aws

Guida per gli sviluppatori

Amazon MQ

Amazon MQ: Guida per gli sviluppatori

Table of Contents

Che cos'è Amazon MQ? 1
Caratteristiche di Amazon MQ 1
Come posso iniziare a utilizzare Amazon MQ? 2
Come posso fornire feedback ad Amazon MQ? 3
Configurazione 4
Fase 1: prerequisiti 4
Iscriviti per un Account AWS 4
Crea un utente con accesso amministrativo 5
Crea un utente e ottieni le tue AWS credenziali 6
Fase 3: ottenere un codice di esempio pronto per l'uso 8
Fasi successive 9
Guida introduttiva: creazione e connessione a un broker ActiveMQ 10
Creare un broker ActiveMQ 10
Guida introduttiva: creazione e connessione a un broker RabbitMQ 13
Crea un broker RabbitMQ 13
Gestione di un broker 16
Connessione ad Amazon MQ 16
Endpoint di servizio 16
Endpoint del broker 17
Connect ad Amazon MQ utilizzando endpoint Dual-stack (IPv4 e) IPv6 17
Connect ad Amazon MQ tramite AWS PrivateLink 17
Autenticazione e autorizzazione 18
Autenticazione e autorizzazione per Amazon MQ for RabbitMQ 18
Autenticazione e autorizzazione per Amazon MQ for ActiveMQ 20
Aggiornamento della versione del motore 21
Aggiornamento manuale della versione del motore 21
Aggiornamento del tipo di istanza 24
Archiviazione 27
Differenze tra i tipi di storage 28
Configurazione di un broker privato 29
Configurazione di un broker privato in Console di gestione AWS 30
Accesso alla console web del broker Amazon MQ senza accessibilità pubblica 31
Pianificazione della manutenzione del broker 32
Riavvio di un broker 35

Riavviare di un broker Amazon MQ ... 35
Eliminazione di un broker ... 36
Eliminazione di un broker Amazon MQ ... 36
Stati del broker ... 36
Assegnazione di tag ... 37
Aggiungere tag nella console Amazon MQ ... 38
Amazon MQ per ActiveMQ ... 40
Amazon MQ per broker ActiveMQ ... 40
Broker ... 40
Utente ... 43
Implementazione di un broker ... 44
broker a istanza singola ... 44
Broker attivo/in standby ... 45
Rete di broker ... 46
Come funziona una rete di broker? ... 47
Come fa una rete di broker a gestire le credenziali? ... 47
Tra regioni ... 47
Failover dinamico con i connettori di trasporto ... 49
Tipi di istanza ... 50
Configurazioni del broker ... 51
Attributes ... 52
Utilizzo dei file di configurazione Spring XML ... 52
Creazione di una configurazione ... 53
Modifica una revisione della configurazione ... 56
Elementi consentiti ... 58
Attributi consentiti ... 61
Raccolte consentite ... 74
Attributi elemento figlio ... 80
Replica dei dati tra regioni ... 87
Broker primari e di replica ... 87
Creazione di un broker CRDR ... 88
Eliminazione di un broker CRDR ... 92
Promozione di un broker CRDR ... 93
Metriche ... 95
Tutorial di ActiveMQ ... 97
Creazione e configurazione di una rete di broker ... 98

Connessione di un'applicazione Java alil broker 103
Integrazione dei broker ActiveMQ con LDAP 109
Fase 3: (Opzionale) Connect a una AWS Lambda funzione 124
Creazione di un utente broker ActiveMQ 126
Modifica un utente del broker ActiveMQ 128
Eliminare un utente del broker ActiveMQ 129
Esempi funzionanti di Java 129
Gestione della versione 141
Versioni dei motori supportate su Amazon MQ for ActiveMQ 142
Aggiornamenti della versione del motore 142
Elenco di versioni del motore supportate 143
Best practice di Amazon MQ per ActiveMQ 143
Non modificare né eliminare mai l'interfaccia di rete elastica Amazon MQ 143
Usa sempre il pooling delle connessioni 144
Usa sempre Failover Transport per la connessione a più endpoint del broker 145
Evita l'uso di selettori di messaggi 146
Preferisci destinazioni virtuali ad abbonamenti durevoli 146
Se utilizzi il peering di Amazon VPC, evita i client IPs nell'intervallo CIDR 10.0.0.0/16 146
Disabilita archiviazione e invio simultaneo per code con consumatori lenti 146
Scegli il tipo di istanza broker corretta per il miglior throughput 147
Scegli il tipo di archiviazione del broker corretto per il miglior throughput 148
Configura la rete di broker nel modo corretto 148
Evita riavvi lenti ripristinando transazioni XA preparate 149
Amazon MQ per RabbitMQ 151
Broker 151
Porte del listener 151
Attributes 41
Gestione della versione 152
Elenco di versioni del motore supportate 153
RabbitMQ 4 154
Supporto versione 157
Aggiornamenti di versione 157
Implementazione di un broker RabbitMQ 158
broker a istanza singola 158
Distribuzione del cluster 159
Tipi di istanza 161

Tipi di istanze per la distribuzione di cluster m7g 162
Tipi di istanze per la distribuzione a singola istanza di m7g 163
Tipi di istanze per la distribuzione a mq.m5 singola istanza 164
Tipi di istanze per la distribuzione in cluster mq.m5 165
Linee guida per il dimensionamento 166
Limiti di risorse predefiniti 167
Limite massimo di risorse 171
Impostazioni predefinite del broker 176
Configurazioni del broker 181
Attributes 52
Creazione di una configurazione 182
Modifica di una revisione della configurazione 185
Valori configurabili 186
Autenticazione e autorizzazione 202
Autenticazione e autorizzazione semplici 19
OAuth Autenticazione e autorizzazione 2.0 19
Autenticazione e autorizzazione IAM 19
Autenticazione e autorizzazione LDAP 19
Autenticazione e autorizzazione HTTP 19
Autenticazione con certificato SSL 20
Autenticazione e autorizzazione semplici 204
OAuth autenticazione e autorizzazione 2.0 206
Autenticazione e autorizzazione IAM 207
Autenticazione e autorizzazione HTTP 209
Autenticazione con certificato SSL 211
Autenticazione e autorizzazione LDAP 214
Plugin 217
Plugin di gestione RabbitMQ 218
Plugin Shovel 218
Plugin federativo 219
Plugin scambio di hash coerente 220
OAuth Plugin 2.0 220
Plugin LDAP 221
Plugin HTTP 221
Plugin per certificati SSL 221
plugin aws 222

Plugin JMS Topic Exchange ... 222
Protocolli ... 222
Supporto JMS ... 222
Client RabbitMQ JMS ... 223
JMS 1.1, 2.0 e 3.1 supportati APIs ... 223
Autenticazione e autorizzazione ... 223
Interoperabilità con le code AMQP su RabbitMQ ... 223
Policy ... 224
Code per il quorum ... 229
Migrazione alle code quorum ... 230
Configurazione delle politiche ... 231
Best practice ... 232
Best practice di Amazon MQ per RabbitMQ ... 232
Configurazione del broker ... 233
Affidabilità dei messaggi ... 235
Ottimizzazione delle prestazioni ... 238
Resilienza della rete ... 243
Tutorial RabbitMQ ... 245
Modifica delle preferenze del broker ... 245
Utilizzo di Python Pika con Amazon MQ per RabbitMQ ... 246
Risoluzione della sincronizzazione della coda sospesa ... 253
Riduzione del numero di connessioni e canali ... 260
Fase 2: Connect un'applicazione basata su JVM al broker ... 261
Fase 3: (Opzionale) Connect a una AWS Lambda funzione ... 265
Utilizzo dell'autenticazione e dell'autorizzazione OAuth 2.0 ... 268
Utilizzo dell'autenticazione e dell'autorizzazione IAM ... 276
Utilizzo dell'autenticazione e dell'autorizzazione LDAP ... 281
Utilizzo dell'autenticazione e dell'autorizzazione HTTP ... 287
Utilizzo dell'autenticazione tramite certificato SSL ... 292
Utilizzo di MTL per AMQP e endpoint di gestione ... 298
Connessione dell'applicazione JMS ... 303
Sicurezza ... 307
Protezione dei dati ... 308
Encryption (Crittografia) ... 309
Crittografia dei dati a riposo ... 309
Crittografia dei dati in transito ... 319

Gestione dell'identità e degli accessi ... 320
Destinatari ... 321
Autenticazione con identità ... 321
Gestione dell'accesso tramite policy ... 322
Funzionamento di Amazon MQ con IAM ... 324
Esempi di policy basate su identità ... 330
Autenticazione e autorizzazione API ... 333
Autenticazione e autorizzazione del broker ... 338
AWS politiche gestite ... 340
Uso di ruoli collegati ai servizi ... 342
Risoluzione dei problemi ... 348
Convalida della conformità ... 350
Resilienza ... 351
Sicurezza dell'infrastruttura ... 351
Best practice di sicurezza ... 351
Preferire broker senza accesso pubblico ... 352
Configurare sempre una mappa di autorizzazione ... 352
Blocca i protocolli non necessari ... 352
Registrazione di log e monitoraggio ... 354
Accesso ai parametri CloudWatch ... 354
Accesso alle CloudWatch metriche utilizzando il Console di gestione AWS ... 355
Metriche per ActiveMQ ... 355
Parametri Amazon MQ per ActiveMQ ... 355
Destinazione ActiveMQ per i parametri (coda e argomento) ... 361
Metriche per RabbitMQ ... 365
Parametri del broker RabbitMQ ... 365
Dimensioni per i parametri del broker RabbitMQ ... 369
Parametri del nodo RabbitMQ ... 369
Le dimensioni per i parametri dei nodi RabbitMQ ... 370
Parametri della coda RabbitMQ ... 371
Dimensioni per i parametri della coda RabbitMQ ... 371
Metriche di rete RabbitMQ ... 372
Dimensioni per i broker RabbitMQ ... 373
Configurazione dei log di Amazon MQ per RabbitMQ ... 373
Registrazione delle chiamate API utilizzando CloudTrail ... 373
Informazioni su Amazon MQ in CloudTrail ... 374

Esempio: voci del file di log di Amazon MQ ... 376
Configurazione dei log di Amazon MQ per ActiveMQ ... 378
Comprensione della struttura di registrazione nei log CloudWatch ... 379
Aggiunta dell'autorizzazione `CreateLogGroup` all'utente Amazon MQ ... 379
Configurare una policy basata sulle risorse per Amazon MQ ... 380
Prevenzione del confused deputy tra servizi ... 382
Risoluzione dei problemi ... 384
I gruppi di log non vengono visualizzati in CloudWatch ... 384
I flussi di log non vengono visualizzati nei gruppi di CloudWatch log ... 384
Quote ... 385
Broker ... 385
Configurazioni ... 386
Utenti ... 387
Storage dei dati ... 388
Throttling delle API ... 389
Risoluzione dei problemi ... 390
Risoluzione dei problemi di ActiveMQ su Amazon MQ ... 390
Risoluzione dei problemi di RabbitMQ su Amazon MQ ... 390
Risoluzione dei problemi: Amazon MQ generale ... 393
Non riesco a connettermi alla console Web o agli endpoint del broker. ... 393
Eccezioni per SSL ... 399
Ho creato un broker ma la creazione non è riuscita. ... 400
Il mio broker si è riavviato e non sono sicuro del motivo. ... 400
Risoluzione dei problemi di ActiveMQ su Amazon MQ ... 401
Recupero dei log CloudWatch ... 401
Connessione al broker dopo un riavvio ... 402
Alcuni client non riescono a connettersi ... 402
Eccezione `JSP` sulla console web ... 403
Risoluzione dei problemi: RabbitMQ su Amazon MQ ... 404
Non riesco a visualizzare le metriche relative alle mie code o ai miei host virtuali. CloudWatch ... 404
Come posso abilitare i plugin in RabbitMQ su Amazon MQ? ... 404
Non riesco a modificare la configurazione di Amazon VPC per il broker. ... 404
Le implementazioni dei cluster hanno messo in pausa le sincronizzazioni delle mie code. ... 405
Il mio broker a istanza singola Amazon MQ for RabbitMQ è in un ciclo di riavvio. ... 405
Ho perso l'accesso a tutti gli account di amministratore sul mio broker. ... 405

BROKER_ENI_DELETED 406
BROKER_OOM 406
RABBITMQ_MEMORY_ALARM 408
Fase 1: Diagnostica di un allarme con memoria elevata 408
Fase 2: Risolve e previene l'allarme di memoria esaurita 411
RABBITMQ_INVALID_KMS_KEY 412
Diagnosi e risoluzione di INVALID_KMS_KEY 413
RABBITMQ_DISK_ALARM 413
Diagnosi e risoluzione dell'allarme relativo al limite del disco 414
RABBITMQ_CLUSTER_DISK_USAGE_TOO_HIGH_FOR_INSTANCE_CHANGE 415
Diagnosi e risoluzione dell'allarme di modifica del tipo di istanza 415
RABBITMQ_INVALID_ASSUMEROLE 416
Diagnosi e risoluzione di RABBITMQ_INVALID_ASSUMEROLE 416
RABBITMQ_INVALID_ARN_LDAP 417
Diagnosi e indirizzamento di RABBITMQ_INVALID_ARN_LDAP 417
RABBITMQ_INVALID_ARN_HTTP 418
Diagnosi e indirizzamento di RABBITMQ_INVALID_ARN_HTTP 419
RABBITMQ_INVALID_ARN_SSL 420
Diagnosi e indirizzamento di RABBITMQ_INVALID_ARN_SSL 420
RABBITMQ_INVALID_ARN 421
Diagnosi e indirizzamento di RABBITMQ_INVALID_ARN 421
Risorse correlate 423
Risorse di Amazon MQ 423
Amazon MQ per risorse ActiveMQ 424
Amazon MQ per risorse RabbitMQ 424
Note di rilascio 426
..... cdlxvi

Che cos'è Amazon MQ?

Amazon MQ è un servizio di broker di messaggi gestito per Apache ActiveMQ Classic e RabbitMQ che gestisce la configurazione, il funzionamento e la manutenzione dei broker di messaggi. Puoi creare un nuovo broker Amazon MQ utilizzando protocolli di messaggistica standard del settore o migrare i broker di messaggi esistenti su Amazon MQ senza riscrivere il codice di messaggistica.

Un broker è un ambiente broker dei messaggi in esecuzione su Amazon MQ. Costituisce l'elemento di base di Amazon MQ. Un broker di messaggistica consente alle applicazioni e ai componenti software di comunicare utilizzando diversi linguaggi di programmazione, sistemi operativi e protocolli di messaggistica formale. Puoi utilizzare i broker Amazon MQ per la comunicazione tra applicazioni e componenti nativi del cloud su larga scala.

Argomenti

- Caratteristiche di Amazon MQ
- Come posso iniziare a utilizzare Amazon MQ?
- Come posso fornire feedback ad Amazon MQ?

Caratteristiche di Amazon MQ

Manutenzione gestita e aggiornamenti delle versioni

Amazon MQ esegue la manutenzione e gli aggiornamenti di versione per un broker di messaggi durante la finestra di manutenzione pianificata.

Monitora i broker con CloudWatch

Amazon MQ è integrato con Amazon CloudWatch in modo da poter visualizzare e analizzare i parametri per i broker e le code. Puoi visualizzare e analizzare i parametri dalla console Amazon MQ, dalla console, dalla CloudWatch riga di comando e dall'API. Le metriche vengono raccolte automaticamente e aggiornate ogni minuto. CloudWatch

Sicurezza

Amazon MQ fornisce la crittografia dei messaggi inattivi e in transito. Le connessioni al broker utilizzano SSL e l'accesso può essere limitato a un endpoint privato all'interno del tuo Amazon VPC.

Inoltre, puoi usare AWS Identity and Access Management(IAM) per controllare le azioni che i tuoi utenti e gruppi IAM possono intraprendere su broker Amazon MQ specifici.

Code quorum per RabbitMQ su Amazon MQ

Le code quorum sono un tipo di coda replicato composto da un nodo leader (replica primaria) e nodi follower (altre repliche). Ogni nodo si trova in una zona di disponibilità diversa, quindi se un nodo è temporaneamente non disponibile, la consegna dei messaggi continua con una replica del leader appena eletto in un'altra zona di disponibilità. Le code quorum sono utili per gestire i messaggi avvelenati, che si verificano quando un messaggio fallisce e viene richiesto più volte.

Replica dei dati tra regioni per ActiveMQ su Amazon MQ

La replica dei dati tra regioni (CRDR) consente la replica asincrona dei messaggi dal broker principale in una regione primaria al broker di replica in una regione di replica. AWS Inviando una richiesta di failover all'API Amazon MQ, l'attuale broker di replica viene promosso al ruolo di broker primario e l'attuale broker primario viene retrocesso al ruolo di replica.

Come posso iniziare a utilizzare Amazon MQ?

Per iniziare a usare ActiveMQ su Amazon MQ, consulta la seguente documentazione:

- Guida introduttiva: creazione e connessione a un broker ActiveMQ
- the section called “Implementazione di un broker”
- Tutorial ActiveMQ
- the section called “Best practice di Amazon MQ per ActiveMQ”

Per iniziare a usare RabbitMQ su Amazon MQ, consulta la seguente documentazione:

- Guida introduttiva: creazione e connessione a un broker RabbitMQ
- the section called “Implementazione di un broker RabbitMQ”
- the section called “Tutorial RabbitMQ”
- the section called “Best practice di Amazon MQ per RabbitMQ”

Per ulteriori informazioni su Amazon MQ REST APIs, consulta l'Amazon MQ REST API Reference.

Per ulteriori informazioni sui AWS CLI comandi Amazon MQ, consulta Amazon MQ nel AWS CLI Command Reference.

Come posso fornire feedback ad Amazon MQ?

Accogliamo con favore e incoraggiamo il tuo feedback sulla documentazione. Puoi utilizzare le icone con il pollice su e il pollice giù sul lato destro per inviare feedback, oppure puoi utilizzare il modulo «Fornisci feedback» collegato di seguito.

Per contattare il team di Amazon MQ, utilizza il forum di discussione di Amazon MQ.

Configurazione di Amazon MQ

Prima di poter utilizzare Amazon MQ, devi completare le fasi seguenti.

Argomenti

- Fase 1: prerequisiti
- Passaggio 2: crea un utente e ottieni AWS le tue credenziali
- Fase 3: ottenere un codice di esempio pronto per l'uso
- Fasi successive

Fase 1: prerequisiti

Iscriviti per un Account AWS

Se non ne hai uno Account AWS, completa i seguenti passaggi per crearne uno.

Per iscriverti a un Account AWS

1. Apri la https://portal.aws.amazon.com/billing/registrazione.
2. Segui le istruzioni online.

   Nel corso della procedura di registrazione riceverai una telefonata o un messaggio di testo e ti verrà chiesto di inserire un codice di verifica attraverso la tastiera del telefono.

   Quando ti iscrivi a un Account AWS, Utente root dell'account AWSviene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come best practice di sicurezza, assegna l'accesso amministrativo a un utente e utilizza solo l'utente root per eseguire attività che richiedono l'accesso di un utente root.

AWS ti invia un'email di conferma dopo il completamento della procedura di registrazione. In qualsiasi momento, puoi visualizzare l'attività corrente del tuo account e gestirlo accedendo a https://aws.amazon.com/e scegliendo Il mio account.

Crea un utente con accesso amministrativo

Dopo esserti registrato Account AWS, proteggi Utente root dell'account AWS AWS IAM Identity Center, abilita e crea un utente amministrativo in modo da non utilizzare l'utente root per le attività quotidiane.

Proteggi i tuoi Utente root dell'account AWS

1. Accedi Console di gestione AWScome proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

   Per informazioni sull'accesso utilizzando un utente root, consulta la pagina Accedere come utente root nella Guida per l'utente di Accedi ad AWS .
2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

   Per istruzioni, consulta Abilitare un dispositivo MFA virtuale per l'utente Account AWS root (console) nella Guida per l'utente IAM.

Crea un utente con accesso amministrativo

1. Abilita il Centro identità IAM.

   Per istruzioni, consulta Abilitazione del AWS IAM Identity Center nella Guida per l'utente di AWS IAM Identity Center .
2. Nel Centro identità IAM, assegna l'accesso amministrativo a un utente.

   Per un tutorial sull'utilizzo di IAM Identity Center directory come fonte di identità, consulta Configurare l'accesso utente con l'impostazione predefinita IAM Identity Center directory nella Guida per l'AWS IAM Identity Center utente.

Accesso come utente amministratore

- Per accedere come utente del Centro identità IAM, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente del Centro identità IAM.

  Per informazioni sull'accesso utilizzando un utente IAM Identity Center, consulta AWS Accedere al portale di accesso nella Guida per l'Accedi ad AWS utente.

Assegnazione dell'accesso ad altri utenti

1. Nel Centro identità IAM, crea un set di autorizzazioni conforme alla best practice per l'applicazione di autorizzazioni con il privilegio minimo.

   Segui le istruzioni riportate nella pagina Creazione di un set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .
2. Assegna al gruppo prima gli utenti e poi l'accesso con autenticazione unica (Single Sign-On).

   Per istruzioni, consulta Aggiungere gruppi nella Guida per l'utente di AWS IAM Identity Center .

Passaggio 2: crea un utente e ottieni AWS le tue credenziali

Gli utenti necessitano di un accesso programmatico se desiderano interagire con l' AWS esterno di. Console di gestione AWS Il modo per concedere l'accesso programmatico dipende dal tipo di utente che accede. AWS

Per fornire agli utenti l'accesso programmatico, scegli una delle seguenti opzioni.

Quale utente necessita dell'accesso programmatico?	Per	Come
IAM	(Consigliato) Utilizza le credenziali della console come credenziali temporanee per firmare le richieste programmatiche a,, o. AWS CLI AWS SDKs AWS APIs	Segui le istruzioni per l'interfaccia che desideri utilizzare. • Per la AWS CLI, consulta Login for AWS local development nella Guida per l'AWS Command Line Interface utente. • Per AWS SDKs, consulta Login for AWS local development nella AWS SDKs and Tools Reference Guide.
Identità della forza lavoro	Utilizza credenziali temporanee per firmare le richieste	Segui le istruzioni per l'interfaccia che desideri utilizzare.

Quale utente necessita dell'accesso programmatico?	Per	Come
(Utenti gestiti nel centro identità IAM)	programmatiche a AWS CLI, AWS SDKs, o. AWS APIs	• Per la AWS CLI, vedere Configurazione dell'uso AWS IAM Identity Center nella AWS CLI Guida per l'utente.AWS Command Line Interface • Per AWS SDKs gli strumenti e AWS APIs, consulta l'autenticazione di IAM Identity Center nella Guida di riferimento AWS SDKs and Tools.
IAM	Utilizza credenziali temporane e per firmare le richieste programmatiche a AWS CLI, AWS SDKs, o. AWS APIs	Seguendo le istruzioni riportate in Utilizzo delle credenziali temporanee con le AWS risorse nella Guida per l'utente IAM.

Quale utente necessita dell'accesso programmatico?	Per	Come
IAM	(Non consigliato) Utilizza credenziali a lungo termine per firmare richieste programmatiche a AWS CLI,, AWS SDKs o. AWS APIs	Segui le istruzioni per l'interfaccia che desideri utilizzare. • Per la AWS CLI, consulta Autenticazione tramite credenziali utente IAM nella Guida per l'utente.AWS Command Line Interface • Per gli strumenti AWS SDKs e gli strumenti, consulta Autenticazione tramite credenziali a lungo termine nella Guida di riferimento agli strumenti e agli AWS SDKs strumenti. • Per AWS APIs, consulta la sezione Gestione delle chiavi di accesso per gli utenti IAM nella Guida per l'utente IAM.

Fase 3: ottenere un codice di esempio pronto per l'uso

I seguenti tutorial mostrano come lavorare con i broker Amazon MQ che utilizzano e Console di gestione AWS come connettersi ai broker Amazon MQ for ActiveMQ e Amazon MQ for RabbitMQ in modo programmatico. Per utilizzare il codice di esempio ActiveMQ Java, devi installare Java Standard Edition Development Kit e apportare alcune modifiche di configurazione al codice di esempio.

Puoi anche creare e gestire broker in modo programmatico utilizzando l'API REST di Amazon MQ e. AWS SDKs

Fasi successive

Ora che sei pronto a utilizzare Amazon MQ, inizia con la creazione di un broker. A seconda del tipo di motore del broker, è possibile quindi connettere un'applicazione Java alil broker Amazon MQ per ActiveMQ o utilizzare la libreria client Java RabbitMQ per connettere un'applicazione basata su JVM alil broker Amazon MQ per RabbitMQ.

Guida introduttiva: creazione e connessione a un broker ActiveMQ

Un broker è un ambiente broker dei messaggi in esecuzione su Amazon MQ. Costituisce l'elemento di base di Amazon MQ. La descrizione combinata della classe dell'istanza del broker (`m5`) e della dimensione (`large`,`medium`) è denominata tipo di istanza del broker (ad esempio,`mq.m5.large`). Per ulteriori informazioni, consulta Cos'è un broker Amazon MQ for ActiveMQ?.

Creare un broker ActiveMQ

La prima attività di Amazon MQ, nonché la più comune, è la creazione di un broker. L'esempio seguente mostra come è possibile utilizzare il Console di gestione AWS per creare un broker di base.

1. Accedere alla console Amazon MQ.
2. Alla pagina Select broker engine (Seleziona motore del broker), scegliere Apache ActiveMQ.
3. Alla pagina Select deployment and storage (Seleziona implementazione e archiviazione), nella sezione Deployment mode and storage type (Modalità di implementazione e tipo di archiviazione), procedere come segue:

   a. Scegliere il tipo di Deployment mode (Modalità di implementazione) (ad esempio, Active/standby broker (Broker attivo/in standby)). Per ulteriori informazioni, consulta Opzioni di implementazione per i broker Amazon MQ for ActiveMQ.

      - Un broker a istanza singola è composto da un broker in una zona di disponibilità. Il broker comunica con l'applicazione e con un volume di archiviazione Amazon EBS o Amazon EFS. Per ulteriori informazioni, consulta Opzione 1: broker a istanza singola Amazon MQ.
      - Un Broker attivo/in standby per alta disponibilità è composto da due broker in due diverse zone di disponibilità, configurate in una coppia ridondante. Questi broker comunicano in modo sincrono con l'applicazione e con Amazon EFS. Per ulteriori informazioni, consulta Opzione 2: active/standby broker Amazon MQ per un'elevata disponibilità.

   b. Scegliere un'opzione per Storage type (Tipo di archiviazione) (ad esempio, EBS). Per ulteriori informazioni, consulta Storage.

> **Note**
>
> Amazon EBS replica i dati all'interno di una singola zona di disponibilità e non supporta la modalità di implementazione ActiveMQ attiva/in standby.

c. Scegli Next (Successivo).

4. Alla pagina Configure settings (Configura impostazioni), nella sezione Details (Dettagli), procedere come segue:

   a. Inserisci il nome del broker.

> **Important**
>
> Non aggiungere informazioni personali di identificazione (PII) o altre informazioni riservate o sensibili nei nomi dei broker. I nomi dei broker sono accessibili ad altri AWS servizi, inclusi CloudWatch i registri. I nomi dei broker non sono destinati ad essere utilizzati per dati privati o sensibili.

> **Note**
>
> Nella sezione Impostazioni aggiuntive, puoi anche configurare quanto segue:
>
> - Configurazioni
> - CloudWatch logs
> - Accesso privato
> - Finestra di manutenzione del broker

   b. Selezionare il tipo di istanza del broker (ad esempio, mq.m5.large). Per ulteriori informazioni, consulta Broker instance types.

5. Nella sezione ActiveMQ Web Console access (Accesso alla console Web di ActiveMQ), specificare nome utente e password. Per i nomi utente e le password del broker si applicano le seguenti limitazioni:

   - Il nome utente può contenere solo caratteri alfanumerici, punti, trattini e tilde (-, ., _ e ~).

- La password deve contenere almeno 12 caratteri, di cui almeno 4 caratteri univoci, e non deve contenere virgole, due punti o il simbolo dell'uguale (,:=).

> **Important**
>
> Non aggiungere informazioni personali di identificazione (PII) o altre informazioni riservate o sensibili nei nomi utente dei broker. I nomi utente dei broker sono accessibili ad altri AWS servizi, inclusi i CloudWatch registri. I nomi utenti dei broker non sono destinati ad essere utilizzati per dati privati o sensibili.

6. Seleziona Deploy (Implementa).

   Mentre Amazon MQ crea il broker, mostra lo stato Creation in progress (Creazione in corso).

   Per creare lil broker sono necessari circa 15 minuti.

   Quando il broker viene creato correttamente, Amazon MQ mostra lo stato Running (In esecuzione).

7. Scegli ***MyBroker***.

   Nella ***MyBroker***pagina, nella sezione Connect, annota l'URL della console web ActiveMQ del tuo broker, ad esempio:

   ```
   https://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:8162
   ```

   Inoltre, annotare gli endpoint del protocollo a livello di collegamento del broker. Di seguito è riportato un esempio di endpoint: OpenWire

   ```
   ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:61617
   ```

Guida introduttiva: creazione e connessione a un broker RabbitMQ

Un broker è un ambiente broker dei messaggi in esecuzione su Amazon MQ. Costituisce l'elemento di base di Amazon MQ. La descrizione combinata della classe dell'istanza del broker (`m5`) e della dimensione (`large`,`medium`) è chiamata tipo di istanza del broker (ad esempio,`mq.m5.large`). Per ulteriori informazioni, consulta Cos'è un broker Amazon MQ for RabbitMQ?

Crea un broker RabbitMQ

La prima attività di Amazon MQ, nonché la più comune, è la creazione di un broker. L'esempio seguente mostra come utilizzare il Console di gestione AWS per creare un broker di base.

Quando crei un broker Amazon MQ per RabbitMQ, segui le best practice di configurazione del broker per RabbitMQ per massimizzare le prestazioni del broker e ottimizzare l'efficienza del throughput dei messaggi.

1. Accedere alla console Amazon MQ.
2. Alla pagina Select broker engine (Seleziona motore broker), scegliere RabbitMQ e quindi Next (Avanti).
3. Alla pagina Select deployment mode (Seleziona la modalità di implementazione), scegliere Deployment mode (Modalità di implementazione), ad esempio, Cluster deployment (Distribuzione cluster), quindi Next (Successivo).

   - Un broker a istanza singola è composto da un broker in una zona di disponibilità dietro un load balancer di rete. Il broker comunica con l'applicazione e con un volume di archiviazione Amazon EBS. Per ulteriori informazioni, consulta Opzione 1: Amazon MQ per broker a istanza singola RabbitMQ.
   - Un'implementazione del cluster RabbitMQ per alta disponibilità rappresenta un raggruppamento logico di tre nodi del broker RabbitMQ dietro un load balancer di rete, in cui ciascuno condivide utenti, code e uno stato distribuito su più zone di disponibilità. Per ulteriori informazioni, consulta Opzione 2: distribuzione di cluster Amazon MQ per RabbitMQ.
4. Alla pagina Configure settings (Configura impostazioni), nella sezione Details (Dettagli), procedere come segue:

   a. Inserisci il nome del broker.

> **Important**
>
> Non aggiungere informazioni personali di identificazione (PII) o altre informazioni riservate o sensibili nei nomi dei broker. I nomi dei broker sono accessibili ad altri servizi, inclusi i log. AWS CloudWatch I nomi dei broker non sono destinati ad essere utilizzati per dati privati o sensibili.

b. Scegli il tipo di istanza Broker (ad esempio, mq.m7g.large). Per ulteriori informazioni, consulta Broker instance types.

5. Alla pagina Configure settings (Configura impostazioni), nella sezione RabbitMQ access (Accesso RabbitMQ), specificare un Username (Nome utente) e Password. Per le credenziali di accesso del broker si applicano le seguenti limitazioni:

   - Il nome utente può contenere solo caratteri alfanumerici, punti e trattini (-, _). Questo valore non deve contenere caratteri tilde (~). Amazon MQ vieta l'utilizzo di `guest` come nome utente.
   - La password deve contenere almeno 12 caratteri, di cui almeno 4 caratteri univoci, e non deve contenere virgole, due punti o il simbolo dell'uguale (,:=).

   > **Important**
   >
   > Non aggiungere informazioni personali di identificazione (PII) o altre informazioni riservate o sensibili nei nomi utente dei broker. I nomi utente dei broker sono accessibili ad altri servizi, inclusi i registri. AWS CloudWatch I nomi utenti dei broker non sono destinati ad essere utilizzati per dati privati o sensibili.

   > **Note**
   >
   > Nella sezione Impostazioni aggiuntive, puoi anche configurare quanto segue:
   >
   > - Configurazioni
   > - CloudWatch logs
   > - Accesso privato
   > - Finestra di manutenzione del broker

6. Scegli Next (Successivo).
7. Alla pagina Review and create (Rivedi e crea), puoi rivedere le selezioni e modificarle, se necessario.
8. Selezionare Create broker (Crea broker).

   Mentre Amazon MQ crea il broker, mostra lo stato Creation in progress (Creazione in corso).

   Per creare lil broker sono necessari circa 15 minuti.

   Quando il broker viene creato correttamente, Amazon MQ mostra lo stato Running (In esecuzione).
9. Scegli ***MyBroker***.

   Nella ***MyBroker***pagina, nella sezione Connect, annota l'URL della console web RabbitMQ del tuo broker, ad esempio:

   ```
   https://b-c8349341-ec91-4a78-ad9c-a57f23f235bb.mq.us-west-2.on.aws
   ```

   Inoltre, annotare l'endpoint sicuro per AMQP del broker. Di seguito è riportato un esempio di un endpoint `amqps` che espone una porta `5671` del listener.

   ```
   amqps://b-c8349341-ec91-4a78-ad9c-a57f23f235bb.mq.us-west-2.on.aws:5671
   ```

Gestione di un broker Amazon MQ

Dopo aver creato un broker, puoi gestire e mantenere i diversi componenti del tuo broker Amazon MQ.

Argomenti

- Connessione ad Amazon MQ
- Autenticazione e autorizzazione per i broker Amazon MQ
- Aggiornamento di una versione del motore del broker Amazon MQ
- Aggiornamento di un tipo di istanza del broker Amazon MQ
- Amazon MQ per i tipi di storage ActiveMQ
- Configurazione di un broker Amazon MQ privato
- Pianificazione della finestra di manutenzione per un broker Amazon MQ
- Riavvio di un broker Amazon MQ
- Eliminazione di un broker Amazon MQ
- Stati dei broker Amazon MQ
- Aggiungere tag alle risorse Amazon MQ

Connessione ad Amazon MQ

Puoi connetterti ad Amazon MQ da altri AWS servizi utilizzando endpoint di servizio ed endpoint broker.

Endpoint di servizio

I seguenti metodi di connessione vengono utilizzati per l'API del servizio Amazon MQ:

Domini	Metodo di connessione
`mq.region.amazonaws.com`	IPv4
`mq.region.api.aws`	Dual-stack (e) IPv4 IPv6
`mq-fips.region.amazonaws.com`	FIPS con solo IPv4

Domini	Metodo di connessione
`mq-fips.region.api.aws`	FIPS con Dual-stack

Endpoint del broker

I seguenti metodi di connessione vengono utilizzati per i broker Amazon MQ:

Domini	Metodo di connessione
`brokerId.mq.region.amazonaws.com`	IPv4
`brokerId.mq.region.on.aws`	Dual-stack (e) IPv4 IPv6 **Note** I broker Amazon MQ for ActiveMQ non supportano il dual-stack.

Connect ad Amazon MQ utilizzando endpoint Dual-stack (IPv4 e) IPv6

Gli endpoint dual-stack supportano sia il traffico che il traffico. IPv4 IPv6 Quando effettui una richiesta a un endpoint dual-stack, l'URL dell'endpoint si risolve in un indirizzo o in un indirizzo. IPv4 IPv6 Per ulteriori informazioni sugli endpoint dual-stack e FIPS, consulta la guida di riferimento SDK.

Amazon MQ supporta gli endpoint dual-stack regionali, il che significa che è necessario specificare la AWS regione come parte del nome dell'endpoint. I nomi degli endpoint dual-stack utilizzano la seguente convenzione di denominazione:. `mq.region.api.aws` Ad esempio, il nome dell'endpoint dual-stack per la Regione `eu-west-1` è `mq.eu-west-1.api.aws`.

Per l'elenco completo degli endpoint Amazon MQ, consulta la Guida AWS generale.

Connect ad Amazon MQ tramite AWS PrivateLink

AWS PrivateLinkendpoint per l'API Amazon MQ con IPv6 supporto IPv4 e connettività privata tra cloud privati virtuali (VPCs) e l'API Amazon MQ senza esporre il traffico alla rete Internet pubblica.

> **Note**
>
> Il supporto per PrivateLink è disponibile solo per l'endpoint dell'API Amazon MQ, non per l'endpoint del broker. Per ulteriori informazioni sulla connessione privata a un endpoint del broker, consulta. Configuring a private Amazon MQ broker

Per accedere all'API Amazon MQ utilizzando PrivateLink, devi prima creare un endpoint VPC di interfaccia nel VPC specifico da cui desideri connetterti. Quando crei l'endpoint VPC, usa il nome del servizio `com.amazonaws.region.mq` o `com.amazonaws.region.mq-fips` per gli endpoint FIPS.

Quando chiami Amazon MQ utilizzando la AWS CLI o l'SDK, devi specificare l'URL dell'endpoint per utilizzare il nome di dominio dual-stack: o. `mq.region.api.aws` `mq-fips.region.api.aws` PrivateLink per Amazon MQ non supporta il nome di dominio predefinito che termina con`amazonaws.com`. Per ulteriori informazioni, consulta gli endpoint Dual-stack e FIPS nella Guida di riferimento SDK.

Il seguente esempio di CLI mostra come chiamarlo `describe-broker-engine-type` nella regione Asia Pacifico (Sydney) tramite un endpoint VPC Amazon MQ.

```
AWS_USE_DUALSTACK=true aws mq describe-broker-engine-types --region ap-southeast-2
```

Per altri modi di configurare l'endpoint nella CLI, consulta Utilizzo degli endpoint nella CLI AWS

Puoi anche determinare l'accesso degli utenti agli endpoint VPC utilizzando le policy degli endpoint VPC. Per ulteriori informazioni, consulta Controllare l'accesso agli endpoint VPC utilizzando le policy degli endpoint.

Autenticazione e autorizzazione per i broker Amazon MQ

Amazon MQ offre diversi metodi di autenticazione e autorizzazione per proteggere l'infrastruttura di messaggistica in base ai requisiti dell'organizzazione.

Autenticazione e autorizzazione per Amazon MQ for RabbitMQ

Amazon MQ for RabbitMQ supporta i seguenti metodi di autenticazione e autorizzazione:

Autenticazione e autorizzazione semplici

Con questo metodo, gli utenti del broker vengono archiviati internamente nel broker RabbitMQ e gestiti tramite la console web o l'API di gestione. Le autorizzazioni per vhost, exchange, code e topic sono configurate direttamente in RabbitMQ. Questo è il metodo predefinito. Per ulteriori informazioni, vedere Autenticazione e autorizzazione semplici.

OAuth Autenticazione e autorizzazione 2.0

In questo metodo, gli utenti del broker e le relative autorizzazioni sono gestiti da un provider di identità OAuth 2.0 (IdP) esterno. L'autenticazione degli utenti e le autorizzazioni alle risorse per vhost, exchange, code e argomenti sono centralizzate tramite il sistema di ambito del OAuth provider 2.0. Ciò semplifica la gestione degli utenti e consente l'integrazione con i sistemi di identità esistenti. Per ulteriori informazioni, vedere Autenticazione e autorizzazione OAuth 2.0.

Autenticazione e autorizzazione IAM

Con questo metodo, gli utenti del broker si autenticano utilizzando le credenziali AWS IAM tramite la federazione in uscita IAM. Le credenziali IAM vengono utilizzate per ottenere i token JWT da AWS Security Token Service (STS) e questi token JWT fungono da token 2.0 per l'autenticazione. OAuth Questo metodo sfrutta il supporto OAuth 2.0 esistente in Amazon MQ for RabbitMQ, dove AWS funge da provider di identità OAuth 2.0. L'autenticazione degli utenti è gestita da AWS IAM, mentre le autorizzazioni delle risorse per vhost, exchange, code e argomenti sono gestite tramite policy IAM e alias di ambito configurati in RabbitMQ. Per ulteriori informazioni, consulta Autenticazione e autorizzazione IAM.

Autenticazione e autorizzazione LDAP

In questo metodo, gli utenti del broker e le relative autorizzazioni sono gestiti da un servizio di directory LDAP esterno. L'autenticazione degli utenti e le autorizzazioni delle risorse sono centralizzate tramite il server LDAP, che consente agli utenti di accedere a RabbitMQ utilizzando le credenziali del servizio di directory esistenti. Per ulteriori informazioni, vedere Autenticazione e autorizzazione LDAP.

Autenticazione e autorizzazione HTTP

In questo metodo, gli utenti del broker e le relative autorizzazioni sono gestiti da un server HTTP esterno. L'autenticazione degli utenti e le autorizzazioni delle risorse sono centralizzate tramite il server HTTP, che consente agli utenti di accedere a RabbitMQ utilizzando il proprio provider di

autenticazione e autorizzazione. Per ulteriori informazioni su questo metodo, consulta Autenticazione e autorizzazione HTTP.

Autenticazione con certificato SSL

Amazon MQ supporta TLS reciproco (mTLS) per i broker RabbitMQ. Il plug-in di autenticazione SSL utilizza i certificati client delle connessioni MTLS per autenticare gli utenti. Con questo metodo, gli utenti del broker vengono autenticati utilizzando certificati client X.509 anziché credenziali di nome utente e password. Il certificato del client viene convalidato rispetto a un'autorità di certificazione (CA) affidabile e il nome utente viene estratto da un campo del certificato, ad esempio Common Name (CN) o Subject Alternative Name (SAN). Questo metodo fornisce un'autenticazione avanzata senza trasmettere credenziali sulla rete. Per ulteriori informazioni, consulta Autenticazione con certificato SSL.

> **Note**
>
> RabbitMQ supporta più metodi di autenticazione e autorizzazione da utilizzare contemporaneamente. Ad esempio, è possibile abilitare sia l'autenticazione OAuth 2.0 che quella semplice (interna). Per ulteriori informazioni, consulta la sezione del tutorial OAuth 2.0 sull'abilitazione dell'autenticazione OAuth 2.0 e semplice (interna) e la documentazione sul controllo degli accessi di RabbitMQ.
>
> Amazon MQ consiglia di creare un utente interno durante il test delle configurazioni di autenticazione. Ciò consente di convalidare la configurazione di accesso utilizzando l'API di gestione RabbitMQ. Per ulteriori informazioni, vedere Convalida dell'accesso.

Autenticazione e autorizzazione per Amazon MQ for ActiveMQ

Amazon MQ for ActiveMQ supporta i seguenti metodi di autenticazione e autorizzazione:

Autenticazione e autorizzazione semplici

Con questo metodo, gli utenti del broker vengono creati e gestiti tramite la console o l'API di Amazon MQ. Gli utenti possono essere configurati con autorizzazioni specifiche per accedere a code, argomenti e alla console Web ActiveMQ. Per ulteriori informazioni su questo metodo, vedere Creazione di un utente del broker ActiveMQ.

Autenticazione e autorizzazione LDAP

Con questo metodo, gli utenti del broker si autenticano tramite le credenziali memorizzate nel server LDAP. È possibile aggiungere, eliminare e modificare utenti e assegnare autorizzazioni ad argomenti e code tramite il server LDAP, fornendo autenticazione e autorizzazione centralizzate. Per ulteriori informazioni su questo metodo, vedere Integrazione dei broker ActiveMQ con LDAP.

Aggiornamento di una versione del motore del broker Amazon MQ

Amazon MQ fornisce regolarmente nuove versioni del motore di brokeraggio per tutti i tipi di motori di broker supportati. Le nuove versioni del motore includono patch di sicurezza, correzioni di bug e altri miglioramenti del motore del broker.

Amazon MQ organizza i numeri di versione in base alle specifiche di versioning semantico as. `X.Y.Z` Nelle implementazioni di Amazon MQ, `X` indica la versione principale, `Y` rappresenta la versione secondaria e `Z` indica il numero di versione della patch. Amazon MQ supporta due tipi di upgrade:

- Aggiornamento della versione principale: si verifica quando cambiano i numeri della versione principale del motore. Ad esempio, l'aggiornamento dalla versione 3.13 di RabbitMQ alla versione 4.2 è considerato un aggiornamento della versione principale.
- Aggiornamento della versione secondaria: si verifica quando cambiano i numeri della versione secondaria del motore. Ad esempio, l'aggiornamento dalla versione 3. 11 alla versione 3. 12 è considerato un aggiornamento secondario della versione.

Puoi aggiornare manualmente il tuo broker in qualsiasi momento alla successiva versione principale o secondaria supportata. Amazon MQ gestisce l'aggiornamento all'ultima versione di patch supportata per tutti i broker durante la finestra di manutenzione programmata. Gli aggiornamenti di versione manuali e automatici avvengono durante la finestra di manutenzione programmata o dopo il riavvio del broker. Amazon MQ aggiorna il tuo broker alla versione secondaria successiva quando la versione secondaria corrente raggiunge la fine del supporto.

Aggiornamento manuale della versione del motore

Puoi aggiornare la versione del motore di un broker utilizzando l' Console di gestione AWS AWS CLI, la o l'API Amazon MQ.

Console di gestione AWS

Per aggiornare la versione del motore di un broker utilizzando il Console di gestione AWS

1. Nella pagina dei dettagli del broker, scegliere Edit (Modifica).
2. In Specifications (Specifiche), per Broker engine version (Versione del motore del broker), scegliere il numero della nuova versione dall'elenco a discesa.
3. Scorrere fino alla parte inferiore della pagina e selezionare Schedule modifications (Pianifica modifiche).
4. Alla pagina Schedule broker modifications (Pianifica modifiche del broker), per When to apply modifications (Quando applicare le modifiche), scegliere una delle opzioni seguenti.

   - Scegliere After the next reboot (Al prossimo riavvio) se si desidera che Amazon MQ completi l'aggiornamento della versione durante la prossima finestra di manutenzione pianificata.
   - Scegliere Immediately (Subito), se si desidera riavviare il broker e aggiornare immediatamente la versione del motore.

   > **Important**
   >
   > I broker a istanza singola sono offline durante il riavvio. Per i broker di cluster, durante il riavvio del broker viene interrotto solo un nodo alla volta.

5. Scegliere Apply (Applica) per completare l'applicazione delle modifiche.

AWS CLI

Per aggiornare la versione del motore di un broker utilizzando il AWS CLI

1. Utilizzare il comando della CLI update-broker e specificare i parametri seguenti, come mostrato nell'esempio.

   - `--broker-id`: ID univoco che Amazon MQ genera per il broker. Puoi analizzare l'ID dall'ARN del broker. Ad esempio, con il seguente ARN, `arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`, l'ID del broker sarebbe `b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`.
   - `--engine-version`: numero di versione del motore del broker a cui effettuare l'aggiornamento.

```
aws mq update-broker --broker-id broker-id --engine-version version-number
```

2. (Facoltativo) Utilizzate il comando CLI reboot-broker per riavviare il broker se desiderate aggiornare immediatamente la versione del motore.

```
aws mq reboot-broker --broker-id broker-id
```

Se non si desidera riavviare il broker e applicare immediatamente le modifiche, Amazon MQ aggiornerà il broker durante la prossima finestra di manutenzione pianificata.

> **Important**
>
> I broker a istanza singola sono offline durante il riavvio. Per i broker di cluster, durante il riavvio del broker viene interrotto solo un nodo alla volta.

API di Amazon MQ

Aggiornamento della versione del motore di un broker tramite l'API di Amazon MQ

1. Usa l'operazione API UpdateBroker. Specificare `broker-id`come parametro del percorso. Negli esempi seguenti si presuppone un broker nella regione `us-west-2`. Per ulteriori informazioni sugli endpoint Amazon MQ disponibili, consultare Quote ed endpoint di Amazon MQ in Riferimenti generali di .

```
PUT /v1/brokers/broker-id HTTP/1.1
Host: mq.us-west-2.amazonaws.com
Date: Mon, 7 June 2021 12:00:00 GMT
x-amz-date: Mon, 7 June 2021 12:00:00 GMT
Authorization: authorization-string
```

Utilizzare `engineVersion` nel payload della richiesta per specificare il numero di versione a cui eseguire l'aggiornamento del broker.

```
{
    "engineVersion": "engine-version-number"
}
```

2. (Facoltativo) Utilizza l'operazione RebootBrokerAPI per riavviare il broker se desideri aggiornare immediatamente la versione del motore. `broker-id`è specificato come parametro di percorso.

```
POST /v1/brokers/broker-id/reboot-broker HTTP/1.1
Host: mq.us-west-2.amazonaws.com
Date: Mon, 7 June 2021 12:00:00 GMT
x-amz-date: Mon, 7 June 2021 12:00:00 GMT
Authorization: authorization-string
```

Se non si desidera riavviare il broker e applicare immediatamente le modifiche, Amazon MQ aggiornerà il broker durante la prossima finestra di manutenzione pianificata.

> **Important**
>
> I broker a istanza singola sono offline durante il riavvio. Per i broker di cluster, durante il riavvio del broker viene interrotto solo un nodo alla volta.

Aggiornamento di un tipo di istanza del broker Amazon MQ

> **Important**
>
> `mq.m7g.x`le istanze sono disponibili solo per i broker Amazon MQ for RabbitMQ. I broker Amazon MQ for ActiveMQ utilizzano solo istanze. `mq.m5.x`

La descrizione combinata della classe dell'istanza del broker (`m7g`) e della dimensione (`large`) è denominata tipo di istanza del broker (ad esempio,). `mq.m7g.large` Quando si sceglie un tipo di istanza, è importante considerare i fattori che influiranno sulle prestazioni del broker:

- il numero di client e di code
- il volume dei messaggi inviati
- messaggi conservati in memoria
- messaggi ridondanti

I tipi di istanze di broker più piccoli (`mq.m7g.medium`) sono consigliati solo per testare le prestazioni delle applicazioni. Consigliamo tipi di istanze broker più grandi (`mq.m7g.large` e superiori) per livelli di produzione di client e code, throughput elevato, messaggi in memoria e messaggi ridondanti.

Ti consigliamo di passare a un tipo di istanza più grande (ad esempio from `micro` to`large`) se riscontri problemi di prestazioni o se stai passando da un ambiente di test a un ambiente di produzione. Per aggiornare il tipo di istanza, puoi utilizzare l' Console di gestione AWS AWS CLI, o l'API Amazon MQ.

Console di gestione AWS

Per eseguire l'aggiornamento a un tipo di istanza più grande utilizzando il Console di gestione AWS, procedi come segue:

1. Accedere alla console Amazon MQ.
2. Nel pannello di navigazione a sinistra, scegli Brokers (broker) e quindi scegli dall'elenco il broker che desideri aggiornare.
3. Nella pagina dei dettagli del broker, scegliere Edit (Modifica).
4. In Specifiche, per Tipo di istanza Broker scegli il nuovo tipo di istanza dall'elenco a discesa.
5. Nella parte inferiore della pagina, scegli Pianifica modifiche.
6. Alla pagina Schedule broker modifications (Pianifica modifiche del broker), per When to apply modifications (Quando applicare le modifiche), scegliere una delle opzioni seguenti.

   - Scegli Dopo il prossimo riavvio, se desideri che Amazon MQ completi l'aggiornamento durante la successiva finestra di manutenzione programmata.
   - Scegli Immediatamente se desideri riavviare il broker e aggiornare immediatamente il tipo di istanza.

   > **Important**
   >
   > I broker a istanza singola sono offline durante il riavvio. Per i broker di cluster, durante il riavvio del broker viene interrotto solo un nodo alla volta.

7. Scegliere Apply (Applica) per completare l'applicazione delle modifiche.

AWS CLI

Per aggiornare il tipo di istanza di un broker utilizzando il AWS CLI

1. Utilizzate il comando CLI modify-broker e specificate i seguenti parametri, come mostrato nell'esempio.

   - `--broker-id`: ID univoco che Amazon MQ genera per il broker.
   - `--host-instance-type`: numero di versione del motore del broker a cui effettuare l'aggiornamento.

   ```
   aws mq modify-broker --broker-id broker-id --host-instance-type instance-type
   ```

2. (Facoltativo) Utilizza il comando CLI reboot-broker per riavviare il broker se desideri aggiornare immediatamente il tipo di istanza.

   ```
   aws mq reboot-broker --broker-id broker-id
   ```

   Se non si desidera riavviare il broker e applicare immediatamente le modifiche, Amazon MQ aggiornerà il broker durante la prossima finestra di manutenzione pianificata.

   > **Important**
   >
   > I broker a istanza singola sono offline durante il riavvio. Per i broker di cluster, durante il riavvio del broker viene interrotto solo un nodo alla volta.

API di Amazon MQ

Per aggiornare il tipo di istanza di un broker utilizzando l'API Amazon MQ

1. Usa l'operazione API UpdateBroker. Specificare `broker-id`come parametro del percorso. Negli esempi seguenti si presuppone un broker nella regione `us-west-2`. Per ulteriori informazioni sugli endpoint Amazon MQ disponibili, consulta Endpoint e quote Amazon MQ nel. Riferimenti generali di AWS

   ```
   PUT /v1/brokers/broker-id HTTP/1.1
   Host: mq.us-west-2.amazonaws.com
   ```

```
Date: Mon, 7 June 2021 12:00:00 GMT
x-amz-date: Mon, 7 June 2021 12:00:00 GMT
Authorization: authorization-string
```

Utilizzalo `host-instance-type` nel payload della richiesta per specificare il tipo di istanza a cui il broker deve effettuare l'upgrade.

```
{
    "host-instance-type": "host-instance-type"
}
```

2. (Facoltativo) Utilizzate l'operazione RebootBrokerAPI per riavviare il broker, se desiderate aggiornare immediatamente la versione del motore. `broker-id`è specificato come parametro di percorso.

```
POST /v1/brokers/broker-id/reboot-broker HTTP/1.1
Host: mq.us-west-2.amazonaws.com
Date: Mon, 7 June 2021 12:00:00 GMT
x-amz-date: Mon, 7 June 2021 12:00:00 GMT
Authorization: authorization-string
```

Se non si desidera riavviare il broker e applicare immediatamente le modifiche, Amazon MQ aggiornerà il broker durante la prossima finestra di manutenzione pianificata.

> **Important**
>
> I broker a istanza singola sono offline durante il riavvio. Per i broker di cluster, durante il riavvio del broker viene interrotto solo un nodo alla volta.

Amazon MQ per i tipi di storage ActiveMQ

Amazon MQ per ActiveMQ supporta Amazon Elastic File System (EFS) e Amazon Elastic Block Store (EBS). Per impostazione predefinita, i broker ActiveMQ utilizzano Amazon EFS per l'archiviazione del broker. Per sfruttare l'elevata durata e la replica in più zone di disponibilità, utilizza Amazon EFS. Per sfruttare la bassa latenza e la velocità effettiva elevata, utilizza Amazon EBS.

Important

- Puoi utilizzare Amazon EBS solo con la famiglia di tipo di istanze del broker `mq.m5`.
- Sebbene sia possibile modificare il tipo di istanza del broker, non è possibile modificare il tipo di archiviazione del broker dopo la creazione del broker.
- Amazon EBS replica i dati all'interno di una singola zona di disponibilità e non supporta la modalità di implementazione ActiveMQ attiva/in standby.

Differenze tra i tipi di storage

Nella tabella seguente viene fornita una breve panoramica delle differenze tra i tipi di archiviazione in memoria, Amazon EFS e Amazon EBS per i broker, ActiveMQ.

Storage Type (Tipo di storage)	Persistenza	Esempio di caso d'uso	Numero massimo approssimativo di messaggi accodati per produttore, al secondo (messaggio di 1 KB)	Replica
In memoria	Non persistente	• Quotazioni di borsa • Aggiornamenti dei dati di posizione • Dati modificati frequentemente	5.000	Nessuno
Amazon EBS	Persistente	• Elevati volumi di testo	500	Copie multiple all'interno di una

Storage Type (Tipo di storage)	Persistenza	Esempio di caso d'uso	Numero massimo approssimativo di messaggi accodati per produttore, al secondo (messaggio di 1 KB)	Replica
		• Elaborazione di ordini		singola zona di disponibilità (AZ)
Amazon EFS	Persistente	Transazioni finanziarie	80	Copie multiple su più copie AZs

Lo storage dei messaggi in memoria fornisce la latenza più bassa e il throughput massimo. Tuttavia, i messaggi vengono persi durante la sostituzione dell'istanza o il riavvio del broker.

Amazon EFS è progettato per essere altamente durevole, replicato su più piattaforme AZs per prevenire la perdita di dati derivante dal guasto di un singolo componente o da un problema che influisce sulla disponibilità di una zona di disponibilità. Amazon EBS è ottimizzato per la velocità effettiva e replicato su più server all'interno di una singola zona di disponibilità.

Configurazione di un broker Amazon MQ privato

Un broker privato non è accessibile al pubblico e non è accessibile dall'esterno del tuo VPC. Prima di configurare un broker privato, visualizza le seguenti informazioni su VPCs sottoreti e gruppi di sicurezza:

- VPCs
  - Le sottoreti e i gruppi di sicurezza di un broker devono trovarsi nello stesso VPC.
  - Quando utilizzi un broker privato, potresti vedere indirizzi IP che non hai configurato con il tuo VPC. Si tratta di indirizzi IP dell'infrastruttura Amazon MQ e non richiedono alcuna azione.
- Sottoreti
  - Se le sottoreti si trovano all'interno di un VPC condiviso, il VPC deve appartenere allo stesso account che ha creato il broker.

- Se non viene fornita alcuna sottorete, verranno utilizzate le sottoreti predefinite nel VPC predefinito.
- Una volta creato il broker, le sottoreti utilizzate non possono essere modificate.
- Per i cluster e active/standby i broker, le sottoreti devono trovarsi in zone di disponibilità diverse.
- Per i broker a istanza singola, è possibile specificare quale sottorete utilizzare e il broker verrà creato all'interno della stessa zona di disponibilità.

- Gruppi di sicurezza
  - Se non viene fornito alcun gruppo di sicurezza, verranno utilizzati i gruppi di sicurezza predefiniti nel VPC predefinito.
  - L'istanza singola, il cluster e active/standby i broker richiedono almeno un gruppo di sicurezza (ad esempio, il gruppo di sicurezza predefinito).

    > **Note**
    >
    > I broker RabbitMQ pubblici non utilizzano sottoreti o gruppi di sicurezza.

  - Una volta creato il broker, il gruppo di sicurezza utilizzato non può essere modificato. I gruppi di sicurezza possono comunque essere modificati.

Configurazione di un broker privato in Console di gestione AWS

Per configurare un broker privato, inizia a creare un nuovo broker in. Console di gestione AWS Quindi, nella sezione Impostazioni di rete, per configurare la connettività del broker, procedi come segue:

1. Scegli l'accesso privato per il tuo broker. Per connetterti a un broker privato, puoi utilizzare IPv4 IPv6, o dual-stack (IPv4 e). IPv6 Per ulteriori informazioni, consulta Connecting to Amazon MQ.
2. Quindi, scegli Usa il VPC, le sottoreti e i gruppi di sicurezza predefiniti oppure seleziona Seleziona VPC, sottoreti e gruppi di sicurezza esistenti. Se non desideri utilizzare il VPC, le sottoreti o i gruppi di sicurezza predefiniti o esistenti, devi crearne uno nuovo per connetterti al broker privato.

   > **Note**
   >
   > Per l'accesso al broker privato, il metodo di connessione sarà lo stesso del tipo di IP selezionato della sottorete. Una volta creato il broker, l'endpoint VPC non può essere

modificato e avrà sempre il tipo IP delle sottoreti selezionate. Se desideri utilizzare un nuovo tipo di IP, devi creare un nuovo broker.

> **Note**
>
> Amazon MQ for ActiveMQ non utilizza endpoint VPC. Quando crei per la prima volta un broker ActiveMQ, Amazon MQ fornisce un'interfaccia di rete elastica (ENI) nel VPC. I gruppi di sicurezza sono collocati nell'ENI e possono essere utilizzati sia per broker pubblici che privati.

Accesso alla console web del broker Amazon MQ senza accessibilità pubblica

Quando disattivi l'accessibilità pubblica per il tuo broker, l'ID dell' AWS account che ha creato il broker può accedere al broker privato. Se disattivi l'accessibilità pubblica per il tuo broker, devi eseguire le seguenti operazioni per accedere alla console web del broker.

1. Creare un'istanza EC2 Linux in `public-vpc` (con un IP pubblico, se necessario).
2. Per verificare che il VPC è configurato correttamente, stabilire una connessione `ssh` all'istanza EC2 e utilizzare il comando `curl` con l'URI del broker.
3. Dal computer, creare un tunnel `ssh` all'istanza EC2 utilizzando il percorso al file della chiave privata e l'indirizzo IP dell'istanza pubblica EC2. Ad esempio:

   ```
   ssh -i ~/.ssh/id_rsa -N -C -q -f -D 8080 ec2-user@203.0.113.0
   ```

   Un server proxy di inoltro viene avviato sul computer.
4. Installa un client proxy, ad esempio, FoxyProxysul tuo computer.
5. Configurare il client proxy usando le impostazioni seguenti:

   - Per tipo di proxy, specificare `SOCKS5`.
   - Per l'indirizzo IP, il nome DNS e il nome server, specificare `localhost`.
   - Per la porta, specific `8080`.
   - Rimuovere eventuali modelli URL esistenti.

- Per il modello URL, specificare `*.mq.*.amazonaws.com*`
- Per il tipo di connessione, specificare `HTTP(S)`.

Quando abiliti il client proxy, puoi accedere alla console Web sul computer.

> **Important**
>
> Se utilizzi un broker privato, potresti visualizzare indirizzi IP che non hai configurato con il tuo VPC. Si tratta di indirizzi IP dell'infrastruttura RabbitMQ su Amazon MQ e non richiedono alcuna azione.

Pianificazione della finestra di manutenzione per un broker Amazon MQ

Periodicamente, Amazon MQ esegue la manutenzione dell'hardware, del sistema operativo o del software del motore di un broker di messaggi durante la finestra di manutenzione. Ad esempio, se hai cambiato il tipo di istanza del broker, Amazon MQ applicherà le modifiche durante la successiva finestra di manutenzione programmata. La durata della manutenzione può durare fino a due ore a seconda delle operazioni pianificate per il tuo broker di messaggi. È possibile ridurre al minimo i tempi di inattività durante una finestra di manutenzione selezionando una modalità di implementazione del broker con elevata disponibilità su più zone di disponibilità (AZ).

Amazon MQ for ActiveMQ fornisce distribuzioni attive/in standby per un'elevata disponibilità. In active/standby modalità, Amazon MQ esegue le operazioni di manutenzione un'istanza alla volta e almeno un'istanza rimane disponibile. Inoltre, puoi configurare una rete di broker con finestre di manutenzione diverse nel corso della settimana. Amazon MQ for RabbitMQ fornisce le distribuzioni di cluster per l'alta disponibilità. Nelle implementazioni di cluster, Amazon MQ esegue le operazioni di manutenzione un nodo alla volta mantenendo almeno due nodi in esecuzione in ogni momento.

Quando crei il tuo broker per la prima volta, puoi pianificare la finestra di manutenzione in modo che si verifichi una volta alla settimana a un'ora specificata. È possibile regolare la finestra di manutenzione di un broker solo fino a quattro intervalli prima della prossima finestra di manutenzione pianificata. Una volta completata la finestra di manutenzione del broker, Amazon MQ reimposta il limite e puoi modificare nuovamente la pianificazione prima che si verifichi la finestra di manutenzione

successiva. La disponibilità del broker non viene influenzata dalla regolazione della finestra di manutenzione del broker.

Per modificare la finestra di manutenzione del broker, puoi utilizzare l' Console di gestione AWS AWS CLI, la o l'API Amazon MQ.

Pianifica la finestra di manutenzione del broker utilizzando il Console di gestione AWS

Per modificare la finestra di manutenzione del broker utilizzando il Console di gestione AWS

1. Accedere alla console Amazon MQ.
2. Nel pannello di navigazione a sinistra, scegli Brokers (broker) e quindi scegli dall'elenco il broker che desideri aggiornare.
3. Nella pagina dei dettagli del broker, scegliere Edit (Modifica).
4. In Maintenance (Manutenzione), eseguire queste operazioni.

    a. Per Start day (Giorno di inizio) scegliere un giorno della settimana, ad esempio, Sunday (Domenica), dal menu a tendina.

    b. Per Start time (Ora di inizio), scegliere l'ora e i minuti del giorno che si desidera pianificare per la prossima finestra di manutenzione del broker, ad esempio,12:00.

    > **Note**
    >
    > Le opzioni per Start time (Ora di inizio) sono configurate nel fuso orario UTC+0.

5. Quindi, seleziona Modifiche alla pianificazione. Quindi scegli Dopo il prossimo riavvio o Immediatamente. Scegliendo Dopo il prossimo riavvio si aggiornerà immediatamente la finestra di manutenzione senza riavviare il broker. Scegliendo Immediatamente si riavvierà immediatamente il broker.
6. Nella pagina dei dettagli del broker, in Maintenance window (Finestra di manutenzione), verificare che sia visualizzata la nuova pianificazione preferita.

Pianifica la finestra di manutenzione del broker utilizzando il AWS CLI

Per regolare la finestra di manutenzione del broker, utilizza il AWS CLI

1. Utilizzare il comando della CLI update-broker e specificare i parametri seguenti, come mostrato nell'esempio.

- `--broker-id`: ID univoco che Amazon MQ genera per il broker. Puoi analizzare l'ID dall'ARN del broker. Ad esempio, con il seguente ARN, `arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`, l'ID del broker sarebbe `b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`.
- `--maintenance-window-start-time`: parametri che determinano l'orario di inizio della finestra di manutenzione settimanale fornito nella seguente struttura.
  - `DayOfWeek`: giorno della settimana, nella sintassi seguente: `MONDAY| TUESDAY | WEDNESDAY | THURSDAY | FRIDAY | SATURDAY | SUNDAY`
  - `TimeOfDay`: ora nel formato 24 ore.
  - `TimeZone`: (opzionale) fuso orario, nel formato paese/città o nel formato UTC. Impostato su UTC per impostazione predefinita.

```
aws mq update-broker --broker-id broker-id \
--maintenance-window-start-time DayOfWeek=SUNDAY,TimeOfDay=13:00,TimeZone=America/Los_Angeles
```

2. (Opzionale) Utilizzare il comando della CLI describe-broker per verificare che la finestra di manutenzione sia stata aggiornata correttamente.

```
aws mq describe-broker --broker-id broker-id
```

Pianifica la finestra di manutenzione del broker utilizzando l'API Amazon MQ

Regolazione della finestra di manutenzione dell broker utilizzando l'API di Amazon MQ

1. Usa l'operazione API UpdateBroker. Specificare `broker-id`come parametro del percorso. Negli esempi seguenti si presuppone un broker nella regione `us-west-2`. Per ulteriori informazioni sugli endpoint Amazon MQ disponibili, consulta Endpoint e quote Amazon MQ nel. Riferimenti generali di AWS

```
PUT /v1/brokers/broker-id HTTP/1.1
Host: mq.us-west-2.amazonaws.com
Date: Wed, 7 July 2021 12:00:00 GMT
x-amz-date: Wed, 7 July 2021 12:00:00 GMT
Authorization: authorization-string
```

Utilizzare il parametro `maintenanceWindowStartTime` e il tipo di risorsa`WeeklyStartTime` nel payload della richiesta.

```
{
"maintenanceWindowStartTime": {
    "dayOfWeek": "SUNDAY",
    "timeZone": "America/Los_Angeles",
    "timeOfDay": "13:00"
  }
}
```

2. (Facoltativo) Utilizza l'operazione DescribeBrokerAPI per verificare che la finestra di manutenzione sia stata aggiornata correttamente. `broker-id`è specificato come parametro di percorso.

```
GET /v1/brokers/broker-id HTTP/1.1
Host: mq.us-west-2.amazonaws.com
Date: Wed, 7 July 2021 12:00:00 GMT
x-amz-date: Wed, 7 July 2021 12:00:00 GMT
Authorization: authorization-string
```

Riavvio di un broker Amazon MQ

Per applicare una nuova configurazione a un broker, puoi riavviare il broker.

> **Note**
>
> Se il broker ActiveMQ non risponde, puoi riavviarlo per eseguire il ripristino da uno stato di errore.

L'esempio seguente mostra come puoi riavviare un broker Amazon MQ utilizzando la Console di gestione AWS.

Riavviare di un broker Amazon MQ

1. Accedere alla console Amazon MQ.
2. Dall'elenco dei broker, scegli il nome del tuo broker (ad esempio, MyBroker).

3. Nella ***MyBroker***pagina, scegli Azioni, Riavvia broker.

> **Important**
>
> I broker a istanza singola saranno offline durante il riavvio. I broker dei cluster saranno disponibili, ma ogni nodo viene riavviato uno alla volta.

4. Nella finestra di dialogo Reboot broker (Riavvia broker), scegliere Reboot (Riavvia).

   Il riavvio di un broker richiede circa 5 minuti. Se il riavvio include modifiche alle dimensioni dell'istanza o viene eseguito su un broker con un'elevata profondità di coda, il processo di riavvio può richiedere più tempo.

Eliminazione di un broker Amazon MQ

Se non utilizzi un broker Amazon MQ (e non prevedi di utilizzarlo in un prossimo futuro), è consigliabile eliminarlo da Amazon MQ per ridurre i AWS costi.

L'esempio seguente mostra come eliminare un broker utilizzando l' Console di gestione AWS.

Eliminazione di un broker Amazon MQ

1. Accedere alla console Amazon MQ.
2. Dall'elenco dei broker, seleziona il tuo broker (ad esempio MyBroker), quindi scegli Elimina.
3. Nella casella Elimina***MyBroker***? nella finestra di dialogo, digitate `delete` e scegliete Elimina.

   L'eliminazione del broker richiede circa 5 minuti.

Stati dei broker Amazon MQ

La condizione corrente di un broker è indicata da uno stato. Nella tabella seguente sono elencati gli stati di un broker Amazon MQ.

Console	"Hello, World!"	Description
Creazione non riuscita	`CREATION_FAILED`	Non è stato possibile creare il broker.

Console	"Hello, World!"	Description
Creazione in corso	`CREATION_IN_PROGRESS`	Il broker è attualmente in fase di creazione.
Eliminazione in corso	`DELETION_IN_PROGRESS`	Il broker è attualmente in fase di eliminazione.
Riavvio in corso	`REBOOT_IN_PROGRESS`	Il broker è attualmente in fase di riavvio.
In esecuzione	`RUNNING`	Il broker è operativo.
Richiesta un'operazione critica	`CRITICAL_ACTION_REQUIRED`	Il broker è in esecuzione, ma è in uno stato degradato e richiede un'azione immediata . Puoi trovare le istruzioni per risolvere il problema scegliendo il codice dell'azione richiesto dall'elenco in Risoluzione dei problemi.

Aggiungere tag alle risorse Amazon MQ

Per organizzare e identificare le risorse Amazon MQ per l'allocazione dei costi, puoi aggiungere tag metadati che identificano lo scopo di un broker o di una configurazione. Questo è particolarmente utile quando si dispone di numerosi broker. È possibile utilizzare i tag di allocazione dei costi per organizzare la AWS fattura in modo che rifletta la propria struttura dei costi. A tale scopo, registrati per ricevere nella fattura AWS dell'account le chiavi e i valori dei tag. Per ulteriori informazioni, consulta Impostazione di un report di allocazione dei costi mensili nella Guida per l'utente di AWS Billing .

Ad esempio, puoi aggiungere tag che rappresentano il centro di costo e lo scopo delle risorse Amazon MQ:

Risorsa	Chiave	Valore
Broker1	Cost Center	34567
	Stack	Production
Broker2	Cost Center	34567
	Stack	Production
Broker3	Cost Center	12345
	Stack	Development

Questo schema di tagging consente di raggruppare due broker che eseguono attività correlate nello stesso centro di costo, mentre esegui il tagging di un broker non pertinente con un tag di allocazione dei costi differente.

Aggiungere tag nella console Amazon MQ

Puoi aggiungere rapidamente tag alle risorse che stai creando nella console Amazon MQ seguendo questi passaggi:

1. Nella pagina Create a broker (Crea un broker), seleziona Additional settings (Impostazioni aggiuntive).
2. In Tags (Tag), seleziona Add tag (Aggiungi tag).
3. Inserisci una coppia Key (Chiave) e Value (Valore).
4. (Facoltativo) Seleziona Add tag (Aggiungi tag) per aggiungere più tag al broker.
5. Seleziona Create broker (Crea broker).

Per aggiungere i tag mentre crei una configurazione:

1. Nella pagina Create configuration (Crea configurazione), seleziona Advanced (Avanzato).
2. In Tags (Tag) nella pagina Create configuration (Crea configurazione), seleziona Add tag (Aggiungi tag).
3. Inserisci una coppia Key (Chiave) e Value (Valore).

4. (Facoltativo) Seleziona Add tag (Aggiungi tag) per aggiungere più tag alla configurazione.
5. Seleziona Create configuration (Crea configurazione).

Dopo aver aggiunto i tag, puoi visualizzare, modificare e rimuovere i tag per le tue risorse nella console Amazon MQ. Puoi anche visualizzare i tag delle tue risorse utilizzando l'API REST. Per ulteriori informazioni, consultare il Riferimento all'API REST di Amazon MQ.

Utilizzo di Amazon MQ per ActiveMQ

Amazon MQ semplifica la creazione di un broker di messaggistica con le risorse di calcolo e archiviazione che soddisfano le tue esigenze. Puoi creare, gestire ed eliminare broker utilizzando l' Console di gestione AWS API REST di Amazon MQ o il AWS Command Line Interface.

I broker Amazon MQ for ActiveMQ possono essere implementati come broker a istanza singola o broker attivi/in standby. Per entrambe le modalità di implementazione, Amazon MQ garantisce un'elevata durata archiviando i dati in modo ridondante.

> **Note**
>
> Amazon MQ utilizza Apache KahaDB come datastore. Altri archivi di dati, come JDBC e LevelDB, non sono supportati.

Puoi accedere ai broker utilizzando qualsiasi linguaggio di programmazione supportato da ActiveMQ e abilitando TLS esplicitamente per i seguenti protocolli:

- AMQP
- MQTT
- MQTT over WebSocket
- OpenWire
- STOMP
- STOMP over WebSocket

Per ulteriori informazioni su Amazon MQ REST APIs, consulta l'Amazon MQ REST API Reference.

Amazon MQ per broker ActiveMQ

Cos'è un broker Amazon MQ for ActiveMQ?

Un broker è un ambiente broker dei messaggi in esecuzione su Amazon MQ. Costituisce l'elemento di base di Amazon MQ. La descrizione combinata della classe di istanza del broker (`m5`) e della dimensione (`large`,`medium`) è denominata tipo di istanza del broker (ad esempio,). `mq.m5.large` Per ulteriori informazioni, consulta Broker instance types.

- Un broker a istanza singola è composto da un broker in una zona di disponibilità. Il broker comunica con l'applicazione e con un volume di archiviazione Amazon EBS o Amazon EFS.
- Un broker attivo/in standby è composto da due broker in due diverse zone di disponibilità, configurate in una coppia ridondante. Questi broker comunicano in modo sincrono con l'applicazione e con Amazon EFS.

Per ulteriori informazioni, consulta Opzioni di implementazione per i broker Amazon MQ for ActiveMQ.

Puoi abilitare aggiornamenti automatici minori della versione a nuove versioni minori del motore del broker, poiché Apache rilascia nuove versioni. Gli aggiornamenti automatici si verificano durante la finestra di manutenzione definita dal giorno della settimana, dall'ora del giorno (in formato 24 ore) e dal fuso orario (UTC per impostazione predefinita).

Per ulteriori informazioni sulla creazione e la gestione di broker, consulta le sezioni seguenti:

- Guida introduttiva: creazione e connessione a un broker ActiveMQ
- Broker
- Broker statuses

Protocolli a livello di collegamento supportati

Puoi accedere ai broker utilizzando qualsiasi linguaggio di programmazione supportato da ActiveMQ e abilitando TLS esplicitamente per i seguenti protocolli:

- AMQP
- MQTT
- MQTT over WebSocket
- OpenWire
- STOMP
- STOMP over WebSocket

Attributes

Un broker ActiveMQ dispone di diversi attributi, ad esempio:

- un nome (`MyBroker`)
- un ID (`b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`)
- un Amazon Resource Name (ARN) (`arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`)
- URL console Web ActiveMQ (`https://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:8162`)

  Per ulteriori informazioni, consultare Console Web nella documentazione di Apache ActiveMQ.

  > **Important**
  >
  > Se si specifica una mappa di autorizzazione che non include il gruppo `activemq-webconsole`, non è possibile utilizzare la console Web ActiveMQ perché il gruppo non è autorizzato a inviare o ricevere messaggi dal broker Amazon MQ.

- Endpoint del protocollo a livello di collegamento:
  - `amqp+ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:5671`
  - `mqtt+ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:8883`
  - `ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:61617`

    > **Note**
    >
    > Questo è un punto OpenWire finale.

  - `stomp+ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:61614`
  - `wss://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:61619`

  Per ulteriori informazioni, consultare Configurazione dei trasporti nella documentazione di Apache ActiveMQ.

> **Note**
>
> Per un active/standby broker, Amazon MQ fornisce due URLs console Web ActiveMQ, ma è attivo un solo URL alla volta. Allo stesso modo, Amazon MQ fornisce due endpoint per ogni protocollo a livello di connessione, ma è attivo un solo endpoint per ogni coppia alla volta. I suffissi `-1` e `-2` indicano una coppia ridondante.

Per un elenco completo di attributi del broker, consultare le sezioni seguenti in Riferimento all'API REST di Amazon MQ:

- ID operazione REST: broker
- ID operazione REST: broker
- ID operazione REST: riavvio broker

Utenti del broker

Un utente ActiveMQ è una persona o un'applicazione che può accedere alle code e agli argomenti di un broker ActiveMQ. È possibile configurare gli utenti in modo che dispongano di autorizzazioni specifiche. Ad esempio, è possibile consentire ad alcuni utenti di accedere alla console Web ActiveMQ.

Un gruppo è un'etichetta semantica. È possibile assegnare un gruppo a un utente e configurare le autorizzazioni per i gruppi in modo che possano inviare, ricevere e amministrare code e argomenti specifici.

> **Important**
>
> Apportare modifiche a un utente non applica le modifiche all'utente in modo istantaneo. Per applicare le modifiche, attendere la finestra di manutenzione successiva o riavviare il broker.

Per informazioni su utente e gruppi, consulta le sezioni seguenti nella documentazione di Apache ActiveMQ.

- Autorizzazione
- Esempio di autorizzazione

Per ulteriori informazioni sulla creazione, la modifica e l'eliminazione di utenti di ActiveMQ, consulta le sezioni seguenti:

- Creazione di un utente broker ActiveMQ
- Utenti

Attributi utente

Per un elenco completo degli attributi utente, consultare le sezioni seguenti in Riferimento all'API REST di Amazon MQ:

- ID operazione REST: utente
- ID operazione REST: utenti

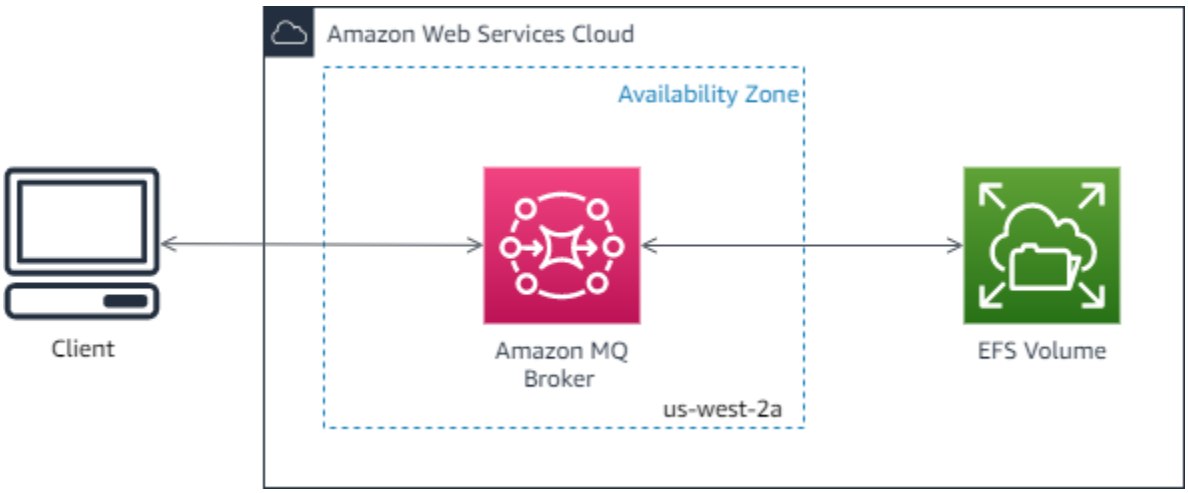
Opzioni di implementazione per i broker Amazon MQ for ActiveMQ

Amazon MQ offre opzioni di distribuzione a singola istanza e cluster per i broker.

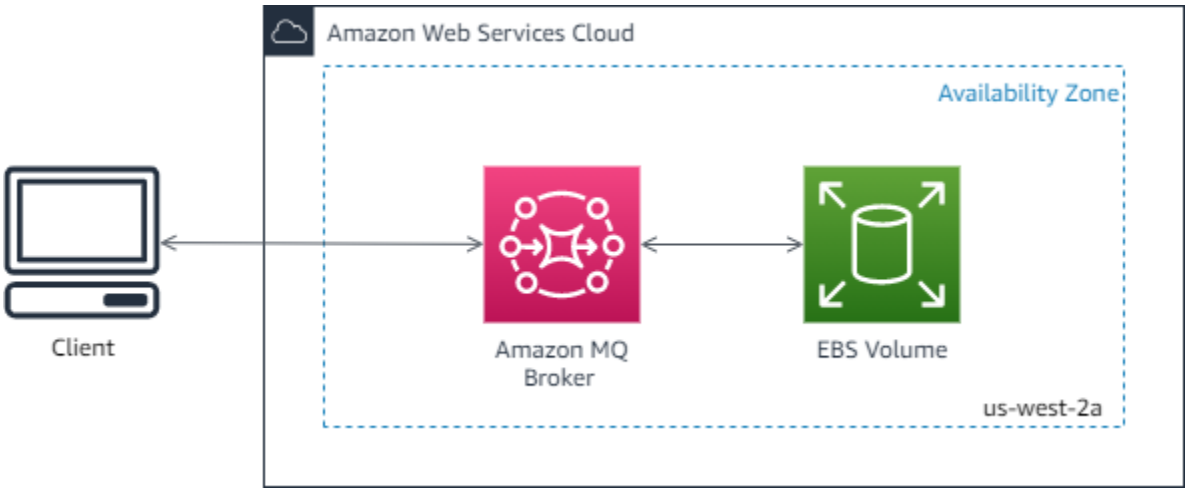
Opzione 1: broker a istanza singola Amazon MQ

Un broker a istanza singola è composto da un broker in una zona di disponibilità. Il broker comunica con l'applicazione e con un volume di archiviazione Amazon EBS o Amazon EFS. I volumi di storage Amazon EFS sono progettati per fornire il massimo livello di durabilità e disponibilità archiviando i dati in modo ridondante su più zone di disponibilità ()AZs. Amazon EBS fornisce archiviazione a livello di blocco ed è ottimizzato per bassa latenza e velocità effettiva elevata. Per ulteriori informazioni sulle classi di archiviazione, consultare Storage.

Il diagramma seguente illustra un broker a istanza singola con storage Amazon EFS replicato su più piattaforme. AZs



Il diagramma seguente illustra un broker a istanza singola con archiviazione Amazon EBS replicata su più server all'interno di una singola zona di disponibilità.



Opzione 2: active/standby broker Amazon MQ per un'elevata disponibilità

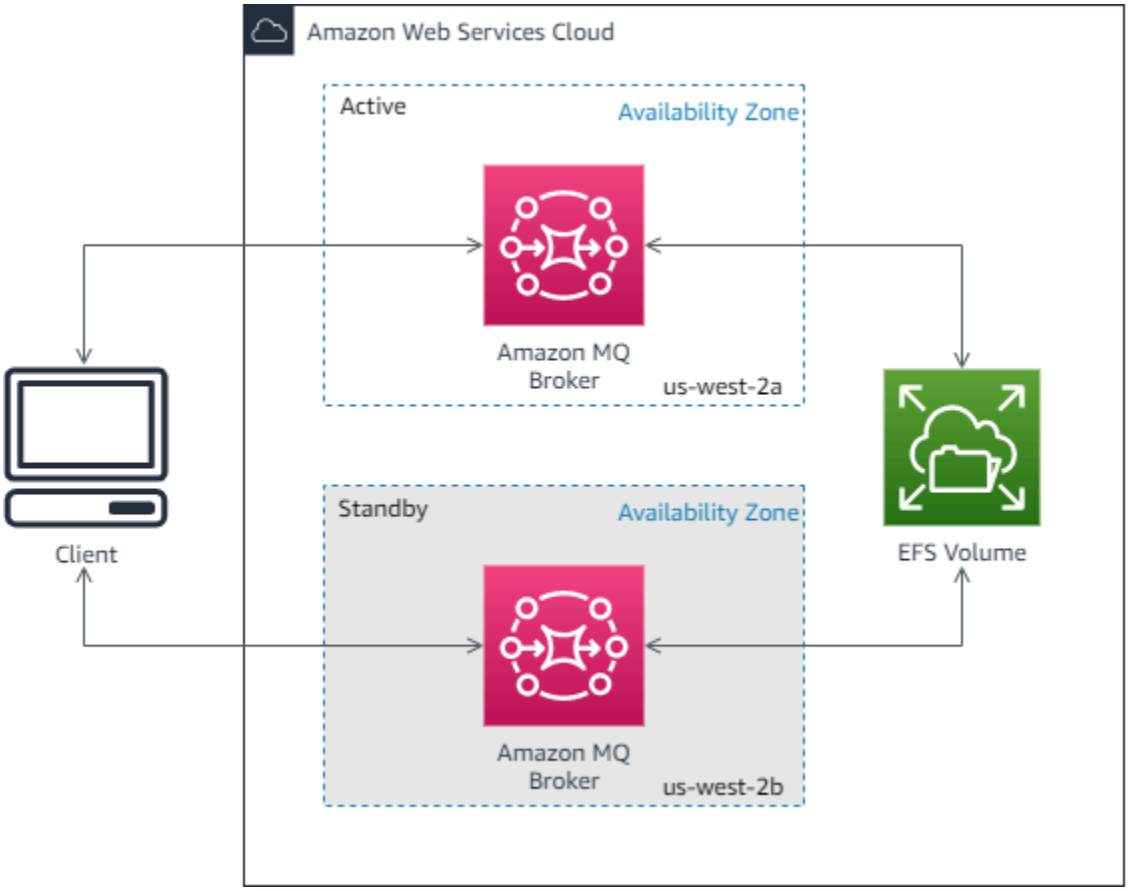
Un broker attivo/in standby è composto da due broker in due diverse zone di disponibilità, configurate in una coppia ridondante. Questi broker comunicano in modo sincrono con l'applicazione e con Amazon EFS. I volumi di storage Amazon EFS sono progettati per fornire il massimo livello di durabilità e disponibilità archiviando i dati in modo ridondante su più zone di disponibilità ()AZs. Per ulteriori informazioni, consulta Storage.

Di solito, solo una delle istanze broker è attiva, mentre l'altra è in standby. Se una delle istanze del broker non funziona correttamente o è in manutenzione, Amazon MQ impiega un breve periodo di tempo per eliminare l'istanza inattiva dal servizio. Ciò consente all'istanza di standby integra di diventare attiva e di iniziare ad accettare le comunicazioni in entrata. Le finestre di manutenzione e

i riavvii del broker avviati causeranno un failover. Quando si riavvia un broker, il failover dura solo qualche secondo.

Per un active/standby broker, Amazon MQ offre due URLs console Web ActiveMQ, ma è attivo un solo URL alla volta. Allo stesso modo, Amazon MQ fornisce due endpoint per ogni protocollo a livello di connessione, ma è attivo un solo endpoint per ogni coppia alla volta. I suffissi `-1` e `-2` indicano una coppia ridondante. Per gli endpoint con protocollo a livello di cavo, è necessario consentire all'applicazione di connettersi a entrambi gli endpoint utilizzando il Failover Transport.

Il diagramma seguente illustra un active/standby broker con storage Amazon EFS replicato su più piattaforme. AZs



Rete di broker Amazon MQ

Amazon MQ supporta la funzionalità di rete di broker ActiveMQ.

Una rete di broker è composta da più broker o broker a singola istanza attivi contemporaneamente. active/standby La creazione di una rete di broker può aumentare la disponibilità, la tolleranza agli errori e il bilanciamento del carico con più istanze di broker.

Come funziona una rete di broker?

Una rete di broker viene stabilita collegando un broker a un altro utilizzando connettori di rete. Un connettore di rete fornisce messaggi su richiesta da un broker all'altro. I connettori di rete sono configurati nella configurazione del broker come connessioni non duplex o duplex. Per connessioni di rete non-duplex, i messaggi vengono inoltrati solo da un broker all'altro. Per le connessioni duplex, i messaggi vengono inoltrati in entrambe le direzioni tra i due broker.

Se il connettore di rete è configurato come duplex, i messaggi vengono inoltrati anche da Broker2 a Broker1.

È possibile utilizzare connessioni non duplex e duplex in una rete di broker. Potresti voler introdurre una connessione duplex a un altro broker per migliorare il traffico o evitare un aumento del limite. Le connessioni duplex sono utili anche per la migrazione parziale da broker locali a broker gestiti da Amazon MQ.

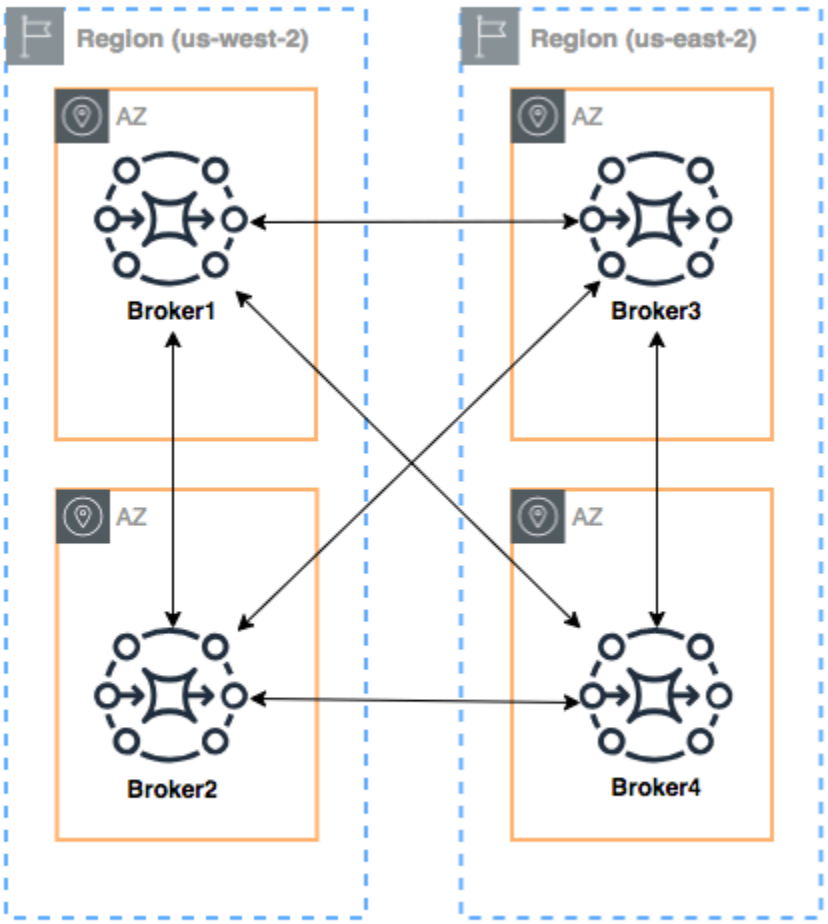
Come fa una rete di broker a gestire le credenziali?

Perché il broker A si connetta al broker B in una rete, il broker A deve usare credenziali valide, come ogni altro produttore o consumatore. Invece di fornire una password nella configurazione del `<networkConnector>` del broker A, devi prima creare un utente sul broker A con gli stessi valori di un altro utente sul broker B (questi sono utenti separati e univoci che condividono gli stessi valori di nome utente e password). Quando specifichi l'attributo `userName` nella configurazione `<networkConnector>`, Amazon MQ aggiungerà la password automaticamente in fase di esecuzione.

> **Important**
>
> Non specificare l'attributo `password` per il `<networkConnector>`. Non è consigliabile archiviare password di testo normale in file di configurazione del broker poiché questo rende le password visibili nella console Amazon MQ. Per ulteriori informazioni, consulta Configure Network Connectors for Your Broker.

Tra regioni

Per configurare una rete di broker che si estende su più AWS regioni, implementa broker in quelle regioni e configura i connettori di rete per gli endpoint di tali broker.



Per configurare una rete di broker come in questo esempio, puoi aggiungere voci `networkConnectors` alle configurazioni del Broker1 e Broker4 che facciano riferimento agli endpoint a livello di collegamento di quei broker.

Connettori di rete per Broker1:

```
<networkConnectors>
    <networkConnector name="1_to_2" userName="myCommonUser" duplex="true"
        uri="static:(ssl://b-9876l5k4-32ji-109h-8gfe-7d65c4b132a1-2.mq.us-
west-2.amazonaws.com:61617)"/>
    <networkConnector name="1_to_3" userName="myCommonUser" duplex="true"
        uri="static:(ssl://b-743c885d-2244-4c95-af67-a85017ff234e-3.mq.us-
east-2.amazonaws.com:61617)"/>
    <networkConnector name="1_to_4" userName="myCommonUser" duplex="true"
        uri="static:(ssl://b-62a7fb31-d51c-466a-a873-905cd660b553-4.mq.us-
east-2.amazonaws.com:61617)"/>
```

```
</networkConnectors>
```

Connettore di rete per Broker2:

```
<networkConnectors>
    <networkConnector name="2_to_3" userName="myCommonUser" duplex="true"
        uri="static:(ssl://b-743c885d-2244-4c95-af67-a85017ff234e-3.mq.us-
east-2.amazonaws.com:61617)"/>
</networkConnectors>
```

Connettori di rete per Broker4:

```
<networkConnectors>
    <networkConnector name="4_to_3" userName="myCommonUser" duplex="true"
        uri="static:(ssl://b-743c885d-2244-4c95-af67-a85017ff234e-3.mq.us-
east-2.amazonaws.com:61617)"/>
    <networkConnector name="4_to_2" userName="myCommonUser" duplex="true"
        uri="static:(ssl://b-9876l5k4-32ji-109h-8gfe-7d65c4b132a1-2.mq.us-
west-2.amazonaws.com:61617)"/>
</networkConnectors>
```

Failover dinamico con i connettori di trasporto

Oltre alla configurazione di elementi `networkConnector`, è possibile configurare le opzioni `transportConnector` del broker per abilitare il failover dinamico e ribilanciare le connessioni quando i broker vengono aggiunti o rimossi dalla rete.

```
<transportConnectors>
  <transportConnector name="openwire" updateClusterClients="true"
 rebalanceClusterClients="true" updateClusterClientsOnRemove="true"/>
</transportConnectors>
```

In questo esempio, `updateClusterClients` e `rebalanceClusterClients` sono impostati su `true`. In questo caso, ai client verrà fornito un elenco di broker nella rete e verrà richiesto loro di eseguire il ribilanciamento se viene aggiunto un nuovo broker.

Opzioni disponibili:

- `updateClusterClients`: trasferisce ai client le informazioni sulle modifiche della topologia dei broker nella rete.

- `rebalanceClusterClients`: causa il ribilanciamento dei client tra i broker quando un nuovo broker viene aggiunto a una rete di broker.
- `updateClusterClientsOnRemove`: aggiorna i clienti con le informazioni sulla topologia quando un broker esce dalla rete di broker.

Quando `updateClusterClients` è impostato su true, i client possono essere configurati per connettersi a un singolo broker in una rete di broker.

```
failover:(ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
east-2.amazonaws.com:61617)
```

Quando un nuovo broker si connette, riceverà un elenco di tutti i broker URIs della rete. Se la connessione al broker ha esito negativo, può dinamicamente passare a uno dei broker forniti se connessi.

Per ulteriori informazioni sul failover, vedi l'argomento relativo alle opzioni del lato broker per il failover nella documentazione di Active MQ.

Tipi di istanze del broker Amazon MQ per ActiveMQ

La descrizione combinata della classe dell'istanza del broker (`m5`) e della dimensione (`large`,`medium`) è denominata tipo di istanza del broker (ad esempio,`mq.m5.large`). La tabella seguente elenca i tipi di istanze del broker Amazon MQ disponibili per i broker ActiveMQ.

Amazon MQ fornisce un preavviso di almeno 90 giorni prima che un tipo di istanza raggiunga la fine del supporto. Ti consigliamo di aggiornare il broker a un nuovo tipo di istanza prima della end-of-support data per evitare interruzioni.

> **Important**
>
> Non puoi creare broker a partire dal `t2.micro` 17 marzo `mq.m4.large` 2025.

Tipo di istanza	VPCU	Memoria (GiB)	Uso consigliato	Storage	Fine del supporto su Amazon MQ
`mq.t3.micro`	2	1	Valutazione	EFS	
`mq.m5.large`	2	8	Produzione	EFS o EBS	
`mq.m5.xlarge`	4	16	Produzione	EFS o EBS	
`mq.m5.2xlarge`	8	32	Produzione	EFS o EBS	
`mq.m5.4xlarge`	16	64	Produzione	EFS o EBS	

Per ulteriori informazioni sulle considerazioni sul throughput, consulta Scegli il tipo di istanza broker corretta per il miglior throughput.

Configurazioni del broker Amazon MQ per ActiveMQ

Una configurazione contiene tutte le impostazioni per il broker ActiveMQ nel formato XML (simile al file `activemq.xml` di ActiveMQ). È possibile creare una configurazione prima di creare qualsiasi broker. È quindi possibile applicare la configurazione a uno o più broker.

> **Important**
>
> Apportare modifiche a una configurazione non applica le modifiche al broker in modo istantaneo. Per applicare le modifiche, attendere la finestra di manutenzione successiva o riavviare il broker.
>
> È possibile eliminare una configurazione solo utilizzando l'`DeleteConfiguration`API. Per ulteriori informazioni, consulta Configurazioni nell'Amazon MQ API Reference.

Attributes

Una configurazione del broker dispone di diversi attributi, ad esempio:

- un nome (`MyConfiguration`)
- un ID (`c-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`)
- un Amazon Resource Name (ARN) (`arn:aws:mq:us-east-2:123456789012:configuration:c-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`)

Per un elenco completo di attributi della configurazione, consultare le sezioni seguenti in Riferimento all'API REST di Amazon MQ:

- ID operazione REST: configurazione
- ID operazione REST: configurazioni

Per un elenco completo degli attributi di revisione, consulta le sezioni seguenti:

- ID operazione REST: revisione configurazione
- ID operazione REST: revisioni configurazione

Utilizzo dei file di configurazione Spring XML

I broker ActiveMQ sono configurati utilizzando file Spring XML. Puoi configurare molti aspetti del broker ActiveMQ, ad esempio destinazioni predefinite, policy di destinazione, policy di autorizzazione e plugin. Amazon MQ controlla alcuni di questi elementi di configurazione, ad esempio trasporti di rete e archiviazione. Altre opzioni di configurazione, ad esempio la creazione di reti di broker, non sono attualmente supportate.

Il set completo di opzioni di configurazione supportate è specificato negli schemi XML Amazon MQ. Scaricare i file zip degli schemi supportati utilizzando i seguenti collegamenti.

- `amazon-mq-active-mq-5.19.1.xsd.zip`
- `amazon-mq-active-mq-5.18.4.xsd.zip`
- `amazon-mq-active-mq-5.17.6.xsd.zip`
- `amazon-mq-active-mq-5.16.7.xsd.zip`

- amazon-mq-active-mq-5.15.16.xsd.zip

È possibile utilizzare questi schemi per convalidare e sterilizzare i file di configurazione. Amazon MQ consente inoltre di fornire configurazioni caricando file XML. Quando carichi un file XML, Amazon MQ sterilizza e rimuove automaticamente parametri di configurazione non validi e non consentiti in base allo schema.

> **Note**
>
> Puoi usare solo valori statici per gli attributi. Amazon MQ sterilizza elementi e attributi che contengono espressioni, variabili e riferimenti a elementi Spring dalla configurazione.

Creazione di una configurazione del broker Amazon MQ per ActiveMQ

Una configurazione contiene tutte le impostazioni per il broker ActiveMQ nel formato XML (simile al file `activemq.xml` di ActiveMQ). È possibile creare una configurazione prima di creare qualsiasi broker. È quindi possibile applicare la configurazione a uno o più broker È possibile applicare una configurazione immediatamente o durante una finestra di manutenzione.

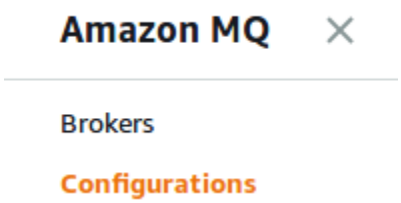
L'esempio seguente mostra come è possibile creare e applicare una configurazione del broker Amazon MQ utilizzando la Console di gestione AWS.

> **Important**
>
> È possibile eliminare una configurazione solo utilizzando l'`DeleteConfiguration`API. Per ulteriori informazioni, consulta Configurazioni nell'Amazon MQ API Reference.

Creazione di una nuova configurazione

Per creare una nuova configurazione del broker, crea prima la nuova configurazione.

1. Accedere alla console Amazon MQ.
2. Nel riquadro a sinistra, espandere il pannello di navigazione e scegliere Configurations (Configurazioni).



3. Nella pagina Configurations (Configurazioni), scegliere Create configuration (Crea configurazione).
4. Nella pagina Create configuration (Crea configurazione), nella sezione Details (Dettagli), digitare il nome in Configuration name (Nome configurazione) (ad esempio, `MyConfiguration`) e selezionare una versione Broker engine (Motore del broker).

   > **Note**
   >
   > Per ulteriori informazioni sulle versioni del motore ActiveMQ supportate da Amazon MQ per ActiveMQ, consultare the section called “Gestione della versione”.

5. Scegli Crea configurazione.

Creazione di una nuova revisione di configurazione

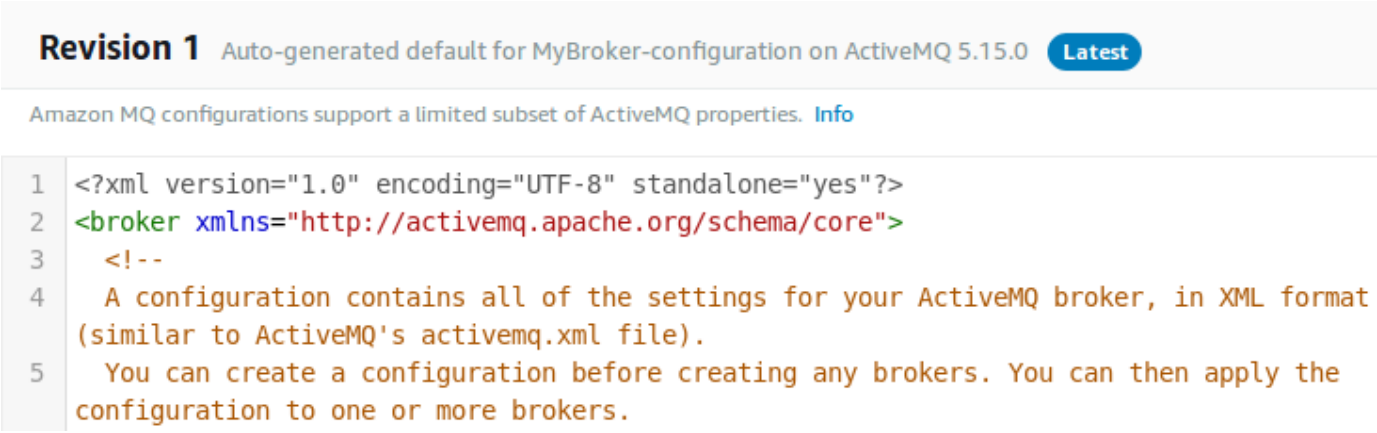
Dopo aver creato una configurazione del broker, dovrai modificare la configurazione utilizzando una revisione della configurazione.

1. Dall'elenco di configurazione, scegli ***`MyConfiguration`***.

   > **Note**
   >
   > La prima revisione di configurazione viene sempre creata automaticamente quando Amazon MQ crea la configurazione.

   `MyConfiguration`Nella pagina vengono visualizzati il tipo di motore del broker e la versione utilizzati dalla nuova revisione della configurazione (ad esempio, Apache ActiveMQ 5.15.16).
2. Nella scheda Configuration details (Dettagli configurazione) vengono visualizzati il numero di revisione della configurazione, la descrizione e la configurazione del broker in formato XML.

> **Note**
>
> La modifica della configurazione corrente crea una nuova revisione della configurazione.



3. Scegliere Edit configuration (Modifica configurazione) e apportare modifiche alla configurazione XML.
4. Scegli Save (Salva).

   Viene visualizzata la finestra di dialogo Save revision (Salva revisione).
5. (Opzionale) Digitare `A description of the changes in this revision`.
6. Scegli Save (Salva).

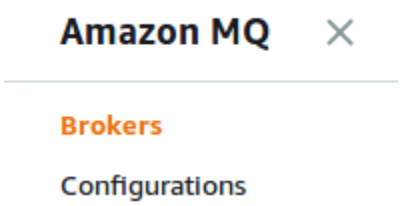
   La nuova revisione della configurazione viene salvata.

> **Important**
>
> La console Amazon MQ sterilizza automaticamente parametri di configurazione non validi e non consentiti in base a uno schema. Per ulteriori informazioni e un elenco completo dei parametri XML consentiti, consultare Amazon MQ Broker Configuration Parameters.

Applicazione di una revisione di configurazione al broker

Dopo aver modificato la configurazione, puoi applicare la revisione della configurazione al tuo broker.

1. Nel riquadro a sinistra, espandere il pannello di navigazione e scegliere Brokers (Broker).



2. Dall'elenco dei broker, seleziona il tuo broker (ad esempio MyBroker), quindi scegli Modifica.
3. Nella *MyBroker* pagina Modifica, nella sezione Configurazione, seleziona una configurazione e una revisione, quindi scegli Pianifica modifiche.
4. Nella sezione Schedule broker modifications (Pianifica modifiche broker) seleziona se applicare le modifiche During the next scheduled maintenance window (Nel corso della finestra di manutenzione pianificata successiva) oppure Immediately (Immediatamente).

> **Important**
>
> I broker a istanza singola sono offline durante il riavvio. Per i broker di cluster, durante il riavvio del broker viene interrotto solo un nodo alla volta.

5. Scegli Applica.

   La revisione della configurazione viene applicata al tuo broker nel momento specificato.

Modifica di una revisione della configurazione di Amazon MQ for ActiveMQ

Potresti voler modificare una revisione della configurazione dopo averla applicata al tuo broker. Utilizza le seguenti istruzioni per modificare una revisione della configurazione.

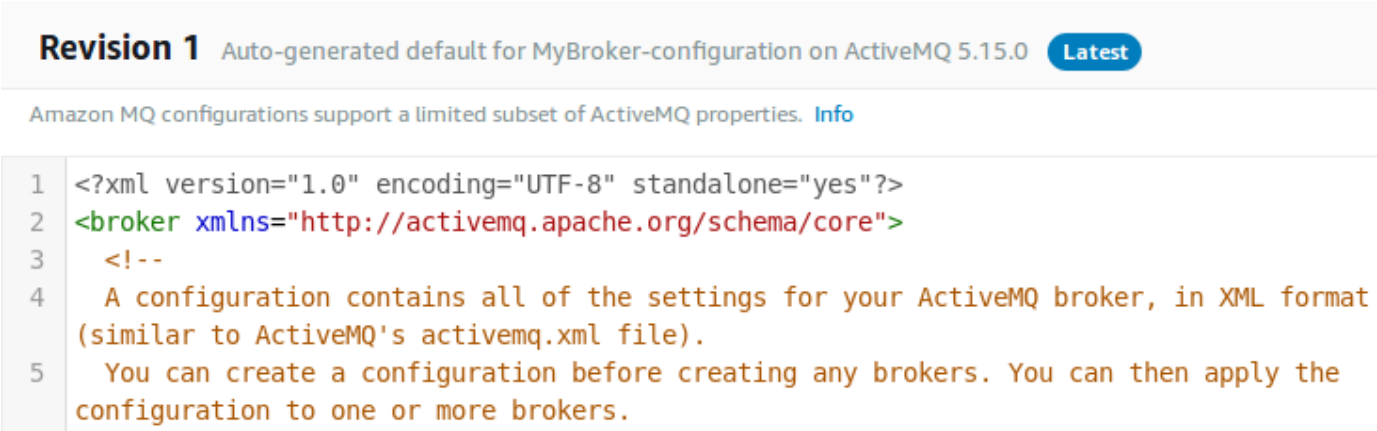
1. Accedere alla console Amazon MQ.
2. Dall'elenco dei broker, seleziona il tuo broker (ad esempio MyBroker), quindi scegli Modifica.
3. Nella ***MyBroker***pagina, scegli Modifica.
4. Nella *MyBroker* pagina Modifica, nella sezione Configurazione, seleziona una configurazione e una revisione, quindi scegli Modifica.

> **Note**
>
> A meno che non si selezioni una configurazione durante la creazione di un broker, la prima revisione della configurazione viene sempre creata quando Amazon MQ crea il broker.

MyBrokerNella pagina vengono visualizzati il tipo e la versione del motore del broker utilizzati dalla configurazione (ad esempio, Apache ActiveMQ 5.15.8).

5. Nella scheda Configuration details (Dettagli configurazione) vengono visualizzati il numero di revisione della configurazione, la descrizione e la configurazione del broker in formato XML.

> **Note**
>
> La modifica della configurazione corrente crea una nuova revisione della configurazione.



6. Scegliere Edit configuration (Modifica configurazione) e apportare modifiche alla configurazione XML.
7. Scegli Save (Salva).

   Viene visualizzata la finestra di dialogo Save revision (Salva revisione).
8. (Opzionale) Digitare `A description of the changes in this revision`.
9. Scegli Save (Salva).

   La nuova revisione della configurazione viene salvata.

> **Important**
>
> La console Amazon MQ sterilizza automaticamente parametri di configurazione non validi e non consentiti in base a uno schema. Per ulteriori informazioni e un elenco completo dei parametri XML consentiti, consultare Amazon MQ Broker Configuration Parameters.

Elementi consentiti nelle configurazioni Amazon MQ

Di seguito è riportato un elenco dettagliato degli elementi consentiti in configurazioni Amazon MQ. Per ulteriori informazioni, consulta XML Configuration nella documentazione di Apache ActiveMQ.

Elemento
`abortSlowAckConsumerStrategy` (attributi)
`abortSlowConsumerStrategy` (attributi)
`authorizationEntry` (attributi)
`authorizationMap` (elementi di raccolta figlio)
`authorizationPlugin` (elementi di raccolta figlio)
`broker` (attributi | elementi di raccolta figlio)
`cachedMessageGroupMapFactory` (attributi)
`compositeQueue` (attributi | elementi di raccolta figlio)
`compositeTopic` (attributi | elementi di raccolta figlio)
`constantPendingMessageLimitStrategy` (attributi)
`discarding` (attributi)
`discardingDLQBrokerPlugin` (attributi)

Elemento
`fileCursor`
`fileDurableSubscriberCursor`
`fileQueueCursor`
`filteredDestination` (attributi)
`fixedCountSubscriptionRecoveryPolicy` (attributi)
`fixedSizedSubscriptionRecoveryPolicy` (attributi)
`forcePersistencyModeBrokerPlugin` (attributi)
`individualDeadLetterStrategy` (attributi)
`lastImageSubscriptionRecoveryPolicy`
`messageGroupHashBucketFactory` (attributi)
`mirroredQueue` (attributi)
`noSubscriptionRecoveryPolicy`
`oldestMessageEvictionStrategy` (attributi)
`oldestMessageWithLowestPriorityEvictionStrategy` (attributi)
`policyEntry` (attributi | elementi di raccolta figlio)
`policyMap` (attributi | elementi di raccolta figlio)
`prefetchRatePendingMessageLimitStrategy` (attributi)
`priorityDispatchPolicy`
`priorityNetworkDispatchPolicy`
`queryBasedSubscriptionRecoveryPolicy` (attributi)

Elemento
queue (attributi)
redeliveryPlugin (attributi | elementi di raccolta figlio)
redeliveryPolicy (attributi)
redeliveryPolicyMap (elementi di raccolta figlio)
retainedMessageSubscriptionRecoveryPolicy (elementi di raccolta figlio)
roundRobinDispatchPolicy
sharedDeadLetterStrategy (attributi | elementi di raccolta figlio)
simpleDispatchPolicy
simpleMessageGroupMapFactory
statisticsBrokerPlugin
storeCursor
storeDurableSubscriberCursor (attributi)
strictOrderDispatchPolicy
tempDestinationAuthorizationEntry (attributi)
tempQueue (attributi)
tempTopic (attributi)
timedSubscriptionRecoveryPolicy (attributi)
timeStampingBrokerPlugin (attributi)
topic (attributi)
transportConnector (attributi)

Elemento	
uniquePropertyMessageEvictionStrategy	(attributi)
virtualDestinationInterceptor	(attributi | elementi di raccolta figlio)
virtualTopic	(attributi)
vmCursor	
vmDurableCursor	
vmQueueCursor	

Elementi e relativi attributi consentiti nelle configurazioni Amazon MQ

Di seguito è riportato un elenco dettagliato degli elementi e dei relativi attributi consentiti nelle configurazioni Amazon MQ. Per ulteriori informazioni, consulta XML Configuration nella documentazione di Apache ActiveMQ.

Elemento	Attributo
abortSlowAckConsumerStrategy	abortConnection
	checkPeriod
	ignoreIdleConsumers
	ignoreNetworkConsumers
	maxSlowCount
	maxSlowDuration
	maxTimeSinceLastAck
	name
abortSlowConsumerStrategy	abortConnection

Elemento	Attributo
	checkPeriod
	ignoreNetworkConsumers
	maxSlowCount
	maxSlowDuration
	name
authorizationEntry	admin
	queue
	read
	tempQueue
	tempTopic
	topic
	write
broker	advisorySupport
	allowTempAutoCreationOnSend
	cacheTempDestinations
	consumerSystemUsagePortion
	dedicatedTaskRunner
	deleteAllMessagesOnStartup
	keepDurableSubsActive
	enableMessageExpirationOnActiveDurableSubs

Elemento	Attributo
	maxPurgedDestinationsPerSweep
	maxSchedulerRepeatAllowed
	monitorConnectionSplits
	networkConnectorStartAsync
	offlineDurableSubscriberTaskSchedule
	offlineDurableSubscriberTimeout
	persistenceThreadPriority
	persistent
	populateJMSXUserID
	producerSystemUsagePortion
	rejectDurableConsumers
	rollbackOnlyOnAsyncException
	schedulePeriodForDestinationPurge
	schedulerSupport
	splitSystemUsageForProducersConsumers
	taskRunnerPriority
	timeBeforePurgeTempDestinations
	useAuthenticatedPrincipalForJMSXUserID

Elemento	Attributo
	useMirroredQueues
	useTempMirroredQueues
	useVirtualDestSubs
	useVirtualDestSubsOnCreation
	useVirtualTopics
cachedMessageGroupMapFactory	cacheSize
compositeQueue	concurrentSend
	copyMessage
	forwardOnly
	name
	sendWhenNotMatched
compositeTopic	concurrentSend
	copyMessage
	forwardOnly
	name
	sendWhenNotMatched
conditionalNetworkBridgeFilterFactory	rateDuration
	rateLimit
	replayDelay
	replayWhenNoConsumers

Elemento	Attributo
	`selectorAware` Supportato in Apache ActiveMQ 5.16.x
`constantPendingMessageLimitStrategy`	`limit`
`discarding`	`deadLetterQueue`
	`enableAudit`
	`expiration`
	`maxAuditDepth`
	`maxProducersToAudit`
	`processExpired`
	`processNonPersistent`
`discardingDLQBrokerPlugin`	`dropAll`
	`dropOnly`
	`dropTemporaryQueues`
	`dropTemporaryTopics`
	`reportInterval`
`filteredDestination`	`queue`
	`selector`
	`topic`

Elemento	Attributo
fixedCountSubscriptionRecoveryPolicy	maximumSize
fixedSizedSubscriptionRecoveryPolicy	maximumSize
	useSharedBuffer
forcePersistencyModeBrokerPlugin	persistenceFlag
individualDeadLetterStrategy	destinationPerDurableSubscriber
	enableAudit
	expiration
	maxAuditDepth
	maxProducersToAudit
	processExpired
	processNonPersistent
	queuePrefix
	queueSuffix
	topicPrefix
	topicSuffix
	useQueueForQueueMessages
	useQueueForTopicMessages
messageGroupHashBucketFactory	bucketCount
	cacheSize
mirroredQueue	copyMessage

Elemento	Attributo
	postfix
	prefix
oldestMessageEvictionStrategy	evictExpiredMessagesHighWat ermark
oldestMessageWithLowestPrio rityEvictionStrategy	evictExpiredMessagesHighWat ermark
policyEntry	advisoryForConsumed
	advisoryForDelivery
	advisoryForDiscardingMessages
	advisoryForFastProducers
	advisoryForSlowConsumers
	advisoryWhenFull
	allConsumersExclusiveByDefault
	alwaysRetroactive
	blockedProducerWarningInterval
	consumersBeforeDispatchStarts
	cursorMemoryHighWaterMark
	doOptimzeMessageStorage
	durableTopicPrefetch
	enableAudit
	expireMessagesPeriod

Elemento	Attributo
	gcInactiveDestinations
	gcWithNetworkConsumers
	inactiveTimeoutBeforeGC
	inactiveTimoutBeforeGC
	includeBodyForAdvisory
	lazyDispatch
	maxAuditDepth
	maxBrowsePageSize
	maxDestinations
	maxExpirePageSize
	maxPageSize
	maxProducersToAudit
	maxQueueAuditDepth
	memoryLimit
	messageGroupMapFactoryType
	minimumMessageSize
	optimizedDispatch
	optimizeMessageStoreInFligh tLimit
	persistJMSRedelivered
	prioritizedMessages

Elemento	Attributo
	`producerFlowControl`
	`queue`
	`queueBrowserPrefetch`
	`queuePrefetch`
	`reduceMemoryFootprint`
	`sendAdvisoryIfNoConsumers`
	`sendFailIfNoSpace`
	`sendFailIfNoSpaceAfterTimeout` Supportato in Apache ActiveMQ 5.16.4 e versioni successive
	`sendDuplicateFromStoreToDLQ`
	`storeUsageHighWaterMark`
	`strictOrderDispatch`
	`tempQueue`
	`tempTopic`
	`timeBeforeDispatchStarts`
	`topic`
	`topicPrefetch`
	`useCache`

Elemento	Attributo
	useConsumerPriority
usePrefetchExtension	
prefetchRatePendingMessageLimitStrategy	multiplier
queryBasedSubscriptionRecoveryPolicy	query
queue	DLQ
	physicalName
redeliveryPlugin	fallbackToDeadLetter
	sendToDlqIfMaxRetriesExceeded
redeliveryPolicy	backOffMultiplier
	collisionAvoidancePercent
	initialRedeliveryDelay
	maximumRedeliveries
	maximumRedeliveryDelay
	preDispatchCheck
	queue
	redeliveryDelay
	tempQueue
	tempTopic
	topic

Elemento	Attributo
	useCollisionAvoidance
	useExponentialBackOff
sharedDeadLetterStrategy	enableAudit
	expiration
	maxAuditDepth
	maxProducersToAudit
	processExpired
	processNonPersistent
storeDurableSubscriberCursor	immediatePriorityDispatch
	useCache
tempDestinationAuthorizationEntry	admin
	queue
	read
	tempQueue
	tempTopic
	topic
	write
tempQueue	DLQ
	physicalName
tempTopic	DLQ

Elemento	Attributo
	physicalName
timedSubscriptionRecoveryPolicy	zeroExpirationOverride
timeStampingBrokerPlugin	recoverDuration
	futureOnly
	processNetworkMessages
	ttlCeiling
topic	DLQ
	physicalName
transportConnector	name
	updateClusterClients
	rebalanceClusterClients
	updateClusterClientsOnRemove
uniquePropertyMessageEvictionStrategy	evictExpiredMessagesHighWatermark
	propertyName
virtualTopic	concurrentSend
	local
	dropOnResourceLimit
	name
	postfix
	prefix

Elemento	Attributo
	`selectorAware`
	`setOriginalDestination`
	`transactedSend`

Attributi elemento padre Amazon MQ

Di seguito è riportata una descrizione dettagliata degli attributi dell'elemento padre. Per ulteriori informazioni, consulta XML Configuration nella documentazione di Apache ActiveMQ.

Argomenti

- broker

broker

`broker` è un elemento di raccolta padre.

Attributes

networkConnectionStartAsincrono

Per ridurre la latenza di rete e per consentire ad altre reti di avviarsi in modo tempestivo, utilizza il tag `<networkConnectionStartAsync>`. Il tag indica al broker di utilizzare un esecutore per avviare le connessioni di rete in parallelo, asincrone a un avvio di broker.

Default: `false`

Configurazione di esempio

```
<broker networkConnectorStartAsync="false"/>
```

Elementi, elementi della raccolta figlio e relativi elementi figlio consentiti nelle configurazioni Amazon MQ

Di seguito è riportato un elenco dettagliato degli elementi, degli elementi della raccolta figlio e dei relativi elementi figlio consentiti nelle configurazioni Amazon MQ. Per ulteriori informazioni, consulta XML Configuration nella documentazione di Apache ActiveMQ.

Elemento	Elemento della raccolta figlio	Elemento figlio
`authorizationMap`	`authorizationEntries`	`authorizationEntry`
		`tempDestinationAuthorizationEntry`
	`defaultEntry`	`authorizationEntry`
		`tempDestinationAuthorizationEntry`
	`tempDestinationAuthorizationEntry`	`tempDestinationAuthorizationEntry`
`authorizationPlugin`	`map`	`authorizationMap`
`broker`	`destinationInterceptors`	`mirroredQueue`
		`virtualDestinationInterceptor`
	`destinationPolicy`	`policyMap`
	`destinations`	`queue`
		`tempQueue`
		`tempTopic`
		`topic`
	`networkConnectors`	`networkConnector`

Elemento	Elemento della raccolta figlio	Elemento figlio
	persistenceAdapter	kahaDB
	plugins	authorizationPlugin
		discardingDLQBroke rPlugin
		forcePersistencyMo deBrokerPlugin
		redeliveryPlugin
		statisticsBrokerPl ugin
		timeStampingBroker Plugin
	systemUsage	systemUsage
	transportConnector	name
		updateClusterClients
		rebalanceClusterCl ients
		updateClusterClien tsOnRemove
compositeQueue	forwardTo	queue
		tempQueue
		tempTopic
		topic
		filteredDestination

Elemento	Elemento della raccolta figlio	Elemento figlio
compositeTopic	forwardTo	queue
		tempQueue
		tempTopic
		topic
		filteredDestination
policyEntry	deadLetterStrategy	discarding
		individualDeadLetterStrategy
		sharedDeadLetterStrategy
	destination	queue
		tempQueue
		tempTopic
		topic
	dispatchPolicy	priorityDispatchPolicy
		priorityNetworkDispatchPolicy
		roundRobinDispatchPolicy
		simpleDispatchPolicy
		strictOrderDispatchPolicy

Elemento	Elemento della raccolta figlio	Elemento figlio
		clientIdFilterDispatchPolicy
	messageEvictionStrategy	oldestMessageEvictionStrategy
		oldestMessageWithLowestPriorityEvictionStrategy
		uniquePropertyMessageEvictionStrategy
	messageGroupMapFactory	cachedMessageGroupMapFactory
		messageGroupHashBucketFactory
		simpleMessageGroupMapFactory
	pendingDurableSubscriberPolicy	fileDurableSubscriberCursor
		storeDurableSubscriberCursor
		vmDurableCursor
	pendingMessageLimitStrategy	constantPendingMessageLimitStrategy
		prefetchRatePendingMessageLimitStrategy
	pendingQueuePolicy	fileQueueCursor

Elemento	Elemento della raccolta figlio	Elemento figlio
		storeCursor
		vmQueueCursor
	pendingSubscriberPolicy	fileCursor
		vmCursor
	slowConsumerStrategy	abortSlowAckConsumerStrategy
		abortSlowConsumerStrategy
	subscriptionRecoveryPolicy	fixedCountSubscriptionRecoveryPolicy
		fixedSizedSubscriptionRecoveryPolicy
		lastImageSubscriptionRecoveryPolicy
		noSubscriptionRecoveryPolicy
		queryBasedSubscriptionRecoveryPolicy
		retainedMessageSubscriptionRecoveryPolicy
timedSubscriptionRecoveryPolicy		
policyMap	defaultEntry	policyEntry

Elemento	Elemento della raccolta figlio	Elemento figlio
	policyEntries	policyEntry
redeliveryPlugin	redeliveryPolicyMap	redeliveryPolicyMap
redeliveryPolicyMap	defaultEntry	redeliveryPolicy
	redeliveryPolicyEntries	redeliveryPolicy
retainedMessageSubscriptionRecoveryPolicy	wrapped	fixedCountSubscriptionRecoveryPolicy
		fixedSizedSubscriptionRecoveryPolicy
		lastImageSubscriptionRecoveryPolicy
		noSubscriptionRecoveryPolicy
		queryBasedSubscriptionRecoveryPolicy
		retainedMessageSubscriptionRecoveryPolicy
		timedSubscriptionRecoveryPolicy
sharedDeadLetterStrategy	deadLetterQueue	queue
		tempQueue
		tempTopic
		topic

Elemento	Elemento della raccolta figlio	Elemento figlio
`virtualDestination Interceptor`	`virtualDestinations`	`compositeQueue`
		`compositeTopic`
		`virtualTopic`

Attributi elemento figlio Amazon MQ

Di seguito è riportata una descrizione dettagliata degli attributi elemento figlio. Per ulteriori informazioni, consulta XML Configuration nella documentazione di Apache ActiveMQ.

Argomenti

- authorizationEntry
- networkConnector
- kahaDB
- systemUsage

authorizationEntry

`authorizationEntry` è un figlio dell'elemento raccolta figlio `authorizationEntries`.

Attributes

admin|read|write

Le autorizzazioni concesse a un gruppo di utenti. Per ulteriori informazioni, consulta Configurare sempre una mappa di autorizzazione.

Se si specifica una mappa di autorizzazione che non include il gruppo `activemq-webconsole`, non è possibile utilizzare la console Web ActiveMQ perché il gruppo non è autorizzato a inviare o ricevere messaggi dal broker Amazon MQ.

Default: `null`

Configurazione di esempio

```
<authorizationPlugin>
```

```
                                <map>
                                    <authorizationMap>
                                        <authorizationEntries>
                                            <authorizationEntry admin="admins,activemq-
webconsole" read="admins,users,activemq-webconsole" write="admins,activemq-webconsole"
 queue=">"/>
                                            <authorizationEntry admin="admins,activemq-
webconsole" read="admins,users,activemq-webconsole" write="admins,activemq-webconsole"
 topic=">"/>
                                        </authorizationEntries>
                                    </authorizationMap>
                                </map>
                                </authorizationPlugin>
```

> **Note**
>
> Il `activemq-webconsole` gruppo in ActiveMQ su Amazon MQ dispone delle autorizzazioni di amministratore per tutte le code e gli argomenti. Tutti gli utenti di questo gruppo avranno accesso come amministratore.

networkConnector

`networkConnector` è un figlio dell'elemento raccolta figlio `networkConnectors`.

Argomenti

- Attributes
- Configurazioni di esempio

Attributes

conduitSubscriptions

Specifica se una connessione di rete in una rete di broker considera più consumatori sottoscritti alla stessa destinazione come un singolo consumatore. Ad esempio, se `conduitSubscriptions` è impostato su `true` e due consumatori si connettono al broker B e consumano da una destinazione, il broker B combina le sottoscrizioni in una singola sottoscrizione logica per la connessione di rete al broker A, in modo che solo una singola copia di un messaggio viene inoltrata dal broker A al broker B.

> **Note**
>
> Impostare `conduitSubscriptions` su `true` può ridurre il traffico di rete ridondante. Tuttavia, utilizzare questo attributo può avere implicazioni per il bilanciamento del carico di messaggi tra i consumatori e potrebbe causare un comportamento errato in alcune situazioni (ad esempio con selettori di messaggi JMS o con argomenti durevoli).

Default: `true`

duplex

Specifica se la connessione nella rete di broker è utilizzata per generare e consumare i messaggi. Ad esempio, se il broker A crea una connessione al broker B in modalità non-duplex, i messaggi possono essere inoltrati solo dal broker A al broker B. Tuttavia, se il broker A crea una connessione duplex verso il broker B, allora il broker B è in grado di inoltrare messaggi al broker A senza dover configurare un `<networkConnector>`.

Default: `false`

nome

Il nome del bridge nella rete di broker.

Default: `bridge`

uri

L'endpoint con protocollo a livello di collegamento per uno dei due (o più) broker in una rete di broker.

Default: `null`

username

Il nome utente comune ai broker in una rete di broker.

Default: `null`

Configurazioni di esempio

> **Note**
>
> Quando utilizzi un `networkConnector` per definire una rete di broker, non includere la password dell'utente comune ai broker.

Una rete di broker con due broker

In questa configurazione, due broker sono connessi in una rete di broker. Il nome del connettore di rete è`connector_1_to_2`, il nome utente comune ai broker è`myCommonUser`, la connessione è`duplex`, e l'URI dell' OpenWire endpoint è preceduto da`static:`, indica una one-to-one connessione tra i broker.

```
<networkConnectors>
                                    <networkConnector name="connector_1_to_2"
 userName="myCommonUser" duplex="true"
                                        uri="static:(ssl://
b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:61617)"/>
                                    </networkConnectors>
```

Per ulteriori informazioni, consulta Configure Network Connectors for Your Broker.

Una rete di broker con più broker

In questa configurazione, broker multipli sono connessi in una rete di broker. Il nome del connettore di rete è`connector_1_to_2`, il nome utente comune ai broker è, la connessione è`myCommonUser`, e l'elenco degli OpenWire endpoint separati da virgole URIs è preceduto da`duplex`, che indica una connessione di failover tra i broker. `masterslave:` Il failover da broker a broker non è randomizzato e i tentativi di riconnessione continuano a tempo indeterminato.

```
<networkConnectors>
                                    <networkConnector name="connector_1_to_2"
 userName="myCommonUser" duplex="true"
                                        uri="masterslave:(ssl://
b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:61617,
                                        ssl://
b-9876l5k4-32ji-109h-8gfe-7d65c4b132a1-2.mq.us-west-2.amazonaws.com:61617)"/>
                                    </networkConnectors>
```

> **Note**
>
> Ti consigliamo di usare il prefisso `masterslave:` per reti di broker. Il prefisso è identico alla sintassi `static:failover:()?randomize=false&maxReconnectAttempts=0` più esplicita.

> **Note**
>
> Questa configurazione XML non consente spazi.

kahaDB

`kahaDB` è un figlio dell'elemento raccolta figlio `persistenceAdapter`.

Attributes

concurrentStoreAndDispatchQueues

Specifica se utilizzare archiviazione e invio simultanei per le code. Per ulteriori informazioni, consulta Disabilita archiviazione e invio simultaneo per code con consumatori lenti.

Default: `true`

cleanupOnStop

> **Supportato in**
>
> Apache ActiveMQ 15.16.x e versioni successive

Se disattivato, la garbage collection (GC) e la pulizia non hanno luogo quando il broker viene terminato, il che accelera il processo di arresto. La maggiore velocità è utile nei casi con database di grandi dimensioni o database del pianificatore.

Default: `true`

journalDiskSyncIntervallo

Intervallo (ms) per quando eseguire una sincronizzazione del disco se `journalDiskSyncStrategy=periodic`. Per ulteriori informazioni, vedere la documentazione di Apache ActiveMQ KahaDB.

Default: `1000`

journalDiskSyncStrategia

> Supportato in
>
> Apache ActiveMQ 15.14.x e versioni successive

Configura la policy di sincronizzazione del disco. Per ulteriori informazioni, vedere la documentazione di Apache ActiveMQ KahaDB.

Default: `always`

> Note
>
> La documentazione di ActiveMQ indica che la perdita di dati è limitata alla durata di `journalDiskSyncInterval`, che ha un valore predefinito di 1s. La perdita di dati può essere più lunga dell'intervallo, ma è difficile essere precisi. Prestare attenzione.

preallocationStrategy

Configura il modo in cui il broker tenterà di preallocare i file journal quando è necessario un nuovo file journal. Per ulteriori informazioni, vedere la documentazione di Apache ActiveMQ KahaDB.

Default: `sparse_file`

Configurazione di esempio

Example

```
<broker xmlns="http://activemq.apache.org/schema/core">
```

```
                                <persistenceAdapter>
                                <kahaDB preallocationStrategy="zeros"
concurrentStoreAndDispatchQueues="false" journalDiskSyncInterval="10000"
journalDiskSyncStrategy="periodic"/>
                            </persistenceAdapter>
                            </broker>
```

systemUsage

`systemUsage` è un figlio dell'elemento raccolta figlio `systemUsage`. Controlla la quantità massima di spazio che il broker utilizzerà prima di rallentare i produttori. Per ulteriori informazioni, vedere Producer Flow Control nella documentazione di Apache ActiveMQ.

Elemento figlio

memoryUsage

`memoryUsage` è un figlio dell'elemento `systemUsage` figlio. Gestisce l'utilizzo della memoria. Utilizzare `memoryUsage` per tenere traccia di quanto di qualcosa viene utilizzato in modo da poter controllare l'utilizzo del working set in modo produttivo. Per ulteriori informazioni, consulta lo schema nella documentazione di Apache ActiveMQ.

Elemento figlio

`memoryUsage` è un figlio dell'elemento `memoryUsage` figlio.

Attributo

percentOfJvmMucchio

Numero intero compreso tra 0 (incluso) e 70 (incluso).

Default: `70`

Attributes

sendFailIfNoSpace

Imposta se un metodo `send()` deve fallire se non c'è spazio libero. Il valore predefinito è false, che blocca il metodo `send()` fino a quando lo spazio non diventa disponibile. Per ulteriori informazioni, vedere lo schema nella documentazione di Apache Active MQ.

Default: `false`

sendFailIfNoSpaceAfterTimeout

Default: `null`

Configurazione di esempio

Example

```
<broker xmlns="http://activemq.apache.org/schema/core">
                                  <systemUsage>
                                  <systemUsage sendFailIfNoSpace="true"
 sendFailIfNoSpaceAfterTimeout="2000">
                                      <memoryUsage>
                                          <memoryUsage  percentOfJvmHeap="60" />
                                      </memoryUsage>>
                                  </systemUsage>
                                  </systemUsage>
                              </broker>
                              </persistenceAdapter>
```

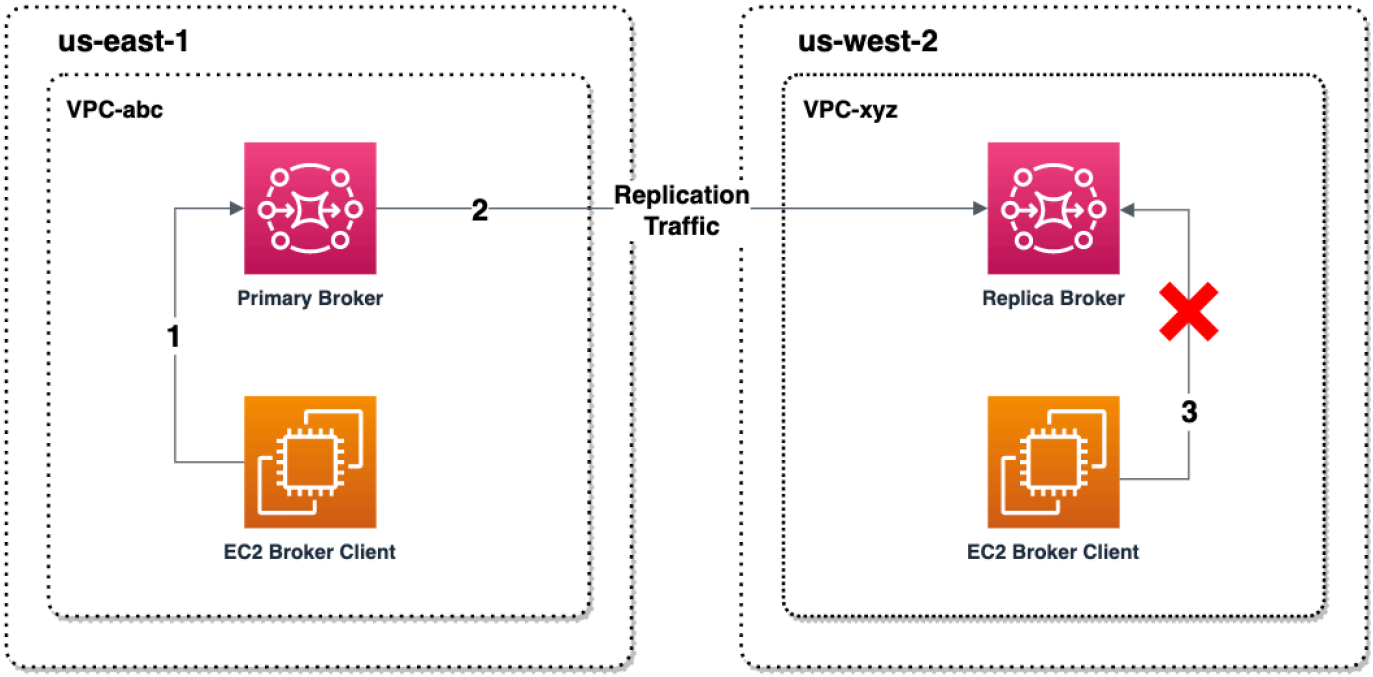
Replica dei dati tra regioni per Amazon MQ per ActiveMQ

Amazon MQ for ActiveMQ offre una funzionalità di replica dei dati tra regioni (CRDR) che consente la replica asincrona dei messaggi dal broker principale in una regione primaria al broker di replica in una AWS regione di replica. Inviando una richiesta di failover all'API Amazon MQ, l'attuale broker di replica viene promosso al ruolo di broker primario e l'attuale broker primario viene retrocesso al ruolo di replica.

Broker primari e di replica per la replica dei dati tra regioni

È possibile creare broker primari e di replica per la replica asincrona dei dati dal broker principale in una regione primaria al broker di replica in una regione di replica. AWS La regione primaria è costituita da una coppia ridondante di broker attivi/in standby denominata broker primario. La regione secondaria è costituita da una coppia ridondante di broker attivi/in standby denominata broker di replica.

Il diagramma seguente illustra un broker di replica in una regione secondaria che riceve dati replicati asincroni dal broker primario nella regione primaria.



I broker primari e di replica fungono da soluzione di ripristino dei dati tra regioni. Se il broker primario nella regione primaria restituisce un errore, è possibile promuovere il broker di replica nella regione secondaria a primario avviando uno switchover o un failover. Il precedente broker primario diventa quindi il broker di replica e il precedente broker di replica viene promosso a broker primario. Per istruzioni sulla creazione di un broker primario e di replica, consulta Creazione di un broker di replica dei dati interregionale Amazon MQ.

> **Note**
>
> Disponibile solo per broker attivi o in standby.
> Non disponibile per le code con mirroring.

Creazione di un broker di replica dei dati interregionale Amazon MQ

Con la replica dei dati tra regioni (CRDR), puoi passare da un broker di messaggi Amazon MQ per ActiveMQ all'altro in due regioni AWS, se necessario. Puoi designare un broker esistente come broker primario e creare una replica per questo broker oppure creare insieme un nuovo broker primario e di replica. Puoi quindi promuovere il broker di replica al ruolo di broker primario utilizzando l'operazione API `Promote` di Amazon MQ. Per ulteriori informazioni sui broker primari e di replica, consulta Broker primari e di replica per la replica dei dati tra regioni.

Le seguenti istruzioni descrivono come creare e configurare un broker di replica utilizzando la Console di gestione Amazon MQ.

Argomenti

- Prerequisiti
- Fase 1 (opzionale): Creazione di un nuovo broker primario
- Fase 2: Creazione di una replica di un broker esistente

Prerequisiti

Per utilizzare la funzionalità di replica dei dati tra aree geografiche, devi esaminare e rispettare i seguenti prerequisiti:

- Versione: la funzionalità di replica dei dati tra regioni è disponibile solo per i broker Amazon MQ per ActiveMQ nelle versioni 5.17.6 e successive.
- Regione: la replica dei dati tra regioni è supportata nelle seguenti regioni: Stati Uniti orientali (Ohio), Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon) e Stati Uniti occidentali (California settentrionale).
- Tipo di istanza: la replica dei dati tra regioni è disponibile solo per istanze di broker di dimensioni `mq.m5.large` o superiori.
- Tipo di implementazione: la replica dei dati tra regioni è disponibile solo per broker attivi/in standby con implementazione in più zone di disponibilità.
- Stato del broker: puoi creare un broker di replica solo per un broker principale con lo stato del broker `Running`.

Fase 1 (opzionale): Creazione di un nuovo broker primario

Creazione di un nuovo broker primario

1. Accedere alla console Amazon MQ.
2. Nella pagina Broker della console Amazon MQ, scegli Crea broker.
3. Alla pagina Select broker engine (Seleziona motore del broker), scegliere Apache ActiveMQ.
4. Alla pagina Select deployment and storage (Seleziona implementazione e archiviazione), nella sezione Deployment mode and storage type (Modalità di implementazione e tipo di archiviazione), procedere come segue:

- Per Modalità distribuzione, scegli Broker attivo/in standby. Un Broker attivo/in standby è composto da due broker in due diverse zone di disponibilità, configurate in una coppia ridondante. Questi broker comunicano in modo sincrono con l'applicazione e con Amazon EFS. Per ulteriori informazioni, consulta Opzioni di implementazione per i broker Amazon MQ for ActiveMQ.

5. Scegli Next (Successivo).
6. Alla pagina Configure settings (Configura impostazioni), nella sezione Details (Dettagli), procedere come segue:

   a. Inserisci il nome del broker.

   > **Important**
   >
   > Non aggiungere informazioni personali di identificazione (PII) o altre informazioni riservate o sensibili nei nomi dei broker. I nomi dei broker sono accessibili ad altri AWS servizi, inclusi CloudWatch i registri. I nomi dei broker non sono destinati ad essere utilizzati per dati privati o sensibili.

   b. Selezionare il tipo di istanza del broker (ad esempio, mq.m5.large). Per ulteriori informazioni, consulta Broker instance types.

7. Nella sezione ActiveMQ Web Console access (Accesso alla console Web di ActiveMQ), specificare nome utente e password. Per i nomi utente e le password del broker si applicano le seguenti limitazioni:

   - Il nome utente può contenere solo caratteri alfanumerici, trattini, punti, caratteri di sottolineature e tilde (- . _ ~).
   - La password deve contenere almeno 12 caratteri, di cui almeno 4 caratteri univoci, e non deve contenere virgole, due punti o il simbolo dell'uguale (,:=).

   > **Important**
   >
   > Non aggiungere informazioni personali di identificazione (PII) o altre informazioni riservate o sensibili nei nomi utente dei broker. I nomi utente dei broker sono accessibili ad altri AWS servizi, inclusi CloudWatch i registri. I nomi utenti dei broker non sono destinati ad essere utilizzati per dati privati o sensibili.

La barra flash verde nella parte superiore della pagina conferma che Amazon MQ sta creando il broker di replica nella regione di ripristino. Puoi anche vedere il ruolo CRDR e lo stato RPO dei tuoi broker. Per disattivare le colonne del ruolo CRDR e dello stato RPO, scegli l'icona a forma di ingranaggio nell'angolo in alto a destra della tabella Broker. Quindi, nella pagina Preferenze, disattiva Ruolo CRDR o Stato RPO.

Fase 2: Creazione di una replica di un broker esistente

1. Nella pagina Broker della console Amazon MQ, scegli Creare broker di replica.
2. Nella pagina Scegli broker primario, seleziona un broker esistente da utilizzare come broker primario CRDR. Quindi, seleziona Next (Successivo).
3. Nella pagina Configura broker di replica, utilizza il menu a tendina per scegliere la regione di replica.
4. Nella sezione Utente della console ActiveMQ per il broker di replica, fornisci un Nome utente e una Password per l'utente della console del broker di replica. Per i nomi utente e le password del broker si applicano le seguenti limitazioni:

   - Il nome utente può contenere solo caratteri alfanumerici, trattini, punti, caratteri di sottolineature e tilde (- . _ ~).
   - La password deve contenere almeno 12 caratteri, di cui almeno 4 caratteri univoci, e non deve contenere virgole, due punti o il simbolo dell'uguale (,:=).

   > **Important**
   >
   > Non aggiungere informazioni personali di identificazione (PII) o altre informazioni riservate o sensibili nei nomi utente dei broker. I nomi utente dei broker sono accessibili ad altri AWS servizi, inclusi i registri. CloudWatch I nomi utenti dei broker non sono destinati ad essere utilizzati per dati privati o sensibili.

5. Nella sezione Utente dei dati di replica per collegare l'accesso tra broker, fornisci un Nome utente e una Password per l'utente che accederà sia al broker principale che a quello di replica. Per i nomi utente e le password del broker si applicano le seguenti limitazioni:

   - Il nome utente può contenere solo caratteri alfanumerici, trattini, punti, caratteri di sottolineature e tilde (- . _ ~).
   - La password deve contenere almeno 12 caratteri, di cui almeno 4 caratteri univoci, e non deve contenere virgole, due punti o il simbolo dell'uguale (,:=).

> **Important**
>
> Non aggiungere informazioni personali di identificazione (PII) o altre informazioni riservate o sensibili nei nomi utente dei broker. I nomi utente dei broker sono accessibili ad altri AWS servizi, inclusi i registri. CloudWatch I nomi utenti dei broker non sono destinati ad essere utilizzati per dati privati o sensibili.

Configurazione di eventuali impostazioni aggiuntive. Quindi, seleziona Next (Successivo).

6. Nella pagina Verifica e crea, esamina i dettagli del broker di replica. Quindi, scegli Crea broker di replica.
7. Quindi, riavvia il broker primario. Questa operazione riavvierà anche il broker di replica. Per istruzioni su come riavviare il broker, consulta. Rebooting a Broker

Per ulteriori informazioni sulla configurazione delle impostazioni aggiuntive per il broker ActiveMQ, consulta Guida introduttiva: creazione e connessione a un broker ActiveMQ

Eliminazione di un broker di replica dei dati interregionale Amazon MQ

Per eliminare un broker CRDR (Cross-Region Data Replication) primario o di replica, è necessario prima annullare l'associazione e quindi riavviare i broker. Le seguenti istruzioni mostrano come annullare l'associazione e riavviare i broker utilizzando la Console di gestione. AWS

1. Nella pagina Broker, seleziona il broker CRDR di cui desideri annullare l'associazione, quindi scegli Modifica.
2. Nella pagina Modifica del broker nella sezione Replica dei dati, scegli Annulla associazione broker.
3. Inserisci «conferma» nella finestra pop-up per confermare la tua scelta. Quindi scegli Annulla associazione broker.
4. Quindi, riavvia il broker primario non associato. Questa operazione riavvierà anche il broker di replica. Per istruzioni su come riavviare il broker, consulta. Rebooting a Broker Dopo il riavvio del broker primario, entrambi i broker non saranno associati e possono essere eliminati singolarmente. Per eliminare il broker, consulta Deleting a broker.

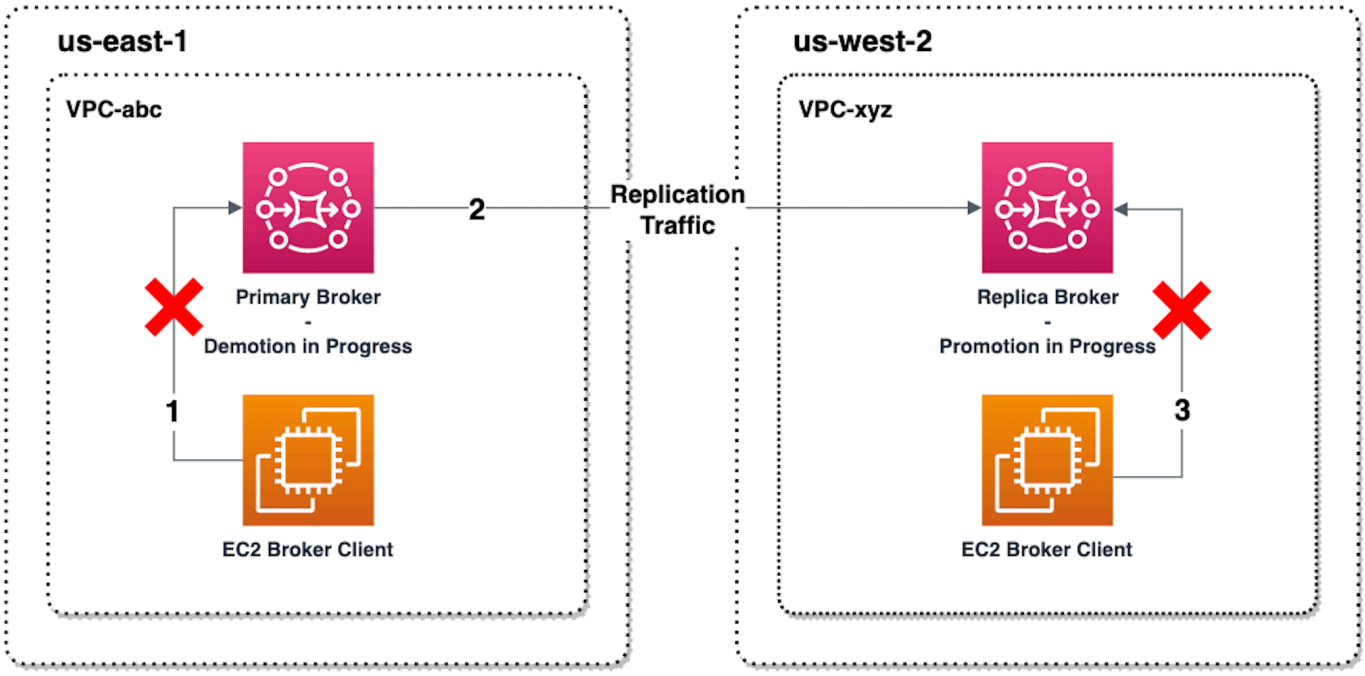
Avvio dello switchover o del failover per promuovere un broker di replica Amazon MQ al ruolo di broker principale

È possibile avviare uno switchover o un failover quando si desidera promuovere il broker di replica al ruolo di broker primario. Quando si promuove il broker di replica, il broker primario viene retrocesso al ruolo di broker di replica.

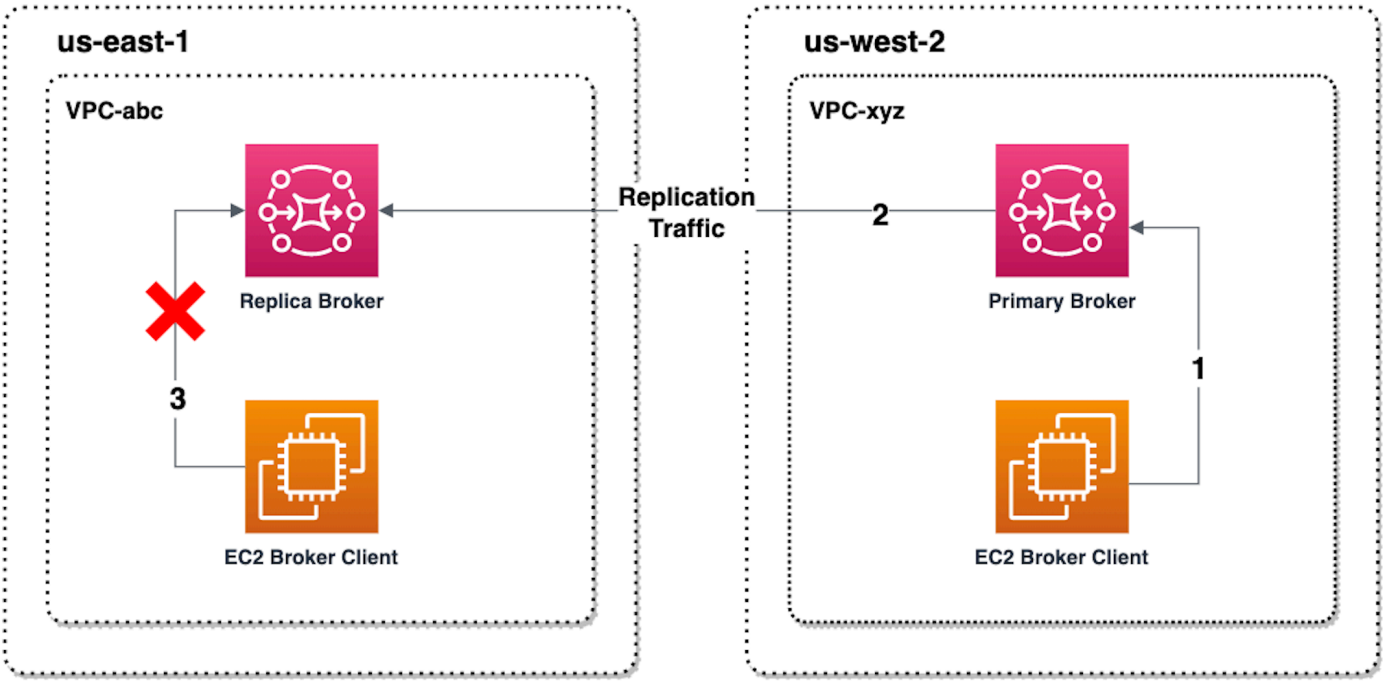
Uno switchover dà priorità alla coerenza rispetto alla disponibilità. È garantito che i broker abbiano lo stesso stato al termine di questa operazione di failover. In caso di transizione, può verificarsi un periodo in cui nessuno dei due broker è disponibile per le connessioni con i client mentre viene stabilita la coerenza tra broker. Entrambi i broker avranno lo stesso stato nel momento in cui la replica viene promossa. Il successo dello switchover dipende dallo stato di entrambe le regioni e della rete interregionale.

Uno failover dà priorità alla disponibilità rispetto alla coerenza. Non è garantito che i broker abbiano stati identici al termine di questa operazione. Con un failover, è garantito che il broker di replica diventi immediatamente disponibile per servire il traffico client, senza attendere la sincronizzazione dei dati di replica o che il principale riceva il segnale di spegnimento. Il successo del failover non dipende né dallo stato della regione primaria originale né dalla rete interregionale.

Il diagramma seguente illustra uno switchover in cui nessuno dei broker accetta connessioni client mentre la coda di replica viene svuotata e gli stati dei broker sono sincronizzati. In questo processo, il client nel VPC del broker principale non è in grado di apportare ulteriori modifiche di stato mentre l'operazione è in corso e il broker principale viene retrocesso a una replica. Quando la coda di replica viene svuotata e i due broker raggiungono lo stesso stato, il client nel VPC del broker di replica non è in grado di connettersi al broker di replica fino al completamento dell'operazione di failover e fino a che il broker di replica non viene promosso a principale.



Il diagramma seguente illustra lo stato del broker dopo il completamento del processo di switchover. Il broker di replica originale è stato ora promosso al ruolo di broker primario e accetta connessioni client. Il client può produrre e utilizzare i dati del broker.



Promozione del broker di replica mediante la console

Per promuovere il broker di replica tramite switchover o failover, segui queste fasi nella console Amazon MQ.

> **Note**
>
> Non è possibile avviare lo switchover o il failover su un broker primario.

1. Passa alla regione del tuo broker di replica. Nella tabella Broker, seleziona il broker di replica esistente che intendi promuovere come primario.
2. Nella Pagina dei dettagli del broker procedi come segue:
    1. Seleziona Promuovi replica.
    2. Nella finestra a comparsa, scegli Switchover o Failover.
    3. Digita "conferma" nella casella di testo per confermare la tua scelta.
    4. Scegli Conferma.

Dopo l'avvio del failover, lo stato del broker cambia in Failover in corso. Quando il failover è completo, la barra di avanzamento blu nella parte superiore della pagina Broker diventa verde.

> **Note**
>
> La configurazione viene replicata solo al momento della creazione della replica del broker. Qualsiasi aggiornamento successivo non viene replicato.

Metriche di replica dei dati tra regioni in Amazon CloudWatch

La funzionalità di replica dei dati tra regioni di Amazon MQ per ActiveMQ offre metriche per mantenere l'affidabilità, la disponibilità e le prestazioni dei broker primari e di replica. Durante il processo di replica, un broker di replica in una regione secondaria riceve dati replicati in modo asincrono dal broker primario nella regione primaria. Se il broker primario nella regione primaria restituisce un errore, è possibile promuovere il broker di replica nella regione secondaria a primario avviando uno switchover o un failover. Per istruzioni sulla visualizzazione delle metriche in Amazon CloudWatch, consultaAccesso ai CloudWatch parametri per Amazon MQ.

Timestamp CRDR

I seguenti timestamp descrivono come vengono calcolate le metriche trovate in Amazon CloudWatch . Esistono cinque timestamp nel processo di replica dei dati:

- Tempo di osservazione corrente (TCO, Time of current observation): l'istante attuale nel tempo.
- Ora di creazione (TC, Time of creation): l'istante in cui un evento è stato creato nella coda di replica dal broker primario. Disponibile sia sui broker primari che su quelli di replica.
- Ora di consegna (TD, Time of delivery): l'istante in cui un evento è stato consegnato con successo al broker di replica. Disponibile solo sui broker di replica.
- Tempo di elaborazione (TP, Time of processing): l'istante in cui un evento è stato elaborato correttamente dal broker di replica. Disponibile solo sui broker di replica.
- Ora del riconoscimento (TA, Time of acknowledgement): l'istante in cui un evento è stato riconosciuto con successo dal broker primario. Disponibile solo sui broker primari.

Stima le prestazioni di switchover/failover con le metriche CRDR CloudWatch

Amazon MQ abilita i parametri per il tuo broker per impostazione predefinita. Puoi visualizzare le metriche del tuo broker accedendo alla CloudWatch console Amazon o utilizzando l' CloudWatch API. Le seguenti metriche sono utili per comprendere le prestazioni di replica e switchover/failover dei broker CRDR:

CloudWatch Metrica Amazon MQ	Motivo dell'utilizzo del CRDR	
`TotalReplicationLag`	Il tempo stimato tra TA e TC dell'ultimo evento non riconosciuto sul broker primario.	
`ReplicationLag`	Il tempo stimato tra TP e TC dell'ultimo evento non riconosciuto sul broker di replica.	

CloudWatch Metrica Amazon MQ	Motivo dell'utilizzo del CRDR	
`PrimaryWaitTime`	Il tempo stimato tra TCO e TC dell'ultimo evento elaborato sul broker primario.	
`ReplicaWaitTime`	Il tempo stimato tra TCO e TP dell'ultimo evento elaborato sul broker di replica.	
`QueueSize`	Il numero totale di eventi non riconosciuti nella coda di replica sul broker primario.	

`TotalReplicationLag` e `ReplicationLag` descrivono la replica ritardata tra il broker primario e quello di replica. Le due metriche possono essere utilizzate anche per stimare il tempo necessario al completamento dell'operazione di switchover o di failover in corso.

`PrimaryWaitTime` e `ReplicaWaitTime` possono essere utilizzati per identificare eventuali problemi in corso con il processo di replica. Se il valore della metrica è in costante aumento, ciò può indicare che il processo di replica è danneggiato o sospeso. La replica lenta può verificarsi a causa di problemi come il partizionamento della rete, l'avvio del broker e il ripristino prolungato.

Tutorial di ActiveMQ

I seguenti tutorial illustrano come creare e connettersi ai broker ActiveMQ. Per utilizzare il codice di esempio ActiveMQ Java, devi installare Java Standard Edition Development Kit e apportare alcune modifiche di configurazione al codice di esempio

Argomenti

- Creazione e configurazione di una rete di broker Amazon MQ
- Connessione di un'applicazione Java alil broker Amazon MQ
- Integrazione dei broker ActiveMQ con LDAP
- Fase 3: (Opzionale) Connect a una AWS Lambda funzione
- Creazione di un utente broker ActiveMQ

- Modifica un utente del broker ActiveMQ
- Eliminare un utente del broker ActiveMQ
- Esempi funzionanti di utilizzo di Java Message Service (JMS) con ActiveMQ

Creazione e configurazione di una rete di broker Amazon MQ

Una rete di broker è composta da più broker a istanza singola simultaneamente attivi o broker attivi/in standby. In questo tutorial imparerai a creare una rete di broker a due broker con una topologia source and sink.

Per una panoramica concettuale e informazioni di configurazione dettagliate, vedi quanto segue:

- Rete di broker Amazon MQ
- Configura la rete di broker nel modo corretto
- `networkConnector`
- `networkConnectionStartAsincrono`
- Reti di broker nella documentazione di ActiveMQ

Puoi utilizzare la console Amazon MQ per creare una rete di broker Amazon MQ. Poiché è possibile avviare la creazione di due broker in parallelo, questo processo richiede circa 15 minuti.

Argomenti

- Prerequisiti
- Fase 1: abilita il traffico tra i broker
- Fase 2: configura i connettori di rete per il broker
- Fasi successive

Prerequisiti

Per creare una rete di broker, devi disporre di quanto segue:

- Due o più broker attivi simultaneamente (denominati `MyBroker1` e `MyBroker2` in questo tutorial). Per ulteriori informazioni sulla creazione di broker, consulta Guida introduttiva: creazione e connessione a un broker ActiveMQ.

- I due broker devono essere nello stesso VPC o in peering. VPCs Per ulteriori informazioni su VPCs, consulta Cos'è Amazon VPC? nella Guida per l'utente di Amazon VPC e Cos'è il peering VPC? nella Amazon VPC Peering Guide.

  > **Important**
  >
  > Se non disponi di un VPC predefinito, di una o più sottoreti o di un gruppo di sicurezza, per prima cosa devi crearli. Per ulteriori informazioni, consultare gli argomenti seguenti nella Guida per l'utente di Amazon VPC:
  >
  > - Creazione di un VPC predefinito
  > - Creazione di una sottorete predefinita
  > - Creazione di un gruppo di sicurezza

- Due utenti con credenziali di accesso identiche per entrambi i broker. Per ulteriori informazioni sulla creazione degli utenti, vedere Creazione di un utente broker ActiveMQ.

  > **Note**
  >
  > Quando si integra l'autenticazione LDAP con una rete di broker, assicurarsi che l'utente esista sia come broker ActiveMQ, sia come utente LDAP.

L'esempio seguente utilizza due broker a istanza singola. Tuttavia, puoi creare reti di broker utilizzando broker attivi/in standby o una combinazione di modalità di distribuzione di broker.

Fase 1: abilita il traffico tra i broker

Dopo aver creato i broker, devi abilitare il traffico tra di loro.

1. Sulla console Amazon MQ, nella pagina MyBroker2, nella sezione Dettagli, in Sicurezza e rete, scegli il nome del tuo gruppo di sicurezza oppure

   Viene visualizzata la pagina Security Groups (Gruppi di sicurezza) del pannello di controllo EC2.
2. Scegli il tuo gruppo di sicurezza dall'elenco.
3. Nella parte inferiore della pagina scegli Inbound (In entrata), quindi scegli Edit (Modifica).
4. Nella finestra di dialogo Modifica regole in entrata, aggiungi una regola per l' OpenWire endpoint.

a. Selezionare Add Rule (Aggiungi regola).

b. Per Type (Tipo) seleziona Custom TCP (TCP personalizzato).

c. Per Port Range, digitate la OpenWire porta ()`61617`.

d. Esegui una delle seguenti operazioni:

- Se desideri limitare l'accesso a un determinato indirizzo IP per Origine, lascia selezionato Personalizzato, quindi immetti l'indirizzo IP di `MyBroker1` seguito da `/32`. (Questo converte l'indirizzo IP in un record CIDR valido). Per ulteriori informazioni consulta Interfacce di rete elastiche.

> **Tip**
>
> Per recuperare l'indirizzo IP di `MyBroker1`, nella console Amazon MQ scegliere il nome del broker e andare alla sezione Details (Dettagli).

- Se tutti i broker sono privati e appartengono allo stesso VPC, per Origine lascia selezionato Personalizzato quindi digita l'ID del gruppo di sicurezza che stai modificando.

> **Note**
>
> Per i broker pubblici, è necessario limitare l'accesso utilizzando gli indirizzi IP.

e. Scegli Save (Salva).

Il broker può ora accettare connessioni in entrata.

Fase 2: configura i connettori di rete per il broker

Dopo aver abilitato il traffico tra i broker, devi configurare i connettori di rete per uno di essi.

1. Modifica la revisione della configurazione per il broker `MyBroker1`.

a. Nella pagina MyBroker1, scegli Modifica.

b. Nella pagina Modifica MyBroker 1, nella sezione Configurazione, scegli Visualizza.

Vengono visualizzati il tipo di motore del broker e la versione utilizzati dalla configurazione (ad esempio, Apache ActiveMQ 5.15.0).

c. Nella scheda Configuration details (Dettagli configurazione) vengono visualizzati il numero di revisione della configurazione, la descrizione e la configurazione del broker in formato XML.

d. Scegli Edit configuration (Modifica configurazione).

e. Nella parte inferiore del file di configurazione, rimuovi il commento dalla sezione `<networkConnectors>` e includi le informazioni riportate di seguito:

- Il `name` per il connettore di rete.
- Lo `username` della console Web ActiveMQ comune a entrambi i broker.
- Abilita le connessioni `duplex`.
- Esegui una delle seguenti operazioni:
  - Se stai collegando il broker a un broker a istanza singola, usa il `static:` prefisso e l' OpenWire endpoint `uri` per. `MyBroker2` Esempio:

```
<networkConnectors>
  <networkConnector name="connector_1_to_2" userName="myCommonUser"
 duplex="true"
    uri="static:(ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
east-2.amazonaws.com:61617)"/>
</networkConnectors>
```

  - Se stai collegando il broker a un broker attivo/in standby, utilizza il `static+failover` trasporto e l' OpenWireendpoint `uri` per entrambi i broker con i seguenti parametri di query. `?randomize=false&maxReconnectAttempts=0` Esempio:

```
<networkConnectors>
  <networkConnector name="connector_1_to_2" userName="myCommonUser"
 duplex="true"
    uri="static:(failover:(ssl://
b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:61617,
    ssl://b-9876l5k4-32ji-109h-8gfe-7d65c4b132a1-2.mq.us-
west-2.amazonaws.com:61617)?randomize=false&amp;maxReconnectAttempts=0)"/>
</networkConnectors>
```

> **Note**
>
> Non includere le credenziali di accesso per l'utente ActiveMQ.

f. Scegli Save (Salva).

g. Nella finestra di dialogo Save revision (Salva revisione), digita `Add network of brokers connector for MyBroker2`.

h. Scegli Save (Salva) per salvare la nuova revisione della configurazione.

2. Modifica `MyBroker1` per applicare immediatamente l'ultima revisione della configurazione.

a. Nella pagina MyBroker1, scegli Modifica.

b. Nella pagina Modifica MyBroker 1, nella sezione Configurazione, scegli Pianifica modifiche.

c. Nella sezione Schedule broker modifications (Pianifica modifiche broker), scegli di applicare le modifiche Immediately (Immediatamente).

d. Scegli Applica.

`MyBroker1` viene riavviato e la revisione della configurazione viene applicata.

La rete di broker viene creata.

Fasi successive

Dopo aver configurato la rete di broker, è possibile testarla tramite la creazione e l'utilizzo di messaggi.

> **Important**
>
> Assicurati di abilitare le connessioni in entrata dal tuo computer locale per il broker `MyBroker1` sulla porta 8162 (per ActiveMQ Web Console) e sulla porta 61617 (per l'endpoint). OpenWire
> Potresti inoltre dover modificare le impostazioni dei gruppi di sicurezza per consentire al produttore e al consumatore di connettersi alla rete di broker.

1. Nella console Amazon MQ, andare alla sezione Connections (Connessioni) e prendere nota dell'endpoint della console Web ActiveMQ per il broker `MyBroker1`.
2. Vai alla console Web ActiveMQ per il broker `MyBroker1`.
3. Per verificare che il bridge di rete sia connesso, scegli Network (Rete).

Nella sezione Network Bridges (Bridge di rete), il nome e l'indirizzo di `MyBroker2` sono elencati nelle colonne Remote Broker (broker remoto) e Remote Address (Indirizzo remoto).

4. Da una macchina che ha accesso al broker `MyBroker2`, crea un consumatore. Esempio:

```
activemq consumer --brokerUrl "ssl://
b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:61617" \
 --user commonUser \
 --password myPassword456 \
 --destination queue://MyQueue
```

   Il consumatore si connette all' OpenWire endpoint di e inizia a consumare i messaggi dalla `MyBroker2` coda. `MyQueue`

5. Da una macchina che ha accesso al broker `MyBroker1`, crea un produttore e invia alcuni messaggi. Esempio:

```
activemq producer --brokerUrl "ssl://
b-9876l5k4-32ji-109h-8gfe-7d65c4b132a1-1.mq.us-east-2.amazonaws.com:61617" \
 --user commonUser \
 --password myPassword456 \
 --destination queue://MyQueue \
 --persistent true \
 --messageSize 1000 \
 --messageCount 10000
```

   Il produttore si connette all' OpenWire endpoint di `MyBroker1` e inizia a produrre messaggi persistenti da mettere in coda. `MyQueue`

Connessione di un'applicazione Java alil broker Amazon MQ

Dopo aver creato un broker ActiveMQ di Amazon MQ, è possibile collegarvi l'applicazione. Di seguito sono riportati esempi che mostrano come è possibile utilizzare il servizio di messaggistica Java (JMS) per creare una connessione alil broker, creare una coda e inviare un messaggio. Per un esempio Java completo e funzionante, consultare Working Java Example.

Puoi connetterti ai broker ActiveMQ utilizzando vari client ActiveMQ. È consigliato l'uso del client ActiveMQ.

Argomenti

- Prerequisiti
- Per creare un produttore del messaggio e inviare un messaggio

- Per creare un consumatore del messaggio e ricevere il messaggio

Prerequisiti

Abilitazione attributi VPC

Per garantire che il broker sia accessibile all'interno del VPC, è necessario abilitare gli attributi VPC `enableDnsHostnames` e `enableDnsSupport`. Per ulteriori informazioni, consultare Supporto del DNS nel VPC nella Guida per l'utente di Amazon VPC.

Abilitazione connessioni in entrata

Quindi, abilita le connessioni in entrata per la tua applicazione.

1. Accedere alla console Amazon MQ.
2. Dall'elenco dei broker, scegli il nome del tuo broker (ad esempio, MyBroker).
3. Nella ***MyBroker***pagina, nella sezione Connessioni, annota gli indirizzi e le porte dell'URL della console web del broker e dei protocolli a livello di cavo.
4. Nella sezione Details (Dettagli), in Security and network (Sicurezza e rete), scegliere il nome del gruppo di sicurezza o

   .

   Viene visualizzata la pagina Security Groups (Gruppi di sicurezza) del pannello di controllo EC2.
5. Scegli il tuo gruppo di sicurezza dall'elenco.
6. Nella parte inferiore della pagina scegli Inbound (In entrata), quindi scegli Edit (Modifica).
7. Nella finestra di dialogo Edit inbound rules (Modifica le regole in entrata), aggiungere una regola per ogni URL o endpoint che si desidera rendere accessibile pubblicamente (nell'esempio seguente viene illustrato come eseguire questa operazione per una console Web del broker).

   a. Selezionare Add Rule (Aggiungi regola).
   b. Per Type (Tipo) seleziona Custom TCP (TCP personalizzato).
   c. Per Port Range (Intervallo porte), digitare la porta della console Web (`8162`).
   d. Per Source (Origine), lasciare selezionato Custom (Personalizzato), quindi inserire l'indirizzo IP del sistema a cui desideri poter accedere alla console Web (ad esempio, `192.0.2.1`).
   e. Scegli Save (Salva).

   Il broker può ora accettare connessioni in entrata.

Aggiunta dipendenze Java

Aggiungere i pacchetti `activemq-client.jar` e `activemq-pool.jar` al percorso di classe Java. L'esempio seguente mostra queste dipendenze in un file `pom.xml` di progetto Maven.

```
<dependencies>
    <dependency>
        <groupId>org.apache.activemq</groupId>
        <artifactId>activemq-client</artifactId>
        <version>5.15.16</version>
    </dependency>
    <dependency>
        <groupId>org.apache.activemq</groupId>
        <artifactId>activemq-pool</artifactId>
        <version>5.15.16</version>
    </dependency>
</dependencies>
```

Per ulteriori informazioni su `activemq-client.jar`, consultare Configurazione iniziale nella documentazione di Apache ActiveMQ.

> **Important**
>
> Nel codice di esempio seguente, produttori e consumatori vengono eseguiti in un singolo thread. Per i sistemi di produzione (o per testare il failover delle istanze del broker), assicurarsi che i produttori e i consumatori vengano eseguiti su host o thread separati.

Per creare un produttore del messaggio e inviare un messaggio

Usa le seguenti istruzioni per creare un produttore di messaggi e ricevere un messaggio.

1. Creare un pool di connessioni di stabilimento JMS per il produttore di messaggi utilizzando l'endpoint del broker e quindi chiamare il metodo `createConnection` rispetto allo stabilimento.

   > **Note**
   >
   > Per un active/standby broker, Amazon MQ fornisce due URLs console Web ActiveMQ, ma è attivo un solo URL alla volta. Allo stesso modo, Amazon MQ fornisce due endpoint per ogni protocollo a livello di connessione, ma è attivo un solo endpoint per ogni coppia

alla volta. I suffissi `-1` e `-2` indicano una coppia ridondante. Per ulteriori informazioni, vedere Opzioni di implementazione per i broker Amazon MQ for ActiveMQ).
Per gli endpoint con protocollo a livello di cavo, è necessario consentire all'applicazione di connettersi a entrambi gli endpoint utilizzando il Failover Transport.

```
// Create a connection factory.
final ActiveMQConnectionFactory connectionFactory = new
 ActiveMQConnectionFactory(wireLevelEndpoint);

// Pass the sign-in credentials.
connectionFactory.setUserName(activeMqUsername);
connectionFactory.setPassword(activeMqPassword);

// Create a pooled connection factory.
final PooledConnectionFactory pooledConnectionFactory = new
 PooledConnectionFactory();
pooledConnectionFactory.setConnectionFactory(connectionFactory);
pooledConnectionFactory.setMaxConnections(10);

// Establish a connection for the producer.
final Connection producerConnection = pooledConnectionFactory.createConnection();
producerConnection.start();

// Close all connections in the pool.
pooledConnectionFactory.clear();
```

> **Note**
>
> I produttori di messaggi devono sempre utilizzare la classe `PooledConnectionFactory`. Per ulteriori informazioni, consulta Usa sempre il pooling delle connessioni.

2. Creare una sessione, una coda denominata `MyQueue` e un produttore di messaggi.

```
// Create a session.
final Session producerSession = producerConnection.createSession(false,
 Session.AUTO_ACKNOWLEDGE);

// Create a queue named "MyQueue".
```

```
final Destination producerDestination = producerSession.createQueue("MyQueue");

// Create a producer from the session to the queue.
final MessageProducer producer =
 producerSession.createProducer(producerDestination);
producer.setDeliveryMode(DeliveryMode.NON_PERSISTENT);
```

3. Creare la stringa del messaggio `"Hello from Amazon MQ!"` e quindi inviare il messaggio.

```
// Create a message.
final String text = "Hello from Amazon MQ!";
TextMessage producerMessage = producerSession.createTextMessage(text);

// Send the message.
producer.send(producerMessage);
System.out.println("Message sent.");
```

4. Eliminare il produttore.

```
producer.close();
producerSession.close();
producerConnection.close();
```

Per creare un consumatore del messaggio e ricevere il messaggio

Utilizzate le seguenti istruzioni per creare un produttore di messaggi e ricevere un messaggio.

1. Creare una connessione di stabilimento JMS per il produttore di messaggi utilizzando l'endpoint del broker e quindi chiamare il metodo `createConnection` rispetto allo stabilimento.

```
// Create a connection factory.
final ActiveMQConnectionFactory connectionFactory = new
 ActiveMQConnectionFactory(wireLevelEndpoint);

// Pass the sign-in credentials.
connectionFactory.setUserName(activeMqUsername);
connectionFactory.setPassword(activeMqPassword);

// Establish a connection for the consumer.
final Connection consumerConnection = connectionFactory.createConnection();
consumerConnection.start();
```

> **Note**
>
> I consumatori dei messaggi non dovrebbero mai utilizzare la classe `PooledConnectionFactory`. Per ulteriori informazioni, consulta Usa sempre il pooling delle connessioni.

2. Creare una sessione, una coda denominata `MyQueue` e un consumatore di messaggi.

```
// Create a session.
final Session consumerSession = consumerConnection.createSession(false,
 Session.AUTO_ACKNOWLEDGE);

// Create a queue named "MyQueue".
final Destination consumerDestination = consumerSession.createQueue("MyQueue");

// Create a message consumer from the session to the queue.
final MessageConsumer consumer =
 consumerSession.createConsumer(consumerDestination);
```

3. Iniziare ad attendere i messaggi quindi riceverlo non appena arriva.

```
// Begin to wait for messages.
final Message consumerMessage = consumer.receive(1000);

// Receive the message when it arrives.
final TextMessage consumerTextMessage = (TextMessage) consumerMessage;
System.out.println("Message received: " + consumerTextMessage.getText());
```

> **Note**
>
> A differenza dei servizi di AWS messaggistica (come Amazon SQS), il consumatore è costantemente connesso al broker.

4. Chiudere il consumatore, la sessione e la connessione.

```
consumer.close();
consumerSession.close();
consumerConnection.close();
```

Integrazione dei broker ActiveMQ con LDAP

> **Important**
>
> Amazon MQ non supporta certificati server emessi da una CA privata.

È possibile accedere ai broker ActiveMQ utilizzando i seguenti protocolli con TLS abilitato:

- AMQP
- MQTT
- MQTT over WebSocket
- OpenWire
- STOMP
- STOMP over WebSocket

Amazon MQ offre una scelta tra l'autenticazione nativa ActiveMQ e l'autenticazione LDAP e l'autorizzazione per gestire le autorizzazioni utente. Per informazioni relative alle limitazioni correlate a nomi utente e password ActiveMQ, consulta Utenti.

Per autorizzare gli utenti e i gruppi ActiveMQ a utilizzare code e argomenti, è necessario modificare la configurazione del broker. Amazon MQ utilizza il plugin di autenticazione semplice di ActiveMQ per limitare la lettura e la scrittura alle destinazioni. Per ulteriori informazioni ed esempi, consulta Configurare sempre una mappa di autorizzazione e `authorizationEntry`.

> **Note**
>
> Attualmente, Amazon MQ non supporta l'autenticazione dei certificati client.

Argomenti

- Integrazione di LDAP con ActiveMQ
- Prerequisiti
- Nozioni di base su LDAP
- Come funziona l'integrazione LDAP

Integrazione di LDAP con ActiveMQ

È possibile autenticare gli utenti Amazon MQ tramite le credenziali memorizzate nel server LDAP (Lightweight Directory Access Protocol). È inoltre possibile aggiungere, eliminare e modificare gli utenti Amazon MQ e assegnare autorizzazioni ad argomenti e code attraverso di esso. Le operazioni di gestione come la creazione, l'aggiornamento e l'eliminazione dei broker richiedono ancora credenziali IAM e non sono integrate con LDAP.

I clienti che desiderano semplificare e centralizzare l'autenticazione e l'autorizzazione del broker Amazon MQ utilizzando un server LDAP possono utilizzare questa funzione. Mantenere tutte le credenziali utente nel server LDAP consente di risparmiare tempo e fatica fornendo una posizione centrale per l'archiviazione e la gestione di tali credenziali.

Amazon MQ fornisce supporto LDAP utilizzando il plugin Apache ActiveMQ JAAS. Qualsiasi server LDAP, ad esempio Microsoft Active Directory od OpenLDAP supportato dal plugin, è supportato anche da Amazon MQ. Per ulteriori informazioni sul plugin, consultare la sezione Sicurezza della documentazione di Active MQ.

Oltre agli utenti, è possibile specificare l'accesso agli argomenti e alle code per un gruppo specifico o un utente tramite il server LDAP. A tale scopo, creare voci che rappresentano argomenti e code nel server LDAP e quindi assegnare autorizzazioni a un utente LDAP specifico o a un gruppo. È quindi possibile configurare i broker per recuperare i dati di autorizzazione dal server LDAP.

> **Important**
>
> Quando si utilizza LDAP, l'autenticazione non fa distinzione tra maiuscole e minuscole, ma l'autorizzazione fa distinzione tra maiuscole e minuscole per il nome utente.

Prerequisiti

Prima di aggiungere il supporto LDAP a un broker Amazon MQ nuovo o esistente, occorre configurare un account di servizio. Questo account di servizio è necessario per avviare una connessione a un server LDAP e deve disporre delle autorizzazioni corrette per effettuare questa connessione. Questo account di servizio configurerà l'autenticazione LDAP per il broker. Eventuali connessioni client successive verranno autenticate tramite la stessa connessione.

Un account del servizio è un account nel server LDAP che ha accesso per avviare una connessione. Si tratta di un requisito LDAP standard ed è necessario fornire le credenziali dell'account di servizio

una sola volta. Dopo aver configurato la connessione, tutte le future connessioni client vengono autenticate tramite il server LDAP. Le credenziali dell'account di servizio sono memorizzate in modo sicuro in un formato crittografato, accessibile solo ad Amazon MQ.

Per l'integrazione con ActiveMQ, sul server LDAP è necessario disporre di una specifica struttura DIT (Directory Information Tree). Per un esempio di file `ldif` che mostra chiaramente questa struttura, consultare Importazione del seguente file LDIF nel server LDAP nella sezione Sicurezza della documentazione di ActiveMQ.

Nozioni di base su LDAP

Per iniziare, accedere alla console Amazon MQ e scegliere LDAP authentication and authorization (Autenticazione e autorizzazione LDAP) durante la creazione di una nuova istanza del broker Amazon MQ o la modifica di un'istanza esistente.

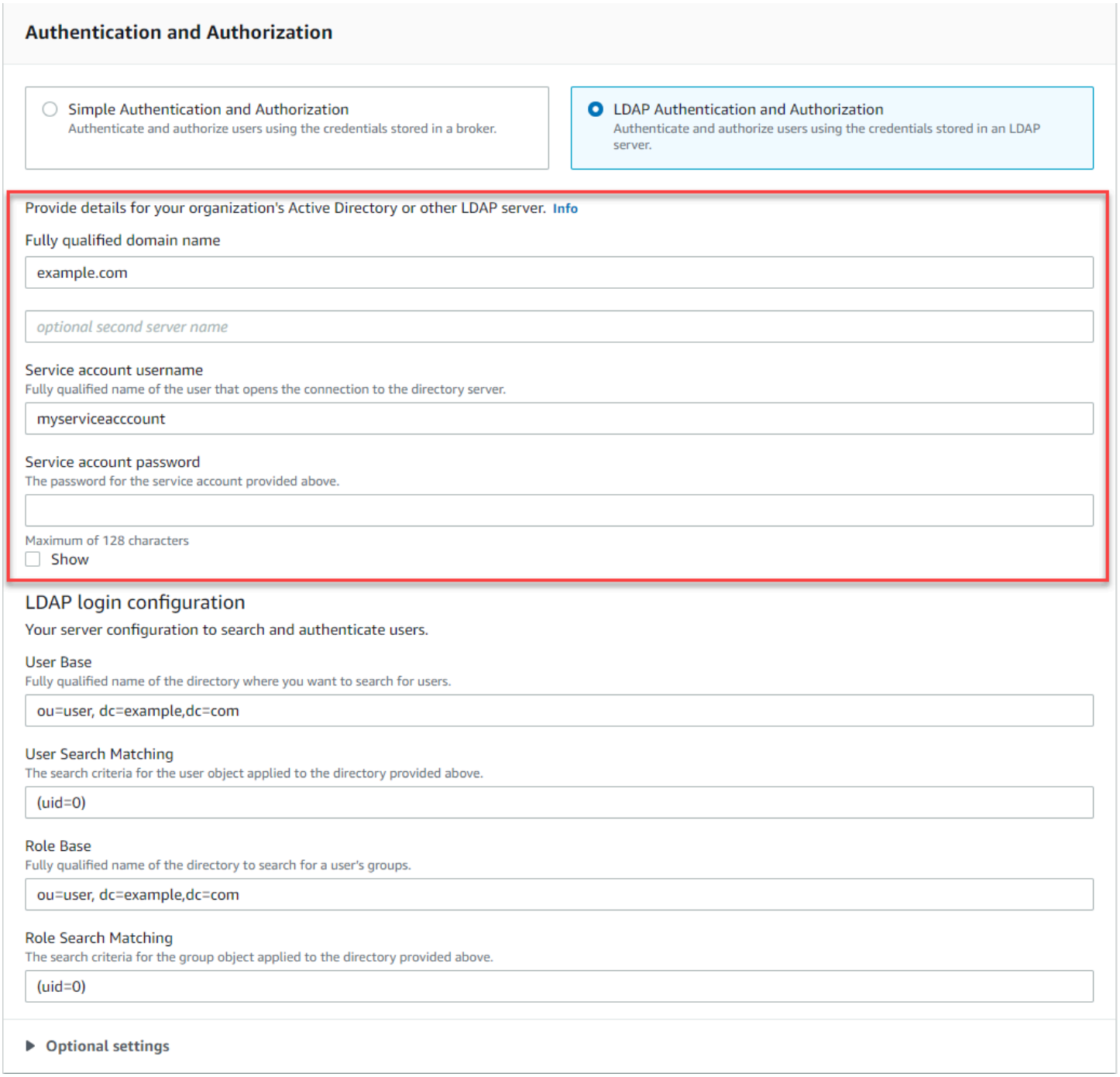
Fornire le informazioni seguenti sull'account di servizio:

- Fully qualified domain name (Nome di dominio completo) Posizione del server LDAP in cui devono essere emesse le richieste di autenticazione e autorizzazione.

  > **Note**
  >
  > Il nome di dominio completo del server LDAP fornito non deve includere il protocollo o il numero di porta. Amazon MQ antepone il nome di dominio completo con il protocollo `ldaps`, a cui aggiungerà il numero di porta `636`.
  > Ad esempio, se si specifica il seguente dominio completo `example.com`, Amazon MQ accederà al server LDAP utilizzando l'URL `ldaps://example.com:636`.
  > Affinché l'host del broker sia in grado di comunicare correttamente con il server LDAP, il nome di dominio completo deve essere risolvibile pubblicamente. Per mantenere il server LDAP privato e sicuro, limitare il traffico in ingresso nelle regole in entrata del server per consentire solo il traffico originato dall'interno del VPC del broker.

- Service account username (Nome utente dell'account del servizio) Il nome distinto dell'utente che verrà utilizzato per eseguire l'associazione iniziale al server LDAP.
- Service account password (Password dell'account del servizio) La password dell'utente che esegue l'associazione iniziale.

L'immagine seguente evidenzia dove fornire questi dettagli.



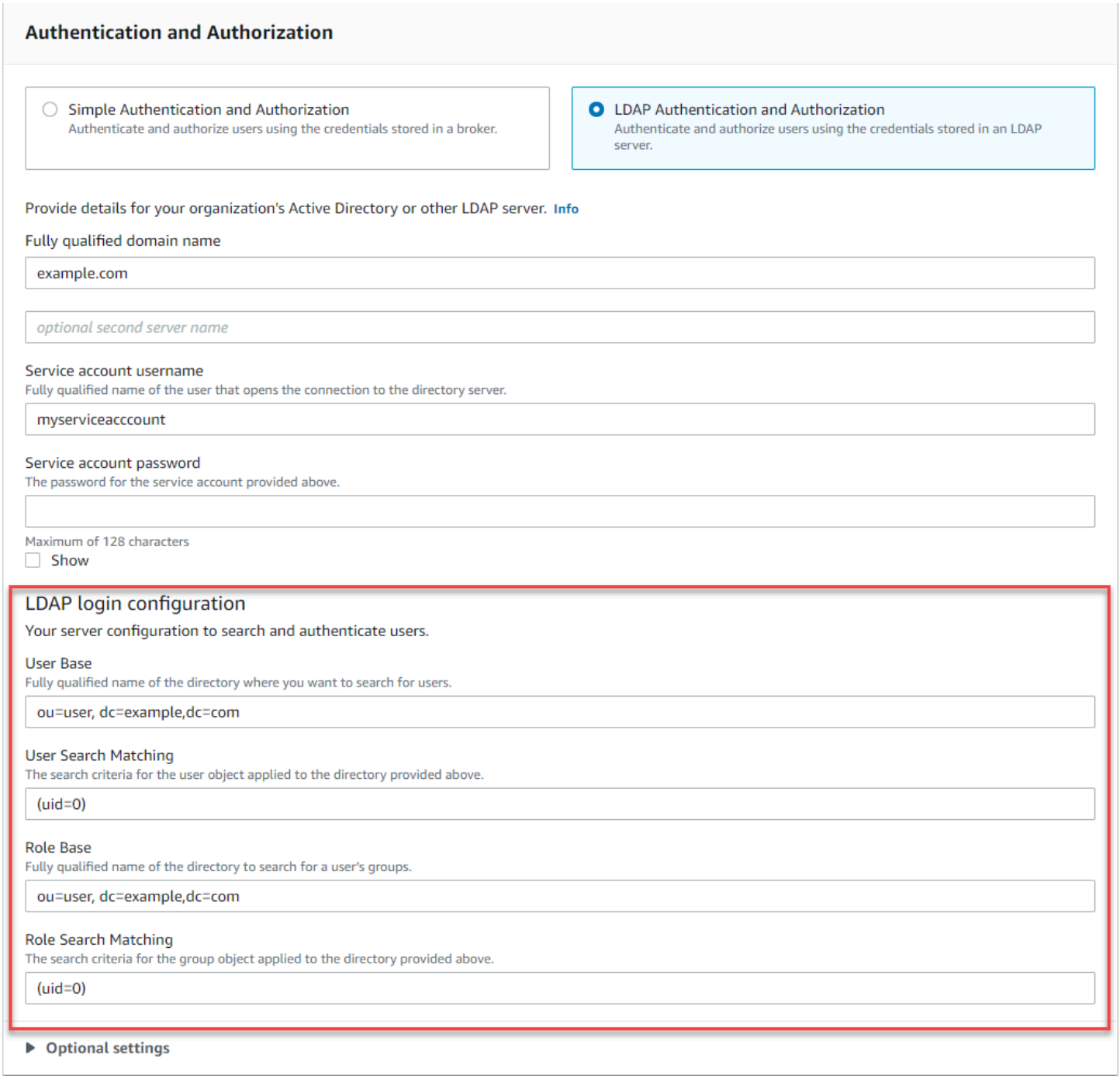
Nella sezione LDAP login configuration (Configurazione dell'accesso con LDAP), fornire le informazioni obbligatorie seguenti:

- User Base (Base utenti) Nome distinto del nodo nella struttura DIT (Directory Information Tree) che verrà cercato per gli utenti.
- User Search Matching (Corrispondenza ricerca utente) Filtro di ricerca LDAP che verrà utilizzato per trovare gli utenti all'interno di `userBase`. Il nome utente del client viene sostituito nel

placeholder `{0}` nel filtro di ricerca. Per ulteriori informazioni, consultare Autenticazione e Autorizzazione.

- Role Base (Base di ruoli) Nome distinto del nodo nel DIT in cui verranno cercati i ruoli. I ruoli possono essere configurati come voci esplicite del gruppo LDAP nella directory. Una voce di ruolo tipica può essere costituita da un attributo per il nome del ruolo, ad esempio nome comune (NC) e un altro attributo, come `member`, con valori che rappresentano i nomi distinti o i nomi utente degli utenti appartenenti al gruppo di ruoli. Ad esempio, data l'unità organizzativa, `group`, è possibile fornire il nome distinto seguente `ou=group,dc=example,dc=com`.
- Role Search Matching (Corrispondenza ricerca ruolo) Filtro di ricerca LDAP che verrà utilizzato per trovare i ruoli all'interno di `roleBase`. Il nome distinto dell'utente abbinato da `userSearchMatching` sarà sostituito nel placeholder `{0}` nel filtro di ricerca. Il nome utente del cliente verrà sostituito al posto del placeholder `{1}`. Ad esempio, se le voci di ruolo nella directory includono un attributo denominato `member`, contenente i nomi utente per tutti gli utenti in tale ruolo, è possibile fornire il seguente filtro di ricerca: `(member:=uid={1})`.

L'immagine seguente evidenzia dove specificare questi dettagli.



Nella sezione Optional settings (Impostazioni opzionali), è possibile fornire le informazioni facoltative seguenti:

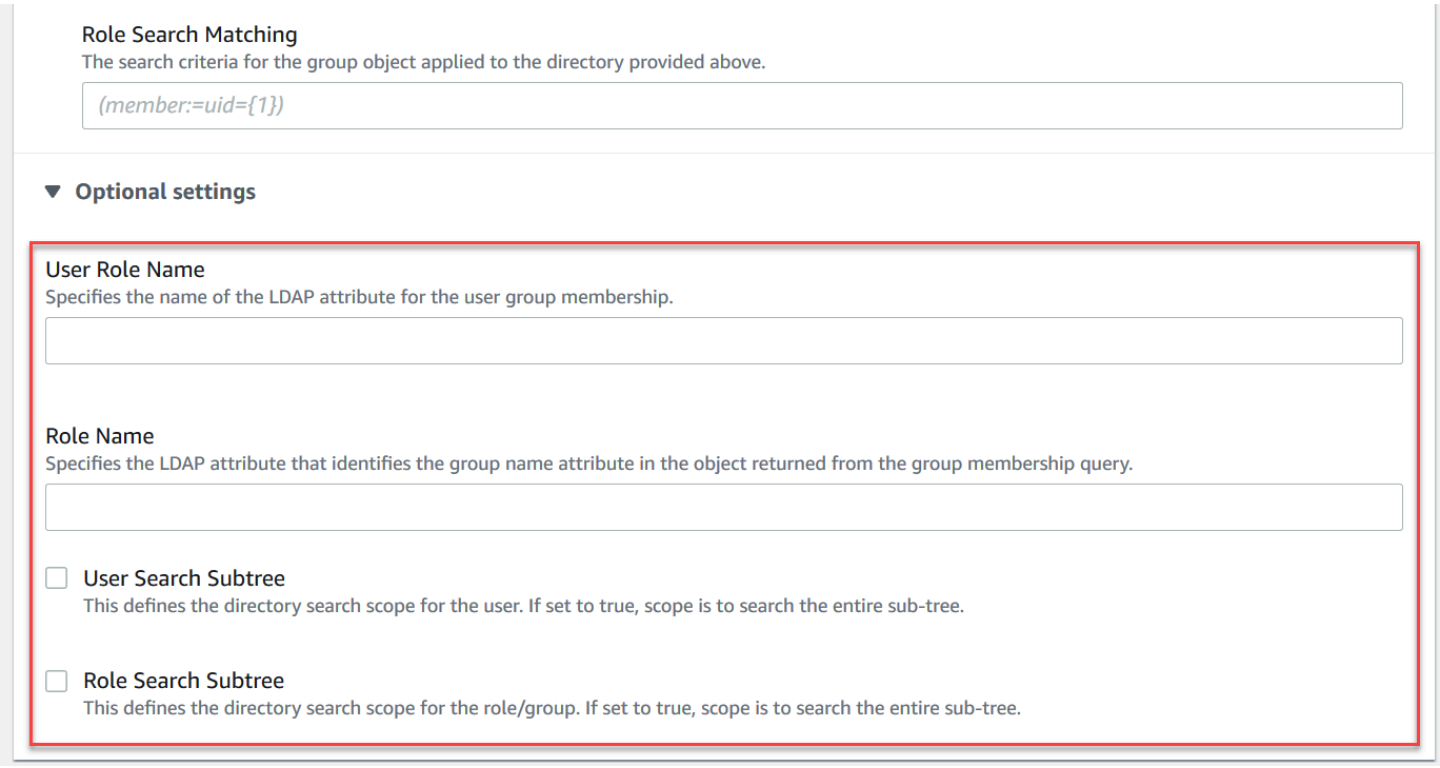
- User Role Name (Nome del ruolo dell'utente) Nome dell'attributo LDAP nella voce di directory dell'utente per l'appartenenza al gruppo dell'utente. In alcuni casi, i ruoli utente possono essere identificati dal valore di un attributo nella voce di directory dell'utente. L'opzione `userRoleName` consente di fornire il nome di questo attributo. Ad esempio, consideriamo la seguente voce utente:

```
dn: uid=jdoe,ou=user,dc=example,dc=com
objectClass: user
uid: jdoe
sn: jane
cn: Jane Doe
mail: j.doe@somecompany.com
memberOf: role1
userPassword: password
```

Per fornire il corretto `userRoleName` per l'esempio precedente, è necessario specificare l'attributo `memberOf`. Se l'autenticazione viene completata correttamente, l'utente viene assegnato al ruolo `role1`.

- Role Name (Nome del ruolo) Attributo del nome del gruppo in una voce del ruolo il cui valore è il nome di tale ruolo. Ad esempio, è possibile specificare `cn` per il nome comune di una voce del gruppo. Se l'autenticazione ha esito positivo, all'utente viene assegnato il valore dell'attributo `cn` per ogni voce di ruolo di cui è membro.
- User Search Subtree (Sottostruttura di ricerca utente) Definisce l'ambito per la query di ricerca utente LDAP. Se true, l'ambito è impostato per cercare l'intera sottostruttura sotto il nodo definito da `userBase`.
- Role Search Subtree (Sottostruttura di ricerca ruolo) Definisce l'ambito per la query di ricerca ruolo LDAP. Se true, l'ambito è impostato per cercare l'intera sottostruttura sotto il nodo definito da `roleBase`.

L'immagine seguente evidenzia dove specificare queste impostazioni opzionali.



Come funziona l'integrazione LDAP

Si può pensare all'integrazione in due categorie principali: la struttura per l'autenticazione e la struttura per l'autorizzazione.

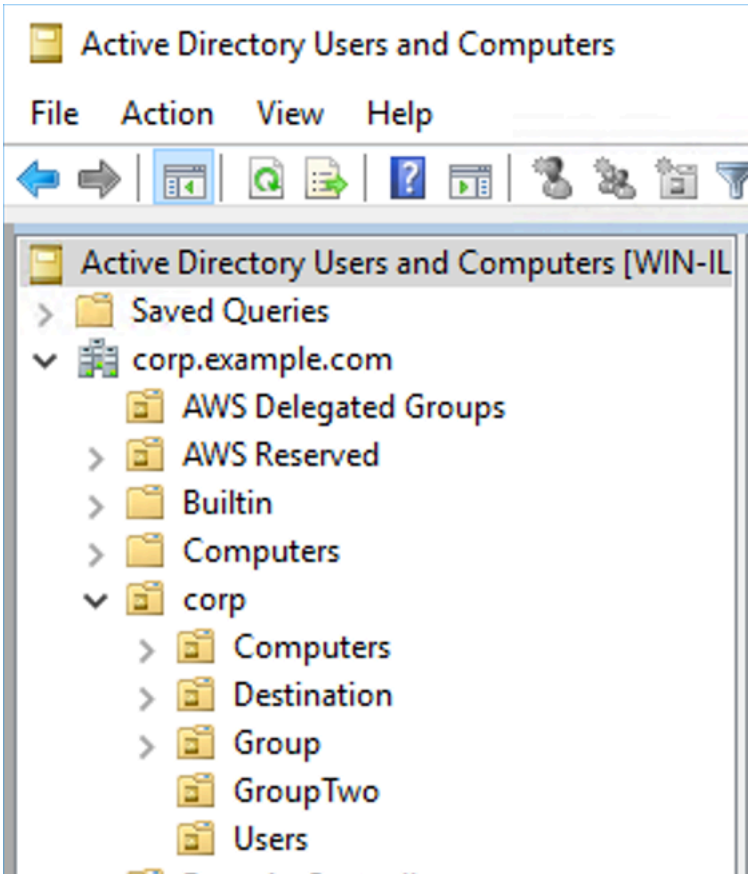
Autenticazione

Per l'autenticazione, le credenziali client devono essere valide. Queste credenziali vengono convalidate rispetto agli utenti della base utenti nel server LDAP.

La base utenti fornita alil broker ActiveMQ deve puntare al nodo nel DIT in cui gli utenti sono archiviati nel server LDAP. Ad esempio, se utilizzi e disponi dei componenti del dominio e AWS Managed Microsoft AD, all'interno di questi `corpexample`, disponi di unità organizzative `corp` e`Users`, utilizzerai quanto segue come base di utenti: `com`

```
OU=Users,OU=corp,DC=corp,DC=example,DC=com
```

il broker ActiveMQ cercherà gli utenti in questa posizione nel DIT al fine di autenticare le richieste di connessione client alil broker.



Poiché il codice sorgente ActiveMQ codifica hardcode il nome dell'attributo per gli utenti in `uid`, è necessario assicurarsi che ogni utente abbia questo attributo impostato. Per semplicità, è possibile utilizzare il nome utente della connessione. Per ulteriori informazioni, consultare il codice sorgente activemq e Configurazione delle mappature dell'ID in utenti Active Directory e computer per Windows Server 2016 (e versioni successive).

Per abilitare l'accesso alla console ActiveMQ per utenti specifici, assicurarsi che appartengano al gruppo `amazonmq-console-admins`.

Autorizzazione

Per l'autorizzazione, le basi di ricerca delle autorizzazioni sono specificate nella configurazione del broker. L'autorizzazione viene eseguita in base alla destinazione (o carattere jolly, set di destinazione) tramite l'elemento `cachedLdapAuthorizationMap`, che si trova nel file di configurazione `activemq.xml`. Per ulteriori informazioni, consultare Modulo di autorizzazione LDAP memorizzato nella cache.

> **Note**
>
> Per poter utilizzare l'`cachedLDAPAuthorizationMap`elemento nel file di `activemq.xml` configurazione del tuo broker, devi scegliere l'opzione Autenticazione e autorizzazione LDAP quando crei una configurazione tramite Console di gestione AWS, oppure impostare la creazione di una configurazione tramite Console di gestione AWS, o impostare la `authenticationStrategy`proprietà su LDAP quando crei una nuova configurazione utilizzando l'API Amazon MQ.

Occorre fornire i seguenti tre attributi come parte dell'elemento `cachedLDAPAuthorizationMap`:

- `queueSearchBase`
- `topicSearchBase`
- `tempSearchBase`

> **Important**
>
> Per evitare che le informazioni sensibili vengano inserite direttamente nel file di configurazione del broker, Amazon MQ blocca l'utilizzo dei seguenti attributi in `cachedLdapAuthorizationMap`:
>
> - `connectionURL`
> - `connectionUsername`
> - `connectionPassword`
>
> Quando crei un broker, Amazon MQ sostituisce i valori forniti tramite o nella `ldapServerMetadata`proprietà della tua richiesta API con gli attributi di cui sopra. Console di gestione AWS

Di seguito è mostrato un esempio funzionante di `cachedLdapAuthorizationMap`.

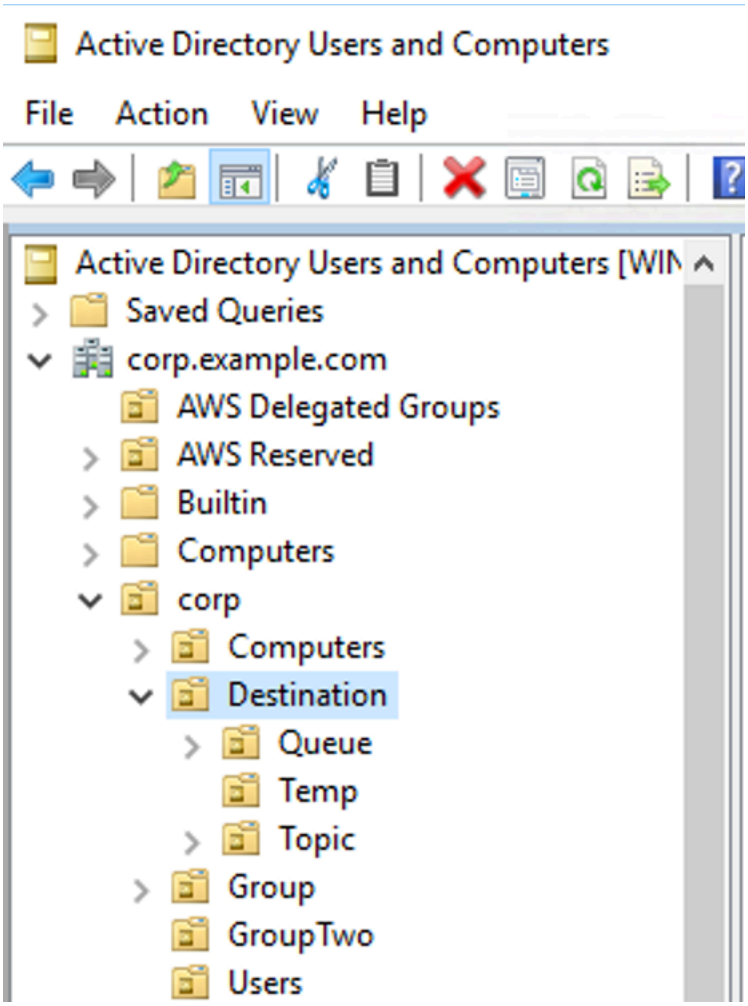
```
<authorizationPlugin>
    <map>
        <cachedLDAPAuthorizationMap
```

```
            queueSearchBase="ou=Queue,ou=Destination,ou=corp,dc=corp,dc=example,dc=com"
            topicSearchBase="ou=Topic,ou=Destination,ou=corp,dc=corp,dc=example,dc=com"
            tempSearchBase="ou=Temp,ou=Destination,ou=corp,dc=corp,dc=example,dc=com"
            refreshInterval="300000"
            legacyGroupMapping="false"
        />
    </map>
</authorizationPlugin>
```

Questi valori identificano le posizioni all'interno del DIT in cui sono specificate le autorizzazioni per ogni tipo di destinazione. Quindi, per l'esempio precedente con AWS Managed Microsoft AD, utilizzando gli stessi componenti di dominio di`corp`, e `examplecom`, dovresti specificare un'unità organizzativa denominata `destination` per contenere tutti i tipi di destinazione. All'interno di quell'unità organizzativa, occorre crearne uno per le destinazioni `queues`, uno per `topics` e uno per `temp`.

Ciò significa che la base di ricerca della coda, che fornisce informazioni di autorizzazione per le destinazioni della coda dei tipi, avrebbe la seguente posizione nel DIT:

```
OU=Queue,OU=Destination,OU=corp,DC=corp,DC=example,DC=com
```



Analogamente, le regole di autorizzazione per gli argomenti e le destinazioni temporanee si trovano allo stesso livello nel DIT:

```
OU=Topic,OU=Destination,OU=corp,DC=corp,DC=example,DC=com
OU=Temp,OU=Destination,OU=corp,DC=corp,DC=example,DC=com
```

All'interno dell'unità organizzativa per ogni tipo di destinazione (coda, argomento, temp), è possibile specificare un carattere jolly o un nome di destinazione specifico. Ad esempio, per fornire una regola di autorizzazione per tutte le code che iniziano con il prefisso DEMO.EVENTS.$., è possibile creare la seguente unità organizzativa:

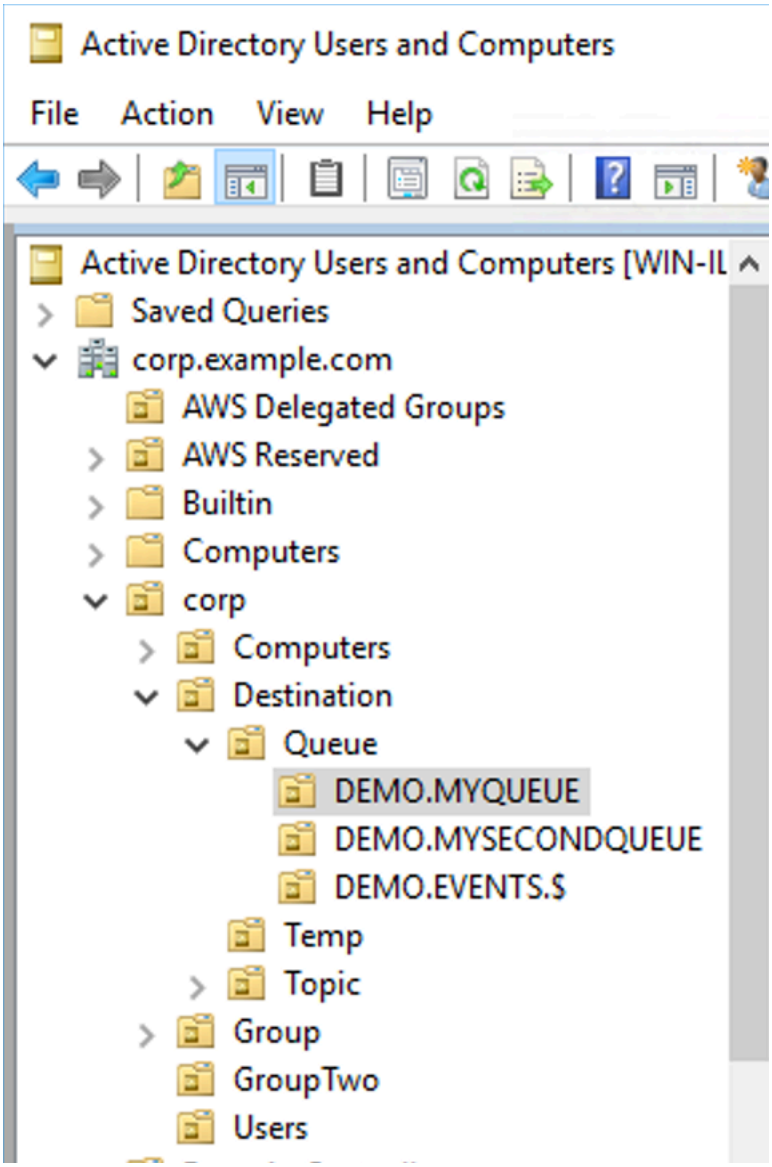
```
OU=DEMO.EVENTS.$,OU=Queue,OU=Destination,OU=corp,DC=corp,DC=example,DC=com
```

> **Note**
>
> L'unità organizzativa `DEMO.EVENTS.$` è all'interno dell'unità organizzativa `Queue`.

Per altre informazioni sui caratteri jolly in ActiveMQ, fare riferimento ai Caratteri jolly

Per fornire regole di autorizzazione per code specifiche, ad esempio DEMO.MYQUEUE, specificare qualcosa di simile al seguente:

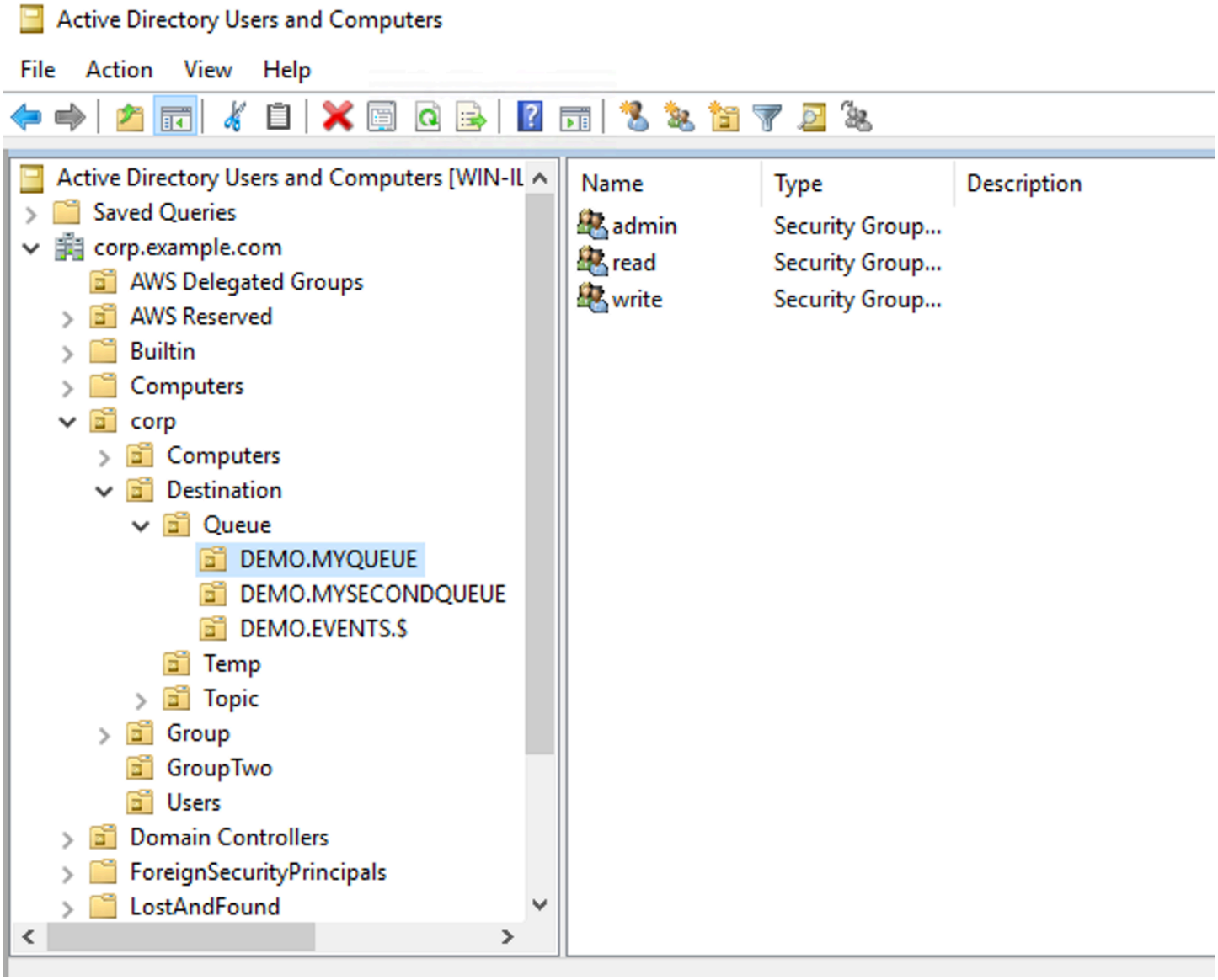
```
OU=DEMO.MYQUEUE,OU=Queue,OU=Destination,OU=corp,DC=corp,DC=example,DC=com
```



Gruppi di sicurezza

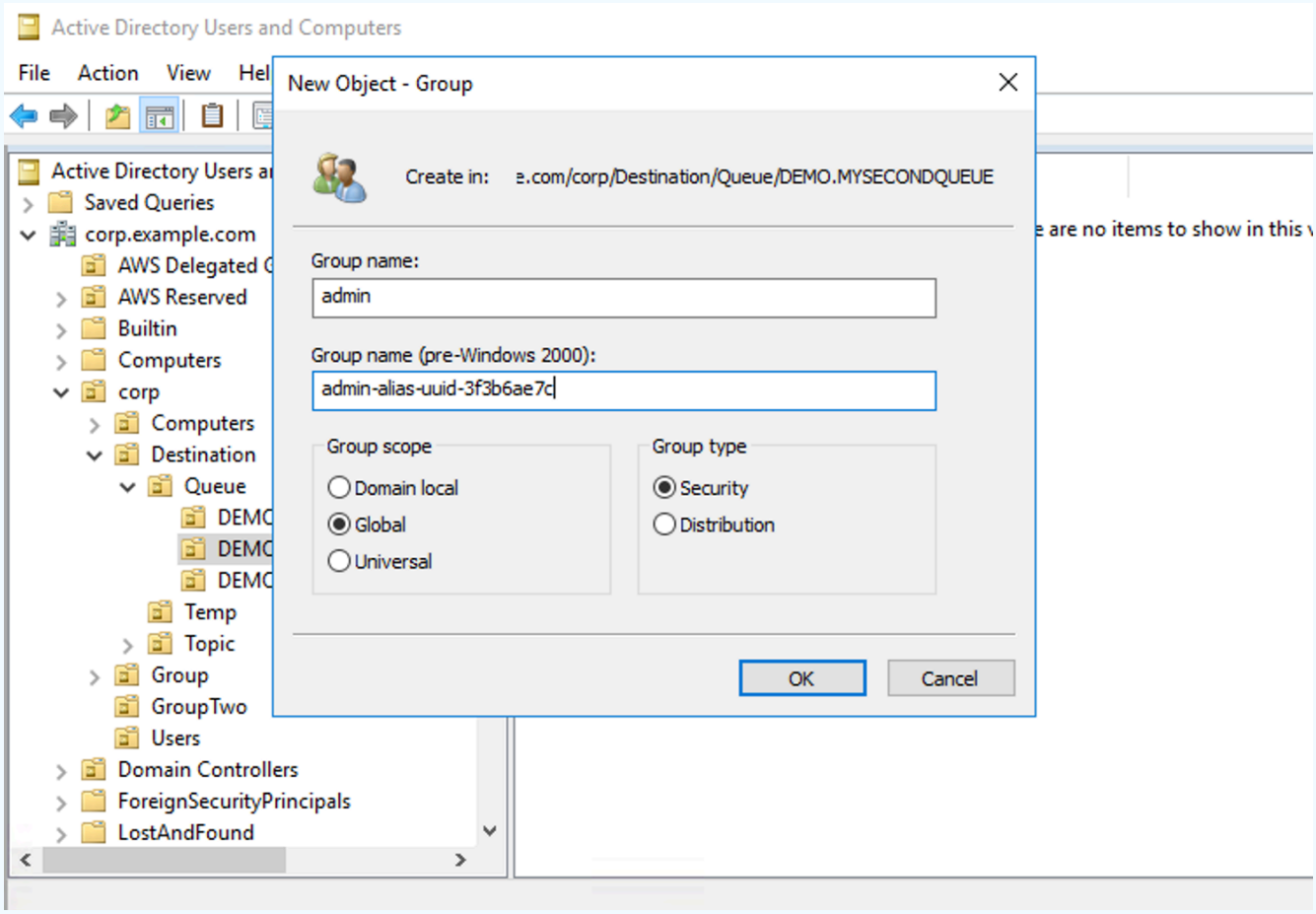
All'interno di ogni unità organizzativa che rappresenta una destinazione o un carattere jolly, è necessario creare tre gruppi di sicurezza. Come tutte le autorizzazioni in ActiveMQ, si tratta di autorizzazioni. read/write/admin Per ulteriori informazioni sulle operazioni di ciascuna utente, consultare Sicurezza nella documentazione di ActiveMQ.

È necessario assegnare un nome a questi gruppi di sicurezza `read`, `write` e `admin`. All'interno di ciascuno di questi gruppi di sicurezza è possibile aggiungere utenti o gruppi, che avranno quindi l'autorizzazione per eseguire le azioni associate. Questi gruppi di sicurezza sono necessari per ogni set di destinazione jolly o singola destinazione.



Note

Quando si crea il gruppo di amministrazione, si verificherà un conflitto con il nome del gruppo. Questo conflitto si verifica perché le regole legacy a Windows 2000 non consentono ai gruppi di condividere lo stesso nome, anche se i gruppi si trovano in posizioni diverse del DIT. Il valore nella casella di testo Pre-Windows 2000 non ha alcun impatto sulla configurazione, ma deve essere univoco a livello globale. Per evitare questo conflitto, è possibile aggiungere un suffisso `uuid` a ciascun gruppo `admin`.



L'aggiunta di un utente al gruppo di sicurezza `admin` per una determinata destinazione consentirà all'utente di creare ed eliminare tale argomento. Aggiungendoli al gruppo di sicurezza `read` consentirà loro di leggere dalla destinazione mentre aggiungerli al gruppo di sicurezza `write` consentirà loro di scrivere nella destinazione.

Oltre ad aggiungere singoli utenti alle autorizzazioni dei gruppi di sicurezza, è anche possibile aggiungere interi gruppi. Tuttavia, poiché ActiveMQ di nuovo specifica a livello di codice i nomi degli

attributi per i gruppi, è necessario assicurarsi che il gruppo che si desidera aggiungere abbia la classe dell'oggetto `groupOfNames`, come mostrato nel codice sorgente activemq.

Per fare ciò, seguire lo stesso processo di `uid` per gli utenti. Consultare Configurazione delle mappature dell'ID in utenti Active Directory e computer per Windows Server 2016 (e versioni successive).

Fase 3: (Opzionale) Connect a una AWS Lambda funzione

AWS Lambda può connettersi e utilizzare i messaggi del tuo broker Amazon MQ. Quando si connette un broker a Lambda, si crea una mappatura delle origini degli eventi che legge i messaggi da una coda e richiama la funzione in modo sincrono. La mappatura dell'origine degli eventi creata legge i messaggi dal broker in batch e li converte in un payload Lambda sotto forma di oggetto JSON.

Connessione del broker a una funzione Lambda

1. Aggiungere le seguenti autorizzazioni del ruolo IAM al ruolo di esecuzione della funzione Lambda.

   - mq: DescribeBroker
   - ec2: CreateNetworkInterface
   - ec2: DeleteNetworkInterface
   - ec2: DescribeNetworkInterfaces
   - ec2: DescribeSecurityGroups
   - ec2: DescribeSubnets
   - ec2: DescribeVpcs
   - registri: CreateLogGroup
   - registri: CreateLogStream
   - registri: PutLogEvents
   - gestore dei segreti: GetSecretValue

   > **Note**
   >
   > Senza le necessarie autorizzazioni IAM, la tua funzione non sarà in grado di leggere correttamente i record dalle risorse di Amazon MQ.

2. (Opzionale) Se hai creato un broker senza accessibilità pubblica, devi effettuare una delle seguenti operazioni per consentire a Lambda di connettersi al broker:

   - Configurare un gateway NAT per sottorete pubblica. Per ulteriori informazioni, consultare Accesso a Internet e ai servizi per funzioni connesse a un VPC nella AWS Lambda Guida per gli sviluppatori.
   - Creare una connessione tra Amazon Virtual Private Cloud (Amazon VPC) e Lambda mediante un endpoint VPC. Il tuo Amazon VPC deve inoltre connettersi agli endpoint AWS Security Token Service (AWS STS) e Secrets Manager. Per ulteriori informazioni, consulta Configuring interface VPC endpoints for Lambda nella AWS Lambda Guida per gli sviluppatori.

3. Configurare il broker come origine dell'evento per una funzione Lambda che utilizza la Console di gestione AWS. Puoi anche usare il comando. `create-event-source-mapping` AWS Command Line Interface

4. Scrivere un codice per la funzione Lambda per elaborare i messaggi utilizzati dal broker. Il payload Lambda recuperato dalla mappatura dell'origine dell'evento dipende dal tipo di motore del broker. Di seguito è riportato un esempio di payload Lambda per una coda Amazon MQ per ActiveMQ.

> **Note**
>
> Nell'esempio, `testQueue` corrisponde al nome della coda.

```
{
  "eventSource": "aws:amq",
  "eventSourceArn": "arn:aws:mq:us-
west-2:112556298976:broker:test:b-9bcfa592-423a-4942-879d-eb284b418fc8",
  "messages": {
    [
      {
        "messageID": "ID:b-9bcfa592-423a-4942-879d-eb284b418fc8-1.mq.us-
west-2.amazonaws.com-37557-1234520418293-4:1:1:1:1",
        "messageType": "jms/text-message",
        "data": "QUJDOkFBQUE=",
        "connectionId": "myJMSCoID",
        "redelivered": false,
        "destination": {
          "physicalname": "testQueue"
        },
```

```
        "timestamp": 1598827811958,
        "brokerInTime": 1598827811958,
        "brokerOutTime": 1598827811959
      },
      {
        "messageID": "ID:b-9bcfa592-423a-4942-879d-eb284b418fc8-1.mq.us-
west-2.amazonaws.com-37557-1234520418293-4:1:1:1:1",
        "messageType":"jms/bytes-message",
        "data": "3DTOOW7crj51prgVLQaGQ82S48k=",
        "connectionId": "myJMSCoID1",
        "persistent": false,
        "destination": {
          "physicalname": "testQueue"
        },
        "timestamp": 1598827811958,
        "brokerInTime": 1598827811958,
        "brokerOutTime": 1598827811959
      }
    ]
  }
}
```

Per maggiori informazioni sulla connessione di Amazon MQ a Lambda, le opzioni supportate da Lambda per un'origine dell'evento Amazon MQ e gli errori di mappatura delle origini degli eventi, vedere Uso di Lambda con Amazon MQ nella AWS Lambda Guida per gli sviluppatori.

Creazione di un utente broker ActiveMQ

Un utente ActiveMQ è una persona o un'applicazione che può accedere alle code e agli argomenti di un broker ActiveMQ. È possibile configurare gli utenti in modo che dispongano di autorizzazioni specifiche. Ad esempio, è possibile consentire ad alcuni utenti di accedere alla console Web ActiveMQ.

Un gruppo è un'etichetta semantica. È possibile assegnare un gruppo a un utente e configurare le autorizzazioni per i gruppi in modo che possano inviare, ricevere e amministrare code e argomenti specifici.

> **Note**
>
> Non è possibile configurare i gruppi indipendentemente dagli utenti. Un'etichetta di gruppo viene creata quando si aggiunge almeno un utente ed eliminata quando si rimuovono tutti gli utenti da essa.

> **Note**
>
> Il `activemq-webconsole` gruppo in ActiveMQ su Amazon MQ dispone delle autorizzazioni di amministratore per tutte le code e gli argomenti. Tutti gli utenti di questo gruppo avranno accesso come amministratore.

Nei seguenti esempi viene mostrato come creare, modificare ed eliminare gli utenti del broker Amazon MQ utilizzando la Console di gestione AWS.

Creare un nuovo utente del broker ActiveMQ

1. Accedere alla console Amazon MQ.
2. Dall'elenco dei broker, scegli il nome del tuo broker (ad esempio MyBroker), quindi scegli Visualizza dettagli.

   Nella ***`MyBroker`***pagina, nella sezione Utenti, sono elencati tutti gli utenti di questo broker.

	Username	Console access	Groups	Pending modifications
○	paolo.santos	No	Devs	
○	jane.doe	Yes	Admins	

3. Selezionare Create user (Crea utente).
4. Nella finestra di dialogo Create user (Crea utente), digitare Username (Nome utente) e Password.
5. (Facoltativo) Digitare i nomi dei gruppi cui appartiene l'utente, separati da virgole (ad esempio: `Devs, Admins`).
6. (Facoltativo) Per abilitare l'accesso dell'utente a ActiveMQ Web Console (Console Web ActiveMQ), scegliere ActiveMQ Web Console (Console Web ActiveMQ).

7. Selezionare Create user (Crea utente).

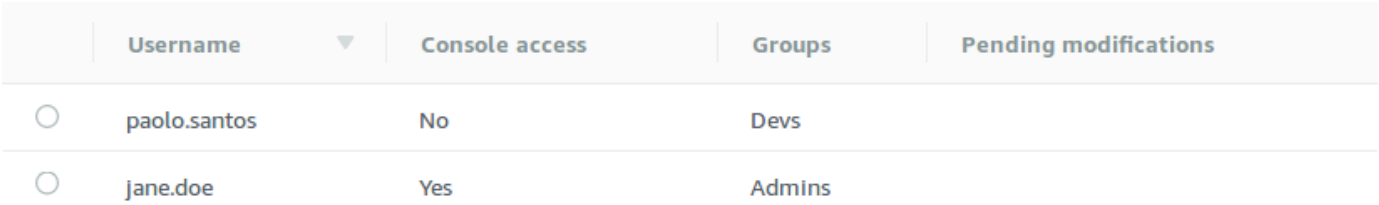
> **Important**
>
> Apportare modifiche a un utente non applica le modifiche all'utente in modo istantaneo. Per applicare le modifiche, attendere la finestra di manutenzione successiva o riavviare il broker.

Modifica un utente del broker ActiveMQ

Per modificare un utente esistente, procedi come segue:

1. Accedere alla console Amazon MQ.
2. Dall'elenco dei broker, scegli il nome del tuo broker (ad esempio MyBroker), quindi scegli Visualizza dettagli.

   Nella ***MyBroker***pagina, nella sezione Utenti, sono elencati tutti gli utenti di questo broker.



3. Seleziona le credenziali di accesso e poi Modifica.

   Viene visualizzata la finestra di dialogo Edit user (Modifica utente).
4. (Facoltativo) Digitare una nuova Password.
5. (Facoltativo) Aggiungere o rimuovere i nomi dei gruppi cui appartiene l'utente, separati da virgole (ad esempio: `Managers, Admins`).
6. (Facoltativo) Per abilitare l'accesso dell'utente a ActiveMQ Web Console (Console Web ActiveMQ), scegliere ActiveMQ Web Console (Console Web ActiveMQ).
7. Per salvare le modifiche apportate all'utente, selezionare Done (Fatto).

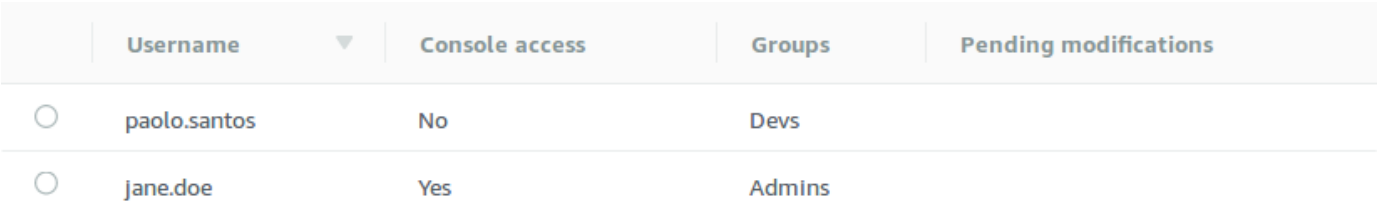
> **Important**
>
> Apportare modifiche a un utente non applica le modifiche all'utente in modo istantaneo. Per applicare le modifiche, attendere la finestra di manutenzione successiva o riavviare il broker.

Eliminare un utente del broker ActiveMQ

Quando non è più necessario un utente, è possibile eliminarlo.

1. Accedere alla console Amazon MQ.
2. Dall'elenco dei broker, scegli il nome del tuo broker (ad esempio, MyBroker), quindi scegli Visualizza dettagli.

   Nella ***MyBroker***pagina, nella sezione Utenti, sono elencati tutti gli utenti di questo broker.

3. Seleziona le tue credenziali di accesso (ad esempio ***MyUser***), quindi scegli Elimina.
4. Per confermare l'eliminazione dell'utente, nella sezione Elimina? ***MyUser*** nella finestra di dialogo, scegli Elimina.

> **Important**
>
> Apportare modifiche a un utente non applica le modifiche all'utente in modo istantaneo. Per applicare le modifiche, attendere la finestra di manutenzione successiva o riavviare il broker.

Esempi funzionanti di utilizzo di Java Message Service (JMS) con ActiveMQ

Nei seguenti esempi viene illustrato come utilizzare ActiveMQ a livello di codice:

- Il codice Java di OpenWire esempio si connette a un broker, crea una coda e invia e riceve un messaggio. Per un'analisi e una spiegazione dettagliata, consulta Connecting a Java application to your broker.
- L'esempio di codice funzionante Java MQTT si connette a un broker, crea un argomento e invia e riceve un messaggio.
- L'esempio di codice funzionante Java STOMP+WSS si connette a un broker, crea una coda e invia e riceve un messaggio.

Prerequisiti

Abilitazione attributi VPC

Per garantire che il broker sia accessibile all'interno del VPC, è necessario abilitare gli attributi VPC `enableDnsHostnames` e `enableDnsSupport`. Per ulteriori informazioni, consultare Supporto del DNS nel VPC nella Guida per l'utente di Amazon VPC.

Abilitazione delle connessioni in entrata

Per utilizzare Amazon MQ a livello di programmazione, devi utilizzare connessioni in entrata.

1. Accedere alla console Amazon MQ.
2. Dall'elenco dei broker, scegli il nome del tuo broker (ad esempio,). MyBroker
3. Nella ***`MyBroker`***pagina, nella sezione Connessioni, annota gli indirizzi e le porte dell'URL della console web del broker e dei protocolli a livello di cavo.
4. Nella sezione Details (Dettagli), in Security and network (Sicurezza e rete), scegliere il nome del gruppo di sicurezza o

   .

   Viene visualizzata la pagina Security Groups (Gruppi di sicurezza) del pannello di controllo EC2.
5. Scegli il tuo gruppo di sicurezza dall'elenco.
6. Nella parte inferiore della pagina scegli Inbound (In entrata), quindi scegli Edit (Modifica).
7. Nella finestra di dialogo Edit inbound rules (Modifica le regole in entrata), aggiungere una regola per ogni URL o endpoint che si desidera rendere accessibile pubblicamente (nell'esempio seguente viene illustrato come eseguire questa operazione per una console Web del broker).

   a. Selezionare Add Rule (Aggiungi regola).

   b. Per Type (Tipo) seleziona Custom TCP (TCP personalizzato).

c. Per Port Range (Intervallo porte), digitare la porta della console Web (`8162`).

d. Per Source (Origine), lasciare selezionato Custom (Personalizzato), quindi inserire l'indirizzo IP del sistema a cui desideri poter accedere alla console Web (ad esempio, `192.0.2.1`).

e. Scegli Save (Salva).

Il broker può ora accettare connessioni in entrata.

Aggiunta di dipendenze Java

OpenWire

Aggiungere i pacchetti `activemq-client.jar` e `activemq-pool.jar` al percorso di classe Java. L'esempio seguente mostra queste dipendenze in un file `pom.xml` di progetto Maven.

```
<dependencies>
    <dependency>
        <groupId>org.apache.activemq</groupId>
        <artifactId>activemq-client</artifactId>
        <version>5.15.16</version>
    </dependency>
    <dependency>
        <groupId>org.apache.activemq</groupId>
        <artifactId>activemq-pool</artifactId>
        <version>5.15.16</version>
    </dependency>
</dependencies>
```

Per ulteriori informazioni su `activemq-client.jar`, consultare Configurazione iniziale nella documentazione di Apache ActiveMQ.

MQTT

Aggiungi il pacchetto `org.eclipse.paho.client.mqttv3.jar` al percorso di classe Java. L'esempio seguente mostra questa dipendenza in un file `pom.xml` di progetto Maven.

```
<dependencies>
                    <dependency>
                        <groupId>org.eclipse.paho</groupId>
                        <artifactId>org.eclipse.paho.client.mqttv3</artifactId>
                        <version>1.2.0</version>
                    </dependency>
```

```
                    </dependencies>
```

Per ulteriori informazioni su `org.eclipse.paho.client.mqttv3.jar`, consulta Eclipse Paho Java Client.

STOMP+WSS

Aggiungi i pacchetti seguenti al percorso di classe Java:

- `spring-messaging.jar`
- `spring-websocket.jar`
- `javax.websocket-api.jar`
- `jetty-all.jar`
- `slf4j-simple.jar`
- `jackson-databind.jar`

L'esempio seguente mostra queste dipendenze in un file `pom.xml` di progetto Maven.

```
<dependencies>
                    <dependency>
                        <groupId>org.springframework</groupId>
                        <artifactId>spring-messaging</artifactId>
                        <version>5.0.5.RELEASE</version>
                    </dependency>
                    <dependency>
                        <groupId>org.springframework</groupId>
                        <artifactId>spring-websocket</artifactId>
                        <version>5.0.5.RELEASE</version>
                    </dependency>
                    <dependency>
                        <groupId>javax.websocket</groupId>
                        <artifactId>javax.websocket-api</artifactId>
                        <version>1.1</version>
                    </dependency>
                    <dependency>
                        <groupId>org.eclipse.jetty.aggregate</groupId>
                        <artifactId>jetty-all</artifactId>
                        <type>pom</type>
                        <version>9.3.3.v20150827</version>
                    </dependency>
                    <dependency>
```

```
            <groupId>org.slf4j</groupId>
            <artifactId>slf4j-simple</artifactId>
            <version>1.6.6</version>
        </dependency>
        <dependency>
            <groupId>com.fasterxml.jackson.core</groupId>
            <artifactId>jackson-databind</artifactId>
            <version>2.5.0</version>
        </dependency>
        </dependencies>
```

Per ulteriori informazioni, consulta STOMP Support nella documentazione di Spring Framework.

Amazon MQExample .java

> **Important**
>
> Nel codice di esempio seguente, produttori e consumatori vengono eseguiti in un singolo thread. Per i sistemi di produzione (o per testare il failover delle istanze del broker), assicurarsi che i produttori e i consumatori vengano eseguiti su host o thread separati.

OpenWire

```
/*
 * Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 *  https://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 *
 */

        import org.apache.activemq.ActiveMQConnectionFactory;
```

```
                    import org.apache.activemq.jms.pool.PooledConnectionFactory;

                    import javax.jms.*;

                    public class AmazonMQExample {

                    // Specify the connection parameters.
                    private final static String WIRE_LEVEL_ENDPOINT
                            = "ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
east-2.amazonaws.com:61617";
                    private final static String ACTIVE_MQ_USERNAME =
 "MyUsername123";
                    private final static String ACTIVE_MQ_PASSWORD =
 "MyPassword456";

                    public static void main(String[] args) throws JMSException {
                        final ActiveMQConnectionFactory connectionFactory =
                                createActiveMQConnectionFactory();
                        final PooledConnectionFactory pooledConnectionFactory =
                                createPooledConnectionFactory(connectionFactory);

                        sendMessage(pooledConnectionFactory);
                        receiveMessage(connectionFactory);

                        pooledConnectionFactory.stop();
                    }

                    private static void
                    sendMessage(PooledConnectionFactory pooledConnectionFactory)
 throws JMSException {
                        // Establish a connection for the producer.
                        final Connection producerConnection =
 pooledConnectionFactory
                                .createConnection();
                        producerConnection.start();

                        // Create a session.
                        final Session producerSession = producerConnection
                                .createSession(false, Session.AUTO_ACKNOWLEDGE);

                        // Create a queue named "MyQueue".
                        final Destination producerDestination = producerSession
                                .createQueue("MyQueue");
```

```
                        // Create a producer from the session to the queue.
                        final MessageProducer producer = producerSession
                                .createProducer(producerDestination);
                        producer.setDeliveryMode(DeliveryMode.NON_PERSISTENT);

                        // Create a message.
                        final String text = "Hello from Amazon MQ!";
                        final TextMessage producerMessage = producerSession
                                .createTextMessage(text);

                        // Send the message.
                        producer.send(producerMessage);
                        System.out.println("Message sent.");

                        // Clean up the producer.
                        producer.close();
                        producerSession.close();
                        producerConnection.close();
                    }

                    private static void
                    receiveMessage(ActiveMQConnectionFactory connectionFactory)
throws JMSException {
                        // Establish a connection for the consumer.
                        // Note: Consumers should not use PooledConnectionFactory.
                        final Connection consumerConnection =
connectionFactory.createConnection();
                        consumerConnection.start();

                        // Create a session.
                        final Session consumerSession = consumerConnection
                                .createSession(false, Session.AUTO_ACKNOWLEDGE);

                        // Create a queue named "MyQueue".
                        final Destination consumerDestination = consumerSession
                                .createQueue("MyQueue");

                        // Create a message consumer from the session to the queue.
                        final MessageConsumer consumer = consumerSession
                                .createConsumer(consumerDestination);

                        // Begin to wait for messages.
                        final Message consumerMessage = consumer.receive(1000);
```

```
                        // Receive the message when it arrives.
                        final TextMessage consumerTextMessage = (TextMessage)
 consumerMessage;
                        System.out.println("Message received: " +
 consumerTextMessage.getText());

                        // Clean up the consumer.
                        consumer.close();
                        consumerSession.close();
                        consumerConnection.close();
                    }

                    private static PooledConnectionFactory
                    createPooledConnectionFactory(ActiveMQConnectionFactory
 connectionFactory) {
                        // Create a pooled connection factory.
                        final PooledConnectionFactory pooledConnectionFactory =
                                new PooledConnectionFactory();

 pooledConnectionFactory.setConnectionFactory(connectionFactory);
                        pooledConnectionFactory.setMaxConnections(10);
                        return pooledConnectionFactory;
                    }

                    private static ActiveMQConnectionFactory
 createActiveMQConnectionFactory() {
                        // Create a connection factory.
                        final ActiveMQConnectionFactory connectionFactory =
                                new ActiveMQConnectionFactory(WIRE_LEVEL_ENDPOINT);

                        // Pass the sign-in credentials.
                        connectionFactory.setUserName(ACTIVE_MQ_USERNAME);
                        connectionFactory.setPassword(ACTIVE_MQ_PASSWORD);
                        return connectionFactory;
                    }
                    }
```

MQTT

```
/*
 * Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
```

```
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 *  https://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 *
 */

                    import org.eclipse.paho.client.mqttv3.*;

                    public class AmazonMQExampleMqtt implements MqttCallback {

                    // Specify the connection parameters.
                    private final static String WIRE_LEVEL_ENDPOINT =
                            "ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
east-2.amazonaws.com:8883";
                    private final static String ACTIVE_MQ_USERNAME =
 "MyUsername123";
                    private final static String ACTIVE_MQ_PASSWORD =
 "MyPassword456";

                    public static void main(String[] args) throws Exception {
                        new AmazonMQExampleMqtt().run();
                    }

                    private void run() throws MqttException, InterruptedException {

                        // Specify the topic name and the message text.
                        final String topic = "myTopic";
                        final String text = "Hello from Amazon MQ!";

                        // Create the MQTT client and specify the connection
 options.
                        final String clientId = "abc123";
                        final MqttClient client = new
 MqttClient(WIRE_LEVEL_ENDPOINT, clientId);
                        final MqttConnectOptions connOpts = new
 MqttConnectOptions();

                        // Pass the sign-in credentials.
```

```
                        connOpts.setUserName(ACTIVE_MQ_USERNAME);
                        connOpts.setPassword(ACTIVE_MQ_PASSWORD.toCharArray());

                        // Create a session and subscribe to a topic filter.
                        client.connect(connOpts);
                        client.setCallback(this);
                        client.subscribe("+");

                        // Create a message.
                        final MqttMessage message = new
MqttMessage(text.getBytes());

                        // Publish the message to a topic.
                        client.publish(topic, message);
                        System.out.println("Published message.");

                        // Wait for the message to be received.
                        Thread.sleep(3000L);

                        // Clean up the connection.
                        client.disconnect();
                    }

                    @Override
                    public void connectionLost(Throwable cause) {
                        System.out.println("Lost connection.");
                    }

                    @Override
                    public void messageArrived(String topic, MqttMessage message)
throws MqttException {
                        System.out.println("Received message from topic " + topic +
": " + message);
                    }

                    @Override
                    public void deliveryComplete(IMqttDeliveryToken token) {
                        System.out.println("Delivered message.");
                    }
                    }
```

STOMP+WSS

```
/*
 * Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 *  https://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 *
 */

                    import
 org.springframework.messaging.converter.StringMessageConverter;
                    import org.springframework.messaging.simp.stomp.*;
                    import org.springframework.web.socket.WebSocketHttpHeaders;
                    import org.springframework.web.socket.client.WebSocketClient;
                    import
 org.springframework.web.socket.client.standard.StandardWebSocketClient;
                    import
 org.springframework.web.socket.messaging.WebSocketStompClient;

                    import java.lang.reflect.Type;

                    public class AmazonMQExampleStompWss {

                    // Specify the connection parameters.
                    private final static String DESTINATION = "/queue";
                    private final static String WIRE_LEVEL_ENDPOINT =
                            "wss://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
east-2.amazonaws.com:61619";
                    private final static String ACTIVE_MQ_USERNAME =
 "MyUsername123";
                    private final static String ACTIVE_MQ_PASSWORD =
 "MyPassword456";

                    public static void main(String[] args) throws Exception {
```

```
                        final AmazonMQExampleStompWss example = new
AmazonMQExampleStompWss();

                        final StompSession stompSession = example.connect();
                        System.out.println("Subscribed to a destination using
session.");
                        example.subscribeToDestination(stompSession);

                        System.out.println("Sent message to session.");
                        example.sendMessage(stompSession);
                        Thread.sleep(60000);
                    }

                    private StompSession connect() throws Exception {
                        // Create a client.
                        final WebSocketClient client = new
StandardWebSocketClient();
                        final WebSocketStompClient stompClient = new
WebSocketStompClient(client);
                        stompClient.setMessageConverter(new
StringMessageConverter());

                        final WebSocketHttpHeaders headers = new
WebSocketHttpHeaders();

                        // Create headers with authentication parameters.
                        final StompHeaders head = new StompHeaders();
                        head.add(StompHeaders.LOGIN, ACTIVE_MQ_USERNAME);
                        head.add(StompHeaders.PASSCODE, ACTIVE_MQ_PASSWORD);

                        final StompSessionHandler sessionHandler = new
MySessionHandler();

                        // Create a connection.
                        return stompClient.connect(WIRE_LEVEL_ENDPOINT, headers,
head,
                                sessionHandler).get();
                    }

                    private void subscribeToDestination(final StompSession
stompSession) {
                        stompSession.subscribe(DESTINATION, new MyFrameHandler());
                    }
```

```
                    private void sendMessage(final StompSession stompSession) {
                        stompSession.send(DESTINATION, "Hello from Amazon
MQ!".getBytes());
                    }

                    private static class MySessionHandler extends
StompSessionHandlerAdapter {
                        public void afterConnected(final StompSession stompSession,
                                                   final StompHeaders stompHeaders) {
                            System.out.println("Connected to broker.");
                        }
                    }

                    private static class MyFrameHandler implements StompFrameHandler
{
                        public Type getPayloadType(final StompHeaders headers) {
                            return String.class;
                        }

                        public void handleFrame(final StompHeaders stompHeaders,
                                                final Object message) {
                            System.out.print("Received message from topic: " +
message);
                        }
                    }
                    }
```

Gestione di Amazon MQ per le versioni del motore ActiveMQ

Apache ActiveMQ organizza i numeri di versione in base alle specifiche di controllo della versione semantico come `X.Y.Z`. Nelle implementazioni di Amazon MQ for ActiveMQ`X`, indica la versione principale`Y`, rappresenta la versione secondaria e indica il numero di versione della patch. `Z` Amazon MQ considera una modifica di versione importante se cambiano i numeri di versione principali. Ad esempio, l'aggiornamento dalla versione 5.17 alla 6.0 è considerato un aggiornamento della versione principale. Una modifica di versione è considerata secondaria se cambia solo il numero della versione secondaria o della patch. Ad esempio, l'aggiornamento dalla versione 5. Da 18 a 5. 19 è considerato un aggiornamento secondario della versione. Quando `autoMinorVersionUpgrade` è attivo, Amazon MQ aggiorna il broker alla versione patch più recente disponibile.

Amazon MQ for ActiveMQ consiglia a tutti i broker di utilizzare l'ultima versione secondaria supportata. Per istruzioni su come aggiornare la versione del motore del broker, consulta Aggiornamento di una versione del motore di brokeraggio Amazon MQ.

Versioni dei motori supportate su Amazon MQ for ActiveMQ

Il calendario di supporto della versione di Amazon MQ indica quando una versione del motore di brokeraggio raggiungerà la fine del supporto. Quando una versione raggiunge la fine del supporto, Amazon MQ aggiorna automaticamente tutti i broker di questa versione alla versione successiva supportata. Questo aggiornamento avviene durante le finestre di manutenzione programmata del broker, entro i 45 giorni successivi alla end-of-support data.

Amazon MQ fornisce un preavviso di almeno 90 giorni prima che una versione raggiunga la fine del supporto. Ti consigliamo di aggiornare il tuo broker prima della end-of-support data prevista per evitare interruzioni. Inoltre, non è possibile creare nuovi broker su versioni la cui scadenza è prevista entro 30 giorni dalla data di fine del supporto.

Versione Apache ActiveMQ	Fine del supporto su Amazon MQ
ActiveMQ 5.19 (consigliato)	
ActiveMQ 5.18	
ActiveMQ 5.17	16 giugno 2025
ActiveMQ 5.16	15 novembre 2024
ActiveMQ 5.15	16 settembre 2024

Quando crei un nuovo broker Amazon MQ per ActiveMQ, puoi specificare qualsiasi versione supportata del motore ActiveMQ. Se non specifichi il numero di versione del motore durante la creazione di un broker, Amazon MQ utilizza automaticamente il numero di versione del motore più recente.

Aggiornamenti della versione del motore

Puoi aggiornare manualmente il tuo broker in qualsiasi momento alla successiva versione principale o secondaria supportata. Quando attivi gli aggiornamenti automatici delle versioni secondarie,

Amazon MQ aggiornerà il tuo broker all'ultima versione di patch supportata durante la finestra di manutenzione.

Per ulteriori informazioni sull'aggiornamento manuale del broker, consulta. the section called "Aggiornamento della versione del motore"

Elenco di versioni del motore supportate

È possibile elencare tutte le versioni minori e principali supportate del motore utilizzando il `describe-broker-instance-options` AWS CLI comando.

```
aws mq describe-broker-instance-options
```

Per filtrare i risultati in base al motore e al tipo di istanza, utilizzare le opzioni `--engine-type` e `--host-instance-type` come mostrato di seguito.

```
aws mq describe-broker-instance-options --engine-type engine-type --host-instance-
type instance-type
```

Ad esempio, per filtrare i risultati per ActiveMQ e il tipo di istanza, *`engine-type`* sostituisci `ACTIVEMQ` con `mq.m5.large` e con. *`instance-type`* `mq.m5.large`

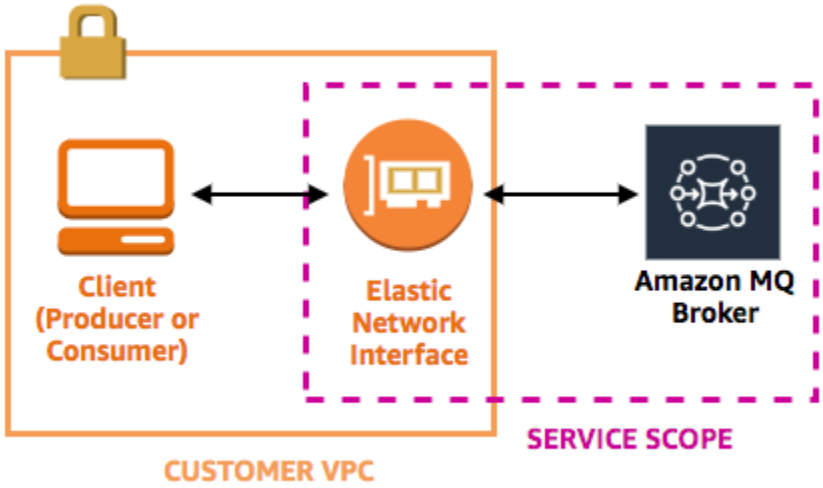
Best practice di Amazon MQ per ActiveMQ

Usa questa sezione per individuare rapidamente le raccomandazioni per massimizzare le prestazioni e ridurre al minimo i costi della velocità effettiva; quando lavori con i broker ActiveMQ su Amazon MQ.

Non modificare né eliminare mai l'interfaccia di rete elastica Amazon MQ

La prima volta che crei un broker Amazon MQ, Amazon MQ esegue il provisioning di un'interfaccia di rete elastica nel Virtual Private Cloud (VPC) nel tuo account e, pertanto, richiede una serie di autorizzazioni EC2. L'interfaccia di rete consente al client (produttore o consumatore) di comunicare con il broker Amazon MQ. L'interfaccia di rete è considerata interna all'ambito del servizio di Amazon MQ, pur facendo parte del VPC dell'account.

> **Warning**
>
> Questa interfaccia di rete deve essere modificata o eliminata. La modifica o l'eliminazione dell'interfaccia di rete può causare una perdita permanente della connessione tra il VPC e il broker.



Usa sempre il pooling delle connessioni

In uno scenario con un singolo produttore e un singolo consumatore (ad esempio il tutorial Guida introduttiva: creazione e connessione a un broker ActiveMQ), puoi utilizzare una singola classe `ActiveMQConnectionFactory` per ogni produttore e consumatore. Esempio:

```
// Create a connection factory.
final ActiveMQConnectionFactory connectionFactory = new
 ActiveMQConnectionFactory(wireLevelEndpoint);

// Pass the sign-in credentials.
connectionFactory.setUserName(activeMqUsername);
connectionFactory.setPassword(activeMqPassword);

// Establish a connection for the consumer.
final Connection consumerConnection = connectionFactory.createConnection();
consumerConnection.start();
```

Tuttavia, in scenari più realistici con più produttori e consumatori, creare un numero elevato di connessioni per più produttori può essere costoso e inefficiente. In questi scenari, è consigliabile raggruppare più richieste del produttore utilizzando la classe `PooledConnectionFactory`. Esempio:

> **Note**
>
> I consumatori dei messaggi non dovrebbero mai utilizzare la classe `PooledConnectionFactory`.

```
// Create a connection factory.
final ActiveMQConnectionFactory connectionFactory = new
 ActiveMQConnectionFactory(wireLevelEndpoint);

// Pass the sign-in credentials.
connectionFactory.setUserName(activeMqUsername);
connectionFactory.setPassword(activeMqPassword);

// Create a pooled connection factory.
final PooledConnectionFactory pooledConnectionFactory = new PooledConnectionFactory();
pooledConnectionFactory.setConnectionFactory(connectionFactory);
pooledConnectionFactory.setMaxConnections(10);

// Establish a connection for the producer.
final Connection producerConnection = pooledConnectionFactory.createConnection();
producerConnection.start();
```

Usa sempre Failover Transport per la connessione a più endpoint del broker

Se è necessario che l'applicazione si connetta a più endpoint del broker, ad esempio quando si utilizza una modalità di implementazione attiva/standby o quando si esegue una migrazione da un broker di messaggistica on-premise ad Amazon MQ, usare il Trasporto del failover per consentire ai consumatori di connettersi in modo casuale a uno dei due. Esempio:

```
failover:(ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
east-2.amazonaws.com:61617,ssl://b-9876l5k4-32ji-109h-8gfe-7d65c4b132a1-2.mq.us-
west-2.amazonaws.com:61617)?randomize=true
```

> **Important**
>
> I broker con zone di disponibilità multiple possono subire failover durante le finestre di manutenzione e il riavvio del broker. Utilizzate il Failover Transport per garantire la disponibilità del broker.

Evita l'uso di selettori di messaggi

È possibile usare selettori JMS per collegare filtri ad abbonamenti ad argomenti (per instradare i messaggi ai consumatori in base ai loro contenuti). Tuttavia, l'uso di selettori JMS riempie il buffer filtro del broker Amazon MQ, impedendo che i messaggi vengano filtrati.

In generale, non consentire ai consumatori di instradare i messaggi perché, per il disaccoppiamento ottimale di consumatori e produttori, questi devono essere entrambi temporanei.

Preferisci destinazioni virtuali ad abbonamenti durevoli

Un abbonamento durevole garantisce che il consumatore riceva tutti i messaggi pubblicati in un argomento, ad esempio, dopo il ripristino di una connessione persa. Tuttavia, l'uso di abbonamenti durevoli preclude anche l'uso di consumatori concorrenti e può causare problemi di prestazioni su scala. Valuta se utilizzare invece destinazioni virtuali.

Se utilizzi il peering di Amazon VPC, evita i client IPs nell'intervallo CIDR `10.0.0.0/16`

Se stai configurando il peering Amazon VPC tra l'infrastruttura locale e il tuo broker Amazon MQ, non devi configurare connessioni client comprese nell'intervallo CIDR. IPs `10.0.0.0/16`

Disabilita archiviazione e invio simultaneo per code con consumatori lenti

Per impostazione predefinita, Amazon MQ è ottimizzato per code con consumatori veloci:

- I consumatori sono considerati veloci se sono in grado di tenere il passo con la frequenza dei messaggi generati dai produttori.
- I consumatori sono considerati lenti se una coda crea un backlog di messaggi non riconosciuti, causando potenzialmente un decremento del throughput del produttore.

Per richiedere ad Amazon MQ di ottimizzare le code con consumatori lenti, impostare l'attributo `concurrentStoreAndDispatchQueues` su `false`. Per un esempio di configurazione, consulta `concurrentStoreAndDispatchQueues`.

Scegli il tipo di istanza broker corretta per il miglior throughput

Il throughput dei messaggi di un tipo di istanza broker dipende dal caso d'uso dell'applicazione e dai seguenti fattori:

- Uso di ActiveMQ in modalità persistente
- Dimensione dei messaggi
- Il numero di produttori e consumatori
- Il numero di destinazioni

Comprensione della relazione tra dimensione dei messaggi, latenza e velocità effettiva

A seconda del caso d'uso, un tipo di istanza broker più grande potrebbe non necessariamente migliorare il throughput del sistema. Quando ActiveMQ scrive messaggi in uno storage durevole, le dimensioni dei messaggi determinano il fattore di limitazione del sistema:

- Se le dimensioni dei messaggi sono inferiori a 100 KB, la latenza di storage persistente è il fattore di limitazione.
- Se le dimensioni dei messaggi sono superiori a 100 KB, il throughput di storage persistente è il fattore di limitazione.

Quando utilizzi ActiveMQ in modalità persistente, la scrittura nello storage avviene normalmente quando ci sono pochi consumatori o quando i consumatori sono lenti. In modalità non persistente, la scrittura nello storage avviene anche con consumatori lenti se la memoria heap dell'istanza broker è piena.

Per determinare il miglior tipo di istanza broker per l'applicazione, ti consigliamo di testare tipi di istanza broker diversi. Per ulteriori informazioni, consultare Broker instance types e anche Misurazione della velocità effettiva per Amazon MQ con il valore di riferimento di JMS.

Casi d'uso per tipi di istanza del broker più grandi

Vi sono tre casi d'uso comune quando tipi di istanza broker più grandi migliorano il throughput:

- Modalità non persistente: quando l'applicazione è meno sensibile alla perdita di messaggi durante il failover dell'istanza del broker (ad esempio, durante la trasmissione di punteggi sportivi), spesso è possibile usare la modalità non persistente di ActiveMQ. In questa modalità, ActiveMQ scrive messaggi nello storage persistente solo se la memoria heap dell'istanza broker è piena. I sistemi che utilizzano la modalità non persistente possono trarre vantaggio dalla maggiore quantità di memoria, CPU ottimizzata e rete più rapida disponibile su tipi di istanza broker più grandi.
- Consumatori veloci: quando sono disponibili consumatori attivi e il flag `concurrentStoreAndDispatchQueues` è abilitato, ActiveMQ consente ai messaggi di passare direttamente dal produttore al consumatore senza inviare messaggi all'archiviazione (anche in modalità persistente). Se l'applicazione può consumare messaggi rapidamente (o è possibile progettare i consumatori affinché lo facciano), l'applicazione può trarre vantaggio da un tipo di istanza broker più grande. Per consentire all'applicazione di consumare messaggi più rapidamente, aggiungi thread consumatore alle istanze dell'applicazione o dimensiona l'applicazione verticalmente o orizzontalmente.
- Transazioni in batch: quando si utilizza la modalità persistente e si inviano più messaggi per transazione, è possibile ottenere una velocità effettiva dei messaggi complessiva più elevata utilizzando tipi di istanza del broker più grandi. Per ulteriori informazioni, consulta Should I Use Transactions? nella documentazione di ActiveMQ.

Scegli il tipo di archiviazione del broker corretto per il miglior throughput

Per sfruttare l'elevata durata e la replica in più zone di disponibilità, utilizza Amazon EFS. Per sfruttare la bassa latenza e la velocità effettiva elevata, utilizza Amazon EBS. Per ulteriori informazioni, consulta Storage.

Configura la rete di broker nel modo corretto

Quando crei una rete di broker, configurala correttamente per l'applicazione:

- Abilita la modalità persistente: poiché (rispetto ai peer) ogni istanza del broker agisce come un produttore o un consumatore, le reti del broker non forniscono la replica distribuita di messaggi. Il primo broker che agisce come consumatore riceve un messaggio e lo mantiene nello storage. Questo broker invia una conferma al produttore e inoltra il messaggio al prossimo broker. Quando il secondo broker riconosce la persistenza del messaggio, il primo broker elimina il messaggio.

  Se la modalità persistente è disattivata, il primo broker riconosce il produttore senza mantenere il messaggio nello storage. Per ulteriori informazioni, consulta Archiviazione di messaggi replicati e

Qual è la differenza tra consegna persistente e non persistente? nella documentazione di Apache ActiveMQ.

- Non disabilitare messaggi informativi per le istanze del broker: per ulteriori informazioni, consulta Messaggio di avviso nella documentazione di Apache ActiveMQ.
- Non utilizzare individuazione di broker multicast: Amazon MQ non supporta l'individuazione di broker tramite multicast. Per ulteriori informazioni, consulta Qual è la differenza tra individuazione, multicast e zeroconf? nella documentazione di Apache ActiveMQ.

Evita riavvi lenti ripristinando transazioni XA preparate

ActiveMQ supporta transazioni distribuite (XA). Sapere in che modo ActiveMQ elabora le transazioni XA può evitare lenti tempi di ripristino per riavvii broker e failover del broker in Amazon MQ

Transazioni XA preparate non risolte vengono riprodotte a ogni riavvio. Se queste rimangono non risolte, il loro numero crescerà nel tempo, aumentando in modo significativo il tempo necessario per avviare il broker. Questo influisce sul tempo di riavvio e di failover. Occorre risolvere queste transazioni con un `commit()` o un `rollback()` per evitare il degrado delle prestazioni nel tempo.

Per monitorare le transazioni XA preparate non risolte, puoi utilizzare la `JournalFilesForFastRecovery` metrica in Amazon Logs. CloudWatch Se questo numero aumenta o è costantemente maggiore di 1, è opportuno recuperare le transazioni non risolte con un codice simile a quello dell'esempio seguente. Per ulteriori informazioni, consulta Quote in Amazon MQ.

Il codice di esempio seguente descrive in dettaglio le transazioni XA preparate e le chiude con un `rollback()`.

```
import org.apache.activemq.ActiveMQXAConnectionFactory;

import javax.jms.XAConnection;
import javax.jms.XASession;
import javax.transaction.xa.XAResource;
import javax.transaction.xa.Xid;

public class RecoverXaTransactions {
    private static final ActiveMQXAConnectionFactory ACTIVE_MQ_CONNECTION_FACTORY;
    final static String WIRE_LEVEL_ENDPOINT =
            "tcp://localhost:61616";;
    static {
```

```
        final String activeMqUsername = "MyUsername123";
        final String activeMqPassword = "MyPassword456";
        ACTIVE_MQ_CONNECTION_FACTORY = new
 ActiveMQXAConnectionFactory(activeMqUsername, activeMqPassword, WIRE_LEVEL_ENDPOINT);
        ACTIVE_MQ_CONNECTION_FACTORY.setUserName(activeMqUsername);
        ACTIVE_MQ_CONNECTION_FACTORY.setPassword(activeMqPassword);
    }

    public static void main(String[] args) {
        try {
            final XAConnection connection =
 ACTIVE_MQ_CONNECTION_FACTORY.createXAConnection();
            XASession xaSession = connection.createXASession();
            XAResource xaRes = xaSession.getXAResource();

            for (Xid id : xaRes.recover(XAResource.TMENDRSCAN)) {
                xaRes.rollback(id);
            }
            connection.close();

        } catch (Exception e) {
        }
    }
}
```

In uno scenario reale, puoi controllare le transizioni XA preparate rispetto al sistema di gestione delle transazioni XA. Puoi quindi decidere se gestire ogni transazione preparata con un `rollback()` o un `commit()`.

Utilizzo di Amazon MQ per RabbitMQ

Amazon MQ semplifica la creazione di un broker di messaggistica con le risorse di calcolo e archiviazione che soddisfano le tue esigenze. Puoi creare, gestire ed eliminare broker utilizzando l' Console di gestione AWS API REST di Amazon MQ o il AWS Command Line Interface.

Questa sezione descrive gli elementi di base di un broker di messaggistica per i tipi di motore ActiveMQ e RabbitMQ, elenca i tipi di istanza del broker Amazon MQ disponibili e i relativi stati e fornisce una panoramica dell'architettura del broker e delle opzioni di configurazione.

Per ulteriori informazioni su Amazon MQ REST APIs, consulta l'Amazon MQ REST API Reference.

Cos'è un broker Amazon MQ for RabbitMQ?

Un broker è un ambiente broker dei messaggi in esecuzione su Amazon MQ. Costituisce l'elemento di base di Amazon MQ. La descrizione combinata della classe dell'istanza del broker (`m7g`) e della dimensione (`large`,`medium`) è denominata tipo di istanza del broker (ad esempio,`mq.m7g.large`).

- Un broker a istanza singola è costituito da un broker in una zona di disponibilità dietro un Network Load Balancer (NLB). Il broker comunica con l'applicazione e con un volume di archiviazione Amazon EBS.
- Un'implementazione cluster è un raggruppamento logico di tre nodi di broker RabbitMQ dietro un load balancer di rete, ognuno dei quali condivide utenti, code e uno stato distribuito su più zone di disponibilità.

Per ulteriori informazioni, consulta Implementazione di un broker RabbitMQ.

Porte del listener

I broker RabbitMQ gestiti da Amazon MQ supportano le seguenti porte listener per la connettività a livello di applicazione tramite. `amqps` Puoi anche utilizzare queste porte per le connessioni client utilizzando la console web RabbitMQ e l'API di gestione. Tutte le connessioni utilizzano la crittografia TLS per motivi di sicurezza.

- Porta listener`5671`: utilizzata per connessioni AMQP sicure effettuate tramite l'URL AMQP sicuro. Questa porta supporta i protocolli AMQP 0-9-1 e AMQP 1.0 in RabbitMQ 4. Ad esempio, dato un

broker con ID broker `b-c8352341-ec91-4a78-ad9c-a43f23d325bb`, distribuito nella regione `us-west-2`, questo è l'URL `amqps` completo del broker: `b-c8352341-ec91-4a78-ad9c-a43f23d325bb.mq.us-west-2.amazonaws.com:5671`.

- Porte listener `443` e `15671` - È possibile utilizzare entrambe le porte listener in modo intercambiabile per accedere a un broker tramite la console web RabbitMQ o l'API di gestione. La porta 443 fornisce l'accesso HTTPS standard, mentre la porta 15671 è la tradizionale porta di gestione RabbitMQ con crittografia TLS.

Attributes

Un broker RabbitMQ dispone di diversi attributi:

- un nome; Ad esempio, `MyBroker`.
- un ID; Ad esempio, `b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`.
- un Amazon Resource Name (ARN); Ad esempio, `arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`.
- un URL della console Web RabbitMQ; Ad esempio, `https://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com`.

  Per ulteriori informazioni, consultare Console Web RabbitMQ nella documentazione di RabbitMQ.
- un endpoint AMQP sicuro. Ad esempio, `amqps://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com`.

Per un elenco completo di attributi del broker, consultare le sezioni seguenti in Riferimento all'API REST di Amazon MQ:

- ID operazione REST: broker
- ID operazione REST: broker
- ID operazione REST: riavvio broker

Gestione delle versioni del motore Amazon MQ per RabbitMQ

RabbitMQ organizza i numeri di versione in base alle specifiche di controllo della versione semantico come `X.Y.Z`. Nelle implementazioni di Amazon MQ for RabbitMQ, `X` indica la versione principale, `Y` rappresenta la versione secondaria e `Z` indica il numero di versione della patch. Amazon MQ

considera una modifica di versione importante se cambiano i numeri di versione principali. Ad esempio, l'aggiornamento dalla versione 3.13 alla 4.0 è considerato un aggiornamento della versione principale. Una modifica di versione è considerata secondaria se cambia solo il numero della versione secondaria o della patch. Ad esempio, l'aggiornamento dalla versione 3. Da 1.28 a 3. 12.13 è considerato un aggiornamento secondario della versione.

Amazon MQ for RabbitMQ consiglia a tutti i broker di utilizzare l'ultima versione supportata RabbitMQ 4.2. Per istruzioni su come aggiornare la versione del motore del broker, consulta Aggiornamento di una versione del motore di brokeraggio Amazon MQ.

Quando crei un nuovo broker Amazon MQ for RabbitMQ, devi solo specificare i numeri di versione principale e secondaria. Ad esempio, RabbitMQ 4.2. Se non specifichi la versione del motore durante la creazione di un broker, Amazon MQ utilizza automaticamente la versione più recente del motore.

> **Important**
>
> Amazon MQ non supporta gli stream. La creazione di uno stream comporterà la perdita di dati.
> Amazon MQ non supporta l'uso della registrazione strutturata in JSON.

Amazon MQ supporta due versioni principali di RabbitMQ:

- RabbitMQ 4

  Amazon MQ supporta RabbitMQ 4.2 nella serie di release RabbitMQ 4 solo sul tipo di istanza mq.m7g in tutte le dimensioni di istanza supportate.
- RabbitMQ 3

  Amazon MQ supporta RabbitMQ 3.13 nella serie di release RabbitMQ 3 su tipi di istanze mq.t3, mq.m5 e mq.m7g in tutte le dimensioni di istanze supportate.

Elenco di versioni del motore supportate

Puoi elencare tutte le versioni minori e principali supportate del motore utilizzando il comando. `describe-broker-instance-options` AWS CLI

```
aws mq describe-broker-instance-options
```

Per filtrare i risultati in base al motore e al tipo di istanza, utilizzare le opzioni `--engine-type` e `--host-instance-type` come mostrato di seguito.

```
aws mq describe-broker-instance-options --engine-type engine-type --host-instance-type instance-type
```

Ad esempio, per filtrare i risultati di RabbitMQ e digitare l'`mq.m7g.large`istanza, sostituisci *`engine-type`* con `RABBITMQ` e *`instance-type`* con. `mq.m7g.large`

RabbitMQ 4

Amazon MQ supporta RabbitMQ 4.2 nella serie di release RabbitMQ 4 solo sul tipo di istanza mq.m7g in tutte le dimensioni di istanza supportate.

> **Important**
>
> È possibile creare nuovi broker solo su RabbitMQ 4.2. Gli aggiornamenti in atto da RabbitMQ 3.13 non sono attualmente supportati.

> **Important**
>
> Il tipo di coda predefinito sui broker Amazon MQ for RabbitMQ 4.2 sarà «quorum». Se non viene specificato alcun argomento sul tipo di coda durante la creazione della coda, verrà creata una coda di quorum.
> Consigliamo vivamente di utilizzare le code quorum su RabbitMQ 4 per esigenze di durabilità, poiché non è garantito che le code classiche siano durevoli in tutti i casi.

Le seguenti modifiche sono state introdotte in RabbitMQ 4 su Amazon MQ

- AMQP 1.0 come protocollo di base: per ulteriori informazioni, consulta Protocolli.
- Pale locali: le pale ora supportano un nuovo protocollo chiamato «locale» oltre a AMQP 0-9-1 e AMQP 1.0. Gli shovel locali si basano internamente su AMQP 1.0 ma invece di utilizzare connessioni TCP separate, utilizzano connessioni intra-cluster tra i nodi del cluster e interne per la pubblicazione e il consumo di messaggi. APIs Può essere utilizzato solo per l'utilizzo e la pubblicazione all'interno dello stesso cluster e può offrire un throughput più elevato utilizzando meno risorse rispetto a AMQP 0-9-1 e AMQP 1.0.

- Le code quorum supportano le priorità dei messaggi: le priorità dei messaggi della coda quorum sono sempre attive e non richiedono una policy per funzionare. Non appena una coda quorum riceve un messaggio con una priorità impostata, abilita l'assegnazione delle priorità. Le code quorum supportano internamente solo due priorità: alta e normale. I messaggi senza una priorità impostata verranno mappati alla normalità così come le priorità da 0 a 4. I messaggi con una priorità superiore a 4 verranno mappati come alta. I messaggi ad alta priorità verranno privilegiati rispetto ai messaggi con priorità normale con un rapporto di 2:1, vale a dire per ogni 2 messaggi ad alta priorità la coda consegnerà 1 messaggio con priorità normale (se disponibile). Pertanto, le code quorum implementano una sorta di elaborazione delle priorità non rigorosa e basata sulla «condivisione equa». Ciò garantisce che si compiano sempre progressi sui messaggi con priorità normale, ma si privilegiano priorità elevate con un rapporto di 2:1.
- Khepri: Khepri viene utilizzato come archivio di metadati predefinito per i broker RabbitMQ 4
- Mutual TLS (mTLS): Amazon MQ supporta il protocollo TLS reciproco (MTLS) per i broker RabbitMQ, consentendo ai clienti di autenticarsi tramite certificati. Per ulteriori informazioni, consulta la configurazione di mTLS.
- Plugin di autenticazione dei certificati SSL: il plug-in di autenticazione SSL utilizza i certificati client delle connessioni MTLS per autenticare gli utenti, consentendo l'autenticazione tramite certificati client X.509 anziché credenziali di nome utente e password. Per ulteriori informazioni, consulta Autenticazione tramite certificato SSL.
- Plugin di autenticazione HTTP: il plug-in di backend di autenticazione HTTP consente di delegare l'autenticazione e l'autorizzazione a un servizio HTTP esterno. Per ulteriori informazioni, consulta Autenticazione e autorizzazione HTTP.
- Supporto JMS: il broker ora supporta i carichi di lavoro JMS con il plug-in di scambio di argomenti JMS abilitato, che consente alle applicazioni JMS di connettersi utilizzando il client JMS RabbitMQ.

Le seguenti funzionalità sono state dichiarate obsolete da RabbitMQ 4 su Amazon MQ

- Mirroring delle code classiche: le code classiche continuano a essere supportate senza modifiche sostanziali per le librerie e le applicazioni client, ma ora sono un tipo di coda non replicato. I client saranno in grado di connettersi a qualsiasi nodo per pubblicare e consumare contenuti da qualsiasi coda classica non replicata. Le code quorum sono consigliate per la replica e la sicurezza dei dati.
- Rimozione del QoS globale: si consiglia ai clienti di impostare il QoS per consumatore (non globale) anziché il QoS globale, in cui viene utilizzato un singolo prefetch condiviso per un intero canale.

- Support per code transitorie e non esclusive: le code transitorie sono code la cui durata è legata all'uptime del nodo su cui sono dichiarate. In un broker a singola istanza, vengono rimosse al riavvio del nodo. In una distribuzione di cluster, vengono rimossi al riavvio del nodo su cui sono ospitati. Ti consigliamo di utilizzare queue TTL per eliminare automaticamente le code inutilizzate e inattive dopo un certo periodo di inattività. Le code esclusive continuano a essere supportate e vengono eliminate una volta rimosse tutte le connessioni alla coda.

Le seguenti modifiche importanti potrebbero influire sulle tue applicazioni durante l'aggiornamento a RabbitMQ 4.2 su Amazon MQ

- Tipo di coda predefinito: Il tipo di coda predefinito su un broker RabbitMQ 4 è impostato sul quorum. Se non viene specificato alcun argomento sul tipo di coda durante la creazione della coda, verrà creata una coda di quorum.
- Il limite di riconsegna predefinito per le code del quorum è impostato su 20: i messaggi che vengono riconsegnati 20 o più volte verranno respinti o eliminati (rimossi). Se uno scenario comune per una coda è di 20 recapiti per messaggio, per tali code è necessario configurare un obiettivo con lettera morta o un limite superiore per evitare la perdita di dati. Il modo consigliato per farlo è tramite una politica.
- amqplib: le versioni amqplib del client Node JS precedenti alla 0.10.7 o qualsiasi libreria client AMQP che utilizza frame_max < 8192 non saranno in grado di connettersi a RabbitMQ
- Limiti predefiniti delle risorse: Amazon MQ per RabbitMQ ha introdotto limiti predefiniti di utilizzo delle risorse per connessioni, canali, consumatori per canale, code, vhost, pale, scambi e dimensione massima dei messaggi. Questi fungono da barriere per proteggere la disponibilità dei broker e possono essere personalizzati utilizzando configurazioni per soddisfare requisiti specifici.

Le seguenti funzionalità non sono supportate su RabbitMQ 4 su Amazon MQ

- Scambi casuali locali: gli scambi casuali locali non sono supportati su Amazon MQ poiché i nodi Amazon MQ sono protetti da un sistema di bilanciamento del carico di rete.
- Message Interceptor: gli intercettori di messaggi RabbitMQ non sono supportati su Amazon MQ.
- Parametri per coda: Amazon MQ non fornirà i parametri della coda RabbitMQ per i broker RabbitMQ 4. AWS CloudWatch Amazon MQ continuerà a fornire parametri a livello di broker. AWS CloudWatch Puoi interrogare i parametri della coda utilizzando l'API di gestione RabbitMQ. Consigliamo di interrogare le metriche per code specifiche a intervalli di un minuto o più.

Supporto per la versione di RabbitMQ

Il calendario di supporto della versione di Amazon MQ riportato di seguito indica quando una versione del motore di brokeraggio raggiungerà la fine del supporto. Quando una versione raggiunge la fine del supporto, Amazon MQ aggiorna automaticamente tutti i broker di questa versione alla versione successiva supportata. Questo aggiornamento avviene durante le finestre di manutenzione programmata del broker, entro 45 giorni dalla end-of-support data.

Amazon MQ fornisce un preavviso di almeno 90 giorni prima che una versione raggiunga la fine del supporto. Ti consigliamo di aggiornare il tuo broker prima della end-of-support data prevista per evitare interruzioni. Inoltre, non è possibile creare nuovi broker su versioni la cui scadenza è prevista entro 30 giorni dalla data di fine del supporto.

Versione RabbitMQ	Fine del supporto su Amazon MQ
4.2 (consigliato)	
3.13	
3.12	17 marzo 2025

Aggiornamenti di versione

Puoi aggiornare manualmente il tuo broker in qualsiasi momento alla successiva versione principale o secondaria supportata. Per ulteriori informazioni sull'aggiornamento manuale del broker, consulta Aggiornamento di una versione del motore di brokeraggio Amazon MQ.

Amazon MQ gestisce gli aggiornamenti all'ultima versione di patch supportata per tutti i broker RabbitMQ utilizzando la versione 3.13 e successive. Gli aggiornamenti manuali e automatici della versione si verificano durante la finestra di manutenzione pianificata o dopo il riavvio del broker.

> **Important**
>
> RabbitMQ consente solo aggiornamenti di versione incrementali (ad esempio da 3.9.x a 3.10.x). Non puoi saltare le versioni secondarie durante l'aggiornamento (ad esempio: da 3.8.x a 3.11.x).

I broker a istanza singola saranno offline durante il riavvio. Per i broker di cluster, le code con mirroring devono essere sincronizzate durante il riavvio. Con code più lunghe, il processo di sincronizzazione delle code può richiedere più tempo. Durante il processo di sincronizzazione della coda, la coda non è disponibile per consumatori e produttori. Quando il processo di sincronizzazione della coda è completo, il broker torna disponibile. Per ridurre al minimo l'impatto, consigliamo di effettuare l'upgrade durante un periodo di traffico limitato. Per ulteriori informazioni sulle best practice per gli aggiornamenti di versione, consulta. Best practice di Amazon MQ per RabbitMQ

Opzioni di distribuzione per i broker Amazon MQ for RabbitMQ

i broker RabbitMQ possono essere creati come broker a istanza singola o in un'implementazione di cluster. Per entrambe le modalità di implementazione, Amazon MQ garantisce un'elevata durata archiviando i dati in modo ridondante.

Puoi accedere ai tuoi broker RabbitMQ utilizzando qualsiasi linguaggio di programmazione supportato da RabbitMQ e abilitando TLS esplicitamente per i seguenti protocolli:

- AMQP (0-9-1)

Argomenti

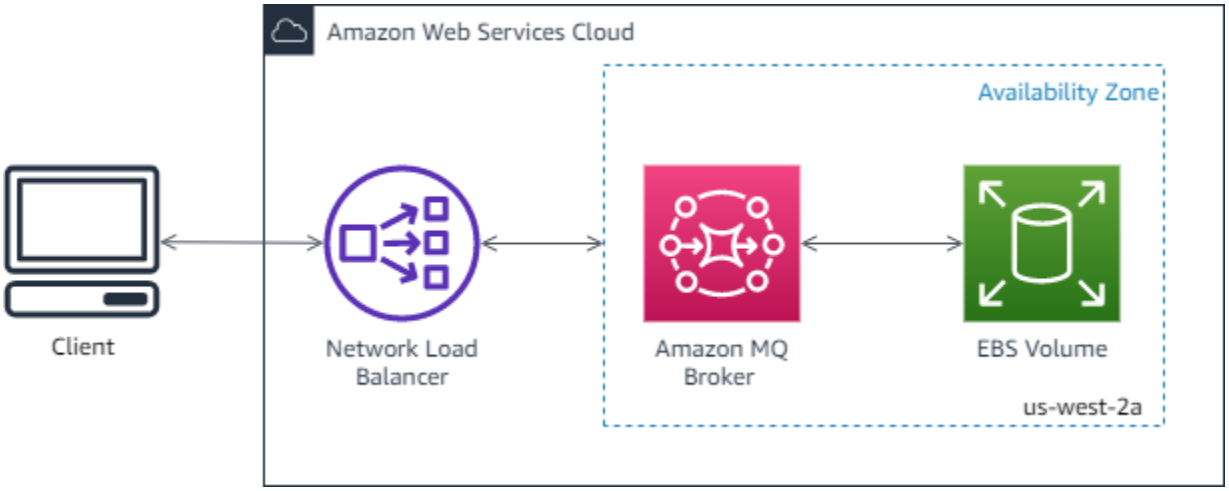
- Opzione 1: Amazon MQ per broker a istanza singola RabbitMQ
- Opzione 2: distribuzione di cluster Amazon MQ per RabbitMQ

Opzione 1: Amazon MQ per broker a istanza singola RabbitMQ

Un broker a istanza singola è composto da un broker in una zona di disponibilità dietro un load balancer di rete. Il broker comunica con l'applicazione e con un volume di archiviazione Amazon EBS. Amazon EBS fornisce archiviazione a livello di blocco ed è ottimizzato per bassa latenza e velocità effettiva elevata.

L'utilizzo di un Network Load Balancer assicura che l'endpoint del broker Amazon MQ for RabbitMQ rimanga invariato se l'istanza del broker viene sostituita durante una finestra di manutenzione o a causa di guasti hardware Amazon sottostanti. EC2 Un load balancer di rete consente alle applicazioni e agli utenti di continuare a utilizzare lo stesso endpoint per connettersi al broker.

Il seguente diagramma illustra un broker a istanza singola Amazon MQ per RabbitMQ.



Opzione 2: distribuzione di cluster Amazon MQ per RabbitMQ

Un'implementazione cluster è un raggruppamento logico di tre nodi di broker RabbitMQ dietro un load balancer di rete, ognuno dei quali condivide utenti, code e uno stato distribuito su più zone di disponibilità.

In un'implementazione cluster, Amazon MQ gestisce automaticamente le policy del broker per abilitare il mirroring classico su tutti i nodi, garantendo un'elevata disponibilità. Ogni coda sottoposta a mirroring è composta da un nodo principale e uno o più mirror. Ogni coda ha il proprio nodo principale. Tutte le operazioni per una determinata coda vengono prima applicate sul nodo principale della coda e poi propagate ai mirror. Amazon MQ crea una policy di sistema predefinita che imposta `ha-mode` su `all` e `ha-sync-mode` su `automatic`. Ciò garantisce che i dati vengano replicati in tutti i nodi del cluster in diverse zone di disponibilità per una maggiore durata.

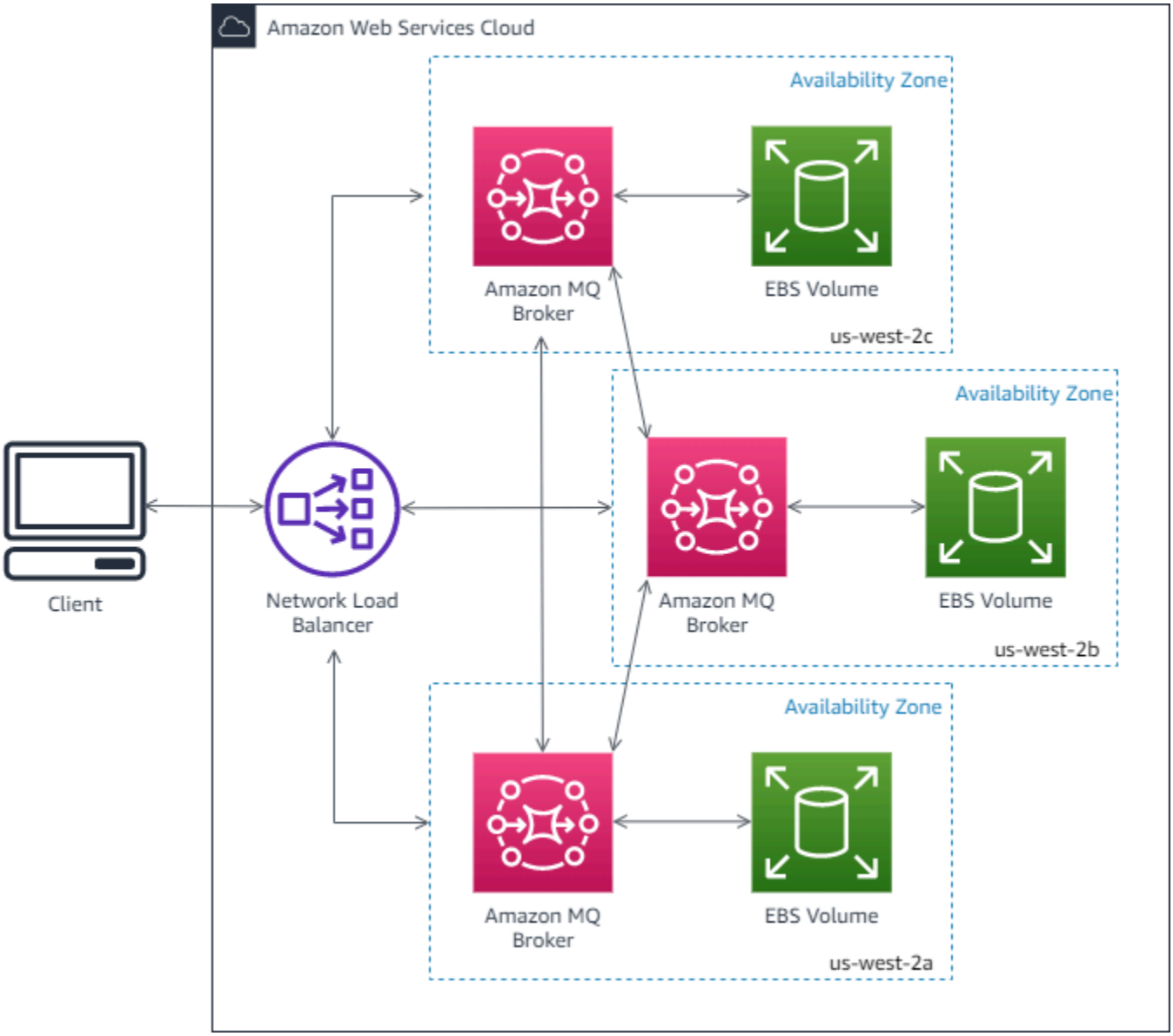
> **Note**
>
> In una distribuzione di cluster, se si verifica un'interruzione della zona di disponibilità, Amazon MQ tenterà automaticamente di riposizionare i nodi RabbitMQ interessati in una zona di disponibilità diversa per mantenere le dimensioni del cluster. Una volta risolta l'interruzione, il cluster verrà automaticamente riequilibrato su tutto il. AZs

> **Note**
>
> Durante una finestra di manutenzione, la manutenzione di un cluster viene eseguita su un nodo alla volta, mantenendo almeno due nodi in esecuzione in ogni momento. Ogni volta

che un nodo viene fermato, le connessioni client verso tale nodo vengono interrotte e devono essere ristabilite. È necessario assicurarsi che il codice client sia progettato per riconnettersi automaticamente al cluster. Per ulteriori informazioni sulla connettività remota, consultare the section called “Fase 1: Ripristino automatico in caso di guasti di rete”.
Dal momento che Amazon MQ imposta `ha-sync-mode: automatic` durante una finestra di manutenzione, le code verranno sincronizzate quando ogni nodo rientra nuovamente nel cluster. La sincronizzazione delle code blocca tutte le altre operazioni di coda. È possibile ridurre l'impatto della sincronizzazione delle code durante le finestre di manutenzione mantenendo brevi le code.

La policy predefinita non deve essere eliminata. Se elimini questa policy, Amazon MQ la ricreerà automaticamente. Amazon MQ garantisce inoltre che le proprietà di alta disponibilità vengano applicate a tutte le altre policy create su un broker del cluster. Se si aggiunge una policy senza le proprietà di alta disponibilità, Amazon MQ le aggiungerà automaticamente. Se si aggiunge una policy con proprietà di alta disponibilità diverse, Amazon MQ le sostituirà. Per ulteriori informazioni sul mirroring classico, consultare Code classiche sottoposte a mirroring.

Il diagramma seguente illustra una distribuzione del broker del cluster RabbitMQ con tre nodi in tre zone di disponibilità, ciascuno con il proprio volume Amazon EBS e uno stato condiviso. Amazon EBS fornisce archiviazione a livello di blocco ed è ottimizzato per bassa latenza e velocità effettiva elevata.



Tipi di istanze del broker Amazon MQ per RabbitMQ

La descrizione combinata della classe di istanza del broker (m7g) e della dimensione (large, medium) è denominata tipo di istanza del broker (ad esempio, mq.m7g.large).

Si consiglia di utilizzare i tipi di istanza mq.m7g per le implementazioni a cluster e a istanza singola.

Amazon MQ fornisce un preavviso di almeno 90 giorni prima che un tipo di istanza raggiunga la fine del supporto. Ti consigliamo di aggiornare il broker a un nuovo tipo di istanza prima della end-of-support data per evitare interruzioni.

> **Important**
>
> Non è possibile effettuare il downgrade di un broker da un tipo di `mq.m5` istanza `mq.m7g` or a un tipo di istanza. `mq.t3.micro`
> Il tipo di `mq.t3.micro` istanza non supporta la distribuzione in cluster.

Tipi di istanze per la distribuzione di cluster m7g

Si consiglia di utilizzare tipi di `mq.m7g.x` istanza con la distribuzione in cluster. La tabella seguente mostra i tipi di `mq.m7g.x` istanze disponibili per la distribuzione in cluster.

Tipo di istanza	VPCU	Memoria (GiB)	Larghezza di banda Network Baseline/ Burst (Gbps)	Uso consigliato	Archiviazione	Dimensione del volume del disco per nodo (GB)
mq.7g.medium	1	4	0,52/12,5	Valutazione	EBS	5
mq.m7g.grande	2	8	0,937/12,5	Produzione	EBS	15
mq.m7g.xlarge	4	16	1,876/12,5	Produzione	EBS	25
mq. 7 g. 2 x grande	8	32	3,75/15,0	Produzione	EBS	45
mq.m7 g. 4xgrande	16	64	7,5/15,0	Produzione	EBS	90
mq.m7 g. 8x grande	32	128	15 gigabit	Produzione	EBS	175

Tipo di istanza	VPCU	Memoria (GiB)	Larghezza di banda Network Baseline/ Burst (Gbps)	Uso consigliato	Archiviazione	Dimensione del volume del disco per nodo (GB)
mq.m7g. 12 x grande	48	192	22,5 Gigabit	Produzione	EBS	260
mq.m7g. 16 x grande	64	256	30 Gigabit	Produzione	EBS	345

Tipi di istanze per la distribuzione a singola istanza di m7g

La tabella seguente mostra i tipi di `mq.m7g.x` istanza disponibili per la distribuzione a istanza singola.

Tipo di istanza	VPCU	Memoria (GiB)	Larghezza di banda Network Baseline/ Burst (Gbps)	Uso consigliato	Archiviazione	Dimensione del volume del disco per nodo (GB)
mq.7g.medium	1	4	0,52/12,5	Valutazione	EBS	200
mq.m7g.grande	2	8	0,937/12,5	Produzione	EBS	200
mq.m7g.xlarge	4	16	1,876/12,5	Produzione	EBS	200

Tipo di istanza	VPCU	Memoria (GiB)	Larghezza di banda Network Baseline/ Burst (Gbps)	Uso consigliato	Archiviazione	Dimensione del volume del disco per nodo (GB)
mq. 7 g. 2 x grande	8	32	3,75/15,0	Produzione	EBS	200
mq.m7 g. 4xgrande	16	64	7,5/15,0	Produzione	EBS	200
mq.m7 g. 8x grande	32	128	15 gigabit	Produzione	EBS	200
mq.m7 g. 12 x grande	48	192	22,5 Gigabit	Produzione	EBS	200
mq.m7 g. 16 x grande	64	256	39 Gigabit	Produzione	EBS	200

Tipi di istanze per la distribuzione a **`mq.m5`** singola istanza

Le tabelle seguenti mostrano i tipi di `mq.m5.x` istanza disponibili per la distribuzione a istanza singola

Tipo di istanza	VPCU	Memoria (GiB)	Larghezza di banda Network Baseline/ Burst (Gbps)	Uso consigliato	Archiviazione	Dimensione del volume del disco per nodo (GB)
mq.t3.micro	2	1	0,064/5,0	Valutazione	EBS	20

Tipo di istanza	VPCU	Memoria (GiB)	Larghezza di banda Network Baseline/ Burst (Gbps)	Uso consigliato	Archiviazione	Dimensione del volume del disco per nodo (GB)
mq.m5. grande	2	8	0,75/10,0	Produzione	EBS	200
mq.m5.x grande	4	16	1,25/10,0	Produzione	EBS	200
mq.m 5,2 x grande	8	32	2,5/10,0	Produzione	EBS	200
mq.m 5,4xgrande	16	64	5,0/10,0	Produzione	EBS	200

Tipi di istanze per la distribuzione in cluster **`mq.m5`**

Le tabelle seguenti mostrano i tipi di `mq.m5.x` istanze disponibili per la distribuzione in cluster

Tipo di istanza	VPCU	Memoria (GiB)	Larghezza di banda Network Baseline/ Burst (Gbps)	Uso consigliato	Archiviazione	Dimensione del volume del disco per nodo (GB)
mq.5.large	2	8	0,75/10,0	Produzione	EBS	200
mq.m5.x grande	4	16	1,25/10,0	Produzione	EBS	200
mq.m 5,2 x grande	8	32	2,5/10,0	Produzione	EBS	200

Tipo di istanza	VPCU	Memoria (GiB)	Larghezza di banda Network Baseline/ Burst (Gbps)	Uso consigliato	Archiviazione	Dimensione del volume del disco per nodo (GB)
mq.m5,4xgrande	16	64	5,0/10,0	Produzione	EBS	200

Linee guida per il dimensionamento di Amazon MQ for RabbitMQ

Puoi scegliere il tipo di istanza del broker che meglio supporta la tua applicazione. Quando scegli un tipo di istanza, considera i fattori che influiranno sulle prestazioni del broker:

- il numero di client e di code
- il volume dei messaggi inviati
- messaggi conservati in memoria
- messaggi ridondanti

`m7g.medium`I tipi di istanze di broker più piccoli sono consigliati solo per testare le prestazioni delle applicazioni. Consigliamo tipi di `m7g.large` istanze broker di dimensioni maggiori o superiori o livelli di produzione di client e code, throughput elevato, messaggi in memoria e messaggi ridondanti.

> **Important**
>
> Non è possibile effettuare il downgrade di un broker da un tipo di `mq.m7g` istanza `mq.m5` or a un tipo di istanza. `mq.t3.micro`

È importante testare i broker per determinare il tipo e la dimensione dell'istanza appropriati per i requisiti di messaggistica del carico di lavoro.

Utilizza sempre i limiti di risorse predefiniti sul broker RabbitMQ 4 per determinare la dimensione dell'istanza appropriata per la tua applicazione in base alle best practice di Amazon MQ. Questi limiti di risorse predefiniti si basano sui tipi, sul tipo di `m7g` istanza e sulle code di quorum.

- Limiti di risorse predefiniti per la distribuzione m7g a singola istanza
- Limiti di risorse predefiniti per la distribuzione di cluster m7g

È possibile aumentare il valore di qualsiasi limite fino ai valori massimi definiti dal tipo di istanza e dalla modalità di distribuzione. Tuttavia, ti consigliamo vivamente di testare le prestazioni del broker con i valori aumentati prima di utilizzarli in produzione.

- Limiti massimi di risorse per la distribuzione m7g a singola istanza
- Limiti massimi di risorse per la distribuzione di cluster m7g
- Limiti massimi di risorse per la distribuzione m5 a istanza singola
- Limiti massimi di risorse per la distribuzione di cluster m5
- Messaggi di errore

> **Note**
>
> I broker RabbitMQ 3.13 non hanno limiti di risorse predefiniti, ma ti consigliamo di utilizzare i valori predefiniti suggeriti.

Limiti di risorse predefiniti

Amazon MQ for RabbitMQ supporta la configurazione dei limiti delle risorse del broker a partire da RabbitMQ 4 in poi. Quando crei un broker, Amazon MQ applica automaticamente i valori predefiniti a questi limiti di risorse. Queste impostazioni predefinite fungono da barriera per proteggere la disponibilità dei broker, soddisfacendo al contempo i modelli di utilizzo comuni dei clienti. Puoi personalizzare il comportamento del broker modificando i valori di configurazione dei limiti per soddisfare meglio i requisiti specifici del carico di lavoro.

Prima di apportare modifiche, tieni presente:

> **Important**
>
> 1. Le modifiche alla configurazione possono influire sulle prestazioni e sulla disponibilità del broker
> 2. Comprendi l'impatto prima di modificare le opzioni di configurazione predefinite

3. Verifica prima le modifiche alla configurazione in ambienti non di produzione
4. Monitora lo stato del broker dopo aver applicato le modifiche

> **Important**
>
> I valori predefiniti e gli intervalli supportati per queste configurazioni variano in base alla versione di RabbitMQ, al tipo di istanza e alla modalità di distribuzione del broker.

> **Important**
>
> Nota: l'associazione o la creazione di un broker con valori di configurazione al di fuori dell'intervallo supportato genererà una risposta di errore.

I limiti di risorse predefiniti applicati per i broker RabbitMQ 4.2 sono

- Limiti di risorse predefiniti per la distribuzione m7g a singola istanza
- Limiti di risorse predefiniti per la distribuzione di cluster m7g

Limiti di risorse predefiniti

> **Important**
>
> Amazon MQ per i broker RabbitMQ 3, l'impostazione predefinita è configurata con il limite massimo di risorse e Amazon MQ non offre la possibilità di sovrascrivere la configurazione del limite di risorse.

Valori predefiniti per i broker a istanza singola

Tipo di istanza	Connessioni per nodo	Canali per nodo	Consumatori per canale	Queues	fantasmi	Pale	Scambi	Dimensione del messaggio in byte
mq.m7g.medium	100	500	10	500	10	30	500	524288
mq.7 g. grande	1.500	4.500	10	1.000	50	50	1.000	524288
mq.m7 g.xlarge	3.000	9.000	10	2.000	100	100	2.000	524288
mq.m 7 g. 2 x grande	6.000	18.000	10	4.000	150	200	4.000	524288
mq.m 7 g. 4 x grande	12.000	36.000	10	8.000	200	400	8.000	524288
mq. 7 g. 8 x grande	24.000	72.000	10	16,000	250	800	16,000	524288
mq. 7 g. 12 x grande	36.000	108.000	10	24.000	300	1.200	24.000	524288
mq.m 7 g. 16 x grande	48.000	144.000	10	32.000	350	1.600	32.000	524288

Valori predefiniti per i broker di cluster

Tipo di istanza	Connessioni per nodo	Canali per nodo	Consumatori per canale	Queues	fantasmi	Pale	Scambi	Dimensione del messaggio in byte
mq.m7g.medium	100	300	10	100	10	10	100	524288
mq.7 g. grande	500	1500	10	1.000	50	30	1.000	524288
mq.m7 g.xlarge	1000	3000	10	2.000	100	60	2.000	524288
mq.m 7 g. 2 x grande	2000	6000	10	4.000	150	120	4.000	524288
mq.m 7 g. 4 x grande	4000	12.000	10	8.000	200	240	8.000	524288
mq. 7 g. 8 x grande	8000	24.000	10	16,000	250	480	16,000	524288
mq. 7 g. 12 x grande	12000	36.000	10	24.000	300	720	24000	524288
mq.m 7 g. 16 x grande	16,000	48.000	10	32.000	350	960	32.000	524288

Limite massimo di risorse Amazon MQ per RabbitMQ

Linee guida per il dimensionamento di m7g con code quorum per la distribuzione a singola istanza

La tabella seguente mostra i valori limite massimi per ogni tipo di istanza per i broker a istanza singola.

Tipo di istanza	Connessioni	Canali	Consumatori per canale	Queues	Fantasmi	Pale	Scambi	Dimensione del messaggio in byte
mq.m7g.medium	300	900	1.000	2.500	10	150	12500	134217728
mq.m7g.grande	5.000	15.000	1.000	20.000	1500	250	100.000	134217728
mq.m7g.xarge	10.000	30.000	1.000	30.000	1.500	500	150.000	134217728
mq. 7 g. 2 x grande	20.000	60.000	1.000	40.000	1.500	1.000	200.000	134217728
mq.m7 g. 4xgrande	40.000	120.000	1.000	60.000	1.500	2000	300.000	134217728
mq.m7 g. 8x grande	80.000	240.000	1.000	80.000	1.500	4000	400.000	134217728
mq. 7 g. 12 x grande	120.000	360.000	1.000	100.000	1.500	6.000	500.000	134217728

Tipo di istanza	Connessioni	Canali	Consumatori per canale	Queues	Fantasmi	Pale	Scambi	Dimensione del messaggio in byte
mq.m7g. 16 x grande	160.000	480.000	1.000	120.000	1.500	8.000	600.000	134217728

Linee guida per il dimensionamento di m7g con code quorum per la distribuzione dei cluster

La tabella seguente mostra i valori limite massimi per ogni tipo di istanza per i broker di cluster.

Tipo di istanza	Connessioni per nodo	Canali per nodo	Consumatori per canale	Queues	Fantasmi	Pale	Scambi	Dimensione del messaggio in byte
mq.m7g.medium	300	900	1.000	500	10	50	500	134217728
mq.m7g.grande	5.000	15.000	1.000	10.000	1.500	150	50.000	134217728
mq.m7g.xarge	10.000	30.000	1.000	15.000	1.500	300	75.000	134217728
mq.m7g.2large	20.000	60.000	1.000	20.000	1.500	600	100.000	134217728
mq.m7g. 4xgrande	40.000	120.000	1.000	30.000	1.500	1200	150.000	134217728

Tipo di istanza	Connessioni per nodo	Canali per nodo	Consumatori per canale	Queues	Fantasmi	Pale	Scambi	Dimensione del messaggio in byte
mq. 7 g. 8 x grande	80.000	240.000	1.000	40.000	1.500	2.400	200.000	134217728
mq. 7 g. 12 x grande	120.000	360.000	1.000	50.000	1.500	3.600	250.000	134217728
mq.m7 g. 16 x grande	160.000	480.000	1.000	60.000	1.500	4.800	300.000	134217728

Limiti massimi di risorse per la distribuzione a istanza singola di M5

La tabella seguente mostra i valori limite massimi per ogni tipo di istanza per i broker a istanza singola.

Tipo di istanza	Connessioni	Canali	Consumatori per canale	Queues	Fantasmi	Pale
m5.large	5.000	15.000	1.000	30.000	1500	250
m5.xlarge	10.000	30.000	1.000	60.000	1500	500
m5.2xlarge	20.000	60.000	1.000	120.000	1500	1.000
m5.4xlarge	40.000	120.000	1000	240.000	1.000	2.000

Limiti massimi di risorse per l'implementazione di cluster m5

La tabella seguente mostra i valori limite massimi per ogni tipo di istanza per i broker di cluster.

Tipo di istanza	Queues	Consumatori per canale	Pale
m5.large	10.000	1.000	150
m5.xlarge	15.000	1.000	300
m5.2xlarge	20.000	1.000	600
m5.4xlarge	30.000	1.000	1200

I seguenti limiti di connessione e canale vengono applicati per nodo:

Tipo di istanza	Connessioni	Canali
m5.large	5000	15.000
m5.xlarge	10.000	30.000
m5.2xlarge	20.000	60.000
m5.4xlarge	40.000	120.000

I valori limite esatti per un broker di cluster possono essere inferiori al valore indicato a seconda del numero di nodi disponibili e del modo in cui RabbitMQ distribuisce le risorse tra i nodi disponibili. Se superi i valori limite, puoi creare una nuova connessione a un altro nodo e riprovare, oppure puoi aggiornare la dimensione dell'istanza per aumentare i limiti massimi

Messaggi di errore

I seguenti messaggi di errore vengono restituiti quando vengono superati i limiti. Tutti i valori si basano sui limiti delle **`m7.large`** singole istanze.

> **Note**
>
> I codici di errore per i seguenti messaggi possono cambiare in base alla libreria client utilizzata.

Connessione

```
ConnectionClosedByBroker 500 "NOT_ALLOWED - connection refused: node
connection limit (5000) is reached"
```

Canale

```
ConnectionClosedByBroker 1500 "NOT_ALLOWED - number of channels opened on
node 'rabbit@ip-10-0-23-173.us-west-2.compute.internal' has reached the
maximum allowed limit of (15,000)"
```

Consumatore

```
ConnectionClosedByBroker: (530, 'NOT_ALLOWED - reached maximum (1,000) of
consumers per channel')
```

Dimensione massima del messaggio

```
(406, 'PRECONDITION_FAILED - message size 524289 is larger than configured
max size 524288')
```

Scambio

```
(406, "PRECONDITION_FAILED - cannot declare exchange 'limit_test_3' in
vhost '/': exchange limit of 10 is reached")
```

> **Note**
>
> I seguenti messaggi di errore utilizzano il formato API di gestione HTTP.

Coda

```
{"error":"bad_request","reason":"cannot declare queue 'my_queue': queue
limit in cluster (10,000) is reached"}]
```

Pala

```
{"error":"bad_request","reason":"Validation failed\n\ncomponent shovel is
limited to 150 per node\n"}
```

Fantasma

```
{"error":"bad_request","reason":"cannot create vhost 'my_vhost': vhost
limit of 1500 is reached"}
```

Impostazioni predefinite del broker Amazon MQ per RabbitMQ

Quando crei un broker Amazon MQ per RabbitMQ, Amazon MQ applica un insieme predefinito di policy del broker e limiti vhost per ottimizzare le prestazioni del tuo broker. Amazon MQ applica limiti vhost solo al valore predefinito (`/`) vhost. Amazon MQ non applicherà policy predefinite ai nuovi vhost creati. Si consiglia di mantenere questi valori predefiniti per tutti i broker nuovi ed esistenti. Tuttavia, è possibile modificare, sostituire o eliminare tali valori predefiniti in qualsiasi momento.

Amazon MQ crea politiche di broker e limiti di vhost diversi per Amazon MQ for RabbitMQ 3 e RabbitMQ 4. Le differenze verranno discusse in dettaglio nelle seguenti sottosezioni.

Amazon MQ crea criteri e limiti in base al tipo di istanza e alla modalità di implementazione del broker scelti al momento della creazione del broker. Le policy predefinite sono denominate in base alla modalità di implementazione, come indicato di seguito:

Amazon MQ per RabbitMQ 3:

- A istanza singola: `AWS-DEFAULT-POLICY-SINGLE-INSTANCE`
- Distribuzione di cluster — && `AWS-DEFAULT-POLICY-CLUSTER-MULTI-AZ` `AWS-DEFAULT-QUORUM-QUEUES-POLICY-CLUSTER-MULTI-AZ`

Amazon MQ per RabbitMQ 4:

- A istanza singola: `AWS-DEFAULT-POLICY-SINGLE-INSTANCE`
- Distribuzione di cluster — && `AWS-DEFAULT-POLICY-CLUSTER` `AWS-DEFAULT-QUORUM-QUEUES-POLICY-CLUSTER-MULTI-AZ`

Per broker a istanza singola, Amazon MQ imposta il valore di priorità della policy su `0`. Per ignorare il valore di priorità predefinito, è possibile creare policy personalizzate con valori di priorità più elevati. Per implementazioni cluster, Amazon MQ imposta il valore di priorità su 1 per le impostazioni predefinite del broker. Per creare policy personalizzate per i cluster, assegnare un valore di priorità superiore a 1.

> **Note**
>
> Nelle implementazioni cluster, le policy del broker `ha-mode` e `ha-sync-mode` sono necessarie per il mirroring classico e la disponibilità elevata. Queste impostazioni sono applicabili solo per Amazon MQ for RabbitMQ 3 e non sono configurate per RabbitMQ 4. Se si elimina la policy predefinita `AWS-DEFAULT-POLICY-CLUSTER-MULTI-AZ`, Amazon MQ utilizza la policy `ha-all-AWS-OWNED-DO-NOT-DELETE` con un valore prioritario di `0`. Ciò assicura che le policy `ha-mode` e `ha-sync-mode` siano ancora in vigore. Se crei una policy personalizzata, Amazon MQ aggiunge automaticamente `ha-mode` e `ha-sync-mode` alle definizioni della policy.

Argomenti

- Descrizioni di policy e limiti
- Valori predefiniti consigliati

Descrizioni di policy e limiti

L'elenco seguente descrive le policy e i limiti predefiniti che Amazon MQ applica a un broker appena creato. I valori per `max-length`, `max-queues` e `max-connections` variano in base al tipo di istanza e alla modalità di implementazione del broker. Questi valori sono elencati nella sezione Valori predefiniti consigliati.

Impostazioni sui broker RabbitMQ 3 e RabbitMQ 4

- **`queue-mode: lazy`** (policy): abilita le code lente. Per impostazione predefinita, le code mantengono una cache in memoria dei messaggi, consentendo al broker di recapitare i messaggi ai consumatori il più rapidamente possibile. Ciò può portare l'esaurimento della memoria del broker e l'attivazione di un allarme per il consumo elevato di memoria. Le code lente tentano di spostare i messaggi sul disco non appena risulta fattibile. Ciò implica una conservazione in memoria di un minor numero di messaggi in normali condizioni operative. Utilizzando le code lente, Amazon MQ per RabbitMQ può supportare carichi di messaggistica notevolmente maggiori e code più lunghe. Si noti che per alcuni casi d'uso, i broker con code lente potrebbero risultare leggermente più lenti. Questo perché i messaggi vengono spostati da un disco a un broker, anziché recapitare i messaggi da una cache in memoria.

> **Modalità di implementazione**
>
> A singola istanza, cluster

- **`max-length: number-of-messages`** (policy): imposta un limite per il numero di messaggi in una coda. Nelle implementazioni cluster, il limite impedisce la sincronizzazione delle code in pausa in casi quali il riavvio del broker o dopo una finestra di manutenzione.

> **Modalità di implementazione**
>
> Cluster

- **`overflow: reject-publish`** (policy): applica le code con una policy `max-length` per rifiutare nuovi messaggi dopo che il numero di messaggi nella coda raggiunge il valore `max-length`. Per evitare la perdita di messaggi se una coda è in uno stato di overflow, le applicazioni client che pubblicano messaggi nel broker devono implementare la conferma del mittente. Per ulteriori informazioni sull'implementazione delle conferme del mittente, consultare Conferma del mittente sul sito Web RabbitMQ.

> **Modalità di implementazione**
>
> Cluster

Impostazioni specifiche di RabbitMQ 3

- **`max-queues: number-of-queues-per-vhost`** (limite vhost): imposta il limite per il numero di code in un broker. Come per la definizione della policy `max-length`, limitando il numero di code nelle implementazioni cluster si impedisce la sincronizzazione delle code in pausa dopo il riavvio del broker o le finestre di manutenzione. La limitazione delle code impedisce inoltre quantità eccessive di utilizzo della CPU per la gestione delle code.

> **Modalità di implementazione**
>
> A singola istanza, cluster

- **`max-connections: number-of-connections-per-vhost`** (limite vhost): imposta il limite per il numero di connessioni client al broker. Limitare il numero di connessioni in base ai valori consigliati impedisce un utilizzo eccessivo della memoria del broker, che potrebbe comportare l'attivazione di un allarme di memoria elevata e la sospensione delle operazioni.

  Modalità di implementazione

  A singola istanza, cluster

Valori predefiniti consigliati

Important

`max-queuese max-connections` si applicano solo ad Amazon MQ per RabbitMQ 3.

Note

I limiti predefiniti `max-length` e `max-queue` vengono testati e valutati in base a una dimensione media del messaggio di 5 kB. Se i messaggi sono significativamente maggiori di 5 kB, sarà necessario regolare e ridurre i limiti `max-length` e `max-queue`.

Nella tabella seguente sono elencati i valori limite predefiniti per un broker appena creato. Amazon MQ applica questi valori in base al tipo di istanza e alla modalità di implementazione del broker.

Tipo di istanza	Deployment mode (Modalità distribuzione)	**`max-length`**	**`max-queues`**	**`max-connections`**
mq.m7g.medium	A istanza singola	N/D	2.500	100
	Cluster	500.000	100	100
mq.m7g.grande	A istanza singola	N/D	20.000	5.000
	Cluster	8.000.000	10.000	5.000

Tipo di istanza	Deployment mode (Modalità distribuzione)	**max-length**	**max-queues**	**max-connections**
mq.7 g.xlarge	A istanza singola	N/D	30.000	10.000
	Cluster	9.000.000	15.000	10.000
mq. 7 g. 2 x grande	A istanza singola	N/D	40.000	20.000
	Cluster	10.000.000	40.000	20.000
mq.m7 g. 4xgrande	A istanza singola	N/D	60.000	40.000
	Cluster	12.000.000	30.000	40.000
mq. 7 g. 8 x grande	A istanza singola	N/D	80.000	80.000
	Cluster	20.000.000	40.000	80.000
mq.m7 g. 12 x grande	A istanza singola	N/D	100.000	120.000
	Cluster	30.000.000	20.000	120.000
mq. 7 g. 16 x grande	A istanza singola	N/D	120.000	160.000
	Cluster	40.000.000	50.000	160.000

Tipo di istanza	Deployment mode (Modalità distribuzione)	**max-length**	**max-queues**	**max-connections**
t3.micro	A istanza singola	N/D	500	500
m5.large	A istanza singola	N/D	20.000	4.000
m5.large	Cluster	8.000.000	10.000	15.000
m5.xlarge	A istanza singola	N/D	30.000	8.000

Tipo di istanza	Deployment mode (Modalità distribuzione)	`max-length`	`max-queues`	`max-connections`
m5.xlarge	Cluster	9.000.000	10.000	20.000
m5.2xlarge	A istanza singola	N/D	60.000	15.000
m5.2xlarge	Cluster	10.000.000	10.000	40.000
m5.4xlarge	A istanza singola	N/D	150.000	30.000
m5.4xlarge	Cluster	12.000.000	10.000	100.000

Configurazione di un broker RabbitMQ

Una configurazione contiene tutte le impostazioni per il tuo broker RabbitMQ in formato Cuttlefish. È possibile creare una configurazione prima di creare qualsiasi broker. È quindi possibile applicare la configurazione a uno o più broker

Attributes

Una configurazione del broker dispone di diversi attributi, ad esempio:

- un nome (MyConfiguration)
- Un ID (c-1234a5b6-78cd-901e-2fgh-3i45j6k178l9)
- Un nome di risorsa Amazon (ARN) (arn:aws:mq:us-east-2:123456789012:configuration:c-1234a5b678cd-901e-2fgh-3i45j6k178l9)

Per un elenco completo di attributi della configurazione, consultare le sezioni seguenti in Riferimento all'API REST di Amazon MQ:

- ID operazione REST: configurazione
- ID operazione REST: configurazioni

Per un elenco completo degli attributi di revisione, consulta le sezioni seguenti:

- ID operazione REST: revisione configurazione
- ID operazione REST: revisioni configurazione

Argomenti

- Creazione e applicazione delle configurazioni del broker RabbitMQ
- Modifica di una revisione della configurazione di Amazon MQ per RabbitMQ
- Valori configurabili per RabbitMQ su Amazon MQ
- Supporto ARN nella configurazione di RabbitMQ

Creazione e applicazione di configurazioni del broker RabbitMQ

Una configurazione contiene tutte le impostazioni per il broker RabbitMQ nel formato Cuttlefish. È possibile creare una configurazione prima di creare qualsiasi broker. È quindi possibile applicare la configurazione a uno o più broker

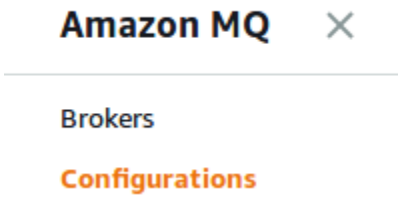
L'esempio seguente mostra come puoi creare e applicare una configurazione del broker RabbitMQ utilizzando la Console di gestione AWS.

> **Important**
>
> È possibile eliminare una configurazione solo utilizzando l'`DeleteConfiguration`API. Per ulteriori informazioni, consulta Configurazioni nell'Amazon MQ API Reference.

Creazione di una nuova configurazione

Per applicare una configurazione al tuo broker, devi prima crearla.

1. Accedere alla console Amazon MQ.
2. Nel riquadro a sinistra, espandere il pannello di navigazione e scegliere Configurations (Configurazioni).



3. Nella pagina Configurations (Configurazioni), scegliere Create configuration (Crea configurazione).
4. Nella pagina Create configuration (Crea configurazione), nella sezione Details (Dettagli), digitare il nome in Configuration name (Nome configurazione) (ad esempio, `MyConfiguration`) e selezionare una versione Broker engine (Motore del broker).

   Per ulteriori informazioni sulle versioni del motore RabbitMQ supportate da Amazon MQ per RabbitMQ, consulta the section called “Gestione della versione”.
5. Scegli Crea configurazione.

Creazione di una nuova revisione di configurazione

Dopo aver creato una configurazione, è necessario modificare la configurazione utilizzando una revisione della configurazione.

1. Dall'elenco di configurazione, scegliete ***`MyConfiguration`***.

   > **Note**
   >
   > La prima revisione di configurazione viene sempre creata automaticamente quando Amazon MQ crea la configurazione.

   `MyConfiguration`Nella pagina vengono visualizzati il tipo di motore del broker e la versione utilizzati dalla nuova revisione della configurazione (ad esempio, RabbitMQ 3.xx.xx).
2. Nella scheda Dettagli configurazione vengono visualizzati il numero di revisione di configurazione, la descrizione e la configurazione del broker in formato Cuttlefish.

   > **Note**
   >
   > La modifica della configurazione corrente crea una nuova revisione della configurazione.
3. Scegli Modifica configurazione e apporta le modifiche alla configurazione Cuttlefish.
4. Scegli Save (Salva).

   Viene visualizzata la finestra di dialogo Save revision (Salva revisione).
5. (Opzionale) Digitare `A description of the changes in this revision`.

6. Scegli Save (Salva).

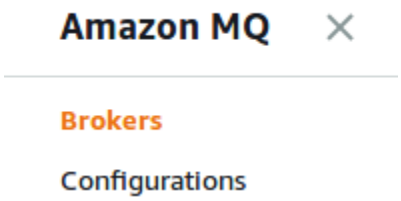
   La nuova revisione della configurazione viene salvata.

   > **Important**
   >
   > Apportare modifiche a una configurazione non applica le modifiche al broker in modo istantaneo. Per applicare le modifiche, attendere la finestra di manutenzione successiva o riavviare il broker.
   > Al momento, non è possibile eliminare una configurazione.

Applicazione di una revisione di configurazione al broker

Dopo aver creato la revisione della configurazione, puoi applicare la revisione della configurazione al tuo broker.

1. Nel riquadro a sinistra, espandere il pannello di navigazione e scegliere Brokers (Broker).

2. Dall'elenco dei broker, seleziona il tuo broker (ad esempio MyBroker), quindi scegli Modifica.
3. Nella *MyBroker* pagina Modifica, nella sezione Configurazione, seleziona una configurazione e una revisione, quindi scegli Pianifica modifiche.
4. Nella sezione Schedule broker modifications (Pianifica modifiche broker) seleziona se applicare le modifiche During the next scheduled maintenance window (Nel corso della finestra di manutenzione pianificata successiva) oppure Immediately (Immediatamente).

   > **Important**
   >
   > I broker a istanza singola sono offline durante il riavvio. Per i broker di cluster, durante il riavvio del broker viene interrotto solo un nodo alla volta.

5. Scegli Applica.

   La revisione della configurazione viene applicata al tuo broker nel momento specificato.

Modifica di una revisione della configurazione di Amazon MQ per RabbitMQ

Le seguenti istruzioni descrivono come modificare una revisione della configurazione per il tuo broker.

1. Accedere alla console Amazon MQ.
2. Dall'elenco dei broker, seleziona il tuo broker (ad esempio MyBroker), quindi scegli Modifica.
3. Nella ***MyBroker***pagina, scegli Modifica.
4. Nella *MyBroker* pagina Modifica, nella sezione Configurazione, seleziona una configurazione e una revisione, quindi scegli Modifica.

   > **Note**
   >
   > A meno che non si selezioni una configurazione durante la creazione di un broker, la prima revisione della configurazione viene sempre creata quando Amazon MQ crea il broker.

   Nella ***MyBroker***pagina, vengono visualizzati il tipo di motore del broker e la versione utilizzati dalla configurazione (ad esempio, RabbitMQ 3.xx.xx).
5. Nella scheda Dettagli configurazione vengono visualizzati il numero di revisione di configurazione, la descrizione e la configurazione del broker in formato Cuttlefish.

   > **Note**
   >
   > La modifica della configurazione corrente crea una nuova revisione della configurazione.
6. Scegli Modifica configurazione e apporta le modifiche alla configurazione Cuttlefish.
7. Scegli Save (Salva).

   Viene visualizzata la finestra di dialogo Save revision (Salva revisione).
8. (Opzionale) Digitare `A description of the changes in this revision`.
9. Scegli Save (Salva).

   La nuova revisione della configurazione viene salvata.

> **Important**
>
> Apportare modifiche a una configurazione non applica le modifiche al broker in modo istantaneo. Per applicare le modifiche, attendere la finestra di manutenzione successiva o riavviare il broker.
> Al momento, non è possibile eliminare una configurazione.

Valori configurabili

È possibile impostare il valore delle seguenti opzioni di configurazione del broker modificando il file di configurazione del broker in. Console di gestione AWS

Oltre ai valori descritti nella tabella seguente, Amazon MQ supporta opzioni di configurazione del broker aggiuntive relative all'autenticazione e all'autorizzazione, nonché ai limiti delle risorse. Per ulteriori informazioni su queste opzioni di configurazione, consulta

- OAuth Configurazione 2.0
- configurazione LDAP
- Configurazione HTTP
- Configurazione SSL
- Configurazione MTLS
- Supporto ARN
- Limiti delle risorse
- Configurazione SSL del client AMQP

Configurazione	Valore predefinito	Valore consigliato	Valori	Versioni applicabili	Description
consumer_timeout	1800000 ms (30 minuti)	1800000 ms (30 minuti)	da 0 a 2.147.483 .647 ms. Amazon MQ supporta anche il	Tutte le versioni	Un timeout nella conferma di consegna al consumatore per rilevare

Configurazione	Valore predefinito	Valore consigliato	Valori	Versioni applicabili	Description
			valore 0, che significa «infinito».		quando i consumatori non impacchettano le consegne.
pulsazione	60 secondi	60 secondi	da 60 a 3600 secondi	Tutte le versioni	Definisce il tempo prima che una connessione venga considerata non disponibile da RabbitMQ.
management.restrictions.operator_policy_changes.disabled	true	true	true, false	Tutte le versioni	Disattiva la modifica delle politiche dell'operatore. Se apporti questa modifica, consigliamo vivamente di includere le proprietà HA nelle policy degli operatori.

Configurazione	Valore predefinito	Valore consigliato	Valori	Versioni applicabili	Description
quorum_queue.property_equivalence.relaxed_checks_on_redeclaration	true	true	true, false	Tutte le versioni	Se impostata su TRUE, l'applicazione evita un'eccezione di canale quando dichiara nuovamente una coda di quorum.
secure.management.http.headers.enabled	true	true	true, false	Tutte le versioni	Attiva le intestazioni di sicurezza HTTP non modificabili.

Configurazione della conferma di consegna da parte del consumatore

Puoi configurare consumer_timeout per rilevare quando i consumatori non accettano le consegne. Se il consumatore non invia una conferma entro il valore di timeout, il canale verrà chiuso. Ad esempio, se utilizzi il valore predefinito 1800000 millisecondi, se il consumatore non invia una conferma di consegna entro 1800000 millisecondi, il canale verrà chiuso. Amazon MQ supporta anche il valore 0, che significa «infinito».

Configurazione del battito cardiaco

È possibile configurare un timeout del battito cardiaco per scoprire quando le connessioni vengono interrotte o non sono riuscite. Il valore del battito cardiaco definisce il limite di tempo prima che una connessione venga considerata inattiva.

Configurazione delle politiche dell'operatore

La policy predefinita dell'operatore su ogni host virtuale presenta le seguenti proprietà HA consigliate:

```
{
"name": "default_operator_policy_AWS_managed",
"pattern": ".*",
"apply-to": "all",
"priority": 0,
"definition": {
"ha-mode": "all",
"ha-sync-mode": "automatic"
}
}
```

Le modifiche alle politiche dell'operatore tramite l'API Console di gestione AWS o Management non sono disponibili per impostazione predefinita. Puoi abilitare le modifiche aggiungendo la riga seguente alla configurazione del broker:

```
management.restrictions.operator_policy_changes.disabled=false
```

Se apporti questa modifica, consigliamo vivamente di includere le proprietà HA nelle policy degli operatori.

Configurazione dei controlli semplificati sulla dichiarazione della coda

Se avete migrato le code classiche alle code quorum ma non avete aggiornato il codice client, potete evitare un'eccezione di canale quando dichiarate nuovamente una coda di quorum configurando quorum_queue.property_equivalence.relaxed_checks_on_redeclaration impostato su true.

Configurazione delle intestazioni di sicurezza HTTP

La configurazione secure.management.http.headers.enabled abilita le seguenti intestazioni di sicurezza HTTP:

- X-Content-Type-Options: nosniff: impedisce ai browser di eseguire lo sniffing dei contenuti, algoritmi utilizzati per dedurre il formato dei file dei siti Web.
- X-Frame-Options: DENY: impedisce ad altri di incorporare il plugin di gestione in un frame del proprio sito Web per ingannare gli altri
- Strict-Transport-Security: max-age=47304000; includeSubDomains: impone ai browser di utilizzare HTTPS quando effettuano connessioni successive al sito Web e ai suoi sottodomini per un lungo periodo di tempo (1,5 anni).

I broker Amazon MQ for RabbitMQ creati con le versioni 3.10 e successive avranno secure.management.http.headers.enabled impostato su true per impostazione predefinita. Puoi attivare queste intestazioni di sicurezza HTTP impostando secure.management.http.headers.enabled su true. Se desideri disattivare queste intestazioni di sicurezza HTTP, imposta secure.management.http.headers.enabled su false.

Configurazione dell'autenticazione e dell'autorizzazione 2.0 OAuth

Per informazioni sulle opzioni di configurazione OAuth 2.0 e sulla configurazione dell'autenticazione OAuth 2.0 per i broker, consulta Configurazioni OAuth 2.0 supportate e Utilizzo dell'autenticazione e dell'autorizzazione OAuth 2.0.

Configurazione dell'autenticazione e dell'autorizzazione LDAP

Per informazioni sulle opzioni di configurazione LDAP e sulla configurazione dell'autenticazione LDAP per i broker, consulta Configurazioni LDAP supportate e. Utilizzo dell'autenticazione e dell'autorizzazione LDAP

Configurazione dell'autenticazione e dell'autorizzazione HTTP

Per informazioni sui valori di configurazione dell'autenticazione HTTP e sulla configurazione dell'autenticazione HTTP per i tuoi broker, consulta Autenticazione e autorizzazione HTTP e. Utilizzo dell'autenticazione e dell'autorizzazione HTTP

> **Note**
>
> Il plug-in di autenticazione HTTP è disponibile solo per Amazon MQ for RabbitMQ versione 4 e successive.

Configurazione dell'autenticazione del certificato SSL

Per informazioni sui valori di configurazione dell'autenticazione del certificato SSL e sulla configurazione dell'autenticazione del certificato SSL per i tuoi broker, consulta Autenticazione del certificato SSL e. Utilizzo dell'autenticazione tramite certificato SSL

> **Note**
>
> Il plug-in di autenticazione del certificato SSL è disponibile solo per Amazon MQ for RabbitMQ versione 4 e successive.

Configurazione degli MTL

Amazon MQ for RabbitMQ supporta il protocollo TLS reciproco (MTLS) per connessioni sicure a vari endpoint e servizi esterni. mTLS offre una maggiore sicurezza richiedendo l'autenticazione del client e del server tramite certificati.

> **Note**
>
> L'uso di autorità di certificazione private per MTL è disponibile solo per Amazon MQ for RabbitMQ versione 4 e successive.

> **Important**
>
> Amazon MQ for RabbitMQ ne impone l'uso AWS ARNs per i file di certificati e chiavi private. Vedi il supporto ARN nella configurazione di RabbitMQ per maggiori dettagli.

In questa pagina

- Endpoint AMQP
- Plugin di gestione RabbitMQ
- Plugin RabbitMQ 2.0 OAuth
- Plugin di autenticazione HTTP RabbitMQ
- Plugin LDAP RabbitMQ
- Connessioni client AMQP

Endpoint AMQP

Configura MTL per le connessioni client all'endpoint AMQP. Viene utilizzato con l'autenticazione del certificato SSL. Per le configurazioni supportate, vedere. Autenticazione con certificato SSL

Plugin di gestione RabbitMQ

Configura MTL per le connessioni all'interfaccia di gestione RabbitMQ.

> **Note**
>
> L'MTLs rigoroso non è supportato per l'API di gestione.

Configurazioni supportate

`aws.arns.management.ssl.cacertfile`

File di autorità di certificazione per la convalida dei certificati client che si connettono all'interfaccia di gestione.

`management.ssl.verify`

Modalità di verifica tra pari. Valori supportati:`verify_none`, `verify_peer`

`management.ssl.depth`

Profondità massima della catena di certificati per la verifica.

`management.ssl.hostname_verification`

Modalità di verifica del nome host. Valori supportati:`wildcard`, `none`

Opzioni SSL non supportate

I seguenti valori di configurazione SSL non sono supportati:

Visualizza l'elenco completo

- `management.ssl.cert`
- `management.ssl.client_renegotiation`
- `management.ssl.dh`
- `management.ssl.dhfile`
- `management.ssl.fail_if_no_peer_cert`
- `management.ssl.honor_cipher_order`
- `management.ssl.honor_ecc_order`

- `management.ssl.key.RSAPrivateKey`
- `management.ssl.key.DSAPrivateKey`
- `management.ssl.key.PrivateKeyInfo`
- `management.ssl.log_alert`
- `management.ssl.password`
- `management.ssl.psk_identity`
- `management.ssl.reuse_sessions`
- `management.ssl.secure_renegotiate`
- `management.ssl.versions.$version`
- `management.ssl.sni`

Plugin RabbitMQ 2.0 OAuth

Configura MTL per le connessioni da Amazon MQ al provider di identità OAuth 2.0. Per le configurazioni supportate, consulta. OAuth autenticazione e autorizzazione 2.0

Plugin di autenticazione HTTP RabbitMQ

Configura MTL per le connessioni da Amazon MQ al server di autenticazione HTTP. Per le configurazioni supportate, consulta. Autenticazione e autorizzazione HTTP

Plugin LDAP RabbitMQ

Configura MTL per le connessioni da Amazon MQ al server LDAP. Per le configurazioni supportate, consulta. Autenticazione e autorizzazione LDAP

Connessioni client AMQP

Configura la verifica peer TLS per le connessioni client AMQP utilizzate da federation e shovel. Per ulteriori informazioni, consulta Configurazione SSL del client AMQP.

> **Important**
>
> Amazon MQ attualmente non supporta la configurazione dei certificati client per le connessioni client AMQP. Di conseguenza, federation e shovel non possono connettersi a broker abilitati a MTLS che richiedono l'autenticazione tramite certificato client.

Configurazione del limite di risorse

Amazon MQ for RabbitMQ supporta la configurazione dei limiti delle risorse del broker a partire da RabbitMQ 4 in poi. Quando crei un broker, Amazon MQ applica automaticamente i valori predefiniti a questi limiti di risorse. Queste impostazioni predefinite fungono da barriera per proteggere la disponibilità dei broker, soddisfacendo al contempo i modelli di utilizzo comuni dei clienti. Puoi personalizzare il comportamento del broker modificando i valori di configurazione dei limiti per soddisfare meglio i requisiti specifici del carico di lavoro. Per ulteriori dettagli sui valori predefiniti e massimi consentiti, consultathe section called “Linee guida per il dimensionamento”.

Nomi delle risorse e chiavi di configurazione

Nome risorsa	Chiave di configurazione
Connessione	`connection_max`
Canale	`channel_max_per_node`
Queue	`cluster_queue_limit`
Vhost	`vhost_max`
Pala	`runtime_parameters.limits.shovel`
Exchange	`cluster_exchange_limit`
Consumatore per canale	`consumer_max_per_channel`
Dimensione massima del messaggio	`max_message_size`

Come superare i limiti delle risorse

Puoi ignorare i limiti delle risorse utilizzando l'API Amazon MQ e la console Amazon MQ.

L'esempio seguente mostra come sovrascrivere il limite predefinito di conteggio delle code utilizzando: AWS CLI

```
aws mq update-configuration --configuration-id <config-id> --data "$(echo
 "cluster_queue_limit=500" | base64 --wrap=0)"
```

Una chiamata riuscita crea una revisione della configurazione. È necessario associare la configurazione al broker RabbitMQ e riavviare il broker per applicare l'override. Per maggiori dettagli, consulta RabbitMQ Broker Configurations

Il limite di risorse sostituisce gli errori

L'associazione o la creazione di un broker con valori di configurazione al di fuori dell'intervallo supportato genera una risposta di errore simile alla seguente:

```
Configuration Revision N for configuration:cluster_queue_limit has limit: of value:
 100000000 larger than maximum allowed limit:5000
```

Supporto ARN nella configurazione di RabbitMQ

Amazon MQ for RabbitMQ supporta i AWS ARNs valori di alcune impostazioni di configurazione di RabbitMQ. Ciò è reso possibile dal plugin della community RabbitMQ rabbitmq-aws. Questo plug-in è sviluppato e gestito da Amazon MQ e può essere utilizzato anche in broker RabbitMQ con hosting autonomo non gestiti da Amazon MQ.

⚠ Considerazioni importanti

- I valori ARN risolti recuperati dal plugin aws vengono passati direttamente al processo RabbitMQ in fase di esecuzione. Non sono memorizzati altrove nel nodo RabbitMQ.
- Amazon MQ for RabbitMQ richiede un ruolo IAM che può essere assunto da Amazon MQ per accedere alla configurazione. ARNs Questo viene configurato tramite impostazione. `aws.arns.assume_role_arn`
- Gli utenti che chiamano CreateBroker o UpdateBroker APIs con una configurazione di broker che include un ruolo IAM devono disporre dell'`iam:PassRole`autorizzazione per quel ruolo.
- Il ruolo IAM deve esistere nello stesso AWS account del broker RabbitMQ. Tutto ARNs nella configurazione deve essere presente nella stessa AWS regione del broker RabbitMQ.

- Amazon MQ aggiunge chiavi condizionali globali IAM `aws:SourceAccount` e `aws:SourceArn` quando assume il ruolo IAM. Questi valori devono essere utilizzati nella policy IAM associata al ruolo in caso di confusa protezione sostitutiva.

In questa pagina

- Chiavi supportate
- Esempi di policy IAM
- Convalida dell'accesso
- Stati di quarantena dei broker correlati
- Scenario di esempio

Chiavi supportate

Ruolo IAM richiesto

`aws.arns.assume_role_arn`

ARN del ruolo IAM che Amazon MQ assume per accedere ad altre risorse. AWS Richiesto quando viene utilizzata qualsiasi altra configurazione ARN.

Endpoint AMQP

Chiave di configurazione	Description
`aws.arns.ssl_options.cacertfile`	File di autorità di certificazione per SSL/TLS le connessioni client. Amazon MQ richiede l'utilizzo di Amazon S3 o l'archiviazione del certificato.

Plugin di gestione RabbitMQ

Chiave di configurazione	Description
`aws.arns.management.ssl.cacertfile`	File di autorità di certificazione per le connessioni all'interfaccia SSL/TLS di gestione. Amazon MQ richiede l'utilizzo di Amazon S3 o l'archiviazione del certificato.

Plugin OAuth RabbitMQ 2.0

Chiave di configurazione	Description
`aws.arns.auth_oauth2.https.cacertfile`	File di autorità di certificazione per connessioni HTTPS OAuth 2.0. Amazon MQ richiede l'utilizzo di Amazon S3 o l'archiviazione del certificato.

Plugin di autenticazione HTTP RabbitMQ

Chiave di configurazione	Description
`aws.arns.auth_http.ssl_options.cacertfile`	File di autorità di certificazione per le connessioni di autenticazione SSL/TLS HTTP. Amazon MQ richiede l'utilizzo di Amazon S3 o l'archiviazione del certificato.
`aws.arns.auth_http.ssl_options.certfile`	File di certificato per connessioni TLS reciproche tra Amazon MQ e il server di autenticazione HTTP. Amazon MQ richiede l'utilizzo di Amazon S3 o l'archiviazione del certificato.
`aws.arns.auth_http.ssl_options.keyfile`	File di chiave privata per connessioni TLS reciproche tra Amazon MQ e il server di autenticazione HTTP. Amazon MQ richiede l'utilizzo Gestione dei segreti AWS per archiviare la chiave privata.

Plugin LDAP RabbitMQ

Chiave di configurazione	Description
`aws.arns.auth_ldap.ssl_options.cacertfile`	File di autorità di certificazione per le connessioni LDAP. SSL/TLS Amazon MQ richiede l'utilizzo di Amazon S3 o l'archiviazione del certificato.
`aws.arns.auth_ldap.ssl_options.certfile`	File di certificato per connessioni TLS reciproche tra Amazon MQ e il server LDAP. Amazon MQ richiede l'utilizzo di Amazon S3 o l'archiviazione del certificato.
`aws.arns.auth_ldap.ssl_options.keyfile`	File di chiave privata per connessioni TLS reciproche tra Amazon MQ e il server LDAP. Amazon MQ richiede l'utilizzo Gestione dei segreti AWS per archiviare la chiave privata.
`aws.arns.auth_ldap.dn_lookup_bind.password`	Password per l'associazione di ricerca DN LDAP. Amazon MQ richiede l'utilizzo Gestione dei segreti AWS di memorizzare la password come valore di testo non crittografato.
`aws.arns.auth_ldap.other_bind.password`	Password per LDAP (altro collegamento). Amazon MQ richiede l'utilizzo Gestione dei segreti AWS di memorizzare la password come valore di testo non crittografato.

Esempi di policy IAM

Per esempi di policy IAM, tra cui i documenti relativi alle policy relative all'assunzione di ruoli e i documenti relativi alle policy relative ai ruoli, consulta l'esempio di implementazione CDK.

Consulta Utilizzo dell'autenticazione e dell'autorizzazione LDAP la procedura per la configurazione Gestione dei segreti AWS e le risorse di Amazon S3.

Convalida dell'accesso

Per risolvere gli scenari in cui i valori ARN non possono essere recuperati, il plugin aws supporta un endpoint dell'API di gestione RabbitMQ che può essere chiamato per verificare se Amazon MQ è in grado di assumere correttamente il ruolo e risolverlo. AWS ARNs In questo modo si evita la necessità di aggiornare la configurazione del broker, aggiornare il broker con la nuova revisione della configurazione e riavviare il broker per testare le modifiche alla configurazione.

> **Note**
>
> L'uso di questa API richiede un utente amministratore RabbitMQ esistente. Amazon MQ consiglia di creare broker di test con un utente interno oltre ad altri metodi di accesso. Scopri come abilitare sia l'autenticazione OAuth 2.0 che quella semplice (interna). Questo utente può quindi essere utilizzato per accedere all'API di convalida.

> **Note**
>
> Sebbene il plugin aws supporti il passaggio di un nuovo ruolo come input all'API di convalida, questo parametro non è supportato da Amazon MQ. Il ruolo IAM utilizzato per la convalida deve corrispondere al valore della configurazione del `aws.arns.assume_role_arn` broker.

Stati di quarantena dei broker correlati

Per informazioni sugli stati di quarantena dei broker relativi ai problemi di supporto ARN, consulta:

- RABBITMQ_INVALID_ASSUMEROLE
- RABBITMQ_INVALID_ARN_LDAP
- RABBITMQ_INVALID_ARN

Scenario di esempio

- Il broker `b-f0fc695e-2f9c-486b-845a-988023a3e55b` è stato configurato per utilizzare il ruolo IAM per accedere al segreto `<role>` Gestione dei segreti AWS `<arn>`
- Se il ruolo fornito ad Amazon MQ non dispone dell'autorizzazione di lettura sul Gestione dei segreti AWS segreto, nei log di RabbitMQ verrà visualizzato il seguente errore:

```
[error] <0.254.0> aws_arn_config: {handle_assume_role,{error,{assume_role_failed,"AWS
 service is unavailable"}}}
```

  Inoltre, il broker entrerà nello stato di quarantena. `INVALID_ASSUMEROLE` Per ulteriori informazioni, vedere INVALID_ASSUMEROLE.
- I tentativi di autenticazione LDAP falliranno con il seguente errore:

```
[error] <0.254.0> LDAP bind failed: invalid_credentials
```

- Correggi il ruolo IAM con le autorizzazioni appropriate
- Chiama l'endpoint di convalida per verificare se RabbitMQ è ora in grado di accedere al segreto:

```
curl -4su 'guest:guest' -XPUT -H 'content-type: application/json' <broker-endpoint>/
api/aws/arn/validate -d '{"assume_role_arn":"arn:aws:iam::<account-id>:role/<role-
name>","arns":["arn:aws:secretsmanager:<region>:<account-id>:secret:<secret-name>"]}'
 | jq '.'
```

Configurazione SSL del client AMQP

Federation e shovel utilizzano AMQP per la comunicazione tra broker upstream e downstream. Per impostazione predefinita, la verifica peer TLS è abilitata per i client AMQP in Amazon MQ for RabbitMQ 4. Con questa impostazione, i client federation e shovel AMQP in esecuzione su broker Amazon MQ eseguiranno la verifica tra pari quando stabiliscono connessioni con il broker upstream.

I client AMQP in esecuzione sui broker Amazon MQ supportano le stesse autorità di certificazione di Mozilla. Se non utilizzi ACM, utilizza un certificato emesso da una CA presente nell'elenco dei certificati CA inclusi da Mozilla. Se il tuo broker locale utilizza certificati di altre autorità di certificazione, la verifica SSL avrà esito negativo. Puoi disabilitare la verifica tra pari TLS per questi casi d'uso.

> **Important**
>
> Amazon MQ attualmente non supporta la configurazione dei certificati client per le connessioni client AMQP. Di conseguenza, federation e shovel non possono connettersi a broker abilitati a MTLS che richiedono l'autenticazione tramite certificato client.

> **Important**
>
> Su Amazon MQ for RabbitMQ 3, le proprietà SSL dei client AMQP sono configurate con i valori predefiniti di RabbitMQ (verify_none). Amazon MQ for RabbitMQ 3 non supporta l'override di queste impostazioni predefinite.

> **Note**
>
> Con l'`verify_peer`impostazione predefinita, puoi stabilire connessioni federative e shovel tra 2 broker Amazon MQ qualsiasi, ma ciò non supporta la creazione della connessione tra il broker Amazon MQ e broker privati o broker locali che utilizzano certificati CA non Amazon MQ. Per connetterti con broker privati o locali, devi disabilitare la verifica tra pari sul broker downstream di Amazon MQ.

Chiave di configurazione SSL del client AMQP

Configurazione	Chiave di configurazione	Valori supportati
Verifica peer SSL del client AMQP	`amqp_client.ssl_options.verify`	`verify_none` , `verify_peer`

Come sostituire la verifica peer SSL del client AMQP

Puoi sostituire la verifica peer SSL del client AMQP utilizzando l'API Amazon MQ e la console Amazon MQ sui broker RabbitMQ 4.

L'esempio seguente mostra come sovrascrivere la verifica peer SSL del client AMQP utilizzando: AWS CLI

```
aws mq update-configuration --configuration-id <config-id> --data "$(echo
 "amqp_client.ssl_options.verify=verify_none" | base64 --wrap=0)"
```

Una chiamata riuscita crea una revisione della configurazione. È necessario associare la configurazione al broker RabbitMQ e riavviare il broker per applicare l'override. Per maggiori dettagli vedi Creating and applying broker configurations

> **Important**
>
> Quando viene utilizzata`verify_none`, la crittografia SSL è ancora attiva, ma l'identità del peer non viene verificata. Utilizza questa impostazione solo quando necessario e assicurati di considerare attendibile il percorso di rete verso il broker di destinazione.

Autenticazione e autorizzazione Amazon MQ per RabbitMQ

Amazon MQ for RabbitMQ supporta i seguenti metodi di autenticazione e autorizzazione:

Autenticazione e autorizzazione semplici

Con questo metodo, gli utenti del broker vengono archiviati internamente nel broker RabbitMQ e gestiti tramite la console web o l'API di gestione. Le autorizzazioni per vhost, exchange, code e topic sono configurate direttamente in RabbitMQ. Questo è il metodo predefinito. Per ulteriori informazioni, vedere Autenticazione e autorizzazione semplici.

OAuth Autenticazione e autorizzazione 2.0

In questo metodo, gli utenti del broker e le relative autorizzazioni sono gestiti da un provider di identità OAuth 2.0 (IdP) esterno. L'autenticazione degli utenti e le autorizzazioni alle risorse per vhost, exchange, code e argomenti sono centralizzate tramite il sistema di ambito del OAuth provider 2.0. Ciò semplifica la gestione degli utenti e consente l'integrazione con i sistemi di identità esistenti. Per ulteriori informazioni, vedere Autenticazione e autorizzazione OAuth 2.0.

Autenticazione e autorizzazione IAM

Con questo metodo, gli utenti del broker si autenticano utilizzando le credenziali AWS IAM tramite la federazione in uscita IAM. Le credenziali IAM vengono utilizzate per ottenere i token JWT da AWS Security Token Service (STS) e questi token JWT fungono da token 2.0 per l'autenticazione. OAuth Questo metodo sfrutta il supporto OAuth 2.0 esistente in Amazon MQ for RabbitMQ, dove AWS funge da provider di identità OAuth 2.0. L'autenticazione degli utenti è gestita da AWS IAM, mentre le autorizzazioni delle risorse per vhost, exchange, code e argomenti sono gestite tramite policy IAM e alias di ambito configurati in RabbitMQ. Per ulteriori informazioni, consulta Autenticazione e autorizzazione IAM.

Autenticazione e autorizzazione LDAP

In questo metodo, gli utenti del broker e le relative autorizzazioni sono gestiti da un servizio di directory LDAP esterno. L'autenticazione degli utenti e le autorizzazioni delle risorse sono centralizzate tramite il server LDAP, che consente agli utenti di accedere a RabbitMQ utilizzando le credenziali del servizio di directory esistenti. Per ulteriori informazioni, vedere Autenticazione e autorizzazione LDAP.

Autenticazione e autorizzazione HTTP

In questo metodo, gli utenti del broker e le relative autorizzazioni sono gestiti da un server HTTP esterno. L'autenticazione degli utenti e le autorizzazioni delle risorse sono centralizzate tramite il server HTTP, che consente agli utenti di accedere a RabbitMQ utilizzando il proprio provider di autenticazione e autorizzazione. Per ulteriori informazioni su questo metodo, consulta Autenticazione e autorizzazione HTTP.

Autenticazione con certificato SSL

Amazon MQ supporta TLS reciproco (mTLS) per i broker RabbitMQ. Il plug-in di autenticazione SSL utilizza i certificati client delle connessioni MTLS per autenticare gli utenti. Con questo metodo, gli utenti del broker vengono autenticati utilizzando certificati client X.509 anziché credenziali di nome utente e password. Il certificato del client viene convalidato rispetto a un'autorità di certificazione (CA) affidabile e il nome utente viene estratto da un campo del certificato, ad esempio Common Name (CN) o Subject Alternative Name (SAN). Questo metodo fornisce un'autenticazione avanzata senza trasmettere credenziali sulla rete. Per ulteriori informazioni, consulta Autenticazione con certificato SSL.

> **Note**
>
> RabbitMQ supporta più metodi di autenticazione e autorizzazione da utilizzare contemporaneamente. Ad esempio, è possibile abilitare sia l'autenticazione OAuth 2.0 che quella semplice (interna). Per ulteriori informazioni, consulta la sezione del tutorial OAuth 2.0 sull'abilitazione dell'autenticazione OAuth 2.0 e semplice (interna) e la documentazione sul controllo degli accessi di RabbitMQ.
>
> Amazon MQ consiglia di creare un utente interno durante il test delle configurazioni di autenticazione. Ciò consente di convalidare la configurazione di accesso utilizzando l'API di gestione RabbitMQ. Per ulteriori informazioni, vedere Convalida dell'accesso.

Autenticazione e autorizzazione semplici

Amazon MQ per gli utenti del broker RabbitMQ

> **Note**
>
> Questo argomento descrive la gestione degli utenti del broker con il meccanismo di autenticazione e autorizzazione interno predefinito di RabbitMQ. Per informazioni su tutti i metodi di autenticazione e autorizzazione supportati, consulta Amazon MQ for RabbitMQ Authentication and Authorization.

A ogni connessione client AMQP 0-9-1 è associato un utente. Questo utente deve essere autenticato. Ogni connessione client ha come destinazione anche un host virtuale (vhost). L'utente deve disporre di una serie di autorizzazioni per questo vhost. Un utente può avere l'autorizzazione per configurare, scrivere e leggere da code e scambi in un vhost. Si specificano le credenziali dell'utente e il vhost di destinazione quando viene stabilita la connessione.

Quando crei per la prima volta un broker Amazon MQ for RabbitMQ, Amazon MQ utilizza le credenziali di accesso per creare un utente RabbitMQ con il tag `administrator`. È quindi possibile aggiungere e gestire gli utenti tramite l'API di gestione RabbitMQ o la console Web RabbitMQ. È inoltre possibile utilizzare la console Web RabbitMQ o l'API di gestione per impostare o modificare le autorizzazioni dell'utente e i tag.

> **Note**
>
> Gli utenti di RabbitMQ non verranno memorizzati o visualizzati tramite l'API utenti Amazon MQ.

> **Important**
>
> Amazon MQ for RabbitMQ non supporta il nome utente «guest» ed eliminerà l'account ospite predefinito quando crei un nuovo broker. Amazon MQ eliminerà inoltre periodicamente qualsiasi account creato dal cliente chiamato «ospite».

Per creare un nuovo utente con l'API di gestione RabbitMQ, utilizzare il seguente endpoint API e corpo della richiesta. Sostituisci *username* e *password* con le tue nuove credenziali di accesso.

```
PUT /api/users/username HTTP/1.1

    {"password":"password","tags":"administrator"}
```

Important

- Non aggiungere informazioni personali di identificazione (PII) o altre informazioni riservate o sensibili nei nomi utente dei broker. I nomi utente dei broker sono accessibili ad altri AWS servizi, inclusi i registri. CloudWatch I nomi utenti dei broker non sono destinati ad essere utilizzati per dati privati o sensibili.
- Se perdi l'accesso a tutti gli account di amministratore, consulta Ripristino dell'accesso al broker per utilizzare l'autenticazione IAM per il ripristino.

La chiave `tags` è obbligatoria ed è un elenco separato da virgole di tag per l'utente. Amazon MQ supporta i tag utente `administrator`, `management`, `monitoring` e `policymaker`.

È possibile impostare le autorizzazioni per un singolo utente utilizzando i seguenti endpoint API e corpo della richiesta. Sostituisci *vhost* e *username* con le tue informazioni. Per il vhost predefinito `/`, utilizzare `%2F`.

```
PUT /api/permissions/vhost/username HTTP/1.1

    {"configure":".*","write":".*","read":".*"}
```

Note

Le chiavi `configure`, `read` e `write` sono tutte obbligatorie.

Utilizzando il carattere jolly `.*`, questa operazione concede all'utente le autorizzazioni di lettura, scrittura e configurazione per tutte le code nel vhost specificato. Per ulteriori informazioni sulla gestione degli utenti tramite l'API di gestione RabbitMQ, consultare API HTTP di gestione di RabbitMQ.

OAuth autenticazione e autorizzazione 2.0 per Amazon MQ for RabbitMQ

Amazon MQ for RabbitMQ supporta diversi metodi di autenticazione e autorizzazione. Per informazioni su tutti i metodi supportati, consulta Autenticazione e autorizzazione per i broker Amazon MQ for RabbitMQ.

Nell'autenticazione e autorizzazione OAuth 2.0, gli utenti del broker e le relative autorizzazioni sono gestiti da un provider di identità OAuth 2.0 esterno (IdP). L'autenticazione degli utenti e le autorizzazioni alle risorse per vhost, exchange, code e topic sono centralizzate tramite il sistema di ambito del OAuth provider 2.0. Ciò semplifica la gestione degli utenti e consente l'integrazione con i sistemi di identità esistenti.

⚠ Considerazioni importanti

- OAuth L'integrazione 2.0 non è supportata sui broker Amazon MQ for ActiveMQ.
- Amazon MQ for RabbitMQ non supporta il certificato server emesso da una CA privata.
- Il plug-in RabbitMQ OAuth 2.0 non supporta gli endpoint di introspezione dei token e i token di accesso opachi. Inoltre, non esegue controlli di revoca dei token.
- È necessario includere l'autorizzazione IAM per abilitare `mq:UpdateBrokerAccessConfiguration` la OAuth versione 2.0 sui broker esistenti.
- Amazon MQ crea automaticamente un utente di sistema denominato `monitoring-AWS-OWNED-DO-NOT-DELETE` con autorizzazioni di solo monitoraggio. Questo utente utilizza il sistema di autenticazione interno di RabbitMQ anche su broker OAuth abilitati alla versione 2.0 ed è limitato al solo accesso all'interfaccia di loopback.

Per informazioni su come configurare OAuth 2.0 per i broker Amazon MQ for RabbitMQ, consulta. Utilizzo dell'autenticazione e dell'autorizzazione OAuth 2.0

In questa pagina

- Configurazioni 2.0 supportate OAuth
- Validazioni aggiuntive per l'autenticazione 2.0 OAuth

Configurazioni 2.0 supportate OAuth

Amazon MQ for RabbitMQ supporta tutte le variabili configurabili nel plug-in RabbitMQ OAuth 2.0, con le seguenti eccezioni:

- `auth_oauth2.https.cacertfile`
- `auth_oauth2.oauth_providers.{id/index}.https.cacertfile`
- `management.oauth_client_secret`

  Poiché Amazon MQ non supporta questa chiave, non supportiamo UAA come IdP.
- `management.oauth_resource_servers.{id/index}.oauth_client_secret`
- `auth_oauth2.signing_keys.{id/index}`

Validazioni aggiuntive per l'autenticazione 2.0 OAuth

Amazon MQ applica anche le seguenti convalide aggiuntive per l'autenticazione OAuth 2.0:

- Tutto URLs deve iniziare con. `https://`
- Algoritmi di firma supportati: `Ed25519` `Ed25519phEd448,Ed448ph,EdDSA,ES256K,,ES256,ES384,ES512,HS256,HS384,HS512,PS256,PS384,` `RS256RS384`, `eRS512`.

Autenticazione e autorizzazione IAM per Amazon MQ for RabbitMQ

Amazon MQ for RabbitMQ supporta diversi metodi di autenticazione e autorizzazione. Per informazioni su tutti i metodi supportati, consulta Autenticazione e autorizzazione per i broker Amazon MQ for RabbitMQ.

L'autenticazione e l'autorizzazione IAM consentono agli utenti del broker di autenticarsi utilizzando le credenziali AWS IAM tramite la federazione in uscita IAM. In questo metodo, le credenziali IAM vengono utilizzate per ottenere token JWT da AWS Security Token Service (STS). Questi token JWT fungono da token OAuth 2.0 per l'autenticazione, sfruttando il supporto OAuth 2.0 esistente in Amazon MQ for RabbitMQ, che AWS funge da provider di identità 2.0. OAuth AWS IAM gestisce l'autenticazione degli utenti, mentre le autorizzazioni delle risorse per host virtuali, scambi, code e argomenti sono gestite tramite policy IAM e alias di ambito configurati in RabbitMQ.

⚠ Considerazioni importanti

- L'autenticazione IAM è supportata nelle versioni 3.13, 4.2 e successive di RabbitMQ. Non è supportato sui broker Amazon MQ for ActiveMQ.
- L'autenticazione IAM richiede che la federazione IAM in uscita sia configurata e disponibile nel tuo account. AWS
- Questo metodo si basa sull'infrastruttura OAuth 2.0 esistente in Amazon MQ for RabbitMQ e AWS funge da provider di identità OAuth 2.0.
- Amazon MQ crea automaticamente un utente di sistema denominato `monitoring-AWS-OWNED-DO-NOT-DELETE` con autorizzazioni di solo monitoraggio. Questo utente utilizza il sistema di autenticazione interno di RabbitMQ anche su broker abilitati all'IAM ed è limitato al solo accesso all'interfaccia di loopback.

In questa pagina

- Come funziona l'autenticazione IAM
- Limitazioni

Come funziona l'autenticazione IAM

L'autenticazione IAM per Amazon MQ for RabbitMQ utilizza la federazione IAM in uscita per consentire alle credenziali AWS IAM di autenticarsi con i broker RabbitMQ. Le credenziali IAM vengono utilizzate per ottenere i token JWT dal AWS Security Token Service (STS) e questi token JWT fungono da token 2.0 per l'autenticazione con il broker RabbitMQ. OAuth

Limitazioni

L'autenticazione IAM per Amazon MQ for RabbitMQ presenta le seguenti limitazioni:

- Configurazione dell'attestazione dell'ambito: non è possibile utilizzare direttamente un'attestazione di ambito perché il token JWT di STS è annidato. La chiave è `sts.amazonaws.com` che richiede l'utilizzo di alias di ambito nella configurazione di RabbitMQ per mappare i ruoli IAM alle autorizzazioni di RabbitMQ. Questa limitazione impedisce inoltre di utilizzare completamente le policy IAM per l'autorizzazione, richiedendo invece la configurazione di RabbitMQ per l'autorizzazione.

Per informazioni su come configurare l'autenticazione e l'autorizzazione IAM per i broker Amazon MQ for RabbitMQ, consulta. Utilizzo dell'autenticazione e dell'autorizzazione IAM

Autenticazione e autorizzazione HTTP per Amazon MQ for RabbitMQ

Amazon MQ for RabbitMQ supporta l'autenticazione e l'autorizzazione degli utenti del broker utilizzando un server HTTP esterno. Per altri metodi supportati, consulta Autenticazione e autorizzazione per i broker Amazon MQ for RabbitMQ.

> **Note**
>
> Il plug-in di autenticazione HTTP è disponibile solo per Amazon MQ for RabbitMQ versione 4 e successive.

> **Considerazioni importanti**
>
> - Il server HTTP deve essere accessibile tramite la rete Internet pubblica. Amazon MQ for RabbitMQ può essere configurato per l'autenticazione sul server HTTP utilizzando il protocollo TLS reciproco.
> - Amazon MQ for RabbitMQ impone l'uso di AWS ARNs per le impostazioni che richiedono l'accesso al file system locale. Vedi il supporto ARN nella configurazione di RabbitMQ per maggiori dettagli.
> - È necessario includere l'autorizzazione IAM per abilitare l'autenticazione HTTP sui broker esistenti. `mq:UpdateBrokerAccessConfiguration`
> - Amazon MQ crea automaticamente un utente di sistema denominato `monitoring-AWS-OWNED-DO-NOT-DELETE` con autorizzazioni di solo monitoraggio. Questo utente utilizza il sistema di autenticazione interno di RabbitMQ anche su broker abilitati per HTTP ed è limitato al solo accesso all'interfaccia di loopback. Amazon MQ impedisce l'eliminazione di questo utente aggiungendo il tag utente protetto.

Per informazioni su come configurare l'autenticazione HTTP per i broker Amazon MQ for RabbitMQ, consulta. Utilizzo dell'autenticazione e dell'autorizzazione HTTP

In questa pagina

- Configurazioni HTTP supportate

- Convalide aggiuntive per le configurazioni HTTP in Amazon MQ

Configurazioni HTTP supportate

Amazon MQ for RabbitMQ supporta tutte le variabili configurabili nel plug-in di autenticazione HTTP RabbitMQ, con le seguenti eccezioni che richiedono. AWS ARNs Per i dettagli sul supporto ARN, vedere Supporto ARN nella configurazione di RabbitMQ.

Configurazioni che richiedono ARNs

`auth_http.ssl_options.cacertfile`

Usa invece `aws.arns.auth_http.ssl_options.cacertfile`

`auth_http.ssl_options.certfile`

Usa invece `aws.arns.auth_http.ssl_options.certfile`

`auth_http.ssl_options.keyfile`

Usa invece `aws.arns.auth_http.ssl_options.keyfile`

Opzioni SSL non supportate

Inoltre, non sono supportate le seguenti opzioni di configurazione SSL:

Visualizza l'elenco completo

- `auth_http.ssl_options.cert`
- `auth_http.ssl_options.client_renegotiation`
- `auth_http.ssl_options.dh`
- `auth_http.ssl_options.dhfile`
- `auth_http.ssl_options.honor_cipher_order`
- `auth_http.ssl_options.honor_ecc_order`
- `auth_http.ssl_options.key.RSAPrivateKey`
- `auth_http.ssl_options.key.DSAPrivateKey`
- `auth_http.ssl_options.key.PrivateKeyInfo`

- `auth_http.ssl_options.log_alert`
- `auth_http.ssl_options.password`
- `auth_http.ssl_options.psk_identity`
- `auth_http.ssl_options.reuse_sessions`
- `auth_http.ssl_options.secure_renegotiate`
- `auth_http.ssl_options.versions.$version`
- `auth_http.ssl_options.sni`
- `auth_http.ssl_options.crl_check`

Convalide aggiuntive per le configurazioni HTTP in Amazon MQ

Amazon MQ applica anche le seguenti convalide aggiuntive per l'autenticazione e l'autorizzazione HTTP:

- `auth_http.http_method`deve essere una delle due o `get` `post`
- Le seguenti configurazioni di percorso devono utilizzare HTTPS: URLs
  - `auth_http.user_path`
  - `auth_http.vhost_path`
  - `auth_http.resource_path`
  - `auth_http.topic_path`
- Se un'impostazione richiede l'uso di un AWS ARN, `aws.arns.assume_role_arn` deve essere fornita.

Autenticazione del certificato SSL per Amazon MQ for RabbitMQ

Amazon MQ for RabbitMQ supporta l'autenticazione degli utenti del broker tramite certificati client X.509. Per altri metodi supportati, consulta Autenticazione e autorizzazione per i broker Amazon MQ for RabbitMQ.

> **Note**
>
> Il plug-in di autenticazione del certificato SSL è disponibile solo per Amazon MQ for RabbitMQ versione 4 e successive.

⚠ Considerazioni importanti

- I certificati client devono essere firmati da un'autorità di certificazione (CA) affidabile. Amazon MQ for RabbitMQ convalida la catena di certificati durante l'autenticazione.
- Amazon MQ for RabbitMQ ne impone l'uso AWS ARNs per le impostazioni relative ai certificati, come i certificati CA, e per le impostazioni che richiedono l'accesso al file system locale. Vedi il supporto ARN nella configurazione di RabbitMQ per maggiori dettagli.
- Amazon MQ crea automaticamente un utente di sistema denominato `monitoring-AWS-OWNED-DO-NOT-DELETE` con autorizzazioni di solo monitoraggio. Questo utente utilizza il sistema di autenticazione interno di RabbitMQ anche su broker abilitati ai certificati SSL ed è limitato al solo accesso all'interfaccia di loopback. Amazon MQ impedisce l'eliminazione di questo utente aggiungendo il tag utente protetto.

Per informazioni su come configurare l'autenticazione del certificato SSL per i broker Amazon MQ for RabbitMQ, consulta. Utilizzo dell'autenticazione tramite certificato SSL

In questa pagina

- Configurazioni SSL supportate
- Convalide aggiuntive per le configurazioni SSL in Amazon MQ

Configurazioni SSL supportate

Amazon MQ for RabbitMQ supporta la SSL/TLS configurazione per le connessioni client. Per i dettagli sul supporto ARN, vedere Supporto ARN nella configurazione di RabbitMQ.

Configurazioni che richiedono ARNs

`ssl_options.cacertfile`

Usa invece `aws.arns.ssl_options.cacertfile`

Configurazioni di accesso con certificato SSL

Le seguenti configurazioni controllano il modo in cui i nomi utente vengono estratti dai certificati client:

`ssl_cert_login_from`

Speciifica quale campo del certificato utilizzare per l'estrazione del nome utente. Valori supportati:

- `distinguished_name`- Usa il nome distinto completo
- `common_name`- Usa il campo Common Name (CN)
- `subject_alternative_name`oppure `subject_alt_name` - Usa il nome alternativo del soggetto

`ssl_cert_login_san_type`

Quando si utilizza Subject Alternative Name, specifica il tipo di SAN. Valori supportati:`dns`,,`ip`,`email`, `uri` `other_name`

`ssl_cert_login_san_index`

Quando si utilizza Subject Alternative Name, specifica l'indice della voce SAN da utilizzare (a base zero). Deve essere un numero intero non negativo.

Opzioni SSL per le connessioni client

Le seguenti opzioni SSL si applicano alle connessioni client:

`ssl_options.verify`

Modalità di verifica tra pari. Valori supportati:`verify_none`, `verify_peer`

`ssl_options.fail_if_no_peer_cert`

Se rifiutare le connessioni se il client non fornisce un certificato. Valore booleano.

`ssl_options.depth`

Profondità massima della catena di certificati per la verifica.

`ssl_options.hostname_verification`

Modalità di verifica del nome host. Valori supportati:`wildcard`, `none`

Opzioni SSL non supportate

Le seguenti opzioni di configurazione SSL non sono supportate:

Visualizza l'elenco completo

- `ssl_options.cert`
- `ssl_options.client_renegotiation`
- `ssl_options.dh`
- `ssl_options.dhfile`
- `ssl_options.honor_cipher_order`
- `ssl_options.honor_ecc_order`
- `ssl_options.key.RSAPrivateKey`
- `ssl_options.key.DSAPrivateKey`
- `ssl_options.key.PrivateKeyInfo`
- `ssl_options.log_alert`
- `ssl_options.password`
- `ssl_options.psk_identity`
- `ssl_options.reuse_sessions`
- `ssl_options.secure_renegotiate`
- `ssl_options.versions.$version`
- `ssl_options.sni`
- `ssl_options.crl_check`

Convalide aggiuntive per le configurazioni SSL in Amazon MQ

Amazon MQ applica anche le seguenti convalide aggiuntive per l'autenticazione dei certificati SSL:

- Se un'impostazione richiede l'uso di un AWS ARN, `aws.arns.assume_role_arn` deve essere fornita.

Autenticazione e autorizzazione LDAP per Amazon MQ for RabbitMQ

Amazon MQ for RabbitMQ supporta l'autenticazione e l'autorizzazione degli utenti del broker utilizzando un server LDAP esterno. Per altri metodi supportati, consulta Autenticazione e autorizzazione per i broker Amazon MQ for RabbitMQ.

⚠ Considerazioni importanti

- Il server LDAP deve essere accessibile tramite la rete Internet pubblica. Amazon MQ for RabbitMQ può essere configurato per l'autenticazione sul server LDAP tramite TLS reciproco.
- Amazon MQ for RabbitMQ ne impone l'uso AWS ARNs per impostazioni LDAP sensibili come le password e per le impostazioni che richiedono l'accesso al file system locale. Vedi il supporto ARN nella configurazione di RabbitMQ per maggiori dettagli.
- È necessario includere l'autorizzazione IAM per abilitare LDAP sui broker esistenti. `mq:UpdateBrokerAccessConfiguration`
- Amazon MQ crea automaticamente un utente di sistema denominato `monitoring-AWS-OWNED-DO-NOT-DELETE` con autorizzazioni di solo monitoraggio. Questo utente utilizza il sistema di autenticazione interno di RabbitMQ anche su broker abilitati per LDAP ed è limitato al solo accesso all'interfaccia di loopback. Amazon MQ impedisce l'eliminazione di questo utente aggiungendo il tag utente protetto.

Per informazioni su come configurare LDAP per i broker Amazon MQ for RabbitMQ, consulta. Utilizzo dell'autenticazione e dell'autorizzazione LDAP

In questa pagina

- Configurazioni LDAP supportate
- Convalide aggiuntive per le configurazioni LDAP in Amazon MQ

Configurazioni LDAP supportate

Amazon MQ for RabbitMQ supporta tutte le variabili configurabili nel plug-in LDAP RabbitMQ, con le seguenti eccezioni che richiedono. AWS ARNs Per i dettagli sul supporto ARN, vedere Supporto ARN nella configurazione di RabbitMQ.

Configurazioni che richiedono ARNs

`auth_ldap.dn_lookup_bind.password`

Usa invece `aws.arns.auth_ldap.dn_lookup_bind.password`

`auth_ldap.other_bind.password`

Usa invece `aws.arns.auth_ldap.other_bind.password`

`auth_ldap.ssl_options.cacertfile`

Usa invece `aws.arns.auth_ldap.ssl_options.cacertfile`

`auth_ldap.ssl_options.certfile`

Usa invece `aws.arns.auth_ldap.ssl_options.certfile`

`auth_ldap.ssl_options.keyfile`

Usa invece `aws.arns.auth_ldap.ssl_options.keyfile`

Opzioni SSL non supportate

Inoltre, non sono supportate le seguenti opzioni di configurazione SSL:

Visualizza l'elenco completo

- `auth_ldap.ssl_options.cert`
- `auth_ldap.ssl_options.client_renegotiation`
- `auth_ldap.ssl_options.dh`
- `auth_ldap.ssl_options.dhfile`
- `auth_ldap.ssl_options.honor_cipher_order`
- `auth_ldap.ssl_options.honor_ecc_order`
- `auth_ldap.ssl_options.key.RSAPrivateKey`
- `auth_ldap.ssl_options.key.DSAPrivateKey`
- `auth_ldap.ssl_options.key.PrivateKeyInfo`
- `auth_ldap.ssl_options.log_alert`
- `auth_ldap.ssl_options.password`
- `auth_ldap.ssl_options.psk_identity`
- `auth_ldap.ssl_options.reuse_sessions`
- `auth_ldap.ssl_options.secure_renegotiate`

- `auth_ldap.ssl_options.versions.$version`
- `auth_ldap.ssl_options.sni`

Convalide aggiuntive per le configurazioni LDAP in Amazon MQ

Amazon MQ applica anche le seguenti convalide aggiuntive per l'autenticazione e l'autorizzazione LDAP:

- `auth_ldap.log`non può essere impostato su `network_unsafe`
- Il server LDAP deve utilizzare LDAPS. `auth_ldap.use_ssl`O `auth_ldap.use_starttls` deve essere abilitato in modo esplicito
- Se un'impostazione richiede l'uso di un AWS ARN, `aws.arns.assume_role_arn` deve essere fornita.
- `auth_ldap.servers`deve essere un indirizzo IP valido o un FQDN valido
- Le seguenti chiavi devono essere un nome distinto LDAP valido:
  - `auth_ldap.dn_lookup_base`
  - `auth_ldap.dn_lookup_bind.user_dn`
  - `auth_ldap.other_bind.user_dn`
  - `auth_ldap.group_lookup_base`

Plugin

Amazon MQ for RabbitMQ supporta anche i seguenti plugin.

- Plugin di gestione RabbitMQ
- Plugin Shovel
- Plugin federativo
- Plugin per lo scambio di hash coerenti
- OAuth Plugin 2
- Plugin LDAP
- Plugin HTTP
- Plugin per certificati SSL

- plugin aws
- Plugin JMS Topic Exchange

Plugin di gestione RabbitMQ

Amazon MQ for RabbitMQ supporta il plug-in di gestione RabbitMQ, che fornisce un'API di gestione basata su HTTP insieme a un'interfaccia utente basata su browser per la console Web RabbitMQ. È possibile utilizzare la console Web e l'API di gestione per creare e gestire utenti e policy del broker.

Plugin Shovel

Amazon MQ for RabbitMQ supporta il plug-in shovel RabbitMQ, che consente di spostare i messaggi dalle code e dagli scambi su un broker all'altro. È possibile utilizzare gli shovel per collegare i broker con accoppiamento debole e implementare i messaggi lontano dai nodi con carichi di messaggi più elevati.

> **Important**
>
> Non è possibile configurare lo shovel tra code o scambi se la destinazione dello shovel è un broker privato.
> Amazon MQ non supporta l'utilizzo di shovel statici.

Sono supportate solo le pale dinamiche. Le pale dinamiche sono configurate utilizzando parametri di runtime e possono essere avviate e interrotte in qualsiasi momento a livello di codice tramite una connessione client. Ad esempio, utilizzando l'API di gestione RabbitMQ, è possibile creare una richiesta PUT al seguente endpoint API per configurare uno shovel dinamico. Nell'esempio, {vhost} può essere sostituito dal nome del vhost del broker e {name} dal nome della nuova pala dinamica.

```
/api/parameters/shovel/{vhost}/{name}
```

Nel corpo della richiesta, è necessario specificare una coda o uno scambio, ma non entrambi. L'esempio seguente configura una pala dinamica tra una coda locale specificata in src-queue e una coda remota definita in dest-queue. Allo stesso modo, è possibile utilizzare i parametri src-exchange e dest-exchange per configurare una pala tra due exchange.

```
{
```

```
"value": {
"src-protocol": "amqp091",
"src-uri": "amqp://localhost",
"src-queue": "source-queue-name",
"dest-protocol": "amqp091",
"dest-uri": "amqps://b-c8352341-ec91-4a78-ad9c-a43f23d325bb.mq.us-
west2.amazonaws.com:5671",
"dest-queue": "destination-queue-name"
}
}
```

Plugin federativo

Amazon MQ supporta scambi e code federati utilizzando il plug-in di federazione RabbitMQ. Con il plugin federativo, è possibile replicare il flusso di messaggi tra code, scambi e consumatori su broker separati. Le code e gli exchange federati utilizzano i point-to-point link per connettersi ai colleghi di altri broker. Mentre gli scambi federati, per impostazione predefinita, instradano i messaggi una sola volta, le code federate possono spostare i messaggi un numero qualsiasi di volte in base alle esigenze dei consumatori.

È possibile utilizzare il plugin federativo per consentire a un broker downstream di utilizzare un messaggio da uno scambio o una coda su un upstream. È possibile abilitare la federazione sui broker downsteam utilizzando la console Web RabbitMQ o l'API di gestione.

> **Important**
>
> Non è possibile configurare la federazione se la coda a monte o lo scambio si trova in un broker privato. È possibile configurare la federazione solo tra code o scambi nei broker pubblici o tra una coda o uno scambio a monte in un broker pubblico e una coda o uno scambio a valle in un broker privato.

Ad esempio, è possibile utilizzare l'API di gestione per configurare la federazione eseguendo le operazioni seguenti.

- Configurare uno o più upstream che definiscono le connessioni di federazione ad altri nodi. È possibile definire connessioni federative utilizzando la console Web RabbitMQ o l'API di gestione. Utilizzando l'API di gestione, puoi creare una richiesta POST a/api/parameters/federation-upstream/%2f/myupstreamcon il seguente corpo della richiesta.

```
{"value":{"uri":"amqp://server-name","expires":3600000}}
```

- Configurare una policy per consentire la federazione delle code o degli scambi. È possibile configurare le policy utilizzando la console Web RabbitMQ o l'API di gestione. Utilizzando l'API di gestione, puoi creare una richiesta POST a/api/policies/%2f/federate-me con il seguente corpo della richiesta.

```
{"pattern":"^amq\.", "definition":{"federation-upstream-set":"all"}, "apply-
to":"exchanges"}
```

> **Note**
>
> Il corpo della richiesta presuppone che gli scambi sul server abbiano nomi che iniziano con amq. L'uso dell'espressione regolare ^amq\. assicurerà che la federazione sia abilitata per tutti gli scambi i cui nomi iniziano con «amq». Gli scambi sul server RabbitMQ possono essere nominati in modo diverso.

Plugin scambio di hash coerente

Amazon MQ for RabbitMQ supporta il plug-in RabbitMQ Consistent Hash Exchange Type. Gli scambi di hash coerenti instradano i messaggi alle code in base a un valore hash calcolato dalla chiave di routing di un messaggio. Data una chiave di routing ragionevolmente uniforme, gli scambi di hash coerenti possono distribuire i messaggi tra le code in modo ragionevole uniforme.

Per le code associate a uno scambio Consistent Hash, la chiave di associazione è una number-as-a-string che determina il peso di associazione di ciascuna coda. Le code con un peso di associazione maggiore riceveranno una distribuzione proporzionalmente superiore dei messaggi dallo scambio di hash coerente a cui sono associati. In una topologia di scambio di hash coerente, i mittenti possono semplicemente pubblicare messaggi nello scambio, ma i consumatori devono essere configurati in modo esplicito per utilizzare i messaggi provenienti da code specifiche.

OAuth Plugin 2.0

Amazon MQ for RabbitMQ supporta il plug-in backend di autenticazione a OAuth 2. Questo plugin è abilitato in modo condizionale in base alla configurazione del broker. Se abilitato, questo plugin fornisce l'autenticazione e l'autorizzazione OAuth 2.0 con integrazione con provider di identità

OAuth 2.0 esterni per la gestione centralizzata degli utenti e il controllo degli accessi. Per ulteriori informazioni sull'autenticazione OAuth 2.0, vedereOAuth autenticazione e autorizzazione 2.0.

Plugin LDAP

Amazon MQ for RabbitMQ supporta il plug-in backend di autenticazione LDAP. Questo plugin è abilitato in modo condizionale in base alla configurazione del broker. Se abilitato, questo plugin fornisce l'autenticazione e l'autorizzazione LDAP con integrazione a servizi di directory LDAP esterni per l'autenticazione e l'autorizzazione centralizzate degli utenti. Per ulteriori informazioni sull'autenticazione LDAP, vedere. Autenticazione e autorizzazione LDAP

Plugin HTTP

Amazon MQ for RabbitMQ supporta il plug-in backend di autenticazione HTTP. Questo plug-in è abilitato in modo condizionale in base alla configurazione del broker. Se abilitato, questo plug-in fornisce l'autenticazione e l'autorizzazione HTTP con integrazione su server HTTP esterni per l'autenticazione e l'autorizzazione centralizzate degli utenti. Per ulteriori informazioni sull'autenticazione HTTP, vedereAutenticazione e autorizzazione HTTP.

> **Note**
>
> Il plug-in di autenticazione HTTP è disponibile solo per Amazon MQ for RabbitMQ versione 4 e successive.

Plugin per certificati SSL

Amazon MQ supporta il protocollo TLS reciproco (mTLS) per i broker RabbitMQ. Il plug-in di autenticazione SSL utilizza i certificati client delle connessioni MTLS per autenticare gli utenti. Questo plugin è abilitato in modo condizionale in base alla configurazione del broker. Se abilitato, fornisce l'autenticazione basata su certificati utilizzando certificati client X.509 per un'autenticazione avanzata senza trasmettere credenziali sulla rete. Per ulteriori informazioni sull'autenticazione dei certificati SSL, consulta. Autenticazione con certificato SSL

> **Note**
>
> Il plug-in di autenticazione del certificato SSL è disponibile solo per Amazon MQ for RabbitMQ versione 4 e successive.

plugin aws

Il plug-in aws è abilitato in modo condizionale da Amazon MQ per RabbitMQ in base alla configurazione del broker. Questo plug-in della community, sviluppato e gestito da Amazon MQ, fornisce il recupero sicuro di credenziali e certificati dai AWS servizi utilizzando AWS ARNs le impostazioni di configurazione di RabbitMQ. Per ulteriori informazioni sul supporto ARN, vedere. ARN support in RabbitMQ configuration

Plugin JMS Topic Exchange

Il plugin JMS Topic Exchange è sempre abilitato da Amazon MQ per RabbitMQ. Funziona con il client RabbitMQ JMS per consentire alle applicazioni JMS nuove ed esistenti di connettersi ad Amazon MQ for RabbitMQ.

> **Note**
>
> Il plug-in JMS Topic Exchange è disponibile solo per Amazon MQ for RabbitMQ versione 4 e successive. È abilitato per impostazione predefinita, ma si attiva solo quando il client RabbitMQ JMS viene utilizzato per eseguire carichi di lavoro JMS.

Protocolli supportati

Puoi accedere ai tuoi broker RabbitMQ utilizzando qualsiasi linguaggio di programmazione supportato da RabbitMQ e abilitando TLS per una delle seguenti specifiche di protocollo:

- AMQP (0-9-1)
- AMQP 1.0
- JMS 1.1
- JMS 2.0
- JMS 3.1

Supporto Amazon MQ per RabbitMQ JMS

Ora puoi eseguire carichi di lavoro JMS 1.1, 2.0 e 3.1 su Amazon MQ per RabbitMQ 4 con il client RabbitMQ JMS.

Client RabbitMQ JMS

Il client RabbitMQ JMS è una libreria client JMS open source di cui hai bisogno per connettere la tua applicazione JMS ai broker Amazon MQ RabbitMQ. Per ulteriori informazioni, visita il repository ufficiale. GitHub

JMS 1.1, 2.0 e 3.1 supportati APIs

A partire da Amazon MQ per RabbitMQ 4 in poi, il plug-in `jms-topic-exchange` è sempre abilitato. Quindi, puoi utilizzare Amazon MQ per RabbitMQ 4 e il client RabbitMQ JMS per il tuo carico di lavoro JMS. Tutti i JMS APIs definiti in JMS 1.1 sono supportati tranne:

- Le sessioni del server non APIs sono supportate.
- APIs Le transazioni XA non sono supportate.
- Il selettore JMS per la destinazione della coda JMS non è supportato.
- L'attributo di `NoLocal` sottoscrizione JMS non è supportato.

Sono supportati tutti i nuovi APIs componenti aggiunti in JMS 2.0 e JMS 3.1, ad eccezione di:

- `JMSProducer.setDeliveryDelay`L'API non è supportata.

Per ulteriori informazioni sulla connessione dell'applicazione JMS al broker Amazon MQ for RabbitMQ, consulta il tutorial su Collegare l'applicazione JMS al broker Amazon MQ for RabbitMQ

Autenticazione e autorizzazione

Sono supportati tutti i meccanismi di autenticazione e autorizzazione elencati in questa sezione. Le credenziali utilizzate per la connessione al broker tramite il client JMS sono le stesse utilizzate per la connessione al broker RabbitMQ utilizzando un client Java AMQP.

Interoperabilità con le code AMQP su RabbitMQ

È possibile utilizzare il client RabbitMQ JMS per inviare messaggi JMS a uno scambio AMQP e utilizzare i messaggi da una coda AMQP (questa funzionalità non supporta gli argomenti JMS). Ciò consente di interoperare o migrare determinati carichi di lavoro JMS verso carichi di lavoro AMQP. Per ulteriori informazioni, consulta la documentazione ufficiale del cliente.

Applicazione delle politiche ad Amazon MQ for RabbitMQ

Puoi applicare politiche e limiti personalizzati con i valori predefiniti consigliati di Amazon MQ. Se sono stati eliminati i limiti e le policy predefiniti consigliati e si desidera ricrearli oppure sono stati creati altri vhost e si desidera applicare policy e limiti predefiniti ai nuovi vhost, è possibile eseguire la procedura seguente.

Important

Nelle versioni 3.13 e precedenti del motore Amazon MQ for RabbitMQ, l'attuale politica dell'operatore predefinita è:

```
vhost name pattern apply-to definition priority/
 default_operator_policy_AWS_managed .* classic_queues {"ha-mode":"all","ha-
sync-mode":"automatic","queue-version":2} 0
```

Nelle versioni 4.0 e successive, la politica dell'operatore predefinita è stata modificata in:

```
vhost name pattern apply-to definition priority/
 default_operator_policy_AWS_managed .* classic_queues {"queue-version":2} 0
```

Questa modifica è necessaria perché il classico mirroring della coda e le impostazioni dei criteri HA non sono supportati in RabbitMQ 4.
Non è possibile creare una policy che si applichi sia alle code con mirroring classiche che alle code quorum. Se desideri che la tua politica si applichi solo alle code quorum, devi impostarla su. `--apply-to quorum_queues` Se si utilizzano le classiche code speculari e le code quorum, è necessario creare una politica separata con `--apply-to:classic_queues` oltre a una politica per le code quorum.

Important

Per eseguire la procedura seguente, è necessario un utente del broker Amazon MQ per RabbitMQ con autorizzazioni di amministratore. È possibile utilizzare l'utente amministratore creato al momento della creazione del broker o un altro utente che potrebbe essere stato creato successivamente. Nella tabella seguente vengono forniti i tag dell'utente amministratore necessari e le autorizzazioni come modelli di espressione regolare (regexp).

Tag	Lettura di regexp	Configurazione di regexp	Scrittura di regexp
`administrator`	`.*`	`.*`	`.*`

Per ulteriori informazioni sulla creazione di utenti RabbitMQ e sulla gestione di tag e autorizzazioni degli utenti, consultare Amazon MQ per gli utenti del broker RabbitMQ.

Applicazione di policy e limiti predefiniti di host virtuali utilizzando la console Web RabbitMQ

1. Accedere alla console Amazon MQ.
2. Nel pannello di navigazione a sinistra selezionare Brokers (Broker).
3. Nell'elenco dei broker, scegliere il nome del broker a cui si desidera applicare la nuova policy.
4. Nella pagina dei dettagli del broker, alla sezione Connections (Connessioni), scegliere l'URL della Console web RabbitMQURL. La console Web RabbitMQ si apre in una nuova scheda o finestra del browser.
5. Accedere alla console Web RabbitMQ con il nome utente e la password dell'amministratore del broker.
6. Nella console Web RabbitMQ, nella parte superiore della pagina, selezionare Admin (Amministratore).
7. Alla pagina Admin (Amministratore), nel pannello di navigazione destro, selezionare Policies (Policy).
8. Alla pagina Policies (Policy), comparirà un elenco delle policy dell'utente connesso del broker. Sotto User policies (Policy utente), espandere Add / update a policy (Aggiungere/aggiornare una policy).
9. Per creare una nuova policy del broker in Add / update a policy (Aggiungere/aggiornare una policy), effettua le seguenti operazioni:

    a. Per Virtual host (Host virtuale), selezionare il nome del vhost a cui si desidera collegare le policy dall'elenco a discesa. Per scegliere il vhost predefinito, scegliere /.

> **Note**
>
> Se non sono stati creati vhost aggiuntivi, l'opzione Virtual host (Host virtuale) non viene visualizzata nella console RabbitMQ e le policy vengono applicate solo al vhost predefinito.

b. Per Name (Nome), immettere un nome per la policy, ad esempio **`policy-defaults`**.

c. Per Pattern (Modello), inserisci il modello regexp **`.*`** in modo che la policy corrisponda a tutte le code nel broker.

d. Per Apply to (Applica a), scegliere Exchanges and queues (Scambi e code) dall'elenco a discesa.

e. Per Priority (Priorità), immetti un numero intero maggiore di tutte le altre policy applicate al vhost. È possibile applicare esattamente un set di definizioni di policy alle code e agli scambi di RabbitMQ in qualsiasi momento. RabbitMQ sceglie la policy corrispondente al valore di priorità più alto. Per ulteriori informazioni sulle priorità delle policy e su come combinare le policy, consultare Policy nella documentazione del server RabbitMQ.

f. Per Definition (Definizione), aggiungere le seguenti coppie chiave-valore:

- **`queue-mode=lazy`**. Scegliere String (Stringa) dall'elenco a discesa.
- **`overflow=reject-publish`**. Scegliere String (Stringa) dall'elenco a discesa.

> **Note**
>
> Non si applica ai broker a istanza singola.

- **`max-length=`**. ***`number-of-messages`*** Sostituisci *`number-of-messages`* con il valore consigliato di Amazon MQ in base alla dimensione dell'istanza e alla modalità di distribuzione del broker, ad esempio **`8000000`** per un `mq.m7g.large` cluster. Scegliere Number (Numero) dall'elenco a discesa.

> **Note**
>
> Non si applica ai broker a istanza singola.

g. Scegliere Add / update policy (Aggiungi/aggiorna policy).

10. Verificare che la nuova policy sia visualizzata nell'elenco delle policy dell'utente.

> **Note**
>
> Per i broker di cluster, Amazon MQ applica automaticamente le definizioni delle policy `ha-mode: all` e `ha-sync-mode: automatic`.

11. Nel pannello di navigazione destro, scegliere Limits (Limiti).
12. Alla pagina Limits (Limiti), compare un elenco dei limiti dell'host virtuale attuali del broker. Sotto Virtual host limits (Limiti dell'host virtuale), espandere Set / update a virtual host limit (Imposta/aggiorna un limite dell'host virtuale).
13. Per creare un nuovo limite vhost, in Set / update a virtual host limit (Imposta/aggiorna un limite dell'host virtuale), effettuare le seguenti operazioni:
    a. Per Virtual host (Host virtuale), selezionare il nome del vhost a cui si desidera collegare le policy dall'elenco a discesa. Per scegliere il vhost predefinito, scegliere /.
    b. Per Limit (Limite), scegliere max-connections (connessioni-max) dalle opzioni a discesa.
    c. Per Value (Valore), inserisci il valore consigliato da Amazon MQ in base alle dimensioni dell'istanza e alla modalità di implementazione del broker, ad esempio **`15000`** per un cluster `mq.m5.large`.
    d. Scegliere Set / update limit (Imposta/aggiorna limite).
    e. Ripetere i passaggi precedenti e per Limit (Limite), scegliere max-code (codice-max) dalle opzioni a discesa.
14. Verificare che i nuovi limiti siano visualizzati nell'elenco Virtual host limits (Limiti dell'host virtuale).

Applicazione di policy e limiti predefiniti di host virtuali utilizzando l'API di gestione RabbitMQ

1. Accedere alla console Amazon MQ.
2. Nel pannello di navigazione a sinistra selezionare Brokers (Broker).
3. Nell'elenco dei broker, scegliere il nome del broker a cui si desidera applicare la nuova policy.
4. Alla pagina del broker, nella sezione Connections (Connessioni), prendere nota dell'URL della Console Web RabbitMQ. Questo è l'endpoint del broker che si utilizza in una richiesta HTTP.
5. Aprire un nuovo terminale o una finestra della riga di comando a scelta.

6. Per creare una nuova policy del broker, inserisci il seguente comando `curl`. Questo comando presuppone una coda sul vhost predefinito `/`, che è codificato come `%2F`. Per applicare la policy a un altro vhost, sostituire `%2F` con il nome del vhost.

   > **Note**
   >
   > Sostituisci *`username`* e *`password`* con le credenziali di accesso dell'amministratore. Sostituisci *`number-of-messages`* con il valore consigliato di Amazon MQ in base alla dimensione dell'istanza e alla modalità di distribuzione del broker. *`policy-name`*Sostituiscilo con un nome per la tua polizza. Sostituiscilo *`broker-endpoint`* con l'URL che hai annotato in precedenza.

   ```
   curl -i -u username:password -H "content-type:application/json" -XPUT \
   -d '{"pattern":".*", "priority":1, "definition":{"queue-mode":lazy,
    "overflow":"reject-publish", "max-length":"number-of-messages"}}' \
   broker-endpoint/api/policies/%2F/policy-name
   ```

7. Per confermare che la nuova policy sia stata aggiunta alle policy dell'utente del broker, inserisci il comando `curl` per elencare tutte le policy del broker.

   ```
   curl -i -u username:password broker-endpoint/api/policies
   ```

8. Per creare un nuovo limite dell'host virtuale `max-connections`, inserisci il comando `curl`. Questo comando presuppone una coda sul vhost predefinito `/`, che è codificato come `%2F`. Per applicare la policy a un altro vhost, sostituire `%2F` con il nome del vhost.

   > **Note**
   >
   > Sostituisci *`username`* e *`password`* con le credenziali di accesso dell'amministratore. Sostituisci *`max-connections`* con il valore consigliato di Amazon MQ in base alla dimensione dell'istanza e alla modalità di distribuzione del broker. Sostituire l'endpoint del broker con l'URL annotato in precedenza.

   ```
   curl -i -u username:password -H "content-type:application/json" -XPUT \
   -d '{"value":"number-of-connections"}' \
   ```

```
broker-endpoint/api/vhost-limits/%2F/max-connections
```

9. Per creare un nuovo limite dell'host virtuale `max-queues`, ripetere il passaggio precedente, modificando il comando curl come illustrato di seguito.

```
curl -i -u username:password -H "content-type:application/json" -XPUT \
-d '{"value":"number-of-queues"}' \
broker-endpoint/api/vhost-limits/%2F/max-queues
```

10. Per confermare l'aggiunta dei nuovi limiti ai limiti dell'host virtuale del broker, inserisci il comando `curl` per elencare tutti i limiti degli host virtuali del broker.

```
curl -i -u username:password broker-endpoint/api/vhost-limits
```

Code quorum per RabbitMQ su Amazon MQ

Le code quorum sono un tipo di coda replicato composto da un leader (replica principale) e dai follower (altre repliche). Se il leader non è più disponibile, quorum queues utilizza l'algoritmo di consenso Raft per eleggere un nuovo nodo leader con la maggioranza dei voti e il leader precedente viene retrocesso a nodo follower nello stesso cluster. I follower rimanenti continuano a replicarsi come prima. Poiché ogni nodo si trova in una zona di disponibilità diversa, se un nodo è temporaneamente non disponibile, la consegna dei messaggi continua con la replica del leader appena eletto in un'altra zona di disponibilità.

Le code quorum sono utili per gestire i messaggi avvelenati, che si verificano quando un messaggio fallisce e viene richiesto più volte.

Non è consigliabile utilizzare le code quorum se:

- usa code transitorie
- hanno lunghi arretrati in coda
- dare priorità alla bassa latenza

Per dichiarare una coda di quorum, imposta l'intestazione su. `x-queue-type quorum`

Argomenti

- Migrazione dalle code classiche alle code quorum su Amazon MQ for RabbitMQ
- Configurazioni delle policy per le code quorum per Amazon MQ for RabbitMQ
- Le migliori pratiche per le code quorum per Amazon MQ for RabbitMQ

Migrazione dalle code classiche alle code quorum su Amazon MQ for RabbitMQ

Puoi migrare le tue classiche code con mirroring alle code quorum sui broker Amazon MQ nella versione 3.13 o successiva creando un nuovo host virtuale sullo stesso cluster o effettuando la migrazione sul posto.

Opzione 1: migrazione dalle classiche code con mirroring alle code quorum con un nuovo host virtuale

Puoi migrare le tue classiche code con mirroring alle code quorum sui broker Amazon MQ nella versione 3.13 o successiva creando un nuovo host virtuale sullo stesso cluster.

1. Nel cluster esistente, crea un nuovo host virtuale (vhost) con il tipo di coda predefinito come quorum.
2. Crea il Plugin federativo dal nuovo vhost con l'URI che punta al vecchio vhost usando le classiche code con mirroring.
3. Usando`rabbitmqadmin`, esporta le definizioni dal vecchio vhost in un nuovo file. È necessario apportare modifiche al file di schema in modo che sia compatibile con le code quorum. Per l'elenco completo delle modifiche da apportare al file, consulta Spostamento delle definizioni nella documentazione sulle code quorum di RabbitMQ. Dopo aver applicato le modifiche necessarie al file, reimporta le definizioni nel nuovo vhost.
4. Crea una nuova politica nel nuovo vhost. Per consigli sulle configurazioni delle policy di Amazon MQ per le code quorum, consulta. Configurazioni delle policy per le code quorum per Amazon MQ for RabbitMQ Quindi, avvia la federazione che hai creato in precedenza dal vecchio vhost al nuovo vhost.
5. Indirizza consumatori e produttori al nuovo vhost.
6. Configura il plug-in Shovel per spostare i messaggi rimanenti. Una volta che la coda è vuota, elimina lo Shovel.

Migrazione dalle classiche code speculari alle code con quorum in atto

Puoi migrare le tue classiche code con mirroring alle code quorum sui broker Amazon MQ nella versione 3.13 o successiva effettuando la migrazione sul posto.

1. Fermate i consumatori e i produttori.
2. Crea una nuova coda temporanea per il quorum.
3. Configura il plug-in Shovel per spostare tutti i messaggi dalla vecchia coda speculare classica alla nuova coda temporanea del quorum. Dopo che tutti i messaggi sono stati spostati nella coda temporanea del quorum, elimina Shovel.
4. Elimina la coda speculare classica di origine. Quindi, ricrea una coda quorum con lo stesso nome e gli stessi collegamenti della coda speculare classica di origine.
5. Crea un nuovo Shovel per spostare i messaggi dalla coda del quorum temporanea alla nuova coda del quorum.

Configurazioni delle policy per le code quorum per Amazon MQ for RabbitMQ

Puoi aggiungere configurazioni di policy specifiche alle code di quorum per il tuo broker RabbitMQ su Amazon MQ.

Quando crei una politica per le code di quorum, devi fare quanto segue:

- Rimuove tutti gli attributi della politica che iniziano con`ha`,`ha-mode`,`ha-params`, `ha-sync-mode`ha-sync-batch-size, `ha-promote-on-shutdown` e. `ha-promote-on-failure`
- Remove `queue-mode`.
- Modifica l'overflow quando è impostato su `reject-publish-dlx`

> **Important**
>
> Amazon MQ for RabbitMQ applica tutti o nessuno degli attributi all'interno di una policy. Non è possibile creare una policy che si applichi sia alle classiche code con mirroring che alle code quorum. Se si desidera che la politica si applichi solo alle code quorum, è necessario impostarla su. `--apply-to quorum_queues` Se si utilizzano le classiche code

speculari e le code quorum, è necessario creare una politica separata con `--apply-to`: `classic_queues` oltre a una politica per le code quorum.

Non è necessario modificare `AWS-DEFAULT` le politiche perché adottano automaticamente il nuovo tipo di coda nel parametro «si applica a». Per ulteriori informazioni sulle politiche predefinite per Amazon MQ for RabbitMQ, consulta. Configurazione delle politiche dell'operatore

Le migliori pratiche per le code quorum per Amazon MQ for RabbitMQ

Consigliamo di utilizzare le seguenti best practice per migliorare le prestazioni quando si lavora con le code quorum.

Gestione dei messaggi avvelenati impostando un limite di recapito

I messaggi Poison si verificano quando un messaggio non va a buon fine e viene recapitato più volte. È possibile impostare un limite di recapito dei messaggi utilizzando l'argomento `delivery-limit` policy per eliminare i messaggi che vengono recapitati più volte. Se un messaggio viene riconsegnato più volte di quanto consentito dal limite di recapito, il messaggio viene quindi eliminato da RabbitMQ. Quando imposti un limite di recapito, il messaggio viene messo in coda vicino alla testa della coda.

Priorità dei messaggi per le code con quorum

Le code quorum non hanno la priorità dei messaggi. Se è necessaria la priorità dei messaggi, è necessario creare più code di quorum. Per ulteriori informazioni sull'assegnazione di priorità ai messaggi con più code di quorum, consulta Priorità dei messaggi nella documentazione di RabbitMQ.

Utilizzo del fattore di replica predefinito

Amazon MQ for RabbitMQ utilizza per impostazione predefinita un fattore di replica di tre (3) nodi per i broker di cluster che utilizzano code quorum. Se apporti modifiche `ax-quorum-initial-group-size`, Amazon MQ imposterà nuovamente per impostazione predefinita il fattore di replica 3.

Best practice di Amazon MQ per RabbitMQ

Segui queste linee guida sulla preparazione alla produzione per massimizzare le prestazioni dei broker e ottimizzare l'efficienza della trasmissione dei messaggi quando lavori con Amazon MQ for RabbitMQ broker.

> **Important**
>
> Al momento, Amazon MQ non supporta i flussi o l'uso della registrazione strutturata in JSON, introdotta in RabbitMQ 3.9.

Argomenti

- Le migliori pratiche per la configurazione del broker e la gestione delle connessioni in Amazon MQ for RabbitMQ
- Le migliori pratiche per la durabilità e l'affidabilità dei messaggi in Amazon MQ for RabbitMQ
- Le migliori pratiche per l'ottimizzazione e l'efficienza delle prestazioni in Amazon MQ for RabbitMQ
- Le migliori pratiche per la resilienza e il monitoraggio della rete in Amazon MQ for RabbitMQ

Le migliori pratiche per la configurazione del broker e la gestione delle connessioni in Amazon MQ for RabbitMQ

La configurazione del broker e la gestione delle connessioni sono il primo passo per prevenire problemi relativi alla velocità di trasmissione dei messaggi del broker, all'utilizzo delle risorse e alla capacità di gestire i carichi di lavoro di produzione. Quando crei e configuri un broker Amazon MQ for RabbitMQ, segui le seguenti best practice per selezionare i tipi di istanza appropriati, gestire le connessioni in modo efficiente e configurare la prelettura dei messaggi per massimizzare le prestazioni del broker.

> **Important**
>
> Amazon MQ for RabbitMQ non supporta il nome utente «guest» ed eliminerà l'account ospite predefinito quando crei un nuovo broker. Amazon MQ eliminerà inoltre periodicamente qualsiasi account creato dal cliente chiamato «ospite».

Fase 1: utilizza le distribuzioni in cluster

Per i carichi di lavoro di produzione, consigliamo di utilizzare distribuzioni in cluster anziché broker a istanza singola per garantire un'elevata disponibilità e resilienza dei messaggi. Le implementazioni in cluster eliminano i singoli punti di errore e offrono una migliore tolleranza agli errori.

Le implementazioni dei cluster consistono in tre nodi broker RabbitMQ distribuiti su tre zone di disponibilità, che forniscono il failover automatico e garantiscono la continuità delle operazioni anche se un'intera zona di disponibilità non è disponibile. Amazon MQ replica automaticamente i messaggi su tutti i nodi per garantire la disponibilità durante i guasti o la manutenzione dei nodi.

Le implementazioni in cluster sono essenziali per gli ambienti di produzione e sono supportate dal Service Level Agreement di Amazon MQ.

Per ulteriori informazioni, consulta la sezione Distribuzione di cluster in Amazon MQ for RabbitMQ.

Passaggio 2: scegli il tipo di istanza del broker corretto

La velocità effettiva dei messaggi di un tipo di istanza del broker dipende dal caso d'uso dell'applicazione. `M7g.mediumdeve` essere utilizzato solo per testare le prestazioni dell'applicazione. L'utilizzo di questa istanza più piccola prima di utilizzare istanze più grandi in produzione può migliorare le prestazioni delle applicazioni. Sui tipi di istanze `m7g.large` e versioni successive, puoi utilizzare le distribuzioni in cluster per un'elevata disponibilità e durabilità dei messaggi. I tipi di istanze broker più grandi possono gestire livelli di produzione di client e code, velocità effettiva elevata, messaggi in memoria e messaggi ridondanti.

Per ulteriori informazioni sulla scelta del tipo di istanza corretto, consulta le linee guida sul dimensionamento in Amazon MQ for RabbitMQ.

Passaggio 3: utilizza le code per il quorum

Le code quorum, con distribuzione in cluster, dovrebbero essere la scelta predefinita per i tipi di coda replicati negli ambienti di produzione per i broker RabbitMQ a partire dalla versione 3.13. Le code quorum sono un tipo di coda replicato moderno che offre elevata affidabilità, velocità effettiva elevata e latenza stabile.

Le code quorum utilizzano l'algoritmo di consenso Raft per fornire una migliore tolleranza agli errori. Quando il nodo leader non è più disponibile, le code quorum eleggono automaticamente un nuovo leader a maggioranza, assicurando che la consegna dei messaggi continui con interruzioni minime. Poiché ogni nodo si trova in una zona di disponibilità diversa, il sistema di messaggistica rimane disponibile anche se un'intera zona di disponibilità diventa temporaneamente non disponibile.

Per dichiarare una coda di quorum, imposta l'intestazione `x-queue-type` su quando crei le code. `quorum`

Per ulteriori informazioni sulle code quorum, incluse le strategie di migrazione e le best practice, consulta Quorum queues in Amazon MQ for RabbitMQ.

Passaggio 4: utilizza più canali

Per evitare interruzioni della connessione, utilizza più canali su una singola connessione. Le applicazioni devono evitare un rapporto tra connessione e canale di 1:1. Si consiglia di utilizzare una connessione per ogni processo e quindi un canale per ogni thread. Evita un uso eccessivo dei canali per evitare fughe di dati.

Le migliori pratiche per la durabilità e l'affidabilità dei messaggi in Amazon MQ for RabbitMQ

Prima di passare all'applicazione in produzione, seguite le seguenti procedure consigliate per prevenire la perdita di messaggi e l'eccessivo utilizzo delle risorse.

Fase 1: Usare messaggi persistenti e code durevoli

I messaggi persistenti possono aiutare a proteggere la durabilità dei dati in situazioni in cui un broker si blocca o si riavvia. I messaggi persistenti vengono scritti su disco non appena arrivano. A differenza delle code lente, tuttavia, i messaggi persistenti vengono memorizzati nella cache sia nella memoria che nel disco, a meno che alil broker non occorra ulteriore memoria. Nei casi in cui è necessaria più memoria, i messaggi vengono rimossi dalla memoria dal meccanismo del broker di RabbitMQ che gestisce la memorizzazione dei messaggi su disco, comunemente indicato come livello di persistenza.

Per abilitare la persistenza dei messaggi, è possibile dichiarare le code come `durable` e impostare la modalità di recapito dei messaggi su `persistent`. Nell'esempio seguente viene illustrata la dichiarazione di una coda duratura utilizzando la libreria client Java RabbitMQ. Quando si lavora con AMQP 0-9-1, è possibile contrassegnare i messaggi come persistenti impostando la modalità di consegna «2".

```
boolean durable = true;
channel.queueDeclare("my_queue", durable, false, false, null);
```

Una volta configurata la coda come duratura, è possibile inviare un messaggio persistente alla coda impostando `MessageProperties` su `PERSISTENT_TEXT_PLAIN` come mostrato nell'esempio seguente.

```
import com.rabbitmq.client.MessageProperties;

channel.basicPublish("", "my_queue",
```

```
            MessageProperties.PERSISTENT_TEXT_PLAIN,
            message.getBytes());
```

Passaggio 2: configura le conferme dell'editore e la conferma di consegna al consumatore

Il processo di conferma dell'invio di un messaggio al broker è noto come conferma dell'editore. Le conferme di Publisher consentono all'applicazione di sapere quando i messaggi sono stati archiviati in modo affidabile. Le conferme di Publisher possono anche aiutare a controllare la frequenza dei messaggi archiviati nel broker. Senza le conferme dell'editore, non vi è alcuna conferma che un messaggio sia stato elaborato correttamente e il broker potrebbe eliminare i messaggi che non è in grado di elaborare.

Analogamente, quando un'applicazione client invia al broker la conferma della consegna e del consumo dei messaggi, si parla di conferma della consegna da parte del consumatore. Sia la conferma che il riconoscimento sono essenziali per garantire la sicurezza dei dati quando si lavora con i broker RabbitMQ.

La conferma di consegna dei consumatori è in genere configurata nell'applicazione client. Quando si lavora con AMQP 0-9-1, la conferma può essere abilitata configurando il metodo. `basic.consume` I client AMQP 0-9-1 possono anche configurare le conferme dell'editore inviando il metodo. `confirm.select`

In genere, la conferma di consegna è abilitata in un canale. Ad esempio, quando si lavora con la libreria client Java RabbitMQ, è possibile utilizzare `Channel#basicAck` per impostare una semplice conferma `basic.ack`positiva come mostrato nell'esempio seguente.

```
// this example assumes an existing channel instance

boolean autoAck = false;
channel.basicConsume(queueName, autoAck, "a-consumer-tag",
     new DefaultConsumer(channel) {
         @Override
         public void handleDelivery(String consumerTag,
                                    Envelope envelope,
                                    AMQP.BasicProperties properties,
                                    byte[] body)
             throws IOException
         {
             long deliveryTag = envelope.getDeliveryTag();
```

```
            // positively acknowledge a single delivery, the message will
            // be discarded
            channel.basicAck(deliveryTag, false);
        }
    });
```

> **Note**
>
> I messaggi non riconosciuti devono essere memorizzati nella cache in memoria. È possibile limitare il numero di messaggi che un consumer pre-recupera mediante la configurazione delle impostazioni pre-fetch per un'applicazione client.

È possibile configurare in modo `consumer_timeout` da rilevare quando i consumatori non confermano le consegne. Se il consumatore non invia una conferma entro il valore di timeout, il canale verrà chiuso e riceverai un. `PRECONDITION_FAILED` Per diagnosticare l'errore, utilizza l'UpdateConfigurationAPI per aumentare il valore. `consumer_timeout`

Fase 3: Mantieni le code brevi

Nelle implementazioni cluster, le code con un numero elevato di messaggi possono causare un eccessivo utilizzo delle risorse. Quando un broker viene utilizzato in modo eccessivo, il riavvio di un broker Amazon MQ per RabbitMQ può causare un ulteriore peggioramento delle prestazioni. In caso di riavvio, i broker sovrasfruttati potrebbero non rispondere nello stato `REBOOT_IN_PROGRESS`.

Durante le finestre di manutenzione, Amazon MQ esegue tutti i lavori di manutenzione su un nodo alla volta per garantire che il broker rimanga operativo. Di conseguenza, potrebbe essere necessario sincronizzare le code quando ogni nodo riprende l'operazione. Durante la sincronizzazione, i messaggi che devono essere replicati nei mirror vengono caricati in memoria dal volume Amazon Elastic Block Store (Amazon EBS) corrispondente per essere elaborati in batch. L'elaborazione dei messaggi in batch consente alle code di sincronizzarsi più velocemente.

Se le code vengono mantenute brevi e i messaggi sono piccoli, le code si sincronizzano correttamente e riprendono l'operazione come previsto. Tuttavia, se la quantità di dati in un batch si avvicina al limite di memoria del nodo, il nodo genera un allarme di memoria elevata, sospendendo la sincronizzazione della coda. Puoi confermare l'utilizzo della memoria confrontando le metriche del nodo del `RabbitMemUsed` broker con quelle del nodo `RabbitMqMemLimit` in. CloudWatch La sincronizzazione non può essere completata finché i messaggi non vengono consumati o eliminati o il numero di messaggi nel batch non viene ridotto.

Se la sincronizzazione della coda è sospesa per una distribuzione del cluster, si consiglia di utilizzare o eliminare i messaggi per ridurre il numero di messaggi nelle code. Una volta ridotta la profondità della coda e completata la sincronizzazione della coda, lo stato del broker diventerà `RUNNING`. Per risolvere una sincronizzazione della coda sospesa, è anche possibile applicare un criterio per ridurre la dimensione di batch di sincronizzazione della coda.

Puoi anche definire politiche di eliminazione automatica e TTL per ridurre in modo proattivo l'utilizzo delle risorse e per evitare NACKs che i consumatori entrino in contatto con i consumatori al minimo. La richiesta di messaggi sul broker richiede un uso intensivo della CPU, quindi un numero elevato di messaggi può influire sulle prestazioni del broker. NACKs

Le migliori pratiche per l'ottimizzazione e l'efficienza delle prestazioni in Amazon MQ for RabbitMQ

Puoi ottimizzare le prestazioni del broker Amazon MQ for RabbitMQ massimizzando il throughput, riducendo al minimo la latenza e garantendo un utilizzo efficiente delle risorse. Completa le seguenti best practice per ottimizzare le prestazioni delle tue applicazioni.

Passaggio 1: Mantieni le dimensioni dei messaggi al di sotto di 1 MB

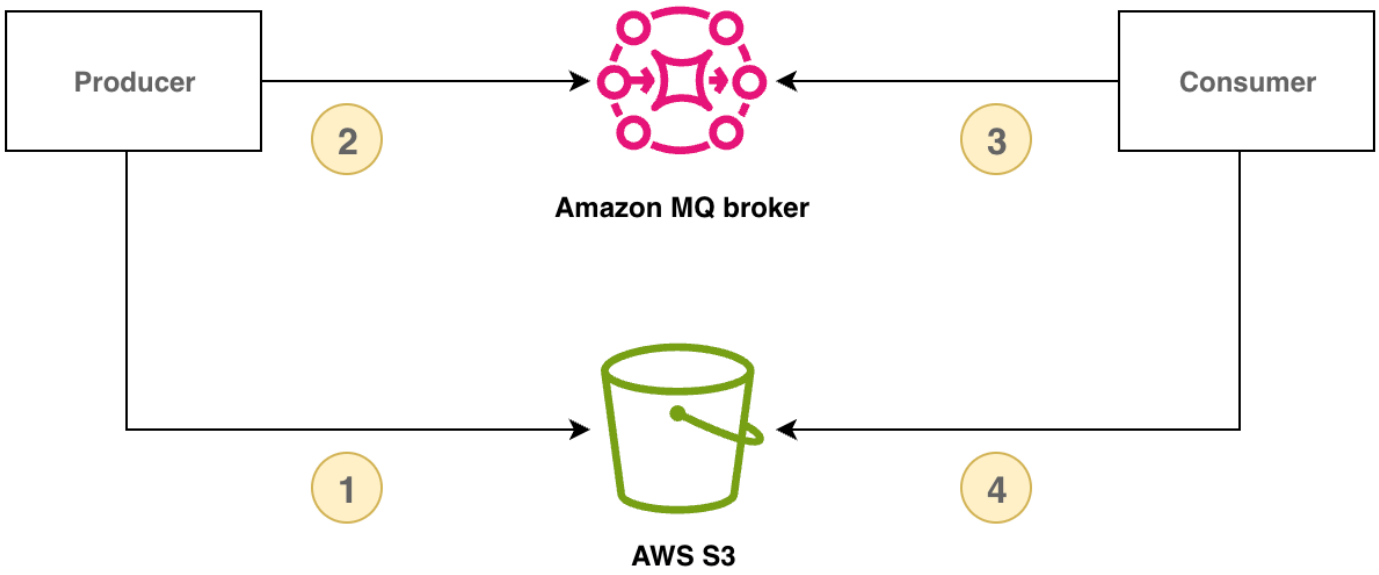
Consigliamo di mantenere i messaggi al di sotto di 1 Megabyte (MB) per prestazioni e affidabilità ottimali.

Per impostazione predefinita, RabbitMQ 3.13 supporta messaggi di dimensioni fino a 128 MB, ma i messaggi di grandi dimensioni possono attivare allarmi di memoria imprevedibili che bloccano la pubblicazione e potenzialmente creano un'elevata pressione della memoria durante la replica dei messaggi tra i nodi. I messaggi sovradimensionati possono inoltre influire sui processi di riavvio e ripristino dei broker, il che aumenta i rischi per la continuità del servizio e può causare un peggioramento delle prestazioni.

Archivia e recupera payload di grandi dimensioni utilizzando lo schema di controllo dei reclami

Per gestire messaggi di grandi dimensioni, puoi implementare il claim check pattern archiviando il payload del messaggio in una memoria esterna e inviando solo l'identificatore di riferimento del payload tramite RabbitMQ. Il consumatore utilizza l'identificatore di riferimento del payload per recuperare ed elaborare il messaggio di grandi dimensioni.

Il diagramma seguente mostra come utilizzare Amazon MQ per RabbitMQ e Amazon S3 per implementare il modello di controllo dei claim.



L'esempio seguente dimostra questo modello utilizzando Amazon MQ, l'AWS SDK per Java 2.x e Amazon S3:

1. Innanzitutto, definisci una classe Message che conterrà l'identificatore di riferimento Amazon S3.

```
class Message {
    // Other data fields of the message...

    public String s3Key;
    public String s3Bucket;
}
```

2. Crea un metodo di pubblicazione che memorizzi il payload in Amazon S3 e invii un messaggio di riferimento tramite RabbitMQ.

```
public void publishPayload() {
    // Store the payload in S3.
    String payload = PAYLOAD;
    String prefix = S3_KEY_PREFIX;
    String s3Key = prefix + "/" + UUID.randomUUID();
    s3Client.putObject(PutObjectRequest.builder()
        .bucket(S3_BUCKET).key(s3Key).build(),
        RequestBody.fromString(payload));

    // Send the reference through RabbitMQ.
    Message message = new Message();
```

```
    message.s3Key = s3Key;
    message.s3Bucket = S3_BUCKET;
    // Assign values to other fields in your message instance.

    publishMessage(message);
}
```

3. Implementa un metodo consumer che recuperi il payload da Amazon S3, elabori il payload ed elimini l'oggetto Amazon S3.

```
public void consumeMessage(Message message) {
    // Retrieve the payload from S3.
    String payload = s3Client.getObjectAsBytes(GetObjectRequest.builder()
        .bucket(message.s3Bucket).key(message.s3Key).build())
        .asUtf8String();

    // Process the complete message.
    processPayload(message, payload);

    // Delete the S3 object.
    s3Client.deleteObject(DeleteObjectRequest.builder()
        .bucket(message.s3Bucket).key(message.s3Key).build());
}
```

Fase 2: Utilizzo e consumatori longevi **`basic.consume`**

L'utilizzo `basic.consume` con un consumatore longevo è più efficiente rispetto all'utilizzo di sondaggi per singoli messaggi. `basic.get` Per ulteriori informazioni, vedere Sondaggi per singoli messaggi.

Fase 3: Configurare il prefetching

È possibile utilizzare il valore di pre-fetching di RabbitMQ per ottimizzare il modo in cui i consumatori consumano i messaggi. RabbitMQ implementa il meccanismo di pre-fetching del canale fornito da AMQP 0-9-1 applicando il conteggio di pre-fetching ai consumatori rispetto ai canali. Il valore di pre-fetching viene utilizzato per specificare quanti messaggi vengono inviati al consumatore in un dato momento. Per impostazione predefinita, RabbitMQ imposta una dimensione del buffer illimitata per le applicazioni client.

Ci sono una varietà di fattori da considerare quando si imposta un conteggio pre-fetching per i consumatori RabbitMQ. Innanzitutto, l'ambiente e la configurazione dei consumatori. Poiché i

consumatori devono conservare tutti i messaggi in memoria durante l'elaborazione, un valore di pre-fetching elevato può influire negativamente sulle prestazioni dei consumatori e, in alcuni casi, far causare potenzialmente a un consumatore un crash generale. Allo stesso modo, il broker RabbitMQ stesso mantiene tutti i messaggi che invia memorizzati nella cache fino a quando non riceve il riconoscimento del consumatore. Un valore di pre-fetching elevato può causare l'esaurimento della memoria del server RabbitMQ rapidamente se il riconoscimento automatico non è configurato per i consumatori e se i consumatori impiegano un tempo relativamente lungo per elaborare i messaggi.

Tenendo presente le considerazioni di cui sopra, si consiglia di impostare sempre un valore di pre-fetching per evitare situazioni in cui un broker RabbitMQ o i suoi consumatori esauriscano la memoria a causa di un numero elevato di messaggi non elaborati o non riconosciuti. Se è necessario ottimizzare i broker per elaborare grandi volumi di messaggi, è possibile testare i broker e i consumatori utilizzando una serie di conteggi pre-fetching per determinare il valore in cui il sovraccarico di rete diventa in gran parte insignificante rispetto al tempo impiegato da un consumatore per elaborare i messaggi.

> **Note**
>
> - Se le applicazioni client sono configurate per confermare automaticamente il recapito dei messaggi ai consumatori, l'impostazione di un valore di pre-fetching non avrà alcun effetto.
> - Tutti i messaggi sottoposti a pre-fetching vengono rimossi dalla coda.

L'esempio seguente dimostra l'impostazione di un valore pre-fetching di `10` per un singolo consumatore che utilizza la libreria client Java RabbitMQ.

```
ConnectionFactory factory = new ConnectionFactory();

Connection connection = factory.newConnection();
Channel channel = connection.createChannel();

channel.basicQos(10, false);

QueueingConsumer consumer = new QueueingConsumer(channel);
channel.basicConsume("my_queue", false, consumer);
```

> **Note**
>
> Nella libreria client Java RabbitMQ, il valore predefinito per il flag `global` è impostato su `false`, quindi l'esempio precedente può essere scritto semplicemente come `channel.basicQos(10)`.

Fase 4: Usare Celery 5.5 o versione successiva con code quorum

Python Celery, un sistema di coda di attività distribuito, può generare molti messaggi non critici quando si verifica un carico di attività elevato. Questa attività aggiuntiva del broker può innescare the section called “ `RABBITMQ_MEMORY_ALARM` ” e portare all'indisponibilità del broker. Per ridurre la possibilità di attivare un allarme di memoria, procedi come segue:

Per tutte le versioni di Celery

1. Disattiva `task_create_missing_queues`per ridurre l'affollamento della coda.
2. Quindi, disattivala `worker_enable_remote_control` per interrompere la creazione dinamica delle code. `celery@...pidbox` Ciò ridurrà il tasso di abbandono delle code per il broker.

   ```
   worker_enable_remote_control = false
   ```

3. Per ridurre ulteriormente l'attività dei messaggi non critici, disattiva Celery non includendolo `-E` o worker-send-task-events`--task-events`contrassegnandolo all'avvio dell'applicazione Celery.
4. Avvia l'applicazione Celery utilizzando i seguenti parametri:

   ```
   celery -A app_name worker --without-heartbeat --without-gossip --without-mingle
   ```

Per le versioni 5.5 e successive di Celery

1. Effettua l'aggiornamento alla versione 5.5 di Celery, la versione minima che supporta le code quorum, o a una versione successiva. Per verificare quale versione di Celery stai utilizzando, usa. `celery --version` Per ulteriori informazioni sulle code quorum, vedere. the section called “Code per il quorum”
2. Dopo l'aggiornamento a Celery 5.5 o versione successiva, configurare a «quorum». `task_default_queue_type`

3. Quindi, è necessario attivare anche le conferme di pubblicazione nelle opzioni di trasporto del broker:

```
broker_transport_options = {"confirm_publish": True}
```

Le migliori pratiche per la resilienza e il monitoraggio della rete in Amazon MQ for RabbitMQ

La resilienza della rete e il monitoraggio delle metriche dei broker sono essenziali per mantenere applicazioni di messaggistica affidabili. Completa le seguenti best practice per implementare meccanismi di ripristino automatici e strategie di monitoraggio delle risorse.

Fase 1: Ripristino automatico in caso di guasti di rete

Si consiglia di abilitare sempre il ripristino automatico della rete per evitare tempi di inattività significativi nei casi in cui le connessioni client ai nodi RabbitMQ non avessero esito positivo. La libreria client Java RabbitMQ supporta il ripristino automatico della rete per impostazione predefinita, a partire dalla versione `4.0.0`.

Il ripristino automatico della connessione viene attivato se viene generata un'eccezione non gestita nel I/O loop della connessione, se viene rilevato un timeout dell'operazione di lettura del socket o se il server perde un heartbeat.

Nei casi in cui la connessione iniziale tra un client e un nodo RabbitMQ non riesce, il ripristino automatico non verrà attivato. Si consiglia di scrivere il codice dell'applicazione per tenere conto degli errori di connessione iniziali riprovando la connessione. Nell'esempio seguente viene illustrato il tentativo di tentativi di errori iniziali di rete utilizzando la libreria client Java RabbitMQ.

```
ConnectionFactory factory = new ConnectionFactory();
// enable automatic recovery if using RabbitMQ Java client library prior to version
 4.0.0.
factory.setAutomaticRecoveryEnabled(true);
// configure various connection settings

try {
  Connection conn = factory.newConnection();
} catch (java.net.ConnectException e) {
  Thread.sleep(5000);
  // apply retry logic
```

```
}
```

> **Note**
>
> Se un'applicazione chiude una connessione utilizzando il metodo `Connection.Close`, il ripristino automatico della rete non verrà attivato o abilitato.

Fase 2: Monitora le metriche e gli allarmi del broker

Ti consigliamo di monitorare regolarmente i CloudWatch parametri e gli allarmi del tuo broker Amazon MQ for RabbitMQ per identificare e risolvere potenziali problemi prima che abbiano un impatto sulla tua applicazione di messaggistica. Il monitoraggio proattivo è essenziale per mantenere un'applicazione di messaggistica resiliente e garantire prestazioni ottimali.

Amazon MQ for RabbitMQ pubblica metriche CloudWatch che forniscono informazioni sulle prestazioni dei broker, sull'utilizzo delle risorse e sul flusso dei messaggi. Le metriche chiave da monitorare includono l'utilizzo della memoria e dell'utilizzo del disco. Puoi impostare CloudWatch allarmi quando il tuo broker si avvicina ai limiti delle risorse o subisce un peggioramento delle prestazioni.

Monitora le seguenti metriche essenziali:

`RabbitMQMemUsed` e **`RabbitMQMemLimit`**

Monitora l'utilizzo della memoria per evitare allarmi di memoria che possono bloccare la pubblicazione dei messaggi.

`RabbitMQDiskFree` e **`RabbitMQDiskFreeLimit`**

Monitora l'utilizzo del disco per evitare problemi di spazio su disco che possono causare guasti al broker.

Per le implementazioni in cluster, monitora anche le metriche specifiche dei nodi per identificare problemi specifici dei nodi.

> **Note**
>
> Per ulteriori informazioni su come prevenire l'allarme di memoria elevata, vedi Risolvere e prevenire l'allarme di memoria elevata.

Tutorial RabbitMQ

I seguenti tutorial illustrano come configurare e utilizzare RabbitMQ su Amazon MQ. Per ulteriori informazioni sull'utilizzo delle librerie client supportate in diversi linguaggi di programmazione, ad esempio Node.js, Python, .NET e altri, vedere Tutorial RabbitMQ nella Guida alle operazioni preliminari di RabbitMQ.

Argomenti

- Modifica delle preferenze del broker
- Utilizzo di Python Pika con Amazon MQ per RabbitMQ
- Risoluzione della sincronizzazione della coda sospesa di RabbitMQ
- Riduzione del numero di connessioni e canali
- Fase 2: Connect un'applicazione basata su JVM al broker
- Fase 3: (Opzionale) Connect a una AWS Lambda funzione
- Utilizzo dell'autenticazione e dell'autorizzazione OAuth 2.0 per Amazon MQ for RabbitMQ
- Utilizzo dell'autenticazione e dell'autorizzazione IAM per Amazon MQ for RabbitMQ
- Utilizzo dell'autenticazione e dell'autorizzazione LDAP per Amazon MQ for RabbitMQ
- Utilizzo dell'autenticazione e dell'autorizzazione HTTP per Amazon MQ for RabbitMQ
- Utilizzo dell'autenticazione del certificato SSL per Amazon MQ for RabbitMQ
- Utilizzo di MTL per AMQP e endpoint di gestione
- Connessione dell'applicazione JMS

Modifica delle preferenze del broker

È possibile modificare le preferenze del broker, ad esempio abilitando o CloudWatch disabilitando i log utilizzando il. Console di gestione AWS

Modifica delle opzioni del broker RabbitMQ

1. Accedere alla console Amazon MQ.
2. Dall'elenco dei broker, seleziona il tuo broker (ad esempio MyBroker), quindi scegli Modifica.
3. Nella *`MyBroker`* pagina Modifica, nella sezione Specifiche, seleziona una versione del motore Broker o un tipo di istanza del Broker.

4. Nella sezione CloudWatch Registri, fai clic sul pulsante di attivazione/disattivazione per abilitare o disabilitare i log generali. Non sono richiesti ulteriori passaggi.

   > **Note**
   >
   > - Per i broker RabbitMQ, Amazon MQ utilizza automaticamente un Service-Linked Role (SLR) per pubblicare i log generali. CloudWatch Per ulteriori informazioni, consulta the section called “Uso di ruoli collegati ai servizi”
   > - Amazon MQ non supporta la registrazione di verifiche per i broker RabbitMQ.

5. Nella sezione Maintenance (Manutenzione), configurare la pianificazione di manutenzione del broker:

   Per aggiornare il broker alle nuove versioni man mano che vengono rilasciate, scegli Abilita gli AWS aggiornamenti automatici delle versioni secondarie. Gli aggiornamenti automatici si verificano durante la finestra di manutenzione definita dal giorno della settimana, dall'ora del giorno (in formato 24 ore) e dal fuso orario (UTC per impostazione predefinita).

6. Scegliere Schedule modifications (Pianifica le modifiche).

   > **Note**
   >
   > Se scegli solo Enable automatic minor version upgrades (Abilita aggiornamenti automatici minori della versione), il pulsante cambia in Save (Salva) perché non è necessario riavviare il broker.

   Le preferenze vengono applicate al broker nel momento specificato.

Utilizzo di Python Pika con Amazon MQ per RabbitMQ

Il seguente tutorial mostra come configurare un client Python Pika client con TLS configurato per connettersi a un broker Amazon MQ per RabbitMQ. Pika è un'implementazione Python del protocollo AMQP 0-9-1 per RabbitMQ. Questo tutorial ti guida attraverso l'installazione di Pika, la dichiarazione di una coda, la configurazione di un editore per inviare messaggi all'exchange predefinito del broker e la configurazione di un consumatore per ricevere messaggi dalla coda.

Argomenti

- Prerequisiti
- Permissions
- Fase uno: creare un client Python Pika di base
- Fase due: creare un mittente e inviare un messaggio
- Fase tre: creare un consumatore e ricevere un messaggio
- Fase quattro: (facoltativa) configurare un ciclo di eventi e utilizzare i messaggi
- Fasi successive

Prerequisiti

Per completare le prime cinque fasi di questo tutorial, hai bisogno di:

- Impostazioni predefinite del broker Amazon MQ per RabbitMQ Per ulteriori informazioni, consulta Creazione di un broker Amazon MQ per RabbitMQ.
- Python 3 installato per il sistema operativo che utilizzi.
- Pika installato utilizzando `pip` di Python. Per installare Pika, apri una nuova finestra del terminale ed esegui quanto segue.

```
$ python3 -m pip install pika
```

Permissions

Per questo tutorial, hai bisogno di almeno un utente di broker Amazon MQ per RabbitMQ con il permesso di scrivere e leggere da un vhost. La tabella seguente descrive le autorizzazioni minime necessarie come modelli di espressioni regolari (regexp).

Tag	Configurazione di regexp	Scrittura di regexp	Lettura di regexp
none		`.*`	`.*`

Le autorizzazioni utente elencate forniscono solo autorizzazioni di lettura e scrittura all'utente, senza concedere l'accesso al plugin di gestione per eseguire operazioni amministrative sul broker. È possibile limitare ulteriormente le autorizzazioni fornendo modelli di regexp che limitano l'accesso

dell'utente alle code specificate. Ad esempio, se modifichi il pattern regexp di lettura in `^[hello world].*`, l'utente avrà il permesso di leggere solo dalle code che iniziano con `hello world`.

Per ulteriori informazioni sulla creazione di utenti RabbitMQ e sulla gestione di tag e autorizzazioni degli utenti, consultare Amazon MQ per gli utenti del broker RabbitMQ.

Fase uno: creare un client Python Pika di base

Per creare una classe base del client Python Pika che definisca un costruttore e fornisca il contesto SSL necessario per la configurazione TLS quando si interagisce con un broker Amazon MQ per RabbitMQ, procedere come segue.

1. Aprire una nuova finestra del terminale, creare una nuova directory per il progetto e andare alla directory.

```
$ mkdir pika-tutorial
$ cd pika-tutorial
```

2. Creare un file `basicClient.py` che contenga quanto segue:

```
import ssl
import pika

class BasicPikaClient:

    def __init__(self, rabbitmq_broker_id, rabbitmq_user, rabbitmq_password,
 region):

        # SSL Context for TLS configuration of Amazon MQ for RabbitMQ
        ssl_context = ssl.SSLContext(ssl.PROTOCOL_TLSv1_2)
        ssl_context.set_ciphers('ECDHE+AESGCM:!ECDSA')

        url = f"amqps://{rabbitmq_user}:
{rabbitmq_password}@{rabbitmq_broker_id}.mq.{region}.amazonaws.com:5671"
        parameters = pika.URLParameters(url)
        parameters.ssl_options = pika.SSLOptions(context=ssl_context)

        self.connection = pika.BlockingConnection(parameters)
        self.channel = self.connection.channel()
```

Ora è possibile definire classi aggiuntive per il mittente e il consumatore che ereditano da `BasicPikaClient`.

Fase due: creare un mittente e inviare un messaggio

Per creare un editore che dichiari una coda e invii un singolo messaggio, procedere come segue.

1. Copiare il contenuto del seguente esempio di codice e salvare localmente come `publisher.py` nella stessa directory creata nella fase precedente.

```
from basicClient import BasicPikaClient

class BasicMessageSender(BasicPikaClient):

    def declare_queue(self, queue_name):
        print(f"Trying to declare queue({queue_name})...")
        self.channel.queue_declare(queue=queue_name)

    def send_message(self, exchange, routing_key, body):
        channel = self.connection.channel()
        channel.basic_publish(exchange=exchange,
                              routing_key=routing_key,
                              body=body)
        print(f"Sent message. Exchange: {exchange}, Routing Key: {routing_key},
 Body: {body}")

    def close(self):
        self.channel.close()
        self.connection.close()

if __name__ == "__main__":

    # Initialize Basic Message Sender which creates a connection
    # and channel for sending messages.
    basic_message_sender = BasicMessageSender(
        "<broker-id>",
        "<username>",
        "<password>",
        "<region>"
    )

    # Declare a queue
    basic_message_sender.declare_queue("hello world queue")
```

```
    # Send a message to the queue.
    basic_message_sender.send_message(exchange="", routing_key="hello world queue",
body=b'Hello World!')

    # Close connections.
    basic_message_sender.close()
```

La la classe `BasicMessageSender` eredita da `BasicPikaClient` e implementa metodi aggiuntivi per rinviare una coda, inviare un messaggio alla coda e chiudere le connessioni. L'esempio di codice inoltra un messaggio allo scambio di default, con una chiave di routing uguale al nome della coda.

2. Sotto a `if __name__ == "__main__":`, sostituire i parametri passati all'istruzione del costruttore `BasicMessageSender` con le seguenti informazioni.

   - **<broker-id>**: ID univoco che Amazon MQ genera per il broker. Puoi analizzare l'ID dall'ARN del broker. Ad esempio, con il seguente ARN, `arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`, l'ID del broker sarebbe `b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`.
   - **<username>** — Il nome utente per un utente broker con autorizzazioni sufficienti per scrivere messaggi al broker.
   - **<password>** — La password per un utente broker con autorizzazioni sufficienti per scrivere messaggi al broker.
   - **<region>**— La AWS regione in cui hai creato il tuo broker Amazon MQ for RabbitMQ. Ad esempio, `us-west-2`.

3. Eseguire il comando sottostante nella stessa directory creata `publisher.py`.

```
$ python3 publisher.py
```

Se il codice viene eseguito correttamente, nella finestra del terminale verrà visualizzato il seguente output.

```
Trying to declare queue(hello world queue)...
Sent message. Exchange: , Routing Key: hello world queue, Body: b'Hello World!'
```

Fase tre: creare un consumatore e ricevere un messaggio

Per creare un consumatore che riceva un solo messaggio dalla coda, procedi come segue.

1. Copiare il contenuto del seguente esempio di codice e salvare localmente come `consumer.py` nella stessa directory.

```
from basicClient import BasicPikaClient

class BasicMessageReceiver(BasicPikaClient):

    def get_message(self, queue):
        method_frame, header_frame, body = self.channel.basic_get(queue)
        if method_frame:
            print(method_frame, header_frame, body)
            self.channel.basic_ack(method_frame.delivery_tag)
            return method_frame, header_frame, body
        else:
            print('No message returned')

    def close(self):
        self.channel.close()
        self.connection.close()


if __name__ == "__main__":

    # Create Basic Message Receiver which creates a connection
    # and channel for consuming messages.
    basic_message_receiver = BasicMessageReceiver(
        "<broker-id>",
        "<username>",
        "<password>",
        "<region>"
    )

    # Consume the message that was sent.
    basic_message_receiver.get_message("hello world queue")

    # Close connections.
    basic_message_receiver.close()
```

Analogamente all'editore creato nel passaggio precedente, `BasicMessageReceiver` eredita `BasicPikaClient` e implementa metodi aggiuntivi per ricevere un singolo messaggio e chiudere le connessioni.

2. Sotto a `if __name__ == "__main__":`, sostituire i parametri passati all'istruzione del costruttore `BasicMessageReceiver` con le seguenti informazioni.
3. Nella directory del progetto, esegui il seguente comando.

```
$ python3 consumer.py
```

Se il codice viene eseguito correttamente, verrà visualizzato il corpo del messaggio e le intestazioni, inclusa la chiave di routing, nella finestra del terminale.

```
<Basic.GetOk(['delivery_tag=1', 'exchange=', 'message_count=0',
 'redelivered=False', 'routing_key=hello world queue'])> <BasicProperties> b'Hello
 World!'
```

Fase quattro: (facoltativa) configurare un ciclo di eventi e utilizzare i messaggi

Per consumare più messaggi da una coda, usa il metodo e la funzione di callback di `basic_consume` di Pika come mostrato di seguito

1. In `consumer.py`, aggiungere la seguente definizione del metodo alla classe `BasicMessageReceiver`.

```
def consume_messages(self, queue):
    def callback(ch, method, properties, body):
        print(" [x] Received %r" % body)

    self.channel.basic_consume(queue=queue, on_message_callback=callback,
 auto_ack=True)

    print(' [*] Waiting for messages. To exit press CTRL+C')
    self.channel.start_consuming()
```

2. In `consumer.py`, sotto a `if __name__ == "__main__":`, invocare il metodo `consume_messages` definito nella fase precedente.

```
if __name__ == "__main__":

    # Create Basic Message Receiver which creates a connection and channel for
 consuming messages.
    basic_message_receiver = BasicMessageReceiver(
        "<broker-id>",
        "<username>",
        "<password>",
        "<region>"
    )

    # Consume the message that was sent.
    # basic_message_receiver.get_message("hello world queue")

    # Consume multiple messages in an event loop.
    basic_message_receiver.consume_messages("hello world queue")

    # Close connections.
    basic_message_receiver.close()
```

3. Eseguire `consumer.py` di nuovo e, in caso di esito positivo, i messaggi in coda verranno visualizzati nella finestra del terminale.

```
[*] Waiting for messages. To exit press CTRL+C
[x] Received b'Hello World!'
[x] Received b'Hello World!'
...
```

Fasi successive

- Per ulteriori informazioni su altre librerie client RabbitMQ supportate, consultaDocumentazione relativa al client RabbitMQ sul sito Web RabbitMQ.

Risoluzione della sincronizzazione della coda sospesa di RabbitMQ

In un'implementazione cluster Amazon MQ per RabbitMQ, i messaggi pubblicati in ogni coda vengono replicati su tre nodi del broker. Questa replica, denominata mirroring, fornisce alta disponibilità per i broker RabbitMQ. Le code in un'implementazione cluster sono costituite da una replica principale su un nodo e uno o più mirror. Ogni operazione applicata a una coda sottoposta

a mirroring, inclusi i messaggi di accodamento, viene prima applicata alla coda principale e quindi replicata attraverso i relativi mirror.

Ad esempio, si consideri una coda sottoposta a mirroring replicata su tre nodi: il nodo principale (`main`) e due mirror (`mirror-1` e `mirror-2`). Se tutti i messaggi in questa coda sottoposta a mirroring vengono propagati correttamente a tutti i mirror, la coda viene sincronizzata. Se un nodo (`mirror-1`) diventa non disponibile per un dato intervallo di tempo, la coda è comunque operativa e può continuare ad accodare i messaggi. Tuttavia, per sincronizzare la coda, i messaggi pubblicati in `main` mentre `mirror-1` non è disponibile devono essere replicati in `mirror-1`.

Per ulteriori informazioni sul mirroring classico, consultare Code classiche sottoposte a mirroring nel sito Web RabbitMQ.

Manutenzione e sincronizzazione delle code

Durante le finestre di manutenzione, Amazon MQ esegue tutti i lavori di manutenzione su un nodo alla volta per garantire che il broker rimanga operativo. Di conseguenza, potrebbe essere necessario sincronizzare le code quando ogni nodo riprende l'operazione. Durante la sincronizzazione, i messaggi che devono essere replicati nei mirror vengono caricati in memoria dal volume Amazon Elastic Block Store (Amazon EBS) corrispondente per essere elaborati in batch. L'elaborazione dei messaggi in batch consente alle code di sincronizzarsi più velocemente.

Se le code vengono mantenute brevi e i messaggi sono piccoli, le code si sincronizzano correttamente e riprendono l'operazione come previsto. Tuttavia, se la quantità di dati in un batch si avvicina al limite di memoria del nodo, il nodo genera un allarme di memoria elevata, sospendendo la sincronizzazione della coda. Puoi confermare l'utilizzo della memoria confrontando le `RabbitMemUsed` metriche del nodo del `RabbitMqMemLimit` broker in CloudWatch. La sincronizzazione non può essere completata finché i messaggi non vengono consumati o eliminati o il numero di messaggi nel batch non viene ridotto.

> **Note**
>
> La riduzione delle dimensioni del batch di sincronizzazione della coda può comportare un numero maggiore di transazioni di replica.

Per risolvere una sincronizzazione della coda in pausa, attenersi alla procedura descritta in questo tutorial, in cui viene illustrata l'applicazione di una policy `ha-sync-batch-size`, quindi riavviare la sincronizzazione della coda.

Argomenti

- Prerequisiti
- Fase 1: applicare una policy ha-sync-batch-size
- Fase 2: riavviare la sincronizzazione della coda
- Fasi successive
- Risorse correlate

Prerequisiti

Per questo tutorial, è necessario disporre di un utente per il broker Amazon MQ per RabbitMQ con autorizzazioni di amministratore. È possibile utilizzare l'utente amministratore creato al momento della creazione del broker o un altro utente che potrebbe essere stato creato successivamente. Nella tabella seguente vengono forniti i tag dell'utente amministratore necessari e le autorizzazioni come modelli di espressione regolare (regexp).

Tag	Lettura di regexp	Configurazione di regexp	Scrittura di regexp
`administrator`	`.*`	`.*`	`.*`

Per ulteriori informazioni sulla creazione di utenti RabbitMQ e sulla gestione di tag e autorizzazioni degli utenti, consultare Amazon MQ per gli utenti del broker RabbitMQ.

Fase 1: applicare una policy **`ha-sync-batch-size`**

Nelle procedure seguenti viene illustrato l'aggiunta di una policy applicabile a tutte le code create nel broker. È possibile utilizzare la console Web RabbitMQ o l'API di gestione RabbitMQ. Per ulteriori informazioni, consultare Plugin di gestione sul sito Web RabbitMQ.

Applicazione di una policy **`ha-sync-batch-size`** utilizzando la console Web RabbitMQ

1. Accedere alla console Amazon MQ.
2. Nel pannello di navigazione a sinistra selezionare Brokers (Broker).
3. Nell'elenco dei broker, scegliere il nome del broker a cui si desidera applicare la nuova policy.
4. Alla pagina de broker, nella sezione Connections (Connessioni), scegliere l'URL della console Web RabbitMQ. La console Web RabbitMQ si apre in una nuova scheda o finestra del browser.

5. Accedi alla console Web RabbitMQ con le credenziali di accesso dell'amministratore del broker.
6. Nella console Web RabbitMQ, nella parte superiore della pagina, selezionare Admin (Amministratore).
7. Alla pagina Admin (Amministratore), nel pannello di navigazione destro, selezionare Policies (Policy).
8. Alla pagina Policies (Policy), comparirà un elenco delle policy dell'utente connesso del broker. Sotto User policies (Policy utente), espandere Add / update a policy (Aggiungere/aggiornare una policy).

   > **Note**
   >
   > Per impostazione predefinita, i cluster Amazon MQ per RabbitMQ vengono creati con una policy del broker iniziale denominata `ha-all-AWS-OWNED-DO-NOT-DELETE`. Amazon MQ gestisce questa policy per garantire che ogni coda del broker venga replicata su tutti e tre i nodi e che le code vengano sincronizzate automaticamente.

9. Per creare una nuova policy del broker, in Add / update a policy (Aggiungere/aggiornare una policy), effettua le seguenti operazioni:

   a. Per Name (Nome), immettere un nome per la policy, ad esempio **`batch-size-policy`**.

   b. Per Pattern (Modello), inserisci il modello regexp **`.*`** in modo che la policy corrisponda a tutte le code nel broker.

   c. Per Apply to (Applica a), scegliere Exchanges and queues (Scambi e code) dall'elenco a discesa.

   d. Per Priority (Priorità), immettere un numero intero maggiore di tutte le altre policy applicate al vhost. È possibile applicare esattamente un set di definizioni di policy alle code e agli scambi di RabbitMQ in qualsiasi momento. RabbitMQ sceglie la policy corrispondente al valore di priorità più alto. Per ulteriori informazioni sulle priorità delle policy e su come combinare le policy, consultare Policy nella documentazione del server RabbitMQ.

   e. Per Definition (Definizione), aggiungere le seguenti coppie chiave-valore:

      - **`ha-sync-batch-size`**=*`100`*. Scegli Numero dall'elenco a discesa.

> **Note**
>
> Potrebbe essere necessario regolare e calibrare il valore di `ha-sync-batch-size` in base al numero e alle dimensioni dei messaggi non sincronizzati nelle code.

- **`ha-mode=all`**. Scegliere String (Stringa) dall'elenco a discesa.

> **Important**
>
> La definizione `ha-mode` è necessaria per tutte le policy correlate all'alta disponibilità. L'omissione si traduce in un errore di convalida.

- **`ha-sync-mode=automatic`**. Scegliere String (Stringa) dall'elenco a discesa.

> **Note**
>
> La definizione `ha-sync-mode` è necessaria per tutte le policy personalizzate. Se omessa, Amazon MQ aggiunge automaticamente la definizione.

f. Scegliere Add / update policy (Aggiungi/aggiorna policy).

10. Verificare che la nuova policy sia visualizzata nell'elenco delle policy dell'utente.

Applicazione di una policy **`ha-sync-batch-size`** utilizzando l'API di gestione RabbitMQ

1. Accedere alla console Amazon MQ.
2. Nel pannello di navigazione a sinistra selezionare Brokers (Broker).
3. Nell'elenco dei broker, scegliere il nome del broker a cui si desidera applicare la nuova policy.
4. Alla pagina del broker, nella sezione Connections (Connessioni), prendere nota dell'URL della Console Web RabbitMQ. Questo è l'endpoint del broker che si utilizza in una richiesta HTTP.
5. Aprire un nuovo terminale o una finestra della riga di comando a scelta.
6. Per creare una nuova policy del broker, inserisci il seguente comando `curl`. Questo comando presuppone una coda sul vhost predefinito `/`, che è codificato come `%2F`.

> **Note**
>
> Sostituisci *username* e inserisci *password* le credenziali di accesso dell'amministratore del broker. Potrebbe essere necessario modificare e calibrare il valore di `ha-sync-batch-size` (*100*) in base al numero e alla dimensione dei messaggi non sincronizzati nelle code. Sostituire l'endpoint del broker con l'URL annotato in precedenza.

```
curl -i -u username:password -H "content-type:application/json" -XPUT \
-d '{"pattern":".*", "priority":1, "definition":{"ha-sync-batch-size":100, "ha-
mode":"all", "ha-sync-mode":"automatic"}}' \
https://b-589c045f-f8ln-4ab0-a89c-co62e1c32ef8.mq.us-west-2.amazonaws.com/api/
policies/%2F/batch-size-policy
```

7. Per confermare che la nuova policy sia stata aggiunta alle policy dell'utente del broker, inserisci il comando `curl` per elencare tutte le policy del broker.

```
curl -i -u username:password https://b-589c045f-f8ln-4ab0-a89c-co62e1c32ef8.mq.us-
west-2.amazonaws.com/api/policies
```

Fase 2: riavviare la sincronizzazione della coda

Dopo aver applicato una nuova policy `ha-sync-batch-size` al broker, riavviare la sincronizzazione della coda.

Riavvio della sincronizzazione della coda utilizzando la console Web RabbitMQ

> **Note**
>
> Per aprire la console Web RabbitMQ, consultare le istruzioni precedenti della fase 1 di questo tutorial.

1. Nella console Web RabbitMQ, nella parte superiore della pagina, selezionare Queues (Code).

2. Alla pagina Queues (Code), in All queues (Tutte le code), individuare le code sospese. Nella riga Policy, la coda dovrebbe elencare il nome della nuova policy che hai creato (ad esempio,). `batch-size-policy`
3. Per riavviare il processo di sincronizzazione con una dimensione del batch ridotta, annulla innanzitutto la sincronizzazione della coda. Quindi riavvia la sincronizzazione della coda.

> **Note**
>
> Se la sincronizzazione si interrompe e non viene completata correttamente, provare a ridurre il valore `ha-sync-batch-size` e riavviare nuovamente la sincronizzazione della coda.

Fasi successive

- Una volta che la coda si è sincronizzata correttamente, puoi monitorare la quantità di memoria utilizzata dai nodi RabbitMQ visualizzando la metrica Amazon. CloudWatch `RabbitMQMemUsed` È inoltre possibile visualizzare il parametro `RabbitMQMemLimit` per monitorare il limite di memoria di un nodo. Per ulteriori informazioni, consultare Accesso ai CloudWatch parametri per Amazon MQ e CloudWatch Metriche disponibili per i broker Amazon MQ for RabbitMQ.
- Per impedire la sincronizzazione delle code sospese, si consiglia di mantenere le code brevi ed elaborare i messaggi. Per carichi di lavoro con messaggi di dimensioni maggiori, si consiglia inoltre di aggiornare il tipo di istanza del broker a una dimensione dell'istanza maggiore con più memoria. Per ulteriori informazioni sui tipi di istanze del broker e sulla modifica delle preferenze del broker, consulta. Modifica delle preferenze del broker
- Quando crei un broker Amazon MQ per RabbitMQ, Amazon MQ applica un set predefinito di policy e limiti dell'host virtuale per ottimizzare le prestazioni del broker. Se il broker non dispone di policy e limiti predefiniti consigliati, consigliamo di crearli tu stesso. Per ulteriori informazioni sulla creazione di policy e limiti vhost predefiniti, consultare https://docs.aws.amazon.com//amazon-mq/latest/developer-guide/rabbitmq-defaults.html.

Risorse correlate

- UpdateBrokerInput— Utilizza questa proprietà broker per aggiornare un tipo di istanza del broker utilizzando l'API Amazon MQ.

- Parametri e policy (Documentazione del server RabbitMQ): ulteriori informazioni sui parametri e sui policy di RabbitMQ sul sito Web di RabbitMQ.
- API HTTP di gestione RabbitMQ: ulteriori informazioni sull'API di gestione RabbitMQ.

Riduzione del numero di connessioni e canali

Le connessioni al tuo broker RabbitMQ su Amazon MQ possono essere chiuse dalle applicazioni client o chiudendole manualmente utilizzando la console web di RabbitMQ. Per chiudere una connessione utilizzando la console web di RabbitMQ, procedi come segue:

1. Accedi Console di gestione AWS e apri la console web RabbitMQ del tuo broker.
2. Nella console RabbitMQ, scegliere la scheda Connessioni.
3. Sulla pagina Connessioni, sotto Tutte le connessioni, scegliere il nome della connessione che si desidera chiudere dall'elenco.
4. Nella pagina dei dettagli della connessione, selezionare Chiudere questa connessione per espandere la sezione, quindi scegliere Forza chiusura. Opzionalmente è possibile sostituire il testo predefinito per Motivo con la tua descrizione. RabbitMQ su Amazon MQ restituirà il motivo specificato al client quando chiudi la connessione.
5. Scegliere OK nella finestra di dialogo per confermare e chiudere la connessione.

Quando si chiude una connessione, verranno chiusi anche tutti i canali associati alla connessione chiusa.

> **Note**
>
> Le applicazioni client possono essere configurate per ristabilire automaticamente le connessioni al broker dopo la chiusura. In questo caso, la chiusura delle connessioni dalla console Web del broker non sarà sufficiente per ridurre il numero di connessioni o canali.

Per i broker senza accesso pubblico, è possibile bloccare temporaneamente le connessioni negando il traffico in entrata sulla porta del protocollo di messaggio appropriata, ad esempio porta `5671` per connessioni AMQP. È possibile bloccare la porta nel gruppo di sicurezza fornito ad Amazon MQ durante la creazione del broker. Per ulteriori informazioni sulla modifica del tuo gruppo di sicurezza, consulta Aggiunta di regole a un gruppo di sicurezza nella Guida per l'utente di Amazon VPC.

Fase 2: Connect un'applicazione basata su JVM al broker

Dopo aver creato un broker RabbitMQ, è possibile collegare l'applicazione ad esso. Di seguito sono riportati esempi che mostrano come è possibile utilizzare la libreria client Java RabbitMQ per creare una connessione al broker, creare una coda e inviare un messaggio. È possibile connettersi ai broker RabbitMQ utilizzando le librerie client RabbitMQ supportate per una varietà di linguaggi. Per ulteriori informazioni sulle librerie client RabbitMQ supportate, consulta Librerie client e strumenti per sviluppatori di RabbitMQ.

Prerequisiti

> **Note**
>
> I seguenti prerequisiti sono applicabili solo ai broker RabbitMQ creati senza accessibilità pubblica. Se stai creando un broker con accessibilità pubblica, è possibile ignorarli.

Abilitazione degli attributi VPC

Per garantire che il broker sia accessibile all'interno del VPC, è necessario abilitare gli attributi VPC `enableDnsHostnames` e `enableDnsSupport`. Per ulteriori informazioni, consultare Supporto del DNS nel VPC nella Guida per l'utente di Amazon VPC.

Abilitazione delle connessioni in entrata

1. Accedere alla console Amazon MQ.
2. Dall'elenco dei broker, scegli il nome del tuo broker (ad esempio,). MyBroker
3. Nella ***`MyBroker`***pagina, nella sezione Connessioni, annota gli indirizzi e le porte dell'URL della console web del broker e dei protocolli a livello di cavo.
4. Nella sezione Details (Dettagli), in Security and network (Sicurezza e rete), scegliere il nome del gruppo di sicurezza o

   .

   Viene visualizzata la pagina Security Groups (Gruppi di sicurezza) del pannello di controllo EC2.
5. Scegli il tuo gruppo di sicurezza dall'elenco.
6. Nella parte inferiore della pagina scegli Inbound (In entrata), quindi scegli Edit (Modifica).

7. Nella finestra di dialogo Edit inbound rules (Modifica le regole in entrata), aggiungere una regola per ogni URL o endpoint che si desidera rendere accessibile pubblicamente (nell'esempio seguente viene illustrato come eseguire questa operazione per una console Web del broker).

   a. Selezionare Add Rule (Aggiungi regola).

   b. Per Type (Tipo) seleziona Custom TCP (TCP personalizzato).

   c. Per Source (Origine), lasciare selezionato Custom (Personalizzato), quindi inserire l'indirizzo IP del sistema a cui desideri poter accedere alla console Web (ad esempio, `192.0.2.1`).

   d. Scegli Save (Salva).

   Il broker può ora accettare connessioni in entrata.

Aggiunta di dipendenze Java

Se stai usando Apache Maven per automatizzare le build, aggiungi la seguente dipendenza al tuo file `pom.xml`. Per ulteriori informazioni sui file Project Object Model in Apache Maven, vedere Introduzione al POM.

```
<dependency>
    <groupId>com.rabbitmq</groupId>
    <artifactId>amqp-client</artifactId>
    <version>5.9.0</version>
</dependency>
```

Se stai usando Gradle per automatizzare le build, dichiarare la seguente dipendenza.

```
dependencies {
    compile 'com.rabbitmq:amqp-client:5.9.0'
}
```

Importazione delle classi **`Connection`** e **`Channel`**

Il client Java RabbitMQ utilizza `com.rabbitmq.client` come pacchetto di primo livello, con le classi API `Connection` e `Channel` che rappresentano una connessione e canale AMQP 0-9-1, rispettivamente. Importare le classi `Connection` e `Channel` prima di utilizzarle, come mostrato nell'esempio seguente.

```
import com.rabbitmq.client.Connection;
```

```
import com.rabbitmq.client.Channel;
```

Creare un **`ConnectionFactory`** e connetterlo al broker

Utilizzare l'esempio seguente per creare un'istanza della classe `ConnectionFactory` con i parametri dati. Utilizzare il metodo `setHost` per configurare l'endpoint del broker annotato in precedenza. Per le connessioni a livello di filo AMQPS, usare la porta `5671`.

```
ConnectionFactory factory = new ConnectionFactory();

factory.setUsername(username);
factory.setPassword(password);

//Replace the URL with your information
factory.setHost("b-c8352341-ec91-4a78-ad9c-a43f23d325bb.mq.us-west-2.amazonaws.com");
factory.setPort(5671);

// Allows client to establish a connection over TLS
factory.useSslProtocol();

// Create a connection
Connection conn = factory.newConnection();

// Create a channel
Channel channel = conn.createChannel();
```

Pubblicazione di un messaggio in uno scambio

Puoi utilizzare `Channel.basicPublish` per pubblicare messaggi in uno scambio. L'esempio seguente utilizza la classe AMQP `Builder` per creare un oggetto di proprietà del messaggio con tipo di contenuto `plain/text`.

```
byte[] messageBodyBytes = "Hello, world!".getBytes();
channel.basicPublish(exchangeName, routingKey,
             new AMQP.BasicProperties.Builder()
               .contentType("text/plain")
               .userId("userId")
               .build(),
               messageBodyBytes);
```

> **Note**
>
> Nota che `BasicProperties` è una classe interna della classe di titolare generata automaticamente, `AMQP`.

Sottoscrizione a una coda e ricezione di un messaggio

È possibile ricevere un messaggio sottoscrivendosi a una coda utilizzando l'interfaccia `Consumer`. Una volta sottoscritti, i messaggi verranno recapitati automaticamente al loro arrivo.

Il modo più semplice per implementare un `Consumer` è usare la sottoclasse `DefaultConsumer`. Un oggetto `DefaultConsumer` può essere passato come parte di una chiamata `basicConsume` per configurare la sottoscrizione come mostrato nell'esempio seguente.

```
boolean autoAck = false;
channel.basicConsume(queueName, autoAck, "myConsumerTag",
     new DefaultConsumer(channel) {
         @Override
         public void handleDelivery(String consumerTag,
                                    Envelope envelope,
                                    AMQP.BasicProperties properties,
                                    byte[] body)
             throws IOException
         {
             String routingKey = envelope.getRoutingKey();
             String contentType = properties.getContentType();
             long deliveryTag = envelope.getDeliveryTag();
             // (process the message components here ...)
             channel.basicAck(deliveryTag, false);
         }
     });
```

> **Note**
>
> Poiché abbiamo specificato `autoAck = false`, è necessario riconoscere i messaggi recapitati al `Consumer`, più convenientemente con il metodo `handleDelivery` come mostrato nell'esempio.

Chiusura della connessione e disconnessione dal broker

Per disconnettersi dal broker RabbitMQ, chiudere sia il canale che la connessione come illustrato di seguito.

```
channel.close();
conn.close();
```

> **Note**
>
> Per ulteriori informazioni sull'utilizzo della libreria client Java RabbitMQ, consulta la Guida all'API del client Java RabbitMQ.

Fase 3: (Opzionale) Connect a una AWS Lambda funzione

AWS Lambda può connettersi e utilizzare i messaggi del tuo broker Amazon MQ. Quando si connette un broker a Lambda, si crea una mappatura delle origini degli eventi che legge i messaggi da una coda e richiama la funzione in modo sincrono. La mappatura dell'origine degli eventi creata legge i messaggi dal broker in batch e li converte in un payload Lambda sotto forma di oggetto JSON.

Connessione del broker a una funzione Lambda

1. Aggiungere le seguenti autorizzazioni del ruolo IAM al ruolo di esecuzione della funzione Lambda.

   - mq: DescribeBroker
   - ec2: CreateNetworkInterface
   - ec2: DeleteNetworkInterface
   - ec2: DescribeNetworkInterfaces
   - ec2: DescribeSecurityGroups
   - ec2: DescribeSubnets
   - ec2: DescribeVpcs
   - registri: CreateLogGroup
   - registri: CreateLogStream
   - registri: PutLogEvents
   - gestore dei segreti: GetSecretValue

> **Note**
>
> Senza le necessarie autorizzazioni IAM, la tua funzione non sarà in grado di leggere correttamente i record dalle risorse di Amazon MQ.

2. (Opzionale) Se hai creato un broker senza accessibilità pubblica, devi effettuare una delle seguenti operazioni per consentire a Lambda di connettersi al broker:

   - Configurare un gateway NAT per sottorete pubblica. Per ulteriori informazioni, consultare Accesso a Internet e ai servizi per funzioni connesse a un VPC nella AWS Lambda Guida per gli sviluppatori.
   - Creare una connessione tra Amazon Virtual Private Cloud (Amazon VPC) e Lambda mediante un endpoint VPC. Il tuo Amazon VPC deve inoltre connettersi agli endpoint AWS Security Token Service (AWS STS) e Secrets Manager. Per ulteriori informazioni, consulta Configuring interface VPC endpoints for Lambda nella AWS Lambda Guida per gli sviluppatori.

3. Configurare il broker come origine dell'evento per una funzione Lambda che utilizza la Console di gestione AWS. Puoi anche usare il comando. `create-event-source-mapping` AWS Command Line Interface

4. Scrivere un codice per la funzione Lambda per elaborare i messaggi utilizzati dal broker. Il payload Lambda recuperato dalla mappatura dell'origine dell'evento dipende dal tipo di motore del broker. Di seguito è riportato un esempio di payload Lambda per una coda Amazon MQ per RabbitMQ.

> **Note**
>
> Nell'esempio RabbitMQ, `test` è il nome della coda e `/` è il nome dell'host virtuale predefinito. Quando si ricevono messaggi, l'origine eventi elenca i messaggi in `test::/`.

```
{
  "eventSource": "aws:rmq",
  "eventSourceArn": "arn:aws:mq:us-
west-2:112556298976:broker:test:b-9bcfa592-423a-4942-879d-eb284b418fc8",
  "rmqMessagesByQueue": {
    "test::/": [
      {
```

```
        "basicProperties": {
          "contentType": "text/plain",
          "contentEncoding": null,
          "headers": {
            "header1": {
              "bytes": [
                118,
                97,
                108,
                117,
                101,
                49
              ]
            },
            "header2": {
              "bytes": [
                118,
                97,
                108,
                117,
                101,
                50
              ]
            },
            "numberInHeader": 10
          }
          "deliveryMode": 1,
          "priority": 34,
          "correlationId": null,
          "replyTo": null,
          "expiration": "60000",
          "messageId": null,
          "timestamp": "Jan 1, 1970, 12:33:41 AM",
          "type": null,
          "userId": "AIDACKCEVSQ6C2EXAMPLE",
          "appId": null,
          "clusterId": null,
          "bodySize": 80
        },
        "redelivered": false,
        "data": "eyJ0aW1lb3V0IjowLCJkYXRhIjoiQ1pybWYwR3c4T3Y0YnFMUXhENEUifQ=="
      }
    ]
  }
```

```
}
```

Per ulteriori informazioni sulla connessione di Amazon MQ a Lambda, sulle opzioni supportate da Lambda per un'origine di eventi Amazon MQ e sugli errori di mappatura delle sorgenti degli eventi, consulta Using Lambda with Amazon MQ nella Developer Guide.AWS Lambda

Utilizzo dell'autenticazione e dell'autorizzazione OAuth 2.0 per Amazon MQ for RabbitMQ

Questo tutorial descrive come configurare l'autenticazione OAuth 2.0 per i broker Amazon MQ for RabbitMQ utilizzando Amazon Cognito come provider 2.0. OAuth

> **Note**
>
> Amazon Cognito non è disponibile in Cina (Pechino) e Cina (Ningxia).

> **Important**
>
> Questo tutorial è specifico per Amazon Cognito, ma puoi utilizzare altri provider di identità ()IdPs. Per ulteriori informazioni, consulta Esempi di autenticazione OAuth 2.0.

In questa pagina

- Prerequisiti per configurare l'autenticazione OAuth 2.0
- Configurazione dell'autenticazione OAuth 2.0 con Amazon Cognito utilizzando AWS CLI
- Configurazione OAuth 2.0 e autenticazione semplice con Amazon Cognito

Prerequisiti per configurare l'autenticazione OAuth 2.0

Puoi impostare le risorse Amazon Cognito richieste in questo tutorial distribuendo lo stack AWS CDK Amazon Cognito stack for RabbitMQ 2 plug-in. OAuth Se stai configurando Amazon Cognito manualmente, assicurati di soddisfare i seguenti prerequisiti prima di configurare la OAuth versione 2.0 sui broker Amazon MQ for RabbitMQ:

Prerequisiti per configurare Amazon Cognito

- Configura un endpoint Amazon Cognito creando un pool di utenti. A tale scopo, consulta il blog intitolato Come usare OAuth 2.0 in Amazon Cognito: scopri le OAuth diverse sovvenzioni 2.0.
- Crea un server di risorse chiamato `rabbitmq` nel pool di utenti con i seguenti ambiti definiti:`read:all`,, `write:all` e. `configure:all` `tag:administrator` Questi ambiti verranno associati alle autorizzazioni RabbitMQ.

  Per informazioni sulla creazione di un server di risorse, consulta Definizione di un server di risorse per il pool di utenti (Console di gestione AWS) nella Amazon Cognito Developer Guide.
- Crea i seguenti client applicativi:
  - Client applicativo per il tipo di pool di utenti`Machine-to-Machine application`. Si tratta di un client riservato con un client segreto che verrà utilizzato per i client RabbitMQ AMQP. Per ulteriori informazioni sui client applicativi e sulla creazione di un client, consulta Tipi di client di app e Creazione di un client di app.
  - Client applicativo per il tipo di pool di utenti`Single-page application`. Si tratta di un client pubblico che verrà utilizzato per accedere gli utenti alla console di gestione di RabbitMQ. Devi aggiornare questo client dell'applicazione per includere l'endpoint del broker Amazon MQ for RabbitMQ che creerai nella procedura seguente come URL di callback consentito. Per ulteriori informazioni, consulta Configurazione dell'accesso gestito con la console Amazon Cognito.

Prerequisito per configurare Amazon MQ

- Un'installazione Docker funzionante per eseguire uno script bash che verifica se la configurazione OAuth 2.0 è riuscita o meno.
- AWS CLI versione >= `2.28.23` per rendere opzionale l'aggiunta di un nome utente e una password durante la creazione del broker.

Configurazione dell'autenticazione OAuth 2.0 con Amazon Cognito utilizzando AWS CLI

La procedura seguente mostra come configurare l'autenticazione OAuth 2.0 per i broker Amazon MQ for RabbitMQ utilizzando Amazon Cognito come IdP. Questa procedura serve AWS CLI a creare e configurare le risorse necessarie.

Nella procedura seguente, assicuratevi di sostituire i valori segnaposto, come ConfigurationID e Revision, con i relativi valori *<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>* effettivi*<2>*.

1. Create una nuova configurazione utilizzando il AWS CLI comando create-configuration, come illustrato nell'esempio seguente.

```
aws mq create-configuration \
  --name "rabbitmq-oauth2-config" \
  --engine-type "RABBITMQ" \
  --engine-version "3.13"
```

   Questo comando restituisce una risposta simile all'esempio seguente.

```
{
    "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-fa3390a5-7e01-4559-
ae0c-eb15b38b22ca",
    "AuthenticationStrategy": "simple",
    "Created": "2025-07-17T16:03:01.759943+00:00",
    "Id": "c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
    "LatestRevision": {
    "Created": "2025-07-17T16:03:01.759000+00:00",
    "Description": "Auto-generated default for rabbitmq-oauth2-config on RabbitMQ
 3.13",
    "Revision": 1
    },
    "Name": "rabbitmq-oauth2-config"
}
```

2. Create un file di configurazione chiamato **`rabbitmq.conf`** a utilizzare OAuth 2.0 come metodo di autenticazione e autorizzazione, come illustrato nell'esempio seguente.

```
auth_backends.1 = oauth2

# FIXME: Update this value with the token signing key URL of your Amazon Cognito
 user pool.
# If you used the AWS CDK stack to deploy Amazon Cognito, this is one of the stack
 outputs.
auth_oauth2.jwks_url = ${RabbitMqOAuth2TestStack.JwksUri}
auth_oauth2.resource_server_id = rabbitmq
# Amazon Cognito does not include an audience field in access tokens
auth_oauth2.verify_aud = false
```

```
# Amazon Cognito does not allow * in its custom scopes. Use aliases to translate
 between Amazon Cognito and RabbitMQ.
auth_oauth2.scope_prefix = rabbitmq/
auth_oauth2.scope_aliases.1.alias = rabbitmq/read:all
auth_oauth2.scope_aliases.1.scope = rabbitmq/read:*/*
auth_oauth2.scope_aliases.2.alias = rabbitmq/write:all
auth_oauth2.scope_aliases.2.scope = rabbitmq/write:*/*
auth_oauth2.scope_aliases.3.alias = rabbitmq/configure:all
auth_oauth2.scope_aliases.3.scope = rabbitmq/configure:*/*

# Allow OAuth 2.0 login for RabbitMQ management console
management.oauth_enabled = true
# FIXME: Update this value with the client ID of your public application client
management.oauth_client_id
 = ${RabbitMqOAuth2TestStack.ManagementConsoleAppClientId}
# FIXME: Update this value with the base JWKS URI (without /.well-known/jwks.json)
auth_oauth2.issuer = ${RabbitMqOAuth2TestStack.Issuer}
management.oauth_scopes = rabbitmq/tag:administrator
```

Questa configurazione utilizza alias di ambito per mappare gli ambiti definiti in Amazon Cognito a ambiti compatibili con RabbitMQ.

3. Aggiorna la configurazione utilizzando il comando update-configuration AWS CLI come mostrato nell'esempio seguente. In questo comando, aggiungi l'ID di configurazione ricevuto nella risposta del passaggio 1 di questa procedura. Ad esempio, **`c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca`**.

```
aws mq update-configuration \
  --configuration-id "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>" \
  --data "$(cat rabbitmq.conf | base64 --wrap=0)"
```

Questo comando restituisce una risposta simile all'esempio seguente.

```
{
    "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-b600ac8e-8183-4f74-
a713-983e59f30e3d",
    "Created": "2025-07-17T16:57:04.520931+00:00",
    "Id": "c-b600ac8e-8183-4f74-a713-983e59f30e3d",
    "LatestRevision": {
        "Created": "2025-07-17T16:57:39.172000+00:00",
        "Revision": 2
    },
```

```
    "Name": "rabbitmq-oauth2-config",
    "Warnings": []
}
```

4. Crea un broker con la configurazione OAuth 2.0 creata nel passaggio 2 di questa procedura. A tale scopo, utilizzate il AWS CLI comando create-broker come illustrato nell'esempio seguente. In questo comando, fornite l'ID di configurazione e il numero di revisione ottenuti rispettivamente nelle risposte dei passaggi 1 e 2. Ad esempio **`c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca`** e **`2`**.

```
aws mq create-broker \
 --broker-name "rabbitmq-oauth2-broker" \
 --engine-type "RABBITMQ" \
 --engine-version "3.13" \
 --host-instance-type "mq.m7g.large" \
 --deployment-mode "CLUSTER_MULTI_AZ" \
 --logs '{"General": true}' \
 --publicly-accessible \
 --configuration '{"Id": "<c-fa3390a5-7e01-4559-ae0c-
eb15b38b22ca>","Revision": <2>}' \
```

Questo comando restituisce una risposta simile all'esempio seguente.

```
{
    "BrokerArn": "arn:aws:mq:us-west-2:123456789012:broker:rabbitmq-oauth2-
broker:b-2a1b5133-a10c-49d2-879b-8c176c34cf73",
    "BrokerId": "b-2a1b5133-a10c-49d2-879b-8c176c34cf73"
}
```

5. Verificate che lo stato del broker passi da `CREATION_IN_PROGRESS` a`RUNNING`, utilizzando il AWS CLI comando describe-broker, come illustrato nell'esempio seguente. In questo comando, fornisci l'ID broker che hai ottenuto nel risultato del passaggio precedente. Ad esempio,. **`b-2a1b5133-a10c-49d2-879b-8c176c34cf73`**

```
aws mq describe-broker \
 --broker-id "<b-2a1b5133-a10c-49d2-879b-8c176c34cf73>"
```

Questo comando restituisce una risposta simile all'esempio seguente. La risposta seguente è una versione abbreviata dell'output completo restituito dal `describe-broker` comando. Questa risposta mostra lo stato del broker e la strategia di autenticazione utilizzata per proteggere il

broker. In questo caso, la strategia di `config_managed` autenticazione indica che il broker utilizza OAuth 2 metodi di autenticazione.

```
{
    "AuthenticationStrategy": "config_managed",
    ...,
    "BrokerState": "RUNNING",
    ...
}
```

Per accedere alla console di gestione RabbitMQ utilizzando OAuth2, l'endpoint del broker deve essere aggiunto come URL di callback valido nel client dell'app Amazon Cognito corrispondente. Per ulteriori informazioni, consulta la Fase 5 della configurazione del nostro stack Amazon Cognito CDK di esempio.

6. Verifica l'autenticazione e l'autorizzazione OAuth 2.0 con lo script seguente. `perf-test.sh`

   Usa questo script bash per testare la connettività al tuo broker Amazon MQ for RabbitMQ. Questo script ottiene un token da Amazon Cognito e verifica se la connessione è stata configurata correttamente. Se è configurato correttamente, vedrai il tuo broker pubblicare e consumare messaggi.

   Se ricevi un `ACCESS_REFUSED` errore, puoi risolvere i problemi delle impostazioni di configurazione utilizzando CloudWatch i log del tuo broker. Puoi trovare il link per il gruppo di CloudWatch log del tuo broker nella console Amazon MQ.

   In questo script, dovrai fornire i seguenti valori:

   - `CLIENT_IDeCLIENT_SECRET`: puoi trovare questi valori nella pagina App client della console Amazon Cognito.
   - Dominio Cognito: puoi trovarlo nella console Amazon Cognito. In Branding, scegli Dominio. Nella pagina Dominio, puoi trovare questo valore nella sezione Server di risorse.
   - Endpoint del broker Amazon MQ: puoi trovare questo valore in Connessioni nella pagina dei dettagli del broker della console Amazon MQ.

   ```
   #! /bin/bash
   set -e

   # Client information
   ```

```
## FIXME: Update this value with the client ID and secret of your confidential
 application client
CLIENT_ID=${RabbitMqOAuth2TestStack.AmqpAppClientId}
CLIENT_SECRET=${RabbitMqOAuth2TestStack.AmqpAppClientSecret}

# FIXME: Update this value with the domain of your Amazon Cognito user pool
RESPONSE=$(curl -X POST ${RabbitMqOAuth2TestStack.TokenEndpoint} \
                -H "Content-Type: application/x-www-form-urlencoded" \
                -d
 "grant_type=client_credentials&client_id=${CLIENT_ID}&client_secret=${CLIENT_SECRET}&scope
configure:all rabbitmq/read:all rabbitmq/tag:administrator rabbitmq/write:all")


# Extract the access_token from the response.
# This token will be passed in the password field when connecting to the broker.
# Note that the username is left blank, the field is ignored by the plugin.
BROKER_PASSWORD=$(echo ${RESPONSE} | jq -r '.access_token')

# FIXME: Update this value with the endpoint of your broker. For
 example, b-89424106-7e0e-4abe-8e98-8de0dada7630.mq.us-east-1.on.aws.
BROKER_DNS=<broker_dns>
CONNECTION_STRING=amqps://:${BROKER_PASSWORD}@${BROKER_DNS}:5671

# Produce/consume messages using the above connection string
QUEUES_COUNT=1
PRODUCERS_COUNT=1
CONSUMERS_COUNT=1
PRODUCER_RATE=1

docker run -it --rm --ulimit nofile=40960:40960 pivotalrabbitmq/perf-test:latest \
    --queue-pattern 'test-queue-%d' --queue-pattern-from 1 --queue-pattern-to
 $QUEUES_COUNT \
    --producers $PRODUCERS_COUNT --consumers $CONSUMERS_COUNT \
    --id "test${QUEUES_COUNT}q${PRODUCERS_COUNT}p${CONSUMERS_COUNT}c
${PRODUCER_RATE}r" \
    --uri ${CONNECTION_STRING} \
    --flag persistent --rate $PRODUCER_RATE
```

Configurazione OAuth 2.0 e autenticazione semplice con Amazon Cognito

Quando crei un broker con autenticazione OAuth 2.0, puoi specificare uno dei seguenti metodi di autenticazione:

- OAuth Solo 2.0: per utilizzare questo metodo, non fornire nome utente e password durante la creazione del broker. La procedura precedente mostra come utilizzare solo il metodo di autenticazione OAuth 2.0.
- Autenticazione OAuth 2.0 e semplice: per utilizzare questo metodo, fornisci un nome utente e una password durante la creazione del broker. Inoltre, `auth_backends.2 = internal` aggiungetelo alla configurazione del broker, come illustrato nella procedura seguente.

Nella procedura seguente, assicuratevi di sostituire i valori segnaposto, ad esempio *<ConfigurationId>* e*<Revision>*, con i valori effettivi.

1. Per utilizzare entrambi i metodi di autenticazione, create la configurazione del broker, come illustrato nell'esempio seguente.

```
auth_backends.1 = oauth2
auth_backends.2 = internal

# FIXME: Update this value with the token signing key URL of your Amazon Cognito
 user pool
auth_oauth2.jwks_url = ${RabbitMqOAuth2TestStack.JwksUri}
auth_oauth2.resource_server_id = rabbitmq
auth_oauth2.verify_aud = false

auth_oauth2.scope_prefix = rabbitmq/
auth_oauth2.scope_aliases.1.alias = rabbitmq/read:all
auth_oauth2.scope_aliases.1.scope = rabbitmq/read:*/*
auth_oauth2.scope_aliases.2.alias = rabbitmq/write:all
auth_oauth2.scope_aliases.2.scope = rabbitmq/write:*/*
auth_oauth2.scope_aliases.3.alias = rabbitmq/configure:all
auth_oauth2.scope_aliases.3.scope = rabbitmq/configure:*/*
```

   Questa configurazione utilizza alias di ambito per mappare gli ambiti definiti in Amazon Cognito a ambiti compatibili con RabbitMQ.

2. Crea un broker che utilizzi entrambi i metodi di autenticazione, come mostrato nell'esempio seguente.

```
aws mq create-broker \
 --broker-name "rabbitmq-oauth2-broker-with-internal-user" \
 --engine-type "RABBITMQ" \
 --engine-version "3.13" \
```

```
--host-instance-type "mq.m7g.large" \
--deployment-mode "CLUSTER_MULTI_AZ" \
--logs '{"General": true}' \
--publicly-accessible \
--configuration '{"Id": "<ConfigurationId>","Revision": <Revision>}' \
--users '[{"Username":"<myUser>","Password":"<myPassword11>"}]'
```

3. Verifica che lo stato del broker e la configurazione per l'impostazione del metodo di autenticazione abbiano avuto esito positivo, come descritto nei passaggi 5 e 6 della Configurazione dell'autenticazione OAuth 2.0 con Amazon Cognito procedura.

Utilizzo dell'autenticazione e dell'autorizzazione IAM per Amazon MQ for RabbitMQ

La procedura seguente mostra come abilitare l'autenticazione e l'autorizzazione AWS IAM per un broker Amazon MQ for RabbitMQ. Dopo aver abilitato IAM, gli utenti possono autenticarsi utilizzando le credenziali AWS IAM per accedere all'API di gestione RabbitMQ e connettersi tramite AMQP. Per i dettagli su come funziona l'autenticazione IAM con Amazon MQ for RabbitMQ, consulta. the section called “Autenticazione e autorizzazione IAM”

Prerequisiti

- AWS credenziali di amministratore per l' AWS account proprietario del broker Amazon MQ for RabbitMQ
- Un ambiente shell configurato con queste credenziali di amministratore (utilizzando profili AWS CLI o variabili di ambiente)
- AWS CLI installata e configurata
- `jq`processore JSON da riga di comando installato
- `curl`strumento da riga di comando installato

Configurazione dell'autenticazione e dell'autorizzazione IAM tramite AWS CLI

1. Imposta le variabili di ambiente

   Imposta le variabili di ambiente richieste per il tuo broker:

```
export AWS_DEFAULT_REGION=<region>
export BROKER_ID=<broker-id>
```

2. Abilita i token JWT in uscita

   Abilita la federazione delle identità web in uscita per il tuo account: AWS

```
ISSUER_IDENTIFIER=$(aws iam enable-outbound-web-identity-federation --query
 'IssuerIdentifier' --output text)
echo $ISSUER_IDENTIFIER
```

   L'output mostra un URL identificativo dell'emittente univoco per il tuo account nel formato. `https://<id>.tokens.sts.global.api.aws`

3. Crea il documento relativo alla policy IAM

   Crea un documento di policy che conceda le autorizzazioni per ottenere token di identità web:

```
cat > policy.json << 'EOF'
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "sts:GetWebIdentityToken",
                "sts:TagGetWebIdentityToken"
            ],
            "Resource": "*"
        }
    ]
}
EOF
```

4. Crea la politica di fiducia

   Recupera l'identità del chiamante e crea un documento sulla politica di fiducia:

```
CALLER_ARN=$(aws sts get-caller-identity --query Arn --output text)
cat > trust-policy.json << EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "$CALLER_ARN"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
EOF
```

5. Crea il ruolo IAM

   Crea il ruolo IAM e allega la policy:

```
aws iam create-role --role-name RabbitMqAdminRole --assume-role-policy-document
 file://trust-policy.json
aws iam put-role-policy --role-name RabbitMqAdminRole --policy-name
 RabbitMqAdminRolePolicy --policy-document file://policy.json
```

6. Configura le impostazioni di OAuth2 RabbitMQ

   Crea un file di configurazione RabbitMQ con le impostazioni di OAuth2 autenticazione e autorizzazione:

```
cat > rabbitmq.conf << EOF
auth_backends.1 = oauth2
auth_backends.2 = internal

auth_oauth2.jwks_url = ${ISSUER_IDENTIFIER}/.well-known/jwks.json
auth_oauth2.resource_server_id = rabbitmq
auth_oauth2.scope_prefix = rabbitmq/
```

```
auth_oauth2.additional_scopes_key = sub
auth_oauth2.scope_aliases.1.alias = arn:aws:iam::$(aws sts get-caller-identity --
query Account --output text):role/RabbitMqAdminRole
auth_oauth2.scope_aliases.1.scope = rabbitmq/tag:administrator rabbitmq/read:*/*
 rabbitmq/write:*/* rabbitmq/configure:*/*
auth_oauth2.https.hostname_verification = wildcard

management.oauth_enabled = true
EOF
```

7. Aggiorna la configurazione del broker

   Applica la nuova configurazione al tuo broker:

```
# Retrieve the configuration ID
CONFIG_ID=$(aws mq describe-broker --broker-id $BROKER_ID --query
 'Configurations[0].Id' --output text)

# Create a new configuration revision
REVISION=$(aws mq update-configuration --configuration-id $CONFIG_ID --data "$(cat
 rabbitmq.conf | base64 --wrap=0)" --query 'LatestRevision.Revision' --output text)

# Apply the configuration to the broker
aws mq update-broker --broker-id $BROKER_ID --configuration Id=$CONFIG_ID,Revision=
$REVISION

# Reboot the broker to apply changes
aws mq reboot-broker --broker-id $BROKER_ID
```

   Attendi il ripristino dello stato di broker `RUNNING` prima di procedere alla fase successiva.

8. Ottieni un token JWT

   Assumi il ruolo IAM e ottieni un token di identità web:

```
# Assume the RabbitMqAdminRole
```

```
ROLE_CREDS=$(aws sts assume-role --role-arn arn:aws:iam::$(aws sts get-caller-
identity --query Account --output text):role/RabbitMqAdminRole --role-session-name
 rabbitmq-session)

# Configure the session with temporary credentials
export AWS_ACCESS_KEY_ID=$(echo "$ROLE_CREDS" | jq -r '.Credentials.AccessKeyId')
export AWS_SECRET_ACCESS_KEY=$(echo "$ROLE_CREDS" | jq -r
 '.Credentials.SecretAccessKey')
export AWS_SESSION_TOKEN=$(echo "$ROLE_CREDS" | jq -r '.Credentials.SessionToken')

# Obtain the web identity token
TOKEN_RESPONSE=$(aws sts get-web-identity-token \
    --audience "rabbitmq" \
    --signing-algorithm ES384 \
    --duration-seconds 300 \
    --tags Key=scope,Value="rabbitmq/tag:administrator")

# Extract the token
TOKEN=$(echo "$TOKEN_RESPONSE" | jq -r '.WebIdentityToken')
```

9. Accedi all'API di gestione RabbitMQ

   Usa il token JWT per accedere all'API di gestione RabbitMQ:

   ```
   BROKER_URL=<broker-id>.mq.<region>.on.aws

   curl -u ":$TOKEN" \
       -X GET https://${BROKER_URL}/api/overview \
       -H "Content-Type: application/json"
   ```

   Una risposta riuscita conferma che l'autenticazione IAM funziona correttamente. La risposta contiene informazioni generali sul broker in formato JSON.

10. Connect tramite AMQP utilizzando il token JWT

    Testa la connettività AMQP utilizzando il token JWT con lo strumento perf-test:

    ```
    BROKER_DNS=<broker-endpoint>
    CONNECTION_STRING=amqps://:${TOKEN}@${BROKER_DNS}:5671
    ```

```
docker run -it --rm --ulimit nofile=40960:40960 pivotalrabbitmq/perf-test:latest \
    --queue-pattern 'test-queue-%d' --queue-pattern-from 1 --queue-pattern-to 1 \
    --producers 1 --consumers 1 \
    --uri ${CONNECTION_STRING} \
    --flag persistent --rate 1
```

Se ricevi un `ACCESS_REFUSED` errore, puoi risolvere i problemi delle impostazioni di configurazione utilizzando i log del tuo broker. CloudWatch Puoi trovare il link per il gruppo CloudWatch Logs log per il tuo broker nella console Amazon MQ.

Utilizzo dell'autenticazione e dell'autorizzazione LDAP per Amazon MQ for RabbitMQ

Questo tutorial descrive come configurare l'autenticazione e l'autorizzazione LDAP per i broker Amazon MQ for RabbitMQ che utilizzano. AWS Managed Microsoft AD

In questa pagina

- Prerequisiti per configurare l'autenticazione e l'autorizzazione LDAP
- Configurazione di LDAP in RabbitMQ tramite CLI AWS

Prerequisiti per configurare l'autenticazione e l'autorizzazione LDAP

Puoi configurare le AWS risorse richieste in questo tutorial distribuendo lo stack AWS CDK per l'integrazione LDAP di Amazon MQ for RabbitMQ con. AWS Managed Microsoft AD

Questo stack CDK crea automaticamente tutte le AWS risorse necessarie AWS Managed Microsoft AD, tra cui utenti e gruppi LDAP, Network Load Balancer, certificati e ruoli IAM. Consulta il pacchetto README per un elenco completo delle risorse create dallo stack.

Se stai configurando le risorse manualmente anziché utilizzare lo stack CDK, assicurati di disporre dell'infrastruttura equivalente prima di configurare LDAP sui tuoi broker Amazon MQ for RabbitMQ.

Prerequisito per configurare Amazon MQ

AWS Versione CLI >= 2.28.23 per rendere opzionale l'aggiunta di un nome utente e una password durante la creazione del broker.

Configurazione di LDAP in RabbitMQ tramite CLI AWS

Questa procedura utilizza la AWS CLI per creare e configurare le risorse necessarie. Nella procedura seguente, assicuratevi di sostituire i valori segnaposto, come ConfigurationID e Revision, con i relativi valori <c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca> effettivi<2>.

1. Creare una nuova configurazione utilizzando il comando `create-configuration` AWS CLI, come illustrato nell'esempio seguente.

```
aws mq create-configuration \
  --name "rabbitmq-ldap-config" \
  --engine-type "RABBITMQ" \
  --engine-version "3.13"
```

   Questo comando restituisce una risposta simile all'esempio seguente.

```
{
"Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-fa3390a5-7e01-4559-ae0c-
eb15b38b22ca",
    "AuthenticationStrategy": "simple",
    "Created": "2025-07-17T16:03:01.759943+00:00",
    "Id": "c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
    "LatestRevision": {
"Created": "2025-07-17T16:03:01.759000+00:00",
    "Description": "Auto-generated default for rabbitmq-ldap-config on RabbitMQ
 3.13",
    "Revision": 1
    },
    "Name": "rabbitmq-ldap-config"
}
```

2. Create un file di configurazione chiamato `rabbitmq.conf` a utilizzare LDAP come metodo di autenticazione e autorizzazione, come illustrato nell'esempio seguente. Sostituisci tutti i valori segnaposto nel modello (contrassegnati con`${RabbitMqLdapTestStack.*}`) con i valori effettivi degli output dello stack dei AWS CDK prerequisiti implementati o dell'infrastruttura equivalente.

```
auth_backends.1 = ldap

# LDAP authentication settings - For more information,
# see https://www.rabbitmq.com/docs/ldap#basic

# FIXME: Replace the ${RabbitMqLdapTestStack.*} placeholders with actual values
# from your deployed prerequisite CDK stack outputs.
auth_ldap.servers.1 = ${RabbitMqLdapTestStack.NlbDnsName}
auth_ldap.dn_lookup_bind.user_dn = ${RabbitMqLdapTestStack.DnLookupUserDn}
auth_ldap.dn_lookup_base = ${RabbitMqLdapTestStack.DnLookupBase}
auth_ldap.dn_lookup_attribute = ${RabbitMqLdapTestStack.DnLookupAttribute}
auth_ldap.port = 636
auth_ldap.use_ssl = true
auth_ldap.ssl_options.verify = verify_peer
auth_ldap.log = network

# AWS integration for secure credential retrieval
# - see: https://github.com/amazon-mq/rabbitmq-aws
# The aws plugin allows RabbitMQ to securely retrieve credentials and certificates
# from AWS services.

# Replace the ${RabbitMqLdapTestStack.*} placeholders with actual ARN values
# from your deployed prerequisite CDK stack outputs.
aws.arns.auth_ldap.ssl_options.cacertfile = ${RabbitMqLdapTestStack.CaCertArn}
aws.arns.auth_ldap.dn_lookup_bind.password =
 ${RabbitMqLdapTestStack.DnLookupUserPasswordArn}
aws.arns.assume_role_arn = ${RabbitMqLdapTestStack.AmazonMqAssumeRoleArn}

# LDAP authorization queries - For more information,
# see: https://www.rabbitmq.com/docs/ldap#authorisation

# FIXME: Replace the ${RabbitMqLdapTestStack.*} placeholders with actual group DN
# values from your deployed prerequisite CDK stack outputs
# Uses Active Directory groups created by the prerequisite CDK stack
auth_ldap.queries.tags = '''
[{administrator, {in_group,
 "${RabbitMqLdapTestStack.RabbitMqAdministratorsGroupDn}"}},
{management,    {in_group,
 "${RabbitMqLdapTestStack.RabbitMqMonitoringUsersGroupDn}"}}]
'''

# FIXME: This provides all authenticated users access to all vhosts
```

```
# - update to restrict access as required
auth_ldap.queries.vhost_access = '''
{constant, true}
'''

# FIXME: This provides all authenticated users full access to all
# queues and exchanges - update to restrict access as required
auth_ldap.queries.resource_access = '''
{for, [    {permission, configure, {constant, true}},
     {permission, write,
      {for, [{resource, queue,    {constant, true}},
             {resource, exchange, {constant, true}}]}},
     {permission, read,
      {for, [{resource, exchange, {constant, true}},
             {resource, queue,    {constant, true}}]}}
    ]
}
'''

# FIXME: This provides all authenticated users access to all topics
# - update to restrict access as required
auth_ldap.queries.topic_access = '''
{for, [{permission, write, {constant, true}},
     {permission, read,  {constant, true}}
    ]
}
'''
```

3. Aggiornate la configurazione utilizzando il comando `update-configuration` AWS CLI come illustrato nell'esempio seguente. In questo comando, aggiungi l'ID di configurazione ricevuto nella risposta del passaggio 1 di questa procedura. Ad esempio, `c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca`.

```
aws mq update-configuration \
  --configuration-id "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>" \
  --data "$(cat rabbitmq.conf | base64 --wrap=0)"
```

Questo comando restituisce una risposta simile all'esempio seguente.

```
{
    "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-b600ac8e-8183-4f74-
a713-983e59f30e3d",
    "Created": "2025-07-17T16:57:04.520931+00:00",
    "Id": "c-b600ac8e-8183-4f74-a713-983e59f30e3d",
    "LatestRevision": {
        "Created": "2025-07-17T16:57:39.172000+00:00",
        "Revision": 2
    },
    "Name": "rabbitmq-ldap-config",
    "Warnings": []
}
```

4. Crea un broker con la configurazione LDAP creata nel passaggio 2 di questa procedura. A tale scopo, utilizzate il comando `create-broker` AWS CLI come illustrato nell'esempio seguente. In questo comando, fornite l'ID di configurazione e il numero di revisione ottenuti rispettivamente nelle risposte dei passaggi 1 e 2. Ad esempio `c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca` e 2.

```
aws mq create-broker \
 --broker-name "rabbitmq-ldap-test-1" \
 --engine-type "RABBITMQ" \
 --engine-version "3.13" \
 --host-instance-type "mq.m7g.large" \
 --deployment-mode "CLUSTER_MULTI_AZ" \
 --logs '{"General": true}' \
 --publicly-accessible \
 --configuration '{"Id": "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>","Revision":
 <2>}'
```

Questo comando restituisce una risposta simile all'esempio seguente.

```
{
    "BrokerArn": "arn:aws:mq:us-west-2:123456789012:broker:rabbitmq-ldap-
broker:b-2a1b5133-a10c-49d2-879b-8c176c34cf73",
    "BrokerId": "b-2a1b5133-a10c-49d2-879b-8c176c34cf73"
```

```
}
```

> **Restrizione dei nomi dei broker**
>
> Il ruolo IAM creato dallo stack CDK prerequisito limita inizialmente i nomi dei broker. `rabbitmq-ldap-test` Assicurati che il nome del tuo broker segua questo schema o il ruolo IAM non sarà autorizzato ad assumere il ruolo per la risoluzione ARN.

5. Verifica che lo stato del broker passi da `CREATION_IN_PROGRESS` a`RUNNING`, utilizzando il comando `describe-broker` AWS CLI come mostrato nell'esempio seguente. In questo comando, fornisci l'ID del broker che hai ottenuto nel risultato del passaggio precedente. Ad esempio,. `b-2a1b5133-a10c-49d2-879b-8c176c34cf73`

```
aws mq describe-broker \
 --broker-id "<b-2a1b5133-a10c-49d2-879b-8c176c34cf73>"
```

Questo comando restituisce una risposta simile all'esempio seguente. La risposta seguente è una versione abbreviata dell'output completo restituito dal `describe-broker` comando. Questa risposta mostra lo stato del broker e la strategia di autenticazione utilizzata per proteggere il broker. In questo caso, la strategia di `config_managed` autenticazione indica che il broker utilizza il metodo di autenticazione LDAP.

```
{
"AuthenticationStrategy": "config_managed",
    ...,
    "BrokerState": "RUNNING",
    ...
}
```

6. Convalida l'accesso a RabbitMQ utilizzando uno degli utenti di test creati dallo stack CDK prerequisito

```
# FIXME: Replace ${RabbitMqLdapTestStack.ConsoleUserPasswordArn} with the actual
 ARN from your deployed prerequisite CDK stack outputs
CONSOLE_PASSWORD=$(aws secretsmanager get-secret-value \
  --secret-id ${RabbitMqLdapTestStack.ConsoleUserPasswordArn} \
  --query 'SecretString' --output text)

# FIXME: Replace BrokerConsoleURL with the actual ConsoleURL retrieved by
# calling describe-broker for the broker created above
# Call management API /api/overview (should succeed)
curl -u RabbitMqConsoleUser:$CONSOLE_PASSWORD \
  https://${BrokerConsoleURL}/api/overview

# Try to create a user (should fail - console user only has monitoring permissions)
curl -u RabbitMqConsoleUser:$CONSOLE_PASSWORD \
  -X PUT https://${BrokerConsoleURL}/api/users/testuser \
  -H "Content-Type: application/json" \
  -d '{"password":"testpass","tags":"management"}'
```

Utilizzo dell'autenticazione e dell'autorizzazione HTTP per Amazon MQ for RabbitMQ

Questo tutorial descrive come configurare l'autenticazione e l'autorizzazione HTTP per i broker Amazon MQ for RabbitMQ utilizzando un server HTTP esterno.

> **Note**
>
> Il plug-in di autenticazione HTTP è disponibile solo per Amazon MQ for RabbitMQ versione 4 e successive.

In questa pagina

- Prerequisiti per configurare l'autenticazione e l'autorizzazione HTTP
- Configurazione dell'autenticazione HTTP in RabbitMQ tramite CLI AWS

Prerequisiti per configurare l'autenticazione e l'autorizzazione HTTP

Puoi configurare le AWS risorse richieste in questo tutorial distribuendo lo stack AWS CDK per l'integrazione dell'autenticazione HTTP di Amazon MQ for RabbitMQ.

Questo stack CDK crea automaticamente tutte le AWS risorse necessarie, inclusi il server di autenticazione HTTP, i certificati e i ruoli IAM. Vedi il pacchetto README per un elenco completo delle risorse create dallo stack.

Se stai configurando le risorse manualmente anziché utilizzare lo stack CDK, assicurati di disporre dell'infrastruttura equivalente prima di configurare l'autenticazione HTTP sui tuoi broker Amazon MQ for RabbitMQ.

Prerequisito per configurare Amazon MQ

AWS Versione CLI >= 2.28.23 per rendere opzionale l'aggiunta di un nome utente e una password durante la creazione del broker.

Configurazione dell'autenticazione HTTP in RabbitMQ tramite CLI AWS

Questa procedura utilizza la AWS CLI per creare e configurare le risorse necessarie. Nella procedura seguente, assicuratevi di sostituire i valori segnaposto con i valori effettivi.

1. Creare una nuova configurazione utilizzando il comando `create-configuration` AWS CLI, come illustrato nell'esempio seguente.

```
aws mq create-configuration \
  --name "rabbitmq-http-config" \
  --engine-type "RABBITMQ" \
  --engine-version "4.2"
```

Questo comando restituisce una risposta simile all'esempio seguente.

```
{
    "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-fa3390a5-7e01-4559-
ae0c-eb15b38b22ca",
    "AuthenticationStrategy": "simple",
    "Created": "2025-07-17T16:03:01.759943+00:00",
    "Id": "c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
    "LatestRevision": {
        "Created": "2025-07-17T16:03:01.759000+00:00",
        "Description": "Auto-generated default for rabbitmq-http-config on RabbitMQ
 4.2",
        "Revision": 1
```

```
    },
    "Name": "rabbitmq-http-config"
}
```

2. Create un file di configurazione chiamato `rabbitmq.conf` a utilizzare HTTP come metodo di autenticazione e autorizzazione, come illustrato nell'esempio seguente. Sostituisci tutti i valori segnaposto nel modello (contrassegnati con`${...}`) con i valori effettivi degli output dello stack dei AWS CDK prerequisiti implementati o dell'infrastruttura equivalente.

```
auth_backends.1 = cache
auth_backends.2 = http
auth_cache.cached_backend = http

# HTTP authentication settings
# For more information, see https://github.com/rabbitmq/rabbitmq-auth-backend-http

# FIXME: Replace the ${...} placeholders with actual values
# from your deployed prerequisite CDK stack outputs.
auth_http.http_method = post
auth_http.user_path = ${HttpServerUserPath}
auth_http.vhost_path = ${HttpServerVhostPath}
auth_http.resource_path = ${HttpServerResourcePath}
auth_http.topic_path = ${HttpServerTopicPath}

# TLS/HTTPS configuration
auth_http.ssl_options.verify = verify_peer
auth_http.ssl_options.sni = test.amazonaws.com

# AWS integration for secure credential retrieval
# For more information, see https://github.com/amazon-mq/rabbitmq-aws

# Replace the ${...} placeholders with actual ARN values
# from your deployed prerequisite CDK stack outputs.
aws.arns.assume_role_arn = ${AmazonMqAssumeRoleArn}
aws.arns.auth_http.ssl_options.cacertfile = ${CaCertArn}
```

3. Aggiorna la configurazione utilizzando il `update-configuration` AWS comando CLI. Usa l'ID di configurazione del passaggio 3.

```
aws mq update-configuration \
  --configuration-id "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>" \
  --data "$(cat rabbitmq.conf | base64 --wrap=0)"
```

Questo comando restituisce una risposta simile all'esempio seguente.

```
{
    "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-fa3390a5-7e01-4559-
ae0c-eb15b38b22ca",
    "Created": "2025-07-17T16:57:04.520931+00:00",
    "Id": "c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
    "LatestRevision": {
        "Created": "2025-07-17T16:57:39.172000+00:00",
        "Revision": 2
    },
    "Name": "rabbitmq-http-config",
    "Warnings": []
}
```

4. Crea un broker con la configurazione HTTP. Utilizza l'ID di configurazione e il numero di revisione dei passaggi precedenti.

```
aws mq create-broker \
  --broker-name "rabbitmq-http-test-1" \
  --engine-type "RABBITMQ" \
  --engine-version "4.2" \
  --host-instance-type "mq.m7g.large" \
  --deployment-mode "SINGLE_INSTANCE" \
  --logs '{"General": true}' \
  --publicly-accessible \
  --configuration '{"Id": "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>","Revision":
 <2>}'
```

Questo comando restituisce una risposta simile all'esempio seguente.

```
{
```

```
    "BrokerArn": "arn:aws:mq:us-west-2:123456789012:broker:rabbitmq-http-
test-1:b-2a1b5133-a10c-49d2-879b-8c176c34cf73",
    "BrokerId": "b-2a1b5133-a10c-49d2-879b-8c176c34cf73"
}
```

5. Verifica che lo stato del broker passi da `CREATION_IN_PROGRESS` a`RUNNING`, utilizzando il comando `describe-broker` AWS CLI.

```
aws mq describe-broker \
  --broker-id "<b-2a1b5133-a10c-49d2-879b-8c176c34cf73>"
```

Questo comando restituisce una risposta simile all'esempio seguente. La strategia di `config_managed` autenticazione indica che il broker utilizza il metodo di autenticazione HTTP.

```
{
    "AuthenticationStrategy": "config_managed",
    ...,
    "BrokerState": "RUNNING",
    ...
}
```

6. Convalida l'accesso a RabbitMQ utilizzando uno degli utenti di test creati dallo stack CDK prerequisito

```
# FIXME: Replace ${RabbitMqHttpAuthElbStack.ConsoleUserPasswordArn} with the actual
 ARN from your deployed prerequisite CDK stack outputs
CONSOLE_PASSWORD=$(aws secretsmanager get-secret-value \
  --secret-id ${RabbitMqHttpAuthElbStack.ConsoleUserPasswordArn} \
  --query 'SecretString' --output text)

# FIXME: Replace BrokerConsoleURL with the actual ConsoleURL retrieved by
# calling describe-broker for the broker created above
# Call management API /api/overview (should succeed)
curl -u RabbitMqConsoleUser:$CONSOLE_PASSWORD \
  https://${BrokerConsoleURL}/api/overview
```

```
# Try to create a vhost (should fail - console user only has management
 permissions)
curl -u RabbitMqConsoleUser:$CONSOLE_PASSWORD \
  -X PUT https://${BrokerConsoleURL}/api/vhosts/test-vhost \
  -H "Content-Type: application/json" \
  -d '{}'
```

Utilizzo dell'autenticazione del certificato SSL per Amazon MQ for RabbitMQ

Questo tutorial descrive come configurare l'autenticazione del certificato SSL per i broker Amazon MQ for RabbitMQ utilizzando un'autorità di certificazione privata.

> **Note**
>
> Il plug-in di autenticazione del certificato SSL è disponibile solo per Amazon MQ for RabbitMQ versione 4 e successive.

In questa pagina

- Prerequisiti per configurare l'autenticazione del certificato SSL
- Configurazione dell'autenticazione del certificato SSL in RabbitMQ tramite CLI AWS

Prerequisiti per configurare l'autenticazione del certificato SSL

L'autenticazione dei certificati SSL utilizza il protocollo TLS reciproco (mTLS) per autenticare i client utilizzando i certificati X.509. Puoi configurare le AWS risorse richieste in questo tutorial distribuendo lo stack AWS CDK per l'integrazione con Amazon MQ for RabbitMQ MTL.

Questo stack CDK crea automaticamente tutte le AWS risorse necessarie, tra cui l'autorità di certificazione, i certificati client e i ruoli IAM. Consulta il pacchetto README per un elenco completo delle risorse create dallo stack.

> **Note**
>
> Prima di distribuire lo stack CDK, imposta la variabile di ambiente. `RABBITMQ_TEST_USER_NAME` Questo valore verrà utilizzato come nome comune (CN) nel

certificato client e deve corrispondere al nome utente utilizzato nei passaggi del tutorial. Ad esempio: `export RABBITMQ_TEST_USER_NAME="myuser"`

Se stai configurando le risorse manualmente anziché utilizzare lo stack CDK, assicurati di disporre dell'infrastruttura equivalente prima di configurare l'autenticazione del certificato SSL sui tuoi broker Amazon MQ for RabbitMQ.

Prerequisito per configurare Amazon MQ

AWS Versione CLI >= 2.28.23 per rendere opzionale l'aggiunta di un nome utente e una password durante la creazione del broker.

Configurazione dell'autenticazione del certificato SSL in RabbitMQ tramite CLI AWS

Questa procedura utilizza la AWS CLI per creare e configurare le risorse necessarie. Nella procedura seguente, assicuratevi di sostituire i valori segnaposto, come ConfigurationID e Revision, con i relativi valori `<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>` effettivi<2>.

1. Creare una nuova configurazione utilizzando il comando `create-configuration` AWS CLI, come illustrato nell'esempio seguente.

```
aws mq create-configuration \
  --name "rabbitmq-ssl-config" \
  --engine-type "RABBITMQ" \
  --engine-version "4.2"
```

Questo comando restituisce una risposta simile all'esempio seguente.

```
{
    "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-fa3390a5-7e01-4559-
ae0c-eb15b38b22ca",
    "AuthenticationStrategy": "simple",
    "Created": "2025-07-17T16:03:01.759943+00:00",
    "Id": "c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
    "LatestRevision": {
        "Created": "2025-07-17T16:03:01.759000+00:00",
```

```
        "Description": "Auto-generated default for rabbitmq-ssl-config on RabbitMQ
 4.2",
        "Revision": 1
    },
    "Name": "rabbitmq-ssl-config"
}
```

2. Crea un file di configurazione chiamato `rabbitmq.conf` per utilizzare l'autenticazione del certificato SSL, come illustrato nell'esempio seguente. Sostituisci tutti i valori segnaposto nel modello (contrassegnati con`${...}`) con i valori effettivi degli output dello stack dei AWS CDK prerequisiti implementati o dell'infrastruttura equivalente.

```
auth_mechanisms.1 = EXTERNAL
ssl_cert_login_from = common_name

auth_backends.1 = internal

# Reject if no client cert
ssl_options.verify = verify_peer
ssl_options.fail_if_no_peer_cert = true

# AWS integration for secure credential retrieval
# For more information, see https://github.com/amazon-mq/rabbitmq-aws

# FIXME: Replace the ${...} placeholders with actual ARN values
# from your deployed prerequisite CDK stack outputs.
aws.arns.assume_role_arn = ${AmazonMqAssumeRoleArn}
aws.arns.ssl_options.cacertfile = ${CaCertArn}
```

3. Aggiornate la configurazione utilizzando il comando `update-configuration` AWS CLI come illustrato nell'esempio seguente. In questo comando, aggiungi l'ID di configurazione ricevuto nella risposta del passaggio 1 di questa procedura. Ad esempio, `c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca`.

```
aws mq update-configuration \
  --configuration-id "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>" \
  --data "$(cat rabbitmq.conf | base64 --wrap=0)"
```

Questo comando restituisce una risposta simile all'esempio seguente.

```
{
    "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-fa3390a5-7e01-4559-
ae0c-eb15b38b22ca",
    "Created": "2025-07-17T16:57:04.520931+00:00",
    "Id": "c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
    "LatestRevision": {
        "Created": "2025-07-17T16:57:39.172000+00:00",
        "Revision": 2
    },
    "Name": "rabbitmq-ssl-config",
    "Warnings": []
}
```

4. Crea un broker con la configurazione di autenticazione del certificato SSL creata nel passaggio 2 di questa procedura. A tale scopo, utilizzate il comando `create-broker` AWS CLI come illustrato nell'esempio seguente. In questo comando, fornite l'ID di configurazione e il numero di revisione ottenuti rispettivamente nelle risposte dei passaggi 1 e 2. Ad esempio `c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca` e 2.

```
aws mq create-broker \
  --broker-name "rabbitmq-ssl-test-1" \
  --engine-type "RABBITMQ" \
  --engine-version "4.2" \
  --host-instance-type "mq.m7g.large" \
  --deployment-mode "SINGLE_INSTANCE" \
  --logs '{"General": true}' \
  --publicly-accessible \
  --configuration '{"Id": "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>","Revision":
 <2>}' \
  --users '[{"Username":"testuser","Password":"testpassword"}]'
```

Questo comando restituisce una risposta simile all'esempio seguente.

```
{
```

```
    "BrokerArn": "arn:aws:mq:us-west-2:123456789012:broker:rabbitmq-ssl-
test-1:b-2a1b5133-a10c-49d2-879b-8c176c34cf73",
    "BrokerId": "b-2a1b5133-a10c-49d2-879b-8c176c34cf73"
}
```

5. Verifica che lo stato del broker passi da `CREATION_IN_PROGRESS` a`RUNNING`, utilizzando il comando `describe-broker` AWS CLI come mostrato nell'esempio seguente. In questo comando, fornisci l'ID del broker che hai ottenuto nel risultato del passaggio precedente. Ad esempio, `b-2a1b5133-a10c-49d2-879b-8c176c34cf73`.

```
aws mq describe-broker \
  --broker-id "<b-2a1b5133-a10c-49d2-879b-8c176c34cf73>"
```

   Questo comando restituisce una risposta simile all'esempio seguente. La risposta seguente è una versione abbreviata dell'output completo restituito dal `describe-broker` comando. Questa risposta mostra lo stato del broker e la strategia di autenticazione utilizzata per proteggere il broker. In questo caso, la strategia di `config_managed` autenticazione indica che il broker utilizza il metodo di autenticazione del certificato SSL.

```
{
    "AuthenticationStrategy": "config_managed",
    ...,
    "BrokerState": "RUNNING",
    ...
}
```

6. Verifica l'autenticazione del certificato SSL con il seguente `ssl.sh` script.

   Usa questo script bash per testare la connettività al tuo broker Amazon MQ for RabbitMQ. Questo script utilizza il certificato del client per l'autenticazione e verifica se la connessione è stata configurata correttamente. Se è configurato correttamente, vedrai il tuo broker pubblicare e consumare messaggi.

Se ricevi un `ACCESS_REFUSED` errore, puoi risolvere i problemi delle impostazioni di configurazione utilizzando CloudWatch i log del tuo broker. Puoi trovare il link per il gruppo di CloudWatch log del tuo broker nella console Amazon MQ.

In questo script, dovrai fornire i seguenti valori:

- `USERNAME`: Il nome comune (CN) del certificato client.
- `CLIENT_KEYSTORE`: percorso del file keystore del client (PKCS12 formato). Se hai utilizzato lo stack CDK prerequisito, il percorso predefinito è. `$(pwd)/certs/client-keystore.p12`
- `KEYSTORE_PASSWORD`: Password per il keystore del cliente. Se hai utilizzato lo stack CDK prerequisito, la password predefinita è. `changeit`
- `BROKER_DNS`: Puoi trovare questo valore in Connessioni nella pagina dei dettagli del broker della console Amazon MQ.

```
#! /bin/bash
set -e

# Client information
## FIXME: Update this value with the client ID and secret of your confidential
 application client
USERNAME=<client_cert_common_name>
CLIENT_KEYSTORE=$(pwd)/certs/client-keystore.p12
KEYSTORE_PASSWORD=changeit

BROKER_DNS=<broker_dns>
CONNECTION_STRING=amqps://${BROKER_DNS}:5671

# Produce/consume messages using the above connection string
QUEUES_COUNT=1
PRODUCERS_COUNT=1
CONSUMERS_COUNT=1
PRODUCER_RATE=1

finch run --rm --ulimit nofile=40960:40960 \
    -v ${CLIENT_KEYSTORE}:/certs/client-keystore.p12:ro \
    -e JAVA_TOOL_OPTIONS="-Djavax.net.ssl.keyStore=/certs/client-
keystore.p12 -Djavax.net.ssl.keyStorePassword=${KEYSTORE_PASSWORD} -
Djavax.net.ssl.keyStoreType=PKCS12" \
    pivotalrabbitmq/perf-test:latest \
```

```
    --queue-pattern 'test-queue-cert-%d' --queue-pattern-from 1 --queue-pattern-to
 $QUEUES_COUNT \
    --producers $PRODUCERS_COUNT --consumers $CONSUMERS_COUNT \
    --id "cert-test${QUEUES_COUNT}q${PRODUCERS_COUNT}p${CONSUMERS_COUNT}c
${PRODUCER_RATE}r" \
    --uri ${CONNECTION_STRING} \
    --sasl-external \
    --use-default-ssl-context \
    --flag persistent --rate $PRODUCER_RATE
```

Utilizzo di MTL per AMQP e endpoint di gestione

Questo tutorial descrive come configurare il TLS reciproco (mTLS) per le connessioni client AMQP e l'interfaccia di gestione RabbitMQ utilizzando un'autorità di certificazione privata.

> **Note**
>
> L'uso di autorità di certificazione private per MTL è disponibile solo per Amazon MQ for RabbitMQ versione 4 e successive.

In questa pagina

- Prerequisiti per configurare gli MTL
- Configurazione degli MTL in RabbitMQ tramite CLI AWS

Prerequisiti per configurare gli MTL

Puoi configurare le AWS risorse richieste in questo tutorial distribuendo lo stack AWS CDK per l'integrazione di Amazon MQ for RabbitMQ MTL con.

Questo stack CDK crea automaticamente tutte le AWS risorse necessarie, tra cui l'autorità di certificazione, i certificati client e i ruoli IAM. Consulta il pacchetto README per un elenco completo delle risorse create dallo stack.

Se stai configurando le risorse manualmente anziché utilizzare lo stack CDK, assicurati di disporre dell'infrastruttura equivalente prima di configurare MTL sui tuoi broker Amazon MQ for RabbitMQ.

Prerequisito per configurare Amazon MQ

AWS Versione CLI >= 2.28.23 per rendere opzionale l'aggiunta di un nome utente e una password durante la creazione del broker.

Configurazione degli MTL in RabbitMQ tramite CLI AWS

Questa procedura utilizza la AWS CLI per creare e configurare le risorse necessarie. Nella procedura seguente, assicuratevi di sostituire i valori segnaposto, come ConfigurationID e Revision, con i relativi valori <c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca> effettivi<2>.

1. Creare una nuova configurazione utilizzando il comando `create-configuration` AWS CLI, come illustrato nell'esempio seguente.

```
aws mq create-configuration \
  --name "rabbitmq-mtls-config" \
  --engine-type "RABBITMQ" \
  --engine-version "4.2"
```

   Questo comando restituisce una risposta simile all'esempio seguente.

```
{
    "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-fa3390a5-7e01-4559-
ae0c-eb15b38b22ca",
    "AuthenticationStrategy": "simple",
    "Created": "2025-07-17T16:03:01.759943+00:00",
    "Id": "c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
    "LatestRevision": {
        "Created": "2025-07-17T16:03:01.759000+00:00",
        "Description": "Auto-generated default for rabbitmq-mtls-config on RabbitMQ
 4.2",
        "Revision": 1
    },
    "Name": "rabbitmq-mtls-config"
}
```

2. Create un file di configurazione chiamato `rabbitmq.conf` a configurare MTL per AMQP e gli endpoint di gestione, come illustrato nell'esempio seguente. Sostituisci tutti i valori segnaposto

nel modello (contrassegnati con`${...}`) con i valori effettivi degli output dello stack dei prerequisiti implementati AWS CDK o dell'infrastruttura equivalente.

```
auth_backends.1 = internal

# TLS configuration
ssl_options.verify = verify_peer
ssl_options.fail_if_no_peer_cert = true
management.ssl.verify = verify_peer

# AWS integration for secure credential retrieval
# For more information, see https://github.com/amazon-mq/rabbitmq-aws

# FIXME: Replace the ${...} placeholders with actual ARN values
# from your deployed prerequisite CDK stack outputs.
aws.arns.assume_role_arn = ${AmazonMqAssumeRoleArn}
aws.arns.ssl_options.cacertfile = ${CaCertArn}
aws.arns.management.ssl.cacertfile = ${CaCertArn}
```

3. Aggiornate la configurazione utilizzando il comando `update-configuration` AWS CLI come illustrato nell'esempio seguente. In questo comando, aggiungi l'ID di configurazione ricevuto nella risposta del passaggio 1 di questa procedura. Ad esempio, `c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca`.

```
aws mq update-configuration \
  --configuration-id "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>" \
  --data "$(cat rabbitmq.conf | base64 --wrap=0)"
```

Questo comando restituisce una risposta simile all'esempio seguente.

```
{
    "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-fa3390a5-7e01-4559-
ae0c-eb15b38b22ca",
    "Created": "2025-07-17T16:57:04.520931+00:00",
    "Id": "c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
    "LatestRevision": {
        "Created": "2025-07-17T16:57:39.172000+00:00",
```

```
        "Revision": 2
    },
    "Name": "rabbitmq-mtls-config",
    "Warnings": []
}
```

4. Crea un broker con la configurazione mTLS creata nel passaggio 2 di questa procedura. A tale scopo, utilizzate il comando `create-broker` AWS CLI come illustrato nell'esempio seguente. In questo comando, fornite l'ID di configurazione e il numero di revisione ottenuti rispettivamente nelle risposte dei passaggi 1 e 2. Ad esempio `c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca` e 2.

```
aws mq create-broker \
  --broker-name "rabbitmq-mtls-test-1" \
  --engine-type "RABBITMQ" \
  --engine-version "4.2" \
  --host-instance-type "mq.m7g.large" \
  --deployment-mode "SINGLE_INSTANCE" \
  --logs '{"General": true}' \
  --publicly-accessible \
  --configuration '{"Id": "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>","Revision":
 <2>}' \
  --users '[{"Username":"testuser","Password":"testpassword"}]'
```

Questo comando restituisce una risposta simile all'esempio seguente.

```
{
    "BrokerArn": "arn:aws:mq:us-west-2:123456789012:broker:rabbitmq-mtls-
test-1:b-2a1b5133-a10c-49d2-879b-8c176c34cf73",
    "BrokerId": "b-2a1b5133-a10c-49d2-879b-8c176c34cf73"
}
```

5. Verifica che lo stato del broker passi da `CREATION_IN_PROGRESS` a`RUNNING`, utilizzando il comando `describe-broker` AWS CLI come mostrato nell'esempio seguente. In questo comando, fornisci l'ID del broker che hai ottenuto nel risultato del passaggio precedente. Ad esempio, `b-2a1b5133-a10c-49d2-879b-8c176c34cf73`.

```
aws mq describe-broker \
  --broker-id "<b-2a1b5133-a10c-49d2-879b-8c176c34cf73>"
```

Questo comando restituisce una risposta simile all'esempio seguente. La risposta seguente è una versione abbreviata dell'output completo restituito dal `describe-broker` comando.

```
{
    "AuthenticationStrategy": "simple",
    ...,
    "BrokerState": "RUNNING",
    ...
}
```

6. Verifica l'autenticazione mTLS con lo script seguente`mtls.sh`.

   Usa questo script bash per testare la connettività al tuo broker Amazon MQ for RabbitMQ. Questo script utilizza il certificato client per l'autenticazione e verifica se la connessione è stata configurata correttamente. Se è configurato correttamente, vedrai il tuo broker pubblicare e consumare messaggi.

   Se ricevi un `ACCESS_REFUSED` errore, puoi risolvere i problemi delle impostazioni di configurazione utilizzando CloudWatch i log del tuo broker. Puoi trovare il link per il gruppo di CloudWatch log del tuo broker nella console Amazon MQ.

   In questo script, dovrai fornire i seguenti valori:

   - `USERNAMEePASSWORD`: le credenziali utente di RabbitMQ che hai creato con il broker.
   - `CLIENT_KEYSTORE`: Percorso del file keystore del client (formato). PKCS12 Se hai utilizzato lo stack CDK prerequisito, il percorso predefinito è. `$(pwd)/certs/client-keystore.p12`
   - `KEYSTORE_PASSWORD`: Password per il keystore del cliente. Se hai utilizzato lo stack CDK prerequisito, la password predefinita è. `changeit`
   - `BROKER_DNS`: Puoi trovare questo valore in Connessioni nella pagina dei dettagli del broker della console Amazon MQ.

```
#! /bin/bash
set -e

# Client information
## FIXME: Update this value with the client ID and secret of your confidential
 application client
USERNAME=<testuser>
PASSWORD=<testpassword>
CLIENT_KEYSTORE=$(pwd)/certs/client-keystore.p12
KEYSTORE_PASSWORD=changeit

BROKER_DNS=<broker_dns>
CONNECTION_STRING=amqps://${USERNAME}:${PASSWORD}@${BROKER_DNS}:5671

# Produce/consume messages using the above connection string
QUEUES_COUNT=1
PRODUCERS_COUNT=1
CONSUMERS_COUNT=1
PRODUCER_RATE=1

finch run --rm --ulimit nofile=40960:40960 \
    -v ${CLIENT_KEYSTORE}:/certs/client-keystore.p12:ro \
    -e JAVA_TOOL_OPTIONS="-Djavax.net.ssl.keyStore=/certs/client-
keystore.p12 -Djavax.net.ssl.keyStorePassword=${KEYSTORE_PASSWORD} -
Djavax.net.ssl.keyStoreType=PKCS12" \
    pivotalrabbitmq/perf-test:latest \
    --queue-pattern 'test-queue-cert-%d' --queue-pattern-from 1 --queue-pattern-to
 $QUEUES_COUNT \
    --producers $PRODUCERS_COUNT --consumers $CONSUMERS_COUNT \
    --id "cert-test${QUEUES_COUNT}q${PRODUCERS_COUNT}p${CONSUMERS_COUNT}c
${PRODUCER_RATE}r" \
    --uri ${CONNECTION_STRING} \
    --use-default-ssl-context \
    --flag persistent --rate $PRODUCER_RATE
```

Connessione dell'applicazione JMS

Questo tutorial mostra come connettere la tua applicazione JMS al broker Amazon MQ for RabbitMQ utilizzando il client RabbitMQ JMS. Imparerai come creare un produttore per inviare messaggi e un consumatore per ricevere messaggi dalle code di RabbitMQ.

Prima di iniziare, aggiungi la dipendenza JMS RabbitMQ appropriata al tuo progetto Maven:

Per JMS 1.1 e 2.0:

```
<dependencies>

  <dependency>
    <groupId>com.rabbitmq.jms</groupId>
    <artifactId>rabbitmq-jms</artifactId>
    <version>2.12.0</version>
  </dependency>

</dependencies>
```

Per JMS 3.1:

```
<dependencies>

  <dependency>
    <groupId>com.rabbitmq.jms</groupId>
    <artifactId>rabbitmq-jms</artifactId>
    <version>3.5.0</version>
  </dependency>

</dependencies>
```

Crea un produttore

Il seguente esempio di codice mostra come scrivere su una coda RabbitMQ utilizzando JMS:

```
import jakarta.jms.*;
import com.rabbitmq.jms.admin.*;

// Setting the connection factory
RMQConnectionFactory factory = new RMQConnectionFactory();
factory.setHost(envProps.getProperty("RABBITMQ_HOST", "localhost"));
factory.setPort(Integer.parseInt(envProps.getProperty("RABBITMQ_PORT", "5672")));
factory.setUsername(envProps.getProperty("RABBITMQ_USERNAME", "guest"));
factory.setPassword(envProps.getProperty("RABBITMQ_PASSWORD", "guest"));
factory.setVirtualHost(envProps.getProperty("RABBITMQ_VIRTUAL_HOST", "/"));
factory.useSslProtocol();
```

```
connection = factory.createConnection();
connection.start();

String queueName = "test-queue-jms";
Session session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);

RMQDestination destination = new RMQDestination(queueName, true, false);

// Send the message to the queue
MessageProducer producer = session.createProducer(destination);
producer.setDeliveryMode(DeliveryMode.PERSISTENT);

String msg_content = "Hello World!!";
TextMessage textMessage = session.createTextMessage(msg_content);
producer.send(textMessage);

System.out.printf("Published to AMQP queue '%s': %s", queueName, msg_content);
```

Crea un consumatore

Il seguente esempio di codice mostra come leggere da una coda RabbitMQ utilizzando JMS:

```
import jakarta.jms.*;
import com.rabbitmq.jms.admin.*;

// Setting the connection factory
RMQConnectionFactory factory = new RMQConnectionFactory();
factory.setHost(envProps.getProperty("RABBITMQ_HOST", "localhost"));
factory.setPort(Integer.parseInt(envProps.getProperty("RABBITMQ_PORT", "5672")));
factory.setUsername(envProps.getProperty("RABBITMQ_USERNAME", "guest"));
factory.setPassword(envProps.getProperty("RABBITMQ_PASSWORD", "guest"));
factory.setVirtualHost(envProps.getProperty("RABBITMQ_VIRTUAL_HOST", "/"));
factory.useSslProtocol();

// Establish the connection and session
jakarta.jms.Connection connection = factory.createConnection();

String queueName = "test-queue-jms";
Session session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);

RMQDestination destination = new RMQDestination();
destination.setDestinationName(queueName);
destination.setAmqp(true);
```

```
destination.setAmqpQueueName(queueName);

// Initialize consumer
MessageConsumer consumer = session.createConsumer(destination);
consumer.setMessageListener(message -> {
    try {
        if (message instanceof TextMessage) {
            TextMessage textMessage = (TextMessage) message;
            System.out.printf("Message: %s%n", textMessage.getText());
        } else if (message instanceof BytesMessage) {
            BytesMessage bytesMessage = (BytesMessage) message;
            byte[] bytes = new byte[(int) bytesMessage.getBodyLength()];
            bytesMessage.readBytes(bytes);
            String content = new String(bytes);
            System.out.printf("Message: %s%n", content);
        } else {
            System.out.printf("Message: [%s]%n", message.getClass().getSimpleName());
        }
    } catch (JMSException e) {
        System.err.printf("Error processing message: %s%n", e.getMessage());
    }
});

connection.start();
```

Sicurezza in Amazon MQ

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di data center e architetture di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra te e te. AWS Il modello di responsabilità condivisa descrive questo aspetto come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel AWS cloud. AWS ti fornisce anche servizi che puoi utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei AWS Programmi di AWS conformità dei Programmi di conformità dei di . Per ulteriori informazioni sui programmi di conformità che si applicano ad Amazon MQ, consulta AWS Services in Scope by Compliance Program AWS .
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. L'utente è anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della propria azienda e le leggi e normative vigenti.

Questa documentazione consente di comprendere come applicare il modello di responsabilità condivisa quando si usa Amazon MQ. Gli argomenti seguenti descrivono come configurare Amazon MQ per soddisfare gli obiettivi di sicurezza e conformità. Scopri anche come utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere le tue risorse Amazon MQ.

Argomenti

- Protezione dei dati in Amazon MQ
- Identity and Access Management per Amazon MQ
- Convalida della conformità per Amazon MQ
- Resilienza in Amazon MQ
- Sicurezza dell'infrastruttura in Amazon MQ
- Best practice di sicurezza per Amazon MQ

Protezione dei dati in Amazon MQ

Il modello di responsabilità AWS condivisa si applica alla protezione dei dati in Amazon MQ. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. Per maggiori informazioni sulla privacy dei dati, consulta le Domande frequenti sulla privacy dei dati. Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al AWS Modello di responsabilità condivisa e GDPR nel AWS Blog sulla sicurezza.

Ai fini della protezione dei dati, consigliamo di proteggere Account AWS le credenziali e configurare i singoli utenti con AWS IAM Identity Center or AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- SSL/TLS Da utilizzare per comunicare con AWS le risorse. È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con AWS CloudTrail. Per informazioni sull'utilizzo dei CloudTrail percorsi per acquisire AWS le attività, consulta Lavorare con i CloudTrail percorsi nella Guida per l'AWS CloudTrail utente.
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di moduli crittografici convalidati FIPS 140-3 per accedere AWS tramite un'interfaccia a riga di comando o un'API, usa un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il Federal Information Processing Standard (FIPS) 140-3.

Ti consigliamo di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori con Amazon MQ o altro Servizi AWS utilizzando la console, l'API o AWS SDKs. AWS CLI I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i la fatturazione o i log di diagnostica. Quando si fornisce un URL a un server esterno,

suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la richiesta al server.

Non utilizzare alcuna informazione personale di identificazione (PII) o altre informazioni riservate o sensibili per i nomi dei broker o i nomi utente durante la creazione di risorse tramite la console Web del broker o l'API Amazon MQ, né per Amazon MQ per ActiveMQ, né per Amazon MQ per i broker RabbitMQ. I nomi utente e i nomi utente dei broker sono accessibili ad altri AWS servizi, inclusi i CloudWatch log. I nomi utenti dei broker non sono destinati ad essere utilizzati per dati privati o sensibili.

> **Important**
>
> TLS 1.3 non è disponibile per i broker RabbitMQ.

Encryption (Crittografia)

I dati utente memorizzati in Amazon MQ sono crittografati a riposo. La crittografia a riposo di Amazon MQ offre sicurezza avanzata grazie alla crittografia dei dati mediante le chiavi di crittografia archiviate in AWS Key Management Service (KMS). Questo servizio consente di ridurre gli oneri operativi e la complessità associati alla protezione dei dati sensibili. La crittografia dei dati inattivi consente di creare applicazioni sicure che rispettano rigorosi requisiti normativi e di conformità per la crittografia.

Tutte le connessioni tra i broker Amazon MQ utilizzano Transport Layer Security (TLS) per fornire la crittografia dei dati in transito.

Amazon MQ crittografa i messaggi a riposo e in transito utilizzando chiavi di crittografia che gestisce e memorizza in modo sicuro. Per ulteriori informazioni, consulta la Guida per gli sviluppatori di AWS Encryption SDK.

Crittografia dei dati a riposo

Amazon MQ si integra con AWS Key Management Service (KMS) per offrire una crittografia trasparente lato server. Amazon MQ esegue sempre la crittografia dei dati a riposo.

Quando crei un broker Amazon MQ for ActiveMQ o un broker Amazon MQ per RabbitMQ, puoi specificare cosa vuoi che Amazon MQ utilizzi per crittografare i AWS KMS key tuoi dati inattivi. Se non specifichi una chiave KMS, Amazon MQ crea per te una chiave KMS di AWS proprietà e

la utilizza per tuo conto. Amazon MQ supporta attualmente chiavi KMS simmetriche. Per ulteriori informazioni sulle chiavi KMS, consulta AWS KMS keys.

Quando si crea un broker, è possibile configurare ciò che Amazon MQ impiega per la chiave di crittografia selezionando una delle opzioni seguenti.

- Amazon MQ owned KMS key (default) (Chiave KMS di proprietà di Amazon MQ (di default)): la chiave è di proprietà ed è gestita da Amazon MQ e non è presente nel tuo account.
- AWS chiave KMS gestita: la chiave KMS AWS gestita (`aws/mq`) è una chiave KMS nel tuo account che viene creata, gestita e utilizzata per tuo conto da Amazon MQ.
- Select existing customer managed KMS key (Seleziona chiave KMS esistente gestita dal cliente): le chiavi KMS gestite dal cliente vengono create e gestite da te in AWS Key Management Service (KMS).

Important

- Non è possibile annullare la revoca di una concessione. Elimina il broker per revocare i diritti di accesso.
- Per i broker Amazon MQ for ActiveMQ che utilizzano Amazon Elastic File System (EFS) per archiviare i dati dei messaggi, potrebbero essere necessarie diverse ore prima che le autorizzazioni per l'uso delle chiavi KMS nel tuo account vengano revocate dopo aver eseguito le azioni richieste.
- Per i broker Amazon MQ per RabbitMQ e Amazon MQ per ActiveMQ che utilizzano EBS per archiviare i dati dei messaggi, se disattivi, pianifichi l'eliminazione o revochi la concessione dell'autorizzazione ad Amazon EBS a utilizzare le chiavi KMS nel tuo account, Amazon MQ non sarà in grado di gestire il broker e questo potrebbe passare a uno stato degradato.
- Se hai disattivato la chiave o pianificato l'eliminazione della chiave, puoi riattivarla o annullare l'eliminazione della chiave e mantenere il tuo broker aggiornato.
- Potrebbero essere necessarie diverse ore per disattivare una chiave o revocare una concessione dopo aver eseguito le azioni richieste.
- Per crittografare o decrittografare CloudWatch i log, non puoi configurare ciò che Amazon MQ utilizza per la tua chiave di crittografia. CloudWatch i log proteggono i dati inattivi utilizzando la crittografia e i gruppi di log sono crittografati. Il servizio CloudWatch logs gestisce la crittografia lato server per impostazione predefinita. Per ulteriori informazioni

su come vengono crittografati i gruppi di log, consulta la Amazon CloudWatch Logs User Guide.

Quando crei un broker di istanze singolo con una chiave KMS per RabbitMQ, vedrai due eventi `CreateGrant` registrati in AWS CloudTrail. Il primo evento è la creazione di una concessione da parte di Amazon MQ per la chiave KMS. Il secondo evento è la creazione da parte di EBS di una sovvenzione da utilizzare.

CreateGrant AWS CloudTrail immissione di registro: broker a istanza singola

mq_grant


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/AmazonMqConsole",
        "accountId": "111122223333",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAIOSFODNN7EXAMPLE",
                "arn": "arn:aws:iam::111122223333:user/AmazonMqConsole",
                "accountId": "111122223333",
                "userName": "AmazonMqConsole"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-02-23T18:59:10Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "mq.amazonaws.com"
    },
    "eventTime": "2018-06-28T22:23:46Z",
    "eventSource": "amazonmq.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "us-west-2",
```

```
    "sourceIPAddress": "203.0.113.0",
    "userAgent": "PostmanRuntime/7.1.5",
    "requestParameters": {
        "granteePrincipal": "mq.amazonaws.com",
        "keyId": "arn:aws:kms:us-east-1:316438333700:key/bdbe42ae-f825-4e78-
a8a1-828d411c4be2",
        "retiringPrincipal": "mq.amazonaws.com",
        "operations": [
            "CreateGrant",
            "Decrypt",
            "GenerateDataKeyWithoutPlaintext",
            "ReEncryptFrom",
            "ReEncryptTo",
            "DescribeKey"
        ]
    },
    "responseElements": {
        "grantId":
 "0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",

    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": false,
    "resources": [
        {
           "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "sessionCredentialFromConsole": "true"
}
```

EBS grant creation

Vedrai un evento per la creazione della sovvenzione EBS.

```
                                    {
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "mq.amazonaws.com"
    },
    "eventTime": "2023-02-23T19:09:40Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "mq.amazonaws.com",
    "userAgent": "ExampleDesktop/1.0 (V1; OS)",
    "requestParameters": {
        "granteePrincipal": "mq.amazonaws.com",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
        "constraints": {
            "encryptionContextSubset": {
                "aws:ebs:id": "vol-0b670f00f7d5417c0"
            }
        },
        "operations": [
            "Decrypt"
        ],
        "retiringPrincipal": "ec2.us-east-1.amazonaws.com"
    },
    "responseElements": {
        "grantId":
 "0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
    },
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
```

```
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "sharedEventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventCategory": "Management"
}
```

Quando crei una distribuzione di cluster con una chiave KMS per RabbitMQ, vedrai cinque eventi `CreateGrant` registrati in AWS CloudTrail. I primi due eventi sono creazioni di sovvenzioni per Amazon MQ. I prossimi tre eventi sono sovvenzioni create da EBS per essere utilizzate da EBS.

CreateGrant AWS CloudTrail immissione di registro: distribuzione del cluster

mq_grant

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/AmazonMqConsole",
        "accountId": "111122223333",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAIOSFODNN7EXAMPLE",
                "arn": "arn:aws:iam::111122223333:user/AmazonMqConsole",
                "accountId": "111122223333",
                "userName": "AmazonMqConsole"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-02-23T18:59:10Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "mq.amazonaws.com"
    },
```

```
    "eventTime": "2018-06-28T22:23:46Z",
    "eventSource": "amazonmq.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "203.0.113.0",
    "userAgent": "PostmanRuntime/7.1.5",
    "requestParameters": {
        "granteePrincipal": "mq.amazonaws.com",
        "keyId": "arn:aws:kms:us-east-1:316438333700:key/bdbe42ae-f825-4e78-
a8a1-828d411c4be2",
        "retiringPrincipal": "mq.amazonaws.com",
        "operations": [
            "CreateGrant",
            "Encrypt",
            "Decrypt",
            "ReEncryptFrom",
            "ReEncryptTo",
            "GenerateDataKey",
            "GenerateDataKeyWithoutPlaintext",
            "DescribeKey"
        ]
    },
    "responseElements": {
        "grantId":
 "0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",

    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": false,
    "resources": [
        {
           "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "sessionCredentialFromConsole": "true"
```

```
}
```

mq_rabbit_grant

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/AmazonMqConsole",
        "accountId": "111122223333",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAIOSFODNN7EXAMPLE",
                "arn": "arn:aws:iam::111122223333:user/AmazonMqConsole",
                "accountId": "111122223333",
                "userName": "AmazonMqConsole"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-02-23T18:59:10Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "mq.amazonaws.com"
    },
    "eventTime": "2018-06-28T22:23:46Z",
    "eventSource": "amazonmq.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "203.0.113.0",
    "userAgent": "PostmanRuntime/7.1.5",
    "requestParameters": {
        "granteePrincipal": "mq.amazonaws.com",
        "retiringPrincipal": "mq.amazonaws.com",
        "operations": [
            "DescribeKey"
        ],
```

```
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
    },
    "responseElements": {
        "grantId":
 "0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",

    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": false,
    "resources": [
        {
           "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "sessionCredentialFromConsole": "true"
}
```

EBS grant creation

Vedrai tre eventi per la creazione delle sovvenzioni EBS.

```
                                    {
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "mq.amazonaws.com"
    },
    "eventTime": "2023-02-23T19:09:40Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "mq.amazonaws.com",
```

```
    "userAgent": "ExampleDesktop/1.0 (V1; OS)",
    "requestParameters": {
        "granteePrincipal": "mq.amazonaws.com",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
        "constraints": {
            "encryptionContextSubset": {
                "aws:ebs:id": "vol-0b670f00f7d5417c0"
            }
        },
        "operations": [
            "Decrypt"
        ],
        "retiringPrincipal": "ec2.us-east-1.amazonaws.com"
    },
    "responseElements": {
        "grantId":
 "0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
    },
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "sharedEventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventCategory": "Management"
}
```


Per ulteriori informazioni sulle chiavi KMS, consultare AWS KMS keys nella Guida per sviluppatori di AWS Key Management Service .

Crittografia dei dati in transito

Amazon MQ per ActiveMQ: Amazon MQ per ActiveMQ richiede un elevato livello di sicurezza TLS (Transport Layer Security) ed effettua la crittografia dei dati in transito tra i broker dell'implementazione Amazon MQ. Tutti i dati trasmessi tra i broker Amazon MQ sono crittografati utilizzando Transport Layer Security (TLS). Questo vale per tutti i protocolli disponibili.

Amazon MQ per RabbitMQ: Amazon MQ per RabbitMQ richiede una crittografia TLS (Transport Layer Security) avanzata per tutte le connessioni client. Il traffico di replica del cluster RabbitMQ transita solo sul VPC del broker e tutto il traffico di rete tra i data AWS center è crittografato in modo trasparente a livello fisico. I broker in cluster Amazon MQ per RabbitMQ attualmente non supportano la crittografia tra nodi per la replica dei cluster. Per ulteriori informazioni, consulta Encrypting e -in-Transit data-in-transit. Data-at-Rest

Protocolli Amazon MQ per ActiveMQ

È possibile accedere ai broker ActiveMQ utilizzando i seguenti protocolli con TLS abilitato:

- AMQP
- MQTT
- MQTT over WebSocket
- OpenWire
- STOMP
- STOMP over WebSocket

Suite di cifratura TLS supportate per ActiveMQ

ActiveMQ su Amazon MQ supporta i pacchetti di crittografia seguenti:

- TLS_ECDHE_RSA_WITH_AES_256_GCM_ SHA384
- TLS_ECDHE_RSA_CON_AES_256_CBC_ SHA384
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_CON_AES_256_GCM_ SHA384
- TLS_DHE_RSA_CON_AES_256_CBC_ SHA256
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_CON_AES_256_GCM_ SHA384

- TLS_RSA_CON_AES_256_CBC_ SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_RSA_CON_AES_128_GCM_ SHA256
- TLS_ECDHE_RSA_CON_AES_128_CBC_ SHA256
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_CON_AES_128_GCM_ SHA256
- TLS_DHE_RSA_CON_AES_128_CBC_ SHA256
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_CON_AES_128_GCM_ SHA256
- TLS_RSA_CON_AES_128_CBC_ SHA256
- TLS_RSA_WITH_AES_128_CBC_SHA

Protocolli Amazon MQ per RabbitMQ

È possibile accedere ai broker RabbitMQ utilizzando i seguenti protocolli con TLS abilitato:

- AMQP (0-9-1)

Suite di cifratura TLS supportate per RabbitMQ

RabbitMQ su Amazon MQ supporta i pacchetti di crittografia seguenti:

- TLS_ECDHE_RSA_CON_AES_256_GCM_ SHA384
- TLS_ECDHE_RSA_CON_AES_128_GCM_ SHA256

Identity and Access Management per Amazon MQ

AWS Identity and Access Management (IAM) è un programma Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle AWS risorse. Gli amministratori IAM controllano chi è autenticato (con accesso effettuato) e autorizzato (che dispone di autorizzazioni) a utilizzare risorse Amazon MQ. IAM è un software Servizio AWS che puoi utilizzare senza costi aggiuntivi.

Argomenti

- Destinatari
- Autenticazione con identità
- Gestione dell’accesso tramite policy
- Funzionamento di Amazon MQ con IAM
- Esempi di policy basate su identità per Amazon MQ
- Autenticazione e autorizzazione API per Amazon MQ
- Autenticazione e autorizzazione del broker
- AWS politiche gestite per Amazon MQ
- Utilizzo di ruoli collegati ai servizi per Amazon MQ
- Risoluzione dei problemi relativi all'identità e all'accesso di Amazon MQ

Destinatari

Il modo in cui utilizzi AWS Identity and Access Management (IAM) varia in base al tuo ruolo:

- Utente del servizio: richiedi le autorizzazioni all’amministratore se non riesci ad accedere alle funzionalità (consulta Risoluzione dei problemi relativi all'identità e all'accesso di Amazon MQ)
- Amministratore del servizio: determina l’accesso degli utenti e invia le richieste di autorizzazione (consulta Funzionamento di Amazon MQ con IAM)
- Amministratore IAM: scrivi policy per gestire l’accesso (consulta Esempi di policy basate su identità per Amazon MQ)

Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi autenticarti come utente IAM o assumendo un ruolo IAM. Utente root dell'account AWS

Puoi accedere come identità federata utilizzando credenziali provenienti da una fonte di identità come AWS IAM Identity Center (IAM Identity Center), autenticazione Single Sign-On o credenziali. Google/ Facebook Per ulteriori informazioni sull’accesso, consulta Come accedere all’ Account AWS nella Guida per l’utente di Accedi ad AWS .

Per l'accesso programmatico, AWS fornisce un SDK e una CLI per firmare crittograficamente le richieste. Per ulteriori informazioni, consulta AWS Signature Version 4 per le richieste API nella Guida per l’utente di IAM.

Account AWS utente root

Quando si crea un Account AWS, si inizia con un'identità di accesso denominata utente Account AWS root che ha accesso completo a tutte Servizi AWS le risorse. Consigliamo vivamente di non utilizzare l'utente root per le attività quotidiane. Per le attività che richiedono le credenziali come utente root, consulta Attività che richiedono le credenziali dell'utente root nella Guida per l'utente di IAM.

Utenti e gruppi

Un utente IAM è una identità che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ti consigliamo di utilizzare credenziali temporanee invece di utenti IAM con credenziali a lungo termine. Per ulteriori informazioni, consulta Richiedere agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee nella Guida per l'utente IAM.

Un gruppo IAM specifica una raccolta di utenti IAM e semplifica la gestione delle autorizzazioni per gestire gruppi di utenti di grandi dimensioni. Per ulteriori informazioni, consulta Casi d'uso per utenti IAM nella Guida per l'utente di IAM.

Ruoli IAM

Un ruolo IAM è un'identità con autorizzazioni specifiche che fornisce credenziali temporanee. Puoi assumere un ruolo passando da un ruolo utente a un ruolo IAM (console) o chiamando un'operazione AWS CLI o AWS API. Per ulteriori informazioni, consulta Metodi per assumere un ruolo nella Guida per l'utente di IAM.

I ruoli IAM sono utili per l'accesso degli utenti federati, le autorizzazioni utente IAM temporanee, l'accesso multi-account, l'accesso multi-servizio e le applicazioni in esecuzione su Amazon EC2. Per maggiori informazioni, consultare Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.

Gestione dell'accesso tramite policy

Puoi controllare l'accesso AWS creando policy e associandole a AWS identità o risorse. Una policy definisce le autorizzazioni quando è associata a un'identità o a una risorsa. AWS valuta queste politiche quando un preside effettua una richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per maggiori informazioni sui documenti delle policy JSON, consulta Panoramica delle policy JSON nella Guida per l'utente IAM.

Utilizzando le policy, gli amministratori specificano chi ha accesso a cosa definendo quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Un amministratore IAM crea le policy IAM e le aggiunge ai ruoli, che gli utenti possono quindi assumere. Le policy IAM definiscono le autorizzazioni indipendentemente dal metodo utilizzato per eseguirle.

Policy basate sull'identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile collegare a un'identità (utente, gruppo o ruolo). Tali policy controllano le operazioni autorizzate per l'identità, nonché le risorse e le condizioni in cui possono essere eseguite. Per informazioni su come creare una policy basata su identità, consultare Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente nella Guida per l'utente IAM.

Le policy basate su identità possono essere policy in linea (con embedding direttamente in una singola identità) o policy gestite (policy autonome collegate a più identità). Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta Scegliere tra policy gestite e policy in linea nella Guida per l'utente di IAM.

Policy basate sulle risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi includono le policy di trust dei ruoli IAM e le policy dei bucket di Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. In una policy basata sulle risorse è obbligatorio specificare un'entità principale.

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non è possibile utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

Liste di controllo degli accessi () ACLs

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

Amazon S3 e Amazon VPC sono esempi di servizi che supportano. AWS WAF ACLs Per ulteriori informazioni ACLs, consulta la panoramica della lista di controllo degli accessi (ACL) nella Amazon Simple Storage Service Developer Guide.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi che possono impostare le autorizzazioni massime concesse dai tipi di policy più comuni:

- Limiti delle autorizzazioni: imposta il numero massimo di autorizzazioni che una policy basata su identità ha la possibilità di concedere a un'entità IAM. Per ulteriori informazioni, consulta Limiti delle autorizzazioni per le entità IAM nella Guida per l'utente di IAM.
- Politiche di controllo del servizio (SCPs): specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa in. AWS Organizations Per ulteriori informazioni, consultare Policy di controllo dei servizi nella Guida per l'utente di AWS Organizations .
- Politiche di controllo delle risorse (RCPs): imposta le autorizzazioni massime disponibili per le risorse nei tuoi account. Per ulteriori informazioni, consulta Politiche di controllo delle risorse (RCPs) nella Guida per l'AWS Organizations utente.
- Policy di sessione: policy avanzate passate come parametro quando si crea una sessione temporanea per un ruolo o un utente federato. Per maggiori informazioni, consultare Policy di sessione nella Guida per l'utente IAM.

Più tipi di policy

Quando a una richiesta si applicano più tipi di policy, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire o meno una richiesta quando sono coinvolti più tipi di policy, consulta Logica di valutazione delle policy nella IAM User Guide.

Funzionamento di Amazon MQ con IAM

Prima di utilizzare IAM per gestire l'accesso ad Amazon MQ, è necessario comprendere quali funzioni IAM sono disponibili per l'uso con Amazon MQ. Per avere una visione di alto livello di come Amazon MQ e altri AWS servizi funzionano con IAM, consulta AWS Services That Work with IAM nella IAM User Guide.

Amazon MQ utilizza IAM for Amazon MQ API operations per creare, aggiornare, eliminare ed elencare broker. Per l'accesso tramite broker per pubblicare e sottoscrivere messaggi, Amazon MQ for ActiveMQ supporta l'autenticazione ActiveMQ nativa e LDAP, mentre Amazon MQ per RabbitMQ supporta l'autenticazione IAM e altri metodi. Per ulteriori informazioni, consulta the section called "Autenticazione e autorizzazione del broker".

Argomenti

- Policy basate su identità Amazon MQ
- Policy basate su risorse Amazon MQ
- Autorizzazione basata su tag Amazon MQ
- Ruoli IAM di Amazon MQ

Policy basate su identità Amazon MQ

Con le policy basate sull'identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte. Amazon MQ supporta specifiche operazioni, risorse e chiavi di condizione. Per informazioni su tutti gli elementi utilizzati in una policy JSON, consulta Documentazione di riferimento degli elementi delle policy JSON IAM nella Guida per l'utente IAM.

Azioni

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento `Action` di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso in una policy. Includere le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Le operazioni delle policy in Amazon MQ utilizzano il seguente prefisso prima dell'operazione: `mq:`. Ad esempio, per concedere a qualcuno l'autorizzazione per eseguire un'istanza Amazon MQ con l'operazione API `CreateBroker` di Amazon MQ, è necessario includere l'operazione `mq:CreateBroker` nella policy. Le istruzioni della policy devono includere un elemento `Action` o `NotAction`. Amazon MQ definisce un proprio set di operazioni che descrivono le attività eseguibili con quel servizio.

Per specificare più azioni in una sola istruzione, separa ciascuna di esse con una virgola come mostrato di seguito:

```
"Action": [
      "mq:action1",
      "mq:action2"
```

È possibile specificare più azioni tramite caratteri jolly (*). Ad esempio, per specificare tutte le azioni che iniziano con la parola `Describe`, includi la seguente azione:

```
"Action": "mq:Describe*"
```

Per visualizzare un elenco delle operazioni Amazon MQ, consulta Operazioni definite da Amazon MQ nella Guida per l'utente di IAM.

Resources

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento JSON `Resource` della policy specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Come best practice, specifica una risorsa utilizzando il suo nome della risorsa Amazon (ARN). Per le azioni che non supportano le autorizzazioni a livello di risorsa, si utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

```
"Resource": "*"
```

In Amazon MQ, le AWS risorse principali sono un broker di messaggi Amazon MQ e la sua configurazione. I broker e le configurazioni Amazon MQ sono associati a ciascuno di essi con Amazon Resource Names (ARNs) univoci, come illustrato nella tabella seguente.

Tipi di risorsa	ARN	Chiavi di condizione
brokers	`arn:aws:mq:us-east-1:123456789012:broker:${brokerName}:${brokerId}`	aws:ResourceTag/${TagKey}
configurations	`arn:${Partition}:mq:${Region}:${Account}:configuration:${configuration-id}`	aws:ResourceTag/${TagKey}

Per ulteriori informazioni sul formato di ARNs, consulta Amazon Resource Names (ARNs) e AWS Service Namespaces.

Ad esempio, per specificare il broker denominato `MyBroker` con brokerId `b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9` nell'istruzione, utilizza il seguente ARN:

```
"Resource": "arn:aws:mq:us-
east-1:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9"
```

Per specificare tutti i broker e le configurazioni che appartengono ad un account specifico, utilizza il carattere jolly (*):

```
"Resource": "arn:aws:mq:us-east-1:123456789012:*"
```

Alcune operazioni Amazon MQ, ad esempio quelle per la creazione di risorse, non possono essere eseguite su una risorsa specifica. In questi casi, è necessario utilizzare il carattere jolly (*).

```
"Resource": "*"
```

L'operazione API `CreateTags` richiede sia un broker che una configurazione. Per specificare più risorse in una singola istruzione, separale con virgole. ARNs

```
"Resource": [
      "resource1",
      "resource2"
```

Per visualizzare un elenco dei tipi di risorse Amazon MQ e relativi ARNs, consulta Resources Defined by Amazon MQ nella IAM User Guide. Per informazioni sulle operazioni con cui è possibile specificare l'ARN di ogni risorsa, consultare Operazioni definite da Amazon MQ.

Chiavi di condizione

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento `Condition` specifica quando le istruzioni vengono eseguite in base a criteri definiti. È possibile compilare espressioni condizionali che utilizzano operatori di condizione, ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di contesto delle condizioni AWS globali nella Guida per l'utente IAM.

Amazon MQ non fornisce chiavi di condizione specifiche del servizio, ma supporta l'utilizzo di alcune chiavi di condizione globali. Per visualizzare un elenco delle chiavi condizione di Amazon MQ, consultare le chiavi condizioni per Amazon MQ nella Guida per l'utente di IAM. Per informazioni su

operazioni e risorse con cui è possibile utilizzare una chiave di condizione, consultare Operazioni definite da Amazon MQ.

Chiavi di condizione	Descrizione	Tipo
aws: RequestTag /$ {} TagKey	Filtra le operazioni in base ai tag passati nella richiesta	Stringa
legge: ResourceTag /$ {} TagKey	Filtra le operazioni in base ai tag associati alla risorsa.	Stringa
leggi: TagKeys	Filtra le operazioni in base alle chiavi di tag passate nella richiesta	Stringa

Esempi

Per visualizzare esempi di policy basate su identità Amazon MQ, consultare Esempi di policy basate su identità per Amazon MQ.

Policy basate su risorse Amazon MQ

Attualmente, Amazon MQ non supporta l'autenticazione IAM utilizzando autorizzazioni basate sulle risorse o policy basate sulle risorse.

Autorizzazione basata su tag Amazon MQ

Puoi collegare i tag alle risorse Amazon MQ o inoltrarli in una richiesta ad Amazon MQ. Per controllare l'accesso basato su tag, fornire informazioni sui tag nell'elemento condizione di una policy utilizzando le chiavi di condizione `mq:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* o `aws:TagKeys`.

Amazon MQ supporta policy basate su tag. Ad esempio, è possibile negare l'accesso a tutte le risorse Amazon MQ che includono un tag con la chiave `environment` e il valore `production`:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "mq:DeleteBroker",
                "mq:RebootBroker",
                "mq:DeleteTags"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/environment": "production"
                }
            }
        }
    ]
}
```

Questa policy si occuperà di `Deny` la possibilità di eliminare o riavviare un broker Amazon MQ che include il tag `environment/production`.

Per ulteriori informazioni sui tag, consulta:

- Aggiungere tag alle risorse Amazon MQ
- Controllo degli accessi tramite tag IAM

Ruoli IAM di Amazon MQ

Un ruolo IAM è un'entità all'interno del tuo AWS account che dispone di autorizzazioni specifiche.

Utilizzo di credenziali temporanee con Amazon MQ

È possibile utilizzare credenziali temporanee per effettuare l'accesso con la federazione, assumere un ruolo IAM o un ruolo multi-account. Ottieni credenziali di sicurezza temporanee chiamando operazioni AWS STS API come AssumeRoleo. GetFederationToken

Amazon MQ supporta l'uso di credenziali temporanee.

Ruoli dei servizi

Questa funzionalità consente a un servizio di assumere un ruolo di servizio per conto dell'utente. Questo ruolo consente al servizio di accedere alle risorse in altri servizi per completare un'azione per conto dell'utente. I ruoli dei servizi sono visualizzati nell'account IAM e sono di proprietà dell'account. Ciò significa che un amministratore IAM può modificare le autorizzazioni per questo ruolo. Tuttavia, questo potrebbe pregiudicare la funzionalità del servizio.

Amazon MQ supporta i ruoli del servizio.

Esempi di policy basate su identità per Amazon MQ

Per impostazione predefinita, gli utenti e i ruoli IAM non dispongono dell'autorizzazione per creare o modificare risorse Amazon MQ. Inoltre, non possono eseguire attività utilizzando l' AWS API Console di gestione AWS AWS CLI, o. Un amministratore IAM deve creare policy IAM che concedono a utenti e ruoli l'autorizzazione per eseguire operazioni API specifiche sulle risorse specificate di cui hanno bisogno. L'amministratore deve quindi allegare queste policy a utenti o IAM che richiedono tali autorizzazioni.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta Creazione di policy nella scheda JSON nella Guida per l'utente IAM.

Argomenti

- Best practice delle policy
- Utilizzo della console Amazon MQ
- Consentire agli utenti di visualizzare le loro autorizzazioni

Best practice delle policy

Le policy basate su identità determinano se qualcuno può creare, accedere o eliminare risorse Amazon MQ nell'account. Queste azioni possono comportare costi aggiuntivi per l' Account AWS. Quando si creano o modificano policy basate sull'identità, seguire queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche AWS gestite che

concedono le autorizzazioni per molti casi d'uso comuni. Sono disponibili nel tuo. Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per maggiori informazioni, consulta Policy gestite da AWS o Policy gestite da AWS per le funzioni dei processi nella Guida per l'utente di IAM.

- Applicazione delle autorizzazioni con privilegio minimo - Quando si impostano le autorizzazioni con le policy IAM, concedere solo le autorizzazioni richieste per eseguire un'attività. È possibile farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegio minimo. Per maggiori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso - Per limitare l'accesso ad azioni e risorse è possibile aggiungere una condizione alle policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio CloudFormation. Per maggiori informazioni, consultare la sezione Elementi delle policy JSON di IAM: condizione nella Guida per l'utente di IAM.
- Utilizzo dello strumento di analisi degli accessi IAM per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali - Lo strumento di analisi degli accessi IAM convalida le policy nuove ed esistenti in modo che aderiscano al linguaggio (JSON) della policy IAM e alle best practice di IAM. Lo strumento di analisi degli accessi IAM offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per maggiori informazioni, consultare Convalida delle policy per il Sistema di analisi degli accessi IAM nella Guida per l'utente di IAM.
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungere le condizioni MFA alle policy. Per maggiori informazioni, consultare Protezione dell'accesso API con MFA nella Guida per l'utente di IAM.

Per maggiori informazioni sulle best practice in IAM, consulta Best practice di sicurezza in IAM nella Guida per l'utente di IAM.

Utilizzo della console Amazon MQ

Per accedere alla console Amazon MQ, è necessario disporre di un set di autorizzazioni minimo. Queste autorizzazioni devono consentirti di elencare e visualizzare i dettagli sulle risorse Amazon MQ presenti nel tuo AWS account. Se crei una policy basata su identità più restrittiva rispetto alle

autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti e ruoli IAM) associate a tale policy.

Per garantire che tali entità possano ancora utilizzare la console Amazon MQ, allega anche la seguente politica AWS gestita alle entità. Per ulteriori informazioni, consulta Aggiunta di autorizzazioni a un utente nella Guida per l'utente di IAM:

```
AmazonMQReadOnlyAccess
```

Non è necessario consentire autorizzazioni minime di console per gli utenti che effettuano chiamate solo verso AWS CLI o l' AWS API. Al contrario, è possibile accedere solo alle operazioni che soddisfano l'operazione API che stai cercando di eseguire.

Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono collegate alla relativa identità utente. Questa politica include le autorizzazioni per completare questa azione sulla console o utilizzando l'API o a livello di codice. AWS CLI AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
```

```
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Autenticazione e autorizzazione API per Amazon MQ

Amazon MQ utilizza la firma delle AWS richieste standard per l'autenticazione delle API. Per ulteriori informazioni, consulta la sezione relativa alla AWS firma delle richieste API nella Riferimenti generali di AWS.

> **Note**
>
> Attualmente, Amazon MQ non supporta l'autenticazione IAM utilizzando autorizzazioni basate sulle risorse o policy basate sulle risorse.

Per autorizzare AWS gli utenti a lavorare con broker, configurazioni e utenti, devi modificare le autorizzazioni della policy IAM.

Argomenti

- Autorizzazioni IAM richieste per creare un broker Amazon MQ
- Riferimento alle autorizzazioni API REST di Amazon MQ
- Riferimento per le autorizzazioni aggiuntive di Amazon MQ
- Autorizzazioni a livello di risorsa supportate per le operazioni API di Amazon MQ

Autorizzazioni IAM richieste per creare un broker Amazon MQ

Per creare un broker, utilizzare la policy IAM `AmazonMQFullAccess` oppure includere le seguenti autorizzazioni EC2 nella policy IAM.

La seguente policy personalizzata è composta da due istruzioni (una condizionale) che concedono le autorizzazioni per manipolare le risorse richieste da Amazon MQ per creare un broker ActiveMQ.

> **Important**
>
> - L'operazione `ec2:CreateNetworkInterface` è necessaria per consentire ad Amazon MQ di creare un'interfaccia di rete elastica (ENI) nell'account a tuo nome.
> - L'operazione `ec2:CreateNetworkInterfacePermission` autorizza Amazon MQ a collegare l'ENI a un broker ActiveMQ.
> - La chiave di condizione `ec2:AuthorizedService` garantisce che le autorizzazioni ENI possano essere concesse solo ad account del servizio Amazon MQ.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Action": [
            "mq:*",
            "ec2:CreateNetworkInterface",
            "ec2:DeleteNetworkInterface",
            "ec2:DetachNetworkInterface",
            "ec2:DescribeInternetGateways",
            "ec2:DescribeNetworkInterfaces",
            "ec2:DescribeRouteTables",
            "ec2:DescribeSecurityGroups",
            "ec2:DescribeSubnets",
            "ec2:DescribeVpcs"
        ],
        "Effect": "Allow",
        "Resource": "*"
    },{
        "Action": [
            "ec2:CreateNetworkInterfacePermission",
            "ec2:DeleteNetworkInterfacePermission",
            "ec2:DescribeNetworkInterfacePermissions"
        ],
        "Effect": "Allow",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "ec2:AuthorizedService": "mq.amazonaws.com"
```

```
            }
        }
    }]
}
```

Per ulteriori informazioni, consultare Passaggio 2: crea un utente e ottieni AWS le tue credenziali e Non modificare né eliminare mai l'interfaccia di rete elastica Amazon MQ.

Riferimento alle autorizzazioni API REST di Amazon MQ

La tabella seguente elenca Amazon MQ REST APIs e le autorizzazioni IAM corrispondenti.

Amazon MQ REST APIs e autorizzazioni richieste

Amazon MQ REST APIs	Autorizzazioni richieste
CreateBroker	mq:CreateBroker
CreateConfiguration	mq:CreateConfiguration
CreateTags	mq:CreateTags
CreateUser	mq:CreateUser
DeleteBroker	mq:DeleteBroker
DeleteUser	mq:DeleteUser
DescribeBroker	mq:DescribeBroker
DescribeConfiguration	mq:DescribeConfiguration
DescribeConfigurationRevision	mq:DescribeConfigurationRevision
DescribeUser	mq:DescribeUser
ListBrokers	mq:ListBrokers
ListConfigurationRevisions	mq:ListConfigurationRevisions
ListConfigurations	mq:ListConfigurations

Amazon MQ REST APIs	Autorizzazioni richieste
ListTags	mq:ListTags
ListUsers	mq:ListUsers
RebootBroker	mq:RebootBroker
UpdateBroker	mq:UpdateBroker
UpdateConfiguration	mq:UpdateConfiguration
UpdateUser	mq:UpdateUser

Riferimento per le autorizzazioni aggiuntive di Amazon MQ

La tabella seguente elenca l'API Amazon MQ e l'autorizzazione IAM aggiuntiva richiesta per funzionalità specifiche, come l'autenticazione OAuth 2.0.

API REST Amazon MQ	Autorizzazione	Description
UpdateBroker	mq:UpdateBrokerAccessConfiguration	È necessaria questa autorizzazione per aggiornare le opzioni di autenticazione e autorizzazione nella configurazione del broker associata. Per ulteriori informazioni, consulta OAuth autenticazione e autorizzazione 2.0 per Amazon MQ for RabbitMQ.

Autorizzazioni a livello di risorsa supportate per le operazioni API di Amazon MQ

Il concetto di autorizzazioni a livello di risorsa indica la possibilità di specificare le risorse su cui gli utenti sono autorizzati a eseguire operazioni. Amazon MQ supporta parzialmente le autorizzazioni a livello di risorsa. Per determinate operazioni di Amazon MQ, puoi controllare se gli utenti sono

autorizzati a utilizzarle in base a condizioni che devono essere soddisfatte o a specifiche risorse che gli utenti sono autorizzati a utilizzare.

La tabella seguente descrive le azioni dell'API Amazon MQ che attualmente supportano le autorizzazioni a livello di risorsa, nonché le risorse, le risorse e le chiavi di condizione supportate per ARNs ogni azione.

> **Important**
>
> Se un'operazione API di Amazon MQ non è presente in questa tabella, significa che non supporta le autorizzazioni a livello di risorsa. Se un'operazione API di Amazon MQ non supporta le autorizzazioni a livello di risorsa, puoi concedere agli utenti l'autorizzazione per utilizzare l'operazione, ma dovrai specificare il carattere jolly * per l'elemento di risorsa della tua dichiarazione di policy.

Operazione API	Tipi di risorsa (*obbligatorio)
CreateConfiguration	configurations*
CreateTags	brokers, configurations
CreateUser	brokers*
DeleteBroker	brokers*
DeleteUser	brokers*
DescribeBroker	brokers*
DescribeConfiguration	configurations*
DescribeConfigurationRevision	configurations*
DescribeUser	brokers*
ListConfigurationRevisions	configurations*

Operazione API	Tipi di risorsa (*obbligatorio)
ListConfigurationRevisions	configurations*
ListTags	brokers, configurations
ListUsers	brokers*
RebootBroker	brokers*
UpdateBroker	brokers*
UpdateConfiguration	configurations*
UpdateUser	brokers*

Autenticazione e autorizzazione del broker

Amazon MQ offre diversi metodi di autenticazione e autorizzazione a seconda del tipo di motore del broker.

Autenticazione e autorizzazione per Amazon MQ for ActiveMQ

Amazon MQ for ActiveMQ supporta i seguenti metodi di autenticazione e autorizzazione:

Autenticazione e autorizzazione semplici

Con questo metodo, gli utenti del broker vengono creati e gestiti tramite la console o l'API di Amazon MQ. Gli utenti possono essere configurati con autorizzazioni specifiche per accedere a code, argomenti e alla console Web ActiveMQ. Per ulteriori informazioni su questo metodo, vedere Creazione di un utente del broker ActiveMQ.

Autenticazione e autorizzazione LDAP

Con questo metodo, gli utenti del broker si autenticano tramite le credenziali memorizzate nel server LDAP. È possibile aggiungere, eliminare e modificare utenti e assegnare autorizzazioni ad argomenti e code tramite il server LDAP, fornendo autenticazione e autorizzazione centralizzate. Per ulteriori informazioni su questo metodo, vedere Integrazione dei broker ActiveMQ con LDAP.

Autenticazione e autorizzazione per Amazon MQ for RabbitMQ

Amazon MQ for RabbitMQ supporta i seguenti metodi di autenticazione e autorizzazione:

Autenticazione e autorizzazione semplici

Con questo metodo, gli utenti del broker vengono archiviati internamente nel broker RabbitMQ e gestiti tramite la console web o l'API di gestione. Le autorizzazioni per vhost, exchange, code e topic sono configurate direttamente in RabbitMQ. Questo è il metodo predefinito. Per ulteriori informazioni, vedere Autenticazione e autorizzazione semplici.

OAuth Autenticazione e autorizzazione 2.0

In questo metodo, gli utenti del broker e le relative autorizzazioni sono gestiti da un provider di identità OAuth 2.0 (IdP) esterno. L'autenticazione degli utenti e le autorizzazioni alle risorse per vhost, exchange, code e argomenti sono centralizzate tramite il sistema di ambito del OAuth provider 2.0. Ciò semplifica la gestione degli utenti e consente l'integrazione con i sistemi di identità esistenti. Per ulteriori informazioni, vedere Autenticazione e autorizzazione OAuth 2.0.

Autenticazione e autorizzazione IAM

Con questo metodo, gli utenti del broker si autenticano utilizzando le credenziali AWS IAM tramite la federazione in uscita IAM. Le credenziali IAM vengono utilizzate per ottenere i token JWT da AWS Security Token Service (STS) e questi token JWT fungono da token 2.0 per l'autenticazione. OAuth Questo metodo sfrutta il supporto OAuth 2.0 esistente in Amazon MQ for RabbitMQ, dove AWS funge da provider di identità OAuth 2.0. L'autenticazione degli utenti è gestita da AWS IAM, mentre le autorizzazioni delle risorse per vhost, exchange, code e argomenti sono gestite tramite policy IAM e alias di ambito configurati in RabbitMQ. Per ulteriori informazioni, consulta Autenticazione e autorizzazione IAM.

Autenticazione e autorizzazione LDAP

In questo metodo, gli utenti del broker e le relative autorizzazioni sono gestiti da un servizio di directory LDAP esterno. L'autenticazione degli utenti e le autorizzazioni delle risorse sono centralizzate tramite il server LDAP, che consente agli utenti di accedere a RabbitMQ utilizzando le credenziali del servizio di directory esistenti. Per ulteriori informazioni, vedere Autenticazione e autorizzazione LDAP.

Autenticazione e autorizzazione HTTP

In questo metodo, gli utenti del broker e le relative autorizzazioni sono gestiti da un server HTTP esterno. L'autenticazione degli utenti e le autorizzazioni delle risorse sono centralizzate tramite il server HTTP, che consente agli utenti di accedere a RabbitMQ utilizzando il proprio provider di autenticazione e autorizzazione. Per ulteriori informazioni su questo metodo, consulta Autenticazione e autorizzazione HTTP.

Autenticazione con certificato SSL

Amazon MQ supporta TLS reciproco (mTLS) per i broker RabbitMQ. Il plug-in di autenticazione SSL utilizza i certificati client delle connessioni MTLS per autenticare gli utenti. Con questo metodo, gli utenti del broker vengono autenticati utilizzando certificati client X.509 anziché credenziali di nome utente e password. Il certificato del client viene convalidato rispetto a un'autorità di certificazione (CA) affidabile e il nome utente viene estratto da un campo del certificato, ad esempio Common Name (CN) o Subject Alternative Name (SAN). Questo metodo fornisce un'autenticazione avanzata senza trasmettere credenziali sulla rete. Per ulteriori informazioni, consulta Autenticazione con certificato SSL.

> **Note**
>
> RabbitMQ supporta più metodi di autenticazione e autorizzazione da utilizzare contemporaneamente. Ad esempio, è possibile abilitare sia l'autenticazione OAuth 2.0 che quella semplice (interna). Per ulteriori informazioni, consulta la sezione del tutorial OAuth 2.0 sull'abilitazione dell'autenticazione OAuth 2.0 e semplice (interna) e la documentazione sul controllo degli accessi di RabbitMQ.
> Amazon MQ consiglia di creare un utente interno durante il test delle configurazioni di autenticazione. Ciò consente di convalidare la configurazione di accesso utilizzando l'API di gestione RabbitMQ. Per ulteriori informazioni, consulta Convalida dell'accesso.

AWS politiche gestite per Amazon MQ

Una politica AWS gestita è una politica autonoma creata e amministrata da AWS. AWS le politiche gestite sono progettate per fornire autorizzazioni per molti casi d'uso comuni, in modo da poter iniziare ad assegnare autorizzazioni a utenti, gruppi e ruoli.

Tieni presente che le policy AWS gestite potrebbero non concedere le autorizzazioni con il privilegio minimo per i tuoi casi d'uso specifici, poiché sono disponibili per tutti i clienti. AWS Si consiglia

pertanto di ridurre ulteriormente le autorizzazioni definendo policy gestite dal cliente specifiche per i propri casi d'uso.

Non è possibile modificare le autorizzazioni definite nelle politiche gestite. AWS Se AWS aggiorna le autorizzazioni definite in una politica AWS gestita, l'aggiornamento ha effetto su tutte le identità principali (utenti, gruppi e ruoli) a cui è associata la politica. AWS è più probabile che aggiorni una policy AWS gestita quando ne Servizio AWS viene lanciata una nuova o quando diventano disponibili nuove operazioni API per i servizi esistenti.

Per ulteriori informazioni, consultare Policy gestite da AWS nella Guida per l'utente di IAM.

Amazon MQ supporta le seguenti politiche AWS gestite:

- AmazonMQApiFullAccess
- AmazonMQApiReadOnlyAccess
- MQFullAccesso Amazon
- AmazonMQReadOnlyAccess
- AmazonMQServiceRolePolicy

AWS politica gestita: Amazon MQService RolePolicy

Non è possibile allegare `AmazonMQServiceRolePolicy` alle entità IAM. Questa policy è associata a un ruolo collegato ai servizi che consente ad Amazon MQ di eseguire operazioni per tuo conto. Per maggiori informazioni su questa policy di autorizzazione e sulle operazioni che consente di eseguire ad Amazon MQ, consultare the section called “Autorizzazioni del ruolo collegato ai servizi per Amazon MQ”.

Aggiornamenti di Amazon MQ alle politiche AWS gestite

Visualizza i dettagli sugli aggiornamenti delle politiche AWS gestite per Amazon MQ da quando questo servizio ha iniziato a tracciare queste modifiche. Per gli avvisi automatici sulle modifiche

apportate a questa pagina, sottoscrivere il feed RSS nella pagina di Cronologia dei documenti di Amazon MQ.

Modifica	Descrizione	Data
Amazon MQ ha iniziato a monitorare le modifiche	Amazon MQ ha iniziato a tracciare le modifiche per le sue politiche AWS gestite.	5 maggio 2021

Utilizzo di ruoli collegati ai servizi per Amazon MQ

Amazon MQ utilizza ruoli collegati ai servizi AWS Identity and Access Management (IAM). Un ruolo collegato al servizio è un tipo di ruolo IAM univoco collegato direttamente ad Amazon MQ. I ruoli collegati ai servizi sono predefiniti da Amazon MQ e includono tutte le autorizzazioni richieste dal servizio per chiamare altri AWS servizi per tuo conto.

Un ruolo collegato ai servizi semplifica la configurazione di Amazon MQ perché ti permette di evitare l'aggiunta manuale delle autorizzazioni necessarie. Amazon MQ definisce le autorizzazioni del ruolo associato ai servizi e, salvo diversamente definito, solo Amazon MQ può assumere il ruolo. Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere allegata a nessun'altra entità IAM.

È possibile eliminare un ruolo collegato al servizio solo dopo avere eliminato le risorse correlate. Questa procedura protegge le risorse di Amazon MQ perché impedisce la rimozione involontaria delle autorizzazioni di accesso alle risorse.

Per informazioni sugli altri servizi che supportano i ruoli collegati ai servizi, consulta i servizi AWS che funzionano con IAM e cerca i servizi che riportano Yes (Sì) nella colonna Service-Linked Role (Ruolo associato ai servizi). Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Autorizzazioni del ruolo collegato ai servizi per Amazon MQ

Amazon MQ utilizza il ruolo collegato al servizio denominato AWSServiceRoleForAmazonMQ: Amazon MQ utilizza questo ruolo collegato al servizio per chiamare AWS i servizi per tuo conto.

Il ruolo collegato al servizio AWSService RoleForAmazon MQ prevede che i seguenti servizi assumano il ruolo:

- `mq.amazonaws.com`

Amazon MQ utilizza la politica di autorizzazione `AmazonMQServiceRolePolicy`, allegata al ruolo collegato al servizio AWSService RoleForAmazon MQ, per completare le seguenti azioni sulle risorse specificate:

- Operazione: `ec2:CreateVpcEndpoint` sulla risorsa `vpc`.
- Operazione: `ec2:CreateVpcEndpoint` sulla risorsa `subnet`.
- Operazione: `ec2:CreateVpcEndpoint` sulla risorsa `security-group`.
- Operazione: `ec2:CreateVpcEndpoint` sulla risorsa `vpc-endpoint`.
- Operazione: `ec2:DescribeVpcEndpoints` sulla risorsa `vpc`.
- Operazione: `ec2:DescribeVpcEndpoints` sulla risorsa `subnet`.
- Operazione: `ec2:CreateTags` sulla risorsa `vpc-endpoint`.
- Operazione: `logs:PutLogEvents` sulla risorsa `log-group`.
- Operazione: `logs:DescribeLogStreams` sulla risorsa `log-group`.
- Operazione: `logs:DescribeLogGroups` sulla risorsa `log-group`.
- Operazione: `CreateLogStream` sulla risorsa `log-group`.
- Operazione: `CreateLogGroup` sulla risorsa `log-group`.

Quando crei un broker Amazon MQ per RabbitMQ, la policy delle autorizzazioni `AmazonMQServiceRolePolicy` consente ad Amazon MQ di eseguire le seguenti attività per tuo conto.

- Creare un endpoint Amazon VPC per il broker utilizzando Amazon VPC, la sottorete e il gruppo di sicurezza fornito. È possibile utilizzare l'endpoint creato per il broker per connettersi alil broker tramite la console di gestione RabbitMQ, l'API di gestione o in modo programmatico.

- Crea gruppi di log e pubblica i log dei broker su Amazon CloudWatch Logs.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVpcEndpoints"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateVpcEndpoint"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:vpc/*",
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:security-group/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateVpcEndpoint"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:vpc-endpoint/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/AMQManaged": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:*:*:vpc-endpoint/*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": "CreateVpcEndpoint"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteVpcEndpoints"
            ],
            "Resource": "arn:aws:ec2:*:*:vpc-endpoint/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/AMQManaged": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:PutLogEvents",
                "logs:DescribeLogStreams",
                "logs:DescribeLogGroups",
                "logs:CreateLogStream",
                "logs:CreateLogGroup"
            ],
            "Resource": [
                "arn:aws:logs:*:*:log-group:/aws/amazonmq/*"
            ]
        }
    ]
}
```

Per consentire a un'entità IAM (come un utente, un gruppo o un ruolo) di creare, modificare o eliminare un ruolo collegato al servizio è necessario configurare le relative autorizzazioni. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi nella Guida per l'utente di IAM.

Creazione di un ruolo collegato ai servizi per Amazon MQ

Non hai bisogno di creare manualmente un ruolo collegato ai servizi. Quando crei un broker per la prima volta, Amazon MQ crea un ruolo collegato ai servizi per chiamare AWS i servizi per tuo conto. Tutti i broker successivi creati utilizzeranno lo stesso ruolo senza crearne di nuovi.

> **Important**
>
> Questo ruolo collegato ai servizi può apparire nell'account se è stata completata un'operazione in un altro servizio che utilizza le funzionalità supportate dal ruolo. Per ulteriori informazioni, consulta Un nuovo ruolo è apparso nel mio account IAM.

Se elimini questo ruolo collegato al servizio, è possibile ricrearlo seguendo lo stesso processo utilizzato per ricreare il ruolo nell'account.

È possibile utilizzare la console IAM anche per creare un ruolo collegato ai servizi con il caso d'uso Amazon MQ. Nella AWS CLI o nell' AWS API, crea un ruolo collegato al servizio con il nome del servizio. `mq.amazonaws.com` Per ulteriori informazioni, consulta Creazione di un ruolo collegato ai servizi nella Guida per l'utente IAM. Se elimini il ruolo collegato ai servizi, è possibile utilizzare lo stesso processo per crearlo nuovamente.

> **Important**
>
> I ruoli collegati ai servizi vengono creati solo per Amazon MQ for RabbitMQ.

Modifica di un ruolo collegato ai servizi per Amazon MQ

Amazon MQ non consente di modificare il ruolo collegato al servizio AWSService RoleForAmazon MQ. È possibile tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta Modifica di un ruolo collegato al servizio nella Guida per l'utente di IAM.

Eliminazione di un ruolo collegato ai servizi per Amazon MQ

Se non è più necessario utilizzare una funzionalità o un servizio che richiede un ruolo collegato al servizio, ti consigliamo di eliminare il ruolo. In questo modo non sarà più presente un'entità non utilizzata che non viene monitorata e gestita attivamente. Tuttavia, è necessario effettuare la pulizia delle risorse associate al ruolo collegato al servizio prima di poterlo eliminare manualmente.

> **Note**
>
> Se il servizio Amazon MQ utilizza tale ruolo quando tenti di eliminare le risorse, è possibile che l'eliminazione non abbia esito positivo. In questo caso, attendi alcuni minuti e quindi ripeti l'operazione.

Per eliminare le risorse Amazon MQ utilizzate da AWSService RoleForAmazon MQ

- Elimina i tuoi broker Amazon MQ utilizzando Amazon MQ CLI o l' Console di gestione AWS API Amazon MQ. Per ulteriori informazioni sull'eliminazione dei broker, consultare ???.

Per eliminare manualmente il ruolo collegato ai servizi mediante IAM

Utilizza la console IAM AWS CLI, o l' AWS API per eliminare il ruolo collegato al servizio AWSService RoleForAmazon MQ. Per ulteriori informazioni, consulta Eliminazione del ruolo collegato al servizio nella Guida per l'utente di IAM.

Regioni supportate per i ruoli collegati ai servizi di Amazon MQ

Amazon MQ supporta l'utilizzo di ruoli collegati ai servizi in tutte le regioni in cui il servizio è disponibile. Per ulteriori informazioni, consulta AWS Regioni ed endpoint.

Nome della Regione	Identità della Regione	Supporto in Amazon MQ
Stati Uniti orientali (Virginia settentri onale)	us-east-1	Sì
Stati Uniti orientali (Ohio)	us-east-2	Sì
Stati Uniti occidentali (California settentri onale)	us-west-1	Sì
Stati Uniti occidentali (Oregon)	us-west-2	Sì
Asia Pacifico (Mumbai)	ap-south-1	Sì
Asia Pacifico (Osaka)	ap-northeast-3	Sì

Nome della Regione	Identità della Regione	Supporto in Amazon MQ
Asia Pacifico (Seoul)	ap-northeast-2	Sì
Asia Pacifico (Singapore)	ap-southeast-1	Sì
Asia Pacifico (Sydney)	ap-southeast-2	Sì
Asia Pacifico (Tokyo)	ap-northeast-1	Sì
Canada (Centrale)	ca-central-1	Sì
Europa (Francoforte)	eu-central-1	Sì
Europa (Irlanda)	eu-west-1	Sì
Europa (Londra)	eu-west-2	Sì
Europa (Parigi)	eu-west-3	Sì
Sud America (San Paolo)	sa-east-1	Sì
AWS GovCloud (US)	us-gov-west-1	No

Risoluzione dei problemi relativi all'identità e all'accesso di Amazon MQ

Utilizza le informazioni seguenti per diagnosticare e risolvere i problemi comuni che possono verificarsi durante l'utilizzo di Amazon MQ e IAM.

Argomenti

- Non sono autorizzato a eseguire un'operazione in Amazon MQ
- Non sono autorizzato a eseguire iam: PassRole
- Desidero consentire a persone esterne al mio AWS account di accedere alle mie risorse Amazon MQ

Non sono autorizzato a eseguire un'operazione in Amazon MQ

Se ti Console di gestione AWS dice che non sei autorizzato a eseguire un'azione, devi contattare l'amministratore per ricevere assistenza. L'amministratore è colui che ti ha fornito le credenziali di accesso.

L'errore di esempio seguente si verifica quando l'`mateojackson`utente tenta di utilizzare la console per visualizzare i dettagli su un *`widget`* ma non dispone `mq:`*`GetWidget`* delle autorizzazioni.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 mq:GetWidget on resource: my-example-widget
```

In questo caso, Mateo chiede al suo amministratore di aggiornare le policy per poter accedere alla risorsa *`my-example-widget`* mediante l'operazione `mq:`*`GetWidget`*.

Non sono autorizzato a eseguire iam: PassRole

Se si riceve un errore che indica che non si è autorizzati a eseguire l'operazione `iam:PassRole`, è necessario aggiornare le policy per poter passare un ruolo ad Amazon MQ.

Alcuni Servizi AWS consentono di passare un ruolo esistente a quel servizio invece di creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

L'errore di esempio seguente si verifica quando un utente IAM denominato `marymajor` cerca di utilizzare la console per eseguire un'operazione in Amazon MQ. Tuttavia, l'azione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per trasmettere il ruolo al servizio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Desidero consentire a persone esterne al mio AWS account di accedere alle mie risorse Amazon MQ

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per i servizi che supportano politiche basate sulle risorse o liste di controllo degli accessi (ACLs), puoi utilizzare tali politiche per consentire alle persone di accedere alle tue risorse.

Per maggiori informazioni, consulta gli argomenti seguenti:

- Per sapere se Amazon MQ supporta queste caratteristiche, consultare Funzionamento di Amazon MQ con IAM.
- Per scoprire come fornire l'accesso alle risorse di tua proprietà, consulta Fornire l'accesso a un utente IAM in Account AWS un altro Account AWS di tua proprietà nella IAM User Guide.
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta Fornire l'accesso a soggetti Account AWS di proprietà di terze parti nella Guida per l'utente IAM.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta Fornire l'accesso a utenti autenticati esternamente (federazione delle identità) nella Guida per l'utente IAM.
- Per informazioni sulle differenze di utilizzo tra ruoli e policy basate su risorse per l'accesso multi-account, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.

Convalida della conformità per Amazon MQ

I revisori di terze parti valutano la sicurezza e la conformità di Amazon MQ nell'ambito di diversi programmi di AWS conformità. Sono inclusi SOC, PCI e HIPAA.

Per sapere se un Servizio AWS programma rientra nell'ambito di specifici programmi di conformità, consulta Servizi AWS la sezione Scope by Compliance Program Servizi AWS e scegli il programma di conformità che ti interessa. Per informazioni generali, consulta Programmi di AWS conformità Programmi di di .

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Scaricamento dei report in AWS Artifact .

La vostra responsabilità di conformità durante l'utilizzo Servizi AWS è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. Per

ulteriori informazioni sulla responsabilità di conformità durante l'utilizzo Servizi AWS, consulta la Documentazione AWS sulla sicurezza.

Resilienza in Amazon MQ

L'infrastruttura AWS globale è costruita attorno a AWS regioni e zone di disponibilità. AWS Le regioni forniscono più zone di disponibilità fisicamente separate e isolate, collegate con reti a bassa latenza, ad alto throughput e altamente ridondanti. Con le zone di disponibilità, puoi progettare e gestire applicazioni e database che eseguono automaticamente il failover tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

Per ulteriori informazioni su AWS regioni e zone di disponibilità, consulta Global Infrastructure.AWS

Sicurezza dell'infrastruttura in Amazon MQ

In quanto servizio gestito, Amazon MQ è protetto dalla sicurezza di rete AWS globale. Per informazioni sui servizi AWS di sicurezza e su come AWS protegge l'infrastruttura, consulta AWS Cloud Security. Per progettare il tuo AWS ambiente utilizzando le migliori pratiche per la sicurezza dell'infrastruttura, vedi Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

Utilizzi chiamate API AWS pubblicate per accedere ad Amazon MQ attraverso la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Best practice di sicurezza per Amazon MQ

I seguenti modelli di progettazione possono migliorare la sicurezza del broker Amazon MQ.

Argomenti

- Preferire broker senza accesso pubblico
- Configurare sempre una mappa di autorizzazione
- Bloccare i protocolli non necessari con i gruppi di sicurezza VPC

Per maggiori informazioni su come Amazon MQ crittografa i dati e per un elenco dei protocolli supportati, consultare Protezione dei dati.

Preferire broker senza accesso pubblico

I broker creati senza accessibilità pubblica sono accessibili solo dal VPC. Ciò riduce notevolmente la suscettibilità del broker agli attacchi Distributed Denial of Service (DDoS) provenienti dalla rete Internet pubblica. Per ulteriori informazioni, consulta Come prepararsi agli attacchi DDo S riducendo la superficie di attacco sul AWS Security Blog.

Configurare sempre una mappa di autorizzazione

Poiché ActiveMQ non dispone di una mappa di autorizzazione configurata per impostazione predefinita, qualsiasi utente autenticato è in grado di eseguire qualsiasi azione sul broker. Pertanto, una best practice prevede di limitare le autorizzazioni per gruppo. Per ulteriori informazioni, consulta `authorizationEntry`.

> **Important**
>
> Se si specifica una mappa di autorizzazione che non include il gruppo `activemq-webconsole`, non è possibile utilizzare ActiveMQ Web Console perché il gruppo non è autorizzato a inviare messaggi o ricevere messaggi dalil broker Amazon MQ.

Bloccare i protocolli non necessari con i gruppi di sicurezza VPC

Per migliorare la sicurezza dei broker privati, devi limitare le connessioni di protocolli e porte non necessari configurando correttamente il tuo Amazon VPC Security Group. Ad esempio, per limitare l'accesso alla maggior parte dei protocolli OpenWire e consentire l'accesso alla console Web, puoi consentire l'accesso solo a 61617 e 8162. Ciò limita l'esposizione bloccando i protocolli non utilizzati OpenWire e consentendo al contempo il normale funzionamento della console Web.

Consenti solo le porte dei protocolli che stai utilizzando.

- AMQP: 5671
- MQTT: 8883
- OpenWire: 61617
- STOMP: 61614

- WebSocket: 61619

Per ulteriori informazioni, consultare:

- Gruppi di sicurezza per VPC
- Gruppo di sicurezza predefinito per VPC
- Utilizzo dei gruppi di sicurezza

Registrazione e monitoraggio dei broker Amazon MQ

Il monitoraggio è una parte importante per mantenere l'affidabilità, la disponibilità e le prestazioni delle soluzioni. AWS È necessario raccogliere i dati di monitoraggio da tutte le parti della AWS soluzione in modo da poter eseguire più facilmente il debug di un errore multipunto, se si verifica. AWS fornisce diversi strumenti per monitorare le risorse Amazon MQ e rispondere a potenziali incidenti:

Puoi utilizzarlo CloudWatch per visualizzare e analizzare i parametri per il tuo broker Amazon MQ. Puoi visualizzare e analizzare le metriche del broker dalla CloudWatch console AWS CLI, dal o dal. CloudWatch AWS CLI CloudWatch i parametri per Amazon MQ vengono interrogati automaticamente dal broker e quindi aggiornati ogni minuto. CloudWatch Per i broker ActiveMQ CloudWatch , monitora solo le prime 1000 destinazioni. Per i broker RabbitMQ, CloudWatch monitora solo le prime 500 destinazioni, ordinate per numero di consumatori.

Per un elenco completo di parametri di Amazon MQ, consultare CloudWatch Parametri disponibili Amazon MQ per i broker ActiveMQ.

Per informazioni sulla creazione di un CloudWatch allarme per una metrica, consulta Create or Edit a CloudWatch Alarm nella Amazon CloudWatch User Guide.

Accesso ai CloudWatch parametri per Amazon MQ

Puoi accedere alle CloudWatch metriche utilizzando l'API Console di gestione AWS AWS CLI, e.

Potresti voler accedere alle CloudWatch metriche senza utilizzare il. Console di gestione AWS

Per accedere ai parametri di Amazon MQ utilizzando AWS CLI, usa il `get-metric-statistics` comando. Per ulteriori informazioni, consulta Get Statistics for a Metric nella Amazon CloudWatch User Guide.

Per accedere ai parametri di Amazon MQ utilizzando l' CloudWatch API, utilizza l'`GetMetricStatistics`azione. Per ulteriori informazioni, consulta Get Statistics for a Metric nella Amazon CloudWatch User Guide.

Accesso alle CloudWatch metriche utilizzando il Console di gestione AWS

L'esempio seguente mostra come accedere ai CloudWatch parametri per Amazon MQ utilizzando Console di gestione AWS.Se hai già effettuato l'accesso alla console Amazon MQ, nella pagina Dettagli del broker, scegli Azioni, Visualizza CloudWatch metriche.

1. Accedi alla console CloudWatch.
2. Nel pannello di navigazione, scegli Metrics (Parametri).
3. Seleziona lo spazio dei nomi del parametro AmazonMQ.
4. Seleziona una delle dimensioni parametro seguenti:

   - Parametri broker
   - Queue Metrics by Broker (Parametri coda per broker)
   - Topic Metrics by Broker (Parametri argomento per broker)

   In questo esempio, è selezionata Broker Metrics (Parametri broker).
5. Puoi ora quindi esaminare i parametri Amazon MQ:

   - Per ordinare i parametri, utilizza l'intestazione della colonna.
   - Per creare il grafico del parametro, seleziona la casella di controllo accanto al parametro.
   - Per filtrare in base a un parametro, scegli il nome del parametro e quindi Add to search (Aggiungi alla ricerca).

CloudWatch Parametri disponibili Amazon MQ per i broker ActiveMQ

Parametri Amazon MQ per ActiveMQ

Metrica	Unità	Description
`AmqpMaximumConnections`	Conteggio	Il numero massimo di clienti che puoi connettere al tuo broker utilizzando AMQP.

Metrica	Unità	Description
		Per ulteriori informazioni sulle quote di connessione, consulta Quotas in Amazon MQ.
`BurstBalance`	Percentuale	La percentuale di crediti per il burst rimanenti sul volume Amazon EBS utilizzato per mantenere i dati dei messaggi per i broker ottimizzati per la velocità effettiva. Se questo saldo raggiunge lo zero, le IOPS fornite dal volume Amazon EBS diminuiranno fino a quando il saldo di burst non si ricaricherà. Per ulteriori informazioni su come funzionano i saldi di burst in Amazon EBS, consulta l'argomento relativo ai crediti di I/O e prestazioni di burst.

Metrica	Unità	Description
`CpuCreditBalance`	Crediti (vCPU/minuti)	**Important** Questo parametro è disponibile solo per il tipo di istanza broker `mq.t2.micro` . I parametri di credito CPU sono disponibili solo a intervalli di 5 minuti. Il numero di crediti CPU ottenuti, che un'istanza ha accumulato da quando è stata lanciata o avviata (incluso il numero di crediti di lancio). Il saldo del credito è disponibile affinché l'istanza broker lo spenda per andare oltre l'utilizzo di base della CPU. I crediti vengono accumulati nel saldo del credito dopo che sono stati ottenuti e vengono rimossi dal saldo del credito una volta spesi. Il saldo del credito ha un limite massimo. Una volta raggiunto il limite, i nuovi crediti guadagnati vengono scartati.

Metrica	Unità	Description
`CpuUtilization`	Percentuale	Percentuale delle unità di elaborazione di Amazon EC2 assegnate e attualmente utilizzate dal broker.
`CurrentConnectionsCount`	Conteggio	Numero di connessioni attualmente attive sul broker corrente.
`EstablishedConnectionsCount`	Conteggio	Il numero totale di connessioni, attive e inattive, che sono state stabilite nel broker.
`HeapUsage`	Percentuale	Percentuale del limite di memoria ActiveMQ JVM utilizzata attualmente dal broker.
`InactiveDurableTopicSubscribersCount`	Conteggio	Il numero di sottoscrizioni ad argomenti durevoli inattive, fino a un massimo di 2.000.
`JobSchedulerStorePercentUsage`	Percentuale	La percentuale di spazio su disco utilizzata dall'archivio del sistema di pianificazione delle attività.
`JournalFilesForFastRecovery`	Conteggio	Numero di file di registro che vengono riprodotti dopo una chiusura senza errori.
`JournalFilesForFullRecovery`	Conteggio	Numero di file di registro che vengono riprodotti dopo una chiusura con errori.

Metrica	Unità	Description
`MqttMaximumConnections`	Conteggio	Il numero massimo di clienti che puoi connettere al tuo broker utilizzando MQTT. Per ulteriori informazioni sulle quote di connessione, consulta Quotas in Amazon MQ.
`NetworkConnectorConnectionCount`	Conteggio	Il numero di nodi collegati al broker in una rete di broker che utilizza. NetworkConnector
`NetworkIn`	Byte	Volume di traffico in entrata per il broker.
`NetworkOut`	Byte	Volume di traffico in uscita per il broker.
`OpenTransactionCount`	Conteggio	Numero totale delle transazioni in esecuzione.
`OpenwireMaximumConnections`	Conteggio	Il numero massimo di client che puoi utilizzare OpenWire per connetterti al tuo broker. Per ulteriori informazioni sulle quote di connessione, consulta Quotas in Amazon MQ.

Metrica	Unità	Description
StompMaximumConnections	Conteggio	Il numero massimo di clienti che puoi connettere al tuo broker utilizzando STOMP. Per ulteriori informazioni sulle quote di connessione, consulta Quotas in Amazon MQ.
StorePercentUsage	Percentuale	Percentuale utilizzata dal limite di storage. Se raggiunge 100, il broker rifiuta i messaggi.
TempPercentUsage	Percentuale	Percentuale di storage temporaneo disponibile utilizzata dai messaggi non persistenti.
TotalConsumerCount	Conteggio	Numero di consumer di messaggi iscritti alle destinazioni sul broker corrente.
TotalMessageCount	Conteggio	Numero di messaggi archiviati sul broker.
TotalProducerCount	Conteggio	Numero di produttori di messaggi attivi nelle destinazioni sul broker corrente.
VolumeReadOps	Conteggio	Numero di operazioni di lettura eseguite sul volume Amazon EBS.
VolumeWriteOps	Conteggio	Numero di operazioni di scrittura eseguite sul volume Amazon EBS.

Metrica	Unità	Description
`WsMaximumConnections`	Conteggio	Il numero massimo di client che puoi utilizzare per connetterti al tuo broker WebSocket. Per ulteriori informazioni sulle quote di connessione, consulta Quotas in Amazon MQ.

Dimensioni per i parametri del broker ActiveMQ

Dimensione	Description
`Broker`	Il nome del broker **Note** Un broker a istanza singola ha il suffisso -1. Un active/standby broker per l'alta disponibilità ha i suffissi -1 e -2 per la sua coppia ridondante.

Destinazione ActiveMQ per i parametri (coda e argomento)

Important

Le seguenti metriche includono i conteggi al minuto per il periodo di votazione. CloudWatch

- `EnqueueCount`
- `ExpiredCount`
- `DequeueCount`
- `DispatchCount`
- `InFlightCount`

Ad esempio, in un periodo CloudWatch di cinque minuti,`EnqueueCount` ha cinque valori di conteggio, ciascuno per una porzione di un minuto del periodo. Le statistiche `Maximum` e `Minimum` offrono il valore minimo e massimo al minuto durante il periodo specificato.

Metrica	Unità	Description
`ConsumerCount`	Conteggio	Numero di consumer iscritti alla destinazione.
`EnqueueCount`	Conteggio	Numero di messaggi inviati alla destinazione, al minuto.
`EnqueueTime`	Tempo (millisecondi)	La end-to-end latenza dal momento in cui un messaggio arriva a un broker fino alla consegna a un consumatore. **Note** `EnqueueTime` non misura la end-to-end latenza dal momento in cui un messaggio viene inviato da un produttore fino a quando raggiunge il broker, né la latenza tra il momento in cui un messaggio viene ricevuto da un broker fino a quando non viene riconosciuto dal broker. Piuttosto , `EnqueueTime` è il numero di milliseco

Metrica	Unità	Description
		ndi dal momento in cui un messaggio viene ricevuto dal broker fino a quando non viene consegnato correttam ente a un consumato re.
`ExpiredCount`	Conteggio	Numero di messaggi non recapitati perché scaduti, al minuto.
`DispatchCount`	Conteggio	Numero di messaggi inviati ai consumer, al minuto.
`DequeueCount`	Conteggio	Numero di messaggi confermati dai consumer, al minuto.
`InFlightCount`	Conteggio	Numero di messaggi inviati ai consumer che non sono stati confermati.
`ReceiveCount`	Conteggio	Il numero di messaggi che sono stati ricevuti dal broker remoto per un connettore di rete duplex.
`MemoryUsage`	Percentuale	Percentuale del limite di memoria utilizzata attualmente dalla destinazione.
`ProducerCount`	Conteggio	Numero di produttori per la destinazione.

Metrica	Unità	Description
`QueueSize`	Conteggio	Numero dei messaggi in coda. **Important** Questo parametro si applica solo alle code.
`TotalEnqueueCount`	Conteggio	Numero totale di messaggi inviati al broker.
`TotalDequeueCount`	Conteggio	Numero totale di messaggi che sono stati utilizzati dai client.

Note

I parametri `TotalEnqueueCount` e `TotalDequeueCount` includono messaggi per argomenti di consulenza. Per ulteriori informazioni sui messaggi degli argomenti di consulenza, consultare la documentazione di ActiveMQ.

Dimensioni per i parametri di destinazione ActiveMQ (coda e argomento)

Dimensione	Description
`Broker`	Il nome del broker. **Note** Un broker a istanza singola ha il suffisso `-1`. Un active/standby broker per l'alta disponibilità ha i suffissi `-1` e `-2` la coppia ridondante.

Dimensione	Description
`Topic` o `Queue`	Nome dell'argomento o della coda.
`NetworkConnector`	Il nome del connettore di rete.

CloudWatch Metriche disponibili per i broker Amazon MQ for RabbitMQ

Parametri del broker RabbitMQ

Metrica	Unità	Description
`ExchangeCount`	Conteggio	Il numero totale di scambi configurati sul broker.
`QueueCount`	Conteggio	Il numero totale di code configurate sul broker.
`ConnectionCount`	Conteggio	Il numero totale di connessioni stabilite nel broker.
`ChannelCount`	Conteggio	Il numero totale di canali stabiliti nel broker.
`ConsumerCount`	Conteggio	Il numero totale di consumatori collegati al broker.
`MessageCount`	Conteggio	Il numero totale di messaggi in coda. **Note** Il numero prodotto è la somma totale dei

Metrica	Unità	Description
		messaggi pronti e sconosciuti sul broker.
`MessageReadyCount`	Conteggio	Il numero totale di messaggi pronti in coda.
`MessageUnacknowledgedCount`	Conteggio	Il numero totale di messaggi non confermati in coda.
`PublishRate`	Conteggio	La frequenza con cui i messaggi vengono pubblicati al broker. Il numero prodotto rappresenta il numero di messaggi al secondo al momento del campionamento.
`ConfirmRate`	Conteggio	La velocità alla quale il server RabbitMQ conferma i messaggi pubblicati. È possibile confrontare questo parametro con `PublishRate` per capire meglio come si sta comportando il broker. Il numero prodotto rappresenta il numero di messaggi al secondo al momento del campionamento.

Metrica	Unità	Description
`AckRate`	Conteggio	La frequenza con cui i messaggi vengono riconosciuti dai consumatori. Il numero prodotto rappresenta il numero di messaggi al secondo al momento del campionamento.
`SystemCpuUtilization`	Percentuale	Percentuale delle unità di elaborazione di Amazon EC2 assegnate e attualmente utilizzate dal broker. Per le implementazioni cluster, questo valore rappresenta l'aggregato di tutti e tre i parametri corrispondenti dei nodi RabbitMQ.
`RabbitMQMemLimit`	Byte	Il limite RAM per un broker RabbitMQ. Per le implementazioni cluster, questo valore rappresenta l'aggregato di tutti e tre i parametri corrispondenti dei nodi RabbitMQ.
`RabbitMQMemUsed`	Byte	Il volume della RAM utilizzata da un broker RabbitMQ. Per le implementazioni cluster, questo valore rappresenta l'aggregato di tutti e tre i parametri corrispondenti dei nodi RabbitMQ.

Metrica	Unità	Description
RabbitMQDiskFreeLimit	Byte	Il limite del disco per un broker RabbitMQ. Per le implementazioni cluster, questo valore rappresenta l'aggregato di tutti e tre i parametri corrispondenti dei nodi RabbitMQ. Questo parametro è diverso in base alla dimensione dell'istanza.
RabbitMQDiskFree	Byte	Il volume totale dello spazio libero disponibile su disco in un broker RabbitMQ. Quando l'utilizzo del disco supera il limite, il cluster bloccherà tutte le connessioni del produttore. Per le implementazioni cluster, questo valore rappresenta l'aggregato di tutti e tre i parametri corrispondenti dei nodi RabbitMQ.
RabbitMQFdUsed	Conteggio	Il numero di descrittori di file utilizzati. Per le implementazioni cluster, questo valore rappresenta l'aggregato di tutti e tre i parametri corrispondenti dei nodi RabbitMQ.
RabbitMQIOReadAverageTime	Conteggio	Il tempo medio (in millisecondi) impiegato da RabbitMQ per eseguire un'operazione di lettura. Il valore è proporzionale alla dimensione del messaggio.

Metrica	Unità	Description
`RabbitMQIOWriteAverageTime`	Conteggio	Il tempo medio (in millisecondi) impiegato da RabbitMQ per eseguire un'operazione di scrittura. Il valore è proporzionale alla dimensione del messaggio.

Dimensioni per i parametri del broker RabbitMQ

Dimensione	Description
`Broker`	Il nome del broker.

Parametri del nodo RabbitMQ

Metrica	Unità	Description
`SystemCpuUtilization`	Percentuale	Percentuale delle unità di elaborazione di Amazon EC2 assegnate e attualmente utilizzate dal broker.
`RabbitMQMemLimit`	Byte	Il limite RAM per un nodo RabbitMQ.
`RabbitMQMemUsed`	Byte	Il volume della RAM utilizzata da un nodo RabbitMQ. Quando l'utilizzo della memoria supera il limite, il cluster bloccherà tutte le connessioni del produttore.
`RabbitMQDiskFreeLimit`	Byte	Il limite del disco per un nodo RabbitMQ. Questo

Metrica	Unità	Description
		parametro è diverso in base alla dimensione dell'istanza.
`RabbitMQDiskFree`	Byte	Il volume totale dello spazio libero disponibile su disco in un nodo RabbitMQ. Quando l'utilizzo del disco supera il limite, il cluster bloccherà tutte le connessioni del produttore.
`RabbitMQFdUsed`	Conteggio	Il numero di descrittori di file utilizzati.

Le dimensioni per i parametri dei nodi RabbitMQ

Dimensione	Description
`Node`	Il nome del nodo. **Note** Un nome del nodo è composto da due parti: un prefisso (di solito `rabbit`) e un nome dell'host. Ad esempio, `rabbit@ip-10-0-0-230.us-west-2.compute.internal` è un nome del nodo con il prefisso `rabbit` e il nome dell'host `ip-10-0-0-230.us-west-2.compute.internal` .
`Broker`	Il nome del broker.

Parametri della coda RabbitMQ

Metrica	Unità	Description
`ConsumerCount`	Conteggio	Il numero di consumatori iscritti alla coda.
`MessageReadyCount`	Conteggio	Il numero di messaggi attualmente disponibili da inviare.
`MessageUnacknowledgedCount`	Conteggio	Il numero di messaggi per i quali il server è in attesa di conferma.
`MessageCount`	Conteggio	Il numero totale di `MessageReadyCount` e `MessageUnacknowledgedCount` (detta anche profondità della coda).

Dimensioni per i parametri della coda RabbitMQ

> **Note**
>
> Amazon MQ per RabbitMQ non pubblicherà parametri per host virtuali e code con nomi contenenti spazi vuoti, tabulazioni o altri caratteri non ASCII.
> Per ulteriori informazioni sui nomi delle dimensioni, consulta Dimension in Amazon CloudWatch API Reference.

Dimensione	Description
`Queue`	Il nome della coda.
`VirtualHost`	Nome dell'host virtuale.

Dimensione	Description
`Broker`	Il nome del broker.

Metriche di rete RabbitMQ

Metrica	Unità	Description
`NetworkOut`	Byte	Il numero di byte inviati dall'istanza su tutte le interfacce di rete. Questo parametro identifica il volume del traffico di rete in uscita da una singola istanza. Il numero segnalato è il numero di byte inviati durante il periodo. Se utilizzi il monitoraggio di base (5 minuti) e la statistica è Sum (Somma), puoi dividere questo numero per 300 per trovare i byte/secondo. Se hai il monitoraggio dettagliato (1 minuto) e la statistica è Sum (Somma), dividi per 60. È inoltre possibile utilizzare la funzione matematica CloudWatch metrica `DIFF_TIME` per trovare i byte al secondo. Ad esempio, se hai rappresentato graficamente CloudWatch asm1, la formula `m1/(DIFF_TIME(m1))` matematica restituisce la metrica NetworkOut in byte/secondo. Per ulteriori informazioni `DIFF_TIME` e altre funzioni matematiche metriche, vedere Uso della matematica metrica. Statistiche significative: somma, media, minimo, massimo
`NetworkIn`	Byte	Il numero di byte ricevuti dall'istanza su tutte le interfacce di rete. Questo parametro identifica il volume del traffico di rete in entrata in una singola istanza. Il numero segnalato è il numero di byte ricevuti durante il periodo. Se utilizzi il monitoraggio di base (5 minuti) e la statistica è Sum (Somma), puoi dividere questo numero per 300 per trovare i byte/secondo. Se hai il monitoraggio dettagliato (1 minuto) e la statistica è Sum (Somma), dividi per 60. È inoltre possibile utilizzare la funzione matematica CloudWatch metrica

Metrica	Unità	Description
		`DIFF_TIME` per trovare i byte al secondo. Ad esempio, se hai rappresentato graficamente CloudWatch as`m1`, la formula `m1/(DIFF_TIME(m1))` matematica metrica restituisce la metrica NetworkIn in byte/secondo. Per ulteriori informazioni `DIFF_TIME` e altre funzioni matematiche metriche, vedere Uso della matematica metrica. Statistiche significative: somma, media, minimo, massimo

Dimensioni per i broker RabbitMQ

Dimensione	Description
`BrokerId`	ID del broker

Configurazione dei log di Amazon MQ per RabbitMQ

Quando abiliti CloudWatch la registrazione per i tuoi broker RabbitMQ, Amazon MQ utilizza un ruolo collegato al servizio per pubblicare i log generali. CloudWatch Se non esiste alcun ruolo collegato al servizio di Amazon MQ quando crei per la prima volta un broker, Amazon MQ ne creerà automaticamente uno. Tutti i broker RabbitMQ successivi utilizzeranno lo stesso ruolo collegato al servizio su cui pubblicare i log. CloudWatch

Per ulteriori informazioni sui ruoli collegati ai servizi, vedere Utilizzo dei ruoli collegati ai servizi nella Guida per l'utente.AWS Identity and Access Management Per ulteriori informazioni sull'utilizzo dei ruoli collegati al servizio da parte di Amazon MQ, consultare the section called “Uso di ruoli collegati ai servizi”.

Registrazione delle chiamate API Amazon MQ tramite AWS CloudTrail

Amazon MQ è integrato con AWS CloudTrail, un servizio che fornisce una registrazione delle chiamate Amazon MQ effettuate da un utente, ruolo o AWS servizio. CloudTrail acquisisce le

chiamate API relative ai broker e alle configurazioni Amazon MQ come eventi, incluse le chiamate dalla console Amazon MQ e le chiamate in codice da Amazon MQ. APIs Per ulteriori informazioni in merito CloudTrail, consulta la Guida per l'AWS CloudTrail utente.

> **Note**
>
> CloudTrail non registra le chiamate API relative alle operazioni di ActiveMQ (ad esempio, l'invio e la ricezione di messaggi) o alla console Web ActiveMQ. Per registrare le informazioni relative alle operazioni di ActiveMQ, puoi configurare Amazon MQ per pubblicare log generali e di audit su Amazon Logs. CloudWatch

Utilizzando le informazioni CloudTrail raccolte, puoi identificare una richiesta specifica a un'API Amazon MQ, l'indirizzo IP del richiedente, l'identità del richiedente, la data e l'ora della richiesta e così via. Se configuri un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3. Se non configuri un percorso, puoi visualizzare gli eventi più recenti nella cronologia degli eventi nella CloudTrail console. Per ulteriori informazioni, consultare Panoramica per la creazione di un percorso nella Guida per l'utente di AWS CloudTrail.

Informazioni su Amazon MQ in CloudTrail

Quando crei il tuo AWS account, CloudTrail è abilitato. Quando si verifica un'attività di evento Amazon MQ supportata, viene registrata in un CloudTrail evento con altri eventi di AWS servizio nella cronologia degli eventi. Puoi visualizzare, ricercare e scaricare eventi recenti per l'account AWS . Per ulteriori informazioni, consulta Visualizzazione degli eventi con cronologia degli CloudTrail eventi nella Guida AWS CloudTrail per l'utente.

Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Puoi creare un percorso per tenere un registro continuo degli eventi nel tuo AWS account. Per impostazione predefinita, quando si crea un percorso utilizzando il Console di gestione AWS, il percorso si applica a tutte le AWS regioni. Il percorso registra gli eventi da tutte le AWS regioni e fornisce i file di log al bucket Amazon S3 specificato. Puoi anche configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consulta gli argomenti seguenti nella Guida per l'utente AWS CloudTrail :

- CloudTrail Servizi e integrazioni supportati
- Configurazione delle notifiche Amazon SNS per CloudTrail
- Ricezione di file di CloudTrail registro da più regioni

- Ricezione di file di CloudTrail registro da più account

Amazon MQ supporta la registrazione sia dei parametri di richiesta che delle risposte per APIs quanto segue come eventi nei file di CloudTrail registro:

- `CreateConfiguration`
- `DeleteBroker`
- `DeleteUser`
- `RebootBroker`
- `UpdateBroker`

> **Note**
>
> RebootBroker i file di registro vengono registrati al riavvio del broker. Durante la finestra di manutenzione, il servizio si riavvia automaticamente e i file di RebootBroker registro non vengono registrati.

> **Important**
>
> Per i seguenti `GET` metodi APIs, i parametri della richiesta vengono registrati, ma le risposte vengono oscurate:
>
> - `DescribeBroker`
> - `DescribeConfiguration`
> - `DescribeConfigurationRevision`
> - `DescribeUser`
> - `ListBrokers`
> - `ListConfigurationRevisions`
> - `ListConfigurations`
> - `ListUsers`
>
> Per quanto segue APIs, i parametri `data` e `password` request sono nascosti da asterischi (): `***`

- `CreateBroker` (POST)
- `CreateUser` (POST)
- `UpdateConfiguration` (PUT)
- `UpdateUser` (PUT)

Ogni evento o voce di log contiene informazioni sul richiedente. Queste informazioni consentono di determinare quanto segue:

- La richiesta è stata effettuata con le credenziali utente o root?
- La richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato?
- La richiesta è stata effettuata da un altro AWS servizio?

Per ulteriori informazioni, consulta CloudTrailUserIdentity Element nella Guida per l'AWS CloudTrail utente.

Esempio: voci del file di log di Amazon MQ

Un trail è una configurazione che consente la distribuzione di eventi come file di log al bucket Amazon S3 specificato. CloudTrail i file di registro contengono una o più voci di registro.

Un evento rappresenta una singola richiesta da qualsiasi origine e include informazioni relative alla richiesta a un'API Amazon MQ, l'indirizzo IP del richiedente, l'identità del richiedente, la data e l'ora della richiesta e così via.

L'esempio seguente mostra una voce di CloudTrail registro per una chiamata `CreateBroker`API.

> **Note**
>
> Poiché i file di CloudTrail registro non sono uno stack trace ordinato di public APIs, non elencano le informazioni in un ordine specifico.

```
{
    "eventVersion": "1.06",
    "userIdentity": {
```

```
        "type": "IAMUser",
        "principalId": "AKIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/AmazonMqConsole",
        "accountId": "111122223333",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "userName": "AmazonMqConsole"
    },
    "eventTime": "2018-06-28T22:23:46Z",
    "eventSource": "amazonmq.amazonaws.com",
    "eventName": "CreateBroker",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "203.0.113.0",
    "userAgent": "PostmanRuntime/7.1.5",
    "requestParameters": {
        "engineVersion": "5.15.9",
        "deploymentMode": "ACTIVE_STANDBY_MULTI_AZ",
        "maintenanceWindowStartTime": {
            "dayOfWeek": "THURSDAY",
            "timeOfDay": "22:45",
            "timeZone": "America/Los_Angeles"
        },
        "engineType": "ActiveMQ",
        "hostInstanceType": "mq.m5.large",
        "users": [
            {
                "username": "MyUsername123",
                "password": "***",
                "consoleAccess": true,
                "groups": [
                    "admins",
                    "support"
                ]
            },
            {
                "username": "MyUsername456",
                "password": "***",
                "groups": [
                    "admins"
                ]
            }
        ],
        "creatorRequestId": "1",
        "publiclyAccessible": true,
        "securityGroups": [
```

```
            "sg-a1b234cd"
        ],
        "brokerName": "MyBroker",
        "autoMinorVersionUpgrade": false,
        "subnetIds": [
            "subnet-12a3b45c",
            "subnet-67d8e90f"
        ]
    },
    "responseElements": {
        "brokerId": "b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9",
        "brokerArn": "arn:aws:mq:us-
east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9"
    },
    "requestID": "a1b2c345-6d78-90e1-f2g3-4hi56jk7l890",
    "eventID": "a12bcd3e-fg45-67h8-ij90-12k34d5l16mn",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```


Configurazione dei log di Amazon MQ per ActiveMQ

Per consentire ad Amazon MQ di pubblicare i log su CloudWatch Logs, devi aggiungere un'autorizzazione al tuo utente Amazon MQ e configurare anche una policy basata sulle risorse per Amazon MQ prima di creare o riavviare il broker.

> **Note**
>
> Quando si attivano i registri e si pubblicano messaggi dalla console Web ActiveMQ, il contenuto del messaggio viene inviato CloudWatch e visualizzato nei registri.

Di seguito vengono descritti i passaggi per configurare CloudWatch i log per i broker ActiveMQ.

Argomenti

- Comprensione della struttura di registrazione nei log CloudWatch
- Aggiunta dell'autorizzazione CreateLogGroup all'utente Amazon MQ
- Configurare una policy basata sulle risorse per Amazon MQ

- Prevenzione del confused deputy tra servizi

Comprensione della struttura di registrazione nei log CloudWatch

È possibile abilitare la registrazione generale e di controllo quando si configurano le impostazioni avanzate del broker, quando si crea un broker o quando si modifica un broker.

La registrazione generale abilita il livello di `INFO` registrazione predefinito (la `DEBUG` registrazione non è supportata) e viene pubblicata `activemq.log` su un gruppo di log del tuo account. CloudWatch Il formato del gruppo di log è simile al seguente:

```
/aws/amazonmq/broker/b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9/general
```

La registrazione di controllo consente la registrazione delle azioni di gestione eseguite utilizzando JMX o utilizzando la console Web ActiveMQ e le pubblica `audit.log` in un gruppo di log dell'account. CloudWatch Il formato del gruppo di log è simile al seguente:

```
/aws/amazonmq/broker/b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9/audit
```

A seconda che si disponga di un broker a istanza singola o un broker attivo/in standby, Amazon MQ crea uno o due flussi di registri all'interno di ogni gruppo di registri. Il formato dei flussi di log è simile al seguente.

```
activemq-b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.log
                activemq-b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-2.log
```

I suffissi `-1` e `-2` denotano singole istanze broker. Per ulteriori informazioni, consulta Working with Log Groups and Log Streams nella Amazon CloudWatch Logs User Guide.

Aggiunta dell'autorizzazione **`CreateLogGroup`** all'utente Amazon MQ

Per consentire ad Amazon MQ di creare un gruppo di log CloudWatch Logs, devi assicurarti che l'utente che crea o riavvia il broker disponga dell'`logs:CreateLogGroup`autorizzazione.

> **Important**
>
> Se non aggiungi l'autorizzazione `CreateLogGroup` all'utente Amazon MQ prima che l'utente crei o riavvi il broker, Amazon MQ non crea il gruppo di registri.

L'esempio seguente della policy basata su IAM concede l'autorizzazioni `logs:CreateLogGroup` per gli utenti ai quali è associata questa policy.

JSON

```
{
                    "Version":"2012-10-17",
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Action": "logs:CreateLogGroup",
                            "Resource": "arn:aws:logs:*:*:log-group:/aws/
amazonmq/*"
                        }
                    ]
                    }
```

> **Note**
>
> Qui il termine utente si riferisce agli utenti IAM e non agli utenti Amazon MQ, che vengono creati quando viene configurato un nuovo broker. Per ulteriori informazioni sulla configurazione degli utenti e delle policy IAM, fare riferimento alla Panoramica della gestione delle identità della Guida per l'utente di IAM.

Per ulteriori informazioni, `CreateLogGroup` consulta Amazon CloudWatch Logs API Reference.

Configurare una policy basata sulle risorse per Amazon MQ

> **Important**
>
> Se non configuri una policy basata sulle risorse per Amazon MQ, il broker non può pubblicare i log su Logs. CloudWatch

Per consentire ad Amazon MQ di pubblicare i log nel tuo gruppo di log CloudWatch Logs, configura una policy basata sulle risorse per consentire ad Amazon MQ di accedere alle seguenti azioni dell'API Logs: CloudWatch

- `CreateLogStream`— Crea un flusso di CloudWatch log di Logs per il gruppo di log specificato.
- `PutLogEvents`— Fornisce gli eventi al flusso di log di CloudWatch Logs specificato.

La seguente politica basata sulle risorse concede l'autorizzazione a e a. `logs:CreateLogStream` `logs:PutLogEvents` AWS

JSON

```
{
                          "Version":"2012-10-17",
                          "Statement": [
                              {
                                  "Effect": "Allow",
                                  "Principal": { "Service":
 "mq.amazonaws.com" },
                                  "Action": [ "logs:CreateLogStream",
 "logs:PutLogEvents" ],
                                  "Resource": "arn:aws:logs:*:*:log-group:/aws/
amazonmq/*"
                              }
                          ]
                      }
```

Questa politica basata sulle risorse deve essere configurata utilizzando il AWS CLI comando seguente. Nell'esempio, sostituire *us-east-1* con le tue informazioni.

```
aws --region us-east-1 logs put-resource-policy --policy-name AmazonMQ-logs \
                      --policy-document "{\"Version\": \"2012-10-17\", \"Statement\":
[{ \"Effect\": \"Allow\", \"Principal\": { \"Service\": \"mq.amazonaws.com\" },
                      \"Action\": [\"logs:CreateLogStream\", \"logs:PutLogEvents\"],
 \"Resource\": \"arn:aws:logs:*:*:log-group:\/aws\/amazonmq\/*\" }]}"
```

> **Note**
>
> Poiché questo esempio utilizza il `/aws/amazonmq/` prefisso, è necessario configurare la politica basata sulle risorse solo una volta per account e per regione. AWS

Prevenzione del confused deputy tra servizi

Il problema confused deputy è un problema di sicurezza in cui un'entità che non dispone dell'autorizzazione per eseguire un'azione può costringere un'entità maggiormente privilegiata a eseguire l'azione. Nel AWS, l'impersonificazione tra servizi può portare al confuso problema del vice. La rappresentazione tra servizi può verificarsi quando un servizio (il servizio chiamante) effettua una chiamata a un altro servizio (il servizio chiamato). Il servizio chiamante può essere manipolato per utilizzare le proprie autorizzazioni e agire sulle risorse di un altro cliente, a cui normalmente non avrebbe accesso. Per evitare che ciò accada, AWS mette a disposizione strumenti che consentono di proteggere i dati relativi a tutti i servizi con responsabili del servizio a cui è stato concesso l'accesso alle risorse del vostro account.

Ti consigliamo di utilizzare le chiavi contestuali `aws:SourceArn` e le condizioni `aws:SourceAccount` globali nella policy basata sulle risorse di Amazon MQ per limitare l'accesso ai CloudWatch log a uno o più broker specifici.

> **Note**
>
> Se si utilizzano entrambe le chiavi di contesto delle condizioni globali, il valore `aws:SourceAccount` e l'account nel valore `aws:SourceArn` devono utilizzare lo stesso ID account nella stessa istruzione di policy.

L'esempio seguente dimostra una policy basata sulle risorse che limita l'accesso ai CloudWatch log a un singolo broker Amazon MQ.

JSON

```
{
                    "Version":"2012-10-17",
                    "Statement": [
                        {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "mq.amazonaws.com"
                        },
                        "Action": [
                            "logs:CreateLogStream",
                            "logs:PutLogEvents"
```

```
                            ],
                            "Resource": "arn:aws:logs:*:*:log-group:/aws/
amazonmq/*",
                            "Condition": {
                                "StringEquals": {
                                "aws:SourceAccount": "123456789012",
                                "aws:SourceArn": "arn:aws:mq:us-
west-1:123456789012:broker:my-broker:123456789012"
                                }
                            }
                            }
                        ]
                        }
```

Puoi anche configurare una politica basata sulle risorse per limitare l'accesso ai CloudWatch log a tutti i broker di un account, come illustrato di seguito.

JSON

```
{
                            "Version":"2012-10-17",
                            "Statement": [
                            {
                                "Effect": "Allow",
                                "Principal": {
                                "Service": [
                                    "mq.amazonaws.com"
                                ]
                                },
                                "Action": [
                                "logs:CreateLogStream",
                                "logs:PutLogEvents"
                                ],
                                "Resource": "arn:aws:logs:*:*:log-group:/aws/
amazonmq/*",
                                "Condition": {
                                "ArnLike": {
                                    "aws:SourceArn":
 "arn:aws:mq:*:123456789012:broker:*"
                                },
                                "StringEquals": {
```

```
                        "aws:SourceAccount": "123456789012"
                    }
                    }
                }
                ]
            }
```

Per ulteriori informazioni sul problema di sicurezza "confused deputy", consulta Problema del "confused deputy" nella Guida per l'utente di IAM.

Risoluzione dei problemi di configurazione dei CloudWatch log con Amazon MQ

In alcuni casi, i CloudWatch log potrebbero non comportarsi sempre come previsto. Questa sezione fornisce una panoramica dei problemi più comuni e illustra come risolverli.

I gruppi di log non vengono visualizzati in CloudWatch

Aggiungere l'autorizzazione `CreateLogGroup` all'utente Amazon MQ e riavviare il broker. Questo consente ad Amazon MQ di creare il gruppo di registri.

I flussi di log non vengono visualizzati nei gruppi di CloudWatch log

Configurare una policy basata sulle risorse per Amazon MQ. Questo consente al broker di pubblicare i suoi log.

Quote in Amazon MQ

Questo argomento elenca i limiti all'interno di Amazon MQ. Molti dei seguenti limiti possono essere modificati per AWS account specifici. Per richiedere un aumento di un limite, consulta la sezione relativa alle quote dei servizi AWS nella Riferimenti generali di Amazon Web Services. I limiti aggiornati non saranno visibili anche dopo l'applicazione dell'aumento del limite. Per ulteriori informazioni sulla visualizzazione dei limiti di connessione correnti in Amazon CloudWatch, consulta Monitoraggio dei broker Amazon MQ tramite Amazon CloudWatch.

Argomenti

- Broker
- Configurazioni
- Utenti
- Storage dei dati
- Throttling delle API

Broker

La tabella seguente elenca le quote relative ai broker Amazon MQ.

Limite	Descrizione
Nome broker	• Deve essere unico nel tuo AWS account. • Deve contenere da 1 a 50 caratteri. • Deve contenere solo caratteri specificati nel set di caratteri ASCII stampabili. • Può contenere solo caratteri alfanumerici, trattini, punti, caratteri di sottolineatura e tilde (- . _ ~).
Numero di broker per regione	50

Limite	Descrizione
Connessioni a livello di collegamento per protocollo per broker più piccoli	**Important** Non si applica ai broker RabbitMQ. 300 per broker del tipo di istanza `mq.*.micro` .
Connessioni a livello di collegamento per protocollo per broker più grandi	**Important** Non si applica ai broker RabbitMQ. 2.000 per broker del tipo di istanza `mq.*.*large` .
Gruppi di sicurezza per broker	5
Destinazioni ActiveMQ (code e argomenti) monitorate in CloudWatch	CloudWatch monitora solo le prime 1000 destinazioni.
Destinazioni RabbitMQ (code) monitorate in CloudWatch	CloudWatch monitora solo le prime 500 destinazioni, ordinate per numero di consumatori.
Tag per broker	50

Configurazioni

La tabella seguente elenca le quote relative alle configurazioni Amazon MQ.

Limite	Descrizione
Nome configurazione	• Deve contenere da 1 a 150 caratteri.

Limite	Descrizione
	• Deve contenere solo caratteri specificati nel set di caratteri ASCII stampabili. • Può contenere solo caratteri alfanumerici, trattini, punti, caratteri di sottolineatura e tilde (- . _ ~).
Revisioni per configurazione	300

Utenti

La tabella seguente elenca le quote relative agli utenti dei broker ActiveMQ di Amazon MQ.

Limite	Descrizione
Username	• Deve contenere da 1 a 100 caratteri. • Deve contenere solo caratteri specificati nel set di caratteri ASCII stampabili. • Può contenere solo caratteri alfanumerici, trattini, punti, caratteri di sottolineatura e tilde (- . _ ~). • Non deve contenere virgole (,).
Password	• Deve contenere da 12 a 250 caratteri. • Deve contenere solo caratteri specificati nel set di caratteri ASCII stampabili. • Deve contenere almeno 4 caratteri univoci. •

Limite	Descrizione
	Non deve contenere virgole (`,`).
Utenti per broker (autenticazione semplice)	250
Gruppi per utente (autenticazione semplice)	20

Storage dei dati

La tabella seguente elenca le quote relative all'archiviazione dei dati di Amazon MQ.

Limite	Descrizione
Capacità di archiviazione per broker più piccolo	20 GB per broker del tipo di istanza `mq.*.micro` . Per ulteriori informazioni sui tipi di istanza Amazon MQ, consultare Broker instance types.
Capacità di archiviazione per broker più grande	200 GB per broker del tipo di istanza `mq.m5.*`. Per ulteriori informazioni sui tipi di istanza Amazon MQ, consultare Broker instance types.
Limite di utilizzo del pianificatore processi supportato da Amazon EBS	**Important** Non si applica ai broker RabbitMQ. 50 GB. Per ulteriori informazioni sull'utilizzo del pianificatore dei processi, consultare `JobSchedulerUsage` nella Documentazione sull'API di Apache ActiveMQ.
Capacità di archiviazione temporanea per broker più piccolo	

Limite	Descrizione
	Important Non si applica ai broker RabbitMQ. 5 GB per broker del tipo di istanza `mq.*.micro` .
Capacità di archiviazione temporanea per broker più grande	**Important** Non si applica ai broker RabbitMQ. 50 GB per broker del tipo di istanza `mq.m5.*`.

Throttling delle API

Le seguenti quote di limitazione vengono aggregate per AWS account, su tutto Amazon MQ per mantenere la larghezza di banda del servizio APIs. Per ulteriori informazioni su Amazon MQ APIs, consulta Amazon MQ REST API Reference.

Important

Queste quote non si applicano alla messaggistica del broker Amazon MQ for ActiveMQ o Amazon MQ for RabbitMQ. APIs Ad esempio, Amazon MQ non limita l'invio o la ricezione di messaggi.

Limite espansione API	Limite frequenza API
100	15

Risoluzione dei problemi di Amazon MQ

In questa sezione sono indicati i problemi comuni che possono verificarsi durante l'utilizzo di broker Amazon MQ e le operazioni possibili per risolverli. Per una risoluzione generale dei problemi, consulta. the section called “Risoluzione dei problemi: Amazon MQ generale” Per la risoluzione dei problemi relativi alla versione specifica del motore, consultate le seguenti sezioni.

Risoluzione dei problemi di ActiveMQ su Amazon MQ

Argomento sulla risoluzione dei problemi	Description
Risoluzione dei problemi generali	Utilizza le informazioni in questa sezione per aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con ActiveMQ sui broker Amazon MQ.
BROKER_ENI_DELETED	ActiveMQ su Amazon MQ genererà `BROKER_ENI_DELETED` un allarme quando elimini l'Elastic Network Interface (ENI) di un broker.
BROKER_OOM	ActiveMQ su Amazon MQ genererà un allarme BROKER_OOM quando il broker subisce un ciclo di riavvio a causa dell'insufficiente capacità di memoria

Risoluzione dei problemi di RabbitMQ su Amazon MQ

Argomento sulla risoluzione dei problemi	Description
Risoluzione dei problemi generali	Diagnostica i problemi più comuni che potresti riscontra re quando lavori con i broker RabbitMQ.

Argomento sulla risoluzione dei problemi	Description
RABBITMQ_MEMORY_ALARM	RabbitMQ genererà un allarme di elevata memoria quando l'utilizzo della memoria da parte del broker, identific ato da una CloudWatch metrica`RabbitMQMemUsed` , supera il limite di memoria, identificato da. `RabbitMQM emLimit`
RABBITMQ_INVALID_KMS_KEY	RabbitMQ su Amazon MQ genererà un codice di azione critica INVALID_KMS_KEY necessario quando un broker creato con una soluzione gestita dal cliente AWS KMS key(CMK) rileva che la chiave (KMS) è disabilitata. AWS Key Management Service
RABBITMQ_INVALID_ASSUMEROLE	RabbitMQ su Amazon MQ genererà un codice INVALID_ASSUMEROLE per l'azione critica richiesta quando il ruolo IAM ARN specificato in non può essere assunto da Amazon MQ. `aws.arns.assume_ro le_arn`

Argomento sulla risoluzione dei problemi	Description
RABBITMQ_INVALID_ARN_LDAP	RabbitMQ su Amazon MQ genererà un codice INVALID_ARN_LDAP per l'azione critica richiesta quando la password ARN dell'account del servizio LDAP non è valida o è inaccessibile.
RABBITMQ_INVALID_ARN_HTTP	RabbitMQ su Amazon MQ genererà un codice INVALID_ARN_HTTP per l'azione critica richiesta quando uno o più certificati SSL o il file chiave per HTTP ARNs auth_backend non sono validi o sono inaccessibili.
RABBITMQ_INVALID_ARN_SSL	RabbitMQ su Amazon MQ genererà un codice INVALID_ARN_SSL per l'azione critica richiesta quando uno o più ARNs certificati CA truststore for EXTERNAL auth_mech anism non sono validi o sono inaccessibili.
RABBITMQ_INVALID_ARN	RabbitMQ su Amazon MQ genererà un codice INVALID_ARN critical action required quando uno o più ARNs componenti della configurazione del broker non sono validi o sono inaccessi bili.

Argomento sulla risoluzione dei problemi	Description
RABBITMQ_DISK_ALARM	L'allarme relativo al limite del disco indica che il volume del disco utilizzato da un nodo RabbitMQ è diminuito a causa dell'elevato numero di messaggi non consumati durante l'aggiunta di nuovi messaggi.

Risoluzione dei problemi: Amazon MQ generale

Utilizza le informazioni contenute in questa sezione per diagnosticare problemi comuni che possono verificarsi durante l'utilizzo di broker Amazon MQ, quali problemi di connessione alil broker e riavvio del broker.

Indice

- Non riesco a connettermi alla console Web o agli endpoint del broker.
- Il mio broker è in esecuzione e posso verificare la connettività utilizzando telnet, ma i miei client non sono in grado di connettersi e restituiscono eccezioni SSL.
- Ho creato un broker ma la creazione non è riuscita.
- Il mio broker si è riavviato e non sono sicuro del motivo.

Non riesco a connettermi alla console Web o agli endpoint del broker.

Se si verificano problemi di connessione alil broker utilizzando la console Web o gli endpoint a livello di connessione, si consiglia di procedere come segue.

1. Controlla se stai tentando di connetterti alil broker da un firewall. Potrebbe essere necessario configurare il firewall per consentire l'accesso alil broker.
2. Controlla se stai tentando di connetterti al tuo broker utilizzando un endpointFIPS. Amazon MQ supporta gli endpoint FIPS solo quando si utilizzano le operazioni API e non per le connessioni a livello di collegamento all'istanza del broker stesso.

3. Controllare se l'opzione Public Accessibility (Accessibilità pubblica) per il broker è impostata su Yes (Sì). Se è impostata su No, controlla le regole di Access Control List (ACL) (Lista di controllo accessi) della sottorete. Se hai creato una rete personalizzata ACLs, potresti dover modificare le regole ACL di rete per consentire l'accesso al tuo broker. Per ulteriori informazioni sulla rete Amazon VPC, consulta Enabling Internet access nella Amazon VPC User Guide
4. Controllare le regole del gruppo di sicurezza del broker. Assicurarsi di consentire le connessioni alle seguenti porte:

> **Note**
>
> Le seguenti porte sono raggruppate in base ai tipi di motore perché ActiveMQ su Amazon MQ e RabbitMQ su Amazon MQ utilizzano porte diverse per le connessioni.

ActiveMQ su Amazon MQ

- Console Web: porta 8162
- OpenWire — Porta 61617
- AMQP: porta 5671
- STOMP: porta 61614
- MQTT: porta 8883
- WSS: porta 61619

RabbitMQ su Amazon MQ

- Console Web e API di gestione: porta 443 e 15671
- AMQP: porta 5671

5. Eseguire i seguenti test di connettività di rete per il tipo di motore del broker.

> **Note**
>
> Per i broker senza accessibilità pubblica, esegui i test da un'istanza Amazon EC2 all'interno dello stesso Amazon VPC del broker Amazon MQ e valutare le risposte.

ActiveMQ on Amazon MQ

Per testare la connettività di rete del tuo ActiveMQ sulla connettività di rete del broker Amazon MQ

1. Aprire una finestra della riga di comando o del terminale.
2. Eseguire il seguente comando `nslookup` per eseguire query sul registro DNS del broker. Per implementazioni attive/in standby, testare sia gli endpoint attivi che quelli in standby. Gli active/standby endpoint vengono identificati con un suffisso `-1` o `-2` aggiunti all'ID univoco del broker. Sostituire l'endpoint con le proprie informazioni.

```
$ nslookup b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-west-2.amazonaws.com
```

Se la query viene eseguita correttamente, verrà prodotto un risultato simile al seguente.

```
Non-authoritative answer:
Server:  dns-resolver-corp-sfo-1.sfo.corp.amazon.com
Address:  172.10.123.456

Name:    ec2-12-345-123-45.us-west-2.compute.amazonaws.com
Address:  12.345.123.45
Aliases:  b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-west-2.amazonaws.com
```

L'indirizzo IP risolto deve corrispondere agli indirizzi IP forniti nella console Amazon MQ. Ciò indica che il nome di dominio si sta risolvendo correttamente nel server DNS ed è possibile passare alla fase successiva.

3. Eseguire il seguente comando `telnet` per testare il percorso di rete per il broker. Sostituire l'endpoint con le proprie informazioni. *`port`*Sostituiscilo con il numero di porta `8162` per la console Web o altre porte a livello di cavo per testare protocolli aggiuntivi, se necessario.

> **Note**
>
> Per le active/standby distribuzioni, riceverai un messaggio di `Connect failed` errore se esegui `telnet` con l'endpoint di standby. Questo è prevedibile, dal momento che l'istanza in standby stessa è in esecuzione, ma il processo ActiveMQ non è in esecuzione e non ha accesso al volume di archiviazione Amazon EFS del

broker. Eseguire il comando per entrambi gli endpoint `-1` e `-2` per assicurarsi di testare sia le istanze attive che quelle in standby.

```
$ telnet b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
west-2.amazonaws.com port
```

Per l'istanza attiva, viene visualizzato un risultato simile al seguente.

```
Connected to b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
west-2.amazonaws.com.
Escape character is '^]'.
```

4. Scegli una delle seguenti operazioni.
   - Se il comando `telnet` ha esito positivo, controllare il parametro `EstablishedConnectionsCount` e confermare che il broker non abbia raggiunto il limite massimo di connessione a livello di collegamento. È anche possibile confermare se il limite è stato raggiunto esaminando i registri `General` del broker. Se questo parametro è superiore a zero, è presente almeno un client attualmente connesso alil broker. Se il parametro mostra zero connessioni, eseguire nuovamente il test del percorso `telnet` e attendere almeno un minuto prima di disconnettersi, poiché i parametri del broker vengono pubblicati ogni minuto.
   - Se il comando `telnet` non riesce, controllare lo stato dell'interfaccia di rete elastica del broker e confermare che lo stato corrisponda a `in-use`. Creare un flusso di log di Amazon VPC per l'interfaccia di rete di ogni istanza ed esaminare flussi di log generati. Cercare gli indirizzi IP del broker quando si esegue il comando `telnet` e confermare che i pacchetti di connessione siano `ACCEPTED`, incluso un pacchetto di ritorno. Per ulteriori informazioni e per vedere un esempio di flusso di log, consultare Esempi di record di flussi di log nella Guida per gli sviluppatori di Amazon VPC.
5. Eseguire il seguente comando `curl` per verificare la connettività alla console Web di amministrazione ActiveMQ.

```
$ curl https://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
west-2.amazonaws.com:8162/index.html
```

Se il comando viene eseguito correttamente, viene prodotto un documento HTML simile al seguente.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://
www.w3.org/TR/html4/loose.dtd">
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
 charset=iso-8859-1" />
        <title>Apache ActiveMQ</title>
        ...
```

RabbitMQ on Amazon MQ

Per testare RabbitMQ sulla connettività di rete del broker Amazon MQ

1. Aprire una finestra della riga di comando o del terminale.
2. Eseguire il seguente comando `nslookup` per interrogare il record DNS del broker. Sostituire l'endpoint con le proprie informazioni.

   ```
   $ nslookup b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-west-2.amazonaws.com
   ```

   Se la query viene eseguita correttamente, verrà prodotto un risultato simile al seguente.

   ```
   Non-authoritative answer:
   Server:  dns-resolver-corp-sfo-1.sfo.corp.amazon.com
   Address:  172.10.123.456

   Name:    rabbit-broker-1c23e456ca78-b9000123b4ebbab5.elb.us-
   west-2.amazonaws.com
   Addresses:  52.12.345.678
             52.23.234.56
             41.234.567.890
             54.123.45.678
   Aliases:  b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-west-2.amazonaws.com
   ```

3. Eseguire il seguente comando `telnet` per testare il percorso di rete per il broker. Sostituire l'endpoint con le proprie informazioni. Puoi sostituirla *`port`* con una porta `443` per la console Web e testare la connessione `5671` AMQP a livello di cavo.

```
$ telnet b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
west-2.amazonaws.com port
```

Se il comando viene eseguito correttamente, verrà visualizzato un risultato simile al seguente.

```
Connected to b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
west-2.amazonaws.com.
Escape character is '^]'.
```

> **Note**
>
> La connessione telnet si chiuderà automaticamente dopo alcuni secondi.

4. Scegli una delle seguenti operazioni.

   - Se il comando `telnet` ha esito positivo, controllare il comando `ConnectionCount` e confermare che il broker non abbia raggiunto il valore impostato nella policy predefinita `max-connections`. È anche possibile confermare se il limite è stato raggiunto esaminando il gruppo di registri `Connection.log` del broker. Se questo parametro è superiore a zero, è presente almeno un client attualmente connesso alil broker. Se il parametro mostra zero connessioni, eseguire nuovamente il test di percorso `telnet`. Potrebbe essere necessario ripetere questo processo se la connessione si chiude prima che il broker abbia pubblicato nuove metriche di connessione su. CloudWatch I parametri vengono pubblicati ogni minuto.
   - Per broker senza accessibilità pubblica, se il comando `telnet` non riesce, controllare lo stato delle interfacce di rete elastiche del broker e confermare che lo stato corrisponda a `in-use`. Creare un flusso di log di Amazon VPC per ogni interfaccia di rete ed esaminare i flussi di log generati. Cercare gli indirizzi IP privati del broker quando si richiama il comando `telnet` e confermare che i pacchetti di connessione siano `ACCEPTED`, incluso un pacchetto di ritorno. Per ulteriori informazioni e per vedere un esempio di flusso di log, consultare Esempi di record di flussi di log nella Guida per gli sviluppatori di Amazon VPC.

> **Note**
>
> Questo passaggio non si applica ai broker RabbitMQ su Amazon MQ con accessibilità pubblica.

5. Eseguire il seguente comando `curl` per verificare la connettività alla console Web di amministrazione RabbitMQ.

```
$ curl https://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
west-2.amazonaws.com:443/index.html
```

Se il comando viene eseguito correttamente, viene prodotto un documento HTML simile al seguente.

```
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="X-UA-Compatible" content="IE=edge" />
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <title>RabbitMQ Management</title>
        ...
```

Il mio broker è in esecuzione e posso verificare la connettività utilizzando **`telnet`**, ma i miei client non sono in grado di connettersi e restituiscono eccezioni SSL.

Il certificato endpoint del broker potrebbe essere stato aggiornato durante la finestra di manutenzione del broker. I certificati del broker Amazon MQ vengono ruotati periodicamente per garantire la disponibilità e la sicurezza continue dei broker.

Consigliamo di utilizzare la certification authority (CA) root di Amazon in Amazon Trust Services per autenticarsi nel negozio di fiducia dei tuoi clienti. Tutti i certificati del broker Amazon MQ sono firmati con questa CA principale. Utilizzando una CA root di Amazon, non sarà più necessario scaricare il nuovo certificato del broker Amazon MQ ogni volta che è presente un aggiornamento del certificato sul broker.

Ho creato un broker ma la creazione non è riuscita.

Se il broker è in uno stato `CREATION_FAILED`, procedere come indicato di seguito.

- Controllare le autorizzazioni IAM. Per creare un broker è necessario utilizzare la policy IAM AWS gestita `AmazonMQFullAccess` o disporre del set corretto di autorizzazioni Amazon EC2 nella policy IAM personalizzata. Per ulteriori informazioni sulle autorizzazioni Amazon EC2 necessarie, consultare Autorizzazioni IAM necessarie per creare un broker Amazon MQ.
- Controllare se la sottorete scelta per il broker si trova in un Amazon Virtual Private Cloud (VPC) condiviso. Per creare un broker Amazon MQ in un Amazon VPC condiviso, crearlo nell'account proprietario di Amazon VPC.

Il mio broker si è riavviato e non sono sicuro del motivo.

Se il broker viene riavviato automaticamente, il motivo può essere uno dei seguenti.

- È possibile che il broker sia stato riavviato a causa di una finestra di manutenzione programmata settimanale. Periodicamente, Amazon MQ esegue la manutenzione dell'hardware, del sistema operativo o del software del motore di un broker di messaggistica. La durata della manutenzione varia, ma può durare fino a due ore, a seconda delle operazioni pianificate per il broker di messaggistica. i broker potrebbero riavviarsi in qualsiasi momento durante la finestra di manutenzione di due ore. Per ulteriori informazioni sulle finestre di manutenzione del broker, consulta. the section called “Pianificazione della manutenzione del broker”
- Il tipo di istanza del broker potrebbe non essere adatto al carico di lavoro dell'applicazione. Ad esempio, l'esecuzione di un carico di lavoro di produzione su un `mq.t3.micro` potrebbe comportare l'esaurimento delle risorse del broker. Un elevato utilizzo della CPU o un elevato utilizzo della memoria del broker può causare il riavvio inaspettato di un broker. Per vedere quanta CPU e memoria vengono utilizzate dal broker, utilizza le seguenti CloudWatch metriche relative al tipo di motore.
  - ActiveMQ su Amazon MQ: `CpuUtilization` verifica la percentuale di unità di calcolo Amazon EC2 allocate attualmente utilizzate dal broker. Controllare `HeapUsage` per la percentuale del limite di memoria ActiveMQ JVM utilizzata attualmente dalil broker.
  - RabbitMQ su Amazon MQ: verifica `SystemCpuUtilization` la percentuale di unità di calcolo Amazon EC2 allocate attualmente utilizzate dal broker. Controllare `RabbitMQMemUsed` per il volume della RAM utilizzata in byte e dividere per `RabbitMQMemLimit` per la percentuale di memoria utilizzata dal nodo RabbitMQ.

Per ulteriori informazioni sui tipi di istanze del broker e su come scegliere il tipo di istanza giusto per il tuo carico di lavoro, consulta. Broker instance types

Risoluzione dei problemi di ActiveMQ su Amazon MQ

Utilizza le informazioni in questa sezione per aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con ActiveMQ sui broker Amazon MQ.

Indice

- Non riesco a visualizzare i log generali o di controllo del mio broker in CloudWatch Logs anche se ho attivato la registrazione.
- Dopo il riavvio o la finestra di manutenzione del broker, non riesco a connettermi al mio broker anche se lo stato è RUNNING. Perché?
- Vedo che alcuni dei miei client si connettono al broker, mentre altri non sono in grado di farlo.
- Vedo un'eccezione org.apache.jasper.JasperException: An exception occurred processing JSP page sulla console ActiveMQ durante l'esecuzione delle operazioni.

Non riesco a visualizzare i log generali o di controllo del mio broker in CloudWatch Logs anche se ho attivato la registrazione.

Se non riesci a visualizzare i log del tuo broker in CloudWatch Logs, procedi come segue.

1. Controllare se l'utente che crea o riavvia il broker dispone dell'autorizzazione `logs:CreateLogGroup`. Se non si aggiunge l'autorizzazione `CreateLogGroup` all'utente prima che l'utente crei o riavvi il broker, Amazon MQ non crea il gruppo di registri.
2. Verifica se hai configurato una policy basata sulle risorse per consentire ad Amazon MQ di pubblicare i log su Logs. CloudWatch Per consentire ad Amazon MQ di pubblicare i log nel tuo gruppo di log CloudWatch Logs, configura una policy basata sulle risorse per consentire ad Amazon MQ di accedere alle seguenti azioni dell'API Logs: CloudWatch
   - `CreateLogStream`— Crea un flusso di CloudWatch log di Logs per il gruppo di log specificato.
   - `PutLogEvents`— Fornisce gli eventi al flusso di log di CloudWatch Logs specificato.

Per ulteriori informazioni sulla configurazione di ActiveMQ su Amazon MQ per pubblicare i log nei CloudWatch log, consulta Configurazione della registrazione.

Dopo il riavvio o la finestra di manutenzione del broker, non riesco a connettermi al mio broker anche se lo stato è **RUNNING**. Perché?

Potresti riscontrare problemi di connessione dopo il riavvio di un broker, dopo aver completato una finestra di manutenzione pianificata o in un evento di fallimento, nei quali l'istanza di standby è attivata. In entrambi i casi, i problemi di connessione a seguito di un riavvio del broker sono probabilmente causati da un numero insolitamente grande di messaggi persistenti nel volume di archiviazione Amazon EFS o Amazon EBS del broker stesso. Durante il riavvio, Amazon MQ sposta i messaggi persistenti dall'archiviazione alla memoria del broker. Per confermare questa diagnosi, puoi monitorare le seguenti metriche per il tuo broker Amazon MQ CloudWatch for ActiveMQ:

- **StoragePercentUsage**: grandi percentuali pari o vicine al 100%, possono causare il rifiuto delle connessioni da parte del broker.
- **JournalFilesForFullRecovery**: indica il numero di file di registro che vengono riprodotti dopo uno spegnimento e un riavvio. Un valore crescente o costantemente superiore a uno indica transazioni non risolte che possono causare problemi di connessione in seguito al riavvio.
- **OpenTransactionCount**: un numero maggiore di zero dopo un riavvio indica che il broker tenterà di archiviare i messaggi consumati in precedenza, causando problemi di connessione.

Per risolvere questo problema, ti consigliamo di risolvere le tue transazioni XA con un `rollback()` o con un `commit()`. Per ulteriori informazioni e per vedere un esempio di codice di risoluzione di transazioni XA utilizzando `rollback()`, consulta recupero di transazioni XA.

Vedo che alcuni dei miei client si connettono al broker, mentre altri non sono in grado di farlo.

Se il tuo broker è nello stato `RUNNING` e alcuni client sono in grado di connettersi con successo al broker, mentre altri non sono in grado di farlo, potresti aver raggiunto il limite di connessioni a livello di filo per il broker. Per verificare di aver raggiunto il limite di connessioni a livello di filo, procedi come segue:

- Controlla i log generali del broker per il tuo broker ActiveMQ su Amazon MQ in Logs. CloudWatch Se il limite è stato raggiunto, vedrai `Reached Maximum Connections` nei registri del broker. Per ulteriori informazioni su CloudWatch Logs for ActiveMQ sui broker Amazon MQ, consulta. the section called “Comprensione della struttura di registrazione nei log CloudWatch”

Una volta raggiunto il limite di connessioni a livello di filo, il broker rifiuterà attivamente ulteriori connessioni in entrata. Per risolvere questo problema, suggeriamo di aggiornare il tipo di istanza di broker. Per ulteriori informazioni sulla scelta del tipo di istanza migliore per un carico di lavoro specifico, consulta Broker instance types.

Se hai confermato che il numero di connessioni a livello di filo è inferiore al limite di connessione del broker, il problema potrebbe essere correlato al riavvio dei client. Controlla i registri del tuo broker per numerose e frequenti voci di `... Inactive for longer than 600000 ms - removing ...`. La voce di registro indica i problemi di riavvio dei client o di connettività. Questo effetto è più evidente quando i client si connettono al broker tramite un load balancer di rete (NLB, Network Load Balancer) con client che spesso si disconnettono e si riconnettono al broker. Tipicamente, questo è osservato soprattutto nei client basati su container.

Per ulteriori dettagli, controlla i registri sul lato client. Il broker ripulirà le connessioni TCP inattive dopo 600000 ms e libererà il socket della connessione.

Vedo un'eccezione **`org.apache.jasper.JasperException: An exception occurred processing JSP page`** sulla console ActiveMQ durante l'esecuzione delle operazioni.

Se utilizzi l'autenticazione e la configurazione semplici `AuthorizationPlugin` per l'autorizzazione di coda e argomento, assicurati di utilizzare l'elemento `AuthorizationEntries` nel file di configurazione XML e permetti l'autorizzazione di gruppo `activemq-webconsole` per tutte le code e gli argomenti. Ciò garantirà che la console Web ActiveMQ possa comunicare con il broker ActiveMQ.

L'esempio seguente, `AuthorizationEntry`, concede le autorizzazioni di lettura e scrittura per tutte le code e gli argomenti al gruppo `activemq-webconsole`.

```
<authorizationEntries>
    <authorizationEntry admin="activemq-webconsole,admins,users" topic=">"
 read="activemq-webconsole,admins,users" write="activemq-webconsole,admins,users" />
    <authorizationEntry admin="activemq-webconsole,admins,users" queue=">"
 read="activemq-webconsole,admins,users" write="activemq-webconsole,admins,users" />
</authorizationEntries>
```

Allo stesso modo, quando integri il tuo broker con LDAP, assicurati di concedere l'autorizzazione al gruppo `amazonmq-console-admins`. Per ulteriori informazioni sull'integrazione LDAP, consulta. the section called “Come funziona l'integrazione LDAP”

Risoluzione dei problemi: RabbitMQ su Amazon MQ

Utilizza le informazioni in questa sezione per aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con RabbitMQ sui broker Amazon MQ.

Indice

- Non riesco a visualizzare le metriche relative alle mie code o ai miei host virtuali. CloudWatch
- Come posso abilitare i plugin in RabbitMQ su Amazon MQ?
- Non riesco a modificare la configurazione di Amazon VPC per il broker.
- Le implementazioni dei cluster hanno messo in pausa le sincronizzazioni delle mie code.
- Il mio broker a istanza singola Amazon MQ for RabbitMQ è in un ciclo di riavvio.
- Ho perso l'accesso a tutti gli account di amministratore sul mio broker.

Non riesco a visualizzare le metriche relative alle mie code o ai miei host virtuali. CloudWatch

Se non riesci a visualizzare le metriche relative alle code o agli host virtuali in CloudWatch, controlla se i nomi delle code o degli host virtuali contengono spazi vuoti, schede o altri caratteri non ASCII.

Amazon MQ non può pubblicare parametri per host virtuali e code con nomi contenenti spazi vuoti, tabulazioni o altri caratteri non ASCII.

Per ulteriori informazioni sui nomi delle dimensioni, consulta Dimension in Amazon CloudWatch API Reference.

Come posso abilitare i plugin in RabbitMQ su Amazon MQ?

RabbitMQ su Amazon MQ attualmente supporta solo il plug-in di gestione, shovel, federazione e scambio consistent-hash di RabbitMQ, che sono abilitati per impostazione predefinita. Per ulteriori informazioni sull'utilizzo dei plugin supportati, consulta the section called “Plugin”.

Non riesco a modificare la configurazione di Amazon VPC per il broker.

Amazon MQ non supporta la modifica della configurazione di Amazon VPC dopo la creazione del broker. Ti ricordiamo che dovrai creare un nuovo broker con la nuova configurazione Amazon VPC e aggiornare l'URL della connessione client con il nuovo URL di connessione del broker.

Le implementazioni dei cluster hanno messo in pausa le sincronizzazioni delle mie code.

Quando si affronta il problema degli allarmi ad alta memoria di RabbitMQ, è possibile che i messaggi su una o più code non possano essere consumati. Queste code possono essere in fase di sincronizzazione dei messaggi tra i nodi, durante i quali le rispettive code non sono disponibili per la pubblicazione e il consumo. Le sincronizzazioni delle code potrebbero essere sospese a causa dell'allarme di memoria elevata e persino contribuire all'allarme di memoria.

Per informazioni sull'interruzione e la ripetizione del tentativo di sincronizzazione della coda in pausa, consulta the section called “Risoluzione della sincronizzazione della coda sospesa”.

Il mio broker a istanza singola Amazon MQ for RabbitMQ è in un ciclo di riavvio.

Un broker a istanza singola Amazon MQ per RabbitMQ che genera un allarme di memoria elevata rischia di diventare non disponibile se si riavvia e non dispone di memoria sufficiente per l'avvio. Ciò può far sì che RabbitMQ entri in un ciclo di riavvio e impedisca ulteriori interazioni con il broker fino a quando il problema non viene risolto. Se il tuo broker è in un ciclo di riavvio, non sarai in grado di applicare le best practice consigliate da Amazon MQ per risolvere l'allarme di memoria elevata.

Per ripristinare il tuo broker, ti consigliamo di eseguire l'aggiornamento a un tipo di istanza più grande con più memoria. A differenza delle implementazioni in cluster, puoi aggiornare un broker a istanza singola quando si verifica un allarme di memoria elevata, poiché non ci sono sincronizzazioni delle code da eseguire tra i nodi durante un riavvio.

Ho perso l'accesso a tutti gli account di amministratore sul mio broker.

Puoi ripristinare l'accesso utilizzando l'autenticazione IAM. Abilita la federazione delle identità web in uscita per il tuo AWS account, crea un ruolo IAM con le autorizzazioni per ottenere token di identità web, configura il tuo broker in modo che accetti l'autenticazione IAM tramite OAuth 2.0, quindi utilizza le credenziali IAM per ottenere un token JWT e crea un nuovo utente amministratore. Per istruzioni dettagliate, vedi the section called “Utilizzo dell'autenticazione e dell'autorizzazione IAM”.

ActiveMQ su Amazon MQ: allarme eliminato dell'interfaccia di rete elastica

ActiveMQ su Amazon MQ genererà un allarme BROKER_ENI_DELETED quando elimini l'Elastic Network Interface (ENI) di un broker. La prima volta che crei un broker Amazon MQ, Amazon MQ esegue il provisioning di un'interfaccia di rete elastica nel Virtual Private Cloud (VPC) nel tuo account e, pertanto, richiede una serie di autorizzazioni EC2.

Questa interfaccia di rete deve essere modificata o eliminata. La modifica o l'eliminazione dell'interfaccia di rete può causare una perdita permanente della connessione tra il VPC e il broker. Se desideri eliminare l'interfaccia di rete, devi prima eliminare il broker.

ActiveMQ su Amazon MQ: allarme di memoria esaurita del broker

ActiveMQ su Amazon MQ genererà un allarme BROKER_OOM quando il broker subisce un ciclo di riavvio a causa dell'insufficiente capacità di memoria. Quando un broker si trova in un ciclo di riavvio, chiamato anche ciclo di rimbalzo, il broker avvia ripetuti tentativi di ripristino entro una breve finestra di tempo. I broker a istanza singola che non possono completare l'avvio a causa dell'elevato utilizzo della memoria potrebbero entrare in un ciclo di riavvio, durante il quale le interazioni con il broker sono limitate.

Amazon MQ abilita i parametri per il tuo broker per impostazione predefinita. Puoi visualizzare le metriche del tuo broker accedendo alla CloudWatch console Amazon o utilizzando l' CloudWatch API. I seguenti parametri sono utili quando si diagnostica l'allarme ActiveMQ BROKER_OOM:

CloudWatch Metrica Amazon MQ	Motivo dell'uso elevato di memoria	
`TotalMessageCount`	I messaggi vengono memorizzati in memoria fino a quando non vengono consumati o eliminati. Un elevato numero di messaggi potrebbe indicare un uso eccessivo delle risorse e può causare un allarme di memoria elevata.	

CloudWatch Metrica Amazon MQ	Motivo dell'uso elevato di memoria	
`HeapUsage`	Percentuale del limite di memoria ActiveMQ JVM utilizzata attualmente dal broker. Una percentuale più alta indica che il broker sta utilizzando risorse significa tive e può causare un allarme OOM.	
`ConnectionCount`	Le connessioni client utilizzan o la memoria e troppe connessioni simultanee possono causare un allarme di memoria elevata.	
`CpuUtilization`	Percentuale delle unità di elaborazione EC2 assegnate e attualmente utilizzate dal broker.	
`TotalConsumerCount`	Per ogni consumatore connesso al broker, un determinato numero di messaggi viene caricato dallo storage in memoria prima che vengano recapitati al consumatore. Un gran numero di connessioni degli utenti potrebbe causare un elevato utilizzo della memoria e un allarme di memoria elevato.	

Per evitare i cicli di riavvio ed evitare l'allarme BROKER_OOM, assicurati che i messaggi vengano consumati rapidamente. Puoi farlo scegliendo il tipo di istanza di broker più efficace e pulendo anche

la Dead Letter Queue per eliminare i messaggi non recapitabili o scaduti. Per saperne di più su come garantire prestazioni efficaci, consulta le best practice di ActiveMQ on Amazon MQ.

Amazon MQ per RabbitMQ: allarme ad alta memoria

Amazon MQ for RabbitMQ genererà un allarme di memoria elevata quando l'utilizzo della memoria da parte del broker, identificato da una CloudWatch metrica`RabbitMQMemUsed`, supera il limite di memoria, identificato da. `RabbitMQMemLimit`

Un broker RabbitMQ che ha generato un allarme di memoria elevata bloccherà tutti i client che pubblicano messaggi. Il tuo broker potrebbe entrare in un ciclo di riavvio, riscontrare una sincronizzazione della coda in pausa o sviluppare altri problemi che complicano la diagnosi e la risoluzione dell'allarme.

Per diagnosticare e risolvere un allarme con memoria elevata, segui innanzitutto tutte le migliori pratiche per RabbitMQ, quindi completa i passaggi seguenti.

> **Important**
>
> - `RabbitMQMemLimit`è impostato da Amazon MQ ed è ottimizzato in modo specifico considerando la memoria disponibile per ogni tipo di istanza host.
> - Amazon MQ non riavvierà un broker con un allarme di memoria elevata e restituirà un'eccezione per Operazioni API `RebootBroker` purché il broker continui a sollevare l'allarme.

Fase 1: Diagnostica di un allarme con memoria elevata

Esistono due modi per diagnosticare allarmi con elevata memoria sul tuo broker Amazon MQ for RabbitMQ. Ti consigliamo di controllare sia la console web di RabbitMQ che le metriche di Amazon MQ. CloudWatch

Diagnostica un allarme con memoria elevata utilizzando la console web di RabbitMQ

La console web RabbitMQ è in grado di generare e visualizzare informazioni dettagliate sull'utilizzo della memoria per ciascun nodo. Puoi trovare queste informazioni eseguendo le seguenti operazioni:

1. Accedi Console di gestione AWS e apri la console web RabbitMQ del tuo broker.

2. Sulla console RabbitMQ, alla pagina Panoramica, scegliere il nome di un nodo dall'elenco Nodi.
3. Nella pagina dei dettagli del nodo, selezionare Dettagli della memoria per espandere la sezione per visualizzare le informazioni sull'utilizzo della memoria del nodo.

Le informazioni sull'utilizzo della memoria fornite da RabbitMQ nella console Web possono aiutarti a determinare quali risorse potrebbero consumare troppa memoria e contribuire all'allarme di memoria elevata. Per ulteriori informazioni sui dettagli sull'utilizzo della memoria disponibili tramite la console web di RabbitMQ, consulta Reasoning About Memory Use sul sito Web RabbitMQ Server Documentation.

Diagnostica allarmi con memoria elevata utilizzando i parametri di Amazon MQ

Amazon MQ abilita i parametri per il tuo broker per impostazione predefinita. Puoi visualizzare i parametri del broker accedendo alla CloudWatch console o utilizzando l'API. CloudWatch I seguenti parametri sono utili quando si diagnostica l'allarme di memoria elevata RabbitMQ.

CloudWatch Metrica Amazon MQ	Motivo dell'uso elevato di memoria	
`MessageCount`	I messaggi vengono memorizzati in memoria fino a quando non vengono consumati o eliminati. Un elevato numero di messaggi potrebbe indicare un uso eccessivo delle risorse e può causare un allarme di memoria elevata.	
`QueueCount`	Le code sono memorizzate nella memoria e un numero elevato di code può causare un allarme di memoria elevata.	
`ConnectionCount`	Le connessioni client utilizzan o la memoria e troppe connessioni simultanee	

CloudWatch Metrica Amazon MQ	Motivo dell'uso elevato di memoria	
	possono causare un allarme di memoria elevata.	
`ChannelCount`	Analogamente alle connessioni, anche i canali stabiliti con ciascuna connessione vengono memorizzati nella memoria dei nodi e un numero elevato di canali può causare un allarme di memoria elevata.	
`ConsumerCount`	Per ogni consumatore connesso al broker, un determinato numero di messaggi viene caricato dallo storage in memoria prima che vengano recapitati al consumatore. Un gran numero di connessioni degli utenti potrebbe causare un elevato utilizzo della memoria e un allarme di memoria elevato.	
`PublishRate`	La pubblicazione di messaggi utilizza la memoria del broker. Se la velocità con cui i messaggi vengono pubblicati al broker è troppo alta e supera significativamente la velocità con cui il broker invia messaggi ai consumatori, il broker potrebbe causare un allarme di memoria elevata.	

Fase 2: Risolve e previene l'allarme di memoria esaurita

> **Note**
>
> Potrebbero essere necessarie fino a diverse ore prima che lo stato RABBITMQ_MEMORY_ALARM venga cancellato dopo aver eseguito le azioni richieste.

Segui tutte le migliori pratiche per RabbitMQ come metodo generale di prevenzione. Per ogni collaboratore specifico identificato, consigliamo la seguente serie di azioni per risolvere e prevenire gli allarmi di memoria elevata di RabbitMQ.

Fonte di elevato utilizzo della memoria	Raccomandazione di Amazon MQ per l'indirizzamento	Raccomandazione di Amazon MQ per la prevenzione
Numero di messaggi	Consuma i messaggi pubblicati nelle code, elimina i messaggi dalle code o elimina le code dal tuo broker.	Abilita le code pigre e imposta o riduci il limite di profondità della coda.
Numero di code	Ridurre il numero di code.	Imposta o riduci il limite di numero di code.
Numero di connessioni	Riduci il numero di connessioni.	Imposta o riduci il limite del numero di connessioni.
Numero di canali	Ridurre il numero di canali.	Imposta un numero massimo di canali per connessione sulle applicazioni client.
Numero di consumatori	Ridurre il numero di consumatori collegati al broker.	Impostare un piccolo limite di pre-recupero consumatore.
Velocità di pubblicazione dei messaggi	Ridurre la velocità con cui i messaggi vengono pubblicati al broker.	Attiva le conferme dell'editore.

Fonte di elevato utilizzo della memoria	Raccomandazione di Amazon MQ per l'indirizzamento	Raccomandazione di Amazon MQ per la prevenzione
Frequenza dei tentativi di connessione del client	Ridurre la frequenza con cui i client tentano di connetter si al broker per pubblicare o consumare messaggi o configurare il broker.	Usa connessioni di durata maggiore per ridurre il numero e la frequenza dei tentativi di connessione.

Una volta risolto l'allarme relativo alla memoria del broker, puoi aggiornare il tipo di istanza host a un'istanza con risorse aggiuntive. Per informazioni su come aggiornare il tipo di istanza del broker, consulta `UpdateBrokerInput`Amazon MQ REST API Reference.

> **Note**
>
> Non puoi effettuare il downgrade di un broker da un tipo di `mq.m5.x` istanza a un tipo di `mq.t3.micro` istanza. Per effettuare il downgrade, devi eliminare il broker e crearne uno nuovo.

RabbitMQ su Amazon MQ: chiave non valida AWS Key Management Service

RabbitMQ su Amazon MQ genererà un codice di azione critica INVALID_KMS_KEY necessario quando un broker creato con una soluzione gestita dal cliente AWS KMS key(CMK) rileva che la chiave (KMS) è disabilitata. AWS Key Management Service Un broker RabbitMQ con un CMK verifica periodicamente che la chiave KMS sia abilitata e che il broker disponga di tutte le autorizzazioni necessarie. Se RabbitMQ non è in grado di verificare che la chiave sia abilitata, il broker viene messo in quarantena e RabbitMQ restituirà INVALID_KMS_KEY.

Senza una chiave KMS attiva, il broker non dispone delle autorizzazioni di base per le chiavi KMS gestite dal cliente. Il broker non può eseguire operazioni crittografiche utilizzando la tua chiave finché non la riattivi e il broker non si riavvia. Un broker RabbitMQ con una chiave KMS disabilitata viene messo in quarantena per evitare il deterioramento. Dopo che RabbitMQ determina che la chiave KMS è nuovamente attiva, il broker viene rimosso dalla quarantena. Amazon MQ non riavvierà un broker

con una chiave KMS disabilitata e restituisce un'eccezione per Operazioni API `RebootBroker` purché il broker continui a disporre di una chiave KMS non valida.

Diagnosi e risoluzione di INVALID_KMS_KEY

Per diagnosticare e risolvere il codice richiesto dall'azione INVALID_KMS_KEY, è necessario utilizzare l'interfaccia a AWS riga di comando (CLI) e la console. AWS Key Management Service

Riabilitazione della chiave KMS

1. Chiama il metodo `DescribeBroker` per recuperare `kmsKeyId` per il tuo broker CMK.
2. AWS Key Management Service Accedere alla console.
3. Nella pagina delle chiavi gestite dal cliente, individua l'ID chiave KMS del broker problematico e verifica che lo stato sia Abilitato.
4. Se la tua chiave KMS è stata disattivata, riattiva la chiave scegliendo Azioni chiave, quindi scegli Abilita. Dopo aver riabilitato la chiave, devi attendere che RabbitMQ rimuova il broker dalla quarantena.

Per verificare che le sovvenzioni necessarie siano ancora associate alla chiave KMS del broker, richiama il `ListGrant` ListGrant metodo per verificarlo `mq_rabbit_grant` e che `mq_grant` siano presenti. Se la concessione o la chiave KMS è stata eliminata, devi eliminare il broker e crearne uno nuovo con tutte operazioni necessarie. Per la procedura di eliminazione di un broker, consulta Eliminare un broker.

Per evitare che l'azione critica INVALID_KMS_KEY richieda il codice, non eliminare o disabilitare manualmente una chiave KMS o una concessione CMK. Se desideri eliminare la chiave, elimina prima il broker.

RabbitMQ su Amazon MQ: allarme limite del disco

L'allarme relativo al limite del disco indica che il volume del disco utilizzato da un nodo RabbitMQ è diminuito a causa dell'elevato numero di messaggi non consumati durante l'aggiunta di nuovi messaggi. RabbitMQ genererà un allarme sul limite del disco quando lo spazio libero su disco del broker, identificato da Amazon CloudWatch Metric`RabbitMQDiskFree`, raggiunge il limite del disco, identificato da. `RabbitMQDiskFreeLimit` `RabbitMQDiskFreeLimit`è impostato da Amazon MQ ed è stato definito considerando lo spazio su disco disponibile per ogni tipo di istanza del broker.

Un broker RabbitMQ su Amazon MQ che ha generato un allarme sul limite del disco diventerà non disponibile per la pubblicazione di nuovi messaggi. Se hai un editore e un consumatore sulla stessa connessione, anche il consumatore non sarà disponibile per ricevere messaggi. Quando si esegue RabbitMQ in un cluster, l'allarme del disco è a livello di cluster. Se un nodo scende al di sotto del limite, tutti gli altri nodi bloccheranno i messaggi in arrivo. A causa di mancanza di spazio sul disco, il broker potrebbe anche ricontrare altri problemi che complicano la diagnosi e la risoluzione dell'allarme.

Amazon MQ non riavvierà un broker con un allarme di disco e restituirà un'eccezione per Operazioni API `RebootBroker`, purché il broker continui a sollevare l'allarme.

> **Note**
>
> Non è possibile effettuare il downgrade di un broker da un tipo di istanza `mq.m5` a un tipo di istanza `mq.t3.micro`. Se desideri effettuare un downgrade, devi eliminare il broker e crearne uno nuovo.

Diagnosi e risoluzione dell'allarme relativo al limite del disco

Amazon MQ abilita i parametri per il tuo broker per impostazione predefinita. Puoi visualizzare le metriche del tuo broker accedendo alla CloudWatch console Amazon o utilizzando l' CloudWatch API. `MessageCount`è una metrica utile per la diagnosi dell'allarme relativo al limite del disco RabbitMQ. I messaggi vengono memorizzati in memoria fino a quando non vengono consumati o eliminati. Un elevato numero di messaggi indica un uso eccessivo dello spazio di archiviazione su disco e può causare un allarme del disco.

Per diagnosticare l'allarme relativo al limite del disco, utilizza la console di gestione Amazon MQ per:

- Crea una nuova connessione per utilizzare i messaggi pubblicati nelle code.
- Rimuovere i messaggi dalle code.
- Elimina le code dal tuo broker.

> **Note**
>
> Potrebbero essere necessarie fino a diverse ore prima che lo stato RABBITMQ_DISK_ALARM venga cancellato dopo aver eseguito le azioni richieste.

Per evitare che l'allarme di limite del disco si ripeta, è possibile aggiornare il tipo di istanza host a un'istanza con risorse aggiuntive. Per informazioni su come aggiornare il tipo di istanza del broker, consulta `UpdateBrokerInput` nella Guida di riferimento delle API REST di Amazon MQ. Ti consigliamo inoltre di mantenere i tuoi editori e i tuoi consumatori su connessioni diverse.

Amazon MQ per RabbitMQ: allarme di modifica del tipo di istanza

`RABBITMQ_CLUSTER_DISK_USAGE_TOO_HIGH_FOR_INSTANCE_CHANGE`indica che la modifica del tipo di istanza del broker richiesta non può procedere a causa dell'elevato utilizzo del disco sul nodo RabbitMQ corrente. Amazon MQ for RabbitMQ genererà questo allarme quando l'utilizzo corrente del disco supera quello che sarebbe disponibile sul tipo di istanza richiesto, come identificato dalla metrica. CloudWatch `RabbitMQDiskFree`

I broker RabbitMQ che inseriscono `RABBITMQ_CLUSTER_DISK_USAGE_TOO_HIGH_FOR_INSTANCE_CHANGE` lo stato continueranno a essere disponibili per le tue applicazioni, ma la modifica del tipo di istanza richiesta non procederà. Amazon MQ consente il riavvio del broker in questo stato, ma non è possibile modificare il tipo di istanza finché l'utilizzo del disco rimane al di sopra della soglia per il tipo di istanza richiesto. Il broker restituirà un'eccezione per le operazioni `ModifyBroker` API che tentano di modificare il tipo di istanza in questo stato.

Diagnosi e risoluzione dell'allarme di modifica del tipo di istanza

Amazon MQ abilita i parametri per il tuo broker per impostazione predefinita. Puoi visualizzare le metriche del tuo broker accedendo alla CloudWatch console o utilizzando l' CloudWatch API. `MessageCounte` le `RabbitMQDiskFree` metriche possono essere utilizzate per la diagnosi. `RABBITMQ_CLUSTER_DISK_USAGE_TOO_HIGH_FOR_INSTANCE_CHANGE`

Per risolvere lo stato di quarantena e consentire la modifica del tipo di istanza, utilizza la console di gestione Amazon MQ per:

- Crea una nuova connessione per utilizzare i messaggi pubblicati nelle code.
- Rimuovere i messaggi dalle code.
- Elimina le code dal tuo broker.

> **Note**
>
> Dopo aver eseguito le azioni richieste, potrebbero essere necessarie diverse ore prima che `RABBITMQ_CLUSTER_DISK_USAGE_TOO_HIGH_FOR_INSTANCE_CHANGE` lo stato venga cancellato.

RabbitMQ su Amazon MQ: IAM Assume Role non valido

RabbitMQ su Amazon MQ genererà un codice INVALID_ASSUMEROLE per l'azione critica richiesta quando il ruolo IAM ARN specificato in `aws.arns.assume_role_arn` non è valido o non può essere assunto da Amazon MQ. Ciò può verificarsi quando il ruolo non esiste, si trova in un AWS account diverso da quello del broker o non dispone del necessario rapporto di fiducia con mq.amazonaws.com.

Un broker in quarantena RABBITMQ_INVALID_ASSUMEROLE non può recuperare le credenziali o i certificati richiesti per l'autenticazione LDAP, rendendo l'autenticazione LDAP non disponibile. Se LDAP è l'unico metodo di autenticazione configurato, gli utenti non saranno in grado di connettersi al broker. Il ruolo IAM è richiesto da Amazon MQ per accedere alle AWS risorse a cui si fa riferimento ARNs nella configurazione del broker, come Gestione dei segreti AWS i segreti o gli oggetti Amazon S3 utilizzati per l'autenticazione LDAP.

Diagnosi e risoluzione di RABBITMQ_INVALID_ASSUMEROLE

Per diagnosticare e risolvere il codice richiesto dall'azione RABBITMQ_INVALID_ASSUMEROLE, devi utilizzare Amazon Logs e la console. CloudWatch AWS Identity and Access Management

Per risolvere il problema «Assumi ruolo» non valido

1. Accedi ad Amazon CloudWatch Logs Insights ed esegui la seguente query sul gruppo `/aws/amazonmq/broker/<broker-id>/general` di log del tuo broker:

```
fields @timestamp, @message
| sort @timestamp desc
| filter @message like /error.*aws_arn_config/
| limit 10000
```

2. Cerca messaggi di errore simili a:

```
[error] <0.254.0> aws_arn_config: {handle_assume_role,{error,
{assume_role_failed,"AWS service is unavailable"}}}
```

3. Controlla la configurazione dei ruoli IAM e risolvi eventuali problemi come:
   - Assicurati che il ruolo esista nello stesso AWS account del broker
   - Verifica che la politica di fiducia consenta a mq.amazonaws.com di assumere il ruolo
   - Verifica che il ruolo disponga delle autorizzazioni appropriate per accedere alle risorse richieste AWS
4. Convalida la correzione utilizzando l'endpoint dell'API di convalida dell'accesso ARN prima di aggiornare la configurazione del broker.
5. Aggiorna la configurazione del broker e riavvia il broker.

RabbitMQ su Amazon MQ: ARN LDAP non valido

RabbitMQ su Amazon MQ genererà un codice INVALID_ARN_LDAP per l'azione critica richiesta quando l'ARN configurato per la password dell'account del servizio LDAP non è valido o è inaccessibile. Questo vale per le password specificate in or, che devono fare riferimento a segreti contenenti password in testo semplice. ARNs `aws.arns.auth_ldap.dn_lookup_bind.password` `aws.arns.auth_ldap.other_bind.password` Gestione dei segreti AWS

Un broker in quarantena RABBITMQ_INVALID_ARN_LDAP non può autenticarsi con l'account del servizio LDAP, rendendo l'autenticazione LDAP non disponibile. Se LDAP è l'unico metodo di autenticazione configurato, gli utenti non saranno in grado di connettersi al broker. L'invalidità ARNs può essere causata da una sintassi ARN non valida, da riferimenti a segreti inesistenti, da segreti situati in una AWS regione diversa da quella del broker o da un secretsmanager insufficiente: autorizzazioni nel ruolo IAM. GetSecretValue

Diagnosi e indirizzamento di RABBITMQ_INVALID_ARN_LDAP

Per diagnosticare e risolvere il codice richiesto dall'azione RABBITMQ_INVALID_ARN_LDAP, devi utilizzare Amazon Logs e la console. CloudWatch

Per risolvere il problema LDAP ARN non valido

1. Accedi ad Amazon CloudWatch Logs Insights ed esegui la seguente query sul gruppo `/aws/amazonmq/broker/<broker-id>/general` di log del tuo broker:

```
fields @timestamp, @message
| sort @timestamp desc
| filter @message like /error.*aws_arn_config/
| limit 10000
```

2. Cerca messaggi di errore simili a:

```
[error] <0.254.0> aws_arn_config: {<<"could not resolve
 ARN 'arn:aws:secretsmanager:xxx' for configuration
 'aws.arns.auth_ldap.dn_lookup_bind.password', error: \"AWS service is unavailable
\"">>,{error,"AWS service is unavailable"}}
```

3. Controlla il segreto di Secrets Manager e risolvi eventuali problemi come:
   - Verifica che il segreto esista nella stessa AWS regione del broker
   - Conferma che la sintassi ARN sia corretta
   - Assicurati che il ruolo IAM abbia secretsmanager: permessi GetSecretValue
4. Convalida la correzione utilizzando l'endpoint dell'API di convalida dell'accesso ARN prima di aggiornare la configurazione del broker.
5. Aggiorna la configurazione del broker e riavvia il broker.

RabbitMQ su Amazon MQ: ARN HTTP non valido

RabbitMQ su Amazon MQ genererà un codice INVALID_ARN_HTTP per l'azione critica richiesta quando uno o più certificati SSL o il file chiave per HTTP ARNs auth_backend non sono validi o sono inaccessibili. Questo vale per gli ARNS specificati in`aws.arns.auth_http.ssl_options.cacertfile`, `aws.arns.auth_http.ssl_options.certfile` o`aws.arns.auth_http.ssl_options.keyfile`, che devono fare riferimento a oggetti Gestione dei segreti AWS e segreti Amazon S3 contenenti certificati e chiave privata.

Un broker in quarantena RABBITMQ_INVALID_ARN_HTTP non può autenticarsi tramite il server HTTP. Se HTTP è l'unico metodo di autenticazione configurato, gli utenti non saranno in grado di connettersi al broker. L'invalidità ARNs può essere causata da una sintassi ARN non valida, da riferimenti a segreti inesistenti, da segreti situati in una AWS regione diversa da quella del broker o da permessi s3: /secretsmanager: GetObject insufficienti nel ruolo IAM. GetSecretValue

Diagnosi e indirizzamento di RABBITMQ_INVALID_ARN_HTTP

Per diagnosticare e risolvere il codice richiesto dall'azione RABBITMQ_INVALID_ARN_HTTP, devi utilizzare Amazon Logs e la console. CloudWatch

Per risolvere il problema HTTP ARN non valido

1. Accedi ad Amazon CloudWatch Logs Insights ed esegui la seguente query sul gruppo `/aws/amazonmq/broker/<broker-id>/general` di log del tuo broker:

```
fields @timestamp, @message
| sort @timestamp desc
| filter @message like /error.*aws_arn_config/
| limit 10000
```

2. Cerca messaggi di errore simili a:

```
[error] <0.209.0> aws_arn_config: {<<"could not resolve ARN 'arn:aws:s3:::xxxx' for
 configuration 'aws.arns.auth_http.ssl_options.certfile', error: \"AWS service is
 unavailable\"">>,{error,"AWS service is unavailable"}}
```

3. Controlla il segreto di S3 Object/Secrets Manager e risolvi eventuali problemi come:

   - Verifica che la risorsa esista nella stessa AWS regione del broker
   - Conferma che la sintassi ARN sia corretta
   - Assicurati che il ruolo IAM abbia le autorizzazioni s3: GetObject e secretsmanager: GetSecretValue

4. Convalida la correzione utilizzando l'endpoint dell'API di convalida dell'accesso ARN prima di aggiornare la configurazione del broker.
5. Aggiorna la configurazione del broker e riavvia il broker.

RabbitMQ su Amazon MQ: ARN SSL non valido

RabbitMQ su Amazon MQ genererà un codice INVALID_ARN_SSL per l'azione critica richiesta quando uno o più ARNs certificati CA truststore for EXTERNAL auth_mechanism non sono validi o sono inaccessibili. Questo vale per gli ARNS specificati in `aws.arns.ssl_options.cacertfile` o`aws.arns.management.ssl.cacertfile`, che devono fare riferimento all'oggetto Amazon S3 o ACM PCA contenente il certificato.

Un broker in quarantena RABBITMQ_INVALID_ARN_SSL non può autenticare i certificati client durante gli handshake TLS reciproci perché non è configurato alcun truststore valido. Se il meccanismo di autenticazione EXTERNAL è l'unico metodo di autenticazione configurato, gli utenti non saranno in grado di connettersi al broker. L'errore ARNs può essere causato da una sintassi ARN non valida, da riferimenti a oggetti S3 inesistenti, da oggetti S3 situati in una AWS regione diversa da quella del broker o da permessi s3: /acm-pca: insufficienti nel ruolo IAM. GetObject GetCertificateAuthorityCertificate

Diagnosi e indirizzamento di RABBITMQ_INVALID_ARN_SSL

Per diagnosticare e risolvere il codice richiesto dall'azione RABBITMQ_INVALID_ARN_SSL, devi utilizzare Amazon Logs e la console. CloudWatch

Per risolvere il problema dell'ARN SSL non valido

1. Accedi ad Amazon CloudWatch Logs Insights ed esegui la seguente query sul gruppo `/aws/amazonmq/broker/<broker-id>/general` di log del tuo broker:

```
fields @timestamp, @message
| sort @timestamp desc
| filter @message like /error.*aws_arn_config/
| limit 10000
```

2. Cerca messaggi di errore simili a:

```
[error] <0.209.0> aws_arn_config: {<<"could not resolve ARN 'arn:aws:acm-pca:xxxx'
 for configuration 'aws.arns.ssl_options.cacertfile', error: \"AWS service is
 unavailable\"">>,{error,"AWS service is unavailable"}}
```

3. Controlla l'oggetto S3/ACM-PCA e risolvi eventuali problemi come:
   - Verifica che il segreto esista nella stessa regione del broker AWS
   - Conferma che la sintassi ARN sia corretta
   - Assicurati che il ruolo IAM disponga delle autorizzazioni s3: /acm-pca: GetObject GetCertificateAuthorityCertificate
4. Convalida la correzione utilizzando l'endpoint dell'API di convalida dell'accesso ARN prima di aggiornare la configurazione del broker.
5. Aggiorna la configurazione del broker e riavvia il broker.

RabbitMQ su Amazon MQ: ARN non valido

RabbitMQ su Amazon MQ genererà un codice INVALID_ARN critical action required quando una o più ARNs configurazioni nel broker non sono valide o inaccessibili. Questo vale ARNs per i certificati SSL, Gestione dei segreti AWS i segreti, gli oggetti Amazon S3 o AWS altri riferimenti di risorse non coperti da codici di quarantena più specifici come RABBITMQ_INVALID_ARN_LDAP o RABBITMQ_INVALID_ASSUMEROLE.

Un broker in quarantena RABBITMQ_INVALID_ARN potrebbe presentare funzionalità degradate a seconda delle funzionalità non valide. ARNs Le funzionalità che dipendono dalle risorse inaccessibili non saranno disponibili e il broker registrerà gli errori indicando quali ARN non è riuscito a risolvere. L'impatto sulla disponibilità del broker dipende dalla necessità o meno di un ARN non valido per le operazioni critiche del broker.

Diagnosi e indirizzamento di RABBITMQ_INVALID_ARN

Per diagnosticare e risolvere il codice richiesto dall'azione RABBITMQ_INVALID_ARN, devi utilizzare Amazon CloudWatch Logs e la console di servizio appropriata per la risorsa interessata. AWS

Per risolvere il problema ARN non valido

1. Accedi ad Amazon CloudWatch Logs Insights ed esegui la seguente query sul gruppo `/aws/amazonmq/broker/<broker-id>/general` di log del tuo broker:

```
fields @timestamp, @message
| sort @timestamp desc
| filter @message like /error.*aws_arn_config/
```

```
| limit 10000
```

2. Cerca messaggi di errore simili a:

```
[error] <0.254.0> aws_arn_config: {<<"could not resolve ARN
 'arn:aws:s3:::bucket-name/certificate.pem' for configuration
 'aws.arns.auth_ldap.ssl_options.cacertfile', error: \"AWS service is unavailable
\"">>,{error,"AWS service is unavailable"}}
```

3. Controlla la AWS risorsa e risolvi eventuali problemi come:

   - Verifica che la risorsa esista nella stessa AWS regione del broker
   - Conferma che la sintassi ARN sia corretta
   - Assicurati che il ruolo IAM disponga delle autorizzazioni appropriate per accedere alla risorsa

4. Convalida la correzione utilizzando l'endpoint dell'API di convalida dell'accesso ARN prima di aggiornare la configurazione del broker.
5. Aggiorna la configurazione del broker e riavvia il broker.

Risorse correlate

Risorse di Amazon MQ

Nella tabella seguente vengono elencate le risorse utili per l'utilizzo di Amazon MQ.

Risorsa	Descrizione
Riferimento all'API REST di Amazon MQ	Descrizione di risorse REST, richieste di esempio, metodi HTTP, schemi, parametri ed errori restituiti dal servizio.
Amazon MQ nel riferimento ai AWS CLI comandi	Descrizioni dei AWS CLI comandi che puoi utilizzare per lavorare con i broker di messaggi.
Amazon MQ nella guida AWS CloudFormation per l'utente	La risorsa `AWS::Amazon MQ::Broker` ti permette di creare i broker Amazon MQ, aggiungere modifiche di configurazione o modificare gli utenti per il broker specificato, restituire le informazioni sulil broker specificato oppure eliminarlo. La risorsa `AWS::Amazon MQ::Configuration` permette di creare configurazioni Amazon MQ, aggiungere modifiche alla configurazione o modificare utenti e restituire informazioni sulla configurazione specificata.
Regioni ed endpoint	Informazioni su regioni ed endpoint di Amazon MQ;
Pagina del prodotto	La pagina Web principale che include informazioni su Amazon MQ.
Forum di discussione	Forum basato su community per sviluppatori per la discussione di questioni tecniche correlate ad Amazon MQ.

Risorsa	Descrizione
AWS Informazioni sull'assistenza Premium	La pagina Web principale per informazioni su AWS Premium Support one-on-one, un canale di supporto a risposta rapida che consente di creare ed eseguire applicazioni sui AWS servizi di infrastruttura

Amazon MQ per risorse ActiveMQ

Nella tabella seguente vengono elencate le risorse utili per l'utilizzo di Apache ActiveMQ.

Risorsa	Descrizione
Guida alle operazioni di base di Apache ActiveMQ	La documentazione ufficiale di Apache ActiveMQ.
ActiveMQ in Action	Una guida ad Apache ActiveMQ che illustra l'anatomia di messaggi JMS, connettori, persistenza messaggi, autenticazione e autorizzazione.
Client multilinguaggio	Un elenco di linguaggi di programmazione e librerie Apache ActiveMQ corrispondenti. Consulta anche ActiveMQ Client e QpidJMS Client.

Amazon MQ per risorse RabbitMQ

Nella tabella seguente vengono elencate le risorse utili per l'utilizzo di RabbitMQ.

Risorsa	Descrizione
La guida introduttiva di RabbitMQ	La documentazione ufficiale di RabbitMQ.

Risorsa	Descrizione
Librerie client e strumenti per sviluppatori di RabbitMQ	Una guida alle librerie client ufficialmente supportate e agli strumenti di sviluppo per lavorare con RabbitMQ utilizzando una varietà di linguaggi di programmazione e piattaforme.
Le migliori pratiche di RabbitMQ	Guida di CloudAMQP alle best practice e alle raccomandazioni per l'utilizzo di RabbitMQ.

Note di rilascio di Amazon MQ

La seguente tabella elenca versioni e miglioramenti delle funzionalità di Amazon MQ.

Data	Aggiornamento della documentazione
19 febbraio 2026	Amazon MQ Amazon MQ ora supporta ActiveMQ 5.19, una nuova versione secondaria del motore. Per ulteriori informazioni, consultare la pagina • Pagina di rilascio di ActiveMQ 5.19 • Gestione di Amazon MQ per le versioni del motore ActiveMQ • Aggiornamento di una versione del motore del broker Amazon MQ • Utilizzo dei file di configurazione Spring XML
22 gennaio 2026	Amazon MQ ora supporta il plug-in di scambio di argomenti JMS per broker su RabbitMQ 4.2 e versioni successive. Puoi utilizzare il client JMS ufficiale di RabbitMQ per eseguire carichi di lavoro JMS sul broker Amazon MQ for RabbitMQ. Supporta JMS 1.1, 2.0 e 3.1. Per ulteriori informazioni, consultare la pagina • Specifiche ufficiali JMS 2.0 (retrocompatibile con JMS 1.1 esteso) • Specifiche ufficiali JMS 3.1 • Limitazione del client RabbitMQ JMS • Connessione dell'applicazione JMS al broker Amazon MQ per RabbitMQ
8 gennaio 2026	Amazon MQ ora supporta l'autenticazione dei certificati SSL per i broker su RabbitMQ 4.2 e versioni successive utilizzando certificati client X.509 e la configurazione TLS (mTLS) reciproca. Puoi configurare l'autenticazione dei certificati SSL e gli MTL tramite Console di gestione AWS, AWS CloudForm ation AWS CLI, o AWS CDK in tutti i paesi in Regioni AWS cui è disponibile Amazon MQ. Per ulteriori informazioni, consulta Autenticazione del certificato SSL e Configurazione degli MTL.

Data	Aggiornamento della documentazione
6 gennaio 2026	Amazon MQ ora supporta l'autenticazione e l'autorizzazione HTTP per i broker su RabbitMQ 4.2 e versioni successive con server HTTP esterni. Puoi configurare l'autenticazione HTTP tramite Console di gestione AWS, AWS CloudFormation AWS CLI, o AWS CDK in tutti i Regioni AWS casi in cui Amazon MQ è disponibile. Per ulteriori informazioni, consulta Autenticazione e autorizzazione HTTP.
20 novembre 2025	Amazon MQ supporta ora RabbitMQ 4.2, una nuova versione principale che introduce il supporto nativo per il protocollo AMQP 1.0, un nuovo archivio di metadati Khepri basato su Raft, pale locali e priorità dei messaggi per le code di quorum. RabbitMQ 4.2 include anche varie correzioni di bug e miglioramenti delle prestazioni per la velocità effettiva e la gestione della memoria. Sebbene questa versione introduca nuove funzionalità, ci sono alcune modifiche importanti. Per ulteriori informazioni, consultare la pagina • RabbitMQ 4 • Note di rilascio open source di RabbitMQ • Configurazione dei limiti delle risorse • Protocolli supportati • Aggiornamenti della versione di Amazon MQ
18 novembre 2024	Amazon MQ ora supporta istanze m7g basate su Graviton3 per RabbitMQ in una gamma di dimensioni, da medie a 16xlarge in Africa (Città del Capo). Per ulteriori informazioni, consulta Tipi di istanze del broker Amazon MQ per RabbitMQ.

Data	Aggiornamento della documentazione
17 novembre 2025	Amazon MQ ora supporta l'autenticazione e l'autorizzazione LDAP per i broker RabbitMQ con servizi di directory LDAP esterni. Puoi configurare LDAP tramite Console di gestione AWS, AWS CloudFormation AWS CLI, o AWS CDK in tutti i Regioni AWS casi in cui Amazon MQ è disponibile. Per ulteriori informazioni, consulta Autenticazione e autorizzazione LDAP per Amazon MQ for RabbitMQ.
22 ottobre 2025	Amazon MQ è ora disponibile nella regione Asia Pacifico (Nuova Zelanda). Per informazioni sulle regioni disponibili, consulta AWS Regioni ed endpoint in Riferimenti generali di AWS .
3 settembre 2025	Amazon MQ ora supporta l'autenticazione e l'autorizzazione OAuth 2.0 per i broker RabbitMQ con provider di identità pubbliche (). IdPs Puoi configura re la OAuth versione 2.0 tramite Console di gestione AWS AWS CloudForm ation AWS CLI, o AWS CDK in tutti i Regioni AWS casi in cui Amazon MQ è disponibile. Per ulteriori informazioni, consulta OAuth autenticazione e autorizzazione 2.0 per Amazon MQ for RabbitMQ.
22 luglio 2025	Amazon MQ ora supporta `m7g` istanze basate su Graviton3 per RabbitMQ in una gamma di dimensioni, da medie a 16xlarge. I cluster RabbitMQ in esecuzione su `m7g` istanze offrono una capacità di carico di lavoro fino al 50% superiore e miglioramenti del throughput fino all'85% rispetto ai cluster Amazon MQ for RabbitMQ comparabili in esecuzione su istanze. `m5` `M7g`le istanze hanno anche dimensioni del volume del disco ottimizzate che variano in base alla dimensione dell'istanza. Per ulteriori informazioni, consulta Broker instance types. `M7g`le istanze su Amazon MQ sono oggi disponibili in tutte le regioni generalmente disponibili ad eccezione delle regioni di Africa (Città del Capo), Canada occidentale (Calgary) ed Europa (Milano).

Data	Aggiornamento della documentazione
8 luglio 2025	Amazon MQ è ora disponibile nella regione Asia Pacifico (Taipei). Per informazioni sulle regioni disponibili, consulta AWS Regioni ed endpoint in Riferimenti generali di AWS .
22 aprile 2025	Ora puoi eliminare le configurazioni del broker Amazon MQ utilizzando l'`DeleteConfiguration` API. Per ulteriori informazioni, consulta Configurazioni nell'Amazon MQ API Reference.
16 aprile 2025	Amazon MQ for RabbitMQ ora supporta l'utilizzo di endpoint dual-stack (IPv4 and IPv6) per connettersi a broker pubblici e privati. Per ulteriori informazioni, consultare Connecting to Amazon MQ e Configuring a private Amazon MQ broker.
7 aprile 2025	Amazon MQ è ora disponibile nelle regioni di Asia Pacifico (Tailandia) e Messico (Centrale). Per informazioni sulle regioni disponibili, consulta AWS Regioni ed endpoint in Riferimenti generali di AWS .
13 febbraio 2025	Gli endpoint FIPS API di Amazon MQ sono ora disponibili nelle regioni Canada (Centrale) e Canada occidentale (Calgary). Per ulteriori informazioni sull'utilizzo degli endpoint FIPS con l'API Amazon MQ, consulta. Connecting to Amazon MQ Per informazioni sulle regioni disponibili, consulta AWS Regioni ed endpoint in Riferimenti generali di AWS .

Data	Aggiornamento della documentazione
12 febbraio 2025	Amazon MQ annuncia le seguenti date di fine del supporto per i tipi di istanze: Broker instance types • `mq.t2.micro` ActiveMQ: 12 maggio 2025 • `mq.m4.large` ActiveMQ: 12 maggio 2025 Non è possibile creare broker a partire dal `mq.t2.micro` 17 marzo `mq.m4.large` 2025.
10 dicembre 2024	Amazon MQ ora supporta l'utilizzo AWS PrivateLink per la connessione tra i tuoi cloud privati virtuali (VPCs) e l'API Amazon MQ senza esporre il traffico alla rete Internet pubblica. Per ulteriori informazioni, consulta the section called “Connect ad Amazon MQ tramite AWS PrivateLink”.
18 novembre 2024	Amazon MQ è ora disponibile nella regione Asia Pacifico (Malesia). Per informazioni sulle regioni disponibili, consulta AWS Regioni ed endpoint in Riferimenti generali di AWS .
14 novembre 2024	Amazon MQ annuncia le seguenti date di fine del supporto per le versioni del motore: Gestione di Amazon MQ per le versioni del motore ActiveMQ • ActiveMQ 5.17:16 giugno 2025 Gestione delle versioni del motore Amazon MQ per RabbitMQ • RabbitMQ 3.11:17 febbraio 2025 • RabbitMQ 3.12:17 marzo 2025 Per ulteriori informazioni sull'aggiornamento alla versione più recente, vedere Aggiornamento di una versione del motore del broker Amazon MQ

Data	Aggiornamento della documentazione
13 novembre 2024	Amazon MQ ora supporta gli endpoint di servizio dual-stack a cui puoi connetterti utilizzando uno o l'altro. IPv4 IPv6 Gli endpoint del servizio regionale dual-stack Amazon MQ possono essere risolti sia con record DNS che A con record DNS. AAAA Per ulteriori informazioni, consulta ???.
25 luglio 2024	Amazon MQ ora supporta ActiveMQ 5.18, una nuova versione secondaria del motore. Per ulteriori informazioni, consulta gli argomenti seguenti: • Pagina di rilascio di ActiveMQ 5.18 • Gestione di Amazon MQ per le versioni del motore ActiveMQ • Aggiornamento di una versione del motore del broker Amazon MQ • Utilizzo dei file di configurazione Spring XML
22 luglio 2024	Amazon MQ ora supporta le code quorum solo sui broker che utilizzano la versione 3.13 e successive. Le code quorum sono un tipo di coda FIFO replicato che utilizza l'algoritmo di consenso Raft per mantenere la coerenza dei dati. Le code quorum forniscono una gestione dei messaggi avvelenati, che può aiutarti a gestire i messaggi non elaborati. Per iniziare a usare le code quorum, consulta. Code quorum per RabbitMQ su Amazon MQ

Data	Aggiornamento della documentazione
2 luglio 2024	Amazon MQ for RabbitMQ ora supporta RabbitMQ 3.13, una versione secondaria. Per tutti i broker che utilizzano la versione 3.13 e successive del motore, Amazon MQ gestisce gli aggiornamenti all'ultima versione di patch supportata durante la finestra di manutenzione. Per ulteriori informazioni, consulta Aggiornamento di una versione del motore del broker Amazon MQ. Linee guida per il dimensionamento di Amazon MQ for RabbitMQsono state aggiornate per includere nuovi limiti per le code, i consumatori per canale e i vantaggi per i broker che utilizzano la versione 3.13 del motore. Per ulteriori informazioni sulle correzioni e le funzionalità di questa versione, consulta le note di rilascio di RabbitMQ 3.13 nell'archivio del server RabbitMQ. GitHub Per ulteriori informazioni sulle versioni supportate di Amazon MQ per RabbitMQ e sugli aggiornamenti dei broker, consulta Gestione delle versioni del motore Amazon MQ per RabbitMQ.
10 giugno 2024	Amazon MQ è ora disponibile nella regione Canada occidentale (Calgary). Per informazioni sulle regioni disponibili, consulta AWS Regioni ed endpoint in Riferimenti generali di AWS .

Data	Aggiornamento della documentazione
10 maggio 2024	Il calendario di supporto della versione di Amazon MQ indica quando una versione del motore di brokeraggio raggiunge la fine del supporto. Quando una versione del motore raggiunge la fine del supporto, Amazon MQ aggiorna automaticamente tutti i broker della versione alla versione secondaria successiva supportata. Amazon MQ fornisce un preavviso di almeno 90 giorni prima che una versione del motore raggiunga la fine del supporto. Per visualizzare il calendario di supporto della versione e la fine del supporto, consulta quanto segue: • Gestione di Amazon MQ per le versioni del motore ActiveMQ • Gestione delle versioni del motore Amazon MQ per RabbitMQ Puoi anche abilitare gli aggiornamenti automatici delle versioni secondarie per consentire al tuo broker di eseguire l'aggiornamento alla versione di patch successiva durante una finestra di manutenzione. Per ulteriori informazioni, consulta Aggiornamento di una versione del motore del broker Amazon MQ
9 maggio 2024	Amazon MQ for RabbitMQ ora supporta RabbitMQ 3.12, una versione secondaria. Tutti i broker della versione 3.12.13 e successive utilizzano Classic Queues versione 2 (CQv2) e tutte le code della versione 3.12.13 e successive si comportano come code pigre. Consigliamo ai broker con versioni precedenti alla 3.12.13 di abilitare CQv2 e rallentare le code o di eseguire l'aggiornamento alla versione più recente di Amazon MQ for RabbitMQ. Per ulteriori informazioni sulle correzioni e funzionalità di questa versione, consultare quanto segue: • Note di rilascio di RabbitMQ 3.12 nell'archivio del server RabbitMQ. GitHub Per ulteriori informazioni sulle versioni supportate di Amazon MQ per RabbitMQ e sugli aggiornamenti dei broker, consulta Gestione delle versioni del motore Amazon MQ per RabbitMQ.

Data	Aggiornamento della documentazione
4 marzo 2024	Amazon MQ per RabbitMQ ora supporta RabbitMQ 3.11.28. Per ulteriori informazioni sulle correzioni e funzionalità di questa versione, consultare quanto segue: • Note di rilascio di RabbitMQ 3.11.28 nell'archivio del server RabbitMQ GitHub • Changelog di RabbitMQ Per ulteriori informazioni sulle versioni supportate di Amazon MQ per RabbitMQ e sugli aggiornamenti dei broker, consulta Gestione delle versioni del motore Amazon MQ per RabbitMQ.
19 gennaio 2024	Amazon MQ for RabbitMQ non supporta il nome utente «guest» ed eliminerà l'account ospite predefinito quando crei un nuovo broker. Amazon MQ eliminerà inoltre periodicamente qualsiasi account creato dal cliente chiamato «ospite».
15 dicembre 2023	Amazon MQ è ora disponibile nella regione Israele (Tel Aviv). Per informazioni sulle regioni disponibili, consulta AWS Regioni ed endpoint in Riferimenti generali di AWS .
11 dicembre 2023	Amazon MQ per RabbitMQ ora supporta RabbitMQ 3.10.25. Per ulteriori informazioni sulle correzioni e funzionalità di questa versione, consultare quanto segue: • Note di rilascio di RabbitMQ 3.10.25 nell'archivio del server RabbitMQ GitHub • Changelog di RabbitMQ Per ulteriori informazioni sulle versioni supportate di Amazon MQ per RabbitMQ e sugli aggiornamenti dei broker, consulta Gestione delle versioni del motore Amazon MQ per RabbitMQ.

Data	Aggiornamento della documentazione
26 ottobre 2023	Amazon MQ ha rilasciato le ultime versioni secondarie di ActiveMQ 5.15.16, 5.16.7, 5.17.6 con un aggiornamento critico. Abbiamo reso obsolete le versioni secondarie precedenti di ActiveMQ e aggiorneremo tutti i broker su qualsiasi versione dalla 5.15 alla 5.15.16, o dalla 5.16 alla 5.16.7 e dalla 5.17 alla 5.17.6. Per ulteriori informazioni sull'aggiornamento del broker ActiveMQ, consulta Gestione di Amazon MQ per le versioni del motore ActiveMQ.
27 settembre 2023	Amazon MQ per RabbitMQ ora supporta RabbitMQ 3.11.20. Per ulteriori informazioni sulle correzioni e funzionalità di questa versione, consultare quanto segue: • Note di rilascio di RabbitMQ 3.11.20 sul repository del server RabbitMQ GitHub • Changelog di RabbitMQ Per ulteriori informazioni sulle versioni supportate di Amazon MQ per RabbitMQ e sugli aggiornamenti dei broker, consulta Gestione delle versioni del motore Amazon MQ per RabbitMQ.
27 luglio 2023	Amazon MQ per RabbitMQ ora supporta RabbitMQ 3.11.16 Per ulteriori informazioni sulle correzioni e funzionalità di questa versione, consultare quanto segue: • Note di rilascio di RabbitMQ 3.11.16 sul repository del server RabbitMQ GitHub • Changelog di RabbitMQ Per ulteriori informazioni sulle versioni supportate di Amazon MQ per RabbitMQ e sugli aggiornamenti dei broker, consulta Gestione delle versioni del motore Amazon MQ per RabbitMQ.

Data	Aggiornamento della documentazione
27 luglio 2023	Amazon MQ per RabbitMQ ora supporta la creazione e l'applicazione di configurazioni al broker RabbitMQ. Per ulteriori informazioni sull'aggiunta di configurazioni al broker, consulta RabbitMQ Broker Configurations. Per ulteriori informazioni sull'utilizzo di questa funzionalità, consulta: • Policy per gli operatori • Modifiche alle policy degli operatori
23 giugno 2023	Amazon MQ supporta ora ActiveMQ 5.17.3, una nuova versione secondaria del motore. Questa versione supporta la nuova funzionalità di replica dei dati tra regioni (CRDR) di Amazon MQ. Per ulteriori informazioni, consulta gli argomenti seguenti: • Per iniziare a usare CRDR, consulta Replica dei dati tra regioni per Amazon MQ per ActiveMQ, nella Guida per sviluppatori. • Pagina di rilascio di ActiveMQ 5.17.3 • Gestione di Amazon MQ per le versioni del motore ActiveMQ • Aggiornamento di una versione del motore del broker Amazon MQ • Utilizzo dei file di configurazione Spring XML
21 giugno 2023	Amazon MQ for ActiveMQ offre ora una funzionalità CRDR (Cross-Region Data Replication) che consente la replica asincrona dei messaggi dal broker principale in una regione primaria al broker di replica in una AWS regione di replica. Se il broker primario nella regione primaria restituisce un errore, è possibile promuovere il broker di replica nella regione secondaria a primario avviando uno switchover o un failover. Per iniziare a usare CRDR, consulta Replica dei dati tra regioni per Amazon MQ per ActiveMQ, nella Guida per sviluppatori.

Data	Aggiornamento della documentazione
18 maggio 2023	Amazon MQ è ora disponibile nelle seguenti regioni: • Asia Pacifico (Melbourne) • Asia Pacifico (Hyderabad) • Europa (Spagna) • Europa (Zurigo) Per informazioni sulle regioni disponibili, consulta AWS Regioni ed endpoint in Riferimenti generali di AWS .
14 aprile 2023	Amazon MQ per RabbitMQ supporta ora RabbitMQ versione 3.9.27. Per ulteriori informazioni sulle correzioni e funzionalità di questa versione, consultare quanto segue: • Note di rilascio di GitHub RabbitMQ 3.9.27 nell'archivio del server RabbitMQ • Changelog di RabbitMQ Per ulteriori informazioni sulle versioni supportate di Amazon MQ per RabbitMQ e sugli aggiornamenti dei broker, consulta Gestione delle versioni del motore Amazon MQ per RabbitMQ.

Data	Aggiornamento della documentazione
14 aprile 2023	Amazon MQ per RabbitMQ supporta ora RabbitMQ versione 3.10.20. Per ulteriori informazioni sulle correzioni e funzionalità di questa versione, consultare quanto segue: • Note di rilascio di RabbitMQ 3.10.20 sul repository del server RabbitMQ GitHub • Changelog di RabbitMQ Per ulteriori informazioni sulle versioni supportate di Amazon MQ per RabbitMQ e sugli aggiornamenti dei broker, consulta Gestione delle versioni del motore Amazon MQ per RabbitMQ.
31 marzo 2023	Amazon MQ for RabbitMQ ha disabilitato la versione 3.10.17 del motore RabbitMQ. Il team di Amazon MQ for RabbitMQ e i manutentori open source di RabbitMQ hanno identificato un problema con la console di gestione RabbitMQ nella versione 3.10.17. Amazon MQ ha ritirato questa versione. Per mitigare l'impatto di questo problema, crea nuovi broker con la versione 3.10.10 mentre lavoriamo per supportare una nuova versione patch di RabbitMQ. Si consiglia di attivare l'opzione di aggiornamento della versione per ottenere automaticamente le ultime correzioni di bug, gli aggiornamenti di sicurezza e i miglioramenti delle prestazioni. Per ulteriori informazioni sulle versioni disponibili di Amazon MQ per RabbitMQ, consulta le versioni del motore Amazon MQ per RabbitMQ.

Data	Aggiornamento della documentazione
1 marzo 2023	Amazon MQ per RabbitMQ supporta ora RabbitMQ versione 3.10.17. Per ulteriori informazioni sulle correzioni e funzionalità di questa versione, consultare quanto segue: • Note di rilascio di RabbitMQ 3.10.17 nell'archivio del server RabbitMQ GitHub • Changelog di RabbitMQ Per ulteriori informazioni sulle versioni supportate di Amazon MQ per RabbitMQ e sugli aggiornamenti dei broker, consulta Gestione delle versioni del motore Amazon MQ per RabbitMQ.
21 febbraio 2023	Amazon MQ for RabbitMQ ora si integra con AWS Key Management Service (KMS) per offrire la crittografia lato server. Ora puoi selezionare la tua CMK gestita dal cliente o utilizzare una AWS chiave KMS gestita nel tuo account. AWS KMS Per ulteriori informazioni, consulta Crittografia dei dati a riposo. Amazon MQ supporta l'utilizzo delle AWS KMS chiavi nei seguenti modi. • Amazon MQ owned KMS key (default) (Chiave KMS di proprietà di Amazon MQ (di default)): la chiave è di proprietà ed è gestita da Amazon MQ e non è presente nel tuo account. • AWS chiave KMS gestita: la chiave KMS AWS gestita (`aws/mq`) è una chiave KMS nel tuo account che viene creata, gestita e utilizzata per tuo conto da Amazon MQ. • Select existing customer managed KMS key (Seleziona chiave KMS esistente gestita dal cliente): le chiavi KMS gestite dal cliente vengono create e gestite da te in AWS Key Management Service (KMS).

Data	Aggiornamento della documentazione
13 gennaio 2023	Amazon MQ per RabbitMQ supporta ora RabbitMQ versione 3.8.34. Per ulteriori informazioni sulle correzioni e funzionalità di questa versione, consultare quanto segue: • Note di rilascio di RabbitMQ 3.8.34 sul repository del server RabbitMQ GitHub • Changelog di RabbitMQ Per ulteriori informazioni sulle versioni supportate di Amazon MQ per RabbitMQ e sugli aggiornamenti dei broker, consulta Gestione delle versioni del motore Amazon MQ per RabbitMQ.
15 dicembre 2022	Amazon MQ per RabbitMQ supporta ora RabbitMQ versione 3.9.24. Per ulteriori informazioni sulle correzioni e funzionalità di questa versione, consultare quanto segue: • Note di rilascio di RabbitMQ 3.9.24 sul repository del server RabbitMQ GitHub • Changelog di RabbitMQ Per ulteriori informazioni sulle versioni supportate di Amazon MQ per RabbitMQ e sugli aggiornamenti dei broker, consulta Gestione delle versioni del motore Amazon MQ per RabbitMQ.
13 dicembre 2022	Amazon MQ è ora disponibile nella regione Medio Oriente (Emirati Arabi Uniti). Per informazioni sulle regioni disponibili, consulta AWS Regioni ed endpoint in Riferimenti generali di AWS .

Data	Aggiornamento della documentazione
14 novembre 2022	Amazon MQ per RabbitMQ supporta ora la versione 3.10, una versione principale del motore. Ora puoi abilitare la versione classica di Queues 2 () sulle tue code RabbitMQ. CQv2 Gli aggiornamenti diretti dalla versione 3.8 alla 3.10 non sono supportati. Per ulteriori informazioni, consulta gli argomenti seguenti: • Note di rilascio di RabbitMQ 3.10.10 • Changelog di RabbitMQ Per ulteriori informazioni sulle versioni supportate di Amazon MQ per RabbitMQ e sugli aggiornamenti dei broker, consulta Gestione delle versioni del motore Amazon MQ per RabbitMQ.
09 novembre 2022	Amazon MQ supporta ora ActiveMQ 5.17.2, una versione secondaria del motore. Per ulteriori informazioni, consulta gli argomenti seguenti: • Pagina di rilascio di ActiveMQ 5.17.2 • Gestione di Amazon MQ per le versioni del motore ActiveMQ • Aggiornamento di una versione del motore del broker Amazon MQ • Utilizzo dei file di configurazione Spring XML
17 agosto 2022	Amazon MQ ora supporta ActiveMQ 5.17.1, una nuova versione principale del motore. Per ulteriori informazioni, consulta gli argomenti seguenti: • Pagina di rilascio di ActiveMQ 5.17.1 • Gestione di Amazon MQ per le versioni del motore ActiveMQ • Aggiornamento di una versione del motore del broker Amazon MQ • Utilizzo dei file di configurazione Spring XML

Data	Aggiornamento della documentazione
14 luglio 2022	Amazon MQ supporta ora ActiveMQ 5.16.5, una versione secondaria del motore. Per ulteriori informazioni, consulta gli argomenti seguenti: • Pagina di rilascio di ActiveMQ 5.16.5 • Gestione di Amazon MQ per le versioni del motore ActiveMQ • Utilizzo dei file di configurazione Spring XML • Aggiornamento di una versione del motore del broker Amazon MQ
4 maggio 2022	Amazon MQ aggiunge un linguaggio inclusivo per l'elemento `networkConnector` nella configurazione del broker. • Creazione e configurazione di una rete di broker Amazon MQ
25 aprile 2022	Amazon MQ Questa versione aggiunge lo stato broker `CRITICAL_ACTION_REQUIRED` e la Proprietà API `ActionRequired` . `CRITICAL_ACTION_REQUIRED` ti informa quando il tuo broker è degradato. `ActionRequired` fornisce un codice che è possibile utilizzare per trovare istruzioni nella Guida per lo sviluppatore su come risolvere il problema. • Risoluzione dei problemi • `ActionRequired` documentazione nei Riferimenti dell'API di Amazon MQ.
20 aprile 2022	Amazon MQ ora supporta ActiveMQ 5.16.4, una versione secondaria del motore. Per ulteriori informazioni, consulta gli argomenti seguenti: • Pagina di rilascio di ActiveMQ 5.16.4 • Gestione di Amazon MQ per le versioni del motore ActiveMQ • Utilizzo dei file di configurazione Spring XML • Aggiornamento di una versione del motore del broker Amazon MQ
1° marzo 2022	Amazon MQ è ora disponibile nella Regione Asia Pacifico (Jakarta). Per informazioni sulle regioni disponibili, consulta AWS Regioni ed endpoint in Riferimenti generali di AWS .

Data	Aggiornamento della documentazione
25 febbraio 2022	Amazon MQ per RabbitMQ ora supporta RabbitMQ versione 3.8.27. Per ulteriori informazioni sulle correzioni e funzionalità di questa versione, consultare quanto segue: • Note di rilascio di RabbitMQ 3.8.27 sul repository del server RabbitMQ GitHub • Changelog di RabbitMQ Per ulteriori informazioni sulle versioni supportate di Amazon MQ per RabbitMQ e sugli aggiornamenti dei broker, consulta Gestione delle versioni del motore Amazon MQ per RabbitMQ.
16 febbraio 2022	Amazon MQ è ora disponibile nella Regione Africa (Città del Capo). Per informazioni sulle regioni disponibili, consulta AWS Regioni ed endpoint in Riferimenti generali di AWS .

Data	Aggiornamento della documentazione
14 febbraio 2022	Amazon MQ per RabbitMQ ora supporta RabbitMQ versione 3.9.13. Aggiornamenti a versioni secondarie automatiche non può essere utilizzat o per eseguire l'aggiornamento da Rabbit 3.8 a 3.9. Per farlo, aggiornare manualmente il broker. Per ulteriori informazioni sulle nuove funzionalità introdotte in RabbitMQ 3.9, consulta la pagina delle note di rilascio per la versione 3.9.0 sul sito Web. GitHub **Note** Al momento, Amazon MQ non supporta i flussi, o utilizzando la registrazione strutturata in JSON, introdotta in RabbitMQ 3.9. Per ulteriori informazioni sulle correzioni e funzionalità di questa versione, consultare quanto segue: • Note di rilascio di RabbitMQ 3.9.13 nell'archivio del server RabbitMQ GitHub • Changelog di RabbitMQ Per ulteriori informazioni sulle versioni supportate di Amazon MQ per RabbitMQ e sugli aggiornamenti dei broker, consulta Gestione delle versioni del motore Amazon MQ per RabbitMQ.
7 febbraio 2022	Amazon MQ per RabbitMQ introduce nuovi parametri broker, consentendo di monitorare l'utilizzo medio delle risorse su tutti i 3 nodi di un'implementazione cluster. Per ulteriori informazioni, consulta gli argomenti seguenti: • the section called “Metriche per RabbitMQ”

Data	Aggiornamento della documentazione
18 gennaio 2022	Amazon MQ per RabbitMQ ora supporta RabbitMQ versione 3.8.26. Per ulteriori informazioni sulle correzioni e funzionalità di questa versione, consultare quanto segue: • Note di rilascio di RabbitMQ 3.8.26 sul repository del server RabbitMQ GitHub • Changelog di RabbitMQ Per ulteriori informazioni sulle versioni supportate di Amazon MQ per RabbitMQ e sugli aggiornamenti dei broker, consulta Gestione delle versioni del motore Amazon MQ per RabbitMQ.
13 gennaio 2022	Amazon MQ presenta il codice di stato `RABBITMQ_MEMORY_ALARM` per informarti quando il tuo broker ha generato un allarme di memoria elevata ed è in uno stato non integro. Amazon MQ fornisce informazioni e consigli dettagliati per consentirti di diagnosticare, risolvere e prevenire allarmi con memoria elevata. Per ulteriori informazioni, consulta gli argomenti seguenti. • the section called “ `RABBITMQ_MEMORY_ALARM` ”
6 gennaio 2022	Quando CloudWatch configuri Logs per i broker Amazon MQ for ActiveMQ, Amazon MQ supporta l'`aws:SourceArn` utilizzo delle chiavi contestuali `aws:SourceAccount` delle condizioni globali nelle policy basate sulle risorse IAM per prevenire il confuso problema del vice. Per ulteriori informazioni, consulta gli argomenti seguenti. • the section called “Prevenzione del confused deputy tra servizi”

Data	Aggiornamento della documentazione
20 dicembre 2021	Amazon MQ per ActiveMQ introduce una serie di nuovi parametri che ti consentono di monitorare il numero massimo di connessioni che puoi effettuare al tuo broker utilizzando protocolli di trasporto supportati diversi, nonché un nuovo permettere aggiuntivo che ti consente di monitorare il numero di nodi collegati al tuo broker in una rete di broker. Per ulteriori informazioni, consulta gli argomenti seguenti. • the section called “Metriche per ActiveMQ”
16 novembre 2021	Amazon MQ per RabbitMQ ora supporta RabbitMQ versione 3.8.23. Per ulteriori informazioni sulle correzioni e funzionalità di questa versione, consultare quanto segue: • Note di rilascio di RabbitMQ 3.8.23 sull'archivio del server RabbitMQ GitHub • Changelog di RabbitMQ Per ulteriori informazioni sulle versioni supportate di Amazon MQ per RabbitMQ e sugli aggiornamenti dei broker, consulta Gestione delle versioni del motore Amazon MQ per RabbitMQ.
12 ottobre 2021	Amazon MQ ora supporta ActiveMQ 5.16.3, una versione secondaria del motore. Per ulteriori informazioni, consulta gli argomenti seguenti: • Pagina di rilascio di ActiveMQ 5.16.3 • Gestione di Amazon MQ per le versioni del motore ActiveMQ • Aggiornamento di una versione del motore del broker Amazon MQ • Utilizzo dei file di configurazione Spring XML

Data	Aggiornamento della documentazione
08 settembre 2021	Amazon MQ per RabbitMQ supporta ora RabbitMQ versione 3.8.22. Questa versione include una correzione per un problema con le code che utilizzano TTL (time to live) per messaggio, identificato nella versione precedentemente supportata, RabbitMQ 3.8.17. Consigliamo di aggiornare i broker esistenti alla versione 3.8.22. Per ulteriori informazioni sulle correzioni e funzionalità di questa versione, consultare quanto segue: • Note di rilascio di RabbitMQ 3.8.22 sul repository del server RabbitMQ GitHub • Changelog di RabbitMQ Per ulteriori informazioni sulle versioni supportate di Amazon MQ per RabbitMQ e sugli aggiornamenti dei broker, consultare Gestione delle versioni del motore Amazon MQ per RabbitMQ
25 agosto 2021	Amazon MQ for RabbitMQ ha temporaneamente disabilitato la versione 3.8.17 del motore RabbitMQ a causa di un problema identificato con le code che utilizzano il protocollo per messaggio (TTL). time-to-live È consigliabile utilizzare la versione 3.8.11.
29 luglio 2021	Amazon MQ per RabbitMQ supporta ora RabbitMQ versione 3.8.17. Per ulteriori informazioni sulle correzioni e funzionalità contenute in questo aggiornamento, consultare quanto segue: • Note di rilascio di GitHub RabbitMQ 3.8.17 nell'archivio del server RabbitMQ • Changelog di RabbitMQ • Gestione delle versioni del motore Amazon MQ per RabbitMQ

Data	Aggiornamento della documentazione
16 luglio 2021	Ora puoi modificare la finestra di manutenzione di un broker Amazon MQ utilizzando Console di gestione AWS AWS CLI, o l'API Amazon MQ. Per ulteriori informazioni sulle finestre di manutenzione dei broker, consultare quanto segue. • Pianificazione della finestra di manutenzione per un broker Amazon MQ
6 luglio 2021	Amazon MQ per RabbitMQ introduce il supporto per il tipo di scambio di hash coerente. Gli scambi hash coerenti instradano i messaggi alle code in base a un valore hash calcolato dalla chiave di routing di un messaggio. Per ulteriori informazioni, consulta gli argomenti seguenti: • Plugin scambio di hash coerente • RabbitMQ Consistent Hash Exchange Type sul repository RabbitMQ GitHub
7 giugno 2021	Amazon MQ ora supporta ActiveMQ 5.16.2, una nuova versione principale del motore. Per ulteriori informazioni, consulta gli argomenti seguenti: • Pagina di rilascio di ActiveMQ 5.16.2 • Gestione di Amazon MQ per le versioni del motore ActiveMQ • Aggiornamento di una versione del motore del broker Amazon MQ • Utilizzo dei file di configurazione Spring XML
26 maggio 2021	Amazon MQ per RabbitMQ è ora disponibile nelle regioni Cina (Pechino) e Cina (Ningxia). Per ulteriori informazioni sulle regioni disponibili, consultare Regioni ed endpoint AWS.
18 maggio 2021	Amazon MQ per RabbitMQ implementa le impostazioni predefinite del broker. Quando crei un broker per la prima volta, Amazon MQ crea una serie di criteri del broker e limiti vhost in base al tipo di istanza e alla modalità di implement azione scelta, al fine di ottimizzare le prestazioni del broker. Per ulteriori informazioni, vedere quanto segue: https://docs.aws.amazon.com//amazon-mq/latest/developer-guide/rabbitmq-defaults.html

Data	Aggiornamento della documentazione
5 maggio 2021	Amazon MQ ora supporta ActiveMQ 5.15.15. Per ulteriori informazioni, consulta gli argomenti seguenti: • Pagina di rilascio di ActiveMQ 5.15.15 • Gestione di Amazon MQ per le versioni del motore ActiveMQ • Utilizzo dei file di configurazione Spring XML
5 maggio 2021	Amazon MQ ha iniziato a tracciare le modifiche alle policy AWS gestite. Per ulteriori informazioni, consulta gli argomenti seguenti: • the section called “AWS politiche gestite”
14 aprile 2021	Amazon MQ è ora disponibile nelle regioni Cina (Pechino) e Cina (Ningxia) . Per ulteriori informazioni sulle regioni disponibili, consultare Regioni ed endpoint AWS.
7 Aprile 2021	Amazon MQ ora supporta RabbitMQ 3.8.11. Per ulteriori informazioni sulle correzioni e funzionalità contenute in questo aggiornamento, consultare quanto segue: • Note di rilascio di RabbitMQ 3.8.11 nell'archivio del server RabbitMQ GitHub • Changelog di RabbitMQ • Gestione delle versioni del motore Amazon MQ per RabbitMQ
1 aprile 2021	Amazon MQ è ora disponibile nella regione Asia Pacifico (Osaka). Per informazioni sulle regioni disponibili, consultare Regioni ed endpoint di Amazon MQ.

Data	Aggiornamento della documentazione
21 dicembre 2020	Amazon MQ ora supporta ActiveMQ 5.15.14. Per ulteriori informazioni, consulta gli argomenti seguenti: • Note di rilascio di ActiveMQ 5.15.14 • Gestione di Amazon MQ per le versioni del motore ActiveMQ • Utilizzo dei file di configurazione Spring XML • **Important** A causa di un problema noto di Apache ActiveMQ in questa versione, il nuovo pulsante Pause Queue (Metti in pausa la coda) nella console Web ActiveMQ non può essere utilizzato con i broker Amazon MQ per ActiveMQ. Per informazioni su questo problema, consultare AMQ-8104.

Data	Aggiornamento della documentazione
04 novembre 2020	Amazon MQ ora supporta RabbitMQ, un popolare broker di messaggis tica open source. Ciò consente di migrare i broker di messaggi esistenti su RabbitMQ senza dover riscrivere il codice. AWS Amazon MQ per RabbitMQ gestisce sia i broker di messaggistica individuali che quelli in cluster e gestisce attività come il provisioning dell'infrastruttura, la configurazione del broker e l'aggiornamento del software. • Amazon MQ supporta RabbitMQ 3.8.6. Per ulteriori informazioni sulle versioni del motore supportate, consultare the section called “Gestione della versione”. • Il Piano gratuito di AWSinclude fino a 750 ore di un broker a singola istanza `mq.t3.micro` e fino a 20 GB di spazio di archiviazione al mese per un anno. Per ulteriori informazioni sui tipi di istanza supportati, consultare Broker instance types. • Con Amazon MQ per RabbitMQ, puoi accedere ai tuoi agenti utilizzando AMQP 0-9-1 e con qualsiasi linguaggio supportato dalle librerie client di RabbitMQ. Per ulteriori informazioni sui protocolli e sulle suite di cifratura supportati, consultare the section called “Protocolli Amazon MQ per RabbitMQ”. • Amazon MQ per RabbitMQ è disponibile in tutte le regioni in cui Amazon MQ è attualmente disponibile. Per ulteriori informazioni su tutte le regioni disponibili, consultare la Tabella delle regioni AWS. Per iniziare a utilizzare Amazon MQ, creare un broker e connettere un'applic azione basata su JVM alil broker RabbitMQ, vedere Guida introduttiva: creazione e connessione a un broker RabbitMQ.
22 ottobre 2020	Amazon MQ supporta ActiveMQ 5.15.13. Per ulteriori informazioni, consulta gli argomenti seguenti: • Note di rilascio di ActiveMQ 5.15.13 • Gestione di Amazon MQ per le versioni del motore ActiveMQ • Utilizzo dei file di configurazione Spring XML

Data	Aggiornamento della documentazione
30 settembre 2020	Amazon MQ è ora disponibile nella regione Europa (Milano). Per informazioni sulle regioni disponibili, consultare Regioni ed endpoint di Amazon MQ.
27 luglio 2020	Puoi autenticare gli utenti Amazon MQ utilizzando le credenziali memorizzate nel tuo Active Directory o in un altro server LDAP. Puoi anche aggiungere, eliminare e modificare gli utenti di Amazon MQ e assegnare autorizzazioni ad argomenti e code. Per ulteriori informazioni, consulta Integrazione di LDAP con ActiveMQ.
17 luglio 2020	Amazon MQ ora supporta il tipo di istanza `mq.t3.micro` . Per ulteriori informazioni, consulta Broker instance types.
30 giugno 2020	Amazon MQ supporta ActiveMQ 5.15.12. Per ulteriori informazioni, consulta gli argomenti seguenti: • Note di rilascio di ActiveMQ 5.15.12 • Gestione di Amazon MQ per le versioni del motore ActiveMQ • Utilizzo dei file di configurazione Spring XML
30 aprile 2020	Amazon MQ supporta un nuovo elemento di raccolta figlio, `systemUsage` , sull'elemento `broker`. Per ulteriori informazioni, consulta systemUsage. Amazon MQ supporta anche tre nuovi attributi sull'elemento figlio `kahaDB`. • `journalDiskSyncInterval` : intervallo (ms) per quando eseguire una sincronizzazione del disco se `journalDiskSyncStrategy=periodic` . • `journalDiskSyncStrategy` : configura la policy di sincronizzazione del disco. • `preallocationStrategy` : configura come il broker cercherà di preallocare i file journal quando è necessario un nuovo file journal. Per ulteriori informazioni, consulta Attributes.

Data	Aggiornamento della documentazione
3 marzo 2020	Amazon MQ supporta due nuove CloudWatch metriche • `TempPercentUsage` : la percentuale di storage temporaneo disponibile utilizzata dai messaggi non persistenti. • `JobSchedulerStorePercentUsage` : la percentuale di spazio su disco utilizzata dall'archivio del sistema di pianificazione delle attività. Per ulteriori informazioni, consulta Monitoring and logging Amazon MQ brokers.
4 febbraio 2020	Amazon MQ è disponibile nelle regioni Asia Pacifico (Hong Kong) e Medio Oriente (Bahrein). Per ulteriori informazioni sulle regioni disponibili, consultare Regioni ed endpoint AWS.
22 gennaio 2020	Amazon MQ supporta ActiveMQ 5.15.10. Per ulteriori informazioni, consulta gli argomenti seguenti: • Note di rilascio di ActiveMQ 5.15.10 • Gestione di Amazon MQ per le versioni del motore ActiveMQ • Utilizzo dei file di configurazione Spring XML
19 dicembre 2019	Amazon MQ è disponibile nelle regioni Europa (Stoccolma) e Sud America (San Paolo). Per ulteriori informazioni sulle regioni disponibili, consultare Regioni ed endpoint AWS.

Data	Aggiornamento della documentazione
16 dicembre 2019	Amazon MQ supporta la creazione di broker ottimizzati per la velocità effettiva utilizzando Amazon Elastic Block Store (EBS), anziché Amazon Elastic File System (Amazon EFS), la soluzione predefinita per l'archiviazione dei broker. Per sfruttare l'elevata durata e la replica in più zone di disponibilità, utilizza Amazon EFS. Per sfruttare la bassa latenza e la velocità effettiva elevata, utilizza Amazon EBS. **Important** • Puoi utilizzare Amazon EBS solo con la famiglia di tipo di istanze del broker `mq.m5`. • Sebbene sia possibile modificare il tipo di istanza del broker, non è possibile modificare il tipo di archiviazione del broker dopo la creazione del broker. • Amazon EBS replica i dati all'interno di una singola zona di disponibilità e non supporta la modalità di implementazione ActiveMQ attiva/in standby. Per ulteriori informazioni, consulta gli argomenti seguenti: • Storage • Scegli il tipo di archiviazione del broker corretto per il miglior throughput • La proprietà `storageType` della risorsa `broker-instance-options` nel Riferimento all'API REST di Amazon MQ • I parametri `BurstBalance`, `VolumeReadOps` e `VolumeWriteOps` nella sezione Monitoring and logging Amazon MQ brokers.
18 ottobre 2019	Sono disponibili due CloudWatch parametri Amazon: `TotalEnqueueCount` e `TotalDequeueCount`. Per ulteriori informazioni, consulta Monitoring and logging Amazon MQ brokers

Data	Aggiornamento della documentazione
11 ottobre 2019	Amazon MQ ora supporta gli endpoint conformi al Federal Information Processing Standard 140-2 (FIPS) nelle aree commerciali degli Stati Uniti. Per ulteriori informazioni, consulta gli argomenti seguenti: • Federal Information Processing Standard (FIPS) 140-2 • Regioni ed endpoint di Amazon MQ
30 settembre 2019	Amazon MQ ora include la possibilità di dimensionare i broker modificando il tipo di istanza host. Per ulteriori informazioni, consulta la proprietà `hostInstanceType` di `UpdateBrokerInput` e la proprietà `pendingHostInstanceType` di `DescribeBrokerOutput`.
30 agosto 2019	Ora puoi aggiornare i gruppi di sicurezza associati a un broker, sia nella console che con `UpdateBrokerInput`.
22 luglio 2019	Amazon MQ si integra con AWS Key Management Service (KMS) per offrire la crittografia lato server. Ora puoi selezionare la tua CMK gestita dal cliente o utilizzare una chiave KMS AWS gestita nel tuo account. AWS KMS Per ulteriori informazioni, consulta Crittografia dei dati a riposo. Amazon MQ supporta l'utilizzo delle AWS KMS chiavi nei seguenti modi. • AWS chiave KMS proprietaria: la chiave è di proprietà di Amazon MQ e non è nel tuo account. • AWS chiave KMS gestita: la chiave KMS AWS gestita (aws/mq) è una chiave KMS nel tuo account che viene creata, gestita e utilizzata per tuo conto da Amazon MQ. • Seleziona le CMK gestite dai clienti esistenti: le CMK gestite dai clienti CMKs vengono create e gestite da te in (KMS). AWS Key Management Service
19 giugno 2019	Amazon MQ è disponibile nelle regioni Europa (Parigi) e Asia Pacifico (Mumbai). Per ulteriori informazioni sulle regioni disponibili, consultare Regioni ed endpoint AWS.

Data	Aggiornamento della documentazione
12 giugno 2019	Amazon MQ è disponibile nella regione Canada (Centrale). Per ulteriori informazioni sulle regioni disponibili, consultare Regioni ed endpoint AWS.
3 giugno 2019	Sono disponibili due nuove CloudWatch metriche Amazon: `EstablishedConnectionsCount` `InactiveDurableSubscribers`. Per ulteriori informazioni, consulta gli argomenti seguenti: • Monitoring and logging Amazon MQ brokers • Monitoring and logging Amazon MQ brokers
10 maggio 2019	L'archiviazione dati per i nuovi tipi di istanza `mq.t2.micro` è limitata a 20 GB. Per ulteriori informazioni, consulta gli argomenti seguenti: • the section called “Storage dei dati” • Broker instance types
29 aprile 2019	Ora puoi utilizzare le policy basate su tag e le autorizzazioni a livello di risorsa. Per ulteriori informazioni, consulta gli argomenti seguenti: • Funzionamento di Amazon MQ con IAM • Autorizzazioni a livello di risorsa supportate per le operazioni API di Amazon MQ
16 aprile 2019	Ora puoi recuperare le informazioni sulle opzioni di istanza broker e sul motore broker utilizzando l'API REST. Per ulteriori informazioni, consulta gli argomenti seguenti: • Opzioni di istanza del broker • Tipi di motore del broker
8 Aprile 2019	Amazon MQ supporta ActiveMQ 5.15.9. Per ulteriori informazioni, consulta gli argomenti seguenti: • Note di rilascio di ActiveMQ 5.15.9 • Gestione di Amazon MQ per le versioni del motore ActiveMQ • Utilizzo dei file di configurazione Spring XML

Data	Aggiornamento della documentazione
4 marzo 2019	È stata migliorata la documentazione per la configurazione del failover dinamico e il ribilanciamento dei client per una rete di broker. Abilita il failover dinamico tramite la configurazione di `transportConnectors` con le opzioni di configurazione `networkConnectors` . Per ulteriori informazioni, consulta gli argomenti seguenti: • Failover dinamico con i connettori di trasporto • Rete di broker Amazon MQ • Amazon MQ Broker Configuration Parameters
27 febbraio 2019	Amazon MQ è disponibile nella regione Europa (Londra) oltre alle seguenti regioni: • Asia Pacifico (Singapore) • Stati Uniti orientali (Ohio) • Stati Uniti orientali (Virginia settentrionale) • Stati Uniti occidentali (California settentrionale) • Stati Uniti occidentali (Oregon) • Asia Pacifico (Tokyo) • Asia Pacifico (Seul) • Asia Pacifico (Sydney) • Europa (Francoforte) • Europa (Irlanda)
24 gennaio 2019	La configurazione predefinita ora include una policy per rimuovere le destinazioni inattive.
17 gennaio 2019	I tipi di istanza `mq.t2.micro` di Amazon MQ ora supportano solo 100 connessioni per protocollo a livello di collegamento. Per ulteriori informazioni, consultare Quotas in Amazon MQ.

Data	Aggiornamento della documentazione
19 dicembre 2018	Puoi configurare una serie di broker Amazon MQ in una rete di broker. Per ulteriori informazioni, consulta le sezioni seguenti: • Rete di broker Amazon MQ • Creating and Configuring a Network of Brokers • Configura la rete di broker nel modo corretto • `networkConnector` • `networkConnectionStartAsincrono`

11 dicembre 2018	Amazon MQ supporta ActiveMQ 5.15.8, 5.15.6 e 5.15.0.
	• Bug risolti e miglioramenti in ActiveMQ:
	  • Note di rilascio di ActiveMQ 5.15.8
	  • Note di rilascio di ActiveMQ 5.15.7

5 dicembre 2018	AWS supporta l'etichettatura delle risorse per monitorare l'allocazione dei costi. È possibile contrassegnare con dei tag le risorse durante la loro creazione oppure visualizzando i dettagli di tale risorsa. Per ulteriori informazioni, consulta l'articolo relativo all'Assegnazione di tag alle risorse.
19 novembre 2018	AWS ha ampliato il suo programma di conformità SOC per includere Amazon MQ come servizio conforme a SOC.
15 ottobre 2018	• Il numero massimo di gruppi per ogni utente è 20. Per ulteriori informazioni, consulta Utenti. • Il numero massimo di connessioni per broker, per protocollo a livello di collegamento è pari a 1.000. Per ulteriori informazioni, consulta Broker.
2 ottobre 2018	AWS ha ampliato il suo programma di conformità HIPAA per includere Amazon MQ come servizio idoneo alla normativa HIPAA.

Data	Aggiornamento della documentazione

27 settembre 2018	Amazon MQ supporta ActiveMQ 5.15.6, oltre a 5.15.0. Per ulteriori informazioni, consulta gli argomenti seguenti:
	• Guida introduttiva: creazione e connessione a un broker ActiveMQ
	• Bug risolti e miglioramenti nella documentazione di ActiveMQ:
	  • Note di rilascio di ActiveMQ 5.15.6
	  • Note di rilascio di ActiveMQ 5.15.5
	  • Note di rilascio di ActiveMQ 5.15.4
	  • Note di rilascio di ActiveMQ 5.15.3
	  • Note di rilascio di ActiveMQ 5.15.2
	  • Note di rilascio di ActiveMQ 5.15.1
	• ActiveMQ Client 5.15.6

31 agosto 2018	• Sono disponibili i seguenti parametri:
	  • `CurrentConnectionsCount`
	  • `TotalConsumerCount`
	  • `TotalProducerCount`
	  Per ulteriori informazioni, consulta la sezione Monitoring and logging Amazon MQ brokers.
	• L'indirizzo IP del broker viene visualizzato nella pagina Details (Dettagli).
	  Note
	  Per broker con accessibilità pubblica disabilitata, viene visualizzato l'indirizzo IP interno.

Data	Aggiornamento della documentazione
30 agosto 2018	Amazon MQ è disponibile nella regione Asia Pacifico (Singapore) oltre alle seguenti regioni: • Stati Uniti orientali (Ohio) • Stati Uniti orientali (Virginia settentrionale) • Stati Uniti occidentali (California settentrionale) • Stati Uniti occidentali (Oregon) • Asia Pacifico (Tokyo) • Asia Pacifico (Seul) • Asia Pacifico (Sydney) • Europa (Francoforte) • Europa (Irlanda)
30 luglio 2018	Puoi configurare Amazon MQ per pubblicare log generali e di audit su Amazon CloudWatch Logs. Per ulteriori informazioni, consulta Monitoring and logging Amazon MQ brokers.
25 luglio 2018	Amazon MQ è disponibile nelle regioni Asia Pacifico (Tokyo) e Asia Pacifico (Seoul) oltre alle seguenti regioni: • Stati Uniti orientali (Ohio) • Stati Uniti orientali (Virginia settentrionale) • Stati Uniti occidentali (California settentrionale) • Stati Uniti occidentali (Oregon) • Asia Pacifico (Sydney) • Europa (Francoforte) • Europa (Irlanda)
19 luglio 2018	Puoi utilizzarlo AWS CloudTrail per registrare le chiamate API di Amazon MQ. Per ulteriori informazioni, consulta Logging Amazon MQ API calls using CloudTrail.

Data	Aggiornamento della documentazione
29 giugno 2018	Oltre a `mq.t2.micro` e `mq.m4.large`, i seguenti tipi di istanza broker sono disponibili per sviluppo, test e carichi di lavoro di produzione regolari che richiedono throughput elevato: • `mq.m5.large` • `mq.m5.xlarge` • `mq.m5.2xlarge` • `mq.m5.4xlarge` Per ulteriori informazioni, consulta Broker instance types.
27 giugno 2018	Amazon MQ è disponibile nella regione Stati Uniti occidentali (California settentrionale) oltre alle seguenti regioni: • Stati Uniti orientali (Ohio) • Stati Uniti orientali (Virginia settentrionale) • Stati Uniti occidentali (Oregon) • Asia Pacifico (Sydney) • Europa (Francoforte) • Europa (Irlanda)

Data	Aggiornamento della documentazione

14 giugno 2018	• Puoi utilizzare la `AWS::Amazon MQ::Broker` AWS CloudFormation risorsa per eseguire le seguenti azioni:
	  • Creare un broker.
	  • Aggiungere modifiche di configurazione o modificare utenti per il broker specificato.
	  • Restituire informazioni sul broker specificato.
	  • Eliminare il broker specificato.
	
	  Note
	  Quando modifichi una proprietà del tipo di proprietà Amazon MQ Broker ConfigurationId o Amazon MQ Broker User, il broker viene riavviato immediatamente.
	• Puoi utilizzare la `AWS::Amazon MQ::Configuration` AWS CloudFormation risorsa per eseguire le seguenti azioni:
	  • Creare una configurazione.
	  • Aggiornare la configurazione specificata.
	  • Restituire informazioni sulla configurazione specificata.
	
	  Note
	  Puoi utilizzarla CloudFormation per modificare, ma non eliminare, una configurazione Amazon MQ.

7 giugno 2018	La console Amazon MQ supporta tedesco, portoghese brasiliano, spagnolo, italiano e cinese tradizionale.
17 maggio 2018	Il limite del numero di utenti per broker è 250. Per ulteriori informazioni, consulta Utenti.
13 marzo 2018	Per creare il broker sono necessari circa 15 minuti. Per ulteriori informazioni, consulta la sezione relativa a come terminare la creazione del broker.

Data	Aggiornamento della documentazione
1 marzo 2018	• Puoi configurare l'archiviazione e l'invio simultanei per Apache KahaDB utilizzando l'attributo `concurrentStoreAndDispatchQueues`. • La `CpuCreditBalance` CloudWatch metrica > è disponibile per il tipo di istanza del broker. `mq.t2.micro`
10 gennaio 2018	Le seguenti modifiche riguardano la console di Amazon MQ: • Nell'elenco di broker, la colonna Creation (Creazione) è nascosta per impostazione predefinita. Per personalizzare la dimensione della pagina e le colonne, scegli  . • Nella ***MyBroker***pagina, nella sezione Connessioni, scegli il nome del tuo gruppo di sicurezza o  apri la console EC2 (anziché la console VPC). La console EC2 consente una configurazione più intuitiva delle regole in entrata e in uscita. Per ulteriori informazioni, consultare la sezione Connecting a Java application to your broker aggiornata.
09 gennaio 2018	• L'autorizzazione per l'ID dell'operazione REST `UpdateBroker` è elencata correttamente come `mq:UpdateBroker` nella console IAM. • L'autorizzazione `mq:DescribeEngine` errata viene rimossa dalla console IAM.

Data	Aggiornamento della documentazione

28 novembre 2017	Corrisponde al rilascio iniziale di Amazon MQ e della Guida per gli sviluppatori di Amazon MQ.
	
	• Amazon MQ è disponibile nelle seguenti regioni:
	  • Stati Uniti orientali (Ohio)
	  • Stati Uniti orientali (Virginia settentrionale)
	  • Stati Uniti occidentali (Oregon)
	  • Asia Pacifico (Sydney)
	  • Europa (Francoforte)
	  • Europa (Irlanda)
	
	  L'uso del tipo di istanza `mq.t2.micro` è soggetto a Crediti CPU e prestazioni di base, con la possibilità di raggiungere un livello superiore (per ulteriori informazioni, consultare il parametro `CpuCreditBalance`). Se la tua applicazione richiede prestazioni fisse, considera l'utilizzo di un tipo di istanza `mq.m5.large`.
	• Puoi creare i broker `mq.m4.large` e `mq.t2.micro`.
	
	  L'uso del tipo di istanza `mq.t2.micro` è soggetto a Crediti CPU e prestazioni di base, con la possibilità di raggiungere un livello superiore (per ulteriori informazioni, consultare il parametro `CpuCreditBalance`). Se la tua applicazione richiede prestazioni fisse, considera l'utilizzo di un tipo di istanza `mq.m5.large`.
	• Puoi utilizzare il motore broker ActiveMQ 5.15.0.
	• Puoi anche creare e gestire broker in modo programmatico utilizzando l'API REST di Amazon MQ e. AWS SDKs
	• Puoi accedere ai broker utilizzando qualsiasi linguaggio di programmazione supportato da ActiveMQ e abilitando TLS esplicitamente per i seguenti protocolli:
	  • AMQP
	  • MQTT
	  • MQTT over WebSocket
	  • OpenWire

Data	Aggiornamento della documentazione
	• STOMP • STOMP over WebSocket • Puoi connetterti ai broker ActiveMQ utilizzando vari client ActiveMQ. È consigliato l'uso del client ActiveMQ. Per ulteriori informazioni, consulta Connecting a Java application to your broker. • Il broker può inviare e ricevere messaggi di qualsiasi dimensione.

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.