per ulteriori informazioni, vedi IBM Cloud Pak and Red Hat OpenShift Container Platform compatibility.

   Per qualsiasi passo che utilizza la CLI Red Hat OpenShift Container Platform , devi essere collegato al tuo cluster OpenShift con `oc login`. Per installare la CLI, vedi Introduzione alla CLI OpenShift.

   Dopo aver installato OpenShift, puoi verificare e ottenere l'accesso al tuo software del contenitore utilizzando la IBM chiave di titolarità che crei in Crea il segreto della chiave di titolarità.
2. Configurare la memoria.

   È necessario definire le classi di memoria in Red Hat OpenShift Container Platform e impostare la propria configurazione di memoria per soddisfare i propri requisiti di dimensione.

   Importante: I gestori code IBM MQ a istanza singola e HA nativa possono utilizzare la modalità di accesso RWO, mentre i gestori code a più istanze richiedono RWX come descritto in “Pianificazione dell'archiviazione per IBM MQ Operator” a pagina 15. I gestori code a più istanze IBM MQ richiedono particolari caratteristiche del file system, che possono essere verificate utilizzando le istruzioni per Verifica di un file system condiviso per IBM MQ.

Un elenco di file system conformi e non conformi noti e note su altri limiti o limitazioni, è disponibile nell' istruzione di test per i file system IBM MQ.

I provider di memoria consigliati possono essere trovati nella pagina CP4I Considerazioni sulla memoria .

3. Immagini speculari (solo air - gap).

   Se il cluster si trova in un ambiente di rete limitato (air-gapped), è necessario eseguire il mirroring delle immagini IBM MQ . A seconda della propria configurazione, potrebbe essere necessario anche eseguire il mirroring di alcuni componenti aggiuntivi. Leggere le seguenti informazioni, quindi eseguire il mirroring delle immagini come richiesto.

   - È necessario eseguire il mirroring delle immagini IBM MQ . Utilizzare i seguenti valori:

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

   - È inoltre necessario eseguire il mirroring di alcuni componenti aggiuntivi richiesti se si intende distribuire almeno un gestore code in cui **tutte** le seguenti istruzioni sono vere:
     - Si sta utilizzando una licenza CP4I .
     - IBM MQ Console è abilitato.
     - Si sta utilizzando il servizio IBM Cloud Pak for Integration Keycloak per l'autorizzazione e l'autenticazione SSO (single sign - on) IBM MQ Console (impostazione predefinita).

     Se tutte le istruzioni precedenti sono vere, SSO viene fornito da Keycloak. Pertanto, oltre che per l'origine del catalogo IBM MQ Operator , è necessario ripetere anche la procedura per ciascuno dei seguenti componenti aggiuntivi richiesti:

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

   Per creare immagini di mirroring, consultare Mirroring delle immagini per un cluster air - gapped.

4. Aggiungere l'origine del catalogo IBM MQ Operator .

   Aggiungi l'origine del catalogo che rende IBM MQ Operator disponibile al tuo cluster utilizzando i seguenti valori:

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

   dove *ARCHITECTURE* fa riferimento all'architettura del sistema e ha un valore `amd64`, `ppc64leo` `s390x`.

   Esistono alcuni componenti aggiuntivi richiesti quando si distribuisce almeno un gestore code in cui **tutte** le seguenti istruzioni sono vere:

   - Si sta utilizzando una licenza CP4I .
   - IBM MQ Console è abilitato.
   - Si sta utilizzando il servizio IBM Cloud Pak for Integration Keycloak per l'autorizzazione e l'autenticazione SSO (single sign - on) IBM MQ Console (impostazione predefinita).

   Se tutte le istruzioni precedenti sono vere, SSO viene fornito da Keycloak. Pertanto, oltre che per l'origine del catalogo IBM MQ Operator , è necessario ripetere anche la procedura per ciascuno dei seguenti componenti aggiuntivi richiesti:

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

   Segui la procedura per le tue origini di catalogo richieste in Aggiunta di origini di catalogo a un cluster.

5. Installare IBM MQ Operator.

   Scegli una delle seguenti due opzioni (utilizza la console o utilizza la CLI):

   - Opzione 1: Installare IBM MQ Operator utilizzando la console OpenShift.
   - Opzione 2: Installa IBM MQ Operator utilizzando la CLI OpenShift.

6. Creare il segreto della chiave di titolarità (solo installazioni online).

   IBM MQ Operator distribuisce le immagini del gestore code estratte da un registro contenitore che esegue un controllo di 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 nello spazio dei nomi in cui verranno installati i gestori code, seguire queste istruzioni per ottenere una chiave di titolarità e creare un segreto di pull.

   Nota: La chiave di titolarità non è richiesta se verranno distribuiti solo i gestori code IBM MQ Advanced for Developers (non Warranted).

   Puoi creare il segreto della chiave di titolarità utilizzando la console OpenShift o la CLI. Il seguente esempio utilizza la CLI:

   a. Ottieni la chiave di titolarità assegnata al tuo ID IBM . 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.
   c. Dalla CLI OpenShift , immetti il seguente comando per creare un segreto di pull dell'immagine denominato `ibm-entitlement-key`.

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

      Dove *entitlement_key* è la chiave di titolarità che hai copiato nel passo b, *user_email>* è l' IBM ID associato al software autorizzato e *namespace* è lo spazio dei nomi in cui hai installato il tuo IBM MQ Operator .

7. Opzionale: Installare CP4I e le relative dipendenze.

   Esistono alcuni componenti aggiuntivi richiesti quando si distribuisce almeno un gestore code in cui **tutte** le seguenti istruzioni sono vere:

   - Si sta utilizzando una licenza CP4I .
   - IBM MQ Console è abilitato.
   - Si sta utilizzando il servizio CP4I Keycloak per l'autorizzazione e l'autenticazione SSO (single sign - on) IBM MQ Console (impostazione predefinita).

   Se tutte le precedenti istruzioni sono true, SSO viene fornito da Keycloak ed è necessario completare le seguenti operazioni aggiuntive:

   - Installare l'operatore IBM Cloud Pak foundational services nella stessa modalità di installazione dell'operatore CP4I . Per le versioni supportate, vedi Versioni del canale operatore per questa release.
   - Installare l'operatore CP4I.
   - Facoltativo: distribuisci l'IU della piattaforma.

     a. Crea lo spazio dei nomi `ibm-common-services` . Quando accedi al tuo cluster OpenShift tramite la CLI, immetti il seguente comando:

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

     b. Distribuisci la IU della piattaforma.

8. Distribuire License Service.

   Ciò è necessario per il monitoraggio dell'utilizzo della licenza dei gestori code. Seguire le istruzioni riportate in Distribuzione del servizio di licenza License Service.
9. Distribuire un gestore code.

   Per istruzioni sulla distribuzione di un gestore code di esempio "avvio rapido" , consultare “Distribuzione di un gestore code semplice utilizzando IBM MQ Operator” a pagina 62.

Attività correlate

“Disinstallazione di IBM MQ Operator” a pagina 52
Puoi utilizzare la console o la CLI Red Hat OpenShift per disinstallare IBM MQ Operator da Red Hat OpenShift.

OpenShift CP4I Operator2.0.0 Aggiornamento di IBM MQ Operator e dei gestori code

Esistono diversi processi di aggiornamento per gli utenti di IBM MQ Operator, a seconda se si utilizzano licenze IBM MQ o licenze IBM Cloud Pak for Integration (CP4I). Completare il passo di aggiornamento per il tipo di distribuzione.

Informazioni su questa attività

Per aggiornare IBM MQ Operator e i gestori code, completare una delle seguenti operazioni:

Procedura

- Opzione 1: **aggiornare le distribuzioni alla versione più recente sul tuo canale Operatore corrente.**

  Per aggiornare le distribuzioni di IBM MQ Operator alla versione più recente sul tuo canale Operatore corrente, vedi “Aggiornamento a un'ultima release di sicurezza del canale IBM MQ Operator” a pagina 43.
- Opzione 2: **Aggiornare le licenze IBM MQ Operator per IBM MQ .**

  Per aggiornare le distribuzioni di IBM MQ Operator in cui vengono utilizzate **solo** IBM MQ licenze, consultare “Aggiornamento di IBM MQ Operator” a pagina 42.
- Opzione 3: **Aggiornare IBM MQ Operator per gli utenti CP4I .**

  Aggiornare le distribuzioni di IBM MQ Operator per utenti di IBM Cloud Pak for Integration. Ciò include se è stato distribuito almeno uno dei gestori code con una licenza CP4I . Consultare “Aggiornamento degli utenti IBM MQ Operator per CP4I” a pagina 47.

OpenShift *Aggiornamento di IBM MQ Operator*

Aggiornare le distribuzioni di IBM MQ Operator in cui vengono utilizzate **solo** IBM MQ licenze.

Prima di iniziare

Importante: Questa attività è riservata agli utenti delle licenze IBM MQ Operator e **solo** IBM MQ . Se si è un utente IBM Cloud Pak for Integration (CP4I) o se è stato distribuito almeno uno dei gestori code utilizzando una licenza CP4I , consultare “Aggiornamento degli utenti IBM MQ Operator per CP4I” a pagina 47.

Informazioni su questa attività

Completare la procedura seguente in base all'aggiornamento necessario.

Nota: La versione 3.2.x di IBM MQ Operator è stata rilasciata sia come release CD che come release SC2 .

Procedura

- Opzione 1:“Aggiornamento a un'ultima release di sicurezza del canale IBM MQ Operator” a pagina 43
- Opzione 2:“Aggiornamento di un LTS IBM MQ Operator 2.0.x al canale 3.2.x SC2/CD” a pagina 44
- Opzione 3: “Aggiornamento di un CD IBM MQ Operator al canale /CD 3.2.x SC2” a pagina 45

Aggiornamento a un'ultima release di sicurezza del canale IBM MQ Operator

L'aggiornamento di IBM MQ Operator consente di aggiornare i propri gestori code.

Prima di iniziare

Importante: Questo argomento è per aggiornare le distribuzioni di IBM MQ Operator all'ultima release di sicurezza sul canale della distribuzione. Se non si applica alla propria distribuzione, fare riferimento ai percorsi di aggiornamento alternativi descritti in “Aggiornamento di IBM MQ Operator e dei gestori code” a pagina 42.

Informazioni su questa attività

Aggiornare prima l'origine del catalogo, quindi i gestori code. Esistono due opzioni, a seconda dell'origine del catalogo utilizzata per distribuire il IBM MQ Operator che si sta aggiornando.

Opzione 1: Origine catalogo specifica per IBM MQ Operator

Una nuova versione di IBM MQ Operator diventa disponibile **solo** in un cluster OpenShift dopo l'aggiornamento dell'origine del catalogo. Questo processo fornisce in modo efficace il controllo manuale degli aggiornamenti, quindi non è necessario utilizzare l'opzione **Manuale** per l'impostazione **`Update approval`** per gli operatori. L'opzione **Manuale** forza l'esecuzione di tutti i possibili aggiornamenti contemporaneamente e può bloccare gli aggiornamenti, quindi utilizzare solo l'opzione **Automatico** . Per ulteriori informazioni, consulta la sezione "Limitazione degli aggiornamenti automatici con una strategia di approvazione" di Installazione degli operatori utilizzando la console Red Hat OpenShift.

Per utilizzare questa opzione, passare a Aggiorna con l'origine del catalogo specifica per IBM MQ Operator.

Opzione 2: IBM Operator Catalog

Con questa opzione, le nuove versioni dell'operatore diventano disponibili e vengono applicate **senza** alcun intervento da parte dell'utente. Quindi, utilizzare questa opzione **solo** per le installazioni in linea in cui si desiderano aggiornamenti **automatici** di IBM MQ Operatore in cui non sono necessarie installazioni deterministiche. Questa opzione può essere utile per ambienti di prova, ma **non è adatta per ambienti di produzione**.

Per utilizzare questa opzione, vai a Upgrade with the IBM Operator Catalog.

Per passare dall'utilizzo del Catalogo operatore IBM all'utilizzo dell'origine del catalogo specifico per IBM MQ Operator, che fornisce un maggiore controllo sugli aggiornamenti, consultare “Passaggio all'origine del catalogo specifico per IBM MQ Operator” a pagina 46.

Procedura

- **Aggiornamento con l'origine del catalogo specifica per IBM MQ Operator**
  a) Applica l'ultima origine del catalogo.

  Seguire le istruzioni in "Aggiungi origini di catalogo specifiche per IBM MQ Operator" in Aggiunta dell'origine di catalogo IBM MQ Operator.

  b) Se si dispone dello stato **Approvazione aggiornamento** per IBM MQ Operator impostato su **Automatico**, l'operatore esegue l'aggiornamento. Se l' **Approvazione aggiornamento** è impostata su **Manuale**, attenersi alla seguente procedura per aggiornare IBM MQ Operator:

  a. Dal riquadro di navigazione, fare clic su **Operatori** > **Operatori installati**.

Vengono visualizzati tutti gli operatori installati nel progetto specificato.

b. Selezionare **Operatore IBM MQ**

c. Passare alla scheda **Sottoscrizione**

d. Fare clic su **Aggiorna disponibile**

e. Fare clic su **Anteprima InstallPlan**

f. Fare clic su **Approva** per completare l'aggiornamento.

L'operatore esegue l'aggiornamento alla nuova versione.

c) Aggiornare i gestori code IBM MQ .

Procedere con le istruzioni in Aggiorna IBM MQ gestori code.

- **Aggiornamento con il IBM Catalogo operatore**

a) Aggiornare IBM MQ Operator a una versione più recente.

Se si dispone di aggiornamenti automatici impostati, al rilascio di un nuovo rilascio di sicurezza IBM MQ Operator completa un aggiornamento. Se non si dispone di aggiornamenti automatici impostati, approvare manualmente l'aggiornamento IBM MQ Operator :

– Se è disponibile un upgrade, **`Upgrade Status`** potrebbe essere "`Upgrade available`".

– In questo caso, potrebbe essere disponibile un controllo che è possibile utilizzare per approvare il **InstallPlan** che aggiorna il IBM MQ Operator.

b) Aggiornare qualsiasi gestore code IBM MQ

Procedere con le istruzioni in Aggiorna IBM MQ gestori code.

- **Aggiorna IBM MQ gestori code**.

È necessario aggiornare qualsiasi gestore code IBM MQ a una versione più recente dopo l'aggiornamento di IBM MQ Operator.

La seguente tabella descrive la versione più recente del gestore code IBM MQ per ogni canale operatore attivo. Utilizzando la relativa versione, seguire la procedura riportata in “Aggiornamento di un gestore code IBM MQ utilizzando Red Hat OpenShift” a pagina 50.

Canale operatore	Gestore code IBM MQ più recente
v3.2 (SC2/CD)	9.4.0.0-r1

OpenShift V 9.4.0 CP4I Operator2.0.0 *Aggiornamento di un LTS IBM MQ Operator 2.0.x al canale 3.2.x SC2/CD*

L'aggiornamento di IBM MQ Operator consente di aggiornare i propri gestori code.

Prima di iniziare

Importante:

- Questa attività è riservata agli utenti delle licenze IBM MQ Operator e **solo** IBM MQ . Se si è un utente IBM Cloud Pak for Integration (CP4I) o se è stato distribuito almeno uno dei gestori code utilizzando una licenza CP4I , consultare “Aggiornamento degli utenti IBM MQ Operator per CP4I” a pagina 47.
- Questo argomento è per l'aggiornamento delle distribuzione di 2.0.x Long Term Support (LTS) IBM MQ Operator al canale Support Cycle 2 (SC2) di IBM MQ Operator 3.2.x **solo**. Se ciò non si applica alla propria distribuzione, consultare i percorsi di aggiornamento alternativi descritti in “Aggiornamento di IBM MQ Operator e dei gestori code” a pagina 42.

Per eseguire l'aggiornamento a IBM MQ Operator 3.2.1 , è necessario che sia in esecuzione Red Hat OpenShift Container Platform 4.12 o versione successiva. Per verificare le versioni compatibili per ogni canale IBM MQ Operator , consultare “Versioni Red Hat OpenShift Container Platform compatibili” a pagina 14. Per aggiornare la piattaforma, consultare Aggiornamento di Red Hat OpenShift.

Procedura

1. **Immagini speculare (solo air - gap).**

   È necessario eseguire il mirroring delle immagini IBM MQ . Completare la procedura al seguente link, utilizzando solo questi valori:

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

   Dovresti omettere la sezione 3.5 "Configura il cluster", perché la connessione al registro delle immagini dovrebbe essere stata impostata durante le installazioni o gli aggiornamenti precedenti.

   Link: Mirroring delle immagini per un cluster air - gapped.
2. **Aggiornare IBM MQ Operator a 3.2.1**

   Consultare “Aggiornamento di IBM MQ Operator utilizzando Red Hat OpenShift” a pagina 48.
3. **Aggiorna le istanze.**

   Per ricevere le funzioni e le correzioni di sicurezza più recenti, aggiornare IBM MQ Operando (Immagine Contenitore gestore code) all'ultima versione CD (9.4.0.0-r1). Consultare “Aggiornamento di un gestore code IBM MQ utilizzando Red Hat OpenShift” a pagina 50.

Aggiornamento di un CD IBM MQ Operator al canale /CD 3.2.x SC2

L'aggiornamento di IBM MQ Operator consente di aggiornare i propri gestori code.

Prima di iniziare

Importante:

- Questa attività è riservata agli utenti delle licenze IBM MQ Operator e **solo** IBM MQ . Se si è un utente IBM Cloud Pak for Integration (CP4I) o se è stato distribuito almeno uno dei gestori code utilizzando una licenza CP4I , consultare “Aggiornamento degli utenti IBM MQ Operator per CP4I” a pagina 47.
- Questo argomento è per l'aggiornamento delle distribuzioni Continuous Delivery (CD) di IBM MQ Operator precedenti alla versione 3.2.0, alla versione 3.2.1 **solo**. Se ciò non si applica alla propria distribuzione, consultare i percorsi di aggiornamento alternativi descritti in “Aggiornamento di IBM MQ Operator e dei gestori code” a pagina 42.

Per eseguire l'aggiornamento a IBM MQ Operator 3.2.1 , è necessario che sia in esecuzione Red Hat OpenShift Container Platform 4.12 o versione successiva. Per verificare le versioni compatibili per ogni canale IBM MQ Operator , consultare “Versioni Red Hat OpenShift Container Platform compatibili” a pagina 14. Per aggiornare la piattaforma, consultare Aggiornamento di Red Hat OpenShift.

Procedura

1. Opzionale: **Aggiornare un IBM MQ Operator che è attualmente alla versione CD precedente a 3.0.0.**

   Se il tuo IBM MQ Operator è attualmente a una versione CD precedente a 3.0.0, segui i passi pertinenti in Migrazione al canale CD corrente dell'operatore IBM MQ (IBM MQ 9.3 documentation), quindi torna qui per eseguire l'aggiornamento all'ultima versione di CD . Notare che questo è un passo prerequisito obbligatorio prima di eseguire l'aggiornamento alla versione 3.2.1.
2. **Immagini speculare (solo air - gap).**

   È necessario eseguire il mirroring delle immagini IBM MQ . Completare la procedura al seguente link, utilizzando solo questi valori:

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

Dovresti omettere la sezione 3.5 "Configura il cluster", perché la connessione al registro delle immagini dovrebbe essere stata impostata durante le installazioni o gli aggiornamenti precedenti.

Link: Mirroring delle immagini per un cluster air - gapped.

3. **Aggiornare IBM MQ Operator a 3.2.1**

   Consultare “Aggiornamento di IBM MQ Operator utilizzando Red Hat OpenShift” a pagina 48.

4. **Aggiornare le istanze.**

   Per ricevere le funzioni e le correzioni di sicurezza più recenti, aggiornare IBM MQ Operando (Immagine Contenitore gestore code) all'ultima versione CD (9.4.0.0-r1). Consultare “Aggiornamento di un gestore code IBM MQ utilizzando Red Hat OpenShift” a pagina 50.

Passaggio all'origine del catalogo specifico per IBM MQ Operator

Se si dispone di un'installazione di IBM MQ Operator da una release precedente e si sta utilizzando IBM Operator Catalog, l'applicazione dell'origine del catalogo specifica è il modo più efficace per controllare completamente la versione del software su un cluster.

Prima di iniziare

Importante: Questa attività deve essere eseguita dall'amministratore del cluster. Vedi Ruoli e autorizzazioniOpenShift.

I seguenti passi vengono completati utilizzando la CLI.

Informazioni su questa attività

Il IBM Operator Catalog è un indice di operatori disponibili per estendere l'API di un cluster Red Hat OpenShift Container Platform per abilitare i prodotti software IBM .

Questa procedura sposta un'installazione di IBM MQ Operator dal catalogo operatore IBM in modo da poter utilizzare l'origine del catalogo specifica per IBM MQ Operator.

Procedura

1. Aggiungi il catalogo IBM MQ Operator .

   Seguire le istruzioni in "Aggiungi origini di catalogo specifiche per IBM MQ Operator" in Aggiunta dell'origine di catalogo IBM MQ Operator.

2. Conferma che l'origine del catalogo IBM MQ Operator è stata creata nel namespace `openshift-marketplace` .

   Esegui il seguente comando:

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

   Output di esempio:

   ```
   oc get catalogsource -n openshift-marketplace

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

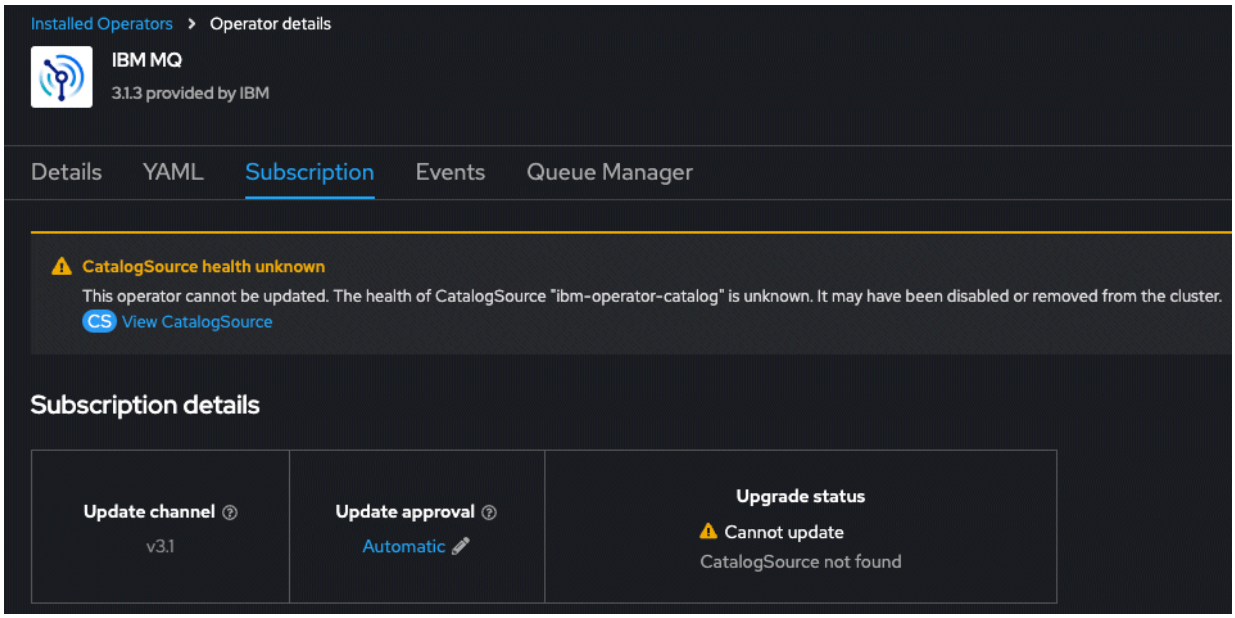
3. Opzionale: Eliminare l'origine del catalogo operatore IBM .

   Avvertenza: È necessario completare questo passo solo se si è certi che non ci sono altri operatori che utilizzano il catalogo degli operatori IBM .

   Esegui il seguente comando:

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

Lo stato di IBM MQ Operator cambia in `CatalogSource not found`. Ciò è previsto.



4. Modificare la sottoscrizione di IBM MQ Operator in modo che punti alla nuova origine di catalogo IBM MQ Operator specifica.
   a) Modificare la sottoscrizione.

      Immetti il seguente comando, sostituendo *OPERATOR - NAMESPACE* con `openshift-operators` per le installazioni a livello di cluster di IBM MQ Operatoro con lo spazio dei nomi specifico in cui è distribuito IBM MQ Operator :

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

   b) Modificare il valore `spec.source` da `ibm-operator-catalog` al nome dell'origine del catalogo creata nel passo “1” a pagina 46.

      Ad esempio:

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

   c) Salvare le modifiche.

      L'installazione di IBM MQ Operator ora punta all'origine del catalogo IBM MQ Operator . Se è stato eliminato il IBM Catalogo operatore, lo stato viene ripristinato da "CatalogSource non trovato" a "Riuscito".

Risultati

L'installazione di IBM MQ Operator ora punta all'origine del catalogo specifico per IBM MQ Operator. Questo ti dà il pieno controllo sugli aggiornamenti per l'operatore.

OpenShift CP4I *Aggiornamento degli utenti IBM MQ Operator per CP4I*

Aggiornare le distribuzioni di IBM MQ Operator in cui viene utilizzata una licenza IBM Cloud Pak for Integration (CP4I).

Prima di iniziare

Importante: Questa attività è per gli utenti CP4I . Ciò include se è stato distribuito almeno uno dei gestori code con una licenza CP4I . Se questo non si applica all'utente, consultare “Aggiornamento di IBM MQ Operator” a pagina 42.

Informazioni su questa attività

Completare una delle seguenti opzioni:

Procedura

- **Opzione 1:** Aggiornare le distribuzioni di 2.0.x Long Term Support (LTS) IBM MQ Operator

  Segui la procedura in Aggiornamento da 2022.2 generando un piano di upgrade.
- **Opzione 2:** Aggiornare una distribuzione 3.0.x o 3.1.x di IBM MQ Operator

  Attieniti alla procedura in Upgrading from 2023.4 generando un piano di upgrade.
- **Opzione 3:** aggiornare altre distribuzioni di IBM MQ Operator

  Seguire i passi pertinenti in Migrazione al canale CD corrente della IBM MQ Operator (IBM MQ 9.3 documentation), quindi tornare qui e procedere con **Opzione 2**. Si noti che questo è un passo prerequisito obbligatorio.

OpenShift CP4I *Aggiornamento di IBM MQ Operator utilizzando Red Hat OpenShift*

Puoi eseguire l'upgrade di IBM MQ Operator utilizzando la CLI o la console web Red Hat OpenShift .

Procedura

Per aggiornare IBM MQ Operator utilizzando Red Hat OpenShift, completare una delle seguenti attività:

- “Aggiornamento di IBM MQ Operator utilizzando la console Red Hat OpenShift” a pagina 48
- “Aggiornamento di IBM MQ Operator utilizzando la CLI di Red Hat OpenShift” a pagina 49

OpenShift CP4I *Aggiornamento di IBM MQ Operator utilizzando la console Red Hat OpenShift*

Il IBM MQ Operator può essere aggiornato utilizzando l'hub dell'operatore.

Prima di iniziare

Nota: L'ultima versione CD di IBM MQ Operator è 3.2.1ed è sia una versione SC2 che CD . Per le ultime note sulla release IBM MQ Operator , consultare Release history for IBM MQ Operator.

Accedi alla tua console del cluster Red Hat OpenShift .

Procedura

1. Esaminare “Supporto versione per IBM MQ Operator” a pagina 13 per determinare a quale canale operatore eseguire l'aggiornamento.
2. Applica l'ultima origine del catalogo.

   Se si utilizza l'origine del catalogo specifica per IBM MQ Operator, piuttosto che `ibm-operator-catalog`, è necessario applicare l'origine del catalogo per la nuova versione di IBM MQ .

   Per passare dall'utilizzo del IBM Catalogo operatore all'utilizzo dell'origine del catalogo specifico per il IBM MQ Operatore ottenere un maggiore controllo sugli aggiornamenti, fare riferimento alla procedura in “Passaggio all'origine del catalogo specifico per IBM MQ Operator” a pagina 46 prima di ritornare al passo completo “3” a pagina 49.

Se si utilizza il catalogo degli operatori IBM (solo alcune installazioni in linea), procedere con il passo “3” a pagina 49.

Seguire le istruzioni in “Aggiunta dell'origine del catalogo IBM MQ Operator” a pagina 33.

3. Aggiornare IBM MQ Operator. Le nuove versioni principali / secondarie di IBM MQ Operator vengono fornite tramite i nuovi canali di sottoscrizione. Per aggiornare il tuo Operatore ad una nuova versione principale / secondaria, dovrai aggiornare il canale selezionato nella tua sottoscrizione IBM MQ Operator .
   a) Dal riquadro di navigazione, fare clic su **Operatori** > **Operatori installati**.
      Vengono visualizzati tutti gli operatori installati nel progetto specificato.
   b) Selezionare **Operatore IBM MQ**
   c) Passare alla scheda **Sottoscrizione**
   d) Fare clic su **Canale**
      Viene visualizzata la finestra **Modifica canale di aggiornamento sottoscrizione** .
   e) Selezionare il canale desiderato e fare clic su **Salva**.
      L'operatore eseguirà l'aggiornamento alla versione più recente disponibile per il nuovo canale. Consultare “Supporto versione per IBM MQ Operator” a pagina 13.

OpenShift CP4I *Aggiornamento di IBM MQ Operator utilizzando la CLI di Red Hat OpenShift*

IBM MQ Operator può essere aggiornato dalla riga di comando.

Prima di iniziare

Nota: L'ultima versione CD di IBM MQ Operator è 3.2.1ed è sia una versione SC2 che CD . Per le ultime note sulla release IBM MQ Operator , consultare Release history for IBM MQ Operator.

Accedere al cluster utilizzando **`oc login`**.

Procedura

1. Esaminare “Supporto versione per IBM MQ Operator” a pagina 13 per determinare a quale canale operatore eseguire l'aggiornamento.
2. Applica l'ultima origine del catalogo.

   Se si utilizza l'origine del catalogo specifica per IBM MQ Operator, piuttosto che `ibm-operator-catalog`, è necessario applicare l'origine del catalogo per la nuova versione di IBM MQ .

   Per passare dall'utilizzo del IBM Catalogo operatore all'utilizzo dell'origine del catalogo specifico per il IBM MQ Operatore ottenere un maggiore controllo sugli aggiornamenti, fare riferimento alla procedura in “Passaggio all'origine del catalogo specifico per IBM MQ Operator” a pagina 46 prima di ritornare al passo completo “3” a pagina 49.

   Se si utilizza il catalogo degli operatori IBM (solo alcune installazioni in linea), procedere con il passo “3” a pagina 49.

   Seguire le istruzioni in “Aggiunta dell'origine del catalogo IBM MQ Operator” a pagina 33.
3. Aggiornare IBM MQ Operator. Le nuove versioni principali / secondarie di IBM MQ Operator vengono fornite tramite i nuovi canali di sottoscrizione. Per aggiornare l'operatore a una nuova versione principale o secondaria, sarà necessario aggiornare il canale selezionato nella sottoscrizione IBM MQ Operator .
   a) Assicurarsi che il canale di aggiornamento IBM MQ Operator richiesto sia disponibile.

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

   b) Applicare una patch a `Subscription` per passare al canale di aggiornamento desiderato (dove *vX.Y* è il canale di aggiornamento desiderato identificato nel passo precedente.

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

OpenShift CP4I *Aggiornamento di un gestore code IBM MQ utilizzando Red Hat OpenShift*

Prima di iniziare

Come parte del processo di aggiornamento dei gestori code IBM MQ , è possibile che l'utente sia stato inviato a questo argomento dalla documentazione IBM Cloud Pak for Integration .

Procedura

Per aggiornare il gestore code IBM MQ utilizzando Red Hat OpenShift, completare una delle seguenti attività:

- “Aggiornamento di un gestore code IBM MQ utilizzando la console Red Hat OpenShift” a pagina 50
- “Aggiornamento di un gestore code IBM MQ utilizzando la CLI Red Hat OpenShift” a pagina 51
- “Aggiornamento di un gestore code IBM MQ in Red Hat OpenShift utilizzando l'IU della piattaforma” a pagina 51

Operazioni successive

Per completare un aggiornamento IBM Cloud Pak for Integration , potrebbe essere necessario tornare alla documentazione di IBM Cloud Pak for Integration .

OpenShift CP4I *Aggiornamento di un gestore code IBM MQ utilizzando la console Red Hat OpenShift*

Un gestore code IBM MQ , distribuito utilizzando IBM MQ Operator, può essere aggiornato in Red Hat OpenShift utilizzando l'hub operatore.

Prima di iniziare

Nota: La versione più recente del gestore code IBM MQ è 9.4.0.0-r1ed è sia una versione SC2 che CD . Per le ultime note sulla release del gestore code IBM MQ , consultare Release history for Queue Manager Container images for use with IBM MQ Operator.

- Accedi alla tua console web del cluster Red Hat OpenShift .
- Assicurarsi che IBM MQ Operator stia utilizzando il canale di aggiornamento desiderato. Vedere “Aggiornamento di IBM MQ Operator utilizzando Red Hat OpenShift” a pagina 48.

Prima di poter aggiornare il gestore code in un ambiente air - gap, è necessario eseguire il mirroring delle immagini IBM Cloud Pak for Integration più recenti mediante la procedura specifica di air - gap in Aggiornamento di un CD IBM MQ Operator al canale 3.2.x SC2/CD.

Procedura

1. Dal riquadro di navigazione, fare clic su **Operatori** > **Operatori installati**.
   Vengono visualizzati tutti gli operatori installati nel progetto specificato.
2. Selezionare **Operatore IBM MQ**.
   Viene visualizzata la finestra **Operatore IBM MQ** .
3. Passare alla scheda **Gestore code** .
   Viene visualizzata la finestra **Dettagli gestore code** .
4. Selezionare il gestore code che si desidera aggiornare.
5. Passare alla scheda YAML.

6. Aggiornare i seguenti campi, se necessario, in modo che corrispondano all'aggiornamento della versione del gestore code IBM MQ desiderato.
   - spec.version
   - spec.license.licence

   Consultare “Cronologia delle release per le immagini del contenitore del gestore code da utilizzare con IBM MQ Operator” a pagina 7 per un'associazione di versioni IBM MQ Operator e immagini del contenitore del gestore code IBM MQ .
7. Salvare l'YAML del gestore code aggiornato.

OpenShift CP4I *Aggiornamento di un gestore code IBM MQ utilizzando la CLI Red Hat OpenShift*

Un gestore code IBM MQ , distribuito utilizzando IBM MQ Operator, può essere aggiornato in Red Hat OpenShift utilizzando la riga comandi.

Prima di iniziare

Nota: La versione più recente del gestore code IBM MQ è 9.4.0.0-r1ed è sia una versione SC2 che CD . Per le ultime note sulla release del gestore code IBM MQ , consultare Release history for Queue Manager Container images for use with IBM MQ Operator.

Per completare queste operazioni, è necessario essere un amministratore del cluster.

- Accedi alla CLI (command line interface) Red Hat OpenShift utilizzando `oc login`.
- Assicurarsi che IBM MQ Operator stia utilizzando il canale di aggiornamento desiderato. Vedere “Aggiornamento di IBM MQ Operator e dei gestori code” a pagina 42.

Prima di poter aggiornare il gestore code in un ambiente air - gap, è necessario eseguire il mirroring delle immagini IBM Cloud Pak for Integration più recenti mediante la procedura specifica di air - gap in Aggiornamento di un CD IBM MQ Operator al canale 3.2.x SC2/CD.

Procedura

Modificare la risorsa **QueueManager** per aggiornare i seguenti campi, se necessario, in modo che corrispondano all'aggiornamento della versione del gestore code IBM MQ desiderato.

- spec.version
- spec.license.licence

Consultare “Supporto versione per IBM MQ Operator” a pagina 13 per un'associazione di canali alle versioni IBM MQ Operator e alle versioni del gestore code IBM MQ .

Utilizzare il seguente comando:

```
oc edit queuemanager my_qmgr
```

dove *my_qmgr* è il nome della risorsa QueueManager che si desidera aggiornare.

CP4I *Aggiornamento di un gestore code IBM MQ in Red Hat OpenShift utilizzando l'IU della piattaforma*

Un gestore code IBM MQ , distribuito utilizzando IBM MQ Operator, può essere aggiornato in Red Hat OpenShift utilizzando IBM Cloud Pak for Integration Platform UI.

Prima di iniziare

Nota: La versione più recente del gestore code IBM MQ è 9.4.0.0-r1ed è sia una versione SC2 che CD . Per le ultime note sulla release del gestore code IBM MQ , consultare Release history for Queue Manager Container images for use with IBM MQ Operator.

- Accedere al IBM Cloud Pak for Integration Platform UI nello spazio dei nomi che contiene il gestore code che si desidera aggiornare.

- Assicurarsi che IBM MQ Operator stia utilizzando il canale di aggiornamento desiderato. Vedere “Aggiornamento di IBM MQ Operator e dei gestori code” a pagina 42.

Prima di poter aggiornare il gestore code in un ambiente air - gap, è necessario eseguire il mirroring delle immagini IBM Cloud Pak for Integration più recenti mediante la procedura specifica di air - gap in Aggiornamento di un CD IBM MQ Operator al canale 3.2.x SC2/CD.

Procedura

1. Dalla home page IBM Cloud Pak for Integration Platform UI , fare clic sulla scheda **Runtime** .
2. I gestori code con aggiornamenti disponibili hanno una **i** blu accanto a **Versione**. Fare clic su **i** per visualizzare **Nuova versione disponibile**.
3. Fare clic sui tre punti all'estrema destra del gestore code che si desidera aggiornare, quindi fare clic su **Modifica versione**.
4. In **Seleziona un nuovo canale o versione**, selezionare la versione di aggiornamento richiesta.
5. Fare clic su **Modifica versione**.

Risultati
Il gestore code è aggiornato.

OpenShift CP4I Disinstallazione di IBM MQ Operator

Puoi utilizzare la console o la CLI Red Hat OpenShift per disinstallare IBM MQ Operator da Red Hat OpenShift.

Procedura

- Opzione 1: disinstallazione di IBM MQ Operator con la console OpenShift .

  Nota: Se IBM MQ Operator è installato su tutti i progetti / namespace sul cluster, ripetere i passi da 2 a 6 della seguente procedura per ogni progetto in cui si desidera eliminare i gestori code.

  a) Accedi alla console web Red Hat OpenShift Container Platform con le credenziali di amministratore del cluster Red Hat OpenShift Container Platform .

  b) Modificare **Progetto** nello spazio dei nomi da cui si desidera disinstallare IBM MQ Operator. Selezionare lo spazio dei nomi dall'elenco a discesa **Progetto** .

  c) Nel riquadro di navigazione, fare clic su **Operatori** > **Operatori installati**.

  d) Fare clic sull'operatore **IBM MQ** .

  e) Selezionare la scheda **Gestori code** per visualizzare i gestori code gestiti da questo IBM MQ Operator.

  f) Eliminare uno o più gestori code.

  Si noti che, sebbene questi gestori code continuino ad essere in esecuzione, potrebbero non funzionare come previsto senza un IBM MQ Operator.

  g) Opzionale: Se appropriato, ripetere i passi da 2 a 6 per ogni progetto in cui si desidera eliminare i gestori code.

  h) Tornare a **Operatori** > **Operatori installati**.

  i) Accanto all'operatore **IBM MQ** , fare clic sul menu a tre punti e selezionare **Disinstalla operatore**.
- Opzione 2: disinstallare IBM MQ Operator con la CLI OpenShift

  a) Accedi al tuo cluster Red Hat OpenShift utilizzando `oc login`.

  b) Se IBM MQ Operator è installato in un singolo spazio dei nomi, completare i seguenti passi secondari:

a. Assicurarsi di trovarsi nel progetto contenente il IBM MQ Operator da disinstallare:

```
oc project project_name
```

b. Visualizzare i gestori code installati nel progetto:

```
oc get qmgr
```

c. Eliminare uno o più gestori code:

```
oc delete qmgr qmgr_name
```

Si noti che, sebbene questi gestori code continuino ad essere in esecuzione, potrebbero non funzionare come previsto senza un IBM MQ Operator.

d. Visualizzare le istanze **ClusterServiceVersion** :

```
oc get csv
```

e. Eliminare il IBM MQ **ClusterServiceVersion**:

```
oc delete csv ibm_mq_csv_name
```

f. Visualizzare le sottoscrizioni:

```
oc get subscription
```

g. Elimina tutte le sottoscrizioni:

```
oc delete subscription ibm_mq_subscription_name
```

h. Se nessun altro utilizza i servizi comuni, è possibile disinstallare l'operatore dei servizi comuni ed eliminare il gruppo di operatori:

i) Disinstallare l'operatore dei servizi comuni, seguendo le istruzioni in Disinstallazione dei servizi di base nella documentazione del prodotto IBM Cloud Pak foundational services .

ii) Visualizzare il gruppo di operatori:

```
oc get operatorgroup
```

iii) Eliminare il gruppo di operatori:

```
oc delete OperatorGroup operator_group_name
```

c) Se il IBM MQ Operator è installato e disponibile per tutti gli spazi dei nomi sul cluster, completare i seguenti passi secondari:

a. Visualizzare tutti i gestori code installati:

```
oc get qmgr -A
```

b. Eliminare uno o più gestori code:

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

Si noti che, sebbene questi gestori code continuino ad essere in esecuzione, potrebbero non funzionare come previsto senza un IBM MQ Operator.

c. Visualizzare le istanze **ClusterServiceVersion** :

```
oc get csv -A
```

d. Eliminare il IBM MQ **ClusterServiceVersion** dal cluster:

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

e. Visualizzare le sottoscrizioni:

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

f. Eliminare le sottoscrizioni:

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

g. Facoltativo: se nessun altro utilizza i servizi comuni, è possibile disinstallare l'operatore dei servizi comuni. Per eseguire questa operazione, seguire le istruzioni in Disinstallazione dei servizi di base nella documentazione del prodotto IBM Cloud Pak foundational services .

Preparazione per IBM MQ creando la propria immagine contenitore

Sviluppa un contenitore auto - costruito. 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 IBM MQ Operator. Vedere “Scelta della modalità di utilizzo di IBM MQ nei contenitori” a pagina 8

Informazioni su questa attività

Procedura

Considerazioni generali quando si crea la propria immagine del gestore code

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.

sicurezza contenitore

Per ridurre al minimo i requisiti di sicurezza di runtime, le immagini del contenitore di esempio vengono installate utilizzando l'installazione dezippabile IBM MQ . Ciò garantisce che non sia impostato alcun bit `setuid` e che il contenitore non debba utilizzare l'escalation dei privilegi. Alcuni sistemi di contenitori definiscono quali ID utente sono in grado di utilizzare e l'installazione non zippabile non fa alcun presupposto sugli utenti del sistema operativo disponibili.

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

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 55
  - “Creazione di un'immagine del gestore code IBM MQ configurata di esempio” a pagina 55

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.

  Segui le istruzioni riportate in Creazione di un'immagine del contenitore su GitHub.
- Opzionale: Se si prevede di configurare l'accesso sicuro utilizzando l'SCC (Security Context Constraint) Red Hat OpenShift Container Platform "limitato" , utilizzare una delle immagini di non installazione IBM MQ .

  I link per scaricare queste immagini sono disponibili nella sezione Contenitori di IBM MQ download.

Risultati

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

Ora è possibile creare un'immagine del gestore code IBM MQ configurata di esempio.

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. A tale scopo, si crea un proprio livello di immagine contenitore, utilizzando l'immagine generica come elemento principale.

Prima di iniziare

Questa attività presuppone che, quando si crea l'immagine del gestore code IBM MQ di esempio, sia stato utilizzato il package "Nessuna installazione" IBM MQ . In caso contrario, non è possibile configurare l'accesso protetto utilizzando l'SCC (Security Context Constraint) Red Hat OpenShift Container Platform "limitato" . L'SCC "limitato" , che viene utilizzato per impostazione predefinita, utilizza ID utente casuali e impedisce l'escalation dei privilegi passando a un utente differente. Il programma di installazione IBM MQ tradizionale basato su RPM si basa su un utente e un gruppo `mqm` e utilizza anche bit `setuid` su programmi eseguibili. Nella versione corrente di IBM MQ, quando si utilizza il package "Nessuna installazione" IBM MQ , non esiste più alcun utente `mqm` né un gruppo `mqm` .

Procedura

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

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

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

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

```
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.

   Nota: Su Red Hat Enterprise Linux (RHEL), utilizzi il comando **`docker`** (RHEL V7) o **`podman`** (RHEL V7 o RHEL V8). Su Linux, sarà necessario eseguire i comandi **`docker`** con **`sudo`** all'inizio del comando, per ottenere ulteriori privilegi.

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:

```
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 visualizzare i processi IBM MQ in esecuzione nel tuo 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 visualizzare 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 è stato immesso il comando **`docker run`** .
- È possibile visualizzare i log di un contenitore utilizzando il comando **`docker logs ${CONTAINER_ID}`** .

Esecuzione di applicazioni di bind locali in contenitori separati

Con la condivisione dello spazio dei nomi del processo tra i contenitori, 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

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` .

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 **dspmq** sul secondo contenitore, per visualizzare lo stato di entrambi i gestori code, immettendo il seguente comando:

```
docker exec secondaryContainer dspmq
```

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 **dspmq**, **amqsput**, **amqsget**e **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 58 per ulteriori informazioni.

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 57 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]
lpiObtainQMDetails --> 545261715

docker exec container crtmqm QM1
AMQ8101: IBM MQ error (893) has occurred.

docker exec container strmqm QM1
AMQ6239: Permission denied attempting to access filesystem location '/var/mqm'.
AMQ7002: An error occurred manipulating a file.

docker exec container endmqm QM1
AMQ8101: IBM MQ error (893) has occurred.

docker exec container dltmqm QM1
AMQ7002: An error occurred manipulating a file.

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

MQ Adv. Creazione del gruppo HA nativo se si creano i propri contenitori

È necessario creare, configurare e avviare tre gestori code per creare il gruppo HA nativo.

Informazioni su questa attività

Il metodo consigliato per la creazione di una soluzione HA nativa è l'utilizzo dell'operatore IBM MQ (consultare HA nativa). In alternativa, se crei i tuoi contenitori, puoi seguire queste istruzioni.

Per creare un gruppo HA nativo, creare tre gestori code su tre nodi con il tipo di log impostato su `log replication`. Si modifica quindi il file `qm.ini` per ciascun gestore code per aggiungere i dettagli di connessione per ognuno dei tre nodi in modo che possano replicare i dati di log l'uno sull'altro.

È necessario, quindi, avviare tutti e tre i gestori code in modo che possano controllare che tutte e tre le istanze possano comunicare tra loro e determinare quale di essi sarà l'istanza attiva e quali saranno le replica.

Nota: Puoi creare un gruppo HA nativo nei tuoi propri contenitori solo in questo modo se stai eseguendo Kubernetes o Red Hat OpenShift.

Procedura

1. Su ognuno dei tre nodi, creare un gestore code, specificando un tipo di log di replica del log e fornendo un nome univoco per ciascuna istanza del log. Ogni gestore code ha lo stesso nome:

   ```
   crtmqm -lr instance_name qmname
   ```

   Ad esempio:

   ```
   node 1> crtmqm -lr qm1_inst1 qm1

   node 2> crtmqm -lr qm1_inst2 qm1

   node 3> crtmqm -lr qm1_inst3 qm1
   ```

2. Una volta creato correttamente ciascun gestore code, una stanza aggiuntiva denominata `NativeHALocalInstance` viene aggiunta al file di configurazione del gestore code, `qm.ini`. Un attributo `Name` viene aggiunto alla sezione specificando il nome istanza fornito.

   È possibile aggiungere facoltativamente i seguenti attributi alla sezione `NativeHALocalInstance` nel file `qm.ini` :

   `KeyRepository`
   L'ubicazione del repository delle chiavi che contiene il certificato digitale da utilizzare per la protezione del traffico di replica del log. L'ubicazione viene fornita in formato di radice, ossia include il percorso completo e il nome file senza un'estensione. Se l'attributo della sezione `KeyRepository` viene omesso, i dati di replica del log vengono scambiati tra le istanze in testo semplice.

   `CertificateLabel`
   L'etichetta del certificato che indica il certificato digitale da utilizzare per la protezione del traffico di replica del log. Se `KeyRepository` viene fornito ma `CertificateLabel` viene omesso, viene utilizzato il valore predefinito `ibmwebspheremq`*`queue_manager`* .

   `CipherSpec`
   La CipherSpec di MQ da utilizzare per proteggere il traffico di replica del log. Se questo attributo della stanza viene fornito, è necessario fornire anche `KeyRepository` . Se `KeyRepository` viene fornito ma `CipherSpec` viene omesso, viene utilizzato il valore predefinito `ANY` .

   `LocalAddress`
   L'indirizzo dell'interfaccia di rete locale che accetta il traffico di replica del log. Se questo attributo della stanza viene fornito, identifica l'interfaccia di rete locale e / o la porta utilizzando il formato "[addr] [(port)]". L'indirizzo di rete può essere specificato come un nome host, in formato decimale puntato IPv4 o in formato esadecimale IPv6 . Se questo attributo viene omesso, il gestore code tenta di collegarsi a tutte le interfacce di rete, utilizza la porta specificata in `ReplicationAddress` nella stanza `NativeHAInstances` che corrisponde al nome dell'istanza locale.

   `HeartbeatInterval`
   L'intervallo di heartbeat definisce la frequenza in millisecondi con cui un'istanza attiva di un gestore code HA nativo invia un heartbeat di rete. L'intervallo valido del valore dell'intervallo di heartbeat è compreso tra 500 (0.5 secondi) e 60000 (1 minuto), un valore esterno a questo intervallo causa l'errore di avvio del gestore code. Se questo attributo viene omesso, viene utilizzato un valore predefinito di 5000 (5 secondi). Ogni istanza deve utilizzare lo stesso intervallo di heartbeat.

HeartbeatTimeout
Il timeout heartbeat definisce il tempo di attesa di un'istanza di replica di un gestore code HA nativo prima di decidere che l'istanza attiva non risponde. L'intervallo valido del valore di timeout dell'intervallo di heartbeat è compreso tra 500 (0.5 secondi) e 120000 (2 minuti). Il valore del timeout di heartbeat deve essere maggiore o uguale all'intervallo di heartbeat.

Un valore non valido causa l'errore di avvio del gestore code. Se questo attributo viene omesso, una replica attende 2 x `HeartbeatInterval` prima di avviare il processo per selezionare una nuova istanza attiva. Ogni istanza deve utilizzare lo stesso timeout heartbeat.

RetryInterval
L'intervallo di nuovi tentativi definisce la frequenza in millisecondi con cui un gestore code HA nativo deve ritentare un link di replica non riuscito. L'intervallo valido per i tentativi è compreso tra 500 (0.5 secondi) e 120000 (2 minuti). Se questo attributo viene omesso, una replica attende 2 x `HeartbeatInterval` prima di ritentare un link di replica non riuscito.

3. Modificare il file `qm.ini` per ogni gestore code e aggiungere dettagli di connessione. Si aggiungono tre stanze `NativeHAInstance` , una per ogni istanza del gestore code nel gruppo HA nativo (inclusa l'istanza locale). Aggiungere i seguenti attributi:

   Nome
   Specificare il nome istanza utilizzato quando è stata creata l'istanza del gestore code.

   ReplicationAddress
   Specificare il nome host, IPv4 decimale puntato o IPv6 l'indirizzo in formato esadecimale dell'istanza. È possibile specificare l'indirizzo come nome host, IPv4 decimale puntato o IPv6 indirizzo in formato esadecimale. L'indirizzo di replica deve essere risolvibile e instradabile da ogni istanza nel gruppo. Il numero di porta da utilizzare per la replica del log deve essere specificato tra parentesi, ad esempio:

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

   Nota: Le stanze `NativeHAInstance` sono identiche in ogni istanza e possono essere fornite utilizzando la configurazione automatica (**`crtmqm -ii`**).

4. Avviare ciascuna delle seguenti tre istanze:

   ```
   strmqm QMgrName
   ```

   Quando le istanze vengono avviate, comunicano per controllare che tutte e tre le istanze siano in esecuzione, quindi decidere quale delle tre è l'istanza attiva, mentre le altre due istanze continuano ad eseguire come repliche.

Esempio

Il seguente esempio mostra la sezione di un file `qm.ini` che specifica i dettagli della HA nativa richiesti per una delle tre istanze:

```
NativeHALocalInstance:
  LocalName=node-1

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

Distribuzione e configurazione dei gestori code nei contenitori

È possibile eseguire una serie di attività per distribuire e configurare i gestori code IBM MQ .

Informazioni su questa attività

Per iniziare a distribuire e configurare i gestori code, consultare i seguenti argomenti.

Procedura

- “Distribuzione e configurazione dei gestori code utilizzando IBM MQ Operator” a pagina 62
- “Distribuzione e configurazione dei gestori code utilizzando Helm” a pagina 103

OpenShift Distribuzione e configurazione dei gestori code utilizzando IBM MQ Operator

Esempi di configurazione; configurazione di HA; connessione dall'esterno di un cluster OpenShift ; integrazione con il pannello di controllo CP4i ; integrazione con la traccia Instana; creazione di un'immagine con file MQSC e INI personalizzati; aggiunta di annotazioni ed etichette personalizzate.

Informazioni su questa attività

Procedura

- “Esempi per configurare un gestore code” a pagina 65.
- “Configurazione dell'alta disponibilità per i gestori code utilizzando IBM MQ Operator” a pagina 74.
- “Configurazione di un instradamento per la connessione a un gestore code dall'esterno di un cluster Red Hat OpenShift” a pagina 83.
- “Integrazione di IBM MQ con la traccia IBM Instana” a pagina 85.
- “Creazione di un'immagine con file MQSC e INI personalizzati, utilizzando la CLI Red Hat OpenShift” a pagina 92.
- “Aggiunta di annotazioni ed etichette personalizzate alle risorse del gestore code” a pagina 94.
- “Disabilitazione dei controlli webhook di runtime” a pagina 94.
- “Disabilitazione degli aggiornamenti dei valori predefiniti per la specifica del gestore code” a pagina 95.

OpenShift CP4I Distribuzione di un gestore code semplice utilizzando IBM MQ Operator

Questo esempio distribuisce un gestore code "quick start", che utilizza la memoria effimera (non persistente) e disattiva la sicurezza IBM MQ . I messaggi non vengono resi persistenti durante i riavvii del gestore code. È possibile modificare la configurazione per modificare molte impostazioni del gestore code.

Informazioni su questa attività

Questa attività offre 3 opzioni per la distribuzione di un gestore code su OpenShift:

1. Distribuire un gestore code con la console OpenShift.
2. Distribuire un gestore code con la CLI OpenShift.
3. Distribuire il gestore code con IBM Cloud Pak for Integration Platform UI.

Procedura

- **Opzione 2: distribuire un gestore code con la console OpenShift .**

  a) Distribuire un gestore code.

  a. Accedi alla console OpenShift con le tue credenziali di amministratore del cluster Red Hat OpenShift Container Platform .

b. Modifica **Project** nello spazio dei nomi in cui hai installato IBM MQ Operator. Selezionare lo spazio dei nomi dall'elenco a discesa **Progetto** .

c. Nel riquadro di navigazione, fare clic su **Operatori** > **Operatori installati**.

d. Nell'elenco del pannello Operatori installati, individuare e fare clic su **IBM MQ**.

e. Fare clic sulla scheda **Gestore code** .

f. Fare clic su **Crea QueueManager** . Viene visualizzato il pannello di creazione dell'istanza che offre due metodi per configurare la risorsa: la **vista Modulo** e **vista YAML**. La **Vista modulo** è selezionata per impostazione predefinita.

b) Configurare il gestore code.

Passo 2 Opzione 1: configurare nella vista **Modulo**.

La **Vista modulo** apre un modulo che è possibile utilizzare per visualizzare o modificare la configurazione della risorsa.

a. Accanto a **Licenza**, fare clic sulla freccia per espandere la sezione di accettazione della licenza.

b. Impostare **Accetta licenza** su **true** se si accetta l'accordo di licenza.

c. Fare clic sulla freccia per aprire l'elenco a discesa e selezionare una licenza. 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 137. È necessario accettare la licenza per distribuire un gestore code.

d. 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` .

Passo 2 Opzione 2: configurare nella vista **YAML**.

La **vista YAML** apre un editor contenente un file YAML di esempio per un `QueueManager`. Aggiornare i valori nel file seguendo la procedura riportata di seguito.

a. Modificare `metadata.namespace` nel nome del proprio progetto (spazio dei nomi).

b. Modificare il valore di `spec.license.license` nella stringa di licenza che corrisponde ai propri requisiti. Consultare “Riferimento per la licenza per mq.ibm.com/v1beta1” a pagina 137 per i dettagli della licenza.

c. Modificare `spec.license.accept` in `true` se si accetta l'accordo di licenza.

d. 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` .

c) Verificare la creazione del gestore code.

È possibile verificare di aver creato un gestore code completando la seguente procedura:

a. Assicurati di essere nello spazio dei nomi in cui hai creato il tuo IBM MQ Operator .

b. Dalla schermata **Home** , fare clic su **Operatori** > **Operatori installati**, quindi selezionare il IBM MQ Operator installato per cui è stato creato il gestore code.

c. Fare clic sulla scheda **Gestore code** . La creazione è completa quando lo stato di `QueueManager` è `Running`.

- **Opzione 2: distribuire un gestore code con la CLI OpenShift .**

a) Crea un file YAML di `QueueManager`

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

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart-cp4i
spec:
  version: 9.4.0.0-r1
  license:
    accept: false
```

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

Importante: Se si accetta l'accordo di licenza, modificare `accept: false` in `accept: true`. Consultare “Riferimento per la licenza per mq.ibm.com/v1beta1” a pagina 137 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 in IBM Cloud Pak for Integration. Per il funzionamento di Single Sign - On, è necessario installare prima altri componenti IBM Cloud Pak for Integration . Consultare “Installazione di IBM MQ Operator per l'utilizzo con CP4I” a pagina 39.

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

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: quickstart
spec:
  version: 9.4.0.0-r1
  license:
    accept: false
    license: L-EHXT-MQCRN9
  web:
    enabled: true
  queueManager:
    name: "QUICKSTART"
    storage:
      queueManager:
        type: ephemeral
```

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 137 per i dettagli sulla licenza.

b) Creare l'oggetto `QueueManager` .

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

c) Verificare la creazione del gestore code.

Verificare di aver creato un gestore code completando la seguente procedura:

a. Convalidare la distribuzione:

```
oc describe queuemanager Queue_Manager_Resource_Name
```

b. Controllare lo stato:

```
oc describe queuemanager quickstart
```

- **Opzione 3: distribuire un gestore code con IBM Cloud Pak for Integration Platform UI.**

  Segui le istruzioni in Distribuzione di un'istanza utilizzando l'IU della piattaforma.

Attività correlate

“Configurazione di un instradamento per la connessione a un gestore code dall'esterno di un cluster Red Hat OpenShift” a pagina 83
Hai bisogno di un instradamento Red Hat OpenShift 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 è disponibile solo nel protocollo TLS quando viene

utilizzato un protocollo TLS 1.2 o superiore. Red Hat OpenShift Container Platform Router utilizza SNI per instradare le richieste al gestore code IBM MQ .

“Connessione al IBM MQ Console distribuito in un cluster Red Hat OpenShift” a pagina 127
Come connettersi al IBM MQ Console di un gestore code distribuito su un cluster Red Hat OpenShift Container Platform .

“Esempi per configurare un gestore code” a pagina 65
È possibile configurare un gestore code modificando il contenuto della risorsa personalizzata QueueManager .

OpenShift CP4I Esempi per configurare un gestore code

È possibile configurare un gestore code modificando il contenuto della risorsa personalizzata QueueManager .

Informazioni su questa attività

Utilizzare i seguenti esempi per configurare un gestore code utilizzando il file YAML QueueManager .

Procedura

- “Esempio: fornitura di file MQSC e INI” a pagina 65
- “Esempio: configurazione di un gestore code con autenticazione TLS reciproca” a pagina 68

OpenShift CP4I *Esempio: fornitura di file MQSC e INI*

Questo esempio crea una Kubernetes ConfigMap ConfigMap contenente due file MQSC e un file INI. Viene quindi distribuito un gestore code che elabora questi file MQSC e INI.

Informazioni su questa attività

I file MQSC e INI possono essere forniti quando un gestore code viene distribuito. I dati MQSC e INI devono essere definiti in una o più Kubernetes ConfigMaps e Segreti. Questi devono essere creati nello spazio dei nomi (progetto) in cui verrà distribuito il gestore code.

Nota: Un segreto Kubernetes deve essere utilizzato quando il file MQSC o INI contiene dati sensibili.

Esempio

Il seguente esempio crea una Kubernetes ConfigMap che contiene due file MQSC e un file INI. Viene quindi distribuito un gestore code che elabora questi file MQSC e INI.

Esempio ConfigMap - applica il seguente YAML nel tuo cluster:

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

Esempio QueueManager - distribuisci il tuo gestore code con la seguente configurazione, utilizzando la riga di comando o la console web Red Hat OpenShift Container Platform :

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: mqsc-ini-qm
spec:
```

```
  version: 9.4.0.0-r1
  license:
    accept: false
    license: L-EHXT-MQCRN9
    use: Production
  web:
    enabled: true
  queueManager:
    name: "MQSCINI"
    mqsc:
    - configMap:
        name: mqsc-ini-example
        items:
        - example1.mqsc
        - example2.mqsc
    ini:
    - configMap:
        name: mqsc-ini-example
        items:
        - example.ini
    storage:
      queueManager:
        type: ephemeral
```

Importante: Se si accetta l'accordo di licenza di IBM MQ Advanced , modificare `accept: false` in `accept: true`. Per dettagli sulla licenza, consultare Riferimento alle licenze per mq.ibm.com/v1beta1 .

Ulteriori informazioni:

- Un gestore code può essere configurato in modo da utilizzare una singola Kubernetes ConfigMap o un segreto (come mostrato in questo esempio) o più ConfigMaps e segreti.
- È possibile scegliere di utilizzare tutti i dati MQSC e INI da una Kubernetes ConfigMap o da un segreto (come mostrato in questo esempio) oppure configurare ciascun gestore code per utilizzare solo un sottoinsieme dei file disponibili.
- I file MQSC e INI vengono elaborati in ordine alfabetico in base alla loro chiave. Quindi `example1.mqsc` verrà sempre elaborato prima di `example2.mqsc`, indipendentemente dall'ordine in cui vengono visualizzati nella configurazione del gestore code.
- Se più file MQSC o INI hanno la stessa chiave, su più Kubernetes ConfigMaps o Secrets, questa serie di file viene elaborata in base all'ordine in cui i file sono definiti nella configurazione del gestore code.
- Quando un pod del gestore code è in esecuzione, le modifiche a Kubernetes ConfigMap non vengono prese in considerazione perché IBM MQ Operator non è a conoscenza della modifica. Se si apportano modifiche a ConfigMap, ad esempio modifiche ai comandi MQSC o ai file INI, è necessario riavviare manualmente i gestori code per rendere effettive tali modifiche. Per i gestori code a istanza singola, eliminare il pod per attivare il riavvio richiesto. Per le distribuzioni della HA nativa, riavvia prima i pod standby eliminandoli. Quando sono di nuovo in uno stato di esecuzione, elimina il pod attivo per riavviarlo. Questo ordine di riavvii garantisce un tempo di inattività minimo per il gestore code.

OpenShift CP4I *Creazione di una PKI autofirmata utilizzando OpenSSL*

IBM MQ ti permette di utilizzare il TLS reciproco per l'autenticazione, dove entrambe le estremità di una connessione forniscono un certificato e i dettagli nel certificato vengono utilizzati per stabilire un'identità con il gestore code. Questo argomento illustra come creare un PKI (Public Key Infrastructure) di esempio utilizzando lo strumento della riga comandi OpenSSL , creando due certificati che possono essere utilizzati in altri esempi.

Prima di iniziare

Verificare che lo strumento della riga comandi OpenSSL sia installato.

Installa IBM MQ cliente aggiungi `samp/bin` e `bin` al *PATH*. È necessario il comando **runmqicred**, che può essere installato come parte di IBM MQ client come segue:

- Linux Windows Per Windows e Linux: installare il client ridistribuibile IBM MQ per il proprio sistema operativo da https://ibm.biz/mq94redistclients

- macOS Per Mac: scaricare e configurare IBM MQ MacOS Toolkit: https://developer.ibm.com/tutorials/mq-macos-dev/

Informazioni su questa attività

Importante: Gli esempi descritti di seguito non sono adatti per un ambiente di produzione e sono intesi solo come esempi per andare avanti rapidamente. La gestione dei certificati è un argomento complesso per gli utenti avanzati. Per la produzione, devi considerare cose come la rotazione, la revoca, la lunghezza della chiave, il ripristino di emergenza e molto altro ancora.

Questi passi sono stati verificati utilizzando OpenSSL 3.1.4.

Procedura

1. Creare una chiave privata da utilizzare per la propria autorità di certificazione interna

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

   Una chiave privata per l'autorità di certificazione interna viene creata in un file denominato *ca.key*. Questo file deve essere mantenuto sicuro e segreto - verrà utilizzato per firmare i certificati per la propria autorità di certificazione interna.
2. Emettere un certificato autofirmato per l'autorità di certificazione interna

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

   `-days` specifica il numero di giorni di validità del certificato CA root.

   Un certificato viene creato in un file denominato *ca.crt*. Questo certificato contiene le informazioni pubbliche sull'autorità di certificazione interna ed è liberamente condivisibile.
3. Crea una chiave privata e un certificato per un gestore code
   a) Crea una chiave privata e una richiesta di firma certificato per un gestore code

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

      Una chiave privata viene creata in un file denominato *example-qm.key*e una richiesta di firma del certificato viene creata in un file denominato *example-qm.csr*
   b) Firma la chiave gestore code con la tua autorità di certificazione interna

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

      `-days` specifica il numero di giorni di validità del certificato.

      Un certificato firmato viene creato in un file denominato *example-qm.crt*
   c) Crea un segreto Kubernetes con il certificato e la chiave del gestore code

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

      Viene creato un Kubernetes segreto denominato *example - qm - tls* . Questo segreto contiene la chiave privata per il gestore code, il certificato pubblico e il certificato CA.
4. Crea una chiave privata e un certificato per una applicazione
   a) Crea una chiave privata e una richiesta di firma certificato per un'applicazione

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

      Una chiave privata viene creata in un file denominato *example-app1.key*e una richiesta di firma del certificato viene creata in un file denominato *example-app1.csr*

b) Firma la chiave gestore code con la tua autorità di certificazione interna

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

`-days` specifica il numero di giorni di validità del certificato.

Un certificato firmato viene creato in un file denominato *example-app1.crt*

c) Crea una memoria chiavi PKCS#12 con la chiave e il certificato dell'applicazione

IBM MQ utilizza un database di chiavi e non singoli file di chiavi. Il gestore code inserito nel contenitore creerà il database delle chiavi per il gestore code da un segreto, ma per le applicazioni client è necessario creare manualmente il database delle chiavi.

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

Dove *PASSWORD* è una password di propria scelta.

Viene creato un keystore in un file denominato *example-app1.p12*. La chiave e il certificato dell'applicazione sono memorizzati all'interno, con una "etichetta" o un "nome descrittivo" di "example-app1", così come il certificato CA.

d) Se stai utilizzando un arm64 Apple Mac, devi configurare un file aggiuntivo che combina l'applicazione e i certificati CA.
Ad esempio:

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

Attività correlate

“Esempio: configurazione di un gestore code con autenticazione TLS reciproca” a pagina 68
Questo esempio distribuisce un gestore code in OpenShift Container Platform utilizzando IBM MQ Operator. Il TLS reciproco viene utilizzato per l'autenticazione, per eseguire l'associazione da un certificato TLS a un'identità nel gestore code.

“Esempio: configurazione della HA nativa utilizzando IBM MQ Operator” a pagina 74
Questo esempio distribuisce un gestore code utilizzando la funzione di alta disponibilità nativa in OpenShift Container Platform utilizzando IBM MQ Operator. Il TLS reciproco viene utilizzato per l'autenticazione, per eseguire l'associazione da un certificato TLS a un'identità nel gestore code.

“Configurazione di un gestore code a più istanze utilizzando IBM MQ Operator” a pagina 80
Questo esempio distribuisce un gestore code a più istanze utilizzando OpenShift Container Platform utilizzando IBM MQ Operator. Il TLS reciproco viene utilizzato per l'autenticazione, per eseguire l'associazione da un certificato TLS a un'identità nel gestore code.

OpenShift CP4I Linux *Esempio: configurazione di un gestore code con autenticazione TLS reciproca*

Questo esempio distribuisce un gestore code in OpenShift Container Platform utilizzando IBM MQ Operator. Il TLS reciproco viene utilizzato per l'autenticazione, per eseguire l'associazione da un certificato TLS a un'identità nel gestore code.

Prima di iniziare

Per completare questo esempio, è necessario prima aver completato i prerequisiti riportati di seguito:

- Creare un progetto / spazio dei nomi OpenShift Container Platform (OCP) per questo esempio.
- Sulla riga comandi, accedere al cluster OCP e passare allo spazio dei nomi precedente.
- Assicurarsi che il file IBM MQ Operator sia installato e disponibile nello spazio dei nomi precedente.

Informazioni su questa attività

Questo esempio fornisce una risorsa personalizzata YAML che definisce un gestore code da distribuire in OpenShift Container Platform. Descrive inoltre i passi aggiuntivi richiesti per distribuire il gestore code con TLS abilitato.

Procedura

1. Creare una coppia di certificati come descritto in “Creazione di una PKI autofirmata utilizzando OpenSSL” a pagina 66.
2. Crea una mappa di configurazione contenente comandi MQSC e un file INI

   Creare una Kubernetes ConfigMap contenente i comandi MQSC per creare una nuova coda e un canale SVRCONN e per aggiungere un record di autenticazione di canale che consenta l'accesso al canale.

   Assicurati di essere nello spazio dei nomi che hai creato precedentemente (vedi Prima di iniziare), quindi immetti il seguente YAML nella console web OCP o utilizzando la riga di comando.

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

   MQSC definisce un canale denominato *MTLS.SVRCONN* e una coda denominata *EXAMPLE.QUEUE*. Il canale è configurato per consentire l'accesso solo ai client che presentano un certificato con un "nome comune" *example-app1*. Questo è il nome comune utilizzato in uno dei certificati creati nel passo “1” a pagina 69. Le connessioni su questo canale con questo nome comune vengono associate a un ID utente *app1*, autorizzato a connettersi al gestore code e ad accedere alla coda di esempio. Il file INI abilita una politica di sicurezza che indica che l'ID utente *app1* non deve necessariamente esistere in un registro utente esterno - esiste solo come nome in questa configurazione.
3. Distribuisci il gestore code

   Creare un nuovo gestore code utilizzando la seguente risorsa personalizzata YAML. Assicurarsi di essere nello spazio dei nomi creato prima di iniziare questa attività, quindi immettere il seguente YAML nella console Web OCP o utilizzando la riga comandi. Verificare che sia stata specificata la licenza corretta e accettarla modificando `false` in `true`.

   ```
   apiVersion: mq.ibm.com/v1beta1
   kind: QueueManager
   metadata:
     name: exampleqm
   spec:
     license:
       accept: false
       license: L-EHXT-MQCRN9
       use: Production
     queueManager:
       name: EXAMPLEQM
       mqsc:
       - configMap:
           name: example-tls-configmap
           items:
           - example-tls.mqsc
       ini:
   ```

```
    - configMap:
        name: example-tls-configmap
        items:
        - example-tls.ini
    storage:
      queueManager:
        type: ephemeral
  version: 9.4.0.0-r1
  pki:
    keys:
      - name: default
        secret:
          secretName: example-qm-tls
          items:
            - tls.key
            - tls.crt
            - ca.crt
```

Tenere presente che il segreto *example - qm - tls* è stato creato nel passo “1” a pagina 69e che ConfigMap *example - tls - configmap* è stato creato nel passo “2” a pagina 69

4. Confermare che il gestore code è in esecuzione

   Il gestore code è in fase di distribuzione. Confermare che si trova nello stato `Running` prima di procedere. Ad esempio:

   ```
   oc get qmgr exampleqm
   ```

5. Verifica la connessione al gestore code

   Per confermare che il gestore code è configurato per la comunicazione TLS reciproca, attenersi alla procedura descritta in “Verifica di una connessione TLS reciproca a un gestore code dal tuo laptop” a pagina 70.

Risultati

Congratulazioni, hai correttamente distribuito un gestore code con TLS abilitato e che utilizza i dettagli forniti nel certificato TLS per autenticarsi con il gestore code e fornire un'identità.

OpenShift CP4I Linux *Verifica di una connessione TLS reciproca a un gestore code dal tuo laptop*

Dopo aver creato un gestore code utilizzando IBM MQ Operator, è possibile verificare che funzioni collegandosi ad esso, inserendo e ricevendo un messaggio. Questa attività illustra come connettersi utilizzando programmi di esempio IBM MQ , eseguendoli su una macchina esterna al cluster Kubernetes , come ad esempio il laptop.

Prima di iniziare

Per completare questo esempio, è necessario aver prima completato i seguenti prerequisiti:

- Installare IBM MQ client. Sono necessari i comandi **amqsputc** e **amqsgetc** , che possono essere installati come parte di IBM MQ client come segue:
  - Linux Windows Per Windows e Linux: installare il client ridistribuibile IBM MQ per il proprio sistema operativo da https://ibm.biz/mq94redistclients
  - macOS Per Mac: scaricare e configurare IBM MQ MacOS Toolkit: https://developer.ibm.com/tutorials/mq-macos-dev/
- Accertarsi di avere i file di chiave e certificato necessari scaricati in una directory sulla macchina e di conoscere la parola d'ordine del keystore. Ad esempio, questi file vengono creati in “Creazione di una PKI autofirmata utilizzando OpenSSL” a pagina 66:
  - `example-app1.p12`
  - `example-app1-chain.crt` (solo se stai utilizzando un Apple Mac arm64)

- Distribuire un gestore code configurato con TLS al cluster OCP, ad esempio seguendo la procedura in "Esempio: configurazione di un gestore code con autenticazione TLS reciproca" a pagina 68

Informazioni su questa attività

Questo esempio utilizza i programmi di esempio IBM MQ in esecuzione su una macchina esterna al cluster Kubernetes , ad esempio il laptop, per connettersi a QueueManager configurato con TLS e per inserire e richiamare messaggi.

Procedura

1. Confermare che il gestore code è in esecuzione

   Il gestore code è in fase di distribuzione. Confermare che si trova nello stato `Running` prima di procedere. Ad esempio:

   ```
   oc get qmgr exampleqm
   ```

2. Trova il nome host del gestore code

   Utilizzare il seguente comando per trovare il nome host completo del gestore code dall'esterno del cluster OCP, utilizzando l'instradamento creato automaticamente: `exampleqm-ibm-mq-qm`:

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

3. Creare una CCDT (IBM MQ Client Channel Definition Table)

   Creare un file denominato `ccdt.json` con il seguente contenuto:

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

   La connessione utilizza la porta 443, perché è la porta su cui è in ascolto il router Red Hat OpenShift Container Platform . Il traffico verrà inoltrato al gestore code sulla porta 1414.

   Se hai utilizzato un nome di canale diverso, dovrai anche modificarlo. Gli esempi TLS reciproci utilizzano un canale denominato *MTLS.SVRCONN*

   Per ulteriori dettagli, consultare Configurazione di una CCDT in formato JSON

4. Creare un file INI client per configurare i dettagli di connessione

   Creare un file denominato `mqclient.ini` nella directory corrente. Questo file verrà letto da **amqsputc** e **amqsgetc**.

   ```
   Channels:
     ChannelDefinitionDirectory=.
     ChannelDefinitionFile=ccdt.json
   SSL:
   ```

```
OutboundSNI=HOSTNAME
SSLKeyRepository=example-app1.p12
SSLKeyRepositoryPassword=password you used when creating the p12 file
```

Assicurarsi di aggiornare la *SSLKeyRepositoryPassword* con la password scelta durante la creazione del file PKCS#12 . Esistono altri modi per impostare la password del keystore, incluso l'utilizzo di una password codificata. Per ulteriori informazioni, consultare Come fornire la password del repository delle chiavi per un IBM MQ MQI client su AIX, Linux, and Windows

Tenere presente che Red Hat OpenShift Container Platform Router utilizza SNI per instradare le richieste al gestore code IBM MQ . L'attributo *OutboundSNI= HOSTNAME* garantisce che il client IBM MQ includa le informazioni necessarie per il funzionamento del router con l'instradamento predefinito configurato da IBM MQ Operator. Per ulteriori informazioni, fare riferimento a “Configurazione di un instradamento per la connessione a un gestore code dall'esterno di un cluster Red Hat OpenShift” a pagina 83.

5. Se si sta utilizzando un arm64 Apple Mac, è necessario configurare una variabile di ambiente aggiuntiva.

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

   Questo file contiene la catena di certificati completa, inclusi i certificati CA e dell'applicazione.
6. Inserire i messaggi nella coda

   Esegui il seguente comando:

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

   Se la connessione al gestore code ha esito positivo, viene emessa la seguente risposta:

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

   Inserire diversi messaggi nella coda, immettendo del testo e premendo ogni volta **Invio** .

   Per terminare, premere due volte **Invio** .
7. Richiama i messaggi dalla coda

   Esegui il seguente comando:

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

   I messaggi aggiunti nel passo precedente sono stati utilizzati e vengono emessi. Dopo pochi secondi, il comando esce.

Risultati

Congratulazioni, hai verificato correttamente la connessione a un gestore code con TLS abilitato e hai mostrato che puoi inserire e ricevere messaggi in modo sicuro al gestore code da un client.

OpenShift CP4I *Esempio: personalizzazione delle annotazioni del servizio di licenza*

IBM MQ Operator aggiunge automaticamente le annotazioni IBM License Service alle risorse distribuite. Questi sono monitorati da IBM License Servicee vengono generati report che corrispondono alla titolarità richiesta.

Informazioni su questa attività

Le annotazioni aggiunte da IBM MQ Operator sono quelle previste in situazioni standard e si basano sui valori della licenza selezionati durante la distribuzione di un gestore code.

Esempio

Se **License** è impostata su `L-RJON-BZFQU2 (IBM Cloud Pak for Integration 2021.2.1)`e **Use** è impostato su `NonProduction`, vengono applicate le seguenti annotazioni:

- cloudpakId: c8b82d189e7545f0892db9ef2731b90d
- cloudpakName: IBM Cloud Pak for Integration
- Contenitori productCharged: qmgr
- productCloudpakRapporto: '4:1'
- productID: 21dfe9a0f00f444f888756d835334909
- productName: IBM MQ Advanced per Non - Production
- productMetric: VIRTUAL_PROCESSOR_CORE
- productVersion: 9.2.3.0

All'interno di IBM Cloud Pak for Integration, le distribuzione di IBM App Connect Enterprise includono una titolarità limitata per IBM MQ. In queste situazioni, queste annotazioni devono essere sovrascritte per garantire che IBM License Service acquisisca l'uso corretto. A tale scopo, utilizzare l'approccio descritto in “Aggiunta di annotazioni ed etichette personalizzate alle risorse del gestore code” a pagina 94.

Ad esempio, se IBM MQ viene distribuito in base alla titolarità IBM App Connect Enterprise , utilizzare l'approccio mostrato nel seguente frammento di codice:

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

Ci sono altri due motivi comuni per cui le annotazioni di licenza potrebbero richiedere modifiche:

1. IBM MQ Advanced è incluso nella titolarità di un altro prodotto IBM .
   - In questa situazione, utilizzare l'approccio precedentemente descritto per IBM App Connect Enterprise.
2. IBM MQ viene distribuito con una licenza IBM Cloud Pak for Integration .
   - Se si dispone di una licenza IBM Cloud Pak for Integration , è possibile decidere di distribuire un gestore code nel rapporto IBM MQ o IBM MQ Advanced . Se esegui la distribuzione in un rapporto IBM MQ , devi assicurarti di non utilizzare alcuna funzionalità avanzata come la HA nativa o Advanced Message Security.
   - In questa situazione, utilizzare le seguenti annotazioni per uso di produzione:

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

   - Utilizzare le seguenti annotazioni per uso non di produzione:

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

```
    productID: 151bec68564a4a47a14e6fa99266deff
    productCloudpakRatio: '8:1'
    productName: IBM MQ for Non-Production
    productMetric: VIRTUAL_PROCESSOR_CORE
```

OpenShift MQ Adv. Configurazione dell'alta disponibilità per i gestori code utilizzando IBM MQ Operator

Informazioni su questa attività

Procedura

- “HA nativa” a pagina 19.
- “Esempio: configurazione della HA nativa utilizzando IBM MQ Operator” a pagina 74.
- “Configurazione di un gestore code a più istanze utilizzando IBM MQ Operator” a pagina 80.

OpenShift MQ Adv. *Configurazione della HA nativa utilizzando IBM MQ Operator*

La HA nativa è configurata utilizzando l'API `QueueManager` e le opzioni avanzate sono disponibili utilizzando un file INI.

La HA nativa è configurata utilizzando il `.spec.queueManager.availability` dell'API di `QueueManager` , ad esempio:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
name: nativeha-example
spec:
  license:
    accept: false
    license: L-EHXT-MQCRN9
    use: Production
  queueManager:
    availability:
      type: NativeHA
  version: 9.4.0.0-r1
```

il campo `.spec.queueManager.availability.type` deve essere impostato su `NativeHA`.

In `.spec.queueManager.availability`, è anche possibile configurare un segreto TLS e le cifrature da utilizzare tra le istanze del gestore code durante la replica. Ciò è fortemente consigliato e una guida dettagliata è disponibile in “Esempio: configurazione della HA nativa utilizzando IBM MQ Operator” a pagina 74.

Attività correlate

“Esempio: configurazione della HA nativa utilizzando IBM MQ Operator” a pagina 74
Questo esempio distribuisce un gestore code utilizzando la funzione di alta disponibilità nativa in OpenShift Container Platform utilizzando IBM MQ Operator. Il TLS reciproco viene utilizzato per l'autenticazione, per eseguire l'associazione da un certificato TLS a un'identità nel gestore code.

OpenShift MQ Adv. Kubernetes *Esempio: configurazione della HA nativa utilizzando IBM MQ Operator*

Questo esempio distribuisce un gestore code utilizzando la funzione di alta disponibilità nativa in OpenShift Container Platform utilizzando IBM MQ Operator. Il TLS reciproco viene utilizzato per l'autenticazione, per eseguire l'associazione da un certificato TLS a un'identità nel gestore code.

Prima di iniziare

Per completare questo esempio, è necessario prima aver completato i prerequisiti riportati di seguito:

- Creare un progetto / spazio dei nomi OpenShift Container Platform (OCP) per questo esempio.
- Sulla riga comandi, accedere al cluster OCP e passare allo spazio dei nomi precedente.
- Assicurarsi che IBM MQ Operator sia installato e sia disponibile nel suddetto spazio dei nomi.

Informazioni su questa attività

Questo esempio fornisce una risorsa personalizzata YAML che definisce un gestore code da distribuire in OpenShift Container Platform. Descrive inoltre i passi aggiuntivi richiesti per distribuire il gestore code con TLS abilitato.

Procedura

1. Creare una coppia di certificati come descritto in “Creazione di una PKI autofirmata utilizzando OpenSSL” a pagina 66.
2. Crea una mappa di configurazione contenente comandi MQSC e un file INI

   Creare una Kubernetes ConfigMap contenente i comandi MQSC per creare una nuova coda e un canale SVRCONN e per aggiungere un record di autenticazione di canale che consenta l'accesso al canale.

   Assicurati di essere nello spazio dei nomi che hai creato precedentemente (vedi Prima di iniziare), quindi immetti il seguente YAML nella console web OCP o utilizzando la riga di comando.

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

   MQSC definisce un canale denominato *MTLS.SVRCONN* e una coda denominata *EXAMPLE.QUEUE*. Il canale è configurato per consentire l'accesso solo ai client che presentano un certificato con un "nome comune" *example-app1*. Questo è il nome comune utilizzato in uno dei certificati creati nel passo “1” a pagina 75. Le connessioni su questo canale con questo nome comune vengono associate a un ID utente *app1*, autorizzato a connettersi al gestore code e ad accedere alla coda di esempio. Il file INI abilita una politica di sicurezza che indica che l'ID utente *app1* non deve necessariamente esistere in un registro utente esterno - esiste solo come nome in questa configurazione.
3. Distribuisci il gestore code

   Creare un nuovo gestore code utilizzando la seguente risorsa personalizzata YAML. Assicurarsi di essere nello spazio dei nomi creato prima di iniziare questa attività, quindi immettere il seguente YAML nella console Web OCP o utilizzando la riga comandi. Verificare che sia stata specificata la licenza corretta e accettarla modificando `false` in `true`.

   ```
   apiVersion: mq.ibm.com/v1beta1
   kind: QueueManager
   metadata:
     name: exampleqm
   spec:
     license:
       accept: false
       license: L-EHXT-MQCRN9
       use: Production
     queueManager:
   ```

```
    name: EXAMPLEQM
    availability:
      type: NativeHA
      tls:
        secretName: example-qm-tls
    mqsc:
    - configMap:
        name: example-nativeha-configmap
        items:
        - example-tls.mqsc
    ini:
    - configMap:
        name: example-nativeha-configmap
        items:
        - example-tls.ini
    storage:
      queueManager:
        type: persistent-claim
  version: 9.4.0.0-r1
  pki:
    keys:
      - name: default
        secret:
          secretName: example-qm-tls
          items:
            - tls.key
            - tls.crt
            - ca.crt
```

Si noti che il segreto *example - qm - tls* è stato creato nel passo “1” a pagina 75e la ConfigMap *example - nativeha - configmap* è stata creata nel passo “2” a pagina 75

Il tipo di disponibilità è impostato su *NativeHA*e l'archiviazione persistente è selezionata. Verrà utilizzata la classe di archiviazione predefinita configurata nel cluster Kubernetes . Se non si dispone di una classe di archiviazione configurata come predefinita o si desidera utilizzare una classe di archiviazione differente, aggiungere `defaultClass: storage_class_name` in `spec.queueManager.storage`.

I tre pod in un gestore code HA nativo replicano i dati sulla rete. Questo link non è codificato per impostazione predefinita, ma questo esempio utilizza il certificato del gestore code per codificare il traffico. È possibile specificare un certificato differente per ulteriore sicurezza. Il segreto TLS HA nativo deve essere un segreto TLS Kubernetes , che ha una struttura particolare (ad esempio, la chiave privata deve essere denominata *tls.key*).

4. Confermare che il gestore code è in esecuzione

   Il gestore code è in fase di distribuzione. Confermare che si trova nello stato `Running` prima di procedere. Ad esempio:

   ```
   oc get qmgr exampleqm
   ```

5. Verifica la connessione al gestore code

   Per confermare che il gestore code è configurato e disponibile, effettuare le operazioni riportate in “Verifica di una connessione TLS reciproca a un gestore code dal tuo laptop” a pagina 70.

6. Forza l'esito negativo del pod attivo

   Per convalidare il ripristino automatico del gestore code, simula un errore pod:

   a) Visualizza i pod attivi e in standby

      Esegui il seguente comando:

      ```
      oc get pods --selector app.kubernetes.io/instance=exampleqm
      ```

      Nota che, nel campo **READY** , il pod attivo restituisce il valore `1/1`, mentre i pod di replica restituiscono il valore `0/1`.

   b) Elimina il pod attivo

      Esegui il seguente comando, specificando il nome completo del pod attivo:

      ```
      oc delete pod exampleqm-ibm-mq-value
      ```

c) Visualizza di nuovo lo stato del pod

Esegui il seguente comando:

```
oc get pods --selector app.kubernetes.io/instance=exampleqm
```

d) Visualizza lo stato del gestore code

Esegui il seguente comando, specificando il nome completo di uno degli altri pod:

```
oc exec -t Pod -- dspmq -o nativeha -x -m EXAMPLEQM
```

Dovresti vedere lo stato che mostra che l'istanza attiva è cambiata, ad esempio:

```
QMNAME(EXAMPLEQM) ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
INSTANCE(inst1) ROLE(Active) REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
```

e) Verificare nuovamente la connessione al gestore code

Per confermare che il gestore code è stato ripristinato, effettuare le operazioni riportate in “Verifica di una connessione TLS reciproca a un gestore code dal tuo laptop” a pagina 70.

Risultati

Congratulazioni, hai distribuito correttamente un gestore code con alta disponibilità nativa e autenticazione TLS reciproca e verificato che venga ripristinato automaticamente quando il pod attivo ha esito negativo.

Visualizzazione dello stato dei gestori code della HA nativa per i contenitori IBM MQ

Per i contenitori IBM MQ , puoi visualizzare lo stato delle istanze Native HA eseguendo il comando **dspmq** all'interno di uno dei pod in esecuzione.

Informazioni su questa attività

È possibile utilizzare il comando **dspmq** in uno dei pod in esecuzione per visualizzare lo stato operativo di un'istanza del gestore code. Le informazioni restituite dipendono dal fatto che l'istanza sia attiva o una replica. Le informazioni fornite dall'istanza attiva sono definitive, le informazioni dai nodi di replica potrebbero non essere aggiornate.

È possibile effettuare le seguenti azioni:

- Visualizzare se l'istanza del gestore code sul nodo corrente è attiva o una replica.
- Visualizza lo stato operativo della HA nativa dell'istanza sul nodo corrente.
- Visualizzare lo stato operativo di tutte e tre le istanze in una configurazione HA nativa.

I seguenti campi di stato vengono utilizzati per riportare lo stato di configurazione della HA nativa:

RUOLO
Specifica il ruolo corrente dell'istanza ed è uno tra `Active`, `Replicao` Unknown.

ISTANZA
Il nome fornito per questa istanza del gestore code quando è stata creata utilizzando l'opzione **-lr** del comando **crtmqm** .

INSYNC
Indica se l'istanza è in grado di assumere il controllo come istanza attiva, se richiesto.

quorum
Riporta lo stato del quorum nel formato *number_of_instances_in - sync/ number_of_instances_configured*.

REPLADDR
L'indirizzo di replica dell'istanza del gestore code.

COLLEGA
Indica se il nodo è connesso all'istanza attiva.

BACKLOG
Indica il numero di KB in cui si trova l'istanza.

CONNETTIN
Indica se l'istanza denominata è connessa a questa istanza.

ALTDATE
Indica la data in cui queste informazioni sono state aggiornate l'ultima volta (vuoto se non sono mai state aggiornate).

ALTTIME
Indica l'ora in cui queste informazioni sono state aggiornate l'ultima volta (vuoto se non sono mai state aggiornate).

Procedura

- Trova i pod che fanno parte del tuo gestore code.

```
oc get pod --selector app.kubernetes.io/instance=nativeha-qm
```

- Esegui il dspmq in uno dei pod

```
oc exec -t Pod dspmq
```

```
oc rsh Pod
```

per una shell interattiva, dove è possibile eseguire direttamente dspmq .

- Per determinare se un'istanza del gestore code è in esecuzione come istanza attiva o come replica:

```
oc exec -t Pod dspmq -o status -m QMgrName
```

Un'istanza attiva di un gestore code denominato BOB riporta il seguente stato:

```
QMNAME(BOB)                STATUS(Running)
```

Un'istanza di replica di un gestore code denominato BOB riporta il seguente stato:

```
QMNAME(BOB)                STATUS(Replica)
```

Un'istanza inattiva riporta il seguente stato:

```
QMNAME(BOB)                STATUS(Ended Immediately)
```

- Per determinare lo stato operativo della HA nativa dell'istanza nel pod specificato:

```
oc exec -t Pod dspmq -o nativeha -m QMgrName
```

L'istanza attiva di un gestore code denominato BOB potrebbe riportare il seguente stato:

```
QMNAME(BOB)              ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
```

Un'istanza di replica di un gestore code denominato BOB potrebbe riportare il seguente stato:

```
QMNAME(BOB)              ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
```

Un'istanza inattiva di un gestore code denominato BOB potrebbe riportare il seguente stato:

```
QMNAME(BOB)              ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
```

- Per determinare lo stato operativo della HA nativo di tutte le istanze nella configurazione della HA nativa:

```
oc exec -t Pod dspmq -o nativeha -x -m QMgrName
```

Se si immette questo comando sul nodo che esegue l'istanza attiva del BOB del gestore code, è possibile che si riceva il seguente stato:

```
QMNAME(BOB)            ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
 INSTANCE(inst1) ROLE(Active)  REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
 INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
 INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
```

Se si immette questo comando su un nodo che esegue un'istanza di replica del BOB del gestore code, è possibile che si riceva il seguente stato, che indica che una delle repliche è in ritardo:

```
QMNAME(BOB)            ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
 INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
 INSTANCE(inst1) ROLE(Active)  REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
 INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(No)  BACKLOG(435)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
```

Se si immette questo comando su un nodo che esegue un'istanza inattiva del BOB del gestore code, è possibile che si riceva il seguente stato:

```
QMNAME(BOB)            ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
 INSTANCE(inst1) ROLE(Unknown) REPLADDR(9.20.123.45) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATE() ALTTIME()
 INSTANCE(inst2) ROLE(Unknown) REPLADDR(9.20.123.46) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATE() ALTTIME()
 INSTANCE(inst3) ROLE(Unknown) REPLADDR(9.20.123.47) CONNACTV(No) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATE() ALTTIME()
```

Se si immette il comando quando le istanze stanno ancora negoziando quali sono attive e quali sono repliche, si riceverà il seguente stato:

```
QMNAME(BOB)              STATUS(Negotiating)
```

Attività correlate

“Esempio: configurazione della HA nativa utilizzando IBM MQ Operator” a pagina 74
Questo esempio distribuisce un gestore code utilizzando la funzione di alta disponibilità nativa in OpenShift Container Platform utilizzando IBM MQ Operator. Il TLS reciproco viene utilizzato per l'autenticazione, per eseguire l'associazione da un certificato TLS a un'identità nel gestore code.

Riferimenti correlati

comando dspmq (visualizza gestori code)

OpenShift MQ Adv. *Ottimizzazione avanzata per Native HA*

Impostazioni avanzate per l'ottimizzazione di intervalli e intervalli. Non dovrebbe essere necessario utilizzare queste impostazioni a meno che i valori predefiniti non corrispondano ai requisiti del sistema.

Le opzioni di base per la configurazione della HA nativa vengono gestite utilizzando l'API `QueueManager`, che IBM MQ Operator utilizza per configurare i file INI del gestore code sottostante. Ci sono alcune opzioni più avanzate che sono configurabili solo utilizzando un file INI, nella stanza NativeHALocalInstance. Consultare anche “Esempio: fornitura di file MQSC e INI” a pagina 65 per ulteriori informazioni su come configurare un file INI.

`HeartbeatInterval`

L'intervallo di heartbeat definisce la frequenza in millisecondi con cui un'istanza attiva di un gestore code HA nativo invia un heartbeat di rete. L'intervallo valido del valore dell'intervallo di heartbeat è compreso tra 500 (0.5 secondi) e 60000 (1 minuto), un valore esterno a questo intervallo causa

l'errore di avvio del gestore code. Se questo attributo viene omesso, viene utilizzato un valore predefinito di 5000 (5 secondi). Ogni istanza deve utilizzare lo stesso intervallo di heartbeat.

HeartbeatTimeout
Il timeout heartbeat definisce il tempo di attesa di un'istanza di replica di un gestore code HA nativo prima di decidere che l'istanza attiva non risponde. L'intervallo valido del valore di timeout dell'intervallo di heartbeat è compreso tra 500 (0.5 secondi) e 120000 (2 minuti). Il valore del timeout di heartbeat deve essere maggiore o uguale all'intervallo di heartbeat.

Un valore non valido causa l'errore di avvio del gestore code. Se questo attributo viene omesso, una replica attende 2 x `HeartbeatInterval` prima di avviare il processo per selezionare una nuova istanza attiva. Ogni istanza deve utilizzare lo stesso timeout heartbeat.

RetryInterval
L'intervallo di nuovi tentativi definisce la frequenza in millisecondi con cui un gestore code HA nativo deve ritentare un link di replica non riuscito. L'intervallo valido per i tentativi è compreso tra 500 (0.5 secondi) e 120000 (2 minuti). Se questo attributo viene omesso, una replica attende 2 x `HeartbeatInterval` prima di ritentare un link di replica non riuscito.

Chiusura gestori code della HA nativa

È possibile utilizzare il comando **endmqm** per terminare un gestore code attivo o di replica che fa parte di un gruppo HA nativo.

Procedura

- Per terminare l''istanza attiva di un gestore code, fare riferimento a Fine dei gestori code della HA nativa nella sezione Configurazione di questa documentazione.

Configurazione di un gestore code a più istanze utilizzando IBM MQ Operator

Questo esempio distribuisce un gestore code a più istanze utilizzando OpenShift Container Platform utilizzando IBM MQ Operator. Il TLS reciproco viene utilizzato per l'autenticazione, per eseguire l'associazione da un certificato TLS a un'identità nel gestore code.

Prima di iniziare

Per completare questo esempio, è necessario prima aver completato i prerequisiti riportati di seguito:

- Creare un progetto / spazio dei nomi OpenShift Container Platform (OCP) per questo esempio.
- Sulla riga comandi, accedere al cluster OCP e passare allo spazio dei nomi precedente.
- Assicurarsi che il file IBM MQ Operator sia installato e disponibile nello spazio dei nomi precedente.

Informazioni su questa attività

Questo esempio fornisce una risorsa personalizzata YAML che definisce un gestore code da distribuire in OpenShift Container Platform. Descrive inoltre i passi aggiuntivi richiesti per distribuire il gestore code con TLS abilitato.

Procedura

1. Determinare una classe di memoria adatta

   È possibile accedere all'archiviazione in un cluster Kubernetes utilizzando più modalità di accesso al volume persistente. Un gestore code a più istanze crea più volumi persistenti: uno per ciascun gestore code e almeno un volume condiviso. Il volume condiviso per un gestore code a più istanze deve utilizzare una classe di memorizzazione `ReadWriteMany` . La classe di archiviazione predefinita in un cluster Kubernetes è in genere per una classe di archiviazione `ReadWriteOnce` (archiviazione blocchi). Ad esempio, se si utilizza Red Hat OpenShift Data Foundation, la classe di archiviazione *ocs - storagecluster - cephfs* fornisce un file system condiviso adatto. La scelta del file system è

molto importante, perché non tutti i file system condivisi gestiscono il blocco dei file nello stesso modo. Consultare Planning file system support on Multiplatforms e Test statement for IBM MQ multi - instance queue manager file systems.

2. Creare una coppia di certificati come descritto in “Creazione di una PKI autofirmata utilizzando OpenSSL” a pagina 66.
3. Crea una mappa di configurazione contenente comandi MQSC e un file INI

   Creare una Kubernetes ConfigMap contenente i comandi MQSC per creare una nuova coda e un canale SVRCONN e per aggiungere un record di autenticazione di canale che consenta l'accesso al canale.

   Assicurati di essere nello spazio dei nomi che hai creato precedentemente (vedi Prima di iniziare), quindi immetti il seguente YAML nella console web OCP o utilizzando la riga di comando.

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

   MQSC definisce un canale denominato *MTLS.SVRCONN* e una coda denominata *EXAMPLE.QUEUE*. Il canale è configurato per consentire l'accesso solo ai client che presentano un certificato con un "nome comune" *example-app1*. Questo è il nome comune utilizzato in uno dei certificati creati nel passo “2” a pagina 81. Le connessioni su questo canale con questo nome comune vengono associate a un ID utente *app1*, autorizzato a connettersi al gestore code e ad accedere alla coda di esempio. Il file INI abilita una politica di sicurezza che indica che l'ID utente *app1* non deve necessariamente esistere in un registro utente esterno - esiste solo come nome in questa configurazione.
4. Distribuisci il gestore code

   Creare un nuovo gestore code utilizzando la seguente risorsa personalizzata YAML. Assicurarsi di essere nello spazio dei nomi creato prima di iniziare questa attività, quindi immettere il seguente YAML nella console Web OCP o utilizzando la riga comandi. Verificare che sia stata specificata la licenza corretta e accettarla modificando `false` in `true`.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: exampleqm
spec:
  license:
    accept: false
    license: L-EHXT-MQCRN9
    use: Production
  queueManager:
    name: EXAMPLEQM
    availability:
      type: MultiInstance
    mqsc:
    - configMap:
        name: example-miqm-configmap
        items:
        - example-tls.mqsc
    ini:
    - configMap:
        name: example-miqm-configmap
        items:
        - example-tls.ini
    storage:
```

```
    defaultClass: STORAGE_CLASS
version: 9.4.0.0-r1
pki:
  keys:
    - name: default
      secret:
        secretName: example-qm-tls
        items:
          - tls.key
          - tls.crt
          - ca.crt
```

Modificare *STORAGE_CLASS* nella classe di memoria identificata nel Passo “1” a pagina 80.

Nota che il segreto *example - qm - tls* è stato creato al passo “2” a pagina 81e che ConfigMap *example - miqm - configmap* è stato creato al passo “3” a pagina 81

Il tipo di disponibilità è impostato su *MultiInstance*, che fa sì che la memoria persistente venga selezionata automaticamente.

5. Confermare che il gestore code è in esecuzione

   Il gestore code è in fase di distribuzione. Confermare che si trova nello stato `Running` prima di procedere. Ad esempio:

   ```
   oc get qmgr exampleqm
   ```

6. Verifica la connessione al gestore code

   Per confermare che il gestore code è configurato e disponibile, effettuare le operazioni riportate in “Verifica di una connessione TLS reciproca a un gestore code dal tuo laptop” a pagina 70.

7. Forza l'esito negativo del pod attivo

   Per convalidare il ripristino automatico del gestore code, simula un errore pod:

   a) Visualizza i pod attivi e in standby

   Esegui il seguente comando:

   ```
   oc get pods --selector app.kubernetes.io/instance=exampleqm
   ```

   Tieni presente che, nel campo **READY** , il pod attivo restituisce il valore `1/1`, mentre il pod standby restituisce il valore `0/1`.

   b) Elimina il pod attivo

   Esegui il seguente comando, specificando il nome completo del pod attivo:

   ```
   oc delete pod exampleqm-ibm-mq-value
   ```

   c) Visualizza di nuovo lo stato del pod

   Esegui il seguente comando:

   ```
   oc get pods --selector app.kubernetes.io/instance=exampleqm
   ```

   d) Visualizza lo stato del gestore code

   Esegui il seguente comando, specificando il nome completo dell'altro pod:

   ```
   oc exec -t Pod -- dspmq -x
   ```

   Dovresti vedere lo stato che mostra che l'istanza attiva è cambiata, ad esempio:

   ```
   QMNAME(EXAMPLEQM)                                         STATUS(Running as standby)
       INSTANCE(exampleqm-ibm-mq-1) MODE(Active)
       INSTANCE(exampleqm-ibm-mq-0) MODE(Standby)
   ```

   e) Verificare nuovamente la connessione al gestore code

   Per confermare che il gestore code è stato ripristinato, effettuare le operazioni riportate in “Verifica di una connessione TLS reciproca a un gestore code dal tuo laptop” a pagina 70.

Risultati

Congratulazioni, hai distribuito correttamente un gestore code a più istanze con autenticazione TLS reciproca e hai verificato che venga ripristinato automaticamente quando il pod attivo ha esito negativo.

Configurazione di un instradamento per la connessione a un gestore code dall'esterno di un cluster Red Hat OpenShift

Hai bisogno di un instradamento Red Hat OpenShift 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 è disponibile solo nel protocollo TLS quando viene utilizzato un protocollo TLS 1.2 o superiore. Red Hat OpenShift Container Platform Router utilizza SNI per instradare le richieste al gestore code IBM MQ .

Informazioni su questa attività

La configurazione richiesta dell' Red Hat OpenShift instradamento dipende dal comportamento SNI (Server Name Indication) della propria applicazione client. IBM MQ supporta due differenti impostazioni di intestazione SNI a seconda del tipo di configurazione e client. Un'intestazione SNI è impostata sul nome host della destinazione del client o, in alternativa, sul nome del canale IBM MQ . Per informazioni sul modo in cui IBM MQ associa un nome canale a un nome host, vedi How IBM MQ fornisce più capacità di certificati.

Se un'intestazione SNI è impostata su un nome di canale IBM MQ o se un nome host è controllato utilizzando l'attributo **OutboundSNI** . I valori possibili sono `OutboundSNI=CHANNEL` (il valore predefinito) o `OutboundSNI=HOSTNAME`. Per ulteriori informazioni, consultare Stanza SSL del file di configurazione client. Si noti che `CHANNEL` e `HOSTNAME` sono i valori esatti che si utilizzano; non sono nomi di variabili che si sostituiscono con un nome canale o un nome host effettivo.

Comportamenti client con impostazioni `OutboundSNI` differenti

Se **OutboundSNI** è impostato su `HOSTNAME`, i seguenti client impostano un nome host SNI purché venga fornito un nome host nel nome connessione:

- Client C
- Client .NET in modalità non gestita
- Java/JMS Client

Se **OutboundSNI** è impostato su `HOSTNAME` e viene utilizzato un indirizzo IP nel nome della connessione, i seguenti client inviano un'intestazione SNI vuota:

- Client C
- Client .NET in modalità non gestita
- Java/JMS Client (che non possono eseguire una ricerca DNS inversa del nome host)

Se **OutboundSNI** è impostato su `CHANNEL`o non è impostato, viene utilizzato un nome canale IBM MQ e viene sempre inviato, se viene utilizzato un nome host o un nome connessione indirizzo IP.

I tipi di client seguenti non supportano l'impostazione di un'intestazione SNI su un nome canale IBM MQ e quindi tentano sempre di impostare l'intestazione SNI su un nome host indipendentemente dall'impostazione **OutboundSNI** :

- Client AMQP
- Client XR

Il client IBM MQ gestito .NET imposta SERVERNAME sul rispettivo nome host se la proprietà **OutboundSNI** è impostata su `HOSTNAME`, che consente a un client IBM MQ gestito .NET di connettersi a un gestore code utilizzando gli instradamenti Red Hat OpenShift .

Se un'applicazione client si connette a un gestore code distribuito in un cluster Red Hat OpenShift tramite IBM MQ Internet Pass-Thru (MQIPT), MQIPT può essere configurato per impostare SNI sul nome host utilizzando la proprietà SSLClientOutboundSNI nella definizione di instradamento.

OutboundSNI, più certificati e instradamenti Red Hat OpenShift

IBM MQ utilizza l'intestazione SNI per fornire più funzionalità di certificati. Se un'applicazione si connette a un canale IBM MQ configurato per utilizzare un certificato differente tramite il campo CERTLABL, l'applicazione deve connettersi con un'impostazione **OutboundSNI** di CHANNEL.

Se la configurazione dell'instradamento Red Hat OpenShift richiede un HOSTNAME SNI, non è possibile utilizzare la funzionalità di più certificati di IBM MQ e non è possibile impostare un'impostazione CERTLABL su qualsiasi oggetto del canale IBM MQ .

Se un'applicazione con un'impostazione **OutboundSNI** diversa da CHANNEL si connette ad un canale con un'etichetta di certificato configurata, l'applicazione viene rifiutata con un MQRC_SSL_INITIALIZATION_ERROR e un messaggio AMQ9673 viene stampato nei log degli errori del gestore code.

Per ulteriori informazioni su come IBM MQ fornisce la funzionalità di più certificati, consultare Come IBM MQ fornisce la funzionalità di più certificati .

Esempio

Le applicazioni client che impostano SNI sul canale MQ richiedono la creazione di un nuovo instradamento Red Hat OpenShift per ogni canale a cui si desidera connettersi. È inoltre necessario utilizzare nomi canale univoci nel cluster Red Hat OpenShift Container Platform , per consentire l'instradamento al gestore code corretto.

È importante che i nomi dei canali MQ non terminino con una lettera minuscola a causa del modo in cui IBM MQ associa i nomi dei canali alle intestazioni SNI.

Per stabilire il nome host richiesto per ciascuno dei tuoi nuovi instradamenti Red Hat OpenShift , devi associare ciascun nome di canale a un indirizzo SNI. Per ulteriori informazioni, vedi How IBM MQ fornisce più capacità di certificati .

Devi quindi creare un nuovo instradamento Red Hat OpenShift per ciascun canale, applicando il seguente `yaml` nel tuo cluster:

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: unique_name_for_the_route
  namespace: namespace_of_your_MQ_deployment
spec:
  host: SNI_address_mapping_for_the_channel
  to:
    kind: Service
    name: name_of_Kubernetes_Service_for_your_MQ_deployment (for example "queue_manager_name-
ibm-mq")
  port:
    targetPort: 1414
  tls:
    termination: passthrough
```

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 "queue_manager_name-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 Red Hat OpenShift Container Platform - normalmente 443.

Attività correlate

“Connessione al IBM MQ Console distribuito in un cluster Red Hat OpenShift” a pagina 127

Come connettersi al IBM MQ Console di un gestore code distribuito su un cluster Red Hat OpenShift Container Platform .

Integrazione di IBM MQ con la traccia IBM Instana

IBM Instana può essere utilizzato per tracciare le transazioni in IBM Cloud Pak for Integration.

Prima di iniziare

Questo documento riguarda la traccia IBM Instana , che è il processo di traccia dei messaggi attraverso un sistema. Non copre il monitoraggio IBM Instana , in cui vengono richiamati i dettagli sullo stato di un gestore code IBM MQ . Per informazioni sul controllo di IBM MQ da parte di IBM Instana , consultare Monitoraggio IBM MQ. Per istruzioni dettagliate sul monitoraggio autenticato, consultare “Configurazione del monitoraggio di IBM Instana autenticato con TLS” a pagina 86.

Nota:

- Questa funzione è supportata solo su Operandi di IBM MQ versione 9.3.1.0-r2 o successiva.
- È possibile eseguire la traccia IBM Instana sulle precedenti versioni di IBM MQ Operator e gestore code, ma non in modo nativo. Consultare Configuring IBM MQ Tracing nella documentazione di IBM Instana .

Prima di poter eseguire la traccia di IBM Instana con l'operatore IBM MQ , è necessario distribuire sia un backend IBM Instana che agent IBM Instana . Per impostazione predefinita, un gestore code IBM MQ comunica con un agente IBM Instana distribuito sullo stesso nodo del pod del gestore code.

Informazioni su questa attività

L'abilitazione dell'integrazione con IBM Instana comporta l'installazione di un'uscita API IBM MQ nel tuo gestore code. L'uscita API invia i dati di traccia agli agent IBM Instana relativi ai messaggi che passano attraverso il gestore code.

L'uscita API aggiunge intestazioni RFH2 a ciascun messaggio. Queste intestazioni contengono informazioni di traccia.

Gli agent IBM Instana sono responsabili dell'invio dei dati di traccia al backend IBM Instana .

Per informazioni sulla distribuzione di un backend IBM Instana e di agent IBM Instana , vedi Abilitazione dei link di monitoraggio Instana nell'IU della piattaforma nella documentazione IBM Instana .

Procedura

Distribuzione standard

- Distribuire un gestore code con la traccia IBM Instana abilitata.

  Per impostazione predefinita, la traccia IBM Instana è disabilitata.

  Se si sta utilizzando la IBM Cloud Pak for Integration Platform UI o la console Web OpenShift :

  1. Fai clic su **Telemetria** > **Traccia** > **Instana**.
  2. Imposta l'opzione **Abilita traccia Instana** su `true`.

  Se si sta eseguendo la distribuzione tramite YAML, utilizzare il seguente frammento:

```
spec:
  telemetry:
    tracing:
      instana:
        enabled: true
```

Distribuzione avanzata

- Comunicare con l'agent IBM Instana su https.

  Per impostazione predefinita, l'uscita IBM Instana per IBM MQ comunica con IBM Instana Agent su http. L'indirizzo host dell'agent è impostato sull'indirizzo IP del nodo su cui è in esecuzione il gestore

code. Ciò corrisponde alla configurazione descritta in Abilitazione del monitoraggio IBM Instana nella documentazione IBM Instana , dove gli agent IBM Instana vengono distribuiti dall'operatore agent IBM Instana come una serie di daemon.

Attualmente la comunicazione tra l'uscita IBM Instana per IBM MQ e l'agente IBM Instana supporta i protocolli http o https. Per utilizzare https, l'agent IBM Instana deve essere prima configurato per utilizzare la codifica TLS. Vedi Impostazione della codifica TLS per l'endpoint dell'agent nella documentazione di IBM Instana . Il protocollo può quindi essere impostato su https nel modo seguente:

Se stai utilizzando la console web OpenShift :

1. Fai clic su **Telemetry** > **Instana**.
2. Espandere l'elenco a discesa **Configurazione avanzata** .
3. Impostare il **protocollo di comunicazione agent Instana** su `https`.

Se si sta eseguendo la distribuzione tramite YAML, utilizzare il seguente frammento:

```
spec:
  telemetry:
    instana:
      enabled: true
      protocol: https
```

- Impostare **`agentHost`**

  Se gli agent IBM Instana non sono stati distribuiti come serie di daemon sul cluster Openshift in cui è in esecuzione il gestore code, è necessario impostare il valore **`agentHost`** sul nome host o sull'indirizzo IP in cui è in esecuzione l'agent IBM Instana . Il valore **`agentHost`** non deve includere un protocollo o una porta.

  Se stai utilizzando la console web OpenShift :

  1. Fai clic su **Telemetry** > **Instana**.
  2. Espandere l'elenco a discesa **Configurazione avanzata** .
  3. Immetti il nome host nella casella di testo **Instana agent host** .

  Se si sta eseguendo la distribuzione tramite YAML, utilizzare il seguente frammento:

```
spec:
  telemetry:
    instana:
      enabled: true
      agentHost: 9.9.9.9
```

Operazioni successive

Consultare anche “Distribuzione di un gestore code semplice utilizzando IBM MQ Operator” a pagina 62.

OpenShift CP4I Operator2.2.0 Configurazione del monitoraggio di IBM Instana autenticato con TLS

Per poter monitorare un gestore code tramite IBM Instana , è necessario configurare sia l'agent che il gestore code.

Prima di iniziare

La sezione "Configurazione" di "Monitoraggio IBM MQ" nella documentazione IBM Instana fornisce informazioni generali relative alla configurazione del controllo IBM Instana . Tuttavia, non include dettagli sulla configurazione del gestore code.

Prima di poter eseguire la traccia di IBM Instana con l'operatore IBM MQ , è necessario distribuire sia un backend IBM Instana che agent IBM Instana . Per farlo, vedi Abilitazione del monitoraggio di IBM Instana nella CP4I Platform UI nella documentazione di IBM Instana .

Procedura

1. Genera certificati.
2. Configurare gli agent IBM Instana.
3. Configurazione del gestore code.
4. Verifica e debug.

Attività correlate

“Integrazione di IBM MQ con la traccia IBM Instana” a pagina 85
IBM Instana può essere utilizzato per tracciare le transazioni in IBM Cloud Pak for Integration.

Generare un certificato e una chiave per l'agent IBM Instana e il gestore code

Per le comunicazioni TLS tra l'agente IBM Instana e il gestore code, entrambi devono avere un certificato e la chiave privata corrispondente.

Prima di iniziare

Questa è la prima di quattro attività per configurare il monitoraggio IBM Instana autenticato con TLS.

Nota: I valori utilizzati nella creazione di questi certificati sono a scopo dimostrativo. Durante la distribuzione in un ambiente di produzione, verificare che l'oggetto e la scadenza del certificato siano appropriati.

Procedura

IBM MQ Gestore code

Per comunicare con l'agent IBM Instana tramite TLS, il gestore code deve avere un certificato e la chiave privata corrispondente. Se hai già questi, salta questa sezione.

1. Generare un certificato e una chiave privata per il gestore code.

   Esegui il seguente comando:

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

AgenteIBM Instana

Per consentire all'agent di eseguire la comunicazione TLS con il gestore code IBM MQ , l'agent deve avere un certificato e la chiave privata corrispondente. Se si dispone già di una chiave privata e di un certificato in un keystore JKS che si desidera utilizzare, ignorare questa sezione.

2. Generare un certificato e una chiave privata per l'agent IBM Instana .

   Esegui il seguente comando:

```
openssl req \
       -newkey rsa:2048 -nodes -keyout application.key \
       -subj "/CN=instana-agent/OU=app team1" \
       -x509 -days 3650 -out application.crt
```

3. Memorizzare il certificato e la chiave privata in un keystore PKCS12 .

Esegui il seguente comando, sostituendo *your_password* con la password che vuoi utilizzare per proteggere il keystore. Eseguire questa sostituzione in tutte le operazioni successive.

```
openssl pkcs12 -export -out application.p12 -inkey application.key -in application.crt
-passout pass:your_password
```

4. Convertire il keystore PKCS12 in un keystore JKS.

   Esegui il seguente comando:

```
keytool -importkeystore \
        -srckeystore application.p12 \
        -srcstoretype pkcs12 \
        -destkeystore application.jks \
        -deststoretype JKS \
        -srcstorepass your_password \
        -deststorepass your_password \
        -noprompt
```

5. Etichettare il certificato.

   Esegui il seguente comando:

```
keytool -changealias -alias "1" -destalias "instana" -keypass your_password -keystore
application.jks -storepass your_password -noprompt
```

6. Importare il certificato del gestore code nel keystore.

   Esegui il seguente comando:

```
keytool -importcert -file server.crt -keystore application.jks -storepass your_password
-alias myca -noprompt
```

Operazioni successive

Ora è possibile configurare gli agenti per il IBM Instana controllo.

Monitoraggio Instana: configurazione degli agent

Montare il keystore sugli agent IBM Instana , quindi configurare il controllo per un gestore code specifico.

Prima di iniziare

Questa attività presuppone che sia stato generato un certificato e una chiave per gli agent IBM Instana e il gestore code.

Procedura

Montaggio del keystore sugli agenti IBM Instana

1. Creare un segreto dal keystore JKS nello spazio dei nomi dell'agente IBM Instana .

   Esegui il seguente comando, sostituendo *keystore_secret_name* col nome che vuoi utilizzare. Eseguire questa sostituzione in tutte le operazioni successive.

```
oc create secret generic keystore_secret_name --from-file=./application.jks -n instana-agent
```

2. Nello spazio dei nomi instana - agent, utilizza il comando `oc edit daemonset instana-agent` per modificare il daemonset instana - agent per includere i seguenti volumi e volumeMount aggiuntivi:

```
volumeMounts:
- name: mq-key-jks-name
   subPath: application.jks
   mountPath: /opt/instana/agent/etc/application.jks
volumes:
- name: mq-key-jks-name
   secret:
      secretName: keystore_secret_name
```

Configurazione del monitoraggio per un determinato gestore code

3. Nello spazio dei nomi instana - agent, utilizza il comando `oc edit configmap instana-agent` per modificare la mappa di configurazione instana - agent.
4. Aggiungere la seguente sezione in `configuration.yaml: |`. Se questa sezione è già stata definita, è sufficiente aggiungere il nuovo gestore code all'elenco.

```
    com.instana.plugin.ibmmq:
      enabled: true
      poll_rate: 60
      queueManagers:
         QUEUE_MANAGER_NAME:
            channel: 'INSTANA.A.SVRCONN'
            keystorePassword: 'your_password'
            keystore: '/opt/instana/agent/etc/application.jks'
            cipherSuite: 'TLS_RSA_WITH_AES_256_CBC_SHA256'
```

   dove

   - *your_password* è la parola d'ordine per il keystore JKS
   - *QUEUE_MANAGER_NAME* è il nome del gestore code IBM MQ sottostante da distribuire, anziché il nome dell'operatore del gestore code.

   Nota: Se *QUEUE_MANAGER_NAME* non è impostato sul nome del gestore code sottostante ed è invece impostato su Operando, il controllo non funzionerà. Il nome sottostante è definito in `spec.queuemanager.name` per l'operatore del gestore code.
5. Elimina i pod instana - agent nello spazio nomi instana - agent. Ciò li fa riavviare e iniziare il controllo con le nuove impostazioni.

Operazioni successive

Si è ora pronti a configurare il gestore code per il monitoraggio IBM Instana.

OpenShift CP4I Operator2.2.0 *Monitoraggio Instana: configurazione del gestore code*

Configurare un gestore code che utilizza TLS per comunicare con l'agent IBM Instana .L'autenticazione per questa connessione viene eseguita utilizzando un SSLPEERMAP.

Prima di iniziare

Questa attività presuppone che l'utente abbia configurato gli agenti per il IBM Instana monitoraggio.

Procedura

1. Configurare il gestore code tramite MQSC e INI.

   MQSC viene utilizzato per impostare un nuovo canale abilitato TLS e quindi configurare tale canale per autenticare l'agent IBM Instana di connessione se dispone di un certificato con i campi richiesti. In questo caso, associamo qualsiasi client di connessione con un certificato che contiene i campi `CN=instana-agent,OU=app team1` all'utente `app1`. MQSC, quindi, concede l'autorizzazione all'utente `app1` per eseguire le operazioni richieste per il monitoraggio IBM Instana .

   Il file INI viene utilizzato per concedere autorizzazioni all'utente esterno `app1`.

   La seguente configmap contiene le impostazioni MQSC e INI richieste. Distribuiscila nel tuo spazio dei nomi del gestore code.

```
apiVersion: v1
data:
  channel.mqsc: |-
    DEFINE CHANNEL('INSTANA.A.SVRCONN') CHLTYPE(SVRCONN) SSLCAUTH(REQUIRED)
SSLCIPH('ANY_TLS12_OR_HIGHER')
    ALTER QMGR CONNAUTH(' ')
    REFRESH SECURITY
    SET CHLAUTH('INSTANA.A.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=*') USERSRC(NOACCESS)
ACTION(REPLACE)
    SET CHLAUTH('*') TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS) ACTION(REPLACE)
```

```
    SET CHLAUTH('INSTANA.A.SVRCONN') TYPE(SSLPEERMAP) SSLPEER('CN=instana-agent,OU=app
team1') USERSRC(MAP) MCAUSER('app1')
    SET AUTHREC PRINCIPAL('app1') OBJTYPE(QMGR) AUTHADD(ALL)
    SET AUTHREC PROFILE('SYSTEM.ADMIN.COMMAND.QUEUE') PRINCIPAL('app1') OBJTYPE(QUEUE)
AUTHADD(PUT,INQ,DSP,CHG)
    SET AUTHREC PROFILE('SYSTEM.**') PRINCIPAL('app1') OBJTYPE(TOPIC) AUTHADD(DSP)
    SET AUTHREC PROFILE('*') PRINCIPAL('app1') OBJTYPE(TOPIC) AUTHADD(DSP)
    SET AUTHREC PROFILE('SYSTEM.**') PRINCIPAL('app1') OBJTYPE(QUEUE) AUTHADD(DSP, CHG, GET)
    SET AUTHREC PROFILE('SYSTEM.**') PRINCIPAL('app1') OBJTYPE(LISTENER) AUTHADD(DSP)
    SET AUTHREC PROFILE('AMQ.*') PRINCIPAL('app1') OBJTYPE(QUEUE) AUTHADD(DSP, CHG)
    REFRESH SECURITY TYPE(CONNAUTH)
  auth.ini: |-
    Service:
      Name=AuthorizationService
      EntryPoints=14
      SecurityPolicy=UserExternal
kind: ConfigMap
metadata:
  namespace: your-queue-manager-namespace
  name: qmgr-monitoring-config
```

dove *your - queue - manager - namespace* è lo spazio dei nomi in cui verrà distribuito il gestore code.

Nota: Se si stanno monitorando code definite dall'utente, è necessario aggiungere ulteriori righe alla configmap MQSC, concedendo le autorizzazioni DSP, CHG e GET a tali code. Ad esempio:

```
SET AUTHREC PROFILE('MYQUEUE') PRINCIPAL('app1') OBJTYPE(QUEUE) AUTHADD(DSP, CHG, GET).
```

Questo esempio utilizza una configmap per i dati MQSC e INI, ma è possibile utilizzare un segreto se eventuali aggiunte effettuate sono riservate. Per informazioni generali sulla distribuzione con MQSC e INI, consultare “Esempio: fornitura di file MQSC e INI” a pagina 65.

2. Per stabilire una connessione TLS, il gestore code deve considerare attendibile il certificato dell'agent IBM Instana . Per ottenere ciò, creare un segreto contenente solo il certificato dell'agent IBM Instana :

```
oc create secret generic instana-certificate-secret --from-file=./application.crt -n your-
queue-manager-namespace
```

3. Il gestore code deve presentare il proprio certificato per l'handshake TLS e richiede l'accesso alla chiave privata associata. Distribuisci un segreto contenente la chiave e il certificato che hai creato in precedenza o che già possiedi:

```
oc create secret tls qm-tls-secret --cert server.crt --key server.key -n your-queue-manager-
namespace
```

Con la configmap e il segreto creati, si è pronti a creare il gestore code stesso.

4. Assicurarsi che il gestore code YAML non imposti la variabile di ambiente **MQSNOAUT** nel contenitore del gestore code.

   In caso contrario, una volta abilitato, il meccanismo di autenticazione non funzionerà. La rimozione della variabile dopo la distribuzione non comporta la riabilitazione del meccanismo e la ricreazione del gestore code.

5. Aggiungere le seguenti sezioni alla definizione del gestore code, dove *MYQM* è il nome del gestore code:

```
spec:
  queueManager:
    name: MYQM   #(a)
    ini:   #(b)
      - configMap:
          items:
            - auth.ini
          name: qmgr-monitoring-config
    mqsc:   #(c)
      - configMap:
          items:
            - channel.mqsc
          name: qmgr-monitoring-config
  pki:
    keys:   #(d)
        - name: default
```

```
          secret:
            items:
            - tls.key
            - tls.crt
            secretName: qm-tls-secret
    trust:   #(e)
      - name: app
        secret:
          items:
            - application.crt
          secretName: instana-certificate-secret
```

Le sezioni contrassegnate della specifica sono descritte come segue:

a. Assicurarsi di aver fornito al gestore code sottostante un nome univoco. Se il gestore code sottostante non ha un nome univoco, il monitoraggio potrebbe non funzionare come previsto. Questo nome deve corrispondere al nome nella mappa di configurazione dell'agent IBM Instana che è stata modificata in precedenza.

b. Le informazioni INI scritte nella configmap vengono aggiunte al gestore code.

c. Le informazioni MQSC scritte nella configmap vengono aggiunte al gestore code.

d. Il certificato del gestore code e la chiave privata vengono aggiunti al keystore del gestore code.

e. Il certificato dell'agent IBM Instana viene aggiunto al truststore del gestore code.

6. Opzionale: Abilitare IBM Instana Traccia sul gestore code monitorato.

   Se si desidera eseguire questa operazione, consultare “Integrazione di IBM MQ con la traccia IBM Instana” a pagina 85.

7. Distribuire il gestore code.

Operazioni successive

Sei ora pronto a verificare ed eseguire il debug del IBM Instana monitoraggio.

OpenShift CP4I Operator2.2.0 *Monitoraggio Instana: verifica e debug*

Per poter monitorare un gestore code tramite IBM Instana , è necessario configurare sia l'agent che il gestore code.

Prima di iniziare

Questa attività presuppone che sia stato configurato il gestore code per il monitoraggio IBM Instana.

Procedura

Verifica

1. Per verificare che la distribuzione sia stata eseguita correttamente, visualizzare il gestore code nel dashboard IBM Instana .

   Il gestore code deve essere visibile nella sezione dei servizi della pagina dell'applicazione e anche nella vista Infrastruttura.

Debug

Nota: Questi passi di debug presuppongono una distribuzione Openshift dell'agent IBM Instana in esecuzione come serie di daemon.

Se non è possibile visualizzare il gestore code nel dashboard IBM Instana , è possibile che il gestore code sia stato configurato in modo non corretto. Utilizzare la seguente procedura per indagare.

2. Identificare il nodo su cui è in esecuzione il pod del gestore code attivo.

   Immettere il seguente comando nello spazio dei nomi del gestore code:

```
oc get pods -o wide -n your-queue-manager-namespace
```

3. Per determinare quale pod dell'agent IBM Instana è in esecuzione sullo stesso nodo del tuo gestore code, esegui lo stesso comando nello spazio dei nomi instana - agent:

```
oc get pods -o wide -n instana-agent-namespace
```

4. Per facilitare la comprensione di eventuali problemi dal lato dell'agent IBM Instana , ottenere i log del pod dell'agent IBM Instana e cercare le voci relative a 'mq' o al nome del gestore code.

   Esegui il seguente comando:

```
oc logs instana-agent-pod -c instana-agent -n instana-agent
```

5. Controllare i log del gestore code.

   Se l'agent ha effettuato un tentativo di connessione al gestore code, i log del gestore code dovrebbero indicare il motivo per cui la connessione non è riuscita. Esegui il seguente comando:

```
oc logs your-queue-manager-name -n your-queue-manager-namespace
```

Risultati

Sono state completate tutte e quattro le attività per configurare il monitoraggio IBM Instana autenticato con TLS.

OpenShift CP4I Creazione di un'immagine con file MQSC e INI personalizzati, utilizzando la CLI Red Hat 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 Red Hat OpenShift per IBM Entitled Registry nel tuo progetto Red Hat OpenShift , attieniti alla procedura per Crea il segreto della chiave di titolarità.

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:9.4.0.0-r1
      RUN printf "DEFINE QLOCAL(foo) REPLACE\n" > /etc/mqm/my.mqsc \
        && printf "Channels:\n\tMQIBindType=FASTPATH\n" > /etc/mqm/my.ini
      LABEL summary "My custom MQ image"
  strategy:
    type: Docker
    dockerStrategy:
      from:
        kind: "DockerImage"
        name: "cp.icr.io/cp/ibm-mqadvanced-server-integration:9.4.0.0-r1"
      pullSecret:
        name: ibm-entitlement-key
  output:
    to:
      kind: ImageStreamTag
      name: 'mymq:latest-amd64'
```

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 fix che si desidera utilizzare (consultare “Cronologia delle release per IBM MQ Operator” a pagina 5 per i dettagli). 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 semplice utilizzando IBM MQ Operator” a pagina 62, aggiungendo la nuova immagine personalizzata in YAML.

Puoi aggiungere il seguente frammento di YAML nel tuo normale `QueueManager` YAML, dove *my - namespace* è il Red Hat OpenShift progetto/namespace 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 semplice utilizzando IBM MQ Operator” a pagina 62
Questo esempio distribuisce un gestore code "quick start", che utilizza la memoria effimera (non persistente) e disattiva la sicurezza IBM MQ . I messaggi non vengono resi persistenti durante i riavvii

del gestore code. È possibile modificare la configurazione per modificare molte impostazioni del gestore code.

OpenShift CP4I Aggiunta di annotazioni ed etichette personalizzate alle risorse del gestore code

Aggiungere annotazioni ed etichette personalizzate ai metadati di QueueManager .

Informazioni su questa attività

Le annotazioni e le etichette personalizzate vengono aggiunte a tutte le risorse tranne le PVC. Se un'annotazione o un'etichetta personalizzata corrisponde a una chiave esistente, viene utilizzato il valore impostato da IBM MQ Operator .

Procedura

- Aggiungere annotazioni personalizzate.

  Per aggiungere annotazioni personalizzate alle risorse del gestore code, incluso il pod, aggiungi le annotazioni in `metadata`. Ad esempio:

  ```
  apiVersion: mq.ibm.com/v1beta1
  kind: QueueManager
  metadata:
    name: quickstart-cp4i
    annotations:
      annotationKey: "value"
  ```

- Aggiungere etichette personalizzate.

  Per aggiungere etichette personalizzate alle risorse del gestore code, incluso il pod, aggiungere le etichette in `metadata`. Ad esempio:

  ```
  apiVersion: mq.ibm.com/v1beta1
  kind: QueueManager
  metadata:
    name: quickstart-cp4i
    labels:
      labelKey: "value"
  ```

OpenShift CP4I Disabilitazione dei controlli webhook di runtime

I controlli del webhook di runtime garantiscono che le classi di memoria siano valide per il tuo gestore code. È possibile disabilitarli per migliorare le prestazioni o perché non sono validi per il proprio ambiente.

Informazioni su questa attività

I controlli webhook di runtime vengono eseguiti sulla configurazione del gestore code. Essi verificano che le classi di memoria siano adatte per il tipo di gestore code selezionato.

È possibile scegliere di disabilitare questi controlli per diminuire il tempo impiegato per la creazione del gestore code o perché i controlli non sono validi per il proprio ambiente specifico.

Nota: Dopo aver disabilitato i controlli webhook di runtime, sono consentiti tutti i valori della classe di archiviazione. Ciò potrebbe causare un gestore code interrotto.

Procedura

- Disabilita i controlli webhook di runtime.

  Aggiungere la seguente annotazione in `metadata`. Ad esempio:

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

```
metadata:
  name: quickstart-cp4i
  annotations:
    "com.ibm.cp4i/disable-webhook-runtime-checks" : "true"
```

OpenShift CP4I Operator2.1.0 Disabilitazione degli aggiornamenti dei valori predefiniti per la specifica del gestore code

IBM MQ Operator aggiorna tutti i valori non specificati nella specifica del gestore code con i relativi valori predefiniti. È possibile disabilitare questo comportamento se si desidera evitare qualsiasi modifica alla specifica del gestore code. I campi di stato del gestore code sono ancora aggiornati.

Procedura

- Disabilita gli aggiornamenti dei valori predefiniti del gestore code.

  Aggiungere la seguente annotazione in `metadata`. Ad esempio:

  ```
  apiVersion: mq.ibm.com/v1beta1
  kind: QueueManager
  metadata:
    name: quickstart-cp4i
    annotations:
      "com.ibm.mq/write-defaults-spec" : "false"
  ```

  Nota: Gli esempi quickstart hanno questa annotazione applicata per impostazione predefinita.

Esecuzione del contenitore IBM MQ con un file system root di sola lettura

È possibile configurare il contenitore IBM MQ per l'esecuzione con un file system root di sola lettura. Ciò impedisce agli aggressori di copiare ed eseguire codice doloso nel contenitore.

Informazioni su questa attività

L'abilitazione del file system root di sola lettura rende immutabili i file del contenitore. Nel file system del contenitore, i file possono essere visualizzati ma non modificati e non possono essere creati nuovi file. I file possono essere modificati o creati solo su un sistema di file montato.

Quando è abilitato un file system root di sola lettura, vengono creati due volumi effimeri `Scratch` e `Tmp` e montati rispettivamente nelle directory `/run` e `/tmp` nel contenitore.

- Il volume `Svuotato` contiene i file, i keystore e altri file utilizzati per la configurazione del gestore code.
- Il volume `Tmp` contiene i file diagnostici, ad esempio i file RAS del gestore code.

Poiché questi volumi sono effimeri, i file su questi volumi vengono persi al riavvio del pod.

Il tipo di volume creato per i dati del gestore code dipende dal tipo di memoria. Per impostazione predefinita, viene montato un volume persistente. Oppure, se il tipo di memoria è `effimero`, viene montato un volume effimero. Se la dimensione dei dati nel volume supera il valore specificato per la proprietà **`sizeLimit`** , Kubernetes può espellere il contenitore e crearne uno nuovo.

Un file system root di sola lettura non è abilitato per impostazione predefinita. Per abilitarla, completare la seguente procedura:

Procedura

1. Utilizzare l'API `spec.securityContext` per abilitare il file system root di sola lettura.

   Per il gestore code, impostare la proprietà **`readOnlyRootFilesystem`** in “.spec.securityContext” a pagina 150 su `true`.

   IBM MQ Operator crea due volumi effimeri, `Scratch` e `Tmp`.
2. Opzionale: Impostare o modificare il tipo di memoria dei dati del gestore code.

Per impostazione predefinita, una richiesta di volume persistente è montata in `/mnt/mqm`. In alternativa, se la proprietà **type** è impostata su `effimero` in “.spec.queueManager.storage.queueManager” a pagina 148, viene creato e montato un volume effimero.

3. Per ogni volume effimero, considerare attentamente la crescita dei dati. Impostare di conseguenza il valore della proprietà **sizeLimit** , incluse le unità SI.
   - Per il volume temporaneo `Scratch` , impostare la proprietà **sizeLimit** in “.spec.queueManager.storage.scratch” a pagina 149. Il valore predefinito è "100M".
   - Per il volume temporaneo `Tmp` , impostare la proprietà **sizeLimit** in “.spec.queueManager.storage.tmp” a pagina 150. Il valore predefinito è "2Gi".
   - Se il valore **type** del volume del gestore code è impostato su `effimero`, impostare la proprietà **sizeLimit** in “.spec.queueManager.storage.queueManager” a pagina 148. Il valore predefinito è "2Gi".

OpenShift V 9.4.0 Configurazione di IBM MQ Console con un registro di base utilizzando IBM MQ Operator

Per accedere a IBM MQ Console, è possibile fornire la propria configurazione al gestore code.

Prima di iniziare

Se si sta distribuendo un gestore code con una licenza IBM MQ Advanced for Developers , è presente una semplice configurazione integrata. Consultare “YAML del gestore code di esempio che descrive come specificare le password per gli utenti admin e app” a pagina 168. Se si sta distribuendo un gestore code di licenze IBM Cloud Pak for Integration , è possibile abilitare l'integrazione con IBM Cloud Pak for Integration Keycloak per accedere a IBM MQ Console utilizzando SSO (Single Sign - On). Vedere “Connessione al IBM MQ Console distribuito in un cluster Red Hat OpenShift” a pagina 127.

Procedura

1. **Creare una password e codificarla utilizzando `securityUtility`.**

   Un `ConfigMap` viene utilizzato per memorizzare le credenziali che si utilizzano per accedere al proprio gestore code. Per migliorare la sicurezza, codificare queste credenziali con il comando securityUtility.

   In alternativa, puoi utilizzare un segreto, che protegge le credenziali nel livello Kubernetes . Tuttavia, gli strumenti di monitoraggio o di risoluzione dei problemi potrebbero esporre il file sottostante in modo non sicuro.
2. Opzionale: **Accedi alla CLI (command line interface) Red Hat OpenShift .**

   Se utilizzi la CLI OpenShift , accedi utilizzando `oc login`.

   In alternativa, puoi utilizzare la console OpenShift .
3. **Crea un `ConfigMap` con la tua configurazione.**

   Per assistenza nella creazione della configurazione XML, consultare IBM MQ Console e REST API security.

   Il seguente esempio crea un utente all'interno del gruppo `MQWebAdminGroup`. Ai componenti di `MQWebAdminGroup` viene assegnato il ruolo `MQWebAdmin` . In questo esempio:
   - **È necessario** sostituire *USERNAME* e *PASSWORD* con i propri valori. Notare che *USERNAME* viene utilizzato due volte nell'esempio.

     È necessario specificare *NAMESPACE* come quello in cui IBM MQ Operator è distribuito e dove il gestore code sarà o già è distribuito.

   a) Utilizza la console OpenShift o la riga di comando per creare il seguente `ConfigMap`:

```
kind: ConfigMap
apiVersion: v1
```

```
metadata:
  name: mqwebuserconfigmap
  namespace: NAMESPACE
data:
  mqwebuser.xml: |
    <?xml version="1.0" encoding="UTF-8"?>
    <server>
      <featureManager>
        <feature>appSecurity-2.0</feature>
        <feature>basicAuthenticationMQ-1.0</feature>
      </featureManager>
      <enterpriseApplication id="com.ibm.mq.console">
        <application-bnd>
          <security-role name="MQWebAdmin">
            <group name="MQWebAdminGroup" realm="defaultRealm"/>
          </security-role>
        </application-bnd>
      </enterpriseApplication>
      <basicRegistry id="basic" realm="defaultRealm">
        <user name="USERNAME" password="PASSWORD"/>
        <group name="MQWebAdminGroup">
          <member name="USERNAME"/>
        </group>
      </basicRegistry>
      <sslDefault sslRef="mqDefaultSSLConfig"/>
    </server>
```

b) Opzionale: Se si utilizza la riga comandi, applicare `ConfigMap`:

```
oc apply -f mqwebuserconfigmap.yaml
```

Per i passi rimanenti, scegliere una delle opzioni seguenti:

- Distribuire un nuovo gestore code con la configurazione per accedere a IBM MQ Console.
- Applicare la configurazione che fornisce l'accesso IBM MQ Console a un gestore code esistente.

4. Opzionale: **Distribuire un nuovo gestore code con la configurazione per accedere a IBM MQ Console**

a) Creare il gestore code.

Impostare i provider di autenticazione e autorizzazione su `manuale` e fornire il `ConfigMap` `mqwebuserconfigmap` appena creato mediante una delle seguenti opzioni:

- Opzione 1: tramite il gestore code YAML

  Aggiungi il seguente codice nella sezione `web` dello YAML del gestore code:

```
...
web:
    enabled: true
    console:
      authentication:
        provider: manual
      authorization:
        provider: manual
    manualConfig:
      configMap:
        name: mqwebuserconfigmap
```

- Opzione 2: tramite la vista modulo della console di OpenShift :

  i) Sulla console OpenShift , selezionare **Operatori** > **Operatori installati**.

  ii) Selezionare la distribuzione di IBM MQ Operator.

  iii) Selezionare **Gestore code** e fare clic su **Crea QueueManager**.

  iv) Selezionare le opzioni rilevanti per il gestore code.

  v) Selezionare **Web** e impostare **Abilita server Web** su `true`.

  vi) Aprire la casella di elenco **Configurazione avanzata** .

  vii) Nella casella di elenco **Console** , impostare il **provider** per **Authentication** e **Authorization** su `manual`.

  viii) Aprire la casella di elenco **Configurazione** .

ix) Aprire la casella di elenco **ConfigMap** e selezionare ConfigMap `mqwebuserconfigmap` creato nel passo “3” a pagina 96.

x) Fai clic su **Crea**.

È ora possibile accedere al IBM MQ Console del nuovo gestore code tramite le credenziali specificate nel `ConfigMap` creato nel passo “3” a pagina 96.

5. Opzionale: **Applicare la configurazione che abilita IBM MQ Console per un gestore code esistente**

   Modificare lo YAML del gestore code per cui si sta abilitando IBM MQ Console:

   a. Sulla console di OpenShift selezionare **Operatori** > **Operatori installati**.

   b. Selezionare la distribuzione di IBM MQ Operator.

   c. Selezionare **Gestore code** e selezionare il nome del proprio gestore code.

   d. Selezionare **YAML**.

   e. Sostituire la sezione `web` esistente dello YAML del gestore code con il seguente codice:

```
...
web:
    enabled: true
    console:
      authentication:
        provider: manual
      authorization:
        provider: manual
    manualConfig:
      configMap:
        name: mqwebuserconfigmap
```

   f. Fare clic su **Salva**.

   Puoi ora accedere al IBM MQ Console del tuo gestore code esistente tramite le credenziali specificate nel `ConfigMap` creato nel passo “3” a pagina 96.

OpenShift V 9.4.0 V 9.4.0 Espansione dei volumi persistenti

Se il provider di memoria supporta l'espansione del volume, utilizzare questa attività per espandere un volume persistente. A seconda del provider di memoria, l'espansione potrebbe verificarsi in linea o non in linea.

Prima di iniziare

Una corretta espansione del volume si basa sul provider di memoria per soddisfare la richiesta di espansione. Fare riferimento alla documentazione dei provider di memoria per determinare se il ridimensionamento in linea è supportato e per informazioni sulle procedure di ridimensionamento non in linea.

Se il provider di memoria non è in grado di soddisfare la richiesta di espansione, la richiesta di volume persistente potrebbe entrare in uno stato con avvertenze o errori. Se l'espansione non riesce, un amministratore OpenShift può ripristinare manualmente lo stato Reclamo volume persistente e annullare l'espansione. Consultare Ripristino da un errore durante l'espansione dei volumi nella documentazione Red Hat OpenShift .

Informazioni su questa attività

Per aiutare con la gestione dell'archiviazione persistente, Kubernetes definisce due risorse API:

- Un PersistentVolume (PV), che è una parte di archiviazione nel cluster di cui è stato eseguito il provisioning da un amministratore o di cui è stato eseguito il provisioning in modo dinamico utilizzando le classi di archiviazione. È possibile eseguire il provisioning in modo statico o dinamico.
- Un'attestazione PersistentVolume(PVC), che è una richiesta di archiviazione da parte di un utente. Funge anche da controllo di richiesta alla risorsa.

Per ulteriori informazioni, vedi Volumi persistenti nella documentazione di Kubernetes .

Avvertenza:

- Se la classe di memoria utilizzata per creare le PVC del gestore code non supporta il ridimensionamento online, si verifica il ridimensionamento offline. Durante il ridimensionamento offline l'intervento dell'utente è necessario per completare l'espansione del volume, in modo che i gestori code riscontrino un tempo di inattività.
- Per il ridimensionamento offline dei volumi condivisi per i gestori code a più istanze, sia i pod attivi che quelli in standby devono essere disattivati contemporaneamente quando si esegue l'intervento dell'utente.
- OpenShift non supporta la riduzione della dimensione delle PVC. Se si tenta di ridurre la dimensione dei volumi persistenti, il gestore code verrà impostato sullo stato 'Non riuscito'.
- Questa procedura non si applica ai volumi effimeri.

Per espandere un PV utilizzato dal contenitore IBM MQ , completa la seguente procedura.

Procedura

1. **Preparazione all'espansione dei volumi**

   a) Decidere quali volumi espandere.

   b) Determinare la classe o le classi di archiviazione utilizzate dai propri volumi.

   Ad esempio:

```
spec:
  queueManager:
    storage:
      persistedData:
        enabled: true
        type: persistent-claim
        class: ocs-storagecluster-cephfs (1)
      queueManager:
        type: persistent-claim
      recoveryLogs:
        enabled: true
        type: persistent-claim
      defaultClass: ocs-storagecluster-ceph-rbd (2)
```

   Note:

   - (1) Se il volume definisce una classe di memoria specifica, viene utilizzata dalle PVC di questo tipo.
   - (2) Se è impostato **`defaultClass`** , questa classe di memoria viene utilizzata per tutti i volumi senza una classe di memoria specifica. Se **`defaultClass`** non è impostato e un tipo di volume non ha specificato una classe, viene utilizzata la classe di archiviazione predefinita per il cluster.

   Puoi inoltre confermare la classe di archiviazione in uso descrivendo le PVC sottostanti. Ad esempio:

```
oc describe pvc pvc-name
```

   c) Verificare che la classe di memoria supporti l'espansione del volume.

   Una classe di archiviazione potrebbe avere la proprietà **`.allowVolumeExpansion`** definita:

   - Se questa proprietà è impostata su `true`, l'espansione del volume è supportata.
   - Se questa proprietà è impostata su `false`o non è definita, la classe di archiviazione non consente l'espansione del volume. In questo caso, fare riferimento alla documentazione del provider di memoria per verificare se questa funzione può essere abilitata.

   È inoltre possibile descrivere una classe di memoria per determinare se supporta l'espansione del volume. Ad esempio:

```
oc describe sc storage-class-name
```

d) Fare riferimento alla documentazione del provider di memoria per verificare se viene utilizzata una procedura online o offline per l'espansione del volume.

Una procedura offline richiede che i pod del gestore code vengano riavviati manualmente, mentre una procedura online non lo fa. Fare riferimento alla documentazione del provider di memoria per le procedure di ridimensionamento non in linea.

e) Verificare se il gestore code ha una condizione di stato con il motivo 'StorageMismatch'.

Se il gestore code ha questa condizione di stato, i volumi elencati nella condizione vengono espansi se si abilita l'espansione del volume. Se non si desidera che ciò accada, modificare i campi della dimensione associati a ciascun tipo di volume nella definizione del gestore code in modo che corrispondano alle PVC di cui è stato eseguito il provisioning. La condizione di stato viene rimossa quando questa operazione viene eseguita per tutti i volumi non corrispondenti.

2. **Espandi volumi**

Avvertenza:

- Se è stato precedentemente modificato uno dei campi della dimensione del volume nella definizione del gestore code, i volumi iniziano ad espandersi quando **`.allowVolumeExpansion`** è impostato su `true` nella definizione del gestore code.
- Il provider di memoria potrebbe avere limitazioni sulla dimensione massima di un volume a causa delle limitazioni del file system o della disponibilità dell'hardware locale. Per evitare errori, convalidare queste limitazioni nella documentazione del provider di memoria prima di espandere i volumi.
- Le riduzioni della dimensione del PVC non sono supportate da OpenShift. Se si espande la dimensione di un volume, non è possibile ridurla. Se il tuo tentativo di eseguire questa operazione non riesce, IBM MQ Operator non può riportare la PVC al suo stato originale.

Esempio di definizione del gestore code che illustra l'espansione del volume:

```
spec:
  queueManager:
    storage:
      allowVolumeExpansion: true (A)
      persistedData:
        enabled: true
        type: persistent-claim
        size: 3Gi (B)
      queueManager:
        type: persistent-claim
        size: 4Gi (B)
      recoveryLogs:
        enabled: true
        type: persistent-claim
        size: 3Gi (B)
```

a) Per consentire l'espansione del volume per il gestore code, impostare il campo **`.spec.queueManager.storage.allowVolumeExpansion`** (A) sul proprio gestore code su `true`.

b) È ora possibile aumentare i campi di dimensione (B) per qualsiasi tipo di volume abilitato. L'applicazione di queste modifiche avvierà l'espansione del volume.

3. **Convalidare che le PVC sono state ridimensionate.**

Note:

- L'espansione del volume può richiedere del tempo. Se la convalida non ha esito positivo, la prima volta considerare l'attesa di alcuni minuti e la convalida di nuovo.
- L'espansione del volume viene completata solo senza l'azione dell'utente quando viene eseguito un ridimensionamento in linea.
- Alcuni provider di memoria completano la dimensione di memoria richiesta. Il volume espanso deve avere la stessa dimensione o una dimensione maggiore della richiesta.

a) Controllare le condizioni di stato del proprio gestore code. Fare riferimento alla seguente tabella per le condizioni, le spiegazioni e le azioni suggerite.

Tabella 1. Condizioni di stato per l'archiviazione

CONDITION	**MESSAGGIO**	**Spiegazione**
`StorageMismatch`	`Storage sizes defined in the QueueManager resource do not match the capacity of one or more provisioned PVCs [pvc-list]. AllowVolumeExpansion is set to false in the QueueManager resource so the MQ Operator will not attempt to reconcile these differences.`	L'espansione del volume non si verifica perché **`.allowVolumeExpansion`** non è stato impostato su `true` nella definizione del gestore code.
`StorageExpansionPending`	`Volume expansion is pending for the following PVCs [pvc-list]`	L'espansione del volume è ancora in corso. Se questa condizione di stato persiste per un periodo di tempo prolungato, seguire la procedura riportata di seguito per raccogliere ulteriori informazioni perché potrebbe verificarsi un ridimensionamento non in linea o un errore di ridimensionamento.
`Failed`	Ci sono molti possibili messaggi relativi alla memoria che possono creare una condizione di stato `'Failed'` . Ad esempio: `'MQ Queue Manager failed to deploy: persistentvolumeclaims "<pvc>" is forbidden: only dynamically provisioned pvc can be resized and the storageclass the provisions the pvc must support resize.'`	Se il gestore code presenta condizioni di stato `'Failed'` con testo che fa riferimento alla memoria, fare riferimento al messaggio all'interno della condizione di stato. Il messaggio di esempio fornito qui è causato dall'utilizzo di una classe di memoria che non supporta l'espansione.

b) Per ogni PVC espanso, verificare che la capacità sia aumentata in modo che corrisponda o sia maggiore del valore specificato nella definizione del gestore code.

I gestori code HA potrebbero avere più PVC di ogni tipo. Per ottenere la capacità di una PVC, immetti questo comando:

```
oc get pvc pvc-name -o template --template '{{.status.capacity.storage}}'
```

c) Controlla che la PVC non abbia condizioni di stato o eventi che suggeriscono un ridimensionamento non riuscito:

```
oc describe pvc pvc-name
```

- La tua PVC potrebbe avere la condizione di stato `FileSystemResizePending` con il messaggio 'Waiting for user to (re-) start a pod to finish file system resize of volume on node'. Questa condizione di stato viene generata per i ridimensionamenti in linea e non in linea. Per un ridimensionamento online, questa condizione di stato scompare senza l'azione dell'utente una volta completato il ridimensionamento online.
- Se la tua PVC ha una condizione di evento o di stato che indica un ridimensionamento non riuscito, vedi Ripristino da un errore durante l'espansione dei volumi nella documentazione di Red Hat OpenShift .

d) Verificare che i pod del gestore code non abbiano condizioni di stato o eventi che suggeriscono un ridimensionamento non riuscito. Per le distribuzioni HA, controllare ogni replica.

```
oc describe pod queue-manager-pod-name
```

- Se il tuo pod ha un evento o una condizione di stato che indica un ridimensionamento non riuscito, vedi Ripristino da un errore durante l'espansione dei volumi nella documentazione di Red Hat OpenShift . Il testo dell'errore potrebbe aiutare a risolvere il problema o impedire che si verifichi lo stesso problema se si tenta di ridimensionare dopo il ripristino.

4. **Riavviare i pod quando si ridimensiona offline**

   Se il provider di memoria utilizza una procedura di ridimensionamento non in linea durante l'espansione dei volumi, per completare l'espansione del volume è necessario riavviare i pod del gestore code che montano i volumi ridimensionati.

   Per i gestori code a più istanze, i log di ripristino e i volumi di dati persistenti sono condivisi tra i pod attivi e in standby. Per il completamento del ridimensionamento di questi volumi, ridurre entrambi i pod contemporaneamente.

   Fare riferimento alla documentazione del provider di memoria per la procedura di ridimensionamento offline.

OpenShift V 9.4.0 V 9.4.0 Arresto di un gestore code (`mq.ibm.com/stop`)

Arrestare un gestore code aggiungendo un'annotazione alla definizione del gestore code.

Informazioni su questa attività

I gestori code creati dall'operatore IBM MQ hanno un `StatefulSet`associato. Questo `StatefulSet` dichiara il numero di `Pods` da distribuire per un determinato tipo di disponibilità del gestore code mediante il campo '`.replicas`'. Assume il valore di 1 (Istanza singola), 2 (Istanza multipla) o 3 (NativeHA).

Nota: La modifica manuale del valore nel campo '`.replicas`' impedisce al gestore code di funzionare correttamente.

In alcuni casi, è possibile che si desideri arrestare il gestore code in modo che `StatefulSet` abbia un conteggio di repliche pari a 0 e non venga distribuito alcun `Pods` . Esempi di quando si potrebbe voler eseguire questa operazione includono durante la manutenzione o una procedura di backup.

Nota: Poiché non vi sono gestori code `Pods` distribuiti quando il gestore code viene arrestato, l'utente e le applicazioni non saranno in grado di accedere al gestore code fino a quando non viene riavviato.

Procedura

- Per arrestare il gestore code, aggiungere la seguente annotazione alla definizione del gestore code nella sezione '`.metadata.annotations`'.

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

```
  name: my-qm
  annotations:
    "mq.ibm.com/stop" : "true"
```

- Per riavviare il gestore code e riportarlo al suo corretto numero di repliche, rimuovere l'annotazione dal gestore code o impostare il suo valore su '`false`'.

Distribuzione e configurazione dei gestori code utilizzando Helm

È possibile distribuire e configurare un gestore code su Kubernetes utilizzando il grafico Helm di esempio.

Informazioni su questa attività

Se non stai utilizzando Red Hat OpenShift Container Platform, il IBM MQ Operator non è supportato. Puoi utilizzare il grafico Helm di esempio per la distribuzione su altri tipi di cluster Kubernetes .

Procedura

- Per informazioni su come utilizzare Helm per distribuire la propria immagine del contenitore IBM MQ , vedi il grafico Esempio IBM MQ Helm

Riferimenti correlati

“Supporto per IBM MQ nei contenitori” a pagina 8
Non tutte le funzioni IBM MQ sono disponibili e supportate nello stesso modo nei contenitori.

OpenShift CD CP4I-SC2 Migrazione a IBM MQ Operator

Questa serie di argomenti descrive i passi chiave per la migrazione di un gestore code IBM MQ esistente in un ambiente contenitore utilizzando IBM MQ Operator in Red Hat OpenShift Container Platform.

Informazioni su questa attività

I client che distribuiscono IBM MQ su Red Hat OpenShift possono essere separati nei seguenti scenari:

1. Creazione di una nuova distribuzione IBM MQ in Red Hat OpenShift per nuove applicazioni.
2. Estensione di una rete IBM MQ in Red Hat OpenShift per nuove applicazioni in Red Hat OpenShift.
3. Spostamento di una distribuzione IBM MQ in Red Hat OpenShift per continuare a supportare le applicazioni esistenti.

È solo per lo scenario 3 che è necessario eseguire la migrazione della configurazione IBM MQ . Gli altri scenari sono considerati nuove distribuzioni.

Questa serie di argomenti è incentrata sullo scenario 3 e descrive i passi chiave per migrare un gestore code IBM MQ esistente in un ambiente contenitore utilizzando IBM MQ Operator. A causa della flessibilità e dell'ampio uso di IBM MQ, ci sono diversi passi facoltativi. Ognuno di questi include una sezione "Ho bisogno di fare questo". La verifica delle proprie esigenze consente di risparmiare tempo durante la migrazione.

È inoltre necessario considerare quali dati migrare:

1. Migrare IBM MQ con la stessa configurazione ma senza alcun messaggio accodato esistente.
2. Migrare IBM MQ con la stessa configurazione e messaggi esistenti.

Una tipica migrazione da versione a versione può utilizzare entrambi gli approcci. In un tipico gestore code IBM MQ nel punto di migrazione, vi sono pochi messaggi memorizzati nelle code, il che rende l'opzione 1 appropriata per molti casi. In caso di migrazione a una piattaforma contenitore, è ancora più comune utilizzare l'opzione 1, per ridurre la complessità della migrazione e consentire una distribuzione verde blu. Pertanto, le istruzioni si concentrano su questo scenario.

L'obiettivo di questo scenario è creare un gestore code nell'ambiente del contenitore che corrisponda alla definizione del gestore code esistente. Ciò consente alle applicazioni collegate alla rete esistenti di

essere semplicemente riconfigurate per puntare al nuovo gestore code, senza modificare alcuna altra configurazione o logica dell'applicazione.

Durante questa migrazione si generano più file di configurazione da applicare al nuovo gestore code. Per semplificare la gestione di questi file, è necessario creare una directory e generarli in tale directory.

Procedura

1. “Verifica della disponibilità delle funzioni richieste” a pagina 104
2. “Estrazione della configurazione del gestore code” a pagina 105
3. Opzionale: “Facoltativo: estrazione e acquisizione delle chiavi e dei certificati del gestore code” a pagina 105
4. Opzionale: “Facoltativo: configurazione di LDAP” a pagina 107
5. Opzionale: “Facoltativo: modifica degli indirizzi IP e dei nomi host nella configurazione IBM MQ” a pagina 115
6. “Aggiornamento della configurazione del gestore code per un ambiente contenitore” a pagina 116
7. “Selezione dell'architettura HA di destinazione per IBM MQ in esecuzione nei contenitori” a pagina 119
8. “Creazione delle risorse per il gestore code” a pagina 120
9. “Creazione del nuovo gestore code su Red Hat OpenShift” a pagina 121
10. “Verifica della nuova distribuzione del contenitore” a pagina 125

OpenShift CD CP4I-SC2 Verifica della disponibilità delle funzioni richieste

IBM MQ Operator non include tutte le funzioni disponibili in IBM MQ Advanceded è necessario verificare che tali funzioni non siano richieste. Altre funzioni sono parzialmente supportate e possono essere riconfigurate in modo da corrispondere a quanto disponibile nel contenitore.

Prima di iniziare

Questo è il primo passo in “Migrazione a IBM MQ Operator” a pagina 103.

Procedura

1. Verificare che l'immagine del contenitore di destinazione includa tutte le funzioni richieste.

   Per le informazioni più recenti, consultare “Scelta della modalità di utilizzo di IBM MQ nei contenitori” a pagina 8.
2. IBM MQ Operator ha una singola porta di traffico IBM MQ , nota come listener. Se si dispone di più listener, semplificarlo per utilizzare un singolo listener nel contenitore. Poiché questo non è uno scenario comune, questa modifica non viene documentata in dettaglio.
3. Se vengono utilizzate le uscite IBM MQ , eseguirne la migrazione nel contenitore eseguendo il layering nei file binari di uscita IBM MQ . Questo è uno scenario di migrazione avanzata e quindi non incluso qui. Per una descrizione della procedura, vedere “Creazione di un'immagine con file MQSC e INI personalizzati, utilizzando la CLI Red Hat OpenShift” a pagina 92.
4. Se il sistema IBM MQ include l'alta disponibilità, esaminare le opzioni disponibili.

   Consultare “Pianificazione dell'alta disponibilità per IBM MQ nei contenitori” a pagina 17.

Operazioni successive

È ora possibile estrarre la configurazione del gestore code.

OpenShift CD CP4I-SC2 Estrazione della configurazione del gestore code

La maggior parte della configurazione è portabile tra gestori code. Ad esempio, gli elementi con cui interagiscono le applicazioni, come le definizioni di code, argomenti e canali. Utilizzare questa attività per estrarre la configurazione dal gestore code IBM MQ esistente.

Prima di iniziare

Questa attività presuppone che sia stato verificato che le funzioni richieste siano disponibili.

Procedura

1. Accedere alla macchina con l'installazione esistente di IBM MQ .
2. Eseguire il backup della configurazione.

   Esegui il seguente comando:

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

   Note di utilizzo per questo comando:

   - Questo comando memorizzi il backup nella directory `tmp` . È possibile memorizzare il backup in un'altra posizione, ma questo scenario presuppone la directory `tmp` per i comandi successivi.
   - Sostituisci *QMGR_NAME* con il nome del gestore code dal tuo ambiente. Se non si è certi del valore, eseguire il comando **dspmq** per visualizzare i gestori code disponibili sulla macchina. Di seguito è riportato l'output del comando **dspmq** di esempio per il gestore code denominato `qm1`:

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

     Il comando **dspmq** richiede l'avvio del gestore code IBM MQ , altrimenti si riceve il seguente errore:

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

     Se necessario, avviare il gestore code immettendo il seguente comando:

     ```
     strmqm QMGR_NAME
     ```

Operazioni successive

Sei ora pronto a estrarre e acquisire le chiavi e i certificati del gestore code.

OpenShift CD CP4I-SC2 Facoltativo: estrazione e acquisizione delle chiavi e dei certificati del gestore code

IBM MQ può essere configurato per codificare il traffico di rete nel gestore code con TLS. Utilizzare questa attività per verificare che il gestore code stia utilizzando TLS, per estrarre chiavi e certificati e per configurare TLS sul gestore code migrato.

Prima di iniziare

Questa attività presuppone che sia stata estratta la configurazione del gestore code.

Informazioni su questa attività

È necessario?

IBM MQ può essere configurato per crittografare il traffico nel gestore code. Questa codifica viene completata utilizzando un repository delle chiavi configurato nel gestore code. I canali IBM MQ quindi

abilitano la comunicazione TLS. Se non si è certi che la comunicazione TLS sia configurata nel proprio ambiente, eseguire il seguente comando per verificare:

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

Se non viene trovato alcun risultato, TLS non viene utilizzato. Tuttavia, ciò non significa che TLS non debba essere configurato nel gestore code migrato. Esistono diversi motivi per cui si potrebbe voler modificare questo comportamento:

- L'approccio di sicurezza nell'ambiente Red Hat OpenShift deve essere migliorato rispetto all'ambiente precedente.
- Se devi accedere al gestore code migrato dall'esterno dell'ambiente Red Hat OpenShift , TLS è richiesto per passare attraverso l'instradamento Red Hat OpenShift .

Nota: I certificati del gestore code con lo stesso DN (Distinguished Name) dell'emittente (CA) non sono supportati. Un certificato deve avere un DN (Distinguished Name) soggetto univoco. Il prodotto verifica che i DN non siano uguali.

Procedura

1. Estrarre i certificati attendibili dall'archivio esistente.

   Se TLS è attualmente in uso sul gestore code, il gestore code potrebbe avere un numero di certificati attendibili memorizzati. Questi devono essere estratti e copiati nel nuovo gestore code. Completare una delle seguenti operazioni facoltative:

   - Per semplificare l'estrazione dei certificati, eseguire il seguente script sul sistema locale:

     ```
     #!/bin/bash

     keyr=$(grep SSLKEYR $1)
     if [ -n "${keyr}" ]; then
        keyrlocation=$(sed -n "s/^.*'\(.*\)'.*$/\1/ p" <<< ${keyr})
        mapfile -t runmqakmResult < <(runmqakm -cert -list -db ${keyrlocation}.kdb -stashed)
        cert=1
        for i in "${runmqakmResult[@]:2}"
        do
           certlabel=$(echo ${i:2} | xargs)
           echo Extracting certificate $certlabel to $cert.cert
           runmqakm -cert -extract -db ${keyrlocation}.kdb -label "$certlabel" -target $
     {cert}.cert -stashed
           cert=$[$cert+1]
        done
     fi
     ```

     Quando si esegue lo script, specificare l'ubicazione del backup IBM MQ come argomento e i certificati vengono estratti. Ad esempio, se lo script è denominato `extractCert.sh` e il backup di IBM MQ si trova in `/tmp/backup.mqsc` , eseguire il seguente comando:

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

   - In alternativa, eseguire i seguenti comandi nell'ordine mostrato:

     a. Identificare l'ubicazione del repository delle chiavi TLS del gestore code:

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

        Output di esempio:

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

        dove il keystore si trova in `/run/runmqserver/tls/key.kdb`

b. In base a queste informazioni sull'ubicazione, interrogare il keystore per determinare i certificati archiviati:

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

Output di esempio:

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

c. Estrarre ciascuno dei certificati elencati. Eseguire questa operazione immettendo il seguente comando:

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

Negli esempi precedentemente visualizzati, ciò equivale ai seguenti comandi:

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

2. Acquisire una nuova chiave e un certificato per il gestore code

Per configurare TLS sul gestore code migrato, generare una nuova chiave e un nuovo certificato. Viene quindi utilizzato durante la distribuzione. In molte organizzazioni ciò significa contattare il team di sicurezza per richiedere una chiave e un certificato. In alcune organizzazioni questa opzione non è disponibile e vengono utilizzati i certificati autofirmati.

Il seguente esempio genera un certificato autofirmato in cui la scadenza è impostata su 10 anni:

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

Vengono creati due nuovi file:

- `qmgr.key` è la chiave privata per il gestore code
- `qmgr.crt` è il certificato pubblico

Operazioni successive

Ora è possibile configurare LDAP.

OpenShift CD CP4I-SC2 Facoltativo: configurazione di LDAP

IBM MQ Operator può essere configurato per utilizzare diversi approcci di sicurezza. Di solito LDAP è il più efficace per una distribuzione enterprise e LDAP viene utilizzato per questo scenario di migrazione.

Prima di iniziare

Questa attività presuppone che l'utente abbia estratto e acquisito le chiavi e i certificati del gestore code.

Informazioni su questa attività

È necessario?

Se si sta già utilizzando LDAP per l'autenticazione e l'autorizzazione, non è richiesta alcuna modifica.

Se non si è certi dell'utilizzo di LDAP, eseguire il seguente comando:

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

Output di esempio:

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

Nell'output sono presenti due attributi di particolare interesse:

AUTHTYPE
Se ha il valore IDPWLDAP, si sta utilizzando LDAP per l'autenticazione.

Se il valore è vuoto o un altro valore, LDAP non è configurato. In questo caso, controllare l'attributo AUTHORMD per verificare se gli utenti LDAP vengono utilizzati per l'autorizzazione.

AUTHORMD
Se questo ha il valore OS, non si sta utilizzando LDAP per l'autorizzazione.

Per modificare l'autenticazione e l'autorizzazione per utilizzare LDAP, completare le seguenti attività:

Procedura

1. Aggiornare il backup IBM MQ per il server LDAP.
2. Aggiornare il backup IBM MQ per informazioni di autorizzazione LDAP.

OpenShift CD CP4I-SC2 Parte 1 LDAP: aggiornamento del backup IBM MQ per il server LDAP

Una descrizione completa di come impostare LDAP non rientra nell'ambito di questo scenario. Questo argomento fornisce un riepilogo del processo, un esempio e riferimenti a ulteriori informazioni.

Prima di iniziare

Questa attività presuppone che l'utente abbia estratto e acquisito le chiavi e i certificati del gestore code.

Informazioni su questa attività

È necessario?

Se si sta già utilizzando LDAP per l'autenticazione e l'autorizzazione, non è richiesta alcuna modifica. Se non si è certi dell'utilizzo di LDAP, consultare “Facoltativo: configurazione di LDAP” a pagina 107.

Ci sono due parti per impostare il server LDAP:

1. Definire una configurazione LDAP.
2. Associare la configurazione LDAP alla definizione del gestore code.

Ulteriori informazioni di supporto per questa configurazione:

- Panoramica repository utente
- Guida di riferimento al comando AUTHINFO

Procedura

1. Definire una configurazione LDAP.

   Modificare il file di `backup.mqsc` per definire un nuovo oggetto **AUTHINFO** per il sistema LDAP. Ad esempio:

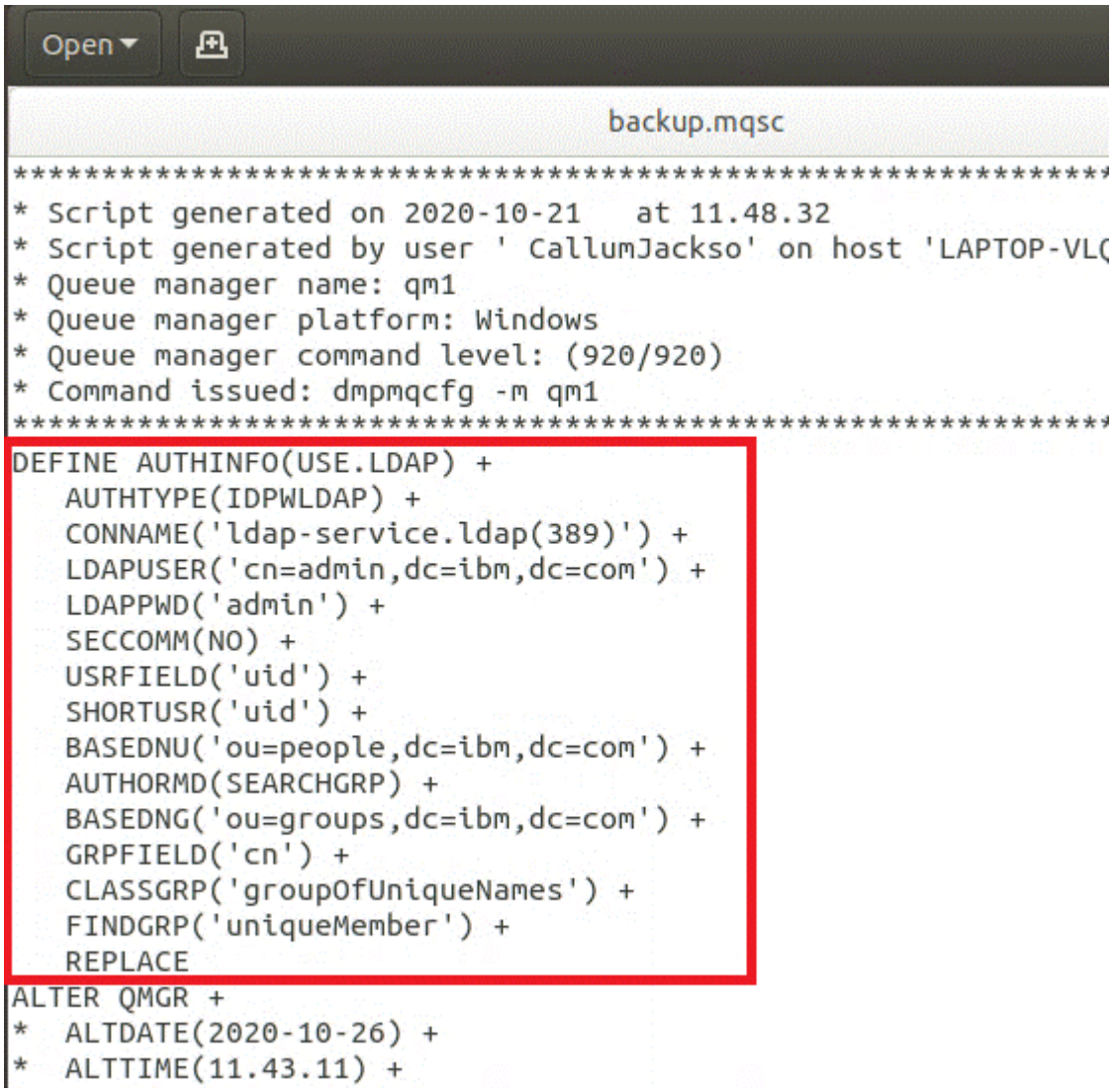
   ```
   DEFINE AUTHINFO(USE.LDAP) +
      AUTHTYPE(IDPWLDAP) +
      CONNAME('ldap-service.ldap(389)') +
      LDAPUSER('cn=admin,dc=ibm,dc=com') +
      LDAPPWD('admin') +
      SECCOMM(NO) +
      USRFIELD('uid') +
      SHORTUSR('uid') +
      BASEDNU('ou=people,dc=ibm,dc=com') +
      AUTHORMD(SEARCHGRP) +
      BASEDNG('ou=groups,dc=ibm,dc=com') +
      GRPFIELD('cn') +
      CLASSGRP('groupOfUniqueNames') +
      FINDGRP('uniqueMember')
      REPLACE
   ```

   dove

   - **CONNAME** è il nome host e la porta corrispondenti al server LDAP. Se esistono più indirizzi per la resilienza, questi possono essere configurati utilizzando un elenco separato da virgole.
   - **LDAPUSER** è il DN (distinguished name) corrispondente all'utente che IBM MQ utilizza durante la connessione a LDAP per interrogare i record utente.
   - **LDAPPWD** è la password che corrisponde all'utente **LDAPUSER** .
   - **SECCOM** specifica se la comunicazione con il server LDAP deve utilizzare TLS. I valori possibili sono:
     - YES: viene utilizzato TLS e il server IBM MQ presenta un certificato.
     - ANON: TLS viene utilizzato senza un certificato presentato dal server IBM MQ .
     - NO: TLS non viene utilizzato durante la connessione.
   - **USRFIELD** specifica il campo nel record LDAP rispetto al quale deve corrispondere il nome utente presentato.
   - **SHORTUSR** è un campo all'interno del record LDAP che non supera i 12 caratteri di lunghezza. Il valore all'interno di questo campo è l'identità asserita se l'autenticazione ha esito positivo.
   - **BASEDNU** è il DN di base da utilizzare per la ricerca LDAP.
   - **BASEDNG** è il DN di base per i gruppi all'interno di LDAP.
   - **AUTHORMD** definisce il meccanismo utilizzato per risolvere l'appartenenza al gruppo per l'utente. Esistono quattro opzioni:
     - SO: interrogare il sistema operativo per i gruppi associati al nome breve.
     - SEARCHGRP: ricercare le voci del gruppo in LDAP per l'utente autenticato.
     - SEARCHUS: ricercare nel record utente autenticato le informazioni di appartenenza al gruppo.
     - SRCHGRPSN: ricercare le voci gruppo in LDAP per il nome utente breve degli utenti autenticati (definito dal campo SHORTUSR).
   - **GRPFIELD** è l'attributo all'interno del record del gruppo LDAP che corrisponde a un nome semplice. Se specificato, può essere utilizzato per definire i record di autorizzazione.
   - **CLASSUSR** è la classe oggetto LDAP che corrisponde a un utente.
   - **CLASSGRP** è la classe di oggetti LDAP che corrisponde a un gruppo.

- **FINDGRP** è l'attributo all'interno del record LDAP che corrisponde all'appartenenza al gruppo.

La nuova voce può essere posizionata in qualsiasi punto all'interno del file, tuttavia potrebbe essere utile avere delle nuove voci all'inizio del file:



2. Associare la configurazione LDAP alla definizione del gestore code.

   È necessario associare la configurazione LDAP alla definizione del gestore code. Immediatamente sotto la voce `DEFINE AUTHINFO` è presente una voce `ALTER QMGR`. Modificare la voce CONNAUTH in modo che corrisponda al nome AUTHINFO appena creato. Ad esempio, nell'esempio precedente, `AUTHINFO(USE.LDAP)` è stato definito, il che significa che il nome è `USE.LDAP`. Modificare quindi `CONNAUTH('SYSTEM.DEFAULT.AUTHINFO.IDPWOS')` in `CONNAUTH('USE.LDAP')`:

Per fare in modo che il passaggio a LDAP avvenga immediatamente, richiamare un comando REFRESH SECURITY aggiungendo una riga immediatamente dopo il comando `ALTER QMGR` :

*backup.mqsc

```
************************************************************************
* Script generated on 2020-10-21   at 11.48.32
* Script generated by user ' CallumJackso' on host 'LAPTOP-VLQKJ5UH'
* Queue manager name: qm1
* Queue manager platform: Windows
* Queue manager command level: (920/920)
* Command issued: dmpmqcfg -m qm1
************************************************************************
DEFINE AUTHINFO(USE.LDAP) +
   AUTHTYPE(IDPWLDAP) +
   CONNAME('ldap-service.ldap(389)') +
   LDAPUSER('cn=admin,dc=ibm,dc=com') +
   LDAPPWD('admin') +
   SECCOMM(NO) +
   USRFIELD('uid') +
   SHORTUSR('uid') +
   BASEDNU('ou=people,dc=ibm,dc=com') +
   AUTHORMD(SEARCHGRP) +
   BASEDNG('ou=groups,dc=ibm,dc=com') +
   GRPFIELD('cn') +
   CLASSGRP('groupOfUniqueNames') +
   FINDGRP('uniqueMember') +
   REPLACE
ALTER QMGR +
*  ALTDATE(2020-10-26) +
*  ALTTIME(11.43.11) +
   CCSID(850) +
   CERTLABL('default') +
   CLWLUSEQ(LOCAL) +
*  COMMANDQ(SYSTEM.ADMIN.COMMAND.QUEUE) +
   CONNAUTH('USE.LDAP') +
*  CRDATE(2020-10-26) +
*  CRTIME(11.43.11) +
*  QMID(qm1_2020-10-26_11.43.11) +
   SSLCRYP(' ') +
   SSLKEYR('/run/runmqserver/tls/key') +
   SUITEB(NONE) +
*  VERSION(09020000) +
   FORCE
REFRESH SECURITY
```

Operazioni successive

Ora è possibile aggiornare il backup IBM MQ per le informazioni di autorizzazione LDAP.

OpenShift CD CP4I-SC2 Parte LDAP 2: aggiornamento del backup IBM MQ per le informazioni di autorizzazione LDAP

IBM MQ fornisce regole di autorizzazione dettagliate che controllano l'accesso agli oggetti IBM MQ . Se l'autenticazione e l'autorizzazione sono state modificate in LDAP, le regole di autorizzazione potrebbero non essere valide e richiedere l'aggiornamento.

Prima di iniziare

Questa attività presuppone che sia stato aggiornato il backup per il server LDAP.

Informazioni su questa attività

È necessario?

Se si sta già utilizzando LDAP per l'autenticazione e l'autorizzazione, non è richiesta alcuna modifica. Se non si è certi dell'utilizzo di LDAP, consultare “Facoltativo: configurazione di LDAP” a pagina 107.

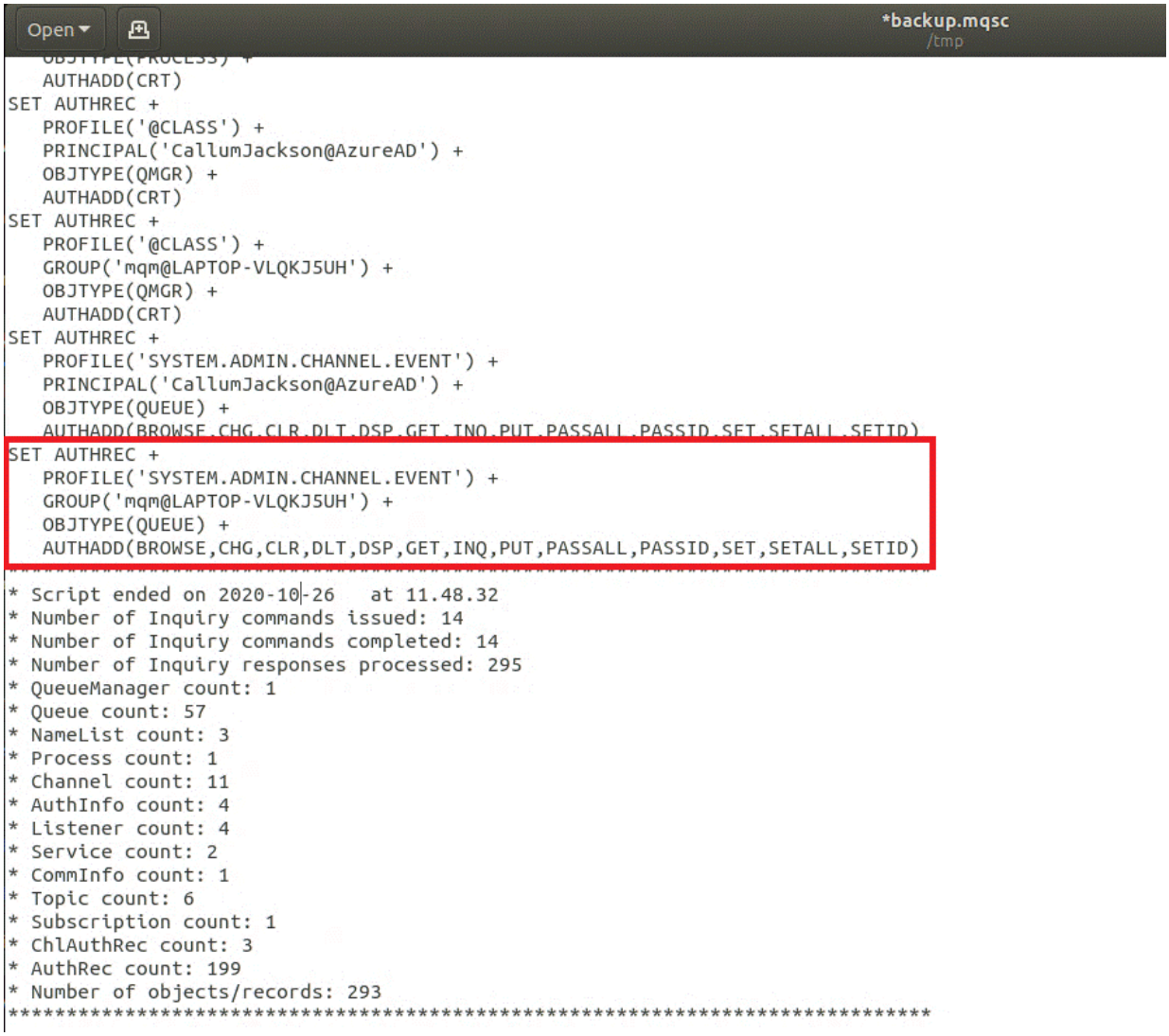
Esistono due parti per aggiornare le informazioni di autorizzazione LDAP:

1. Rimuovere tutte le autorizzazioni esistenti dal file.
2. Definire nuove informazioni di autorizzazione per LDAP.

Procedura

1. Rimuovere tutte le autorizzazioni esistenti dal file.

   Nel file di backup, vicino alla fine del file, vengono visualizzate diverse voci che iniziano con SET AUTHREC:



Individuare le voci esistenti ed eliminarle. L'approccio più semplice consiste nel rimuovere tutte le regole SET AUTHREC esistenti, quindi creare nuove voci basate sulle voci LDAP.

2. Definire nuove informazioni di autorizzazione per LDAP

   A seconda della configurazione del gestore code e del numero di risorse e gruppi, questa potrebbe essere un'attività che richiede tempo o semplice. Nel seguente esempio si assume che il gestore code abbia solo una singola coda denominata Q1e si desidera consentire l'accesso al gruppo LDAP apps .

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

   Il primo comando AUTHREC aggiunge l'autorizzazione per accedere al gestore code e il secondo fornisce l'accesso alla coda. Se è richiesto l'accesso a una seconda coda, è necessario un terzo comando AUTHREC , a meno che non si decida di utilizzare i caratteri jolly per fornire un accesso più generico.

   Ecco un altro esempio. Se un gruppo di amministratori (denominato `admins`) ha bisogno di accesso completo al gestore code, aggiungere i seguenti comandi:

```
SET AUTHREC PROFILE('*') OBJTYPE(QUEUE) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(TOPIC) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(CHANNEL) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(CLNTCONN) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(AUTHINFO) GROUP('admins') AUTHADD(ALL)
```

```
SET AUTHREC PROFILE('*') OBJTYPE(LISTENER) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(NAMELIST) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(PROCESS) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(SERVICE) GROUP('admins') AUTHADD(ALL)
SET AUTHREC PROFILE('*') OBJTYPE(QMGR) GROUP('admins') AUTHADD(ALL)
```

Operazioni successive

Sei ora pronto a modificare gli indirizzi IP e i nomi host nella IBM MQ configurazione.

OpenShift CD CP4I-SC2 Facoltativo: modifica degli indirizzi IP e dei nomi host nella configurazione IBM MQ

Per la configurazione di IBM MQ potrebbero essere specificati indirizzi IP e nomi host. In alcune situazioni queste possono rimanere, mentre in altre situazioni devono essere aggiornate.

Prima di iniziare

Questa attività presuppone che sia stato configurato LDAP.

Informazioni su questa attività

È necessario?

Innanzitutto, determinare se si dispone di indirizzi IP o nomi host specificati, a parte la configurazione LDAP definita nella sezione precedente. A tale scopo, eseguire il seguente comando:

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

Output di esempio:

```
****************************************************
DEFINE AUTHINFO(USE.LDAP) +
   AUTHTYPE(IDPWLDAP) +
   CONNAME('ldap-service.ldap(389)') +
--
DEFINE AUTHINFO('SYSTEM.DEFAULT.AUTHINFO.IDPWLDAP') +
   AUTHTYPE(IDPWLDAP) +
   ADOPTCTX(YES) +
   CONNAME(' ') +
--
   REPLACE
DEFINE AUTHINFO('SYSTEM.DEFAULT.AUTHINFO.CRLLDAP') +
   AUTHTYPE(CRLLDAP) +
   CONNAME(' ') +
```

In questo esempio la ricerca restituisce tre risultati. Un risultato corrisponde alla configurazione LDAP definita in precedenza. Questo può essere ignorato, perché il nome host del server LDAP rimane lo stesso. Gli altri due risultati sono voci di connessione vuote, quindi possono essere ignorati. Se non si dispone di voci aggiuntive, è possibile ignorare il resto di questo argomento.

Procedura

1. Comprendere le voci restituite.

   IBM MQ può includere indirizzi IP, nomi host e porte in molti aspetti della configurazione. Possiamo classificarli in due categorie:

   a. **Posizione di questo gestore code**: le informazioni sull'ubicazione che questo gestore code utilizza o pubblica, che altri gestori code o applicazioni all'interno di una rete IBM MQ possono utilizzare per la connettività.

   b. **Ubicazione delle dipendenze del gestore code**: le ubicazioni di altri gestori code o sistemi di cui questo gestore code deve essere a conoscenza.

Poiché questo scenario è concentrato solo sulle modifiche a questa configurazione del gestore code, gestiamo solo gli aggiornamenti di configurazione per la categoria (a). Tuttavia, se a questa ubicazione del gestore code fanno riferimento altri gestori code o applicazioni, potrebbe essere necessario aggiornare le relative configurazioni per corrispondere alla nuova ubicazione di questo gestore code.

Ci sono due oggetti chiave che potrebbero contenere informazioni che devono essere aggiornate:

- Listener: rappresentano l'indirizzo di rete su cui è in ascolto IBM MQ .
- CLUSTER RECEIVER canale: se il gestore code fa parte di un cluster IBM MQ , questo oggetto esiste. Specifica l'indirizzo di rete a cui altri gestori code possono connettersi.

2. Nell'output originale del comando `grep 'CONNAME\|LOCLADDR\|IPADDRV' -B 3 backup.mqsc` , identificare se sono definiti canali CLUSTER RECEIVER. In caso affermativo, aggiornare gli indirizzi IP.

   Per identificare se sono definiti canali CLUSTER RECEIVER, individuare le voci con CHLTYPE (CLUSRCVR) nell'output originale:

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

   Se le voci esistono, aggiornare CONNAME con l'instradamento IBM MQ Red Hat OpenShift . Questo valore è basato sull'ambiente Red Hat OpenShift e utilizza una sintassi prevedibile:

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

   Ad esempio, se la distribuzione del gestore code è denominata qm1 all'interno dello spazio dei nomi cp4i e *openshift_app_route_hostname* è `apps.callumj.icp4i.com`, l'URL di instradamento è il seguente:

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

   Il numero di porta per l'instradamento è generalmente 443. A meno che l'amministratore di Red Hat OpenShift non lo dica in modo diverso, questo è di solito il valore corretto. Utilizzando queste informazioni, aggiornare i campi CONNAME . Ad esempio:

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

   Nell'output originale del comando `grep 'CONNAME\|LOCLADDR\|IPADDRV' -B 3 backup.mqsc` , verificare se esistono voci per LOCLADDR o IPADDRV. Se lo fanno, eliminarli. Non sono rilevanti in un ambiente contenitore.

Operazioni successive

Si è ora pronti ad aggiornare la configurazione del gestore code per un ambiente contenitore.

OpenShift CD CP4I-SC2 Aggiornamento della configurazione del gestore code per un ambiente contenitore

Quando è in esecuzione in un contenitore, alcuni aspetti della configurazione sono definiti dal contenitore e potrebbero essere in conflitto con la configurazione esportata.

Prima di iniziare

Questa attività presuppone che si disponga di ha modificato la configurazione IBM MQ di indirizzi IP e nomi host.

Informazioni su questa attività

I seguenti aspetti di configurazione sono definiti dal contenitore:

- Le definizioni del listener (che corrispondono alle porte esposte).
- L'ubicazione di qualsiasi archivio TLS potenziale.

Pertanto, è necessario aggiornare la configurazione esportata:

1. Rimuovere tutte le definizioni di listener.
2. Definire l'ubicazione del repository delle chiavi TLS.

Procedura

1. Rimuovere tutte le definizioni di listener.

   Nella configurazione di backup, cercare `DEFINE LISTENER`. Deve essere compreso tra le definizioni `AUTHINFO` e `SERVICE` . Evidenziare l'area ed eliminarla.

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

2. Definire l'ubicazione del repository chiavi TLS.

   Il backup del gestore code contiene la configurazione TLS per l'ambiente originale. Questo è diverso dall'ambiente del contenitore e quindi sono necessari un paio di aggiornamenti:

   - Modificare la voce **CERTLABL** in `default`
   - Modificare l'ubicazione del repository delle chiavi TLS (**SSLKEYR**) in `/run/runmqserver/tls/key`

   Per trovare l'ubicazione dell'attributo **SSLKEYR** nel file, ricercare **SSLKEYR**. Di solito viene trovata una sola voce. Se vengono trovate più voci, verificare che si stia modificando l'oggetto **QMGR** come mostrato nella seguente figura:

*backup.mqsc

```
********************************************************************
* Script generated on 2020-10-21   at 11.48.32
* Script generated by user ' CallumJackso' on host 'LAPTOP-VLQKJ5UH'
* Queue manager name: qm1
* Queue manager platform: Windows
* Queue manager command level: (920/920)
* Command issued: dmpmqcfg -m qm1
********************************************************************
DEFINE AUTHINFO(USE.LDAP) +
   AUTHTYPE(IDPWLDAP) +
   CONNAME('ldap-service.ldap(389)') +
   LDAPUSER('cn=admin,dc=ibm,dc=com') +
   LDAPPWD('admin') +
   SECCOMM(NO) +
   USRFIELD('uid') +
   SHORTUSR('uid') +
   BASEDNU('ou=people,dc=ibm,dc=com') +
   AUTHORMD(SEARCHGRP) +
   BASEDNG('ou=groups,dc=ibm,dc=com') +
   GRPFIELD('cn') +
   CLASSGRP('groupOfUniqueNames') +
   FINDGRP('uniqueMember') +
   REPLACE
ALTER QMGR +
*  ALTDATE(2020-10-26) +
*  ALTTIME(11.43.11) +
   CCSID(850) +
   CERTLABL('default') +
   CLWLUSEQ(LOCAL) +
*  COMMANDQ(SYSTEM.ADMIN.COMMAND.QUEUE) +
   CONNAUTH('USE.LDAP') +
*  CRDATE(2020-10-26) +
*  CRTIME(11.43.11) +
*  QMID(qm1_2020-10-26_11.43.11) +
   SSLCRYP(' ') +
   SSLKEYR('/run/runmqserver/tls/key') +
   SUITEB(NONE) +
*  VERSION(09020000) +
   FORCE
REFRESH SECURITY
```

Operazioni successive

Sei ora pronto a selezionare l'architettura di destinazione per IBM MQ in esecuzione nei contenitori.

OpenShift CD CP4I-SC2 Selezione dell'architettura HA di destinazione per IBM MQ in esecuzione nei contenitori

Scegli tra una singola istanza (un singolo pod Kubernetes), più istanze (due pod) e la HA nativa (un pod di replica attivo e due pod di replica standby) per soddisfare i tuoi requisiti di alta disponibilità.

Prima di iniziare

Questa attività presuppone che sia stata aggiornata la configurazione del gestore code per un ambiente contenitore.

Informazioni su questa attività

IBM MQ Operator fornisce tre opzioni di alta disponibilità:

- **Istanza singola**: viene avviato un singolo contenitore (Pod) ed è responsabilità di Red Hat OpenShift riavviare in caso di errore. A causa delle caratteristiche di una serie con stato all'interno di Kubernetes, ci sono diverse situazioni in cui questo failover potrebbe richiedere un periodo di tempo prolungato o richiedere il completamento di un'azione amministrativa.
- **A più istanze**: vengono avviati due contenitori (ciascuno in un pod separato), uno in modalità attiva e un altro in standby. Questa topologia consente un failover molto più rapido. Richiede un file system Read Write Many che soddisfi i requisiti IBM MQ .
- **HA nativa**: tre contenitori (ciascuno in un pod separato), ciascuno con una istanza del gestore code. Un'istanza è il gestore code attivo, che elabora i messaggi e scrive nel log di ripristino. Ogni volta che viene scritto il log di ripristino, il gestore code attivo invia i dati alle altre due istanze, note come repliche. Se il pod che esegue il gestore code attivo ha esito negativo, una delle istanze di replica del gestore code assume il ruolo attivo e dispone dei dati correnti con cui operare.

In questa attività si sceglie solo l'architettura HA di destinazione. I passi per la configurazione dell'architettura scelta sono descritti in un'attività successiva in questo scenario (“Creazione del nuovo gestore code su Red Hat OpenShift” a pagina 121).

Procedura

1. Esamina le tre opzioni.

   Per una descrizione completa di queste opzioni, vedere “Pianificazione dell'alta disponibilità per IBM MQ nei contenitori” a pagina 17.
2. Selezionare l'architettura HA di destinazione.

   Se non sei sicuro di quale opzione scegliere, inizia con l'opzione **Singola istanza** e verifica se soddisfa i tuoi requisiti di alta disponibilità.

Operazioni successive

Si è ora pronti a creare le risorse del gestore code.

OpenShift CD CP4I-SC2 Creazione delle risorse per il gestore code

Importa la configurazione di IBM MQ e i certificati e chiavi TLS nell'ambiente Red Hat OpenShift .

Prima di iniziare

Questa attività presuppone che si disponga di l'architettura di destinazione selezionata per IBM MQ in esecuzione nei contenitori.

Informazioni su questa attività

Nelle sezioni precedenti sono state estratte, aggiornate e definite due risorse:

- IBM MQ configurazione
- Chiavi e certificati TLS

È necessario importare queste risorse nell'ambiente Red Hat OpenShift prima che il gestore code venga distribuito.

Procedura

1. Importare la configurazione IBM MQ in Red Hat OpenShift.

   Le seguenti istruzioni presuppongono che si disponga della configurazione IBM MQ nella directory corrente, in un file denominato `backup.mqsc`. Altrimenti, è necessario personalizzare il nome file in base al proprio ambiente.

   a) Accedere al cluster utilizzando `oc login`.

   b) Caricare la configurazione IBM MQ in un `configmap`.

      Esegui il seguente comando:

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

   c) Verificare che il file sia stato caricato correttamente.

      Esegui il seguente comando:

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

2. Importare le IBM MQ risorse TLS

   Come discusso in “Facoltativo: estrazione e acquisizione delle chiavi e dei certificati del gestore code” a pagina 105, TLS potrebbe essere richiesto per la distribuzione del gestore code. In tal caso, è necessario disporre già di un numero di file che terminano con `.crt` e `.key`. È necessario aggiungerli ai segreti Kubernetes per il gestore code a cui fare riferimento al momento della distribuzione.

   Ad esempio, se si dispone di una chiave e di un certificato per il gestore code, potrebbero essere richiamati:

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

   Per importare questi file, eseguire il seguente comando:

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

   Kubernetes fornisce questo utile programma di utilità quando stai importando una chiave pubblica e privata corrispondente. Se si dispone di ulteriori certificati da aggiungere, ad esempio nel truststore del gestore code, eseguire il seguente comando:

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

   Ad esempio, se i file da importare sono `trust1.crt`, `trust2.crt` e `trust3.crt`, il comando è il seguente:

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

Operazioni successive

Ora è possibile creare il nuovo gestore code su Red Hat OpenShift.

OpenShift CD CP4I-SC2 Creazione del nuovo gestore code su Red Hat OpenShift

Distribuire una singola istanza o un gestore code a più istanze su Red Hat OpenShift.

Prima di iniziare

Questa attività presuppone che l'utente abbia creato le risorse del gestore codee abbia installato IBM MQ Operator in Red Hat OpenShift.

Informazioni su questa attività

Come descritto in “Selezione dell'architettura HA di destinazione per IBM MQ in esecuzione nei contenitori” a pagina 119, ci sono tre possibili topologie di distribuzione. Pertanto, questo argomento fornisce tre diversi modelli:

- Modello 1: distribuire un gestore code a istanza singola.
- Modello 2: distribuire un gestore code a più istanze.
- Modello 3: distribuire un gestore code HA nativo.

Importante: Completare solo uno dei tre modelli, in base alla topologia preferita.

Procedura

- **Modello 1: distribuire un singolo gestore code dell'istanza.**

  Il gestore code migrato viene distribuito a Red Hat OpenShift utilizzando un file YAML. Di seguito è riportato un esempio, basato sui nomi utilizzati negli argomenti precedenti:

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: qm1
spec:
  version: 9.4.0.0-r1
  license:
    accept: true
    license: L-BMSF-5YDSLR
    use: "Production"
  pki:
    keys:
    - name: default
      secret:
        secretName: my-tls-migration
        items:
          - tls.key
          - tls.crt
  web:
    enabled: true
  queueManager:
    name: QM1
    mqsc:
      - configMap:
          name: my-mqsc-migrated
          items:
            - backup.mqsc
```

  A seconda dei passaggi che hai eseguito, potrebbe essere necessario personalizzare il precedente YAML. Per aiutarvi con questo, ecco una spiegazione di questo YAML:

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

  Definisce l'oggetto, il tipo e il nome Kubernetes . L'unico campo che richiede la personalizzazione è quello `name` .

```
 spec:
  version: 9.4.0.0-r1
  license:
    accept: true
```

```
    license: L-BMSF-5YDSLR
    use: "Production"
```

Ciò corrisponde alle informazioni sulla versione e sulla licenza per la distribuzione. Se è necessario personalizzarlo, utilizzare le informazioni fornite in “Riferimento per la licenza per mq.ibm.com/v1beta1” a pagina 137.

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

Perché il gestore code sia configurato per utilizzare TLS, deve fare riferimento ai certificati e alle chiavi pertinenti. Il campo `secretName` fa riferimento al segreto Kubernetes creato nella sezione Importa le risorse IBM MQ TLS e l'elenco di elementi (`tls.key` e `tls.crt`) sono i nomi standard assegnati da Kubernetes quando si utilizza la sintassi `oc create secret tls` . Se si dispone di ulteriori certificati da aggiungere al truststore, è possibile aggiungerli in modo simile, ma gli elementi sono i nomi file corrispondenti utilizzati durante l'importazione. Ad esempio, il seguente codice può essere utilizzato per creare i certificati del truststore:

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

```
  pki:
    trust:
    - name: default
      secret:
        secretName: my-extra-tls-migration
        items:
          - trust1.crt
          - trust2.crt
          - trust3.crt
```

Importante: Se TLS non è obbligatorio, eliminare la sezione TLS di YAML.

```
  web:
    enabled: true
```

Ciò abilita la console Web per la distribuzione

```
  queueManager:
    name: QM1
```

Definisce il nome del gestore code come QM1. Il gestore code viene personalizzato in base ai requisiti dell'utente, ad esempio il nome del gestore code originale.

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

Il codice precedente estrae la configurazione del gestore code importata nella sezione Importa la configurazione IBM MQ . Se sono stati utilizzati nomi diversi, è necessario modificare `my-mqsc-migrated` e `backup.mqsc`.

Si noti che lo YAML di esempio presuppone che la classe di memoria predefinita per l'ambiente Red Hat OpenShift sia definita come una classe di memoria RWX o RWO. Se non è definito un valore predefinito nel proprio ambiente, è necessario specificare la classe di memoria da utilizzare. È possibile eseguire questa operazione estendendo YAML nel modo seguente:

```
  queueManager:
    name: QM1
```

```
    storage:
      defaultClass: my_storage_class
      queueManager:
        type: persistent-claim
```

Aggiungere il testo evidenziato, con l'attributo della classe personalizzato per corrispondere al proprio ambiente. Per rilevare i nomi delle classi di memoria nell'ambiente, eseguire il seguente comando:

```
oc get storageclass
```

Di seguito viene riportato un output di esempio restituito da questo comando:

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

Il seguente codice mostra come fare riferimento alla configurazione IBM MQ importata nella sezione Importa la configurazione IBM MQ . Se sono stati utilizzati nomi diversi, è necessario modificare `my-mqsc-migrated` e `backup.mqsc`.

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

Il gestore code a istanza singola è stato distribuito. Questo completa il modello. Sei ora pronto a verificare la distribuzione del nuovo contenitore.

- **Modello 2: distribuzione di un gestore code a più istanze.**

  Il gestore code migrato viene distribuito a Red Hat OpenShift utilizzando un file YAML. Il seguente esempio è basato sui nomi utilizzati nelle precedenti sezioni.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: qm1mi
spec:
  version: 9.4.0.0-r1
  license:
    accept: true
    license: L-BMSF-5YDSLR
    use: "Production"
  pki:
    keys:
    - name: default
      secret:
        secretName: my-tls-migration
        items:
          - tls.key
          - tls.crt
  web:
    enabled: true
  queueManager:
    name: QM1
    availability: MultiInstance
    storage:
      defaultClass: aws-efs
      persistedData:
        enabled: true
      queueManager:
        enabled: true
      recoveryLogs:
        enabled: true
    mqsc:
      - configMap:
          name: my-mqsc-migrated
          items:
            - backup.mqsc
```

Ecco una spiegazione di questo YAML. La maggior parte della configurazione segue lo stesso approccio di distribuzione di un singolo gestore code dell'istanza, pertanto vengono illustrati solo gli aspetti relativi alla disponibilità e alla memoria del gestore code.

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

Specifica il nome del gestore code come `QM1`e imposta la distribuzione su `MultiInstance` invece che sulla singola istanza predefinita.

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

Un gestore code a più istanze IBM MQ dipende dalla memoria RWX. Per impostazione predefinita, un gestore code viene distribuito in modalità a istanza singola e, pertanto, sono richieste ulteriori opzioni di archiviazione quando si passa alla modalità a più istanze. Nel precedente esempio YAML, vengono definiti tre volumi di archiviazione persistenti e una classe di volume persistente. Questa classe di volume persistente deve essere una classe di archiviazione RWX. Se non si è sicuri dei nomi delle classi di memoria nel proprio ambiente, è possibile eseguire il seguente comando per rilevarli:

```
oc get storageclass
```

Di seguito viene riportato un output di esempio restituito da questo comando:

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

Il seguente codice mostra come fare riferimento alla configurazione IBM MQ importata nella sezione Importa la configurazione IBM MQ . Se sono stati utilizzati nomi diversi, è necessario modificare `my-mqsc-migrated` e `backup.mqsc`.

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

È stato distribuito il gestore code a più istanze. Questo completa il modello. Sei ora pronto a verificare la distribuzione del nuovo contenitore.

- **Modello 3: distribuire un gestore code HA nativo.**

  Per un esempio di creazione di un gestore code HA nativo, consultare “Esempio: configurazione della HA nativa utilizzando IBM MQ Operator” a pagina 74.

OpenShift CD CP4I-SC2 Verifica della nuova distribuzione del contenitore

Ora che IBM MQ è distribuito su Red Hat OpenShift, puoi verificare l'ambiente utilizzando gli esempi IBM MQ .

Prima di iniziare

Questa attività presuppone che sia stato creato il nuovo gestore code su Red Hat OpenShift.

Importante: Questa attività presuppone che TLS non sia abilitato nel gestore code.

Informazioni su questa attività

In questa attività si eseguono gli esempi IBM MQ dall'interno del contenitore del gestore code migrato. Tuttavia, è possibile utilizzare le proprie applicazioni in esecuzione da un altro ambiente.

Hai bisogno delle seguenti informazioni:

- Nome utente LDAP
- Password LDAP
- IBM MQ Nome canale
- Nome coda

Questo codice di esempio usa le seguenti impostazioni. Si prega di notare che le impostazioni saranno diverse.

- Nome utente LDAP: mqapp
- Password LDAP: mqapp
- IBM MQ Nome canale: DEV.APP.SVRCONN
- Nome coda: Q1

Procedura

1. Esegui nel contenitore IBM MQ in esecuzione.

   Utilizzare il seguente comando:

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

   dove `qm1-ibm-mq-0` è il pod che abbiamo distribuito in “Creazione del nuovo gestore code su Red Hat OpenShift” a pagina 121. Se la distribuzione è stata richiamata in modo diverso, personalizzare questo valore.
2. Inviare un messaggio.

   Eseguire i seguenti comandi:

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

   Viene richiesta una password, quindi è possibile inviare un messaggio.
3. Verificare che il messaggio sia stato ricevuto correttamente.

   Eseguire l'esempio GET:

   ```
   ./amqsgetc Q1 QM1
   ```

Risultati

È stato completato il “Migrazione a IBM MQ Operator” a pagina 103.

Operazioni successive

Utilizzare le seguenti informazioni come supporto per scenari di migrazione più complessi:

Migrazione dei messaggi accodati

Per migrare i messaggi in coda esistenti, seguire le istruzioni riportate nel seguente argomento per l'esportazione e l'importazione dei messaggi dopo che il nuovo gestore code è stato creato: Utilizzo del programma di utilità dmpmqmsg tra due sistemi.

Connessione a IBM MQ dall'esterno dell'ambiente Red Hat OpenShift

Il gestore code distribuito può essere esposto ai client IBM MQ e ai gestori code all'esterno dell'ambiente Red Hat OpenShift . Il processo dipende dalla versione di IBM MQ che si collega all'ambiente Red Hat OpenShift . Vedere “Configurazione di un instradamento per la connessione a un gestore code dall'esterno di un cluster Red Hat OpenShift” a pagina 83.

Funzionamento di IBM MQ nei contenitori

Se hai bisogno di operare o interagire con i gestori code IBM MQ in esecuzione nei contenitori, consulta i seguenti argomenti per ulteriori informazioni.

Procedura

- “Utilizzo di IBM MQ mediante IBM MQ Operator” a pagina 127.
- “Visualizzazione dello stato dei gestori code della HA nativa” a pagina 134.
- “Arresto manuale delle istanze del gestore code della HA nativa” a pagina 136.

OpenShift CP4I Utilizzo di IBM MQ mediante IBM MQ Operator

Procedura

- “Connessione al IBM MQ Console distribuito in un cluster Red Hat OpenShift” a pagina 127.
- “Monitoraggio quando si utilizza IBM MQ Operator” a pagina 128.
- “Backup e ripristino della configurazione del gestore code utilizzando la CLI Red Hat OpenShift” a pagina 133.

OpenShift CP4I Connessione al IBM MQ Console distribuito in un cluster Red Hat OpenShift

Come connettersi al IBM MQ Console di un gestore code distribuito su un cluster Red Hat OpenShift Container Platform .

Informazioni su questa attività

L'URL IBM MQ Console è disponibile nella pagina dei dettagli `QueueManager` nella console Web Red Hat OpenShift o in IBM Cloud Pak for Integration Platform UI. In alternativa, è possibile trovarlo dalla CLI Red Hat OpenShift immettendo il seguente comando:

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

Se stai utilizzando una licenza IBM Cloud Pak for Integration , IBM MQ Console utilizza Keycloak per la gestione dell'identità e dell'accesso. Vedi Identity and Access management nella documentazione di IBM Cloud Pak for Integration .

Se si utilizza una licenza IBM MQ , il IBM MQ Console non è preconfigurato ed è necessario configurarlo da soli. Per ulteriori informazioni, vedi Configurazione di utenti e ruoli. Per un esempio, consultare “Configurazione di IBM MQ Console con un registro di base utilizzando IBM MQ Operator” a pagina 96.

Attività correlate

“Configurazione di un instradamento per la connessione a un gestore code dall'esterno di un cluster Red Hat OpenShift” a pagina 83

Hai bisogno di un instradamento Red Hat OpenShift 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 è disponibile solo nel protocollo TLS quando viene utilizzato un protocollo TLS 1.2 o superiore. Red Hat OpenShift Container Platform Router utilizza SNI per instradare le richieste al gestore code IBM MQ .

Concessione di autorizzazioni per IBM MQ Console

Le autorizzazioni per IBM MQ Console sono gestite in modo diverso in base al tuo uso della licenza.

Informazioni su questa attività

- Se stai utilizzando una licenza IBM Cloud Pak for Integration , IBM MQ Console utilizza Keycloak per la gestione dell'identità e dell'accesso.
  - Vedi Identity and Access management nella documentazione di IBM Cloud Pak for Integration .
  - Se hai precedentemente configurato gli utenti con IAM su versioni precedenti di IBM MQ Operator, vedi Migrazione degli utenti da IAM a Keycloak.
- Se si utilizza una licenza IBM MQ , il IBM MQ Console non è preconfigurato ed è necessario configurarlo da soli.
  - Per ulteriori informazioni su utenti e ruoli, consultare Configurazione di utenti e ruoli.
  - Per un semplice esempio, consultare “Configurazione di IBM MQ Console con un registro di base utilizzando IBM MQ Operator” a pagina 96.
  - In alternativa, è possibile installare l'operatore IBM Cloud Pak for Integration per configurare Keycloak come descritto in precedenza.

Monitoraggio quando si utilizza IBM MQ Operator

I gestori code gestiti da IBM MQ Operator possono produrre metriche compatibili con Prometheus.

Puoi visualizzare queste metriche utilizzando lo stack di monitoraggioRed Hat OpenShift Container Platform (OCP). Aprire la scheda **Metriche** in OCP, quindi fare clic su **Observe** > **Metriche**. Le metriche del gestore code sono abilitate per impostazione predefinita, ma possono essere disabilitate impostando **`.spec.metrics.enabled`** su `false`.

Prometheus è un database di serie temporali e un motore di valutazione delle regole per metriche. I contenitori IBM MQ espongono un endpoint di metrica che può essere interrogato da Prometheus. Le metriche vengono generate dagli argomenti di sistema MQ per il monitoraggio e la traccia dell'attività.

OpenShift Container Platform include uno stack di monitoraggio preconfigurato, preinstallato e con aggiornamento automatico che utilizza un server Prometheus . Lo stack di controllo OpenShift Container Platform deve essere configurato per monitorare i progetti definiti dall'utente. Per ulteriori informazioni, consultare Abilitazione del monitoraggio per i progetti definiti dall'utente. IBM MQ Operator crea un `ServiceMonitor` quando crei un `QueueManager` con le metriche abilitate, che l'operatore Prometheus può quindi rilevare.

Metriche pubblicate quando si utilizza IBM MQ Operator

I contenitori dei gestori code possono pubblicare metriche compatibili con Red Hat OpenShift Monitoring.

Metrica	Tipo	Descrizione
`ibmmq_qmgr_commit_total`	`counter`	Conteggio commit
`ibmmq_qmgr_cpu_load_fifteen_minute_average_percentage`	`gauge`	Carico CPU - media di quindici minuti
`ibmmq_qmgr_cpu_load_five_minute_average_percentage`	`gauge`	Carico CPU - media di cinque minuti
`ibmmq_qmgr_cpu_load_one_minute_average_percentage`	`gauge`	Carico CPU - media di un minuto

Metrica	Tipo	Descrizione
ibmmq_qmgr_destructive_get_bytes_total	counter	Totale estrazioni distruttive dell'intervallo - conteggio byte
ibmmq_qmgr_destructive_get_total	counter	Totale estrazioni distruttive dell'intervallo - conteggio
ibmmq_qmgr_durable_subscription_alter_total	counter	Conteggio modifiche sottoscrizioni durature
ibmmq_qmgr_durable_subscription_create_total	counter	Conteggio creazioni sottoscrizioni durature
ibmmq_qmgr_durable_subscription_delete_total	counter	Conteggio eliminazione sottoscrizioni durature
ibmmq_qmgr_durable_subscription_resume_total	counter	Conteggio ripristini sottoscrizioni durature
ibmmq_qmgr_errors_file_system_free_space_percentage	gauge	File system errori MQ - spazio disponibile
ibmmq_qmgr_errors_file_system_in_use_bytes	gauge	File system errori MQ - byte in uso
ibmmq_qmgr_expired_message_total	counter	Conteggio messaggi scaduti
ibmmq_qmgr_failed_browse_total	counter	Conteggio esplorazioni non riuscite
ibmmq_qmgr_failed_mqcb_total	counter	Conteggio MQCB non riusciti
ibmmq_qmgr_failed_mqclose_total	counter	Conteggio MQCLOSE non riusciti
ibmmq_qmgr_failed_mqconn_mqconnx_total	counter	Conteggio MQCONN/MQCONNX non riusciti
ibmmq_qmgr_failed_mqget_total	counter	MQGET non riusciti - conteggio
ibmmq_qmgr_failed_mqinq_total	counter	Conteggio MQINQ non riusciti
ibmmq_qmgr_failed_mqopen_total	counter	Conteggio MQOPEN non riusciti
ibmmq_qmgr_failed_mqput1_total	counter	Conteggio MQPUT1 non riusciti
ibmmq_qmgr_failed_mqput_total	counter	Conteggio MQPUT non riusciti

Metrica	Tipo	Descrizione
ibmmq_qmgr_failed_mqset_total	counter	Conteggio MQSET non riusciti
ibmmq_qmgr_failed_mqsubrq_total	counter	Conteggio MQSUBRQ non riusciti
ibmmq_qmgr_failed_subscription_create_alter_resume_total	counter	Conteggio creazioni/modifiche/ripristini sottoscrizioni non riusciti
ibmmq_qmgr_failed_subscription_delete_total	counter	Conteggio errore di eliminazione sottoscrizione
ibmmq_qmgr_failed_topic_mqput_mqput1_total	counter	Conteggio MQPUT/MQPUT1 di argomento non riusciti
ibmmq_qmgr_fdc_files	gauge	Conteggio file FDC MQ
ibmmq_qmgr_log_file_system_in_use_bytes	gauge	File system log - byte in uso
ibmmq_qmgr_log_file_system_max_bytes	gauge	File system log - massimo di byte
ibmmq_qmgr_log_in_use_bytes	gauge	Log - byte in uso
ibmmq_qmgr_log_logical_written_bytes_total	counter	Log - byte logici scritti
ibmmq_qmgr_log_max_bytes	gauge	Log - massimo di byte
ibmmq_qmgr_log_occupied_by_reusable_extents_bytes	gauge	Log - byte occupati dalle estensioni riutilizzabili
ibmmq_qmgr_log_physical_written_bytes_total	counter	Log - byte fisici scritti
ibmmq_qmgr_log_primary_space_in_use_percentage	gauge	Log - spazio primario corrente in uso
ibmmq_qmgr_log_required_for_media_recovery_bytes	gauge	Log - byte necessari per il ripristino supporti
ibmmq_qmgr_log_workload_primary_space_utilization_percentage	gauge	Log - utilizzo spazio primario carico di lavoro
ibmmq_qmgr_log_write_latency_seconds	gauge	Log - latenza scrittura

Metrica	Tipo	Descrizione
ibmmq_qmgr_log_write_size_bytes	gauge	Log - dimensione scrittura
ibmmq_qmgr_mqcb_total	counter	Conteggio MQCB
ibmmq_qmgr_mqclose_total	counter	Conteggio MQCLOSE
ibmmq_qmgr_mqconn_mqconnx_total	counter	Conteggio MQCONN/MQCONNX
ibmmq_qmgr_mqctl_total	counter	Conteggio MQCTL
ibmmq_qmgr_mqdisc_total	counter	Conteggio MQDISC
ibmmq_qmgr_mqinq_total	counter	Conteggio MQINQ
ibmmq_qmgr_mqopen_total	counter	Conteggio MQOPEN
ibmmq_qmgr_mqput_mqput1_bytes_total	counter	Conteggio byte totale MQPUT/MQPUT1 dell'intervallo
ibmmq_qmgr_mqput_mqput1_total	counter	Conteggio totale MQPUT/MQPUT1 dell'intervallo
ibmmq_qmgr_mqset_total	counter	Conteggio MQSET
ibmmq_qmgr_mqstat_total	counter	Conteggio MQSTAT
ibmmq_qmgr_mqsubrq_total	counter	Conteggio MQSUBRQ
ibmmq_qmgr_non_durable_subscription_create_total	counter	Conteggio creazioni sottoscrizioni non durature
ibmmq_qmgr_non_durable_subscription_delete_total	counter	Conteggio eliminazione sottoscrizioni non durature
ibmmq_qmgr_non_persistent_message_browse_bytes_total	counter	Esplorazione messaggi non permanenti - conteggio byte
ibmmq_qmgr_non_persistent_message_browse_total	counter	Esplorazione messaggi non permanenti - conteggio
ibmmq_qmgr_non_persistent_message_destructive_get_total	counter	Estrazione distruttiva di messaggi non permanenti - conteggio
ibmmq_qmgr_non_persistent_message_get_bytes_total	counter	Ottenimento messaggi non permanenti - conteggio byte

Metrica	Tipo	Descrizione
ibmmq_qmgr_non_persistent_message_mqput1_total	counter	Conteggio MQPUT1 messaggi non permanenti
ibmmq_qmgr_non_persistent_message_mqput_total	counter	Conteggio MQPUT messaggi non permanenti
ibmmq_qmgr_non_persistent_message_put_bytes_total	counter	Inserimento messaggi non permanenti - conteggio byte
ibmmq_qmgr_non_persistent_topic_mqput_mqput1_total	counter	Non permanente - conteggio MQPUT/MQPUT1 di argomento
ibmmq_qmgr_persistent_message_browse_bytes_total	counter	Esplorazione messaggi permanenti - conteggio byte
ibmmq_qmgr_persistent_message_browse_total	counter	Esplorazione messaggi permanenti - conteggio
ibmmq_qmgr_persistent_message_destructive_get_total	counter	Estrazione distruttiva di messaggi permanenti - conteggio
ibmmq_qmgr_persistent_message_get_bytes_total	counter	Ottenimento messaggi permanenti - conteggio byte
ibmmq_qmgr_persistent_message_mqput1_total	counter	Conteggio MQPUT1 messaggi permanenti
ibmmq_qmgr_persistent_message_mqput_total	counter	Conteggio MQPUT messaggi permanenti
ibmmq_qmgr_persistent_message_put_bytes_total	counter	Inserimento messaggi permanenti - conteggio byte
ibmmq_qmgr_persistent_topic_mqput_mqput1_total	counter	Permanente - conteggio MQPUT/MQPUT1 di argomento
ibmmq_qmgr_published_to_subscribers_bytes_total	counter	Pubblicato per i sottoscrittori - conteggio byte
ibmmq_qmgr_published_to_subscribers_message_total	counter	Pubblicato per i sottoscrittori - conteggio messaggi
ibmmq_qmgr_purged_queue_total	counter	Conteggio code eliminate

Metrica	Tipo	Descrizione
ibmmq_qmgr_queue_manager_file_system_free_space_percentage	gauge	File system Gestore code - spazio disponibile
ibmmq_qmgr_queue_manager_file_system_in_use_bytes	gauge	File system Gestore code - byte in uso
ibmmq_qmgr_ram_free_percentage	gauge	Percentuale RAM disponibile
ibmmq_qmgr_ram_usage_estimate_for_queue_manager_bytes	gauge	Totale byte della RAM - stima per il gestore code
ibmmq_qmgr_rollback_total	counter	Conteggio rollback
ibmmq_qmgr_system_cpu_time_estimate_for_queue_manager_percentage	gauge	Tempo CPU di sistema - stima percentuale per il gestore code
ibmmq_qmgr_system_cpu_time_percentage	gauge	Percentuale di tempo CPU di sistema
ibmmq_qmgr_topic_mqput_mqput1_total	counter	Totale dell'intervallo di MQPUT/MQPUT1 di argomenti
ibmmq_qmgr_topic_put_bytes_total	counter	Inserimento totale byte di argomento dell'intervallo
ibmmq_qmgr_trace_file_system_free_space_percentage	gauge	File system traccia MQ - spazio disponibile
ibmmq_qmgr_trace_file_system_in_use_bytes	gauge	File system traccia MQ - byte in uso
ibmmq_qmgr_user_cpu_time_estimate_for_queue_manager_percentage	gauge	Tempo CPU utente - stima percentuale per il gestore code
ibmmq_qmgr_user_cpu_time_percentage	gauge	Percentuale di tempo CPU utente

Informazioni correlate

Metriche pubblicate negli argomenti di sistema

OpenShift CP4I Backup e ripristino della configurazione del gestore code utilizzando la CLI Red Hat 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

Accedere al cluster utilizzando **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 .

  a) Ottenere il nome del pod per il gestore code.
  Ad esempio, 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
  ```

  b) 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.

  a) Ottenere il nome del pod per il gestore code.
  Ad esempio, 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
  ```

  b) Eseguire il comando **runmqsc** sul pod, indirizzando il contenuto del file backup.mqsc .

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

MQ Adv. Visualizzazione dello stato dei gestori code della HA nativa

Per i contenitori personalizzati, puoi visualizzare lo stato delle istanze Native HA utilizzando il comando **dspmq** .

Informazioni su questa attività

È possibile utilizzare il comando **dspmq** per visualizzare lo stato operativo di un'istanza del gestore code su un nodo. Le informazioni restituite dipendono dal fatto che l'istanza sia attiva o una replica. Le informazioni fornite dall'istanza attiva sono definitive, le informazioni dai nodi di replica potrebbero non essere aggiornate.

È possibile effettuare le seguenti azioni:

- Visualizzare se l'istanza del gestore code sul nodo corrente è attiva o una replica.
- Visualizza lo stato operativo della HA nativa dell'istanza sul nodo corrente.
- Visualizzare lo stato operativo di tutte e tre le istanze in una configurazione HA nativa.

I seguenti campi di stato vengono utilizzati per riportare lo stato di configurazione della HA nativa:

RUOLO
Specifica il ruolo corrente dell'istanza ed è uno tra Active, Replicao Unknown.

ISTANZA
Il nome fornito per questa istanza del gestore code quando è stata creata utilizzando l'opzione **-lr** del comando **crtmqm** .

INSYNC
Indica se l'istanza è in grado di assumere il controllo come istanza attiva, se richiesto.

quorum
Riporta lo stato del quorum nel formato *number_of_instances_in - sync/ number_of_instances_configured*.

REPLADDR
L'indirizzo di replica dell'istanza del gestore code.

COLLEGA
Indica se il nodo è connesso all'istanza attiva.

BACKLOG
Indica il numero di KB in cui si trova l'istanza.

CONNETTIN
Indica se l'istanza denominata è connessa a questa istanza.

ALTDATE
Indica la data in cui queste informazioni sono state aggiornate l'ultima volta (vuoto se non sono mai state aggiornate).

ALTTIME
Indica l'ora in cui queste informazioni sono state aggiornate l'ultima volta (vuoto se non sono mai state aggiornate).

Procedura

- Per determinare se un'istanza del gestore code è in esecuzione come istanza attiva o come replica:

  ```
  dspmq -o status -m QMgrName
  ```

  Un'istanza attiva di un gestore code denominato BOB riporta il seguente stato:

  ```
  QMNAME(BOB)                STATUS(Running)
  ```

  Un'istanza di replica di un gestore code denominato BOB riporta il seguente stato:

  ```
  QMNAME(BOB)                STATUS(Replica)
  ```

  Un'istanza inattiva riporta il seguente stato:

  ```
  QMNAME(BOB)                STATUS(Ended Immediately)
  ```

- Per determinare lo stato operativo della HA nativa dell'istanza sul nodo corrente:

  ```
  dspmq -o nativeha -m QMgrName
  ```

  L'istanza attiva di un gestore code denominato BOB potrebbe riportare il seguente stato:

  ```
  QMNAME(BOB)              ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
  ```

  Un'istanza di replica di un gestore code denominato BOB potrebbe riportare il seguente stato:

  ```
  QMNAME(BOB)              ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
  ```

  Un'istanza inattiva di un gestore code denominato BOB potrebbe riportare il seguente stato:

  ```
  QMNAME(BOB)              ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
  ```

- Per determinare lo stato operativo della HA nativo di tutte le istanze nella configurazione della HA nativa:

```
dspmq -o nativeha -x -m QMgrName
```

Se si immette questo comando sul nodo che esegue l'istanza attiva del BOB del gestore code, è possibile che si riceva il seguente stato:

```
QMNAME(BOB)            ROLE(Active) INSTANCE(inst1) INSYNC(Yes) QUORUM(3/3)
 INSTANCE(inst1) ROLE(Active)  REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
 INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
 INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
```

Se si immette questo comando su un nodo che esegue un'istanza di replica del BOB del gestore code, è possibile che si riceva il seguente stato, che indica che una delle repliche è in ritardo:

```
QMNAME(BOB)            ROLE(Replica) INSTANCE(inst2) INSYNC(Yes) QUORUM(2/3)
 INSTANCE(inst2) ROLE(Replica) REPLADDR(9.20.123.46) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
 INSTANCE(inst1) ROLE(Active)  REPLADDR(9.20.123.45) CONNACTV(Yes) INSYNC(Yes) BACKLOG(0)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
 INSTANCE(inst3) ROLE(Replica) REPLADDR(9.20.123.47) CONNACTV(Yes) INSYNC(No)  BACKLOG(435)
CONNINST(Yes) ALTDATE(2022-01-12) ALTTIME(12.03.44)
```

Se si immette questo comando su un nodo che esegue un'istanza inattiva del BOB del gestore code, è possibile che si riceva il seguente stato:

```
QMNAME(BOB)            ROLE(Unknown) INSTANCE(inst3) INSYNC(no) QUORUM(0/3)
 INSTANCE(inst1) ROLE(Unknown) REPLADDR(9.20.123.45) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATE() ALTTIME()
 INSTANCE(inst2) ROLE(Unknown) REPLADDR(9.20.123.46) CONNACTV(Unknown) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATE() ALTTIME()
 INSTANCE(inst3) ROLE(Unknown) REPLADDR(9.20.123.47) CONNACTV(No) INSYNC(Unknown)
BACKLOG(Unknown) CONNINST(No) ALTDATE() ALTTIME()
```

Se si immette il comando quando le istanze stanno ancora negoziando quali sono attive e quali sono repliche, si riceverà il seguente stato:

```
QMNAME(BOB)             STATUS(Negotiating)
```

Riferimenti correlati

comando dspmq (visualizza gestori code)

MQ Adv. Arresto manuale delle istanze del gestore code della HA nativa

È possibile utilizzare il comando **endmqm** per terminare un gestore code attivo o di replica che fa parte di un gruppo HA nativo.

Procedura

- Per terminare l''istanza attiva di un gestore code, fare riferimento a Fine dei gestori code della HA nativa nella sezione Configurazione di questa documentazione.

OpenShift CP4I Informazioni di riferimento per IBM MQ nei contenitori

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

OpenShift CP4I Riferimento API per IBM MQ Operator

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

Riferimento API per mq.ibm.com/v1beta1

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

Riferimento per la licenza per mq.ibm.com/v1beta1

Versioni di licenza correnti

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

Valore di `spec.license.license`	Valore di `spec.license.use`	Informazioni sulla licenza	Versioni IBM MQ applicabili
`L-JTPV-KYG8TF`	`Production` o `NonProduction`	IBM Cloud Pak for Integration 16.1.0	9.4.0
`L-BMSF-5YDSLR`	`Production` o `NonProduction`	IBM Cloud Pak for Integration Edizione limitata 16.1.0	9.4.0
`L-EHXT-MQCRN9`	`Production`	IBM MQ Advanced 9.4	9.4.0
`L-CLXQ-ADXTK3`	`Development`	IBM MQ Advanced for Developers (Non Warranted) 9.4	9.4.0

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

Versioni di licenza precedenti

Consultare Vere versioni di licenza nella documentazione di IBM MQ 9.3 .

Riferimento API per QueueManager (mq.ibm.com/v1beta1)

QueueManager

Un QueueManager è un IBM MQ che fornisce servizi di accodamento e pubblicazione / sottoscrizione alle applicazioni. Documentazione di IBM MQ : https://ibm.biz/BdPZqj. Riferimento licenza: https://ibm.biz/BdPZfq..

Campo	Descrizione
`apiVersion` 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.
`kind` 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.
`metadata`	
`spec` QueueManagerSpec	Lo stato desiderato del QueueManager.

Campo	Descrizione
`status` QueueManagergestore code	Lo stato osservato di QueueManager.

.spec

Lo stato desiderato del QueueManager.

Viene visualizzato in:

- "QueueManager" a pagina 137

Campo	Descrizione
`affinity`	Regole di affinità Kubernetes standard. Per ulteriori informazioni, consultare https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.17/#affinity-v1-core.
`annotations` Annotazioni	Il campo delle annotazioni funge da pass-through per le annotazioni Pod. Gli utenti possono aggiungere qualsiasi annotazione a questo campo e applicarlo al pod. Le annotazioni qui sovrascrivono le annotazioni predefinite, se fornite. Richiede l'operatore MQ 1.3.0 o superiore.
Array `imagePullSecrets` LocalObjectReference	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.
`labels` Etichette	Il campo Etichette funge da pass-through per le etichette Pod. Gli utenti possono aggiungere qualsiasi etichetta a questo campo e applicarlo al pod. Le etichette qui sovrascrivono le etichette predefinite, se fornite. Richiede l'operatore MQ 1.3.0 o superiore.
`license` Licenza	Impostazioni che controllano l'accettazione della licenza e quali metriche di licenza utilizzare.
`pki` 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 .
`queueManager` Configurazione QueueManager	Impostazioni per il contenitore Gestore code e il gestore code sottostante.
`securityContext` SecurityContext	Impostazioni di protezione da aggiungere al securityContextdi Queue Manager Pod.
`telemetry` Telemetria	Impostazioni per la configurazione di Open Telemetry. Richiede l'operatore MQ 2.2.0 o superiore.
`template` 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.

Campo	Descrizione
`terminationGracePeriodSeconds` 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. Il valore predefinito è 30 secondi.
`tracing` TracingConfig	Impostazioni per l'integrazione di traccia con il dashboard Operazioni Cloud Pak for Integration .
`version` Stringa	Impostazione che controlla la versione di MQ che verrà utilizzata (obbligatorio). Ad esempio: `9.1.5.0-r2` specifica MQ versione 9.1.5.0, utilizzando la seconda revisione dell'immagine contenitore. Le correzioni specifiche del contenitore vengono spesso applicate nelle revisioni, come le correzioni all'immagine di base.
`web` WebServerConfigurazione	Impostazioni per il server Web MQ .

.spec.annotations

Il campo delle annotazioni funge da pass-through per le annotazioni Pod. Gli utenti possono aggiungere qualsiasi annotazione a questo campo e applicarlo al pod. Le annotazioni qui sovrascrivono le annotazioni predefinite, se fornite. Richiede l'operatore MQ 1.3.0 o superiore.

Viene visualizzato in:

- ".spec" a pagina 138

.spec.imagePullSecrets

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

Viene visualizzato in:

- ".spec" a pagina 138

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?.

.spec.labels

Il campo Etichette funge da pass-through per le etichette Pod. Gli utenti possono aggiungere qualsiasi etichetta a questo campo e applicarlo al pod. Le etichette qui sovrascrivono le etichette predefinite, se fornite. Richiede l'operatore MQ 1.3.0 o superiore.

Viene visualizzato in:

- ".spec" a pagina 138

.spec.license

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

Viene visualizzato in:

- ".spec" a pagina 138

Campo	Descrizione
`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/BdPZfq per i valori validi.
`metric` Stringa	Impostazione che specifica quale metrica di licenza utilizzare. Ad esempio, `ProcessorValueUnit`, `VirtualProcessorCore` o `ManagedVirtualServer`. Il valore predefinito è `ProcessorValueUnit` quando si usa una licenza di MQ e `VirtualProcessorCore` quando si utilizza una licenza di Cloud Pak for Integration .
`use` Stringa	Impostazione che controlla il modo in cui verrà utilizzato il software, dove la licenza supporta più utilizzi. Consultare https://ibm.biz/BdPZfq per i valori validi.

.spec.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:

- ".spec" a pagina 138

Campo	Descrizione
Array `keys` PKISource	Chiavi private da aggiungere al repository delle chiavi del gestore code.
Array `trust` PKISource	Certificati da aggiungere al repository delle chiavi del gestore code.

.spec.pki.keys

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

Viene visualizzato in:

- ".spec.pki" a pagina 140

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

.spec.pki.keys.secret

Fornisci una chiave utilizzando un segreto Kubernetes .

Viene visualizzato in:

- ".spec.pki.keys" a pagina 140

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 .

.spec.pki.trust

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

Viene visualizzato in:

- ".spec.pki" a pagina 140

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

.spec.pki.trust.secret

Fornisci una chiave utilizzando un segreto Kubernetes .

Viene visualizzato in:

- ".spec.pki.trust" a pagina 141

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 .

.spec.queueManager

Impostazioni per il contenitore Gestore code e il gestore code sottostante.

Viene visualizzato in:

- ".spec" a pagina 138

Campo	Descrizione
availability Disponibilità	Impostazioni di disponibilità per il gestore code, ad esempio se utilizzare o meno una coppia active - standby o un'alta disponibilità nativa.
debug booleano	Indica se registrare o meno i messaggi di debug dal codice specifico del contenitore al log del contenitore. Il valore predefinito è false.
image Stringa	L'immagine contenitore che verrà utilizzata.
imagePullPolicy Stringa	Impostazione che controlla quando il kubelet tenta di estrarre l'immagine specificata. Il valore predefinito è IfNotPresent.
Array ini INISource	Impostazioni per fornire INI per il gestore code. Richiede l'operatore MQ 1.1.0 o superiore.
livenessProbe QueueManagerLivenessProbe	Impostazioni che controllano il probe di attività.
logFormat Stringa	Quale formato di log utilizzare per questo contenitore. Utilizza JSON per i log formattati JSON dal contenitore. Utilizzare Basic per i messaggi in formato testo. Il valore predefinito è Basic.
metrics QueueManagerMetriche	Impostazioni per le metriche in stile Prometheus.
Array mqsc MQSCSource	Impostazioni per fornire MQSC per il gestore code. Richiede l'operatore MQ 1.1.0 o superiore.

Campo	Descrizione
name Stringa	Nome del gestore code MQ sottostante, se diverso da metadata.name. Utilizzare questo campo se si desidera un nome del gestore code che non sia conforme alle regole Kubernetes per i nomi (ad esempio, un nome che include lettere maiuscole).
readinessProbe QueueManagerReadinessProbe	Impostazioni che controllano il probe di disponibilità.
recoveryLogs RecoveryLogs	Impostazioni per i log di ripristino di MQ . Richiede l'operatore MQ 2.4.0 o superiore.
resources Risorse	Impostazioni che controllano i requisiti delle risorse.
route Instrada	Impostazioni per l'instradamento del gestore code. Richiede l'operatore MQ 1.4.0 o superiore.
startupProbe StartupProbe	Impostazioni che controllano il probe di avvio. Si applica solo alle distribuzioni MultiInstance e NativeHA . Richiede l'operatore MQ 1.5.0 o superiore.
storage QueueManagerStorage	Impostazioni di archiviazione per controllare l'utilizzo da parte del gestore code di volumi persistenti e classi di archiviazione.

.spec.queueManager.availability

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

Viene visualizzato in:

- ".spec.queueManager" a pagina 141

Campo	Descrizione
tls TL	Impostazioni TLS facoltative per la configurazione della comunicazione protetta tra le repliche NativeHA . Richiede l'operatore MQ 1.5.0 o superiore.
type Stringa	Il tipo di disponibilità da utilizzare. Utilizza SingleInstance per un singolo pod, che verrà riavviato automaticamente (in alcuni casi) da Kubernetes. Utilizzare MultiInstance per una coppia di pod, uno dei quali è il gestore code active e l'altro è uno standby. Utilizzare NativeHA per la replica nativa ad alta disponibilità (richiede MQ Operator 1.5.0 o superiore). Il valore predefinito è SingleInstance. Per ulteriori dettagli, consultare http://ibm.biz/BdqAQa .
updateStrategy Stringa	La strategia di aggiornamento da utilizzare per i gestori code MultiInstance e NativeHA . Utilizzare RollingUpdate per abilitare gli aggiornamenti a rotazione automatica ogni volta che viene modificata la configurazione del gestore code. Utilizzare OnDelete per disabilitare gli aggiornamenti a rotazione automatica, le modifiche del gestore code verranno applicate solo quando i pod vengono eliminati (incluse le eliminazioni di pod attivate da fattori esterni). Il valore predefinito è RollingUpdate. Richiede l'operatore MQ 1.6.0 o superiore.

.spec.queueManager.availability.tls

Impostazioni TLS facoltative per la configurazione della comunicazione protetta tra le repliche NativeHA . Richiede l'operatore MQ 1.5.0 o superiore.

Viene visualizzato in:

- “.spec.queueManager.availability” a pagina 142

Campo	Descrizione
`cipherSpec` Stringa	Il nome di CipherSpec per TLS NativeHA .
`secretName` Stringa	Il nome del segreto Kubernetes .

.spec.queueManager.ini

Origine dei file di configurazione INI.

Viene visualizzato in:

- “.spec.queueManager” a pagina 141

Campo	Descrizione
`configMap` ConfigMapINISource	ConfigMap rappresenta una ConfigMap Kubernetes che contiene informazioni INI.
`secret` SecretINISource	Il segreto rappresenta un segreto Kubernetes che contiene informazioni INI.

.spec.queueManager.ini.configMap

ConfigMap rappresenta una ConfigMap Kubernetes che contiene informazioni INI.

Viene visualizzato in:

- “.spec.queueManager.ini” a pagina 143

Campo	Descrizione
`items` schiera	Le chiavi all'interno dell'origine Kubernetes che devono essere applicate.
`name` Stringa	Il nome dell'origine Kubernetes .

.spec.queueManager.ini.secret

Il segreto rappresenta un segreto Kubernetes che contiene informazioni INI.

Viene visualizzato in:

- “.spec.queueManager.ini” a pagina 143

Campo	Descrizione
`items` schiera	Le chiavi all'interno dell'origine Kubernetes che devono essere applicate.
`name` Stringa	Il nome dell'origine Kubernetes .

.spec.queueManager.livenessProbe

Impostazioni che controllano il probe di attività.

Viene visualizzato in:

- “.spec.queueManager” a pagina 141

Campo	Descrizione
`failureThreshold` intero	Numero minimo di errori consecutivi perché l'analisi venga considerata non riuscita dopo l'esito positivo. L'impostazione predefinita è 1.

Campo	Descrizione
`initialDelaySeconds` intero	Numero di secondi dopo l'avvio del contenitore prima dell'avvio dell'analisi. Il valore predefinito è 90 secondi per SingleInstance. Il valore predefinito è 0 secondi per le distribuzioni MultiInstance e NativeHA . Ulteriori informazioni: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes.
`periodSeconds` intero	La frequenza (in secondi) con cui eseguire l'analisi. Il valore predefinito è 10 secondi.
`successThreshold` intero	Numero minimo di esiti positivi consecutivi perché il probe venga considerato riuscito dopo l'errore. L'impostazione predefinita è 1.
`timeoutSeconds` intero	Numero di secondi dopo i quali l'analisi va in timeout. Il valore predefinito è 5 secondi. Ulteriori informazioni: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes.

.spec.queueManager.metrics

Impostazioni per le metriche in stile Prometheus.

Viene visualizzato in:

- ".spec.queueManager" a pagina 141

Campo	Descrizione
`enabled` booleano	Indica se abilitare o meno un endpoint per le metriche compatibili con Prometheus. L'impostazione predefinita è true.

.spec.queueManager.mqsc

Origine dei file di configurazione MQSC.

Viene visualizzato in:

- ".spec.queueManager" a pagina 141

Campo	Descrizione
`configMap` ConfigMapMQSCSource	ConfigMap rappresenta un ConfigMap Kubernetes che contiene informazioni MQSC.
`secret` SecretMQSCSource	Il segreto rappresenta un segreto Kubernetes che contiene informazioni MQSC.

.spec.queueManager.mqsc.configMap

ConfigMap rappresenta un ConfigMap Kubernetes che contiene informazioni MQSC.

Viene visualizzato in:

- ".spec.queueManager.mqsc" a pagina 144

Campo	Descrizione
`items` schiera	Le chiavi all'interno dell'origine Kubernetes che devono essere applicate.
`name` Stringa	Il nome dell'origine Kubernetes .

.spec.queueManager.mqsc.secret

Il segreto rappresenta un segreto Kubernetes che contiene informazioni MQSC.

Viene visualizzato in:

- “.spec.queueManager.mqsc” a pagina 144

Campo	Descrizione
`items` schiera	Le chiavi all'interno dell'origine Kubernetes che devono essere applicate.
`name` Stringa	Il nome dell'origine Kubernetes .

.spec.queueManager.readinessProbe

Impostazioni che controllano il probe di disponibilità.

Viene visualizzato in:

- “.spec.queueManager” a pagina 141

Campo	Descrizione
`failureThreshold` intero	Numero minimo di errori consecutivi perché l'analisi venga considerata non riuscita dopo l'esito positivo. L'impostazione predefinita è 1.
`initialDelaySeconds` intero	Numero di secondi dopo l'avvio del contenitore prima dell'avvio dell'analisi. Il valore predefinito è 10 secondi per SingleInstance. Il valore predefinito è 0 per le installazioni MultiInstance e NativeHA . Ulteriori informazioni: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes.
`periodSeconds` intero	La frequenza (in secondi) con cui eseguire l'analisi. Il valore predefinito è 5 secondi.
`successThreshold` intero	Numero minimo di esiti positivi consecutivi perché il probe venga considerato riuscito dopo l'errore. L'impostazione predefinita è 1.
`timeoutSeconds` intero	Numero di secondi dopo i quali l'analisi va in timeout. Il valore predefinito è 3 secondi. Ulteriori informazioni: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes.

.spec.queueManager.recoveryLogs

Impostazioni per i log di ripristino di MQ . Richiede l'operatore MQ 2.4.0 o superiore.

Viene visualizzato in:

- “.spec.queueManager” a pagina 141

Campo	Descrizione
`logFilePages` intero	I dati del log di ripristino sono contenuti in una serie di file. La dimensione del file di log è specificata in unità di pagine da 4 KB.

.spec.queueManager.resources

Impostazioni che controllano i requisiti delle risorse.

Viene visualizzato in:

- “.spec.queueManager” a pagina 141

Campo	Descrizione
`limits` Limiti	Impostazioni CPU & memoria.
`requests` Richieste	Impostazioni CPU & memoria.

.spec.queueManager.resources.limits

Impostazioni CPU & memoria.

Viene visualizzato in:

- ".spec.queueManager.resources" a pagina 145

Campo	Descrizione
`cpu`	
`memory`	

.spec.queueManager.resources.requests

Impostazioni CPU & memoria.

Viene visualizzato in:

- ".spec.queueManager.resources" a pagina 145

Campo	Descrizione
`cpu`	
`memory`	

.spec.queueManager.route

Impostazioni per l'instradamento del gestore code. Richiede l'operatore MQ 1.4.0 o superiore.

Viene visualizzato in:

- ".spec.queueManager" a pagina 141

Campo	Descrizione
`enabled` booleano	Indica se abilitare o meno l'instradamento. L'impostazione predefinita è true.

.spec.queueManager.startupProbe

Impostazioni che controllano il probe di avvio. Si applica solo alle distribuzioni MultiInstance e NativeHA . Richiede l'operatore MQ 1.5.0 o superiore.

Viene visualizzato in:

- ".spec.queueManager" a pagina 141

Campo	Descrizione
`failureThreshold` intero	Numero minimo di errori consecutivi per il probe da considerare non riuscito. Il valore predefinito è 24.
`initialDelaySeconds` intero	Numero di secondi dopo l'avvio del contenitore prima dell'avvio dell'analisi. Il valore predefinito è 0 secondi. Ulteriori informazioni: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes.
`periodSeconds` intero	La frequenza (in secondi) con cui eseguire l'analisi. Il valore predefinito è 5 secondi.
`successThreshold` intero	Numero minimo di successi consecutivi per il probe da considerare riuscito. L'impostazione predefinita è 1.

Campo	Descrizione
`timeoutSeconds` intero	Numero di secondi dopo i quali l'analisi va in timeout. Il valore predefinito è 5 secondi. Ulteriori informazioni: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes.

.spec.queueManager.storage

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

Viene visualizzato in:

- ".spec.queueManager" a pagina 141

Campo	Descrizione
`allowVolumeExpansion` booleano	Indica se consentire o meno l'espansione dei volumi.
`defaultClass` Stringa	La classe di memoria da applicare a tutti i volumi permanenti di questo gestore code per impostazione predefinita. I volumi persistenti specifici possono definire la propria classe di archiviazione che sovrascriverà questa impostazione della classe di archiviazione predefinita. Se `type of availability` è `SingleInstance` o `NativeHA`, la classe di memoria può essere di tipo ReadWriteUna volta o ReadWriteMolti. Se `type of availability` è `MultiInstance`, la classe di memoria deve essere di tipo ReadWriteMany.
`defaultDeleteClaim` booleano	Indica se tutti i volumi devono essere eliminati o meno quando viene eliminato il gestore code. Volumi persistenti specifici possono definire il proprio valore per deleteClaim che sovrascriverà questa impostazione di richiesta defaultDelete. Il valore predefinito è false.
`persistedData` QueueManagerOptionalVolume	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` QueueManagerVolume	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	Dettagli del volume persistente per i log di ripristino di MQ . Obbligatorio quando si utilizza il gestore code a più istanze.
`scratch` Svuota	Impostazioni per il volume temporaneo vuoto del gestore code. Questo volume verrà montato come cartella '/run' sul contenitore. Applicabile solo se il file system root è impostato su sola lettura. Richiede l'operatore MQ 3.0.0 o superiore.
`tmp` Tmp	Impostazioni per il volume temporaneo Tmp del gestore code. Questo volume verrà montato sul contenitore come cartella '/tmp'. I file di dati diagnostici, come il file zip prodotto dal comando runmqras, verranno creati in questo volume. Applicabile solo se il file system root è impostato su sola lettura. Richiede l'operatore MQ 3.0.0 o superiore.

.spec.queueManager.storage.persistedData

Dettagli PersistentVolume per i dati persistenti di MQ , inclusi la configurazione, code e messaggi. Obbligatorio quando si utilizza il gestore code a più istanze.

Viene visualizzato in:

- ".spec.queueManager.storage" a pagina 147

Campo	Descrizione
`class` Stringa	Classe di memoria da utilizzare per questo volume. Valido solo se `type` è `persistent-claim`. Se `type of availability` è `SingleInstance` o `NativeHA`, la classe di memoria può essere di tipo ReadWriteUna volta o ReadWriteMolti. Se `type of availability` è `MultiInstance`, la classe di memoria deve essere di tipo ReadWriteMany.
`deleteClaim` booleano	Indica se questo volume deve essere eliminato o meno quando viene eliminato il gestore code.
`enabled` booleano	Indica se questo volume deve essere abilitato o meno come volume separato o se deve essere posizionato sul volume `queueManager` predefinito. Il valore predefinito è false.
`size` Stringa	Dimensione del PersistentVolume da passare a Kubernetes, incluse le unità SI. Valido solo se `type` è `persistent-claim`. Ad esempio, `2Gi`. Il valore predefinito è `2Gi`.
`sizeLimit` Stringa	Limite dimensione quando si utilizza un volume `ephemeral` . I file sono ancora scritti in una directory temporanea, quindi è possibile utilizzare questa opzione per limitare la dimensione. Valido solo se `type` è `ephemeral` e il filesystem root è impostato su sola lettura. Richiede l'operatore MQ 3.0.0 o superiore.
`type` Stringa	Tipo di volume da utilizzare. Scegli `ephemeral` per utilizzare l'archiviazione non persistente o `persistent-claim` per utilizzare un volume persistente. Il valore predefinito è `persistent-claim`.

.spec.queueManager.storage.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:

- ".spec.queueManager.storage" a pagina 147

Campo	Descrizione
`class` Stringa	Classe di memoria da utilizzare per questo volume. Valido solo se `type` è `persistent-claim`. Se `type of availability` è `SingleInstance` o `NativeHA`, la classe di memoria può essere di tipo ReadWriteUna volta o ReadWriteMolti. Se `type of availability` è `MultiInstance`, la classe di memoria deve essere di tipo ReadWriteMany.
`deleteClaim` booleano	Indica se questo volume deve essere eliminato o meno quando viene eliminato il gestore code.
`size` Stringa	Dimensione del PersistentVolume da passare a Kubernetes, incluse le unità SI. Valido solo se `type` è `persistent-claim`. Ad esempio, `2Gi`. Il valore predefinito è `2Gi`.
`sizeLimit` Stringa	Limite dimensione quando si utilizza un volume `ephemeral` . I file sono ancora scritti in una directory temporanea, quindi è possibile utilizzare questa opzione per limitare la dimensione. Valido solo se `type` è `ephemeral` e il filesystem root è impostato su sola lettura. Richiede l'operatore MQ 3.0.0 o superiore.

Campo	Descrizione
`type` Stringa	Tipo di volume da utilizzare. Scegli `ephemeral` per utilizzare l'archiviazione non persistente o `persistent-claim` per utilizzare un volume persistente. Il valore predefinito è `persistent-claim`.

.spec.queueManager.storage.recoveryLogs

Dettagli del volume persistente per i log di ripristino di MQ . Obbligatorio quando si utilizza il gestore code a più istanze.

Viene visualizzato in:

- ".spec.queueManager.storage" a pagina 147

Campo	Descrizione
`class` Stringa	Classe di memoria da utilizzare per questo volume. Valido solo se `type` è `persistent-claim`. Se `type of availability` è `SingleInstance` o `NativeHA`, la classe di memoria può essere di tipo ReadWriteUna volta o ReadWriteMolti. Se `type of availability` è `MultiInstance`, la classe di memoria deve essere di tipo ReadWriteMany.
`deleteClaim` booleano	Indica se questo volume deve essere eliminato o meno quando viene eliminato il gestore code.
`enabled` booleano	Indica se questo volume deve essere abilitato o meno come volume separato o se deve essere posizionato sul volume `queueManager` predefinito. Il valore predefinito è false.
`size` Stringa	Dimensione del PersistentVolume da passare a Kubernetes, incluse le unità SI. Valido solo se `type` è `persistent-claim`. Ad esempio, `2Gi`. Il valore predefinito è `2Gi`.
`sizeLimit` Stringa	Limite dimensione quando si utilizza un volume `ephemeral` . I file sono ancora scritti in una directory temporanea, quindi è possibile utilizzare questa opzione per limitare la dimensione. Valido solo se `type` è `ephemeral` e il filesystem root è impostato su sola lettura. Richiede l'operatore MQ 3.0.0 o superiore.
`type` Stringa	Tipo di volume da utilizzare. Scegli `ephemeral` per utilizzare l'archiviazione non persistente o `persistent-claim` per utilizzare un volume persistente. Il valore predefinito è `persistent-claim`.

.spec.queueManager.storage.scratch

Impostazioni per il volume temporaneo vuoto del gestore code. Questo volume verrà montato come cartella '/run' sul contenitore. Applicabile solo se il file system root è impostato su sola lettura. Richiede l'operatore MQ 3.0.0 o superiore.

Viene visualizzato in:

- ".spec.queueManager.storage" a pagina 147

Campo	Descrizione
`sizeLimit` Stringa	Limite di dimensione del volume effimero, incluse le unità SI. Ad esempio, `2Gi`. Valido solo quando il filesystem root è impostato su sola lettura. Richiede l'operatore MQ 3.0.0 o superiore.

.spec.queueManager.storage.tmp

Impostazioni per il volume temporaneo Tmp del gestore code. Questo volume verrà montato sul contenitore come cartella '/tmp'. I file di dati diagnostici, come il file zip prodotto dal comando runmqras, verranno creati in questo volume. Applicabile solo se il file system root è impostato su sola lettura. Richiede l'operatore MQ 3.0.0 o superiore.

Viene visualizzato in:

- ".spec.queueManager.storage" a pagina 147

Campo	Descrizione
`sizeLimit` Stringa	Limite di dimensione del volume effimero, incluse le unità SI. Ad esempio, `2Gi`. Valido solo quando il filesystem root è impostato su sola lettura. Richiede l'operatore MQ 3.0.0 o superiore.

.spec.securityContext

Impostazioni di protezione da aggiungere al securityContextdi Queue Manager Pod.

Viene visualizzato in:

- ".spec" a pagina 138

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 questo valore su `true` se si sta utilizzando 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` influisce sull'oggetto SCC (Security Context Constraints) che è possibile utilizzare e l'avvio del gestore code potrebbe non riuscire se non si è autorizzati ad utilizzare un SCC che consente l'utente root. Il valore predefinito è false. Per ulteriori informazioni, consultare https://docs.openshift.com/container-platform/latest/authentication/managing-security-context-constraints.html.
`readOnlyRootFilesystem` booleano	Indica se abilitare o meno le impostazioni del file system root di sola lettura per il gestore code. Il valore predefinito è false. Richiede l'operatore MQ 3.0.0 o superiore.
`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.

.spec.telemetry

Impostazioni per la configurazione di Open Telemetry. Richiede l'operatore MQ 2.2.0 o superiore.

Viene visualizzato in:

- ".spec" a pagina 138

Campo	Descrizione
`tracing` Traccia	Impostazioni per la traccia Open Telemetry.

.spec.telemetry.tracing

Impostazioni per la traccia Open Telemetry.

Viene visualizzato in:

- ".spec.telemetry" a pagina 150

Campo	Descrizione
`instana` Instana	Impostazioni per la traccia Instana.

.spec.telemetry.tracing.instana

Impostazioni per la traccia Instana.

Viene visualizzato in:

- ".spec.telemetry.tracing" a pagina 151

Campo	Descrizione
`agentHost` Stringa	Il nome host dell'agent Instana a cui inviare i dati di traccia. Questo non deve includere un protocollo.
`enabled` booleano	Indica se abilitare o meno la traccia Instana. Il valore predefinito è false.
`protocol` Stringa	Il protocollo da utilizzare nella comunicazione con l'agente Instana. http e https sono supportati.

.spec.template

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:

- ".spec" a pagina 138

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.

.spec.tracing

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

Viene visualizzato in:

- ".spec" a pagina 138

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

Campo	Descrizione
`collector` TracingCollector	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. Il valore predefinito è false.
`namespace` Stringa	Spazio dei nomi in cui è installato il dashboard Operazioni di Cloud Pak for Integration .

.spec.tracing.agent

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

Viene visualizzato in:

- ".spec.tracing" a pagina 151

Campo	Descrizione
`image` Stringa	L'immagine contenitore che verrà utilizzata.
`imagePullPolicy` Stringa	Impostazione che controlla quando il kubelet tenta di estrarre l'immagine specificata. Il valore predefinito è `IfNotPresent`.
`livenessProbe` TracingProbe	Impostazioni che controllano il probe di attività.
`readinessProbe` TracingProbe	Impostazioni che controllano il probe di disponibilità.

.spec.tracing.agent.livenessProbe

Impostazioni che controllano il probe di attività.

Viene visualizzato in:

- ".spec.tracing.agent" a pagina 152

Campo	Descrizione
`failureThreshold` intero	Numero minimo di errori consecutivi perché l'analisi venga considerata non riuscita dopo l'esito positivo. L'impostazione predefinita è 1.
`initialDelaySeconds` intero	Numero di secondi dopo che il contenitore è stato avviato prima dell'avvio delle analisi di attività. Il valore predefinito è 10 secondi. Ulteriori informazioni: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes.
`periodSeconds` intero	La frequenza (in secondi) con cui eseguire l'analisi. Il valore predefinito è 10 secondi.
`successThreshold` intero	Numero minimo di esiti positivi consecutivi perché il probe venga considerato riuscito dopo l'errore. L'impostazione predefinita è 1.
`timeoutSeconds` intero	Numero di secondi dopo i quali l'analisi va in timeout. Il valore predefinito è 2 secondi. Ulteriori informazioni: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes.

.spec.tracing.agent.readinessProbe

Impostazioni che controllano il probe di disponibilità.

Viene visualizzato in:

- ".spec.tracing.agent" a pagina 152

Campo	Descrizione
`failureThreshold` intero	Numero minimo di errori consecutivi perché l'analisi venga considerata non riuscita dopo l'esito positivo. L'impostazione predefinita è 1.
`initialDelaySeconds` intero	Numero di secondi dopo che il contenitore è stato avviato prima dell'avvio delle analisi di attività. Il valore predefinito è 10 secondi. Ulteriori informazioni: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes.
`periodSeconds` intero	La frequenza (in secondi) con cui eseguire l'analisi. Il valore predefinito è 10 secondi.
`successThreshold` intero	Numero minimo di esiti positivi consecutivi perché il probe venga considerato riuscito dopo l'errore. L'impostazione predefinita è 1.
`timeoutSeconds` intero	Numero di secondi dopo i quali l'analisi va in timeout. Il valore predefinito è 2 secondi. Ulteriori informazioni: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes.

.spec.tracing.collector

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

Viene visualizzato in:

- ".spec.tracing" a pagina 151

Campo	Descrizione
`image` Stringa	L'immagine contenitore che verrà utilizzata.
`imagePullPolicy` Stringa	Impostazione che controlla quando il kubelet tenta di estrarre l'immagine specificata. Il valore predefinito è `IfNotPresent`.
`livenessProbe` TracingProbe	Impostazioni che controllano il probe di attività.
`readinessProbe` TracingProbe	Impostazioni che controllano il probe di disponibilità.

.spec.tracing.collector.livenessProbe

Impostazioni che controllano il probe di attività.

Viene visualizzato in:

- ".spec.tracing.collector" a pagina 153

Campo	Descrizione
`failureThreshold` intero	Numero minimo di errori consecutivi perché l'analisi venga considerata non riuscita dopo l'esito positivo. L'impostazione predefinita è 1.
`initialDelaySeconds` intero	Numero di secondi dopo che il contenitore è stato avviato prima dell'avvio delle analisi di attività. Il valore predefinito è 10 secondi. Ulteriori informazioni: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes.

Campo	Descrizione
`periodSeconds` intero	La frequenza (in secondi) con cui eseguire l'analisi. Il valore predefinito è 10 secondi.
`successThreshold` intero	Numero minimo di esiti positivi consecutivi perché il probe venga considerato riuscito dopo l'errore. L'impostazione predefinita è 1.
`timeoutSeconds` intero	Numero di secondi dopo i quali l'analisi va in timeout. Il valore predefinito è 2 secondi. Ulteriori informazioni: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes.

.spec.tracing.collector.readinessProbe

Impostazioni che controllano il probe di disponibilità.

Viene visualizzato in:

- ".spec.tracing.collector" a pagina 153

Campo	Descrizione
`failureThreshold` intero	Numero minimo di errori consecutivi perché l'analisi venga considerata non riuscita dopo l'esito positivo. L'impostazione predefinita è 1.
`initialDelaySeconds` intero	Numero di secondi dopo che il contenitore è stato avviato prima dell'avvio delle analisi di attività. Il valore predefinito è 10 secondi. Ulteriori informazioni: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes.
`periodSeconds` intero	La frequenza (in secondi) con cui eseguire l'analisi. Il valore predefinito è 10 secondi.
`successThreshold` intero	Numero minimo di esiti positivi consecutivi perché il probe venga considerato riuscito dopo l'errore. L'impostazione predefinita è 1.
`timeoutSeconds` intero	Numero di secondi dopo i quali l'analisi va in timeout. Il valore predefinito è 2 secondi. Ulteriori informazioni: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes.

.spec.web

Impostazioni per il server Web MQ .

Viene visualizzato in:

- ".spec" a pagina 138

Campo	Descrizione
`console` Console	Impostazioni per la console Web di MQ . Richiede l'operatore MQ 3.0.0 o superiore.
`enabled` booleano	Indica se abilitare o meno il server Web. Il valore predefinito è false.
`manualConfig` ManualConfig	Impostazioni per fornire la configurazione XML del server Web. Richiede l'operatore MQ 3.0.0 o superiore.

.spec.web.console

Impostazioni per la console Web di MQ . Richiede l'operatore MQ 3.0.0 o superiore.

Viene visualizzato in:

- ".spec.web" a pagina 154

Campo	Descrizione
`authentication` Autenticazione	Impostazioni di autenticazione per la console Web di MQ . Richiede l'operatore MQ 3.0.0 o superiore.
`authorization` Autorizzazione	Impostazioni di autorizzazione per la console Web di MQ . Richiede l'operatore MQ 3.0.0 o superiore.

.spec.web.console.authentication

Impostazioni di autenticazione per la console Web di MQ . Richiede l'operatore MQ 3.0.0 o superiore.

Viene visualizzato in:

- ".spec.web.console" a pagina 154

Campo	Descrizione
`provider` Stringa	Il fornitore di autenticazione da utilizzare per la console Web di MQ . Utilizza `integration-keycloak` per utilizzare SSO (single sign - on) con la IU della piattaforma Cloud Pak for Integration (Keycloak). Il valore predefinito è `integration-keycloak` se si utilizza una licenza Cloud Pak for Integration o `manual` se si utilizza una licenza MQ . Utilizzare `manual` se si desidera fornire la propria configurazione.

.spec.web.console.authorization

Impostazioni di autorizzazione per la console Web di MQ . Richiede l'operatore MQ 3.0.0 o superiore.

Viene visualizzato in:

- ".spec.web.console" a pagina 154

Campo	Descrizione
`provider` Stringa	Il provider di autorizzazione da utilizzare per la console Web di MQ . Utilizza `integration-keycloak` per utilizzare i ruoli forniti da Cloud Pak for Integration Keycloak. Utilizzare `manual` se si desidera fornire la propria configurazione. Il valore predefinito è `integration-keycloak` se si utilizza una licenza Cloud Pak for Integration o `manual` se si utilizza una licenza MQ .

.spec.web.manualConfig

Impostazioni per fornire la configurazione XML del server Web. Richiede l'operatore MQ 3.0.0 o superiore.

Viene visualizzato in:

- ".spec.web" a pagina 154

Campo	Descrizione
`configMap` ConfigMap	ConfigMap rappresenta una ConfigMap Kubernetes che contiene la configurazione XML del server Web.
`secret` Segreto	Il segreto rappresenta un segreto Kubernetes che contiene la configurazione XML del server web. L'utilizzo di un segreto protegge le credenziali nel livello Kubernetes , ma è possibile che gli strumenti di monitoraggio o di risoluzione dei problemi possano esporre il file sottostante in modo non sicuro. Per migliorare la sicurezza, codificare le credenziali utilizzando "securityUtility".

.spec.web.manualConfig.configMap

ConfigMap rappresenta una ConfigMap Kubernetes che contiene la configurazione XML del server Web.

Viene visualizzato in:

- ".spec.web.manualConfig" a pagina 155

Campo	Descrizione
name Stringa	Il nome dell'origine Kubernetes .

.spec.web.manualConfig.secret

Il segreto rappresenta un segreto Kubernetes che contiene la configurazione XML del server web. L'utilizzo di un segreto protegge le credenziali nel livello Kubernetes , ma è possibile che gli strumenti di monitoraggio o di risoluzione dei problemi possano esporre il file sottostante in modo non sicuro. Per migliorare la sicurezza, codificare le credenziali utilizzando "securityUtility".

Viene visualizzato in:

- ".spec.web.manualConfig" a pagina 155

Campo	Descrizione
name Stringa	Il nome dell'origine Kubernetes .

.stato

Lo stato osservato di QueueManager.

Viene visualizzato in:

- "QueueManager" a pagina 137

Campo	Descrizione
adminUiUrl Stringa	URL per l'interfaccia utente Admin.
availability Disponibilità	Lo stato di disponibilità per il gestore code.
Array conditions QueueManagerStatusCondition	Le condizioni rappresentano le ultime osservazioni disponibili dello stato del gestore code.
Array endpoints QueueManagerStatusEndpoint	Informazioni sugli endpoint che questo gestore code sta esponendo, come gli endpoint API o UI.
metadata Metadati	I metadati rappresentano ulteriori informazioni per il gestore code, incluso lo statoKeycloak dell'integrazione.
name Stringa	Il nome del gestore code.
phase Stringa	Fase dello stato del gestore code.
versions QueueManagerStatusVersion	Versione di MQ utilizzata e altre versioni disponibili da IBM Entitled Registry.

.status.availability

Lo stato di disponibilità per il gestore code.

Viene visualizzato in:

- ".stato" a pagina 156

Campo	Descrizione
`initialQuorumEstablished` booleano	Se è stato stabilito o meno un quorum iniziale per NativeHA.

.status.conditions

QueueManagerStatusCondition definisce le condizioni del Gestore code.

Viene visualizzato in:

- ".stato" a pagina 156

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

.status.endpoints

QueueManagerStatusEndpoint definisce gli endpoint per QueueManager.

Viene visualizzato in:

- ".stato" a pagina 156

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.

.status.metadata

I metadati rappresentano ulteriori informazioni per il gestore code, incluso lo statoKeycloak dell'integrazione.

Viene visualizzato in:

- ".stato" a pagina 156

Campo	Descrizione
`integrationKeycloak` IntegrationKeycloak	QueueManagerStatusIntegrationKeycloak definisce lo stato Integration -Keycloak per QueueManager.

.status.metadata.integrationKeycloak

QueueManagerStatusIntegrationKeycloak definisce lo stato Integration -Keycloak per QueueManager.

Viene visualizzato in:

- ".status.metadata" a pagina 157

Campo	Descrizione
`clientName` Stringa	

.status.versions

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

Viene visualizzato in:

- ".stato" a pagina 156

Campo	Descrizione
`available` QueueManagerStatusVersion Disponibile	Altre versioni di MQ disponibili da IBM Entitled Registry.
`reconciled` Stringa	La specifica versione di IBM MQ utilizzata. Se viene specificata un'immagine personalizzata, potrebbe non corrispondere alla versione di MQ effettivamente utilizzata.

.status.versions.available

Altre versioni di MQ disponibili da IBM Entitled Registry.

Viene visualizzato in:

- ".status.versions" a pagina 158

Campo	Descrizione
`channels` schiera	I canali disponibili per l'aggiornamento automatico della versione di MQ .
Array `versions` Versions	Versioni specifiche di MQ disponibili.

.status.versions.available.versions

QueueManagerStatusVersion definisce una versione di MQ.

Viene visualizzato in:

- ".status.versions.available" a pagina 158

Campo	Descrizione
Array `licenses` Licenses	Le licenze applicabili per questa versione di QueueManager.
`name` Stringa	Versione `name` per questa versione di QueueManager. Questi sono valori validi per il campo `spec.version` .

.status.versions.available.versions.licenses

QueueManagerStatusLicense definisce una licenza.

Viene visualizzato in:

- ".status.versions.available.versions" a pagina 158

Campo	Descrizione
`displayName` Stringa	Nome di visualizzazione per la licenza.
`link` Stringa	Link al contenuto della licenza.

Campo	Descrizione
`matchesCurrentType` booleano	Se la licenza corrisponde o meno al tipo di licenza attualmente utilizzata.
`name` Stringa	Nome della licenza.

OpenShift CP4I *Condizioni di stato per QueueManager (mq.ibm.com/v1beta1)*

I campi **`status.conditions`** vengono aggiornati per riflettere la condizione della risorsa `QueueManager` . In generale, le condizioni descrivono situazioni anomale. Un gestore code in uno stato di integrità e pronto non ha condizioni **`Error`** o **`Pending`** . Potrebbe avere alcune condizioni **`Warning`** di avviso.

Le seguenti condizioni sono state definite per una risorsa `QueueManager` :

Tabella 2. Condizioni di stato del gestore code

Componente	Tipo di condizione	Codice di errore	Messaggio di avvertenza
QueueManager[3]	Bloccato	OperatorDependency	Per installare, questa istanza richiede la configurazione di Keycloak da parte di [IBM Cloud Pak for Integration]. Questa istanza rimarrà nello stato [In sospeso] finché Keycloak non viene riportato come [KeycloakReady] nella risorsa Cp4iServicesBinding per questa QueueManager.
			Per installare, questa istanza richiede l'operatore [IBM IAM]. Questa istanza rimarrà in stato [Bloccato] fino a quando l'operatore non verrà installato da [IBM Cloud Pak foundational services].
	In sospeso	Creazione	Il gestore code MQ è in fase di distribuzione
	In sospeso	OidcPending	Il gestore code MQ è in attesa della registrazione del client OIDC
	In sospeso	Arrestato	Il gestore code MQ è stato arrestato poiché l'annotazione 'mq.ibm.com/stop' è presente ed è impostata su 'true' nella definizione QueueManager . Quando viene arrestato, il conteggio di repliche di QueueManager StatefulSet è impostato a zero, rimuovendo tutti i pod del gestore code MQ .
	Errore	Non superato	Distribuzione del gestore code MQ non riuscita
	Avviso	UnsupportedVersion	Un operando è stato installato da un operatore che non è supportato nella versione OCP *< ocp_version>*. Questo operando non è supportato.
	Avviso	CP4I-LTS Supporto	Un CP4I-LTS operando *< mq_version>* è stato installato ma è gestito da un operatore che non è qualificato per la durata del supporto esteso. Questo operando non è idoneo per la durata di supporto esteso.
	Avviso	CP4I-LTS Supporto	Un CP4I-LTS operando *< mq_version>* è stato installato ma la versione OCP *< ocp_version>* non si qualifica per la durata del supporto esteso. Questo operando non è idoneo per la durata di supporto esteso.

[3] Le condizioni `Creating` e `Failed` monitorano l'avanzamento generale della distribuzione del gestore code. Se stai usando una licenza IBM Cloud Pak for Integration e la console web è abilitata, la condizione `OidcPending` registra lo stato del gestore code in attesa del completamento della registrazione del client OIDC con IAM.

Tabella 2. Condizioni di stato del gestore code (Continua)			
Componente	**Tipo di condizione**	**Codice di errore**	**Messaggio di avvertenza**
Pod[4]	In sospeso	PodPending	Il pod per il gestore code MQ è in fase di distribuzione
	Errore	PodFailed	Il pod per il gestore code MQ è in fase di distribuzione
Memoria[5]	In sospeso	StoragePending	È in corso il provisioning della memoria per il gestore code MQ
	Avviso	StorageEphemeral	Utilizzo della memoria temporanea per un gestore code MQ di produzione
	Avviso	StorageExpansionin sospeso	L'espansione del volume è in sospeso per le seguenti PVC [< elenco di pvcs>]
	Avviso	StorageMismatch	Le dimensioni di memoria definite nella risorsa QueueManager non corrispondono alla capacità di una o più PVC di cui è stato eseguito il provisioning [< elenco di pvcs>]. AllowVolumeExpansion è impostato su false nella risorsa QueueManager , quindi l'operatore MQ non tenterà di riconciliare queste differenze.
	Errore	StorageFailed	Impossibile eseguire il provisioning della memoria per il gestore code MQ

Linux Annotazioni di licenza durante la creazione della propria immagine del contenitore IBM MQ

Le annotazioni di licenza ti permettono di tenere traccia dell'utilizzo in base ai limiti definiti sul contenitore, piuttosto che sulla macchina sottostante. Configura i tuoi client per distribuire il contenitore con annotazioni specifiche che IBM License Service utilizza per tracciare l'utilizzo.

Quando si distribuisce un'immagine del contenitore IBM MQ auto - costruito, ci sono due approcci comuni alla licenza:

- Licenza dell'intera macchina che esegue il contenitore.
- Licenza del contenitore in base ai limiti associati.

---

[4] Le condizioni del pod monitorano lo stato dei pod durante la distribuzione di un gestore code. Se viene visualizzata una condizione `PodFailed` , anche la condizione generale del gestore code verrà impostata su `Failed`.

[5] Le condizioni di archiviazione monitorano l'avanzamento (condizione`StoragePending`) delle richieste per creare volumi per l'archiviazione persistente e riportano errori di bind di ritorno e altri errori. Le condizioni di storage monitorano anche l'avanzamento delle espansioni di volume e avvisano di non corrispondenze tra le dimensioni di storage definite nella definizione del gestore code e la dimensione delle PVC distribuite. Se si verifica un errore durante il provisioning della memoria, la condizione `StorageFailed` viene aggiunta all'elenco delle condizioni e la condizione generale del gestore code viene impostata su `Failed`.

Entrambe le opzioni sono disponibili per i client e ulteriori dettagli possono essere trovati nella pagina IBM Container Licenses in Passport Advantage.

Se il contenitore IBM MQ deve essere concesso in licenza in base ai limiti del contenitore, IBM License Service deve essere installato per tenere traccia dell'utilizzo. Ulteriori informazioni relative agli ambienti supportati e alle istruzioni di installazione sono disponibili nella pagina ibm - licensing - operator su GitHub.

IBM License Service è installato sul cluster Kubernetes in cui è distribuito il contenitore IBM MQ e le annotazioni del pod vengono utilizzate per tracciare l'utilizzo. Pertanto i client devono distribuire il pod con annotazioni specifiche che IBM License Service utilizza. In base alla tua titolarità e alle funzionalità distribuite nel contenitore, utilizza una o più delle seguenti annotazioni.

Nota: Molte delle annotazioni contengono una o entrambe le seguenti righe:

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

È necessario modificare queste righe prima di utilizzare l'annotazione:

- Per `productChargedContainers`, è necessario scegliere `"All"`o sostituire il nome effettivo del contenitore.
- Per `productMetric`, è necessario scegliere uno dei valori offerti.

Annotazioni da utilizzare con una titolarità di prodotto IBM MQ

Se hai una titolarità del prodotto IBM MQ , seleziona di seguito l'annotazione che corrisponde alla titolarità che hai acquistato e che vuoi utilizzare.

- “IBM MQ” a pagina 164
- “IBM MQ Avanzate” a pagina 164
- “IBM MQ per ambiente di non produzione” a pagina 164
- “IBM MQ Avanzate per ambienti non di produzione” a pagina 164
- “IBM MQ Advanced per gli sviluppatori” a pagina 164

Le annotazioni IBM MQ da utilizzare con le configurazioni IBM MQ Multi Instance High Availability sono le seguenti. Consultare anche “Selezione delle annotazioni corrette per le configurazioni HA (High Availability)” a pagina 163.

- “Istanze multiple del contenitore IBM MQ” a pagina 164
- “IBM MQ Istanza multipla contenitore avanzato” a pagina 165
- “IBM MQ Contenitore a più istanze per l'ambiente di non produzione” a pagina 165
- “IBM MQ Istanza multipla del contenitore avanzata per ambiente di non produzione” a pagina 165

Annotazioni da utilizzare con la titolarità prodotto CP4I

Se si dispone della titolarità IBM Cloud Pak for Integration (CP4I), selezionare l'annotazione riportata di seguito che corrisponde alla titolarità acquistata e che si desidera utilizzare.

- “Titolarità IBM MQ con CP4I” a pagina 165
- “IBM MQ Avanzato con titolarità CP4I” a pagina 165
- “IBM MQ per l'ambiente di non produzione con titolarità CP4I” a pagina 165
- “IBM MQ Avanzate per ambienti non di produzione con titolarità CP4I” a pagina 165

Le annotazioni CP4I da utilizzare con le configurazioni IBM MQ Multi Instance High Availability sono le seguenti. Consultare anche “Selezione delle annotazioni corrette per le configurazioni HA (High Availability)” a pagina 163.

- “Titolarità IBM MQ Container Multi Instance with CP4I” a pagina 166

- “Titolarità IBM MQ Advanced Container Multi Instance with CP4I” a pagina 166
- “IBM MQ Container Multi Instance for Non - Production Environment con titolarità CP4I” a pagina 166
- “Titolarità IBM MQ Advanced Container Multi Instance for Non - Production Environment con CP4I” a pagina 166

Selezione delle annotazioni corrette per le configurazioni HA (High Availability)

IBM MQ Istanza multipla

Quando distribuisci una coppia di gestori code in una configurazione ad alta disponibilità a più istanze IBM MQ , devi utilizzare la stessa annotazione su entrambe le istanze. È necessario selezionare una delle seguenti annotazioni, a seconda della titolarità acquistata:

- Titolarità autonoma IBM MQ o IBM MQ Advanced
  - “Istanze multiple del contenitore IBM MQ” a pagina 164
  - “IBM MQ Istanza multipla contenitore avanzato” a pagina 165
  - “IBM MQ Contenitore a più istanze per l'ambiente di non produzione” a pagina 165
  - “IBM MQ Istanza multipla del contenitore avanzata per ambiente di non produzione” a pagina 165
- IBM Cloud Pak for Integration titolarità
  - “Titolarità IBM MQ Container Multi Instance with CP4I” a pagina 166
  - “Titolarità IBM MQ Advanced Container Multi Instance with CP4I” a pagina 166
  - “IBM MQ Container Multi Instance for Non - Production Environment con titolarità CP4I” a pagina 166
  - “Titolarità IBM MQ Advanced Container Multi Instance for Non - Production Environment con CP4I” a pagina 166

Quando vengono utilizzati con la titolarità IBM Cloud Pak for Integration , i rapporti di titolarità nelle annotazioni garantiscono che venga registrato il consumo di titolarità corretto. Quando vengono utilizzate con titolarità IBM MQ o IBM MQ Advanced autonome, le annotazioni riportate nel License Service per ogni istanza devono essere associate alle parti di titolarità IBM MQ nel modo seguente:

- IBM MQ Advanced container Istanza multipla
  - 1 x IBM MQ Advanced **e** 1 x IBM MQ Advanced High Availability Replica **o**
  - 2 x IBM MQ Advanced[6]
- IBM MQ Advanced container Istanza multipla per l'ambiente di non produzione
  - 1 x IBM MQ Advanced **e** 1 x IBM MQ Advanced High Availability Replica **o**
  - 2 x IBM MQ Advanced per ambiente non di produzione)[6]
- Istanze multiple del contenitore IBM MQ
  - 1 x IBM MQ **e** 1 x IBM MQ High Availability Replica **o**
  - 2 x IBM MQ[6]
- IBM MQ Contenitore a più istanze per l'ambiente di non produzione
  - 1 x IBM MQ **e** 1 x IBM MQ High Availability Replica **o**
  - 2 x IBM MQ per ambiente non di produzione) [6]

IBM MQ HA nativa

Se si stanno distribuendo tre gestori code in un quorum HA nativo, solo l'istanza attiva utilizza la titolarità. Tutte le istanze devono avere la stessa annotazione. A seconda della titolarità acquistata, è necessario selezionare una delle seguenti opzioni:

[6] Questa opzione di titolarità è sub - ottimale e deve essere utilizzata solo se non è disponibile alcuna titolarità della parte di replica ad alta disponibilità pertinente.

- Titolarità autonoma IBM MQ o IBM MQ Advanced
  - “IBM MQ Avanzate” a pagina 164
  - “IBM MQ Avanzate per ambienti non di produzione” a pagina 164
- IBM Cloud Pak for Integration titolarità
  - “IBM MQ Avanzato con titolarità CP4I” a pagina 165
  - “IBM MQ Avanzate per ambienti non di produzione con titolarità CP4I” a pagina 165

Annotazioni

Il resto di questo argomento descrive in dettaglio il contenuto di ogni annotazione.

IBM MQ

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

IBM MQ Avanzate

```
productID: "208423bb063c43288328b1d788745b0c"
productName: "IBM MQ Advanced"
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

IBM MQ per ambiente di non produzione

```
productID: "151bec68564a4a47a14e6fa99266deff"
productName: "IBM MQ for Non-Production Environment"
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

IBM MQ Avanzate per ambienti non di produzione

```
productID: "21dfe9a0f00f444f888756d835334909"
productName: "IBM MQ Advanced for Non-Production Environment"
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

IBM MQ Advanced per gli sviluppatori

```
productID: "2f886a3eefbe4ccb89b2adb97c78b9cb"
productName: "IBM MQ Advanced for Developers (Non-Warranted)"
productMetric: "FREE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

Istanze multiple del contenitore IBM MQ

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

IBM MQ Istanza multipla contenitore avanzato

```
productID: "bd35bff411bb47c2a3f3a4590f33a8ef"
productName: "IBM MQ Advanced Container Multi Instance"
productMetric: "PROCESSOR_VALUE_UNIT" | "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

IBM MQ Contenitore a più istanze per l'ambiente di non produzione

```
productID: "af11b093f16a4a26806013712b860b60"
productName: "IBM MQ Container Multi Instance for Non-Production Environment"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

IBM MQ Istanza multipla del contenitore avanzata per ambiente di non produzione

```
productID: "31f844f7a96b49749130cd0708fdbb17"
productName: "IBM MQ Advanced Container Multi Instance for Non-Production Environment"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
```

Titolarità IBM MQ con CP4I

```
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
cloudpakName: "IBM Cloud Pak for Integration"
productID: "c661609261d5471fb4ff8970a36bccea"
productName: "IBM MQ"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productCloudpakRatio: "4:1"
```

IBM MQ Avanzato con titolarità CP4I

```
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
cloudpakName: "IBM Cloud Pak for Integration"
productID: "208423bb063c43288328b1d788745b0c"
productName: "IBM MQ Advanced"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productCloudpakRatio: "2:1"
```

IBM MQ per l'ambiente di non produzione con titolarità CP4I

```
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
cloudpakName: "IBM Cloud Pak for Integration"
productID: "151bec68564a4a47a14e6fa99266deff"
productName: "IBM MQ for Non-Production Environment"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productChargedContainers:  "All" | "NAME_OF_CONTAINER"
productCloudpakRatio: "8:1"
```

IBM MQ Avanzate per ambienti non di produzione con titolarità CP4I

```
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
cloudpakName: "IBM Cloud Pak for Integration"
productID: "21dfe9a0f00f444f888756d835334909"
productName: "IBM MQ Advanced for Non-Production Environment"
productMetric: "VIRTUAL_PROCESSOR_CORE"
```

```
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productCloudpakRatio: "4:1"
```

Titolarità IBM MQ Container Multi Instance with CP4I

```
productName: "IBM MQ Container Multi Instance"
productID: "2dea73b866b648b6b4abe2a85eb76964"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productCloudpakRatio: "10:3"
cloudpakName: "IBM Cloud Pak for Integration"
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
```

Titolarità IBM MQ Advanced Container Multi Instance with CP4I

```
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
cloudpakName: "IBM Cloud Pak for Integration"
productID: "bd35bff411bb47c2a3f3a4590f33a8ef"
productName: "IBM MQ Advanced Container Multi Instance"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productCloudpakRatio: "5:3"
```

IBM MQ Container Multi Instance for Non - Production Environment con titolarità CP4I

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

Titolarità IBM MQ Advanced Container Multi Instance for Non - Production Environment con CP4I

```
cloudpakId: "c8b82d189e7545f0892db9ef2731b90d"
cloudpakName: "IBM Cloud Pak for Integration"
productID: "31f844f7a96b49749130cd0708fdbb17"
productName: "IBM MQ Advanced Container Multi Instance for Non-Production Environment"
productMetric: "VIRTUAL_PROCESSOR_CORE"
productChargedContainers: "All" | "NAME_OF_CONTAINER"
productCloudpakRatio: "10:3"
```

IBM MQ Advanced for Developers immagine contenitore

Un'immagine del contenitore precostruita è disponibile per IBM MQ Advanced for Developers. Questa immagine è disponibile da IBM Container Registry. Questa immagine può essere utilizzata con Docker, Podman, Kubernetese altri ambienti del contenitore.

Immagini disponibili

Le immagini IBM MQ sono memorizzate in IBM Container Registry:

- IBM MQ Advanced for Developers 9.4.0.0: `icr.io/ibm-messaging/mq:9.4.0.0-r1`

Riferimento rapido

- Licenza:
  - Riferimento per la licenza per mq.ibm.com/v1beta1 e Apache License 2.0. Tenere presente che la licenza IBM MQ Advanced for Developers non consente un'ulteriore distribuzione e che i termini limitano l'uso a una macchina dello sviluppatore.
- Dove archiviare i problemi:
  - GitHub
- Disponibile per le seguenti architetture CPU:
  - amd64
  - s390x
  - ppc64le

Utilizzo

Esegui IBM MQ Advanced for Developers in un contenitore.

Vedi la documentazione sull'utilizzo per i dettagli su come eseguire un contenitore.

Per poter utilizzare l'immagine, devi accettare i termini della licenza IBM MQ impostando la variabile di ambiente **LICENSE** .

Variabili di ambiente supportate

LANG
Impostare la lingua in cui si desidera stampare la licenza.

LICENSE
Impostare `accept` per accettare le condizioni di licenza IBM MQ Advanced for Developers .

Impostare `view` per visualizzare le condizioni di licenza.

Deprecated **MQ_ADMIN_PASSWORD**
Specificare la password dell'utente admin.

Deve contenere almeno 8 caratteri.

Non esiste alcuna password predefinita per l'utente admin.

V 9.4.0 V 9.4.0 Da IBM MQ 9.4.0, questa variabile non è più fornita. Lo YAML di esempio in questo argomento mostra come puoi creare tu stesso questa variabile e proteggerlo con un segreto.

Deprecated **MQ_APP_PASSWORD**
Specificare la password dell'utente dell'app.

Se impostato, questo fa sì che il canale **DEV.APP.SVRCONN** diventi protetto e consenta solo le connessioni che forniscono un ID utente e una password validi.

Deve contenere almeno 8 caratteri.

Non esiste alcuna password predefinita per l'utente dell'applicazione.

V 9.4.0 V 9.4.0 Da IBM MQ 9.4.0, questa variabile non è più fornita. Lo YAML di esempio in questo argomento mostra come puoi creare tu stesso questa variabile e proteggerlo con un segreto.

MQ_DEV
Impostare `false` per interrompere la creazione degli oggetti predefiniti.

METRICA_ABILITATA_MQ
Imposta `true` per generare le metriche Prometheus per il tuo gestore code.

MQ_LOGGING_CONSOLE_SOURCE
Specifica un elenco separato da virgole di origini per i log che sono sottoposti a mirroring nell'ubicazione **stdout** del contenitore.

I valori validi sono `qmgr`, `web` e `mqsc`.

Il valore predefinito è `qmgr, web`.

Il valore facoltativo è `mqsc`. Questa opzione può essere utilizzata per riflettere i contenuti di `autocfgmqsc.LOG` nel log del contenitore.

MQ_LOGGING_FORMATO_CONSOLE
Modifica il formato dei log stampati nell'ubicazione **stdout** del contenitore.

Impostare `base` per utilizzare un formato leggibile semplice. Questo è il valore predefinito.

Imposta `json` per utilizzare un formato JSON (un oggetto JSON su ogni riga).

ID_ESCLUDI_CONSOLE_LOGGING_MQ_
Specificare un elenco separato da virgole di ID messaggio per i messaggi di log esclusi.

I messaggi di log vengono ancora visualizzati nel file di log sul disco, ma non vengono stampati nell'ubicazione **stdout** del contenitore.

Il valore di default è `AMQ5041I,AMQ5052I,AMQ5051I,AMQ5037I,AMQ5975I`.

MQ_QMGR_NAME
Impostare il nome con cui si desidera creare il gestore code.

Per ulteriori informazioni sulla configurazione dello sviluppatore predefinito supportata dall'immagine IBM MQ Advanced for Developers , vedi la documentazione della configurazione dello sviluppatore predefinito.

YAML del gestore code di esempio che descrive come specificare le password per gli utenti admin e app

Per gli utenti degli ID utente **admin** e **app** , è necessario fornire le password durante la distribuzione di un gestore code utilizzando la licenza `Development` . Di seguito è riportato un esempio di YAML del gestore code che mostra come eseguire questa operazione con IBM MQ Operator.

Il comando riportato di seguito crea un segreto contenente le password per gli utenti **admin** e **app** .

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

Il seguente YAML utilizza tali password quando si distribuisce un gestore code.

```
apiVersion: mq.ibm.com/v1beta1
kind: QueueManager
metadata:
  name: qm-dev
spec:
  license:
    accept: false
    license: L-CLXQ-ADXTK3
    use: Development
  web:
    enabled: true
  template:
    pod:
      containers:
        - env:
            - name: MQ_DEV
              value: "true"
            - name: MQ_CONNAUTH_USE_HTP
              value: "true"
            - name: MQ_ADMIN_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: my-mq-dev-passwords
                  key: dev-admin-password
            - name: MQ_APP_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: my-mq-dev-passwords
                  key: dev-app-password
          name: qmgr
  queueManager:
    storage:
```

```
      queueManager:
        type: persistent-claim
    name: QUICKSTART
  version: 9.4.0.0-r1
```

Risoluzione dei problemi di IBM MQ nei contenitori

Se stai avendo problemi con l'esecuzione di IBM MQ in un contenitore, puoi utilizzare le tecniche qui descritte per aiutarti a diagnosticare e risolvere i problemi.

Procedura

- “Risoluzione dei problemi di riavvii non pianificati di IBM MQ nei contenitori” a pagina 169.
- “Risoluzione dei problemi con IBM MQ Operator” a pagina 170.

OpenShift CP4I Kubernetes Risoluzione dei problemi di riavvii non pianificati di IBM MQ nei contenitori

Nella maggior parte dei sistemi di gestione dei contenitori come Red Hat OpenShift Container Platform e Kubernetes, i contenitori vengono comunemente riavviati. Non è normale che un contenitore sia di lunga durata. Questo argomento illustra il ciclo di vita del contenitore, il modo in cui è possibile esaminare un riavvio e le cause di un riavvio del contenitore non pianificato.

Se non sono stati riscontrati problemi con la propria distribuzione IBM MQ e questa continua ad essere eseguita come previsto, è probabile che la soluzione stia funzionando come previsto. È possibile che nel log del contenitore venga visualizzato un messaggio simile al seguente:

```
Signal received: terminated
```

Ciò significa che il segnale SIGTERM è stato inviato al contenitore MQ , chiedendogli di terminare. I contenitori Linux hanno la responsabilità di rispondere ai segnali POSIX , che sono messaggi standard inviati a un programma per attivare il comportamento.

Quando il contenitore IBM MQ riceve un segnale SIGTERM, emette un comando `endmqm -w -r -tp` per arrestare il gestore code. Una volta arrestato il gestore code, il contenitore verrà arrestato. Se l'arresto del gestore code richiede molto tempo, è possibile che venga inviato un segnale SIGKILL, che terminerà immediatamente i processi Linux . La quantità di tempo tra SIGTERM e SIGKILL è nota come "periodo di tolleranza di terminazione" in Kubernetesed è configurabile sulla risorsa `QueueManager` (se si utilizza IBM MQ Operator) o direttamente sulla risorsa Pod. Il valore predefinito è 30 secondi, di cui un secondo è riservato per l'arresto del contenitore e il resto viene assegnato a IBM MQ. Ad esempio, nel caso predefinito, viene emesso un `endmqm -w -tp 29` , che indica al gestore code di avere 29 secondi per l'arresto.

Motivi per lo sfratto del pod

Il segnale SIGTERM viene utilizzato da Kubernetes (e quindi Red Hat OpenShift Container Platform) per terminare correttamente un pod. Vedi Terminazione dei pod nella documentazione di Kubernetes . Kubernetes utilizza i termini "Pod Disruption" e "eviction" per il processo con cui i pod sui nodi vengono terminati volontariamente o involontariamente. Ci sono molte ragioni per cui un pod potrebbe essere sfrattato, tra cui:

- **Risoluzione da kubelet**. Ciò può essere dovuto a una serie di motivi, tra cui:
  - Il pod può essere terminato perché il Nodo è in fase di arresto (forse come parte di un aggiornamento cluster continuo)
  - Il pod può essere terminato a causa della "pressione" del nodo (dove il kubelet termina proattivamente i pod per recuperare le risorse su un nodo). L'amministratore del cluster Kubernetes può configurare le soglie di eliminazione che potrebbero variare tra i cluster.

  - Il pod può essere terminato perché il pod non ha eseguito correttamente la relativa analisi di attività. Un'analisi attività può essere configurata in Kubernetes per verificare che un pod sia ancora integro. IBM MQ Operator imposta un probe di attività del gestore code che richiama il comando **dspmq** per verificare uno stato di esecuzione valido. Se il gestore code non è in uno stato di integrità o se l'esecuzione del probe richiede troppo tempo, il kubelet considererà questo un errore. Le soglie per il numero di errori da tollerare sono configurabili, sulla risorsa `QueueManager` (se si utilizza IBM MQ Operator) o direttamente sulla risorsa Pod.
- **Prevenzione mediante lo scheduler Kubernetes**. Ciò può verificarsi se il programma di pianificazione Kubernetes deve eseguire un pod con priorità più elevata
- **Nodo tainted** Un nodo può essere "contaminato" e i pod che non tollerano il taint vengono sfrattati. I taint vengono utilizzati dagli amministratori Kubernetes per "respingere" i pod da nodi specifici. Ad esempio, per dire che i pod IBM MQ non devono più essere eseguiti su nodi che hanno hardware speciale che ora è riservato ad altri carichi di lavoro.
- **Richiedi tramite l'API di eliminazione**. Questo può essere richiamato da un amministratore per eliminare i pod
- **Pod garbage collection**. Ciò può verificarsi se il Nodo non è più in servizio o viene rimosso tramite l'API Kubernetes .

Determinazione del motivo per cui un pod del gestore code è stato eliminato

Le potenziali fonti di informazioni che aiutano a capire perché un pod è stato sfrattato includono:

- **Eventi cluster**. Ad esempio, Visualizzazione delle informazioni sull'evento di sistema in un cluster OpenShift Container Platform .
- **Eventi di verifica cluster**. Consultare Visualizzazione dei log di controllo in Red Hat OpenShift Container Platform.
- **Nodi sotto pressione** Ricercare i nodi sotto la pressione della CPU, della rete o della memoria. È possibile visualizzarlo nello stato Nodo. Si noti che nel momento in cui si arriva a guardare, il nodo potrebbe non essere più sotto pressione.
- **Red Hat OpenShift Container Platform Monitoraggio** o altre metriche di monitoraggio potrebbero essere in grado di visualizzare elementi come problemi di latenza del disco. Un'utile metrica Prometheus è `ibmmq_qmgr_log_write_latency_seconds`. Queste informazioni provengono dagli argomenti relativi alle statistiche di MQ .

Informazioni correlate

Documentazione Kubernetes sulla pianificazione, la prevenzione e l'eliminazione

OpenShift CP4I Risoluzione dei problemi con IBM MQ Operator

Se si stanno riscontrando dei problemi con IBM MQ Operator, utilizzare le tecniche descritte per facilitarne la diagnostica e la soluzione.

Procedura

- “Raccolta delle informazioni per la risoluzione dei problemi per i gestori code distribuiti con IBM MQ Operator” a pagina 170
- “Risoluzione dei problemi: accesso ai dati del gestore code” a pagina 172

OpenShift CP4I Raccolta delle informazioni per la risoluzione dei problemi per i gestori code distribuiti con IBM MQ Operator

Raccolta delle informazioni per la risoluzione dei problemi che devono essere fornite al supporto IBM quando si genera un nuovo caso di supporto.

Procedura

1. Raccogliere le informazioni sul provider cloud.

   Questo è il provider cloud che ospita il tuo cluster Red Hat OpenShift (ad esempio, IBM Cloud).
2. Raccogliere le informazioni sull'architettura.

   L'architettura del cluster Red Hat OpenShift è una delle seguenti:

   - Linux for x86-64
   - Linux on Power Systems (ppc64le)
   - Linux for IBM Z
3. Raccogliere le informazioni di distribuzione IBM MQ .

   a) Accedi al tuo cluster Red Hat OpenShift , utilizzando una shell `bash/zsh` .

   b) Impostare le seguenti variabili di ambiente:

   ```
   export QM=QueueManager_name
   export QM_NAMESPACE=QueueManager_namespace
   export MQ_OPERATOR_NAMESPACE=mq_operator_namespace
   ```

   dove *QueueManager_name* è il nome della risorsa QueueManager , *QueueManager_namespace* è lo spazio dei nomi in cui è distribuito e *mq_operator_namespace* è lo spazio dei nomi in cui è distribuito IBM MQ Operator . Potrebbe essere uguale allo spazio dei nomi QueueManager .

   c) Immettere i seguenti comandi e fornire tutti i file di output risultanti al supporto IBM .

   ```
   # OCP / Kubernetes: Version
   oc version -o yaml > ocversion.yaml

   # QueueManager: YAML
   oc get qmgr $QM -n $QM_NAMESPACE -o yaml > "queue-manager-$QM.yaml"

   # MQ Queue Manager: Pods
   oc get pods -n $QM_NAMESPACE -o wide --selector "app.kubernetes.io/instance=$QM" > "qm-
   pods-$QM.txt"

   # MQ Queue Manager: Pod YAML
   oc get pods -n $QM_NAMESPACE -o yaml --selector "app.kubernetes.io/instance=$QM" > "qm-
   pods-$QM.yaml"

   # MQ Queue Manager: Pod Logs
   for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/
   instance=$QM" | cut -d ' ' -f 1); do oc logs -n $QM_NAMESPACE --previous "$p" > "qm-logs-
   previous-$p.txt"; oc logs -n $QM_NAMESPACE $p > "qm-logs-$p.txt";done

   # MQ Queue Manager: Describe Pods
   for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/
   instance=$QM" | cut -d ' ' -f 1); do oc describe pod $p -n $QM_NAMESPACE > "qm-pod-
   describe-$p.txt"; done

   # MQ Web UI: Console Log
   for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/
   instance=$QM" | cut -d ' ' -f 1); do oc cp -n $QM_NAMESPACE --retries=10 "$p:var/mqm/web/
   installations/Installation1/servers/mqweb/logs/console.log" "web-$p-console.log"; done

   # MQ Web UI: Messages Log
   for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/
   instance=$QM" | cut -d ' ' -f 1); do oc cp -n $QM_NAMESPACE --retries=10 "$p:var/mqm/web/
   installations/Installation1/servers/mqweb/logs/messages.log" "web-$p-messages.log"; done

   # MQ Queue Manager: routes defined by operator
   oc get routes -n $QM_NAMESPACE -o yaml --selector "app.kubernetes.io/instance=$QM" > "qm-
   routes-$QM.yaml"

   # MQ Queue Manager: routes to QM
   oc get routes -n $QM_NAMESPACE -o yaml --field-selector "spec.to.name=$QM-ibm-mq" > "qm-
   routes2-$QM.yaml"

   # MQ Queue Manager: stateful set
   oc get statefulset -n $QM_NAMESPACE -o yaml ${QM}-ibm-mq > "qm-statefulset-$QM.yaml"

   # MQ Queue Manager: revisions of the stateful set
   oc get controllerrevisions.apps -o yaml -n $QM_NAMESPACE --selector "app.kubernetes.io/
   ```

```
instance=$QM" > "qm-statefulset-revisions-$QM.yaml"

# MQ Queue Manager: Pod events
for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/
instance=$QM" | cut -d ' ' -f 1); do oc get -o custom-columns="LAST
SEEN:.lastTimestamp,TYPE:.type,REASON:.reason,KIND:.involvedObject.kind,NAME:.involvedObje
ct.name,MESSAGE:.message" event -n $QM_NAMESPACE --field-selector
involvedObject.name="$p" > "qm-pod-events-$p.txt"; done

# MQ Queue Manager: StatefulSet events
oc get events -n $QM_NAMESPACE -o custom-columns="LAST
SEEN:.lastTimestamp,TYPE:.type,REASON:.reason,KIND:.involvedObject.kind,NAME:.involvedObje
ct.name,MESSAGE:.message" --field-selector involvedObject.name="${QM}-ibm-mq" > "qm-
statefulset-events-$QM.txt"

# MQ Queue Manager: services
oc get services -n $QM_NAMESPACE -o yaml --selector "app.kubernetes.io/instance=$QM" >
"qm-services-$QM.yaml"

# MQ Queue Manager: PVCs
oc get pvc -n $QM_NAMESPACE -o yaml --selector "app.kubernetes.io/instance=$QM" > "qm-
pvcs-$QM.yaml"

# MQ Operator: Version
oc get csv -n $QM_NAMESPACE | grep "^ibm-mq\|NAME" > mq-operator-csv.txt

# Cloud Pak Foundational Services: Version
oc get csv -n $QM_NAMESPACE | grep "^ibm-common-service-operator\|NAME" > common-services-
csv.txt

# Cloud Pak for Integration: Version (if applicable)
oc get csv -n $QM_NAMESPACE | grep "^ibm-integration-platform-navigator\|NAME" > cp4i-
csv.txt

# Output from runmqras (this may take a while to execute)
for p in $(oc get pods -n $QM_NAMESPACE --no-headers --selector "app.kubernetes.io/
instance=$QM" | cut -d ' ' -f 1); do timestamp=$(TZ=UTC date +"%Y%m%d_%H%M%S"); oc exec
-n $QM_NAMESPACE $p -- runmqras -workdirectory "/tmp/runmqras_$timestamp" -section
logger,mqweb,nativeha,trace; oc cp -n $QM_NAMESPACE --retries=10 "$p:tmp/
runmqras_$timestamp/" .; done

# MQ Operator: Pod Log
oc logs -n $MQ_OPERATOR_NAMESPACE $(oc get pods -n $MQ_OPERATOR_NAMESPACE --no-headers --
selector app.kubernetes.io/name=ibm-mq,app.kubernetes.io/managed-by=olm | cut -d ' ' -f
1) > mq-operator-log.txt
```

Nota:

La maggior parte di questi comandi richiede l'accesso allo spazio dei nomi in cui viene distribuito il gestore code. Tuttavia, la raccolta del log IBM MQ Operator potrebbe richiedere anche l'accesso **amministratore del cluster** se IBM MQ Operator è installato **nell'ambito del cluster**.

Attività correlate

Raccolta di informazioni per la risoluzione dei problemi per IBM Support

OpenShift CP4I Risoluzione dei problemi: accesso ai dati del gestore code

Utilizza lo strumento di controllo PVC per ottenere l'accesso ai file su una PVC del gestore code in cui non può essere stabilita una shell remota per il pod del gestore code. Ciò potrebbe essere dovuto al fatto che il pod è in uno stato **`Error`** o **`CrashLoopBackOff`** . Questo strumento è progettato per essere utilizzato con i gestori code distribuiti da IBM MQ Operator.

Prima di iniziare

Per utilizzare lo strumento di controllo PVC. è necessario avere accesso al proprio spazio nomi del gestore code.

Informazioni su questa attività

Per risolvere i problemi, è possibile accedere ai dati memorizzati nelle PVC (Persistent Volume Claims) associate a un determinato gestore code. Per fare ciò, utilizza uno strumento per montare le PVC in un

insieme di pod inspector. Puoi quindi ottenere una shell remota in qualsiasi pod inspector per leggere i file.

A seconda del tipo di distribuzione, vengono creati tra uno e tre pod inspector. I volumi specifici per un determinato pod di un gestore code Native - HA o Multi - Instance sono disponibili sul pod inspector PVC associato. I volumi condivisi sono disponibili su tutti gli ispettori. Il nome del pod inspector contiene il nome del pod del gestore code associato.

Procedura

1. Scarica lo strumento MQ PVC inspector.

   Lo strumento è disponibile qui: https://github.com/ibm-messaging/mq-pvc-tool.
2. Assicurati di essere collegato al cluster.
3. Individuare il nome del gestore code e lo spazio dei nomi in cui è in esecuzione il gestore code.
4. Eseguire lo strumento inspector sul tuo gestore code.
   a) Eseguire il seguente comando, specificando il nome del gestore code e il relativo nome spazio dei nomi.

   ```
   ./pvc-tool.sh queue_manager_name queue_manager_namespace_name
   ```

   b) Una volta completato lo strumento, immetti il seguente comando per visualizzare i pod inspector in fase di creazione.

   ```
   oc get pods
   ```

5. Visualizza i file montati nel pod inspector.
   a) Ogni pod di controllo PVC è associato a un pod del gestore code, quindi potrebbero esserci più pod di controllo. Accedi a uno di questi pod, immettendo il seguente comando:

   ```
   oc rsh pvc-inspector-pod-name
   ```

   L'utente viene collocato nella directory contenente le directory PVC montate.
   b) Elenca le directory PVC immettendo il seguente comando:

   ```
   ls
   ```

   c) Visualizzare un elenco delle PVC immettendo il seguente comando al di fuori della sessione della shell remota:

   ```
   oc get pvc
   ```

   d) Ripulisci i pods creati dallo strumento, immettendo il seguente comando:

   ```
   oc delete pods -l tool=mq-pvc-inspector
   ```

Informazioni particolari

Queste informazioni sono state sviluppate per prodotti e 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. Tuttavia, è responsabilità dell'utente valutare e verificare il funzionamento di qualsiasi prodotto, programma o servizio non 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:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
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
Legal and Intellectual Property Law
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 COMMERCIABILITÀ 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.

Tutti i riferimenti a siti Web non dell'IBM contenuti in questo documento sono forniti solo per consultazione e non rappresenta in alcun modo un'approvazione di tali siti. I materiali reperibili in tali siti Web non fanno parte dei materiali relativi a questo prodotto IBM e l'utilizzo di tali siti è responsabilità dell'utente.

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 poterli illustrare nel modo più completo possibile, gli esempi riportano nomi di persone, 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 IBM 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, in altri paesi o entrambi.

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

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

Questo prodotto include il software sviluppato da Eclipse Project (https://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: